2050062

Informatica V° anno

Informatica V° anno

A cura di

CESD s.r.l.

CESD EDITRICE

Tutti i diritti riservati

©CESD s.r.l.

Stampato da:

CESD s.r.l.

Stampato nel 2007

Indice degli argomenti

UNITÀ DIDATTICA 1
RISORSE DI SISTEMA E ORGANIZZAZIONE DEGLI ARCHIVI

1.1 – Gli archivi

Archivi generalità

L'uso degli archivi deriva dalla necessità di conservare dati e informazioni in modo permanente perché potranno essere utili in momenti successivi: questo vale per dati di tipo personale, come quelli contenuti in una normale rubrica telefonica, ma anche per i documenti utili alla vita di uno Stato, di un'azienda o di un Ente.

In un'azienda l'esecuzione delle normali attività sia amministrative che operative, la definizione e la scelta delle politiche commerciali, di quelle finanziarie e di quelle relative al personale, sono strettamente legate all'elaborazione di insiemi di dati che vengono raccolti e conservati in archivi. Archivi dei movimenti contabili, archivi dei clienti e dei fornitori, archivi del personale, archivi di magazzino e riguardanti la produzione: questi sono gli insiemi di dati tipici di un'azienda e il trattamento delle informazioni in essi contenute occupa una parte rilevante dell'attività delle aziende. L'elaborazione di grandi volumi di dati è di grande importanza anche in tutti gli altri settori della società moderna. Basti pensare alle problematiche degli Enti pubblici per la gestione dell'anagrafe tributaria, degli Istituti di credito con l'archivio dei conti correnti, dei liberi professionisti con la possibilità, per esempio da parte di un avvocato, un notaio o un commercialista, di avere a disposizione il testo di tutte le leggi in vigore, oppure dei centri di ricerca scientifica con l'elaborazione di dati sperimentali o degli enti ospedalieri per la conservazione dei risultati di analisi cliniche. Da questi esempi risulta evidente l'impor-tanza dei diversi aspetti della gestione degli archivi di dati, che riguardano strumenti, attività e persone.

Aspetti principali dell'automazione

Gli aspetti principali della gestione automatizzata degli archivi sono rappresentati da:la tipologia dei supporti utilizzati per registrare le informazioni le attrezzature hardware dedicate alla gestione delle unità

di memorizzazione; gli strumenti software per la costruzione di programmi applicativi e dell'interfaccia per l'utente la definizione dell'organizzazione degli archivi. E per rendere efficiente l'accesso ai dati e veloci le operazioni di ritrovamento.

Proprietà

In generale un archivio è un insieme organizzato di informazioni caratterizzate da alcune proprietà fondamentali:

- tra esse esiste un nesso logico (cioè sono in qualche modo inerenti ad un medesimo argomento);
- sono rappresentate secondo un formato che ne rende possibile l'interpretazione;
- sono registrate con un supporto su cui è possibile scrivere e rileggere informazioni anche a distanza di tempo;
- sono organizzate in modo da rendere facile la consultazione.

Consideriamo un esempio di uso molto comune: l'elenco telefonico è un archivio di dati in cui le informazioni riguardano gli abbonati al telefono di una provincia; per ogni abbonato sono riportati nell'ordine generalità, indirizzo, numero di telefono; tutte queste informazioni sono stampate su fogli di carta. Le informazioni vengono raccolte in questo archivio perché si riferiscono agli abbonati di una stessa provincia e all'interno della provincia di uno stesso comune (nesso logico).

Formato delle informazioni

La disposizione delle informazioni nelle righe (formato delle informazioni), nello stesso ordine per tutti gli abbonati, rende facile la lettura e l'interpretazione da parte della persona che consulta l'elenco. Il supporto è costituito dalla carta delle pagine dell'elenco.

Gli abbonati sono stampati seguendo l'ordine alfabetico dei cognomi, all'interno della 5 per comune, per permettere un veloce reperimento del numero di telefono che corrisponde alla persona cercata (organizzazione dei dati).

Nella società moderna, con il crescere del volume delle informazioni e dell'importanza di renderle disponibili in tempi rapidi, la facile reperi-

bilità dei dati è diventata una caratteristica a fondamentale degli archivi e l'uso di archivi gestiti da un calcolatore è nata la prassi nella quasi totalità delle attività organizzate.

1.2 – Operazioni sugli archivi

La gestione di un archivio di dati, di qualunque tipo esso sia, viene realizzata attraverso seguenti operazioni principali:

- la creazione dell'archivio stesso cioè tutto ciò che riguarda la realizzazione, sul supporto di memorizzazione, dello spazio destinato a contenere i dati;
- la consultazione o interrogazione, cioè il reperimento all'interno del-l'archivio delle informazioni necessarie per l'elaborazione desiderata. È ovvio infatti che la possibilità di reperire le informazioni volute è il motivo principale dell'esistenza di archivio. L'archivio della contabilità di un'azienda viene interrogato per conoscere quali le fatture emesse che non sono ancora state pagate dai clienti, archivio anagrafico di un comune può essere consultato per avere un elenco di i bambini che nell'anno in corso raggiungono l'età scolare e devono quindi frequentare la prima classe;

Consultazione degli archivi

- l'inserimento di nuovi dati dopo che l'archivio è stato creato: un esempio di aggiunta di nuovi dati è l'inserimento di un nuovo abbonato nell'elenco telefonico, oppure la memorizzazione, nell'archivio fornitori di un'azienda, dei dati di una ditta è diventata nuova fornitrice;
- la modifica o aggiornamento dei dati già presenti nell'archivio, ma non più corrispondenti ai dati reali: sé un utente telefonico trasferisce la propria residenza, rimanendo nella medesima città, nella successiva edizione dell'elenco telefonico: stampato il nuovo indirizzo. Le operazioni di ingresso o uscita di merci da un magazzino comportano una variazione, nell'archivio di

Operazioni principali per gestire un archivio

magazzino, della giacenza relativa a quella merce;

- la cancellazione di informazioni che non si vogliono più conservare perché non più un nesso logico rispetto alle altre informazioni presenti nell'archivio: è il caso di un abbonato al telefono che si trasferisce in un'altra città e quindi deve essere tolto dall'elenco telefonico o di un articolo che viene cancellato dall'archivio degli articoli di un'azienda di distribuzione perché non più commercializzato;
- l'ordinamento (sort) dei dati secondo un determinato criterio. Questa operazione risponde all'esigenza di un'organizzazione degli archivi tale da consentire una consultazione. Appare ovvio che ricercare un nome in un elenco ordinato alfabeticamente è molto più veloce che eseguire la medesima operazione in un elenco disordinato. Così pure conviene che un archivio di fatture sia ordinato logicamente se si vuole calcolare il fatturato suddiviso per mese;
- la fusione tra due o più archivi, ovvero la realizzazione di un nuovo archivio utilizzando i dati contenuti negli archivi di partenza.

In un archivio le informazioni, in genere, sono raggruppate secondo un'unità logica: nel caso dell'elenco telefonico i dati relativi ad ogni abbonato, in un archivio notarile i dati contenuti nell'incartamento relativo a un cliente.

Organizzazione logica

Questi insiemi di informazioni logicamente organizzate e riferite a un unico soggetto vengono chiamati con il termine record (che in italiano significa registrazione); le singole informazioni che compongono il record si chiamano campi; l'elenco dei campi che lo compongono viene detto tracciato del record.

Per esempio il tracciato del record dell'archivio costituito dall'elenco telefonico può essere descritto con i seguenti campi:

- cognome;

- titolo professionale;
- nome;
- altre descrizioni;
- indirizzo;
- numero di telefono.

La creazione di un archivio richiede la definizione preliminare delle seguenti specifiche:

Creazione

- nome dell'archivio, che lo identifica e serve a ricordarne il contenuto;
- per esempio "archivio fornitori" oppure "archivio anagrafico";
- tracciato record, in altre parole quali informazioni compongono il record;
- supporto da usare per archiviare i dati (fogli di carta, dischi o nastri magnetici, dischi ottici);
- dimensione massima dell'archivio: per esempio il numero massimo di scaffali occupati in un archivio cartaceo o di abbonati in un elenco telefonico;
- il modo con cui i dati sono strutturati e collegati tra loro, cioè l'organizzazione dell'archivio, ci sono diverse possibilità di organizzazione e la scelta del supporto di archiviazione è spesso legata al tipo di organizzazione e alle modalità di consultazione previste. La decisione su queste specifiche fa parte dell'analisi del problema e deve precedere la fase di realizzazione fisica dell'archivio, che può consistere nella sistemazione delle informazioni già esistenti in un archivio ben organizzato oppure nella pre-disposizione del supporto per la registrazione delle nuove informazioni che verranno successivamente inserite.

1.3 – Supporto fisico

Supporti fisici

Il tipo di supporto dove fisicamente sono contenute le informazioni è strettamente legato alle diverse esigenze di utilizzazione delle informazioni stesse e quindi al modo in cui l'archivio viene consultato.
In qualsiasi azienda moderna, anche di piccole dimensioni, è necessario gestire una mole di dati tale da rendere impensabile o comunque molto onerosa sia la memorizzazione su supporti tradizionali, sia ancor più l'elaborazione e la gestione non automatizzata di questi dati.
Per ragioni di velocità nella ricerca e nell'elaborazione dei dati, e di spazio nella loro memorizzazione, si è passati da archivi registrati su supporti cartacei contenuti in armadi (schedari), adatti al reperimento manuale da parte dell'uomo, a supporti ideati per essere trattati in modo automatico dai computer.

Archivi detti file

Gli archivi memorizzati su tali supporti vengono detti file, perché in inglese la parola archivio viene tradotta con file, anche se in informatica questo termine serve a indicare in modo generico qualsiasi informazione che può essere registrata sui supporti di memoria, come un testo, un programma, un comando del sistema operativo.
Nelle applicazioni con il computer un archivio può contenere qualsiasi tipo di informazione non solo di tipo testuale, ma anche di tipo multimediale. Nelle procedure gestionali comunque, nella maggior parte dei casi, gli archivi sono costituiti da insiemi di record omogenei, nel senso che ciascun archivio possiede un tracciato predefinito e uguale per tutti i record in esso contenuti (file di record).
Le apparecchiature, esterne al calcolatore e ad esso collegate, atte a leggere e scrivere le informazioni contenute nei file, vengono dette unità periferiche di memoria o semplicemente periferiche.
I dati destinati ad essere elaborati nei computer devono essere memorizzati in modo opportuno infatti la memoria centrale di un elaboratore elettronico è costituita da componenti, in ognuno dei quali è possibile distinguere con sicurezza due stati distinti per esempio, la conduzione oppure la non conduzione di corrente) a cui sono associati

per conven-zione i valori O e 1. Tali componenti sono quindi elementari e lo stato che essi assumono viene chiamato bit.

È opportuno sottolineare che ogni trasferimento di dati dalla periferica verso la memoria centrale (operazione di input con l'ingresso dei dati in memoria), come pure ogni trasferimento di dati dalla memoria centrale verso la periferica (operazione di output con l'uscita dei dati dalla memoria), riguarda non un singolo carattere, bensì un insieme di caratteri, detto blocco.

Blocco

La dimensione del blocco varia da sistema a sistema e assume valori dell'ordine di alcuni kilobyte quali: 2KB, 4KB, 8KB corrispondenti, rispettivamente, a 2048, 4096, 8192 byte, in quanto 1 KB corrisponde a 210= 1024 byte Per le operazioni di trasferimento tra periferica e memoria centrale, il computer utilizza una particolare zona di lavoro della memoria centrale detta buffer di I/O (input/ output).

Il blocco è l'unità fisica (o record fisico) di memorizzazione dei dati sulla memoria di massa e non deve essere confuso con il record logico definito nel paragrafo precedente: può accadere che un blocco contenga più record logici. In questo modo le operazioni di lettura e scrittura su un file non riguardano singoli record logici, ma gruppi di record: complessivamente diminuisce così il numero di accessi alla periferica, che sono operazioni lente rispetto agli accessi ai dati contenuti in memoria centrale. Per esempio, supponiamo che la lunghezza del record logico di un file sia di 400 caratteri (e quindi 400 byte), e che la dimensione del blocco sul disco sia 2048 byte: il blocco può contenere 5 record, lasciando 48 byte inutilizzati.

Record

Si definisce fattore di blocco di un file il numero di record logici contenuti in un blocco: nel nostro caso il fattore di blocco risulta uguale a 5; un file con fattore di blocco uguale a 1 viene detto a record sbloccati. L'esempio presentato fa riferimento a un file i cui record hanno tutti la medesima lunghezza; questa è la scelta più frequente, poiché permette di semplificare le operazioni di lettura, di scrittura e di controllo. Oltre ai file con record a lunghezza fissa esistono file con

File con record a lunghezza fissa e variabile

record a lunghezza variabile: in essi è necessario specificare la posizione dove termina il singolo record (end of record). Ci sono fondamentalmente due modi per risolvere il problema: il numero di caratteri di ogni record è indicato in un campo di dimensione fissa all'interno del record stesso oppure, in alternativa, ogni record è seguito da una speciale sequenza per indicare la fine del medesimo.
Il DOS utilizza la seconda di queste strategie e impiega la coppia di caratteri ASCII di codice decimale 13 e 10 (Carriage Return e Line Feed), in sostanza una coppia di caratteri con il significato di 'Vai a capo", come marcatura di fine record.
Le informazioni che permettono l'identificazione e il controllo dei file registrati, sono contenute in particolari gruppi di caratteri, detti label. Queste informazioni hanno di solito formati standard e vengono gestiti dal sistema operativo del computer.

Label

Le label più significative sono:

- la label di volume, che è il primo blocco registrato sul nastro, contiene informazioni atte a identificare la bobina, tra cui, per esempio, il nome del proprietario;
- la label di file, che è posta prima dell'inizio di un file, contiene informazioni sul nome e il codice del file, la data di registrazione e il periodo di conservazione del file stesso, oltre al numero d'ordine della bobina, nel caso che il file sia multivolume;
- la label di fine file (in inglese End Of File o EOF) è il blocco contenente il gruppo di caratteri che chiude ogni file; ne esiste uno solo per ogni file, anche per quelli multivolume;
- la label di fine volume (in inglese End Of Volume o EOV) è posta alla fine di un nastro per indicare che è terminata la bobina, ma non il file.

Oltre a questi blocchi informativi, sul nastro vengono registrati blocchi di un solo carattere, detti tape mark (TM), per il controllo del nastro.
Esistono tratti non registrati che separano un blocco dall'altro o una

label dall'altra, detti gap.

L'uso dei nastri come supporto per la memorizzazione delle informazioni, è conveniente per l'alto numero di informazioni immagazzinate con scarso ingombro e per il basso costo delle bobine stesse che sono anche riutilizzabili, anche se lo sfregamento delle testine di lettura e scrittura sulla superficie del nastro ne comporta il deterioramento nel tempo. Svantaggi sono invece le probabili alterazioni della magnetizzazione, se le bobine non vengono conservate in un ambiente adatto, e soprattutto l'accesso obbligatoriamente sequenziale alle informazioni.

Tape mark

Infatti la registrazione o la consultazione dei dati avvengono mentre il nastro, svolgendosi da una bobina e avvolgendosi sull'altra, scorre a velocità costante sotto le testine delle unità a nastri, quindi per reperire un'informazione è necessario rileggere il nastro fino a ritrovare quella che interessa.

Modelli recenti di unità a nastri consentono tuttavia lo scorrimento del nastro in modo bidirezionale: questo significa che non è necessario Ravvolgere il nastro in modo completo per effettuare operazioni di lettura o scrittura, rendendo complessivamente più veloci gli accessi.

1.4 – Il collegamento delle periferiche

Dopo avere esaminato i supporti fisici dei file e le relative periferiche (device), vediamo come i dati in essi contenuti siano trasferiti nella memoria centrale del computer per essere elaborati.

Le varie periferiche prese in esame in precedenza hanno tempi di lavoro molto superiori a quelli della CPU; il collegamento diretto tra periferiche e la CPU comporterebbe quindi periodi inaccettabili di attesa e di inattività dell'unità centrale.

Periferiche

Il colloquio tra unità centrale e periferica avviene attraverso dispositivi di controllo della comunicazione detti interfacce e secondo regole comuni a emittente e ricevente dette protocollo di comunicazione.

Quando la CPU esegue un'applicazione e richiede le prestazioni di una periferica, comunica con essa per ottenere le informazioni sulle caratteristiche fondamentali: per esempio, per un disco, il numero di cilindri, testine e settori. Questo consente di potersi poi riferire al disco, secondo una visione logica, come a un insieme di blocchi sequenziali numerati.

Un'interfaccia è in sostanza una scheda di circuiti elettrici che, inserita nella struttura di un computer, permette di eseguire il trasferimento di informazioni dal computer alle sue periferiche e viceversa: la scheda di interfaccia è connessa al bus mediante opportuni connettori sulla piastra madre (motherboard) del computer; il collegamento con la periferica è realizzato attraverso le porte di I/O presenti sulla scheda e accessibili dall'esterno.

Interfaccia

La porta può essere seriale, quando i caratteri vengono trasmessi un bit per volta, o parallela, quando i bit del carattere vengono trasmessi contemporaneamente. Per poter connettere apparecchiature periferiche di produttori diversi, il collegamento deve rispettare gli standard fissati a livello internazionale: seguendo questa esigenza, negli anni recenti si è affermato un nuovo standard dì interfaccia seriale denominato USB (Universal Serial Bus). Questa tecnologia di connessione presenta alcuni vantaggi: possibilità di connettere fino a 127 dispositivi in sequenza sulla stessa porta USB, alta velocità di trasferimento, connessione e configurazione rapida di nuove periferiche anche con computer funzionante.

USB

I dispositivi hardware della periferica, per servire adeguatamente le richieste dell'unità centrale, devono compiere operazioni fisiche per cercare i dati contenuti nelle tracce e attendere che il settore passi sotto la testina per l'operazione di I/O.

Le unità periferiche, come dischi e stampanti, non possono essere controllate da una struttura integrata nell'unità centrale, in quanto hanno caratteristiche molto diverse da essa in termini di velocità e modalità operative. Negli elaboratori attuali si tende a separare

nettamente l'unità centrale dalle unità di controllo delle periferiche (controller).

Controller

In generale, un controller è un dispositivo capace di pilotare una periferica, attraverso l'elaborazione di funzioni logiche, secondo un programma scritto in un linguaggio relativamente semplice e riconducibile alle funzioni svolte dall'apparecchiatura da controllare.

L'unità centrale invia alle periferiche solo comandi di alto livello, indipendenti dal particolare dispositivo che lo dovrà eseguire: la successione di controlli richiesta dai dispositivi per espletare la loro operazione è trasparente all'utente e all'unità centrale.

1.5 – Operazioni sui file

Sui file possono essere eseguite le seguenti operazioni fondamentali:

l'apertura di un archivio stabilisce un collegamento tra la memoria centrale e il file registrato sulla memoria di massa, riservando un buffer per le operazioni di I/O e individuando nella tabella dei descrittori di file le informazioni necessarie per accedere ai blocchi fisici del file.

Il comando di apertura deve essere eseguito prima di iniziare qualsiasi operazione di lettura o scrittura.

L'operazione di lettura copia in memoria centrale dalla memoria di massa il contenuto dei campi di un record registrato nel file.

L'operazione di scrittura trasferisce sulla memoria di massa il contenuto del record composto in memoria centrale con i valori assegnati ai campi.

Il posizionamento individua il record sul quale si deve leggere o scrivere oppure a partire dal quale si deve iniziare una lettura o scrittura sequenziale.

L'operazione di riscrittura aggiorna nel file su memoria di massa il contenuto di un record modificato durante l'elaborazione.

La cancellazione di un record elimina le informazioni che non servono

più nelle applicazioni utilizzate. L'operazione di cancellazione va effettuata con molta cautela: per questo di solito, prima di cancellare, si legge il record, si visualizza il suo contenuto e si chiede all'utente la conferma della cancellazione.

La chiusura del file interrompe il collegamento tra memoria centrale e file, liberando la memoria riservata per le operazioni di I/O e aggiornando le informazioni sul file nella tabella dei descrittori. Perciò è opportuno chiudere un archivio quando non sono previste ulteriori operazioni di lettura e scrittura.

Lo schema seguente riassume le operazioni possibili a seconda delle diverse organizzazioni assegnate ai file.

L'operazione di rewind (riavvolgimento) è l'operazione che posiziona il file all'inizio in modo che una successiva lettura causi il trasferimento del primo record del file.

In alcuni sistemi, oltre all'operazione di rewind, sono possibili altre operazioni di posizionamento anche per i file sequenziali.

Si noti che la riscrittura e la cancellazione sono operazioni di scrittura e vengono di solito precedute da un'operazione di lettura per facilitare il controllo da parte dell'utente.

1.6 – Le applicazioni gestionali

Le applicazioni informatiche, che riguardano problemi gestionali nelle aziende, negli studi professionali o negli Enti pubblici, sono basate sull'uso di dati organizzati in archivi. I dati devono essere raccolti, registrati sui supporti di memoria, per metterli poi a disposizione degli utenti con procedure di interrogazione o di stampa di report. Gli archivi sono di solito: anagrafiche, movimenti, parametri.

Procedure

Le procedure per la gestione degli archivi comprendono da una parte programmi per la creazione degli archivi e la loro manutenzione nel tempo, e dall'altra programmi per ritrovare i dati che servono attraverso l'interrogazione degli archivi. I dati che servono possono

essere ottenuti con operazioni di interrogazione che presentano i dati con visualizzazioni sul monitor o stampe su carta.

Dati

~~Le stampe sono costituite da elenchi su carta normale, come nelle liste alfabetiche oppure da moduli prestampati attraverso la stampa dei dati in posizioni predefinite del foglio, come nelle bollette telefoniche.~~

Con il termine basi di dati (in inglese database) si indicano in informatica gli archivi di dati, organizzati in modo integrato attraverso tecniche di modellazione dei dati e gestiti sulle memorie di massa dei computer attraverso appositi software, con l'obiettivo di raggiungere una grande efficienza nel trattamento e nel ritrovamento dei dati, superando anche i limiti presenti nelle organizzazioni tradizionali degli archivi.

Le problematiche relative alle basi di dati nella storia dell'informatica risalgono al 1970 e il passaggio dalla teoria all'applicazione sui computer con l'uso di software efficiente ha prodotto, nel corso degli anni successivi, strumenti software per aumentare la produttività nell'elaborazione di informazioni, soprattutto nei casi che riguardano la gestione di grandi quantità di dati.

Problematiche

La gestione delle basi di dati si è poi consolidata anche nelle applicazioni per i personal computer. A grandi linee, possiamo dire che il database è una collezione di archivi di dati ben organizzative ben strutturati, in modo che possano costituire una base di lavoro per utenti con programmi diversi. Per esempio i dati relativi agli articoli del magazzino di un'azienda possono essere utilizzati dal programma che stampa le fatture, oppure dal programma che stampa i listini di magazzino. Quando si parla di efficienza e di produttività di un'organizzazione di archivi si intende naturalmente la possibilità di ritrovare facilmente le informazioni desiderate, anche attraverso criteri di ricerca diversi, in termini di velocità nell'elaborazione, di sicurezza dei dati e integrità delle registrazioni, specialmente quando la gestione si riferisce a una mole di dati rilevante.

Efficienza

Deve essere garantita la consistenza degli archivi, cioè i dati in essi

contenuti devono essere significativi ed essere effettivamente utilizzabili nelle applicazioni dell'azienda.

Consistenza

I dati devono quindi essere protetti per impedire perdite accidentali dovute a cadute del sistema, guasti hardware o interventi dannosi da parte di utenti e di programmi; la protezione deve riguardare anche gli interventi dolosi sui dati, dovuti ad accessi non autorizzati con operazioni di lettura, modifica o cancellazione.

In sostanza sicurezza significa impedire che il database venga danneggiato da interventi accidentali o non autorizzati; integrità significa garantire che le operazioni effettuate sul database da utenti autorizzati non provochino una perdita di consistenza ai dati.

Sicurezza

I prodotti software per la gestione di database vengono indicati con il termine DBMS (DataBase Management System). Si tenga presente anche la differenza tra database, come insieme di dati e DBMS come sistema per la gestione del database, così come nell'uso degli archivi tradizionali c'è una distinzione tra archivi di dati e file system.

DBMS

1.7 – I limiti dell'organizzazione convenzionale degli archivi

Le tecniche di gestione delle basi di dati nascono per superare i problemi e i limiti insiti nelle tradizionali organizzazioni degli archivi in modo non integrato.

Tecniche di gestione

Limiti e problemi che sono evidenziati dal seguente esempio giocattolo. Il termine giocattolo precisa la natura dell'esempio: si tratta di un esempio collocato in un mondo ideale nel quale sono possibili cose impossibili nella vita reale come, per esempio, identificare una persona con il solo cognome.

Risulta tuttavia efficace per presentare in modo semplice i concetti e per facilitare l'apprendimento.

Si vogliono gestire le informazioni dei clienti di un'azienda bancaria e per questo è stata sviluppata una semplice applicazione.

Per ciascun conto vengono gestiti: il numero del conto, il nome

dell'intestatario, il suo indirizzo con il codice di avviamento postale e il saldo.

I progettisti hanno deciso di organizzare le informazioni su due file:

- il primo con le informazioni anagrafiche del cliente integrate con il numero di conto;
- il secondo abbina al numero di conto il relativo saldo.

Nell'approccio tradizionale le informazioni vengono organizzate in archivi i cui record hanno i seguenti tracciati.

Questa organizzazione delle informazioni permette di gestire con semplicità i casi di clienti che condividono i conti, come si vede nel caso di Gialli e Verdi che condividono il conto 5100.

Per la gestione dei dati sono stati sviluppati programmi, scritti in un linguaggio di programmazione tradizionale (per esempio Cobol), che prevedono l'immissione dei dati anagrafici, il versamento su conto, il prelievo dal conto, l'aggiornamento delle anagrafiche e la produzione di un rapporto contenente l'elenco dei clienti, ordinato alfabeticamente, con il saldo.

Il tradizionale approccio basato sugli archivi (file based) favorisce una serie di inconvenienti individuabili nel seguente elenco:

- dipendenza dai dati. I programmi sono dipendenti dagli archivi che essi gestiscono, nel senso che tutti i linguaggi di programmazione (per esempio il Cobol o il C) richiedono di specificare, all'interno del programma, quali sono gli archivi utilizzati e la struttura dei loro record: questo significa che qualsiasi modifica alla struttura del record, richiede la modifica di tutti i program-mi che utilizzano quel tipo di record. L'accesso ai dati è determinato dal tipo di organizzazione degli archivi, dalle chiavi stabilite per i record e dall'ordine con cui i campi compaiono nella struttura del record; l'organizzazione scelta vincola il programmatore nell'uso delle operazioni che si possono effettuare sugli archivi;

Dipendenza dei dati

- interrogazioni predefinite e difficoltà nell'accesso ai dati. È possibile accedere ai dati solo tramite le applicazioni specifiche, cioè tramite un numero limitato di interrogazioni predefinite. Per ogni altra esigenza informativa bisogna sviluppare applicazioni ad hoc;

Interrogazioni

- isolamento dei dati e file di differente formato. I dati sono dispersi tra molti file e in differenti formati a causa, per esempio, dell'uso di differenti linguaggi nello sviluppo di diverse parti di un'applicazione: di conseguenza diventa difficile collegare i dati tra di loro e integrarli. Ridondanza e inconsistenza dei dati. La ridondanza nei dati è molto frequente nell'approccio tradizionale basato su file indipendenti. La duplicazione dei dati è accompagnata dalla necessità di inserire gli stessi dati più volte, con l'aumento dello spazio occupato dai dati, ma, soprattutto, la ridondanza può portare all'incongruenza, nel caso in cui un dato venga aggiornato in un archivio e non in un altro, oppure siano presenti valori diversi per lo stesso dato.

Isolamento

1.8 – Organizzazione degli archivi mediante basi di dati

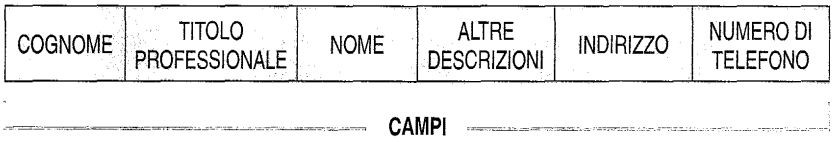
Una base di dati non è solo una qualsiasi collezione di informazioni che esistono per un lungo periodo di tempo sul disco, ma è una collezione di informazioni accomunate dal fatto che esse forniscono la descrizione di uno specifico aspetto del mondo reale. Possiamo definire una base di dati come segue:

il database è una collezione di dati logicamente correlati e condivisi, che ha lo scopo di soddisfare i fabbisogni informativi di una specifica organizzazione. In un archivio le informazioni, in genere, sono raggruppate secondo un'unità logica.

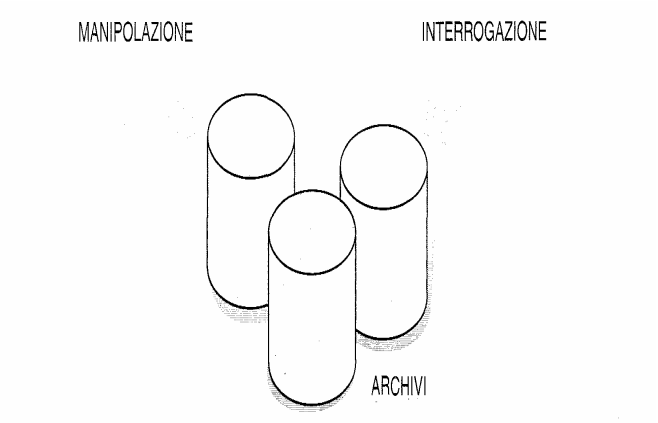
Questi insiemi di informazioni logicamente organizzate e riferite ad un unico soggetto vengono chiamati con il termine record; le singole informazioni che compongono il record si chiamano campi; l'elenco

Record

dei campi viene detto tracciato del record.



La creazione di un archivio richiede la definizione preliminare delle seguenti specifiche: nome dell'archivio.
Il tracciato record, in altre parole quali informazioni compongono il record supporto da usare per archiviare i dat. Dimensione massima dell'archivio.
Prima di essere usato, un archivio deve essere creato.



La gestione di un archivio di dati, di qualunque tipo esso sia, viene realizzata attraverso le seguenti operazioni:

- operazioni di manipolazione: l'inserimento, la modifica o aggiornamento;
- la cancellazione operazioni di interrogazione: consultazione o interrogazione, visualizzazione, stampa.

I dati, congiuntamente con la loro descrizione, sono gestiti da un unico sistema, chiamato DBMS (DataBase Management System), che ne permette la gestione e ne regola gli accessi.

DBMS

Un DBMS deve essere in grado di: Permettere la creazione di una nuova base di dati, definendo gli archivi che la compongono, la loro articolazione, le correlazioni logiche tra archivi, i limiti nell'accesso ai

dati e i vincoli imposti alla loro manipolazione.

La creazione della base dati avviene mediante un apposito linguaggio che prende il nome di linguaggio di definizione dei dati, ovvero DDL (Data Definition Language).

DDL

Facilitare gli utenti nell'inserimento, nella cancellazione e variazione dei dati nel database sfruttando uno specifico linguaggio che prende il nome di linguaggio di manipolazione dei dati, detto DML, acronimo di Data Manipulation Language.

Rendere possibile l'estrazione di informazioni dal database interrogando la base dati mediante un apposito linguaggio di interrogazione, QL (Query Language).

QL

Inoltre un DBMS deve risolvere i problemi che si presentano con l'approccio tradizionale e la gestione degli archivi deve avere le seguenti caratteristiche fondamentali:

- facilità di accesso: il ritrovamento dei dati è facilitato e svolto con grande velocità, anche nel caso di database molto grandi e con richieste provenienti contemporaneamente da più utenti;

Facilità di accesso

- indipendenza dalla struttura logica dei dati: i programmi applicativi sono indipendenti dalla struttura logica con cui i dati sono organizzati negli archivi; quindi è possibile apportare modifiche alla definizione delle strutture della base di dati senza modificare il software applicativo;
- indipendenza dalla struttura fisica dei dati: i programmi applicativi sono indipendenti dai dati fisici, cioè è possibile modificare i supporti con cui i dati sono registrati e le modalità di accesso alle memoria di massa senza modifiche alle applicazioni;
- eliminazione della ridondanza: gli stessi dati non compaiono più volte in archivi diversi, in quanto il Database è costituito da archivi integrati di dati;

Eliminazione di ridondanza e inconsistenza

- eliminazione della inconsistenza: il database non può presentare campi uguali con valori diversi in archivi diversi;

- integrità dei dati: vengono previsti controlli per evitare anomalie ai dati causate dai programmi e dalle applicazioni degli utenti; le operazioni sui dati richieste dagli utenti vengono eseguite fino al loro completamento per assicurare la consistenza;

Integrità

- utilizzo da parte di più utenti: i dati organizzati in un unico database possono essere utilizzati da più utenti con i loro programmi, consentendo anche una visione solo parziale del database da parte del singolo utente, che può rimanere estraneo al resto dei contenuti nel database;
- controllo della concorrenza: il DBMS garantisce che le operazioni svolte da utenti diversi in modo concorrente non interferiscano una con l'altra;
- sicurezza dei dati: sono previste procedure di controllo sia per impedire accessi non autorizzati ai dati contenuti nel database, sia per la protezione da guasti accidentali.

1.9 – I modelli per il database

I database è un modello della realtà considerata: i contenuti della base di dati rappresentano gli stati in cui si trova la realtà da modellare.

I cambiamenti che vengono apportati alla base di dati rappresentano gli eventi che avvengono nell'ambiente in cui opera l'azienda.

L'uso efficace dei dati organizzati in un database presuppone un attento lavoro di progettazione iniziale, che viene fatto con riferimento ai dati che si vogliono memorizzare e successivamente elaborare.

Progettazione

Il progetto è indipendente dal computer, dai supporti fisici destinati a contenere le informazioni e dalle caratteristiche del DBMS.

Questo è un elemento di novità rispetto all'organizzazione convenzionale degli archivi per la quale il lavoro e la programmazione erano molto legate alle caratteristiche delle risorse dell'hardware e del linguaggio a disposizione. Questo modo di operare consente alla base

di dati di evolvere nel tempo insieme alla realtà aziendale per la quale è stata progettata, con il crescere delle esigenze degli utenti e della necessità di informazioni. È in questo contesto che deve essere inquadrata la precedente definizione di database nella quale si parla di archivi logicamente correlati.

Nel rappresentare gli aspetti significativi della porzione di mondo reale che si vuole modellare devono essere identificate, mediante opportuni schemi, le entità di interesse, la loro articolazione e le correlazioni tra le entità individuate.

Il modello costruito è indipendente dalle modalità con le quali la basi di dati verrà realizzata e ne rappresenta il modello concettuale.

Costruzione del modello

Tra i numerosi modelli proposti per la progettazione concettuale, quello maggiormente diffuso è il modello Entità/Associazioni.

Il modello Entità/Associazioni, indicato come modello E/R dal termine inglese Entity/Relationship, è il modello adottato in questo testo per la progettazione concettuale e sarà sviluppato nell'Unità Didattica successiva. Nella costruzione del modello E/R di una realtà si individuano gli oggetti che la compongono dette entità, gli attributi, che rappresentano le caratteristiche delle entità individuate, e infine le associazioni che individuano le correlazioni logiche tra entità.

Modello E/R

1.10 – Archivi classici

Per svolgere una qualsiasi attività gestionale, amministrativa, o statistica è necessario utilizzare grandi quantità di dati ovvero archiviare informazioni. La modalità di archiviazione condiziona fortemente la consultazione e l'uso delle informazioni tanto da rendere indispensabile uno studio approfondito, fatto a priori, per la scelta dell'organizzazione più idonea per l'archivio.

In Informatica un Archivio può essere considerato come un insieme di file, ciascuno dei quali contiene informazioni tra loro omogenee e relative ad una certa classe di oggetti. Un *file* è quindi un insieme di

Archivio – file

registrazioni o record, ciascuno dei quali descrive le proprietà specifiche del singolo oggetto, ed è individuato tramite un indirizzo logico che corrisponde alla posizione che la registra. Ciascuna registrazione a sua volta è costituita da un insieme prefissato di informazioni dette campi, ciascuna delle quali individua un preciso attributo dell'oggetto. Tra i campi di solito ci sarà un campo privilegiato, detto chiave primaria, che ha la proprietà di individuare univocamente il record; i campi che non hanno questa proprietà vengono detti chiavi secondarie. Ogni campo è comunque caratterizzato da un nome, che permette di fare riferimento ad una proprietà comune agli oggetti in questione e da un valore che specifica la proprietà del singolo oggetto astrazione occupa nel file.

Struttura dati – Archivio

Un archivio può essere considerato come una struttura dati astratta sulla quale è possibile definire un insieme di operazioni. Anche se non esiste una standardizzazione sul tipo e sul numero di tali operazioni, è importante almeno ricordare quali sono le operazioni più frequenti che compongono i programmi applicativi:

- creazione archivio: sulla base dell'organizzazione scelta, in fase di progettazione, si creano tutti i sottoarchivi (files) previsti dai programmi applicativi;
- inserimento dati in un archivio già esistente: questa opera-zione deve permettere l'aggiunta di un qualsiasi elemento in un qualsiasi sottoarchivio e le eventuali correlazioni tra le informazioni inserite;
- cancellazione di una registrazione con chiave primaria K: questa operazione prevede innanzi tutto la ricerca dell'informazione, poi in base alla tecnica adottata per l'archiviazione, la cancellazione. Anche qui bisogna prevedere gli aggiornamenti da effettuare sulle informazioni correlate alla chiave K cancellata;
- visita di un archivio: in base all'organizzazione data, questa operazione può essere sequenziale e non, ovvero una visita può essere compiuta sia in un ordine qualsiasi che secondo una certa

logica;

- ordinamento di un archivio: si definisce ordine fisico la sequenza in cui sono disposte le singole registrazioni sul supporto di Memoria di Massa, mentre si parla di ordine logico facendo riferimento alla sequenza dei valori crescenti e decrescenti della chiave primaria;
- copia di un archivio: operazione che viene effettuata periodicamente per il riaggiornamento e anche per avere copie di sicurezza delle parti fondamentali di un archivio.

1.11 – Archivi sequenziali

Per archivi sequenziali si intende un'organizzazione di dati su memoria di massa in cui le registrazioni possono essere pensate come disposte in blocchi contigui. Essi sono caratterizzati dalle seguenti proprietà:

1. per ogni record del file, escluso uno, è definita un'operazione di accesso ad un altro record designato come successivo;
2. non esistono strutture dati ausiliarie associate al file che possano agevolare le operazioni di ricerca.

Su essi l'accesso può essere sequenziale o diretto e ciò dipende dal supporto di memorizzazione a disposizione, rispettivamente nastri magnetici o dischi. Tra le operazioni previste su un archivio sequenziale bisogna ricordare le modalità per i nuovi inserimenti.

Se l'archivio è ordinato i nuovi inserimenti vengono effettuati su un file appositamente creato e, periodicamente, si riorganizza l'archivio con una fusione (***merge***). Se l'archivio non è ordinato i nuovi inserimenti vengono fatti o nelle posizioni dei record cancellati, o in coda al file.

L'aggiornamento di un record, per l'accesso diretto, avviene semplicemente attraverso una riscrittura del record.

Per la cancellazione, il record viene marcato con un apposito simbolo ed in fase di riorganizzazione, quel record non viene copiato, si effettua cioè una cancellazione logica e non fisica. forniti dagli alimenti ed i reali fabbisogni dell'organismo

1.12 – Organizzazione a lista

In molte applicazioni risulta particolarmente importante poter mantenere costantemente aggiornato ed ordinato un archivio, così con l'organizzazione a liste è possibile assegnare agli elementi un ordinamento fisico arbitrario, garantendo comunque la sequenzialità logica tra gli stessi.

Si costruisce così una lista definita come una successione di elementi che occupano in memoria posizioni qualsiasi, in modo però che ciascuno di essi sia legato al successivo mediante un puntatore.

Di seguito è schematizzato un generico archivio in cui le informazioni sono memorizzate sequenzialmente ma sono anche ordinate in ordine alfabetico, rispetto alla chiave primaria, logicamente:

- nel record ***0*** del file è memorizzato nel campo ***Punt***, l'indirizzo logico della chiave più piccola;
- nel campo ***Punt*** di ogni record è memorizzato l'indirizzo della successiva chiave in ordine alfabetico;
- il campo ***Punt*** dell'ultimo record conterrà–***1*** (o altro valore rappresentante il ***null***).

Per inserire un nuovo elemento con chiave K, si occupa un record vuoto, quindi si scorre la lista partendo dall'indirizzo memorizzato nel campo Punt del record 0, utilizzando due puntatori, uno al record da esaminare e uno al precedente: quando si incontra una chiave ***Ki > K*** si registra nel campo Punt del record da inserire, l'indirizzo di Ki e nel record precedente, l'indirizzo logico del record inserito.

Per recuperare gli spazi resi vuoti dalle cancellazioni, è possibile

creare una lista dei cancellati, così da rendere possibile l'inserimento delle informazioni negli spazi resi disponibili dai record cancellati.

Organizzazione sequenziale con indici

Le organizzazioni *sequenziali* e *sequenziali con indice* rispondono in modo adeguato a buona parte delle problematiche legate alla gestione di grandi masse di dati ma non risultano efficienti nei casi in cui gli archivi sono dinamici e di frequente consultazione. L'organizzazione in grado di rispondere con maggiore efficienza viene detta ad *accesso diretto* perché si basa sull'idea di ricavare direttamente, tramite una funzione di trasformazione, dal valore della chiave primaria l'indirizzo della registrazione ad essa associata. La funzione viene implementata tramite un algoritmo di randomizzazione e il metodo di accesso viene indicato come ***indirizzamento hash***. Negli archivi casuali il problema fondamentale è quello di definire una funzione che associ ad ogni record logico l'indirizzo logico in cui memorizzarlo, per cui occorre realizzare, mediante un algoritmo, un'associazione chiave-indirizzo tale che ad ogni valore **K** della chiave faccia corrispondere un indirizzo **h(K)**. Ciò implica che la chiave, che fornisce gli attributi necessari per individuare logica-mente un certo record, contiene anche tutte le informazioni atte a reperire la posizione in cui la registrazione è memorizzata. La situazione ideale si otterrebbe con una funzione che, ad ogni valore ipotetico della chiave, faccia corrispondere un indirizzo distinto, in modo che ogni elemento possa essere raggiunto con un unico accesso (*trasformazione perfetta)*. Ciò comporterebbe, tra l'altro, che il nu-mero di record che compongono l'archivio fisico sia pari al numero di tutti i possibili valori assegnabili alla chiave. Poiché nella realtà il numero di chiavi reali è di gran lunga più basso di quello delle chiavi ipotetiche, un'organizzazione di questo tipo porterebbe ad un inaccettabile spreco di memoria.

Organizzazioni sequenziali

Algoritmo di randomizzazione

Per questo motivo si preferisce optare per algoritmi di conversione che, pur non risultando trasformazioni perfette, non si discostino molto

dal modello reale

Scelta dell'algoritmo di randomizzazione

La scelta dell'algoritmo di calcolo risulta cruciale e richiede un'attenta considerazione dei seguenti aspetti:

- l'algoritmo usato per la trasformazione deve produrre lo stesso indirizzo a partire dalla medesima chiave;
- se per la memorizzazione dell'archivio è stata riservata una certa area compresa, ad esempio, tra l'indirizzo i_1 ei_n, la trasformazione di una qualsiasi chiave ***K*** deve produrre sempre un indirizzo compreso tra i_1 **e** i_n;
- l'algoritmo deve essere il più semplice possibile, in modo da non appesantire le operazioni di conversione di chiave e di abbassare il tempo di accesso;
- la trasformazione deve coprire l'intero intervallo degli indirizzi, poiché, se l'algoritmo non genera mai un indirizzo ***i***, lo spazio di memoria ad esso associato non verrà mai occupato;
- occorrerebbe evitare che due o più chiavi vengano trasformate dall'algoritmo di randomizzazione nello stesso indirizzo (***collisione***). Poiché ciò non si può evitare, occorre mettere a punto procedure particolari per poter allocare le varie registrazioni (***sinonimi***) ad indirizzi distinti

Metodi di randomizzazione

Dato c delle collisioni, la ricerca non deve tendere verso un algoritmo perfetto, ma verso la realizzazione di un algoritmo di randomizzazione efficiente. In generale la conversione della chiave in indirizzo si articola nei seguenti passi:

- se la chiave non è numerica occorre *convertirla* in un valore numerico (chiave numerica), cercando di non perdere informazioni nella trasformazione;
- si *applica* alla chiave trasformata l'algoritmo di randomiz-

zazione scelto;

- si *opera* sul numero ottenuto per adattare questo valore al *range* di indirizzi di memoria effettivamente disponibile.

La scelta del miglior metodo di randomizzazione è ovviamente legata alla tipologia della chiave, per cui è buona norma, una volta individuati i metodi idonei, procedere ad una simulazione della distribuzione e delle intensità dei sinonimi per verificare quale sia quello più adatto. Purtroppo solo in casi molto particolari è possibile trovare un algoritmo che non generi collisioni (***trasformazione perfetta***) ed è quindi inevitabile trovarsi nella necessità di gestire il problema connesso con i sinonimi. In ogni caso la gestione dei sinonimi risulta più semplice se nella progettazione si tengono presenti le seguenti regole generali:

- scegliere l'algoritmo di randomizzazione più idoneo ed una tecnica di caricamento opportuna che, come abbiamo visto, è praticamente impossibile eludere il problema Fare in modo che i record richiamati più spesso siano memorizzati in posizione *home*, cioè rappresentino l'elemento di testa delle liste di sinonimi;
- prestare particolare attenzione alla scelta di un metodo efficiente per la memorizzazione e il reperimento dei sinonimi che si sono generati;
- prevedere una riorganizzazione periodica dell'archivio Data Base.

La diffusione che si è avuta dei mezzi informatici, dovuta agli straordinari progressi conseguiti sia nell'hardware che nel software, ha indotto sia le grosse che le piccole organizzazioni (aziende, istituzioni, ecc.) ad avvalersi dei Sistemi informatici, per evitare le lentezze e i rischi connessi all'esecuzione manuale delle procedure per l'elaborazione, per la conservazione e la comunicazione di informazioni utili e

necessarie alla vita delle organizzazioni di cui è permeata la nostra società Il *sistema informatico* dunque oggi assolve ad un ruolo irrinunciabile rispetto ai *bisogni informativi* delle organizzazioni.

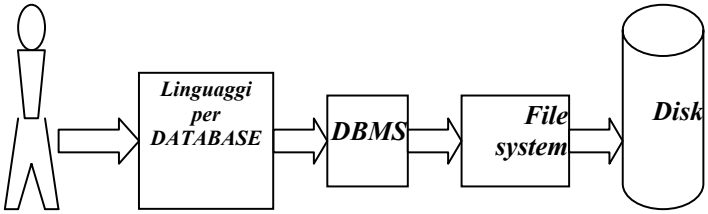
Esso rappresenta *il sottoinsieme del sistema informativo dedicato alla gestione automatica di informazioni* e fa uso di risorse materiali di tipo informatico (elaboratori, periferiche varie, dischi, nastri, ecc.). una vera e propria svolta nell'organizzazione e nella gestione dei sistemi informatici è venuta dall'avvento dei ***DATA BASE*** Un *Data* Base può essere definito come un insieme di informazioni strettamente correlate, memorizzate su un supporto di memoria di massa, costituenti un tutt'uno, che possono essere manipolate da più programmi applicativi.

Il sistema preposto alla gestione dei dati, che provvede, sulla base delle specifiche dell'utente, alla loro generazione, ricerca e aggiornamento, prende il nome di ***Data Base Management System*** (DBMS).

Tra Data Base e DBMS esiste una forte interazione, per cui spesso si tende a considerarli due parti di un unico oggetto: il DBMS rappresenta la parte attiva, il Data Base quella passiva, sulla quale il DBMS opera.

Il DBMS si differenzia da un file system per numerosi aspetti:

- Filesystem è parte integrante del S.O.. Un DBMS utilizza il file system per gestire le registrazioni in modo trasparente per l'utente con il quale vi è possibilità di interazione diretta;
- le capacità di un File System si limitano alla gestione di dati non volatili e alla loro organizzazione fisica. Un DBMS permette un accesso efficiente ai dati, attraverso il mantenimento di un'astrazione di essi tramite un modello. Inoltre il DBMS si preoccupa di gestire le transazioni, fornendo un accesso corretto e concorrente al Data Base e di controllare la congruità e la sicurezza del sistema.


Linguaggi
per
DATABASE
DBMS
File
system
Disk

UNITÀ DIDATTICA 2
LO SVILUPPO DEL PROGETTO INFORMATICO

2.1 – Il progetto

Lo sviluppo di un progetto è un sistema complesso di attività con l'obiettivo di ottenere un risultato che chiameremo prodotto. In un progetto compaiono l'attività di studio, di ideazione, l'attività di progettazione, l'attività di realizzazione e l'attività di produzione.

Sinteticamente introduciamo il significato delle attività presenti in un progetto. Per conoscere come sviluppare un progetto possiamo pensare a un esempio nel campo immobiliare. Inizialmente essi dovranno conoscere quali sono le richieste in termini di obiettivi e finalità.

Conclusa la fase di studio, inizia la fase di ideazione e progettazione. Il risultato della fase di progettazione porterà alla realizzazione.

Il prodotto di un progetto informatico è l'insieme dei moduli software e di archivi atti a soddisfare gli obiettivi definiti

In informatica la base del controllo di qualità di processo è l'insieme delle attività (in parte svolte in rigorosa successione cronologica e in parte svolte in parallelo) e delle loro relazioni rappresentate in un modello chiamato ciclo di vita del software che, di tatto, definisce una metodologia per la progettazione, lo sviluppo e la manutenzione del software. La metodologia racchiude le attività in fasi e ne descrive gli obiettivi e i prodotti da ottenere.

Sviluppo

Abbiamo visto il controllo di qualità di prodotto, il controllo di qualità di processo e la qualità totale validi per qualunque attività produttiva. Ora applichiamo questi concetti nel campo informatico.

Per applicare il controllo di qualità di prodotto nei progetti informatici si parte con la scelta delle caratteristiche da misurare. È più facile definire le caratteristiche di un prodotto materiale e tangibile quale per esempio, un'auto, un elettrodomestico, una casa, un

vestito. è decisamente più difficile identificare le caratteristiche di un prodotto intangibile o di un servizio. La difficoltà aumenta anche quando dobbiamo determinare l'unità di misura e il relativo intervallo di qualità.

Nei progetti informatici possiamo dichiarare le seguenti principali caratteristiche, così come sono enunciate nella norma ISO 9126, riguardante i fattori di qualità del software e la loro misurazione:

- funzionalità le procedure software coincidono con i requisiti prefissati e non ci sono attività ripetitive o inutili;
- affidabilità, i programmi gestiscono anche le situazioni particolari, cioè non si bloccano per motivi banali;
- usabilità, riutilizzo semplice per gli utenti;
- efficienza, le elaborazioni sorto veloci e occupano poche risorse;
- manutenibilità, la documentazione è adeguata e consente di intervenire celermente con modifiche ai programmi;
- portabilità, è possibile trasferire il software su altre piattaforme hardware o in altri sistemi operativi

2.2 – La metodologia

Abbiamo visto come applicare il controllo di qualità di prodotto nell'ambito informatica Ora vediamo come applicare il controllo di qualità di processo e come sia fondamentale adottare un processo che può essere indicato come ciclo di vita del software. Adottare una metodologia significa organizzare il lavoro.

La metodologia è un approccio strutturato per l'esecuzione dell'intera attività di sviluppo dei progetti e dei relativi sistemi, coerentemente con gli obiettivi e le necessità da soddisfare.

Programmi e dati

Un sistema informatico, normalmente, è un insieme di programmi, archivi, apparecchiature informatiche e reti telematiche atto a risolvere le esigenze per cui è stato progettato.

I sistemi gestiscono in modo organico e integrato le funzioni e le informazioni di competenza dell'area in cui operano, al fine di evitare dispersioni di energie e duplicazioni di dati.

L'organizzazione del lavoro di sviluppo dì un progetto informatico assicura, oltre a un elevato grado di qualità, anche un risparmio di risorse.

Le principali risorse, che beneficiano per il fatto di operare in un ambiente organizzato, cioè definito, pianificato e coordinato, sono il tempo e le persone. È evidente che serve un minor tempo e un minore numero di persone se ognuno sa esattamente quando e cosa deve fare garantire la riuscita di un progetto informatico dipende quindi molto dalla struttura e dalla sequenza dei processi previsti per lo sviluppo, oltre che dall'impegno e dalle capacità delle persone che vi operano.

La metodologia indica al progettista il metodo da seguire per sviluppare il progetto. La costruzione di un progetto informatico si ottiene tramite l'esecuzione di diverse attività, caratterizzate da obiettivi ben precisi da raggiungere.

La metodologia identifica queste attività e la organizza per ottimizzare il lavoro. Suddividiamo quindi la metodologia in sezioni che chiamiamo fasi o passi sequenza.

Le fasi di progettazione e di transizione, come pure le fasi di documentazione, di prove (testing) di formazione sono eseguibili in parallelo, cioè in contemporanea, in quanto non necessitano l'una dei risultati dell'altra Questa impostazione è chiamata metodologia in cascata. Altre metodologie prevedono la riesecuzione o riciclo delle fasi con l'intento di raffinare, in modo crescente, i. risultati di ogni fase (metodologia aspirale).

Per concludere citiamo l'approccio prototipale; esso punta direttamente alla fase di realizzazione e il prodotto si avvicina progressivamente alla versione definitiva con successivi interventi migliorativi.

L'approccio prototipale è utilizzabile solo in caso di progetti di dimensioni molto ridotte. Ogni fase è costituita dalle attività da svolgere, dai risultati da raggiungere e dai controlli da eseguire: le attività descrivono dettagliatamente che cosa deve essere fatto: i risultati da raggiungere sono il prodotto della fase; i controlli da eseguire consentono la verifica dell'andamento.

La metodologia, quindi, fornisce la struttura e la sequenza delle attività da svolgere. Questo non è sufficiente: infatti servono riferimenti di carattere temporale, cioè occorre indicare il tempo previsto per io svolgimento di ogni attività.

Tempificazione

La tempificazione serve per la pianificazione delle risorse umane.

È ovvio che ciò rende possibile la stima dei costi dei progetto.

La prima fase del processo di sviluppo, detta anche dei requisiti, riguarda la conoscenza degli obiettivi. Conoscere ciò che si dovrà sviluppare è il requisito di successo indispensabile per operare nella stessa direzione.

Per esempio, non è possibile progettare una autovettura senza conoscere le finalità per cui vuole lanciare sui mercato.

Potremmo progettare una autovettura a benzina, diesel o elettrica; possiamo decidere tra la carrozzeria a due volumi, monovolume o station wagon. Magari la richiesta è quella di commercializzare autovetture economiche, quindi il progetto deve dimensionare le caratteristiche dell'auto per contenete i costi. Questi ed altri aspetti, esterni al prevedibile meccanico ingegneristico, che, se trascurati, porterebbero il fallimento dei progetto.

2.3 – La qualità per i prodotti software

In informatica la base del controllo di qualità di processo è l'insieme delle attività (in parte svolte in rigorosa successione cronologica e in parte svolte in parallelo) e delle loro relazioni rappresentate in un modello chiamato ciclo di vita del software che di fatto, defini-

sce una metodologia per la progettazione, lo sviluppo e la manutenzione dei software.
La metodologia racchiude le attività in fasi e ne descrive gli obiettivi e i prodotti da ottenere.

Controllo

Abbiamo visto il controllo di qualità di prodotto, il controllo di qualità di processo e a qualità totale validi per qualunque attività produttiva. Ora applichiamo questi concetti nel campo informatico.
Per applicare il controllo di qualità di prodotto nei progetti informatici si parte con la scelta delle caratteristiche da misurare. È più facile definire le caratteristiche di un prodotto materiale e tangibile quale, per esempio, un'auto, un elettrodomestico, una casa, un vestito. è decisamente più difficile identificare la caratteristiche di un prodotto intangibile o di un servizio.
La difficoltà aumenta anche quando dobbiamo determinare l'unità di misura e il relativo intervallo di qualità.
Nei progetti informatici possiamo dichiarare le seguenti principali caratteristiche, così come sono enunciate nella norma ISO 9125, riguardante i fattori di qualità del software e la loro misurazione:

- funzionalità, le procedure software coincidono con i requi-siti prefissati e non ci sono attività ripetitive o inutili;
- affidabilità, i programmi gestiscono anche le situazioni particolari, cioè non si bloccano per motivi banali;
- usabilità, l'utilizzo è semplice per gli utenti;
- efficienza, le elaborazioni sono veloci e occupano poche risorse;
- manutenibilità: la documentazione è adeguata e consente di intervenire celermente con modifiche ai programmi;
- portabilità, è possibile trasferire il software su altre piattaforme hardware o in altri sistemi operativi.

Le caratteristiche principali sono ulteriormente dettagliate in sotto-caratteristiche, che specificano le proprietà che il software poi deve

possedere: alcune di queste sono esterne, cioè orientate all'utente finale dei software, altre sono interne, cioè orientate alle attività di sviluppo.

2.4 – La conoscenza degli obiettivi

Conoscenza degli obiettivi

La prima fase del processo di sviluppo, detta analisi dei requisiti, riguarda la conoscenza degli obiettivi. Conoscere ciò che si dovrà sviluppare è il fattore di successo indispensabile per operare nella giusta direzione. Riprendendo l'esempio sopra accennato, non è possibile progettare una buona autovetture senza conoscere le finalità per cui la si vuole lanciare sul mercato.

Potremmo progettare un'autovettura a benzina, diesel o elettrica; possiamo decidere tra la carrozzeria a due volumi, tre volumi, monovolume o station wagon magari la richiesta è quella di commercializzare autovetture economiche, quindi il progetto deve dimensionare le caratteristiche dell'auto per contenere i costi.

Questi esempi mostrano aspetti, esterni al prevedibile contesto meccanico ingegneristico, che, se trascurati, comporterebbero il fallimento del progetto.

Al pari dell'esempio precedente, anche nell'ambito di progetti informatici emerge evidente la necessità dei progettista di approfondire la conoscenza dei fattori rilevanti per un corretto risultato, oltre a possedere la necessaria professionalità informatica.

Contemporaneamente alla necessità di conoscere gli obiettivi da raggiungere e il sistema da costruire) occorre la conoscenza globale dell'ambito di progetto (il sistema esistente).

Ciò si ottiene tramite l'indagine e l'approfondimento della situazione esistente.

Per proseguire con l'esempio dell'autovettura è importante sapere la situazione attuale del mercato delle auto, estendendo la conoscenza anche per altre marche.

Se volessimo vendere in America dovremmo progettare l'auto con guida a destra. Un'automobile pensata per le strade europee deve essere di lunghezza e di larghezza inferiore a quella pensata per il mercato americano in quanto la situazione esistente delle vie di comunicazione è molto diversa.

Il mercato americano privilegia il cambio automatico al contrario del mercato europeo. Queste caratteristiche della situazione esistente devono essere conosciute per non incorrere in grossolani errori.

La rilevazione della situazione esistente, nell'ambito di un progetto informatico, si preoccupa di individuare in quale realtà di Sistema informativo si deve integrare il nuovo progetto.

Dobbiamo quindi conoscere quali siano gli archivi informatizzati già presenti nel sistema per evitare duplicazioni di dati che sarebbero causa di possibili, anzi probabili, discordanza delle informazioni.

Per esempio, se esistesse già un archivio della contabilità aziendale o un archivio dei clienti non sarebbe opportuno crearne altri simili, sia per motivi di omogeneità e completezza sia per motivi di integrazione di clienti di un'azienda devono essere registrati in un unico archivio che garantisce l'unicità della visione e degli aggiornamenti.

In caso contrario saremmo costretti a eseguire un lavoro aggiuntive che elabori tutti gli archivi della contabilità per raccordarli in termini di codici di conto, in, modo da ottenere una contabilità generale unificata dell'azienda, in presenza dell'archivio clienti duplicato, nel caso per esempio di variazione dell'indirizzo di un cliente, saremmo costretti a eseguire gli aggiornamenti su più archivi con l'evidente aumento dei costi e con il rischio dì disallineamento dei dati. Inoltre in un progetto informatico occorre conoscere le fasi di lavoro degli altri progetti per integrare organicamente il nuovo progetto nei complesso dei progetti esistenti.

Questa fase anche chiamata analisi preliminare. Un elemento importante da evidenziare e da tenere costantemente in evidenza, in questa fase e nelle successive, infatti le soluzioni di progetto

ipotizzabili devono rispettare i vincoli che l'ambiente esterno ci impone.

Vincoli

Un esempio di vincolo informatico è costituito dai dati e dai programmi esistenti e con i quali il nuovo prodotto deve colloquiare. Un altro esempio di vincolo è il tipo di architettura hardware e di software di sistema su cui dovrà operare il progetto.

Conoscenza

I vincoli e la conoscenza di quanto si vuole ottenere, compongono lo scenario dal quale il progettista attinge per lo sviluppo del progetto.

Analisi dei requisiti

Occorre osservare che l'analisi preliminare deve tenere conto di altri importanti aspetti riguardanti l'uso del software:

1. definizione della specifiche di sicurezza per la protezione dei dati da accessi non autorizzati;
2. definizione delle specifiche di riservatezza per la protezione dei dati sensibili;
3. definizione delle specifiche di ergonomia. E di usabilità dei software per favorire il lavoro degli utenti finali.
4. il risultato di questa prima fase è la documentazione dell'area dei progetto e una serie di ipotesi di prodotto:
5. la documentazione del progetto è quanto ottenuto dalla ricerca e dallo studio della materia oggetto dei progetto;
6. l'ipotesi di prodotto è il frutto dell'attività di ideazione che consentirà di concretizzare l'idea dei prodotto.

Per ottenere la conoscenza necessaria è indispensabile poter disporre di esperti della materia ai quali affidare l'intervista conoscitiva. Siamo in presenza di un'importante attività di ricerca e di studio; essa deve portare il progettista ad essere l'esporto della materia di pertinenza dei progetto.

L'intervista è una tecnica, utilizzata nelle sviluppo di progetti informatici, che ha come obiettivo quello di conoscere, comprendere e documentare la materia oggetto dei progetto: si basa sui metodo dell'indagine conoscitiva, di analisi, o di progettazione architet-

turale. Ha l'obiettivo di determinare e di descrivere tutte le componenti dei progetto. È la fase di maggiore importanza: il risultato di questo lavoro diventa la base e la guida di tutto il progetto.

La documentazione delle componenti diventa il consolidamento della conoscenza dei problema. Ciò consente la verifica della comprensione degli obiettivi, la divulgazione delta conoscenza sia per io sviluppo delle fasi successive, sia per la futura manutenzione dell'applicazione informatica.

Fasi di lavoro

Questa fase del processo di sviluppo tratta la definizione di tre aspetti fondamentali:

1. dati, cioè le informazioni che caratterizzano il progetto;
2. funzioni, cioè le funzionalità richieste al progetto;
3. flusso del dati, cioè le modalità di acquisizione e di uscita dei dati input e output.

In un progetto informatico una componente fondamentale è costituita dai dati.

L'attività di determinazione dei dati consiste nel decidere quali sono le informazioni necessarie per la riuscita dei progetto.

Nella fase precedente, chiamata conoscenza degli obiettivi, vengono già individuati i dati più importanti, in questa fase di analisi si devono individuare tutti i dati gestiti dal progetto.

L'attività di descrizione dei dati serve a documentare la risorsa informativa di competenza del progetto.

L'altra componente fondamentale di un progetto informatico è costituita dalle funzioni.

L'attività di determinazione delle funzioni consiste nei decidere quali siano le cose che il sistema dovrà fare.

Le funzioni più importanti vengono già individuate nella precedente fase di conoscenza degli obiettivi.

Analisi preliminare
Documentazione

Nella fase di analisi, invece, è necessario scompone e dettagliarle sino al livello che consenta di documentare tutte le attività che

dovranno essere soddisfatte da! progetto: questa attività si chiama scomposizione funzionale.

La rappresentazione grafica della gerarchia delle funzioni che fanno parte di un progetto prende il nome di funzionigramma.

Si tratta di una struttura ad albero: ogni nodo contiene la descrizione sintetica di una funzione.

Macrofunzioni

Le funzioni che descrivono attività complesse (macrofunzioni) vengono suddivise in funzioni di dettaglio che indicano con maggiore precisione le operazioni da compiere per realizzare la macrofunzione.

Abbiamo visto le due componenti basilari di un sistema informatico:

- i dati dei progetto, che saranno memorizzati in archivi elettronici;
- le funzioni del progetto, che saranno risolte con i pro-grammi.

La terza componente da conoscere e documentare nella fase di analisi è costituita dal flusso dei dati tra le funzioni.

Flusso di dati

Documentare il flusso dei dati tra le funzioni significa abbinare a ogni funzione, quali dati utilizza in input e quali dati produce in output. Abbiamo visto le due componenti basilari di un sistema informatico: i dati dei progetto, che saranno memorizzati in archivi elettronici e le funzioni del progetto, che saranno risolte con i programmi.

Intervista

Consideriamo per esempio una funzione per il censimento dei dipendenti che si compone di tre funzioni.

Realizzazione

La fase della realizzazione genera il prodotto dei progetto. Dopo lo studio, l'ideazione e la concretizzazione della scelte di progetto, si arriva alla sua realizzazione.

Moduli software-programmi

Trattandosi di un progetto informatico il prodotto è costituito da moduli software (chiamati comunemente programmi) e dagli archivi utilizzati dai programmi stessi.

I programmi sviluppati devono essere provati, già in questa fase, per garantire la qualità dei software.

L'obiettivo di questo tipo di prova è la verifica del singolo modulo software ed è eseguita dalla stessa persona che ha sviluppato il modulo.

Queste prove sono diverse da quelle che saranno eseguite dall'utente nella successiva fase dì prove (system test) infatti l'obiettivo di queste ultime diventa la verifica globale del progetto. Il prodotto della documentazione è, normalmente, composto dai manuale per l'utente.

System test

Il manuale deve avere la caratteristica, molto importante, di descrivere completamente tutte le funzionalità dei progetto con un opportuno indice, per cui l'operatore che volesse eseguire una funzione dei proprio lavoro, troverà facilmente il modulo software dei progetto che deve attivare. La descrizione deve essere completa e nello stesso tempo deve essere di facile consultazione e di dimensioni ridotte. Si pensi a manuali di centinaia di pagine; l'autore dei manuale ha un impegno enorme nella stesura e anche nei successivi aggiornamenti; inoltre l'utente riscontra difficoltà nel leggere tante pagine per poter operare. in aggiunta al manuale per l'utente, che abbiamo definito completo, sintetico e focalizzato sulle funzioni, è molto utile pensare a una documentazione interattiva inglobata nelle interfacce video. Questo manuale elettronico in linea help in linea fornisce il significato dei campi previsti nell'interfaccia utilizzando le defini-zioni registrate nella fase dl analisi, In altre parole funziona a richiesta: per esempio, quando l'utente chiede di vedere la descri-zione dei campo su cui è posizionato, premendo il tasto di help (tipicamente il tasto F1) appare a video un testo con la descrizione del dato. Questo secondo tipo di manuale è focalizzato sui dati, al contrario dei manuale per l'utente che è focalizzato sulle funzioni, in un ambiente multimediale sì possono pensare ausili per l'utente di tipo visivo e vocale.

Caratteristiche manuale

Test

1. Completa le seguenti frasi scegliendo tra le parole elencate alla fine della domanda:

   L'attività di ……. consolida l'astrazione del prodotto formalizzando tramite disegni, modelli e prototipi.

   L'attività di …… crea l'astrazione del prodotto e ne definisce le caratteristiche.

   L'attività di …….. studia l'area di competenza del progetto per conoscere approfonditamente la materia.

   L'attività di ……. termina il progetto e inizia la produzione.

   L'attività di ……. crea realmente i primi esemplari del prodotto.

2. Quale è l'obiettivo della metodologia?
   a) La qualità del software di progetto.
   b) Un'organizzazione del lavoro tale da dare il prodotto desiderato.
   c) Il modello di qualità generale.
   d) La struttura degli archivi elettronici.

3. Quali sono gli aspetti fondamentali trattati nella fase di analisi?
   a) Obiettivi, strumenti, tecniche.
   b) Intervista,modello, metodologia.
   c) Dati, archivi, programmi.
   d) Dati, funzioni,flusso dei dati.

UNITÀ DIDATTICA 3

MODELLAZIONE DEI DATI

3.1 – Introduzione e progettazione concettuale, logica e fisica

Nella precedente Unità didattica abbiamo descritto le tecniche per raccogliere le richieste degli utenti di un'applicazione informatica.

Il passo successivo consiste nello sviluppare la base di dati dell'applicazione. Sviluppo che passa attraverso diverse fasi di progettazione dette progettazione concettuale, logica e fisica.

La progettazione concettuale è la sintesi tra la visione degli utenti e la visione dei progettisti dell'applicazione. Deve possedere due caratteristiche antitetiche: da una parte deve essere assolutamente precisa per non lasciare dubbi in merito alla caratteristiche della base di dati che si sta progettando, dall'altro deve essere espressa tramite formalismi sufficientemente semplici da permetterne la lettura e la comprensione anche da parte di utenti non tecnici. Gli utenti devono infatti essere certi che i progettisti abbiano compreso a fondo tutte le loro esigenze.

Modello Entità/Associazioni

Il modello Entità/Associazioni presenta tali caratteristiche e si concretizza in un documento con rappresentazioni grafiche, che verranno spiegate nelle unità didattiche successive.

Lo scopo di questa Unità didattica è di descrivere i concetti e i formalismi utilizzati nella costruzione del modello entità/associazioni.

Il modello entità/associazioni, pur essendo di gran lunga il modello più utilizzato nella progettazione concettuale, non ha una rappresentazione standardizzata.

Quando si parla di modello entità/associazioni tutti sono concordi nel ritenere che nel modello devono comparire entità, associazioni e attributi, e che cosa si intenda con tali termini: esistono però diversi modi di rappresentarli in un modello.

In questa dispensa abbiamo optato per l'uso di una forma di rappre-

sentazione grafica basata sull'Unified Modelling Language (UML).

UML

L'UML, linguaggio unificato di modellazione, è un linguaggio grafico per visualizzare, specificare, costruire e documentare tutte le costruzioni di sistemi software.

L'UML è stato approvato da parte del comitato OMG (Object Management Group) e si sta affermando come linguaggio standard nella progettazione di sistemi software object oriented.

Nel presentare il modello entità/associazioni indicheremo anche quali sono gli altri formalismi maggiormente diffusi. Occorre osservare che il modello entità/associazioni è da molti indicato con il nome di modello entità/relazioni.

Modello Entità /Associazioni

Il termine relazione è utilizzato, in tal caso, come sinonimo di associazione, ossia per indicare un collegamento logico tra entità differenti. Ma il termine relazione è importante nell'ambito del modello relazionale dei dati, dove assume un significato completamente diverso.

Per evitare ambiguità, in questo libro, sarà utilizzato sempre il termine entità/associazioni per indicare questo tipo di modello.

Modellare i dati significa costruire una rappresentazione semplificata della realtà osservata o di un problema aziendale, individuandone gli elementi caratterizzanti e i legami intercorrenti tra essi.

La progettazione di un modello di dati avviene a livelli diversi:

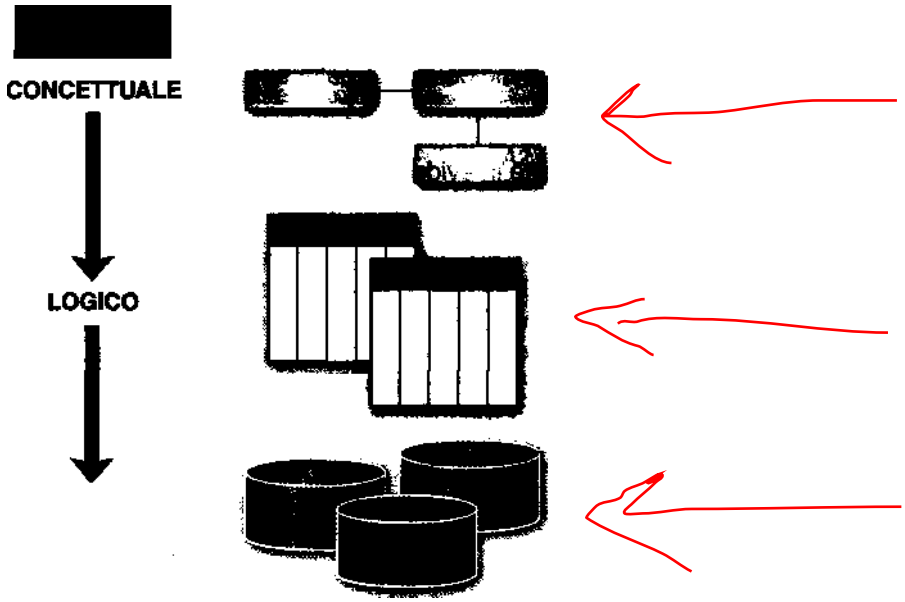
Livelli di progettazione

- livello concettuale (o esterno) rappresenta la realtà dei dati e le relazioni tra si attraverso uno schema;
- livello logico rappresenta il modo attraverso il quale i dati sono organizzati negli archivi elettronici: descrive quindi la composizione e il formato dei dati nel loro detto di struttura logica di dati.

Il livello logico viene derivato dal livello concettuale applicando alcune semplici regole di trasformazione. livello fisico rappresenta l'effettiva installazione degli archivi elettronici: esso indica l'ubica-

zione dei dati nelle memorie di massa (dischi).

Il livello fisico è quindi l'implementazione del livello logico sui supporti per la registrazione fisica dei dati: partizioni, puntatori, blocchi fisici, cluster, indici.



L'attività di progettazione consente prima di tutto di costruire una rappresentazione astratta della realtà in modo indipendente dalla struttura dei dati.

Modello concettuale

Il modello concettuale viene definito attraverso lo schema dei dati, cioè una rappresentazione sintetica (di solito presentata in forma grafica) degli elementi fondamentali caratterizzano la realtà osservata.

Questa rappresentazione è indipendente da:

- i valori che verranno assegnati ai dati;
- le applicazioni degli utenti che utilizzano i dati;
- le visioni parziali dei dati da parte degli utenti.

Il modello concettuale rappresenta un patrimonio importante per le aziende, poiché scrive i dati esistenti in azienda: il suo valore informativo può essere utilizzato sia nel campo informatico sia nell'ambito gestionale e diventa un supporto per i diversi aziendali.

Passaggio dal modello concettuale al modello logico

Con il passaggio al modello logico, l'insieme dei dati viene dotato di una struttura che deve facilitare:

- la manipolazione o il trattamento dei dati, cioè la possibilità di inserire, modificare e cancellare i dati;
- l'interrogazione, cioè la possibilità di ritrovare i dati, richiesti da un'applicazione, in modo semplice e veloce.

Queste strutture di dati vengono poi implementate sulle memorie di massa, realizzando in pratica il modello fisico, rappresentato dai file registrati nei blocchi del disco.

Riassumendo, a partire dalla realtà considerata, vengono individuati i dati che sono significativi, nel senso che sono caratterizzanti, e viene definito lo schema concettuale rappresentativo della realtà.

Le strutture logiche dei dati, derivate dallo schema concettuale, vengono implementate e memorizzate negli archivi su un supporto fisico di registrazione (memorie di massa).

II modello entità/associazioni (in inglese Entity/Relationship), introdotto nel 1976 matematico Peter P. Chen, è uno strumento per analizzare le caratteristiche di una realtà in modo indipendente dagli eventi che in essa accadono, cioè per costruire un modello concettuale dei dati indipendente dalle applicazioni. Il risultato di questo lavoro è la definizione di una rappresentazione grafica, detta schema E/R (Entity/Relationship), che mette in evidenza gli aspetti fondamentali del modello concettuale, con i dati caratterizzanti e le associazioni tra essi. Esso diventa uno strumento molto utile nel realizzare la transizione dalla descrizione di un problema allo schema formale degli archivi. Il modello descrive lo schema concettuale di un problema o di una gestione aziendale e non si occupa dell'efficienza delle operazioni di manipolazione e ritrovamento dei dati sugli archivi fisici. Risulta di facile comprensione anche per persone che non si occupano di computer ed è sostenuto da alcuni concetti e regole che lo rendono preciso e

E/R

rigoroso. Gli elementi di un modello entità/associazioni sono:

- entità;
- associazioni;
- attributi.

Entità

L'entità è un oggetto (concreto o astratto) che ha un significato anche quando viene considerato in modo isolato ed è di interesse per la realtà che si vuole modellare movimento contabile, una prova soste-nuta da uno studente. Le entità possono essere classificate secondo un certo criterio di omogeneità definendo il tipo di entità attraverso un nome. Per esempio gli studenti di una scuola sono classificabili nel tipo entità Studente, i diversi modelli di automobile sono classificabili nel tipo entità Automobile. Ciascuno studente rappresenta un'istanza dell'entità Studente.

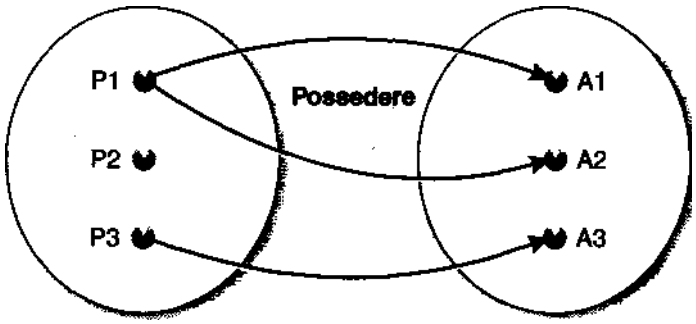
3.2 – L'associazione

Associazione

L'associazione è un legame che stabilisce un'interazione tra le entità.

Nella figura sottostante sono mostrate alcune persone, un certo numero di automobile un insieme di archi per indicare l'associazione di possesso che si viene a stabilire tra le persone e le automobili.

L'associazione ha nome Possedere e ha un verso che è specificato tramite le frecce che collegano l'entità Persona con l'entità Auto-mobile.



In figura si vede che P1 possiede le automobili A1 e A2, P2 non possiede alcuna automobile mentre P3 possiede l'automobile A3.

Descrizione

Descriviamo questa situazione in linguaggio naturale con le seguenti frasi: una persona può possedere una o più automobili; un'automobile è posseduta da una sola persona. Possiamo dire che tra l'entità Persona e l'entità Automobile esiste l'associazione Possedere.

La rappresentazione grafica convenzionalmente usata per indicare un'associazione a linea che unisce le due entità interessate; il nome dell'associazione compare linea con il simbolo della punta di una freccia per indicare il senso di lettura della associazione.

Di norma i nomi delle entità sono sostantivi mentre i nomi delle associazioni sono in questo modo si cerca di stabilire una corrispondenza tra rappresentazione associazioni e frasi del linguaggio naturale che le descrivono.

Le caratteristiche di ogni attributo sono:

Caratteristiche

- il formato;
- la dimensione;
- l'opzionalità.

Il formato di un attributo indica il tipo di valori che assume; i tre formati di base sono:

Formato

- carattere;
- numerico;
- data/ora.

La dimensione indica la quantità massima di caratteri o cifre inseribili. L'opzionalità indica la possibilità di non essere sempre valorizzato:

Dimensione

Opzionalità

- l'attributo è obbligatorio se deve avere valore non nullo (per esempio il nome di una persona in un'anagrafica);
- facoltativo se sono accettabili valori nulli (per esempio il titolo di studio di una persona).

Il valore nullo, in inglese Null, (da non confondere con la stringa di caratteri blank o con un numero di valore zero) rappresenta un'informazione mancante in quanto inapplicabile (per esempio il numero del certificato elettorale di un ragazzo con meno di 18 anni) o sconosciuta (la data effettiva di consegna di una mercé ordinata e non ancora arrivata in magazzino).

Attributo

I diversi valori assunti dagli attributi determinano le diverse istanze dell'entità. L'insieme dei possibili valori assunti da un attributo si chiama dominio dell'attributo.

I valori appartenenti al dominio sono omogenei tra loro, cioè sono dello stesso tipo.

Gli attributi sono elencati nella parte inferiore del rettangolo che rappresenta l'entità, con una linea di demarcazione tra il nome dell'entità e la Vista di attributi. La figura rappresenta l'entità Automobile con i relativi attributi.

Le associazioni possono avere attributi. Si consideri per esempio l'associazione di nome Acquistare che associa una persona all'automobile acquistata.

L'acquisto avviene in una certa data (DataAcquisto) e con un certo prezzo (PrezzoAcquisto). Il prezzo di acquisto non è un attributo di Automobile (un automobile è caratterizzata da un prezzo di listino, ma potrebbe avere un prezzo di acquisto differente per ogni vettura venduta). Analogamente non è un attributo della persona.

Un ragionamento analogo si può fare per l'attributo DataAcquisto.

Si può dire che i due attributi sono caratteristici dell'abbinamento tra una persona e l'automobile acquistata, ossia dell'associazione Acquistare tra le entità Persona e Automobile.

Rappresentazione UML

La rappresentazione UML di questa situazione è la seguente: nella rappresentazione UML gli attributi dell'associazione compaiono nella parte inferiore di un riquadro identico a quello usato per rappresentare le entità, collegato alla linea che rappresenta l'associazione.

Il nome dell'associazione è riportato sulla linea che collega le entità e lo spazio, che nel caso delle entità è occupato dal nome, rimane vuoto.

È importante sottolineare che la differente visione del problema non porta a una corretta in un caso ed errata nell'altro. Non solo, come vedremo di apprendimento, applicando le regole di passaggio dal modello concettuale al modello logico, le due rappresentazioni portano alle stesse strutture.

Regola

Una regola molto importante richiede di definire solo gli attributi elementari e quindi gli attributi che si ottengono con le elaborazioni, cioè gli attributi derivati. Il mancato rispetto della regola provoca inefficienza dovuta alle elaborazioni necessarie per aggiornare questo tipo di attributi. Per esempio l'età di una persona è un attributo derivato dall'attributo elementare della data di nascita; il saldo di un conto corrente è derivato dalla somma algebrica degli importi dei movimenti effettuati sul conto.

Primary Key

Si indica con il termine chiave o chiave primaria un insieme minimale di attributi che permettono di distinguere tra loro le istanze di una stessa entità. Esempi di chiavi sono il codice di un prodotto, la matricola di un dipendente, la chiave composta dal codice studente insieme alla data e al codice della materia per le prove scolastiche. Nel caso delle tre entità Persona, Acquisto, Automobile gli attributi Codice fiscale e Modello sono le chiavi primarie per le entità Persona e Automobile, mentre Acquisto non ha chiave. La chiave primaria di un'entità viene riconosciuta dalla presenza dell'acronimo: {PK} posto tra parentesi graffe, accanto all'attributo chiave. nel caso di chiave formata da più attributi l'acronimo {PPK} chiave primaria parziale è posto accanto ognuno degli attributi che compongono la chiave. L'uso dei codici come chiave per identicicare le istanze di un'entità è ormai abitudine nei moderni sistemi: ciò è dovuto anche al fatto che in modo sempre più esteso vengono previsti, nell'automazione dei sistemi informativi, strumenti di rilevazione dei dati diversi dalla

Partial Primary Key

tastiera (lettori di caratteri, scanner per codici a barre), più veloci e più affidabili nelle operazioni di input.

Nelle altre rappresentazioni il simbolo grafico convenzionalmente usato per rappresentare l'attributo è la linea che parte dall'entità e termina con un piccolo cerchio. Nella descrizione grafica, gli attributi chiave vengono evidenziati sottolineandone il nome oppure colorando il cerchietto dell'attributo.

3.3 – Le associazioni tra entità

La molteplicità di un'associazione è il numero di possibili istanze di un'entità che viene messo in corrispondenza con un'istanza dell'altra entità che partecipa associazione.

Il numero minimo e massimo di possibili istanze viene rappresentato mediante una coppia di valori separati da punti: 1..1, 0..1, 1..N. Al valore minimo e massimo sono associati gli importanti concetti di obbligatorietà e cardinalità dell'associazione. Il valore minimo assume, in genere, uno dei due valori O e 1. Lo O indica che la partecipazione è facoltativa, mentre il valore 1 indica che la partecipazione è obbligatoria.

Obbligatorietà e Cardinalità

Valore minimo e massimo

Il valore massimo definisce la cardinalità della partecipazione all'associazione. Esso assume, in genere uno dei due valori 1 oppure N per indicare una o molte partecipazioni all'associazione.

La cardinalità può quindi essere a uno oppure a molti e pertanto le associazioni tra due entità si classificano nei seguenti tipi:

- associazione uno a uno o biunivoca, indicata con 1:1;
- associazione uno a molti, indicata con 1: N;
- associazione molti a molti, indicata con N: N.

Non è sempre facile attribuire questi valori e solo un attento esame della specifica situazione permette di definirli con esattezza.

Un metodo utile per risolvere i dubbi consiste nel rappresentare

l'associazione con di istanze delle entità collegate con archi, come mostrato negli esempi successivi.

Associazione 1:1 (uno a uno) o biunivoca

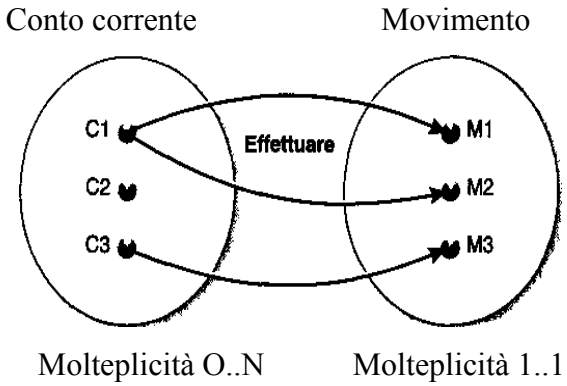
Biunivoca

Un'associazione si dice uno a uno, o biunivoca, e si indica con 1:1, quando ogni a della prima entità si deve associare a una sola istanza della seconda e viceversa. Per esempio l'associazione tra l'entità Studente e l'entità Diploma, in una scuola superiore, è biunivoca perché a ogni studente corrisponde un solo diploma e viceversa a un diploma corrisponde un solo studente**.** Consideriamo ora le entità Classe e Docente e l'associazione Coordinare che collega un docente con la classe di cui è coordinatore.

Associazione 1:N (uno a molti) o semplice

Semplice

Un'associazione si dice uno a molti, o semplice, e si indica con 1:N, quando ogni istanza della prima entità s;può associare a una o più entità, mentre a ogni istanza della seconda entità si deve associare una sola istanza della prima. Per esempio nella gestione dei movimenti su un conto corrente, ogni conto può effettuare una o più operazioni, ma ogni movimento deve riferirsi a un solo conto corrente. Quindi l'associazione Effettuare tra l'entità ContoCorrente e l'entità Movimento è uno a molti. Analizziamo in dettaglio la situazione.



Osservando gli archi in partenza dalle istanze di ContoCorrente possiamo dire che mero minimo di volte che il conto corrente viene

associato ai movimenti è 0 (partecipazione facoltativa) e che la cardinalità è a molti. Infatti su un conto corrente si possono fare nessun movimento o molti movimenti. La molteplicità è O..N (minima 0,massima N).

Se osserviamo il diagramma dalla parte di Movimento, il numero minimo di istanze è 1 perché tutti i movimenti devono riferirsi a un conto corrente (partecipazione obbligatoria).

La cardinalità è a uno, perché a un'istanza di Movimento corrisponde una ed una sola istanza di ContoCorrente.

La molteplicità è 1..1 (minima 1, massima 1).

Osservando gli archi in partenza dalle istanze di ContoCorrente possiamo dire che mero minimo di volte che il conto corrente viene associato ai movimenti è 0 (partecipazione facoltativa) e che la cardinalità è a molti. Infatti su un conto corrente si possono fare nessun movimento o molti movimenti.

La molteplicità è O..N (minima 0,massima N).

Se osserviamo il diagramma dalla parte di Movimento, il numero minimo di istanze è 1 perché tutti i movimenti devono riferirsi a un conto corrente (partecipazione obbligatoria).

La cardinalità è a uno, perché a un'istanza di Movimento corrisponde una ed una sola istanza di ContoCorrente.

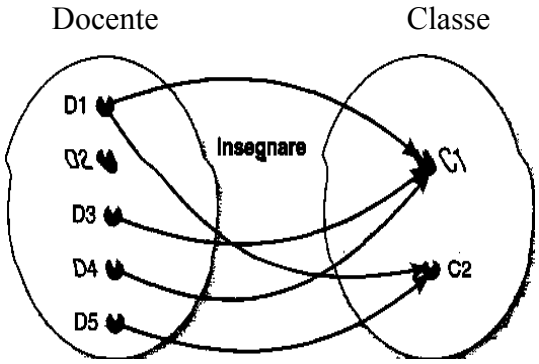
La molteplicità è 1..1 (minima 1, massima 1).

Associazione 1:N (uno a molti) o complessa

Complessa

Un'associazione si dice molti a molti, o complessa, e si indica con N:N, se ad ogni istanza della prima entità si possono associare a una o più istanze della seconda entità e a ogni istanza della seconda entità si possono associare una o più istanze della prima.

Consideriamo per esempio le entità Docente e Classe e l'associazione insegnare che associa i docenti di una scuola alle classi dove insegnano:



Ogni docente insegna in più classi e in ogni classe insegnano più docenti.

La situazione dalla parte docenti è chiara: un docente può insegnare in una o più classi ma potrebbe anche non insegnare in alcuna classe, avendo incarichi di altro come il docente D2.

Il numero minimo di istanze è O (partecipazione facoltativa) e la cardinalità è a molti. Infatti a un'istanza di Docente corrispondono più istanze di classe. La molteplicità è O..N (minima O, massima N).

Se analizziamo l'associazione partendo dalle classi ci troviamo di fronte alla difficoltà di definire il numero minimo di docenti per una classe. Abbiamo indicato la situazione un "?" al posto del valore minimo di istanze. Per superare la difficoltà dobbiamo specificare meglio il contesto nel quale è collocata l'associazione. Supponiamo che il caso in esame sia quello di un complesso scolastico formato da una scuola media inferiore e da una scuola elementare. Di conseguenza non ci sono mai meno di 2docenti per classe. Questo porta a definire il numero minimo di istanze a partecipazione obbligatoria. Inoltre la cardinalità è a molti, perché a ogni classe corrispondono più insegnanti. La molteplicità è 2..N (minima 2, massima N).

Test

1. Completa le frasi seguenti utilizzando una tra le parole elencate alla fine della domanda:

   Il livello……..…… rappresenta l'effettiva installazione degli archivi elettronici

   Il livello……..…… rappresenta la realtà dei dati e le relazioni tra essi attraverso uno schema

   Il livello…..…… rappresenta il modo attraverso il quale i dati sono organizzati negli archivi

   elettronici.

   top, opzionale, down, fisico, schematico, uno a uno, uno a molti, logico, molti a molti, obbligatorio, concettuale

2. La struttura di dati nel modello logico deve facilitare:
   a) la progettazione del modello concettuale;
   b) la scelta della chiave tra gli attributi di un'entità;
   c) le operazioni di manipolazione e di interrogazione;
   d) le operazioni di derivazione del modello fisico.

3. Per stabilire la natura dell'associazione tra le entità E1 ed E2, si rappresentano E1 ed E2 come insiemi e si collegano con archi. I collegamenti uscenti da E1 sono O..N e quelli uscenti da E2 sono 2..N. Quali delle seguenti affermazioni sono vere?
   a) Si tratta di un'associazione 1:1 tra E1 ed E2.
   b) Si tratta di un'associazione 1: N tra E1 ed E2.
   c) Si tratta di un'associazione N:1 tra E1 ed E2.
   d) Si tratta di un'associazione N:N tra E1 ed E2.
   e) La partecipazione di E1 all'associazione è obbligatoria.
   f) La partecipazione di E1 all'associazione è facoltativa.
   g) La partecipazione di E2 all'associazione è obbligatoria,
   h) La partecipazione di E2 all'associazione è facoltativa.

4. Quale tra le seguenti frasi esprime meglio il significato di schema Entity/ Relationship?
   a) Rappresentazione grafica del modello logico.
   b) Rappresentazione grafica del modello concettuale.
   c) Rappresentazione grafica del modello fisico.
   d) Rappresentazione grafica del modello sequenziale.

5. Si vogliono archiviare i dati relativi agli articoli pubblicati sulle riviste specializzate in un determinato settore, organizzandoli per rivista, per argomento e per autore.
6. Si vuole gestire un magazzino di prodotti organizzato per reparti, tenendo sotto controllo anche la situazione dei fornitori; in particolare si vuole poi rispondere alle seguenti esigenze:
   a) produrre listini di prodotti con descrizione, prezzi e sconti;
   b) produrre elenchi di prodotti con i relativi fornitori;
   c) controllare i prodotti sotto scorta.

UNITÀ DIDATTICA 4
MODELLO RELAZIONALE

4.1 – Forma normale di Boyce-Codd

BCNF

Una relazione è in forma normale di Boyce-Codd (BCNF, Boyce-Codd Normal Form) quando rispetta le caratteristiche fondamentali del modello relazionale (1FN) e in essa ogni determinante è una chiave candidata, cioè ogni attributo dal quale dipendono altri attributi può svolgere la funzione di chiave. La BCNF può essere espressa anche nel seguente modo:

- se in una relazione vale la dipendenza funzionale A -> B, allora l'insieme di attributi A deve contenere una chiave (e quindi può svolgere la funzione di chiave).

Da questo fatto discende immediatamente che una relazione che soddisfa la BCNF è anche in seconda e in terza forma normale, in quanto la BCNF esclude che un determinante possa essere composto solo da una parte della chiave, come avviene per le violazioni alla 2FN, o che possa essere esterno alla chiave, come avviene per le violazioni alla 3FN.Terza forma normale, ma non è vero l'opposto, come si evidenzia dall'esempio. Consideriamo una relazione che descrive l'allocazione delle sale operatorie di un ospedale. Le sale operatorie sono prenotate, giorno per giorno, in orari previsti, per effettuare interventi su pazienti ad opera dei chirurghi dell'ospedale. Nel corso di una giornata una sala operatoria è occupata sempre dal medesimo effettua più interventi, in ore diverse. Noti i valori di Paziente e sono noti anche: ora dell'intervento, chirurgo, e sala operatoria 3 attributi della relazione Interventi sono descritti nello schema:

Esempio

Interventi (Paziente, Data Intervento,Ora Intervento,Chirurgo,Sala)

In base alla precedente descrizione del caso in esame, nella relazione Interventi valgono le dipendenze funzionali:

- (Paziente, Data Intervento)→Ora Intervento,Chirurgo, Sala
- (Chirurgo, Data Intervento,Ora Intervento) →Paziente, Sala
- (Sala, Data Intervento,Ora Intervento) →Paziente, Chirurgo
- (Chirurgo, Data Intervento) →Sala

Ci sono tre insiemi di attributi che possono svolgere la funzione di chiave:

- (Paziente, Data Intervento), (Chirurgo,Data Intervento,Ora Intervento), (Sala, Data Intervento,Ora Intervento).

Scegliamo come chiave primaria la coppia di attributi:

- (Paziente, Data Intervento).

Paziente	Data Intervento	Ora Intervento	Chirurgo	Sala
Rossi	25/10/2005	8.00	Romano	Sala2
Negri	26/10/2005	9.30	Veronesi	Sala1
Viola	25/10/2005	10.30	De Bakey	Sala1
Verdi	25/10/2005	11.30	Romano	Sala2

Il significato delle forme normali viene di seguito riassunto.

Forme normali

Prima forma normale

Una relazione SI dice in prima forma normale (1FN) quando rispetta i requisiti fondamentali del modello relazionale, in particolare ogni attributo è elementare, non ci sono righe uguali e non ci sono attributi ripetitivi.

Seconda forma normale

Una relazione è in seconda forma normale (2FN) quando è in prima forma normale e non ci sono attributi non-chiave che dipendono parzialmente dalla chiave.

Terza forma normale

Una relazione è in terza forma normale (3FN) quando è in seconda forma normale e non ci sono attributi non-chiave che dipendono transitivamente dalla chiave Forma normale di Boyce-Codd.

Una relazione è in forma normale di Boyce-Codd (BCNF) quando è in prima forma normale e in essa ogni determinante *è* una chiave candidata. Oltre a quelle presentate finora, nella teoria dei database relazionali esistono altre forme normali di ordine superiore al terzo che risolvono situazioni di anomalia nelle operazioni sulle tabelle.
Più precisamente la quarta e la quinta forma normale risolvono i problemi che si possono creare quando nella relazione sono presenti attributi multivalore, cioè attributi che possono assumere più valori in corrispondenza dello stesso valore di un altro attributo. Queste forme inoltre servono a rendere minimo il numero degli attributi che formano le chiavi composte. Sebbene sia possibile definire tabelle anche in quarta (4FN) e quinta forma normale (5FN), di solito è sufficiente rappresentare le relazioni nel livello di normalizzazione 3FN che, come si può dimostrare, ha il pregio di essere sempre ottenibile senza perdita di informazioni e senza perdita di dipendenze funzionali. Non è così invece per la forma normale di Boyce - Codd: ci sono relazioni che non possono essere normalizzate nella forma di Boyce - Codd senza perdita di dipendenze funzionali, come si vede dall'esempio seguente:

Normalizzazione

- supponiamo di voler tenere traccia dei pazienti che devono essere sottoposti a più terapie chirurgiche, in diversi reparti, per la terapia di patologie complesse. Una relazione che soddisfa a tale esigenza è mostrata nella tabella a lato. Ogni n-upla della relazione TerapieComplesse associa un paziente al chirurgo che lo ha operato e al reparto nel quale è avvenuto l'intervento. Valgono le seguenti dipendenze funzionali: Chirurgo -> Reparto {Paziente, Reparto} -> Chirurgo

Interventi

Paziente Rossi	Reparto Cardiochirurgia	Chirurgo De Bakey
Rossi	Chir. Generale	Romano
Bianchi	Chir. Generale	Romano
Bianchi	Chir. Oncologica	Veronesi
Verdi	Chir. Generale	Lanzetta

Infatti un chirurgo è inquadrato in un determinato reparto e ogni paziente, quando è ricoverato in un certo reparto, è operato da un dato chirurgo. [Paziente, Reparto] è chiave per TerapieComplesse, per via della seconda dipendenza funzionale. La relazione è in 3FN, ma non è in BCNF, a causa della prima dipendenza funzionale che ha come determinante Chirurgo che non è chiave.
Se cerchiamo di scomporre la relazione con la tecnica illustrata in precedenza a partire dalla dipendenza funzionale Chirurgo -> Reparto, ottenga-mo le due relazioni Chirurghi e Pazienti.

Chirurghi

Chirurgo De Bakey	Reparto Cardiochirurgia
Romano	Chir. Generale
Veronesi	Chir. Oncologica
Lanzetta	Chir. Generale

Pazienti

Paziente Rossi	Chirurgo De Bakey
Rossi	Romano
Bianchi	Romano
Bianchi	Veronesi
Verdi	Lanzetta

Questa decomposizione non conserva la seconda delle due dipendenze funzionali definite sulla relazione originaria. L'effetto di questa perdita di dipendenze funzionali può avere effetti indesiderati.
Se, per esempio, si volesse registrare il fatto (errato) che il paziente Bianchi è stato operato da Lametta nel reparto di Cardiochirurgia, ci si limiterebbe a inserire nella tabella Paziente la coppia di valori: ("Bianchi", "Lanzetta") che è lecita. Solo quando si cerca di ricostruire i dati della relazione TerapieComplesse, congiungendo le tabelle Pazienti e Chirurghi, si ottiene la n-upla: ("Bianchi", "Lanzetta", "Chir. Generale") che permette di evidenziare l'errore nei dati. L'esempio mostra l'esistenza di relazioni che non sono in BCNF e non vengono portate a tale livello di normalizzazione, ma lasciate in 3NF, anche a costo di una certa ridondanza dei dati, perché la BCNF può portare alla perdita di dipendenze funzionali.
Le forme normali di ordine superiore contengono la stessa quantità di informazioni di quelle inferiori e non è obbligatorio il passaggio a queste forme normali: il modello relazionale richiede come obbligatoria solo la prima forma normale.
Tuttavia il passaggio alle forme normali superiori consente di distinguere e separare con precisione gli oggetti, senza perdita di informazioni, anche se viene generata una ridondanza di dati che però è costantemente sotto controllo.

4.2 – L'integrità referenziale

Nella definizione dei concetti fondamentali del modello relazionale è già stata descritta una regola di integrità sui dati, l'integrità sull'entità o vincolo di chiave, che non consente valori nulli e valori duplicati per la chiave.

Referential integrity

Il modello relazionale possiede regole altre regole di integrità dei dati e, in particolare, sussistono vincoli di tupla e vincoli di integrità referenziale.

Ordini

NumeroOrdine	DataOrdine	CodCliente
2	12/08/2005	Luci
3	12/08/2005	
4	20/12/2005	Levi
5	21/12/2005	Luci
6	32/13/2005	Meta
7	12/01/2006	Lami
8	13/01/2006	Vite

Clienti

Codice	RagioneSociale	Indirizzo
Lami	Lamiere per Auto	Torino
Levi	Levigatoria Toscana	Firenze
Luci	Lucidatura Metalli	Tarante
Meta	Metallurgica Sarda	Cagliari
	Viteria Lombarda	Sondrio

Osserviamo le due tabelle Ordini e Clienti; in esse sono stati evidenziati alcuni campi che invalidano i corrispondenti record della tabella. L'ordine 3 non è riferito ad alcun cliente, la data dell'ordine 6 è assurda, l'ordine 8 è riferito a un cliente inesistente; alla Viteria Lombarda non è abbinato alcun valore della chiave e, di conseguenza, non è possibile collegarla ad alcun ordine.

Per prevenire queste situazioni sono state definite opportune regole di integrità sui dati tra le quali ricordiamo la già citata integrità di chiave che è, appunto, la regola d'integrità violata in Clienti.

I vincoli di tupla esprimono condizioni che devono essere soddisfatte dai valori di ciascuna n-upla indipendentemente dalle altre. La data dell'ordine 6 è un esempio di violazione di un vincolo di dominio, in quanto il valore 32/13/2005 non ricade nell'insieme delle date ammissibili.

La natura del problema analizzato porta a definire i vincoli ai quali

devono sottostare i dati. Per esempio l'attributo CodiceCliente di Ordini non può assumere valori nulli per prevenire situazioni quali quella dell'ordine avente il numero 3. L'integrità referenziale (referential integrity) è un insieme di regole del modello relazionale che garantiscono l'integrità dei dati quando si hanno relazioni associate tra loro attraverso la chiave esterna: queste regole servono per rendere valide le associazioni tra le tabelle e per eliminare gli errori di inserimento, cancellazione o modifica di dati collegati tra loro.
L'integrità referenziale viene rispettata quando per ogni valore non nullo della chiave esterna, esiste un valore corrispondente della chiave primaria nella tabella associata. Per esempio, nel database relazionale che contiene la tabella dei clienti e la tabella degli ordini viste sopra, il codice del cliente della tabella Ordini è associato alla chiave della tabella Clienti.
Clienti (Codice, RagioneSociale, Indirizzo)
Ordini (NumeroOrdine, DataOrdine, CodiceCliente)

Applicare l'integrità referenziale al database significa garantire che un valore, presente nella tabella Ordini per la chiave esterna CodiceCliente, abbia un corrispondente valore di Codice in una delle righe della tabella Clienti. Inoltre non si deve consentire la cancellazione di un cliente dalla tabella Clienti se ci sono righe nella tabella Ordini che si riferiscono ad esso. L'integrità referenziale, se applicata, non permette che si presentino situazioni come quella dell'ordine 8, riferito a un cliente inesistente, di cancellare la riga relativa al cliente di codice Lami per la presenza di un ordine collegato, oppure di modificare il valore della chiave della Metallurgica Sarda per la medesima ragione.
In generale data una relazione R1 avente come chiave l'attributo K1 e la relazione R2 avente come chiave esterna KE2 associata a K1, le regole dell'integrità referenziale impongono che:

- ogni valore di KE2 deve avere un valore uguale di K1 in una

delle righe della tabella R1;

- oppure il valore di KE2 deve essere nullo.

Quando viene applicata l'integrità referenziale, è necessario osservare le seguenti regole pratiche:

non è possibile immettere un valore nella chiave esterna della tabella associata, setale valore non esiste tra le chiavi della tabella primaria. È possibile, comunque,immettere un valore nullo nella chiave esterna, per rappresentare il fatto che le righe non sono correlate, non è possibile eliminare una n-upla dalla tabella primaria, se esistono righe legate ad essa attraverso la chiave esterna nella tabella correlata. Inoltre non si può modificare, come è ovvio, il valore alla chiave nella tabella primaria, se ad essa corrispondono righe nella tabella correlata.

4.3 – Osservazioni sul modello relazionale

Come si è potuto notare, il modello relazionale è più intuitivo e più espressivo per la strutturazione dei dati, rispetto ai modelli gerarchico e reticolare. La teoria dei database relazionali è costruita a partire da sicuri fondamenti matematici e utilizza un linguaggio rigoroso: questo consente di sviluppare definizioni, teoremi e dimostrazioni.

Dal punto di vista informatico presenta una grande semplicità nell'uso e nell'implementazione, anche se è relativamente più lento nella ricerca e occupa più spazio su memoria di massa rispetto ai database creati e gestiti da DBMS basati sugli altri modelli.

Il trattamento dei dati avviene per gruppi di record, anziché per singoli record, come avviene nelle organizzazioni convenzionali degli archivi. ritrovamento delle informa-zioni viene realizzato operando sulle righe e sulle colonne delle tabelle, con gli operatori di selezione, proiezione e congiunzione. Le operazioni sulle relazioni producono nuove relazioni, alle quali si possono ulterior-mente applicare gli operatori.

Quindi non è necessario specificare la sequenza del percorso che deve essere seguito per accedere ai dati contenuti nel database, mentre i modelli gerarchico e reticolare sono strettamente condizionati dal tipo di cammino insito nella struttura dei dati.

Nello schema tabellare del modello relazio-nale inoltre l'ordine con il quale le righe compaiono nella tabella è ininfluente.

Il modello relazionale ha portato poi benefici nel lavoro di progettazione del database: il progettista del database può costruire il modello dei dati considerando con attenzione le entità, le associazioni e le dipendenze tra gli attributi nel modello della realtà.

Il passaggio dal modello concettuale al modello logico può essere realizzato con semplici regole. Per quanto riguarda infine il processo di normalizzazione delle relazioni, è opportuno osservare che nella progettazione del modello relazionale non sempre è conveniente arrivare a gradi elevati di normalizzazione, anche perché nella realtà è raro trovare situazioni che vengono tradotte con tabelle normalizzate con ordine superiore alla terza forma normale. Tenuto conto delle caratteristiche della realtà che si vuole rappresentare (e nello stesso tempo dell'implementazione più o meno efficace dell'operazione di congiunzione nello specifico prodotto utilizzato per la gestione del database), occorre cercare un livello equilibrato di normalizzazione, per evitare di appesantire la fase di ritrovamento dei dati con un numero eccessivo di operazioni di congiunzione tra le tabelle.

La normalizzazione può essere considerata il passo finale del

processo complessivo di progettazione della base di dati. Inoltre può consentire di attuare un controllo sul modello realizzato attraverso alcune semplici regole pratiche.

Test

1. Quali delle seguenti frasi esprime meglio la definizione di relazione di grado 2?
   a) Una tabella bidimensionale.
   b) Un sottoinsieme di un qualsiasi insieme.
   c) Una tabella con due colonne.
   d) Una tabella con due righe.

2. Se il dominio dell'attributo A1 è formato da un insieme di 5 oggetti e quello dell'attributo A2 ne contiene 3, che cosa si può dire in merito alla cardinalità di una relazione con attributi A1 e A2?
   a) La cardinalità è 25.
   b) La cardinalità è minore o uguale a 15.
   c) La cardinalità è minore di 15.
   d) La cardinalità è minore o uguale a 25.
   e) La cardinalità è minore o uguale a 9.

3. Associa a ciascun termine indicato con la numerazione da a ad f la definizione corretta tra quelle elencate con numerazione da 1 a 6:
   a) Grado
   b) Dominio
   c) Relazione
   d) Tu pia
   e) Cardinalità
   f) Chiave
      1. Insieme di n-uple
      2. Riga di una tabella
      3. Attributo che identifica una n-upla
      4. Insieme dei valori assunti da un attributo
      5. Numero delle colonne della tabella
      6. Numero delle n-uple

4. Organizzare le informazioni relative agli abbonamenti di una casa editrice che pubblica più riviste: ogni rivista possiede molti abbonati e un abbonato può sottoscrivere più abbonamenti a riviste diverse. Dopo aver progettato il modello della base di dati, rappresentare le seguenti interrogazioni:
   a) città di residenza degli abbonati a una rivista;
   b) titolo e prezzo dell'abbonamento di tutte le riviste;
   c) cognome e nome degli abbonati che hanno sottoscritto un abbonamento a una qualsiasi delle riviste nel primo trimestre dell'anno in corso;
   d) titolo e periodicità della rivista con prezzo dell'abbonamento superiore a una cifra prefissata.

UNITÀ DIDATTICA 5
IL LINGUAGGIO SQL

5.1 – Caratteristiche generali

Abbiamo osservato che tramite i servizi di un DBMS, ci si aspetta di poter gestire gli archivi di un sistema informativo con notevole efficienza e che esso metta a disposizione uno specifico linguaggio per:

- definire e creare il database;
- effettuare le diverse operazioni di gestione dei dati, quali l'inserimento, la cancellazione e la variazione dei record di un archivio;
- interrogare il database a scopo informativo.

Il linguaggio deve permettere di fare tutto questo facilmente ed essere basato su costrutti semplici e facili da imparare. Le sue caratteristiche, infine, devono essere standardizzate in modo che un utente, cambiando DBMS, non debba apprendere un nuovo linguaggio per usare la base dei dati.

Il linguaggio SQL è nato con l'intento di soddisfare a queste richieste nei database relazionali. Le interrogazioni che si possono costruire con SQL sono una estensione di quelle che si possono realizzare con sequenze di operazioni relazionali, in quanto con SQL, è possibile, effettuare calcoli, raggruppamenti e ordinamenti. Non è un linguaggio procedurale ed è divenuto uno standard fra i linguaggi per la gestione dei database relazionali.

Riferendoci a quanto affermato nelle unità didattiche precedenti affermiamo che anche SQL consente all'utente di definire la struttura delle relazioni e controllare gli accessi (riferirsi alle funzioni di DDL); modifica i dati; consente di interrogare (funzioni di QL).

Il linguaggio SQL utilizza i caratteri, cifre decimali, operatori aritmetici e di confronto più altri caratteri che assumono particolari significati nella sintassi delle istruzioni.

Da tener presente la terminologia del linguaggio SQL è analoga a quella di ACCESS dove un database è costituito da tabelle, attributi e record. Rispetto ai linguaggi di programmazione,questo è un linguaggio che consente di aumentare la produttività nel progetto e nello sviluppo delle applicazioni orientate alla gestione di dati.

5.2 – Comandi di definizione e manipolazione

Creazione

Il comando per creare la tabella è: CREATE TABLE, seguito di norma dal nome della tabella e dell'elenco degli attributi.
Per ogni attributo occorre specificare il nome e il tipo di dato. Gli attributi possono essere qualificati mediante diverse clausole con le quali è possibile definire la chiave primaria, le chiavi esterne, l'obbligatorietà e il valore di default di un campo.

Modifica

La struttura di una tabella può essere modificata in un secondo tempo con il comando Alter Table, per aggiungere una nuova colonna ADD a quelli già esistenti, oppure per toglier una colonna DROP.
I comandi illustrati rappresentano la parte del linguaggio SQL che fa riferimento alla categoria dei comandi definiti da Data Definition Language. I valori degli attributi nelle righe della tabella possono essere inseriti, aggiornati o cancellati rispettivamente con i comandi Insert, update e delete.

Il comando Select

Select

L'aspetto più importante del linguaggio SQL è costituito dalla possibilità di porre interrogazioni in modo molto semplice alla base di dati per ritrovare le informazioni interessano Queste prestazionisono fornite dal comando SELECT che e nello stesso tempo molto potente e molto semplice da usare. Con il comando Select vengono attivate le interrogazioni sulle relazioni e le operazioni relazionali per ottenere nuove tabelle. Inoltre l'utente viene liberato da tutti problemi che riguardano le modalità e i percorsi per ritrovare i dati,

cioè la loro collocazione fisica nelle memoria di massa: deve solo specificare al sistema quali dati vuole ottenere.

La struttura generale del comando Select è la seguente: ***Sintassi***

- SELECT
- FROM
- WHERE

Accanto alla parola Select vengono indicati i nomi degli attributi (le colonne) da elencare (se è necessario elencare tutti gli attributi basta scrivere il segno di asterisco * dopo la parola Select); dopo From vengono indicati i nomi delle tabelle su cui deve operare il comando Select; dopo la clausola Where si specifica la condizione che deve essere soddisfatta dai campi delle righe: possono comparire anche più condizioni combinate con gli operatori AND, OR e NOT. ***Clausole***

Per esempio l'elenco con cognome, nome e codice fiscale dei dipendenti con funzione di Impiegato si ottiene con il comando Select nella forma:

- SELECT Cognome, Nome, CodFisc
- FROM Personale
- WHERE Funzione = Impiegato.

In questo primo esempio accanto alla parola Select sono stati specificati solo alcuni attributi tra quelli presenti nella tabella.

Per richiedere tutti i dati dei dipendenti che abitano in provincia di Milano, si usa il comando Select nella forma:

- SELECT *
- FROM Personale
- WHERE Prov = MI

Nel secondo esempio il simbolo * sostituisce l'elencazione di tutti gli attributi della tabella Personale.

Il comando Select possiede due predicati ALL e DISTINCT. ***Predicati ALL e DISTINCT***

Il predicato ALL indica la richiesta di ottenere come risultato dell'interrogazione tutte le righe che soddisfano alle condizioni contenute nel comando. Questo predicato è di default, cioè se non viene fatta nessuna specificazione vengono visualizzate tutte le righe della tabella che rispondono alle condizioni poste.

Selezione e proiezione

Le operazioni di selezione, proiezione e congiunzione su una base di dati relazionale vengono realizzate in pratica attraverso il comando Select, secondo le diverse forme consentite dalla sintassi di questo comando. Se si vuole rispettare una delle regole del modello relazionale che non consente la presenza di righe uguali all'interno della stessa tabella, basta porre accanto alla parola Select la clausola Distinct: con questa specificazione la tabella ottenuta non contiene righe duplicate.

La selezione e la proiezione sono state già state utilizzate in pratica negli esempi di uso del comando Select nel paragrafo precedente. L'operazione di selezione, che consente di ricavare da una relazione un'altra relazione contenente solo le righe che soddisfano ad una certa condizione, viene realizzata nel linguaggio SQL utilizzando la clausola Where del comando Select. Per esempio per ottenere l'elenco con tutti i dati dei dipendenti che svolgono la funzione di Dirigente, si opera una selezione sulla tabella Personale estraendo le righe per le quali l'attributo Funzione contiene il valore "Dirigente".

Pseudocodifica

In pseudocodifica si scrive:

- Selezione di Personale
- per Funzione = "Dirigente"

La precedente interrogazione viene codificata in linguaggio SQL con il comando:

- SELECT *
- FROM Personale
- WHERE Funzione = Dirigente

Proiezione

L'operazione di proiezione, che permette di ottenere una relazione contenente solo alcuni attributi della relazione di partenza, si realizza indicando accanto alla parola Select l'elenco degli attributi richiesti.

Descrizione del comando

Il comando Select può operare su più tabelle, indicandone i nomi (separati da virgola) dopo la parola From; scrivendo poi dopo la parola Where i nomi degli attributi che si corrispondono nelle due tabelle (legati tra loro dal segno =), si realizza in pratica l'operazione di congiunzione di due tabelle secondo un attributo comune. Qualora ci sia l'esigenza di maggiore chiarezza nella descrizione del comando oppure ci siano due attributi con lo stesso nome in due tabelle diverse, è opportuno, come già visto, indicare il nome della tabella e il nome dell'attributo separati dal punto.

Per esempio si supponga di aver creato accanto alla tabella Personale anche la tabella Dipendenza con il comando CREATE TABLE Dipendenza (CodF, smallint, Descrizione char(20) Indirizzo char(25)); contenente per ogni dipendenza il codice, usato nei dati dei dipendenti, la descrizione e l'indirizzo della filiale. Dopo aver visto come si rappresentano nel linguaggio SQL le operazioni fondamenta del modello relazionale, si può facilmente comprendere a questo punto come realizzabile la combinazione di diverse operazioni (proiezione e selezione, congiunzione e selezione, congiunzione e proiezione, congiunzione, selezione e proiezione) usando:

- struttura Select...From... Where... secondo tutte le sue possibili varianti.

5.3 – Le funzioni di aggregazione

All'interno del comando Select possono essere usate funzioni predefinite che agiscono sui valori contenuti in insiemi di righe della tabella e che per questo motivo si chiamano funzioni di aggregazione.

Funzione COUNT

La funzione COUNT conta il numero di righe presenti in una

tabella. La sintassi del linguaggio SQL richiede di specificare come argomento della funzione il nome di un attributo oppure il carattere * (asterisco): nel primo caso non vengono conteggiate le righe che hanno valore Null nella colonna dell'attributo specificato; nel secondo caso, indicando l'asterisco, la funzione Count (*) calcola il numero delle righe della tabella, incluse quelle con campi di tipo Null.

In sostanza la funzione Count (*) serve per ottenere la cardinalità di una relazione.

Condizioni di ricerca

La funzione calcola solo il numero delle righe, indipendentemente dai valori in esse memorizzati. Le condizioni di ricerca sono utilizzate insieme alle clausole Where e Having per minare i criteri di selezione rispettivamente delle righe e dei raggruppamenti. scrittura delle condizioni si usano i segni del confronto =, <, >, <>, >, <=. Una condizione di ricerca è costruita anche mettendo insieme più condizioni legate tra loro con gli operatori AND e OR, precedute eventualmente dall'operazione NOT. L'ordinedi applicazione degli operatori è il seguente: NOT viene applicato prima di AND prima di OR. Le condizioni di ricerca possono utilizzare anche altre parole del linguaggio che indicano operatori o predicati, con i quali è possibile rendere ancora più raffi: interrogazioni alla base di dati.

BETWEEN

BETWEEN: le condizioni di ricerca sono utilizzate insieme alle clausole Where e Having per minare i criteri di selezione rispettivamente delle righe e dei raggruppamenti. scrittura delle condizioni si usano i segni del confronto =, <, >, <>, >, <=.

Operatore IN

Una condizione di ricerca è costruita anche mettendo insieme più condizioni contemporaneamente. L'operatore IN controlla se un valore appartiene ad un insieme specificato di valori, cioè è possibile richiedere le righe che hanno i valori di 1111 attributo compresi in una lista di valori indicati dopo la parola In all'interno della condizione scritta dopo Where. L'operatore LIKE confronta il valore di un attributo di tipo carattere con un modello di stringa che può contenere caratteri jolly (o metacaratteri).

LIKE

Jolly

I caratteri jolly sono:

- (underscore) per indicare un singolo carattere qualsiasi in quella posizione della stringa:
- % (percento) per indicare una sequenza qualsiasi di caratteri in quella posizione della stringa.

Per esempio:

- LIKE 'xyz%' vengono ricercate tutte le stringhe che iniziano con i caratteri 'xyz'
- LIKE '%xyz' serve a ricercare tutte le stringhe che finiscono con i caratteri 'xyz'
- LIKE '%xyz%' per tutte le stringhe che contengono al loro interno i caratteri 'xyz';
- LIKE 'xyz' controlla le stringhe di 4 caratteri che finiscono con 'xyz'.

L'operatore Like utilizzato con un modello di stringa che non contiene caratteri jolly è del tutto equivalente all'operatore =.

IS NULL

Il predicato IS NULL confronta il valore in una colonna con il valore Null. L'uso di questo predicato e il solo modo per controllare la presenza del valore Null in una co1on E possibile inserire l'operatore di negazione NOT per valutare la condizione opposte. In altre parole per controllare se un attributo non ha valore Null. L'operatore Is viene utilizzato solo con la parola Null.

View

Il linguaggio SQL consente di decidere le modalità con le quali gli utenti possono vedere le tabelle del database, creando una finestra, detta VIEW (vista), su alcuni o su tutti i dati contenuti in una o più tabelle.

La vista viene identificata con un nome assegnato in fase di creazione con il comando CREATE VIEW.

Con le viste è possibile decidere anche che un utente debba avere una visione solo parziale dei dati registrati nel database, mentre non

può vedere altri dati: in questo modo si elimina il rischio di modifiche indesiderate su dati che non riguardano il lavoro di un determinato utente e nello stesso tempo l'utente non rimane distratto dai dati che sono ininfluenti sul tipo di elaborazione che deve fare. L'utente può comunque operare sulla vista con i comandi illustrato in precedenza, come se fosse una tabella del database.

Le viste sono finestre dinamiche sulle tabelle del database, in quanto ogni modifica ai dati sulla tabella primaria è disponibile per l'utente attraverso la vista. Vale anche l'opposto: ogni modifica sui dati della vista si riflette sui dati della tabella primaria. Non tutte le viste sono, però, aggiornabili: infatti le viste che contengono risultati di funzioni di aggregazione non possono essere modificate

La creazione della vista viene realizzata con l'uso dell'istruzione Select all'interno del comando Create View.

Il linguaggio SQL fornisce un insieme di comandi per salvaguardare l'integrità dei de contro situazioni di malfunzionamento del sistema o di danneggiamento dei dati:

Lock e Unlock

- LOCK TABLE e UNLOCK TABLE per limitare e ripristinare l'accesso di determina utenti a una tabella;
- RECOVER TABLE che consente di recuperare una tabella da una copia di sicurezza, nel caso in cui si verifichi un'interruzione anomala del processo di elaborazione;
- CHECK TABLE per controllare la corrispondenza dei dati di una tabella con i suoi indici.
- REPAIR TABLE per ricostruire gli indici di ima tabella nel caso in cui il control eseguito con il precedente comando Check Table non sia andato a buon fine.

Uno degli aspetti più interessanti del comando Select è costituito dalla possibilità di inserire un comando Select all'interno della struttura di un altro comando Select, ponendo un'interrogazione all'interno di un'altra interrogazione, costruendo cioè interrogazioni

nidificate (sub query). Questa caratteristica spiega la presenza del termine structured nella sigla del linguaggio SQL. per indicare un linguaggio che consente di costruire interrogazioni complesse e ben strutturate.

Sub query

La condizione scritta dopo Where confronta il valore di un attributo con il risultato di un altro comando Select. Una subquery può restituire un valore singolo, nessun valore oppure un insieme di valori, ma deve comunque avere una singola colonna o espressione accanto alla sua Select.

Esempio di sub query

Per esempio è possibile ottenere l'elenco con cognome e nome dei dipendenti che hanno lo stipendio base inferiore allo stipendio medio di tutti i dipendenti usando il comando Select espresso nella forma:

- SEIJECT Cognome, Nome
- FROM Personale
- WHERE StipBase <
- (SELECT AVG (StipBase)
- FROM Personale)

Il comando Select nidificato restituisce il valore calcolato del valore medio degli stipendi; questo numero viene usato poi nell'interrogazione principale per il confronto con i valori dell'attributo StipBase nel criterio di selezione delle righe della tabella, scritto dopo Where. Di seguito viene presentato un altro esempio di subquery con uso del comando Select in una forma che contiene molte delle parole-chiave presentate precedentemente, e che illustra in modo efficace la potenza espressiva del comando Select per indicare con grande concisione un insieme complesso di operazioni.

Supponiamo di voler ricercare i dipendenti, elencando in ordine alfabetico cognome, nome e descrizione della filiale dove lavorano, per i quali lo stipendio risulta uguale al valore massimo tra tutti gli stipendi dei dipendenti con la funzione di Impiegato:

- Cognome, Nome, Descrizione

- Personale, Dipendenza
- Filiale = CodFis
- AND StipBase = (SELECT MAX(StipBase) FROM Personale
- WHERE Funzione = ‘Impiegato)

L’interrogazione si ottiene con la congiunzione tra le due tabelle Personale e Dipendenza. realizzata attraverso l’attributo comune del codice filiale; la condizione di selezione sulle righe risultanti confronta lo stipendio di ogni dipendente con il valore ottenuto da una sottointerrogazione che restituisce un numero, ottenuto calcolando con la funzione Ma il valore massimo tra tutti i valori di StipBase.
Nella costruzione delle subquery si possono usare alcune clausole che consentono a effettuare interrogazioni più complesse con poche righe di codice SQL.

ANY

Il predicato ANY indica che la subquery può restituire zero, oppure uno, oppure u insieme di valori, e che la condizione di ricerca è vera se il confronto è vero per alme uno dei valori restituiti.
La condizione di ricerca è falsa se la subquery restituisce un insieme vuoto oppure se confronto è falso per ciascuno dei valori restituiti dalla subquery.
Il seguente esempio di interrogazione serve per ottenere le informazioni dei dipende che non sono impiegati e che hanno lo stipendio superiore a quello di uno qualsiasi gli impiegati:

- SELECT Cognome, Nome, Funzione FROM Personale
- WHERE Funzione <> Impiegato
- AND StipBase > ANY (SEIJECT StipBase
- FROM Personale
- WHERE Funzione = Impiegato);

ALL

Il predicato ALL indica che la subquery può restituire zero, oppure uno, oppure un insieme di valori, e che la condizione di ricerca è

vera se il confronto è vero per ciascuno dei valori restituiti.
La condizione di ricerca è falsa se il confronto è falso per almeno uno tra i valori. Sostituendo l'interrogazione precedente con la seguente che contiene All al posto si possono estrarre tutte le righe dei dipendenti che non sono impiegati e che h stipendio superiore a quello di tutti gli impiegati.

- SELJECT Cognome, Nome, Funzione FROM Personale
- WHERE Funzione <> Impiegato
- AND StipBase > AL (SELECT StipBase
- FROM Personale
- WHERE Funzione = Impiegato).

È logico immaginare che il numero delle righe ottenute con l'interrogazione con All sia inferiore al numero di righe restituite dall'interrogazione contenente And. Le clausole Any e All possono essere tralasciate nelle espressioni di confronto se si è in grado di stabilire che la subquery restituirà sicuramente un solo valore. In questo caso la condizione di ricerca è vera se è vero il confronto tra il valore dell'attributo e il valore restituito dalla subquery. Il predicato IN serve a controllare se il valore di un attributo è compreso tra quelli restituiti dalla subquery effettuata con la Select nidificata. È possibile utilizzare NOT IN per estrarre solo le righe della tabella principale per, le quali nessuna riga della tabella ottenuta con la subquery contiene un valore uguale.

NOT IN

Si osservi che la condizione di ricerca, Where Attributo IN (SELECT
è equivalente a:

Condizioni di ricerca

Where Attributo = ANY (SELECT

Analogamente la condizione di ricerca, Where Attributo NOT IN (SELECT

È equivalente a:

Where Attributo < > ALL (SELECT

EXISTS

Exists

Il predicato EXISTS controlla se vengono restituite righe dall'esecuzione della subquery: La condizione di ricerca è vera se la Select nidificata produce una o più righe come risultato, è falsa se la subquery restituisce un insieme vuoto. Per esempio se si vuole ottenere l'elenco dei dipendenti con cognome e nome solo se esistono dipendenti di sesto livello, si può usare il comando Select nel seguente formato:

- SELECT Cognome, Nome
- FROM Personale
- WHHRE EXISTS (SELECT *
- FROM Personale :1
- WHERE Livello = 6)

5.4 – I comandi per la sicurezza

Chi crea le relazioni del database, può stabilire anche il diritto di accesso per utenti specifici o per tutti, nel caso di accessi multipli.

Concedere permessi

Il comando Grant concede i permessi, specificando il tipo di accesso, le tabelle sulle quali è consentito l'accesso e l'elenco degli utenti ai quali è permesso di accedere. Il tipo di accesso può riguardar il diritto di modificare la struttura con l'aggiunta di nuove colonne, oppure di modificare dei dati contenuti nella tabella,oppure l'uso del comando Select. Per concedere il diritto di modifica sulla tabella si usa il comando Grant:

- GRANT UPDATE
- ON
- TO

Revoca dei permessi

La revoca dei permessi con annullamento dei diritti di accesso viene effettuato con il comando REVOKE che ha una sintassi analoga a quella del comando GRANT:

- Revoke Update
- ON
- FROM

I permessi che possono essere concessi o revocati agli utenti sono indicati con le seguenti parole chiave che vanno specificate dopo Grant o Revoke:

- ALTER: aggiunge o elimina colonne oppure per modificare i tipi di dati
- DELETE: elimina righe dalla tabella
- INDEX:crea indici
- INSERT: inserisce nuove righe nelle tabelle
- SELECT: per ritrovare i dati nelle tabelle
- UPDATE: cambia i valori nella tabella
- ALL: per tutti i permessi.

I permessi e i diritti di accesso possono riguardare, non solo le tabelle, ma, anche le viste create con il comando CREATE VIEW.

Test

1. Completa le frasi seguenti utilizzando una tra le parole elencate alla fine della domanda:

   Per eliminare una tabella dal database si usa il comando……..

   Per fare un'interrogazione al database si usa il comando……

   Per modificare i valori nelle righe di una tabella si usa il comando……

   Per inserire nuove tabelle nel database si usa il comando……

   Per aggiungere una colonna a una tabella si usa il comando……..

   INSERT, SELECT, CREATE TABLE, UPDATE,ALTER TABLE, DELETE, DROP TABLE.

2. Scrivere le istruzioni SQL per la creazione delle tabelle degli studenti, delle materie e dei docenti.

UNITÀ DIDATTICA 6
CARATTERISTICHE GENERALI DI PROGRAMMAZIONE

6.1 – Caratteristiche generali del Visual Basic

Visual basic

Per ottenere un aiuto contestuale dall'help di Visual Basic sulla sintassi di funzioni o istruzioni, oppure su proprietà, eventi o metodi, basta selezionare la parola chiave e premere il tasto F1.

Gli operatori possono essere di tre tipi: aritmetici, di relazione e logici. Gli operatori aritmetici sono:

- + per l'addizione,
- per la sottrazione,
- per la moltiplicazione,
- /per la divisione con quoziente decimale,
- \per la divisione tra numeri interi e per ottenere il quoziente intero,
- MOD per il calcolo del resto della divisione tra interi,
- ^per l'elevamento a potenza.

Esempio

Per esempio:

- 6\4 = 1;7\3 = 2;2\3 = 0.
- 6 MOD 4 = 2; 7 MOD 3 = 1; 2 MOD 3 = 2;.

Dato un numero N intero qualsiasi, N è dispari se:

- N MOD 2 = 1, pari se N MOD 2 = 0.

Gli operatori di relazione sono utilizzati per confrontare il contenuto di due variabili e sono indicati con i simboli:

- < minore di,
- <= minore o uguale di,
- <maggiore,
- >= maggiore o uguale di,

- <>diverso.

Gli operatori logici sono: AND per il prodotto logico (congiunzione), OR per la somma logica (disgiunzione), NOT per la negazione, XOR per l'OR esclusivo.
I dati utilizzati all'interno di un programma possono essere:

1. costanti, se non cambiano il loro valore durante l'esecuzione del programma;
2. variabili, se cambiano il valore.

Le costanti utilizzate nel programma vengono precedute dalla parola CONST, secondo frasi del tipo:

CONST Nome = espressione

Per esempio:

CONST PiGreco = 3.14 CONST Risposta = “SI”

Se la costante contiene caratteri il valore della costante viene racchiuso tra virgolette, per costanti di tipo numerico la separazione tra cifre intere e decimali è indicata con carattere. (punto).
La dichiarazione delle variabili utilizzate nel programma inizia con la parola DIM.

DIM

DIM Nome AS tipo

I nomi delle variabili devono iniziare con una lettera e possono contenere numeri e lettere fino a un massimo di 40 caratteri.
I dati trattati in un programma possono essere:
numerici, quali età, importi, stipendi, misure;
alfanumerici (o stringhe), quali nomi, descrizioni, codici.
I tipi principali per le variabili in Visual Basic sono:

- Boolean: tipo di dati con solo due valori possibili, ovvero True (-1) o False (0). Le variabili di tipo Boolean vengono

memorizzate come numeri a 16 bit (2 byte).

- Integre: tipo di dati contenente variabili memorizzate come numeri interi a 16 bit (2 byte)
- Currency: tipo di dati utilizzato per calcoli monetari o a virgola fissa in cui la precisione è fondamentale.
- Single: tipo di dati che contiene variabili a virgola mobile e precisione singola a 32 bit (4 byte), per valori negativi e per valori positivi. (7 cifre significative).
- Double: tipo di dati che contiene numeri a virgola mobile e doppia precisione a 64 bit (8 byte) per i valori negativi, e per i valori positivi.(15 cifre significative).

Double

- Date: tipo di dati utilizzato per memorizzare date e orari come numeri reali. Le variabili di tipo Date vengono memorizzate come numeri a 64 bit (8 byte). Il valore a sinistra del separatore decimale rappresenta una data e il valore a destra rappresenta un orario.
- String: tipo di dati utilizzato per memorizzare una sequenza di caratteri contigui. Può includere lettere. numeri, spazi e segni di punteggiatura. Il tipo String può contenere stringhe di lunghezza fissa con lunghezza compresa tra O e circa 63 KB di caratteri; è possibile stabilire la lunghezza della stringa al momento della definizione della variabile indicando dopo String il numero dei caratteri preceduto da un asterisco. Per esempio: Dim Nome As String * 15 specifica che la variabile Nome può contenere al massimo 15 caratteri.

String

- Variant: è il tipo di dati in cui vengono trasformate tutte le variabili se non sono dichiarate esplicitamente come tipo diverso utilizzando l'istruzione Dim. Variant è un tipo di dati speciale che può contenere qualsiasi tipo sia numerico che alfanumerico. È possibile utilizzare Variant al posto di qualsiasi tipo per gestire i dati in modo più flessibile.

Variant

Esempi di dichiarazione di variabili:

- Dim Contatore As integer
- Dim Anni As integer
- Dim Statura As single
- Dim AreaCerchio As double Dim AreaTriang As double Dim Nome As string

Dim Trovato As boole

As boole

L'identificatore di una variabile o di una costante è una sequenza qualsiasi di caratteri alfabetici e cifre, che inizia comunque con una lettera; si può usare anche il carattere _ per definire nomi composti, per esempio Area_Cerchio.

Gli identificatori rappresentati con nomi composti sono spesso scritti con tutti i caratteri di seguito e utilizzando l'iniziale maiuscola per ciascun nome, per esempio AreaCerchio.

All'interno del programma possono essere inserite frasi contenenti commenti o annotazioni del programmatore, con le quali è possibile documentare il significato delle variabili o delle costanti utilizzate, oppure la funzione svolta da una parte del programma.

Le frasi di commento sono precedute dal carattere apice '.

Per esempio:

- ‘dichiarazione delle variabili;
- Dim Età As integer “età di una persona”

As integer

Nella finestra dell'Ed/tordi Visual Basic le righe di commento sono colorate in verde. L'istruzione di assegnazione permette di attribuire un valore a una variabile e la sintassi è del tipo: variabile = espressione.

II valore dell'espressione viene assegnato alla variabile scritta a sinistra del simbolo =. Per esempio, il calcolo dell'area di un cerchio viene indicato con l'istruzione: Area = Raggio*Raggio *3.14

Per le variabili alfanumeriche, il valore da assegnare va racchiuso

tra virgolette. Peresempio: lingua = “Inglese”

Nelle espressioni possono poi comparire le funzioni, ossia sottopro-grammi predefiniti (built-in) del linguaggio che, ricevendo un valore, restituiscono un valore calcolato. Per esempio, la funzione predefinita Sqr(X) calcola la radice quadrata del numero X. Quindi, per calcolare l'ipotenusa di un triangolo rettangolo, si può scrivere un'istruzione del tipo: ipot = SQR (cat ^ 2 + cat 2^2)

II valore delle espressioni logiche può essere True o False e, quindi, il risultato del calcolo delle espressioni può essere assegnato a variabili dichiarate di tipo boolean. Per esempio, data la dichiarazione:

- DimX, Y, Z Asboolean

Si possono scrivere le seguenti istruzioni:

- X = A>B
- Y = Not B
- Z = A< B or C<D

Si osservi che la seconda istruzione è equivalente a:

Y = A<= B, perché NOT (A>B) è equivalente a A <=B.

Inputbox

Per consentire la gestione dell'input da tastiera da parte dell'utente, i linguaggio Visual Basic mette a disposizione la funzione InputBox. Tale funzione visualizza una area di dialogo standard in cui viene richiesto all'utente di immettere un valore reale.

La finestra di dialogo di InputBox contiene una casella di testo in cui l'utente agitare un valore alfanumerico e scegliere il pulsante OK /o Annulla. Se viene scelto il pulsante OK o se viene premuto il tasto Invio, la funzione InputBox restituisce la stringa digitata dall'utente. Se viene scelto il pulsante Annulla,-;:ne restituisce una stringa vuota ("").

Per esempio l'istruzione:

StringaNome = InputBox(“Inserisci il nome, “Nome”)

Assegna alla variabile StringaNome il valore inserito dall'utente tramite finestra di dialogo della funzione InputBox. Tale funzione MsgBox, invece, permette di mandare un messaggio all’utente con una finestra di dialogo predefinita contenente anche un'icona che ricorda il tipo di messaggio (errore, avvertimento, informazione) e con uno o più tra i pulsanti standard Si, No, Annulla e ?. Per esempio, la seguente istruzione produce sul video una finestra di dialogo con la frase "Fine lavoro'MsgBox "Fine lavoro", O, "Messaggio per l'utente". II secondo parametro, zero, fa comparire nella finestra un pulsante OK. Il valore zero può essere sostituito anche dalla costante predefinita vbOKOnly. La finestra di dialogo ha come titolo la frase "Messaggio per l'utente". La strutturaci selezione viene rappresentata in Visual Basic secondo lo schema:

Schema

- IF condizione THEN
- istruzionel
- ELSE
- istruzione2 ENDIF

Se la condizione è vera, viene eseguita l'istruzionel, altrimenti viene eseguita l'istruzione2. Istruzionel e istruzione2 possono indicare, come accade nella maggior parte dei casi, non una sola istruzione, ma un gruppo di istruzioni. La condizione è un'espressione booleana di cui viene valutata la verità: vengono quindi utilizzati i segni del confronto: <, >, =, >=, <=, <>, e gli operatori booleani AND, NOI, OR, XOR per costruire espressioni logiche combinando tra loro più condizioni, -a ripetizione si rappresenta in Visual Basic con la struttura DO... LOOP UNTIL:

Loop until

- DO
- istruzioni

- LOOP UNTIL condizione

La condizione deve essere un'espressione che rappresenta un valore True o False. Le istruzioni comprese tra Do e Loop vengono eseguite una prima volta, dopo di che viene verificata la condizione scritta dopo Until: se la condizione risulta vera si prosegue con l'istruzione successiva, altrimenti si ripete l'esecuzione delle istruzioni a partire dalla prima istruzione dopo Do.

La struttura di ripetizione precondizionale viene realizzata con la struttura:

- DO WHILE...LOOP
- DO WHILE condizione
- *istruzioni* **LOOP**

Le istruzioni comprese tra *Do* e *Loop* vengono ripetute mentre la condizione scritta vicino a *While* si mantiene vera.

La struttura di **ripetizione con contatore** è rappresentata con la struttura **FOR...NEXT:**

- **FOR** *contatore - iniziale* **TO** *finale*
- *istruzioni* **NEXT** *contatore*

Le istruzioni comprese tra *for e Next* vengono ripetute tante volte quante occorrono per passare dal valore *iniziale* della variabile *conta-tore* al valore *finale,* incrementando di 1 a ogni esecuzione.

6.2 – Le strutture derivate

In aggiunta alle precedenti strutture di controllo, il linguaggio Visual Basic possiede altre varianti delle strutture di ripetizione, che possono essere considerate come strutture derivate da quelle fondamentali:

- **DO UNTIL** *condizione*

- *istruzioni* **LOOP**
- **DO**
- *istruzioni*
- **LOOP WHILE** *condizione*
- **DO UNTIL** *condizione*
- *istruzioni* **LOOP**
- **DO**
- *istruzioni*
- **LOOP WHILE** *condizione*

Variabili di controllo

Dopo le parole Select Case viene indicato il nome della variabile Variabile Di Controllo (o variabile selettore) di cui si deve controllare il valore per decidere quale strada seguire tra quelle possibili.
Accanto ai valori previsti devono essere scritte le istruzioni da eseguire nel caso in cui la variabile assuma quei valori.
Nella costruzione della stringa SQL contenente il comando da eseguire, si deve fare particolare attenzione alla sintassi con l'uso delle virgolette e degli spazi.
Si osservi inoltre che, se le variabili sono di tipo numerico, possono essere concatenate senza delimitazione; per le variabili di tipo stringa occorre usare invece l'apice come delimitatore.
Per esempio, per aggiornare il telefono di un utente di cui viene fornito il cognome, si deve usare la seguente sintassi per il comando *Update* nella stringa SQL.

6.3 – Compilazione di un'applicazione eseguibile

MDE

Dopo aver completato l'applicazione in Access è possibile creare la versione eseguibile da consegnare all'utente finale attraverso la trasformazione del file.mdb in file.mete.MDE (Microsoft Database Executable) rappresenta il formato che si ottiene attraverso la compilazione del codice contenuto nel database. La creazione del

database eseguibile è possibile solo per i database creati con le versioni 2002 e 2003 di Access.

Queste versioni di Access contengono nel menu Strumenti, Utilità database, la scelta Converti database e la sottoscelta In formato Access 2002-2003 che consente la conversione verso l'alto di versioni precedenti di database. Per avviare la creazione del file MDE si deve scegliere, dal menu Strumenti, Utilità database e poi Crea copia di file MDE. Se il database contiene codice Visual Basic, la creazione del file MDE provoca la compilazione di tutti i moduli, la rimozione di tutto il codice sorgente modificabile e la compattazione del database di destinazione.

Si faccia attenzione al fatto che eventuali errori, presenti nel codice, impediscono la creazione del file eseguibile. Per questo motivo è buona norma, prima di creare il file MDE, eseguire la precompilazione del codice del database: dall'ambiente Visual Basic, si deve scegliere, nel menu Debug, la scelta contrassegnata da Compila seguita dal nome del database per individuare ed eliminare eventuali errori nel codice Visual Basic. Nel database MDE è poi possibile eseguire il codice Visual Basic, ma non visualizzarlo o modificarlo.

II database nel formato eseguibile consente di proteggere le maschere e i report da modifiche indesiderate o da utenti non autorizzati. Infatti, dopo avere salvato il database di Access come file MDE, non è più possibile effettuare le seguenti operazioni:

- modifica o creazione di maschere, report o moduli in *Visualizzazione Struttura;*
- modifica del codice, in quanto un file MDE non contiene codice sorgente;
- importazione o esportazione di maschere, report o moduli.

È possibile invece importare o esportare tabelle e query.

In ogni caso è opportuno mantenere una copia del database di partenza, perché eventuali modifiche al database e ai suoi oggetti

possono essere fatti sul file.*mdb* e richiedono una nuova compilazione per creare il file.*mde* corretto

6.4 – Accesso ai database con ADO.NET

ADO.NET è la tecnologia di accesso ai dati che rappresenta un'evoluzione di ADO con lo scopo di costruire applicazioni che siano scalabili, cioè utilizzabili con database: piccole e di grandi dimensioni.

Si differenzia da ADO, visto nei paragrafi precedenti, per gli oggetti utilizzati e per le modalità di accesso ai database, in particolare in ADO.NET non è più disponibile l'oggetto *Recordset.* Gli oggetti principali di ADO.NET per costruire applicazioni Web, con accesso ai database in rete, sono:

- Connection, per stabilire la connessione al database
- Command, per eseguire i comandi di manipolazione o interrogazione al database;
- DataReader, per ottenere i dati richiesti dal database.

I DataReader vengono poi associati ai controlli server di ADO.NET per visualizzare i dati in forma tabellare nelle pagine Web, per l'utente che utilizza il browser. I controlli server sono:

- DataGrid
- Repeater
- DataList.

Lo schema seguente riassume le modalità di interazione tra le applicazioni Web e i database secondo la tecnologia ASP. NET.

Il metodo ExecuteReader applicato all'oggetto di tipo Commandesegue il comando SQL, memorizzato in una stringa, e crea i dati all'interno dell'oggetto DataReader. Esso viene associato poi, attraverso il metodo DataBind, al controllo definito nella pagina

ASP per la visualizzazione dei dati. Per leggere e scrivere i dati occorre prima di tutto stabilire una connessione con il database attraverso un oggetto di tipo Connection che è diverso a seconda del database utilizzato: questo oggetto si chiama SqlConnection per i database SQLServer e OleDbConnection per i database /Access.
Lo spazio dei nomi (namespace):

- per il primo è System.Data.SqlClient;
- per il secondo è System.Data.OleDb.

Per questo motiveMejpagine ASP.NET che accedono ai database di Access contengono come prima riga latiichiarazione di importazione degli oggetti OleDB (namespace del Provider OleDb):

<% @Import Namespace="System.Data.OleDb" %>

DataGrid

L'esempio seguente mostra l'uso degli oggetti ADO.NET per accedere alla tabella di un database di Access. Per la visualizzazione dei dati viene utilizzato il controllo DataGrid, che si chiama così perché organizza i dati in forma tabellare con una griglia di righe e colonne.

PAGINA ASP.NET *(RicercaPerNazione.aspx)*

```
<%@ImportNamespace="System.Data.OleDb" %>

<script runat="server">
SubPage_Load
Lib1.Text="RicercaMusicisti perNazione "
EndSub
Sub submit(sender As Object, e As EventArgs)
Dim    dbconn as OleDbConnection
Dim    dbcomm as 0leDbCommand
Dim    dbread as 0leDbDataReader
```

L’alimentazione è una necessità imprescindibile per l’uomo: dagli alimenti egli ricava l’energia necessaria per tutte le attività vitali e il materiale indispensabile per la costruzione e riparazione dei tessuti. Lo stato di salute generale di un individuo dipende in gran parte dall’equilibrio tra nutrienti forniti dagli alimenti ed i reali fabbisogni dell’organismo.

UNITÀ DIDATTICA 7
DATABASE NEL WEB

7.1 – Software per Database

Il termine Web Server indica in generale il software per la gestione di un computer host che mette a disposizione dati o applicativi e al quale si possono connettere altri computer in rete. Esistono in commercio diversi prodotti software per le funzioni di Web Server: tra questi Internet Information Server (IIS), per il sistema operativo Windows NT, adatto per gestire reti locali complesse e con un elevato numero di utenti, e Personal Web Server (PWS) per i personal computer con sistema operativo Windows.

IIS

PWS

L'installazione di un Web Server sul proprio computer personale, eventualmente collegato a una rete locale, offre alcuni vantaggi importanti, permette agli sviluppatori di software per Web di creare le applicazioni Web sui loro personal computer prima di fare l'upload su Internet trasforma un personal computer di una rete locale in un server della rete Intranet nelle piccole aziende o nei dipartimenti delle grandi aziende. Per piccole organizzazioni può diventare l'Internet server, purché sia disponibile un adeguato collegamento con la rete telefonica e siano state acquisite le autorizzazioni per avere nomi di dominio per Internet.

L'uso più comune è la costruzione e la validazione di un sito Web per Internet, utilizzando il proprio personal computer, per la visualizzazione di pagine HTML (Hyper Text Markup Language) oppure per l'esecuzione di applicativi per Web, come se si usasse una connessione a un sito Internet, con gli stessi browser e le stesse modalità operative.

HTML

Il test delle pagine diventa più veloce perché si usa una connessione locale. Dopo aver verificato le funzionalità delle pagine Web si può eseguire l'upload sul server del proprio Internet Service Provider [Personal] Web Server è disponibile sul CD di installazione del

sistema operativo Windows ed è facile da installare e da configurare.
Dopo l'installazione può essere gestito dal sottomenu dei programmi di Internet Explorer. Esso inoltre, quando è attivo, è identificato con un'apposita icona che compare nella barra delle applicazioni in basso a destra sul desktop di Windows trasforma un personal computer di una rete locale in un server della rete Intranet nelle piccole aziende o nei dipartimenti delle grandi aziende e per piccole organizzazioni può diventare l'Internet server, purché sia disponibile un adeguato collegamento con la rete telefonica e siano state acquisite le autorizzazioni per avere nomi di dominio per Internet.
L'uso più comune è la costruzione e la validazione di un sito Web per Internet, utilizzando il proprio personal computer, per la visualizzazione di pagine HTML (Hyper Text Markup Language) oppure per l'esecuzione di applicativi per Web, come se si usasse una connessione a un sito Internet, con gli stessi browser e le stesse modalità operative.
Il test delle pagine diventa più veloce perché si usa una connessione locale. Dopo aver verificato le funzionalità delle pagine Web si può eseguire l'upload sul server del proprio Internet Service Provider [Persona] Web Server è disponibile sul CD di installazione del sistema operativo Windows ed è facile da installare e da configurare.
Dopo l'installazione può essere gestito dal sottomenu dei programmi di Internet Explorer. Esso inoltre, quando è attivo, è identificato con un'ap-posita icona che compare nella barra delle applicazioni in basso a destra sul desktop di Windows.
Per fissare le proprietà delle cartelle di Personal Web Server.
Fare clic con il pulsante destro del mouse sull'icona della cartella C:\WEBSHARE\ WWWROOT per aprire il menu di scelta rapida e scegliere Condivisione.
Nella finestra che si apre, selezionare la linguetta Condivisione. Condividere la cartella con il nome WWWROOT e impostare i permessi di accesso come sola lettura o accesso completo.

Fare clic sulla linguetta Condivisione Web e verificare che la condivisione per HTTP (HyperText Transfer Protocol) sia di sola lettura. Ripetere la procedura anche per la cartella

C:\WEBSHARE\SCRIPTS.

Per essa occorre verificare che nella condivisione per HTTP sia fissata l'opzione Esegui procedure.

Dopo l'installazione è opportuno riavviare il computer per assicurarsi che venga attivato il Web Server con i servizi HTTP. In seguito Persona] Web Server verrà avviato automaticamente a ogni riavvio del computer.

Personal Web Server

Per verificare che il Web Server funzioni, si può utilizzare un computer connesso in rete per identificare del computer su cui è installato il Personal Web Server. Per controllare che il server Web sia attivo, basta indicare nella casella indirizzo:

http://nomecomputer, dove nome computer indica il nome con il quale il computer con il Web Server viene identificato.

Questo indirizzo WWWROOT. che è la home page creata nella cartella per pagine HTML.

Per avviare o interrompere Personal Web Server:

1. fare doppio clic sull'icona Personal Web Server nella barra delle applicazioni;
2. nella finestra che si apre fare clic sul pulsante Avvia se in quel momento è interrotto o sul pulsante Interrompi se è attivo.

Per visualizzare un qualsiasi altro file, basta scrivere un indirizzo del tipo http://nomecomputer/prova.htm dove prova indica il nome di un file qualsiasi in formato HTML.

Per eseguire uno script presente nella cartella Scripts, si deve scrivere l'indirizzo http://nomecomputer/scripts/nomescript dove nomescript indica il nome di un file qualsiasi eseguibile in ambiente Web.

Scripts

Il nome, le caratteristiche e le proprietà del Personal Web Server

possono essere visualizzate facendo doppio clic sulla sua icona nella barra delle applicazioni del desktop (in basso a destra).

Se si vuole impedire temporaneamente agli altri utenti connessi in rete di accedere alle risorse del Web Server, basta interromperne l'attività facendo clic sul pulsante Interrompi all'interno della finestra del Personal Web Server. Si può accedere alle cartelle create dal Personal Web Server anche dal computer su cui esso e installato con le stesse modalità viste in precedenza.

Local host

In questo caso il nome del computer può essere sostituito con local host; per esempio, per vedere la home page dal browser, basta fornire l'indirizzo http://localhost essendo local host il nome che identifica il server Web della macchina locale.

Il server local host corrisponde all'indirizzo IP 127.0.0.1. In cui compare una pagina di test, che conferma l'attivazione del server Web. L'installazione del Web Server su un personal computer consente di accedere a pagine in formato HTML, a script eseguibili e a basi di dati condivise, con le stesse modalità e gli stessi prodotti software che vengono normalmente utilizzati per la connessione ai siti Internet: è quindi lo strumento più semplice per realizzare una Intranet aziendale.

Le pagine e gli eseguibili devono essere registrati nelle cartelle create in fase di installazione od organizzati in sottocartelle di WWWROOT, per essere accessibili con indirizzi HTTP standard.

Esempio

Esempio

Creare un piccolo sito Web formato da alcune pagine e verificarne la visualizzazione tramite il browser.

Il sito Web è costituito in pratica da una sottocartella creata all'interno della cartella C:\WEBSHARE\WWWROOT. Creiamo quindi una nuova cartella in WWWROOT con il nome Primo Web. Costruiamo poi una prima pagina Web che deve svolgere il ruolo di Home Page, inserendo un titolo, qualche riga di testo ed eventualmente un'imma-

gine. In fondo alla pagina inseriamo anche un collegamento ipertestuale ad un'altra pagina di nome Pag2.htm.

Home page

La Home Page deve essere salvata, in formato HTML, nella cartella Primo Web con il nome Default. htm, in modo che possa essere riconosciuta come pagina di partenza del sito Web dal programma browser. Creiamo poi un'altra pagina contenente un collegamento ipertestuale al file Default. htm, per dare la possibilità all'utente di tornare alla Home Page del sito Web. Salviamo anche questa seconda pagina in formato HTML con il nome Pag2.htm.

A questo punto possiamo verificare le pagine Web utilizzando il programma browser. Nella casella dell'indirizzo scriviamo:

http://localhost/Primo Web

Il browser cerca la Home Page nella sottocartella Primo Web del Personal Web Server residente sul computer e visualizza il file Default. htm.

Se invece si vuole visualizzare direttamente la pagina Pag2.htm, occorre specificare l'indirizzo:

http://localhost/PrimoWebiPag2.htm

Le stesse operazioni possono essere eseguite da qualsiasi altro computer connesso alla stessa rete del computer, sostituendo local host con il nome direte del computer sul quale è installato il Personal Web Server; per esempio:

http://pc 1/Primo Web

essendo pc1 il nome con il quale il computer, contenente il sito Primo Web, è identificato all'interno della rete.

Questo semplice esempio fornisce un'idea pratica di cosa significa utilizzare le risorse di una rete Intranet attraverso le tecniche e i protocolli tipici di Internet.

In informatica il termine driver indica un programma che consente ad

una periferica hardware di comunicare con l'unità centrale di un computer: il programma driver ritrova i dati richiesti e li trasferisce al programma richiedente.

Il driver per database è un programma standard che consente di trasferire i dati presenti in un database: quando un programma vuole accedere al database creato con un altro applicativo, deve utilizzare il driver per quel database.

ODBC

ODBC (Open Database Connectivity) è l'interfaccia software standard in ambiente Windows che consente ai programmatori di scrivere in modo semplice le applicazioni per connettersi ai database e ritrovare i dati in essi contenuti. I database possono essere creati con prodotti DBMS diversi: ODBC possiede i driver software per database ed evita quindi di dover scrivere un programma diverso per ciascun tipo di database.

Data Source Name

La sigla DSN (Data Source Name) indica il nome della sorgente di dati; è il nome che identifica il database quando viene utilizzato da un'applicazione Web in Internet o in una Intranet aziendale. Per visualizzare le proprietà di ODBC e per fissarne le impostazioni eseguire:

ODBC

1. fare clic sul pulsante Start, scegliere Impostazioni, quindi Pannello di controllo e infine fare doppio clic sull'icona Fonti dati ODBC (32 bit);
2. fare clic sulla scheda DSN utente, DSN di sistema o DSN su file, a seconda del tipo di origine dati;
3. per definire una nuova origine dati per un driver installato, scegliere il pulsante Aggiungi;
4. per modificare la definizione di un'origine dati esistente:
   - selezionare l'origine dati dal l'elenco.
   - scegliere il pulsante Configura.
   - completare le finestre di dialogo.

L'origine dei dati può essere:

- DSN utente (User DSN) per fissare una sorgente di dati per uno specifico utente DSN di sistema (System DSN per le sorgenti di dati disponibili per molti utenti che accedono allo stesso computer di rete.
- DSN su file (File DSN) per configurare un DSN come file con estensione.dsn che può essere condiviso da più utenti che dispongono dello stesso driver installato.

In ambiente Windows si possono utilizzare anche altre tecnologie per stabilire la connessione delle applicazioni con un database.

Tecnologia OLE DB

La tecnologia OLE DB è simile a ODBC, ma consente, in aggiunta, di accedere anche ad altri tipi di dati, diversi da quelli standard organizzati in un database costituiti da campi stringa e numerici. Con essa infatti si possono anche trattare documenti multimediali e messaggi di e-mail. OLE DB è comunque completamente compatibile con ODBC, nel senso che un'applicazione può accedere a database con i driver ODBC utilizzando le funzioni di OLE DB.

Gli oggetti ActiveX per l'accesso ai dati, indicati con la sigla ADO (active data objects) sono oggetti software che contengono al loro interno le funzioni di OLE DB, offrendo a programmatore la possibilità di operare con istruzioni di alto livello nelle applicazioni e rendendo quindi più semplice la connessione al database.

7.2 – Pubblicare i dati con pagine statiche e dinamiche

Pagine statiche e pagine dinamiche

L'elaborazione dei dati organizzati in un database può rispondere anche all'esigenza di mettere a disposizione informazioni per gli utenti collegati ad una rete aziendale Intranet oppure alla rete Internet.

Access offre la possibilità di realizzare, in modo facile, pagine Web che contengono de provenienti da un database. Si possono pubblicare tutti gli oggetti di Access: tabell, query, maschere e report.

La pubblicazione può essere di tipo:

- statico, quando i dati estratti dal database diventano il contenuto di una pagina Web cd viene consultata dagli utenti tramite browser: questo riguarda soprattutto i dati ci variano con minore frequenza o i dati consolidati;
- dinamico, quando l'utente ha la possibilità di vedere i dati aggiornati di un database cI sta all'interno di un sito Internet, attraverso interrogazioni che vengono eseguite r momento in cui effettua la connessione alla pagina Web.

Per creare una pagina Web con i dati in formato HTML statico necessitano i seguenti passi:

1. nella finestra del database fare clic sul nome della tabella, della query, della maschera o del report che si desidera esportare, quindi scegliere Esporta dal menu File;
2. nella casella di riepilogo Tipo file selezionare Documenti HTML (*.html;*.htm). Scegliere il disco o la cartella di destinazione con Salva in;
3. assegnare un nome al file nella casella Nome file. Selezionare la casella di controllo Salva formattato: in questo modo nel browser Web i report vengono visualizzati nel formato report, mentre tabelle, query e maschere vengano visualizzate nel formato Foglio dati di Access;
4. selezionare la casella di controllo Avvio automatico se si desidera visualizzare i risultati nel browser Web predefinito. Fare clic sul pulsante Salva. Per il salvataggio in modo formattato si può anche scegliere opzionalmente un modello di pagina HTML predefinito, se disponibile. A differenza del foglio dati, un report viene esportato come un insieme di più file HTML, con un file per ogni pagina stampata. I nomi dei file vengono creati utilizzando il nome dell'oggetto e un suffisso. per esempio Clienti. html, ClieritiPagina2.htm, Clienti

Pagina3.html e così via. Nelle pagine Web vengono creati anche in modo automatico bottoni di navigazione tra le diverse pagine del report. Se l'oggetto da esportare (query, maschera o report) contiene una query con parametri, vengono richiesti i valori dei parametri ed esportati i risultati. Le pagine Web ottenute con il formato statico rappresentano gli output delle visualizzazioni, delle query o dei report con i dati esistenti al momento della creazione dei file HTML. Non essendoci nessun legame con il database, in caso di modifiche al database, è necessario effettuare di nuovo l'esportazione per poter visualizzare i nuovi dati con un browser. Per pubblicare un file HTML statico su un server Web, occorre poi copiare il file HTML statico in una sottocartella della cartella principale del server Web. Le pagine dinamiche vengono generate dal server nel momento in cui l'utente formula la richiesta di visualizzazione di una tabella o di esecuzione di una query, anche parametrica. Access è in grado di creare due tipi di pagine dinamiche: il file IDC/HTX (Internet Database Connector /HyperText) la pagina ASP Active Server Page eFile HTML.

Creazione pagine dinamiche

È possibile creare file HTML in modo dinamico a partire da tabelle, query e maschere. Non è possibile invece creare pagine dinamiche con i report. L'uso di file ASP o IDC/HTX corrisponde alla necessità di ottenere pagine Web con dati provenienti da una sorgente di dati ODBC, che vengono modificati spesso. Ogni volta che un utente apre o aggiorna un file ASP o IDC/HTX da un browser Web, il server Web crea un file HTML in modo dinamico, quindi lo invia al browser.
Per utilizzare file HTML generati dal server (ASP o JDC/HTX):

1. creare sotto la cartella principale predefinita (C:\ Webshare\ Wwwroot per Personal Web Server) una cartella in cui memorizzare i file ASP o IDC/HTX.;
2. definire i permessi per la cartella: Sola lettura, Esegui proce-

dure, Esegui script;

3. copiare nella cartella i file ASP o IDC/HTX, oltre ai file HTML o eventuali file di immagini collegati ad essi.

Il database, contenente i dati che servono, può essere copiato in questa cartella oppure si può fare riferimento ad esso specificandone il percorso4. Definire l'origine dati ODBC come DSN di sistema o come DSN utente sul server Web, secondo la procedura illustrata in precedenza. In particolare occorre fare attenzione alla scelta del nome logico assegnato all'origine dati ODBC, perché deve essere lo stesso nome specificato al momento della creazione dei file ASP o IDC/HTX.

Poiché l'uso delle pagine dinamiche presuppone la specificazione di una fonte dati, nel caso in cui si voglia pubblicare il database sul server di una rete locale o sul server di un Es Internet Provider, occorre prendere accordi con l'amministratore di rete per rendere disponibile il database a tutti gli utenti della rete o a quelli che si collegano tramite Internet.

Con la prima tra le modalità di creazione di pagine dinamiche viste nel paragrafo prece dente, Access crea un file IDC e un file HTX insieme, per ottenere dati da un'origine dati ODBC e per formattarli come documento HTML.

Il file IDC è un file di testo che contiene informazioni su come connettersi a un'origine dati ODBC e un'istruzione SQL da eseguire.

File IDC

Il file HTX è un file HTML che contiene un modello di pagina Web, con parole chiave che controllano la formattazione dei dati e segnaposti che specificano dove i dati devono essere inseriti nel documento HTML quando vengono restituiti dall'origine dati, come specificato nell'istruzione SQL del file IDC.

In aggiunta viene anche creata una pagina HTML per fornire valori da tastiera per query di tipo parametrico.

Per creare pagine dinamiche in formato IDC/HTX:

1. nella finestra del database fare clic sul nome della tabella, della query o della maschera che si desidera esportare, quindi scegliere Esporta dal menu File;
2. nella casella Tipo file selezionare Microsoft IIS 1-2 (*.htx;*.idc);
3. scegliere l'unità o la cartella di destinazione nella casellaSalva in;
4. specificare il nome del file nella casella Nome file;
5. fare clic sul pulsante Salva;
6. immettere le informazioni appropriate nella finestra di dialogo Opzioni di output:
   - nella casella Modello HTML immettere eventualmente il percorso di un modello HTML, se esistente;
   - nella casella Nome origine dati immettere il nome logico del database assegnato come DSN nell'origine dati ODBC;
   - nelle caselle Utente connesso nome e Password utente immettere un nome utente e una password che consentano agli utenti di accedere al database di Access dalla pagina Web Se non si immette un nome utente e una password, verrà utilizzato il nome utente predefinito Amministratore senza password.

Se il foglio dati contiene una query con parametri, Access simula la finestra di dialogo Immissione valore parametro delle query parametriche e crea una pagina HTML aggiuntiva che contiene una casella di testo di un form HTML per immettere i valori dei parametri e un pulsante di comando per eseguire la query. Esempio Creare una pagina Web dinamica per ottenere la lista dei clienti di una città fornita dall'utente al momento dell'esecuzione della query.
In Access occorre definire una query parametrica che visualizzi il nome della società. l'indirizzo e la città dei clienti appartenenti ad una

città prefissata e scelta dall'utente al momento dell'esecuzione. Nella riga dei criteri della query, sotto la colonna del campo Città, si scrive la domanda che verrà posta all'utente per la richiesta della città nella finestra Immissione valore parametro: [città]. Si salva poi la query con il nome Query.Utilizzando la procedura presentata in precedenza per l'esportazione di dati in formato IDC/HTX, si sceglie Query come oggetto da esportare: Access crea automaticamente tre file con lo stesso nome e con tre estensioni diverse: html, hix e idc. Il loro contenuto può essere visualizzato usando Blocco Note di Windows. Il primo file HTML viene creato perché la query è parametrica e contiene un form per acquisire la città da controllare.

L'azione associata al form è la chiamata del file IDC.

Queryl,HTML

<HTML>

<HEAD>

<NErA HTTP-EQUIV= Content-Type' CONTENT= text/html;char set=windows-1252> <TITLE> Query</TITLE>

<BODY>

<FORM METI-IOD= GET ACTION= Queryl. IDC >

[ciUtà <INPUT TYPE=Text NANE- [città]><P> <INPUT TYPE=Submit VALUE =Esegui query>

</FORM>

</BODY>

</HTML>

Il secondo file (IDC) contiene le informazioni sull'origine dei dati e sui comando SQL che realizza la query. Se specificati al momento dell'esportazione da Access, in questo file compaiono anche il nome utente e la password. Il valore fornito dall'utente per la città è una stringa; questo giustifica l'uso degli apici nella condizione scritta dopo Where nel comando SQL:

(Clienti.Città) = % città]%.

Il secondo membro della condizione riceve il valore dalla casella di testo del form HTML presente nel precedente file Query1HTML.
Si osservi attentamente che come Datasource viene specificato il nome logico dbl assegnato come DSN dell'origine dati ODBC (Pannello di controllo di Windows, Fonte dati ODBC).
Il file IDC richiama a sua volta il file HTX come Template, cioè modello, della pagina Web da creare in modo dinamico.

```
Queryl.IDC
Datasource: dbl
Template Queryl. HTX
SQL Statement:SELECT Clienti.MomeSocietà, Clienti.Indirizzo,
Clienti.Città
+FROM Clienti
+WHERE (((Clienti.Città)=%{quale città]%));
Password:
Username:
```

Il terzo file (HTX rappresenta il modello in formato HTML della pagina Web che deve essere creata dal Web Server. I dati vengono presentati in forma tabellare con un titolo per la tabella e un'intestazione per ogni colonna. Nella seconda parte del file compare la descrizione delle righe di dettaglio con l'indicazione dei segnaposto, racchiusi tra una coppia di caratteri %, uno per ciascun valore da visualizzare sulla stessa riga della tabella (nell'esempio, %NomeSocietà %, %lndjrjzzo%, %Città%).

```
Query i.HTX
<HTML>
<HEAD>
<META HTTP-EQUIV= 'Content-Type CONTENT= ' text/html;
charset =windows- > <TITLE>Queryl< /TITLE>
</HEAD>
<BODY>
```

<TABLE BORDER = l BGCOLOR=#ffffff CELLSPACIMG=O><FOMT FACE= ‘Arial
COLOR=
<TBEAD>
<TR>
<TR BGCOLOR=#cOcOcO BORDERC0L0R= ><FCNT SIZE=2 FACE =Arial COLOR=<TR BGCODOR=#cOcOcO BORDER COLOR=#000000 ><FOMT SIZE=2 FACE=Arial COLOR=
<TR BGCOLOR=#cOcOcO BORDERCOLOR=#000000 ><FONT SIZE=2 FACE=L COLOR=
</TR>
</THEAD>
<TBODY>
<%Begin Detail%>
<TR VALIGN =TOP>
<TD BORDERCOLOR=#cOcOcO ><FONT SIZE=2 FACE=’Arial COLOR= # 000000 ><%NomeSocietà%>
< /FONT>< /TD>
<TD BORDERCOLOR= ><FONT SIZE=2 FACE= Arial COLOR=0000000><%Indirizzo%>
</FONT></TJD>
<TD BORDERCOLOR=#cOcOcO ><FONT SIZE-2 FACE= Arial’ COLOR=# 000000 ><%Città%>
< /FONT>< /TD>
<%EndDeta 1%> </TBODY>
<TFOQT></TFOOT>
</TABLE>
</BODY>
</HTML>

Supponiamo di aver creato una sottocartella WebClienti nella cartella principale (T/VWWROOT) del Web Server e di aver trasferito in essa i file Queryl.1-UML, Queryl.IDC e Queryl.HTX generati automaticamente da Access. Dopo aver stabilito il permesso di esecuzione sulla cartella Web Clienti, è possibile eseguire la query dal browser

scrivendo nella riga dell'indirizzo:

http://localhost/WebClienti/Query i.HTML

sostituendo eventualmente il nome localhost con il nome del computer contenente i file, se si vuole eseguire la stessa interrogazione da un computer diverso e connesso alla rete.

Dopo aver fornito la città cercata, il Web Server lancia l'esecuzione del programma IDC che. a sua volta, esegue il comando SQL e genera la pagina Web in modo dinamico. usando il file HTX come modello.

7.3 – Le pagine ASP

Le pagine ASP (Active Server Page) consentono l'uso di pagine dinamiche che prelevano dati da un'origine dati ODBC e li presentano in una pagina con formato HTML pronta per essere visualizzata con un qualsiasi browser. Il file ASP contiene al suo interno script per il Web Server che specificano come connettersi all'origine dati e i tag HTML per formattare i dati restituiti. Quando si esportano query o maschere di Access, il file ASP contiene anche controlli ActiveX e codice VBScript, cioè codice per costruire script formato da parole chiave del linguaggio Visual Basic.

Pagine ASP

Visual Basic Script

Per creare pagine dinamiche in formato ASP

1. nella finestra del database fare clic sul nome della tabella, della query o della maschera che si desidera esportare, quindi scegliere Esporta dal menu File;
2. nella casella Tipo file selezionare Microsoft Active Server Pages (*.asp);
3. scegliere l'unità o la cartella di destinazione nella casella Salva in;
4. specificare il nome del file nella casella Nome file;
5. fare clic sul pulsante Salva;
6. immettere le informazioni appropriate nella finestra di dialogo

Opzioni di output:

- nella casella Modello HTML immettere eventualmente il percorso di un modello HTML, se esistente;
- nella casella Nome origine dati immettere il nome logico del database assegnato come DSN nell'origine dati ODBC;
- nelle caselle Utente connesso come e Password utente immettere un nome utente e una password che consentano agli utenti di accedere al database di Access dalla pagina 'Net. Se non si immette un nome utente e una password, verrà utilizzato il nome utente predefinito Amministratore senza password.

Questi parametri sono identici a quelli richiesti per le pagine in formato IDC/HTX. Per le pagine ASP ci sono due informazioni aggiuntive da specificare:

- URL del server, cioè l'indirizzo HTTP del sito Web dove occorre pubblicare le pagine per esempio: http://local host/WebClienti
- Timeout sessione (minuti), per esempio I minuto, cioè il tempo dopo il quale viene interrotta l'elaborazione delle pagine Web, per non creare lunghe e inutili attese per l'utente, nei casi in cui il Web Server non riesca a concludere positivamente la richiesta di dati.

Anche per le pagine ASP, se il foglio dati contiene una query con parametri, Access simula la finestra di dialogo Immissione valore parametro delle query parametriche creando una pagina HTML aggiuntiva che contiene una casella di testo di un form HT per immettere i valori dei parametri e un pulsante di comando per eseguire la query Viene riproposto l'esempio già visto per le pagine dinamiche in formato IDC/HTX, realizzato con una pagina ASP.
Creare una pagina Web dinamica informato ASP per ottenere la lista

dei clienti di u; città fornita dall'utente al momento dell'esecuzione della query.

Query

Per questo esempio si fa riferimento alla stessa Query di Access dell'esempio nel pari grafo precedente. Utilizzando la procedura per l'esportazione di dati in formato ASP sceglie Queryl come oggetto da esportare: Access crea automaticamente due file, ai qua diamo per comodità il nome Query2, con due estensioni diverse: HTML e ASP. Il loro contenuto può essere visualizzato usando Blocco Note di Windows.

La pagina ASP raggruppa in un unico file sia le istruzioni per acquisire i dati dall'origine ODBC, sia i tag per formattare i dati ottenuti in una pagina HTML.

Le righe degli script sono racchiuse tra una coppia di <%... %> e contengono parole chiave del linguaggio Visual Basic (VBScript).

Nella prima parte viene stabilita una connessione con il database, indicato con il nome logico dhl (DSN di ODBC); si costruisce poi un Recordset (indicato con rs) utilizzando il comando SQL. Esso contiene al suo interno il parametro [città], che ottiene il valore dalla casella di testo del form contenuto nel file Query2.HTML.

Nella seconda parte sono contenuti i tag per costruire la pagina HTML in modo dinamico: i dati vengono presentati in forma tabellare con un titolo per la tabella e le intestazioni per le colonne. Leggendo poi sequenzialmente il Recordset ottenuto dall'interrogazione SQL, le righe della tabella vengono riempite con i valori dei campi richiesti dal problema. Si noti l'uso dei metodi MoveFirst, MoveNext e delle strutture di controllo If... Then Else e Do While... Loop, già viste nel linguaggio Visual Basic.

Query2.ASP

ahlIIL>

<HEAID>

<MFTg HTTP EQUIV=ConLent-Type CONTENT-:ul;:

<TITLE>Query2</TITLE>

```
a/HEAD>
<BODY>
<I
Il lsObject(Session( ‘dbl conn ) ) Jien Set conn - Session(
Else
Ee conn = Sereer.CreateObjectLACDB.Connection
mn. open dbl ‘,
Set Session( dblconn)
%>

If IsObject(Session(Querylrs)) Then
Set rs = Session(’Queryl_rs)
Else
sql = SELECT Clienti.NomeSocietà, Clienti.Indirizzo, Clienti.Città
FROM Clienti WHERE Clienti.Città =
& Request.QueryString( [ città] u) &
Set rs = Server.CreateObject( ADODB.Recordset)
rs.Open sql, conn, 3, 3
If rs.eof Then
rs AddNew
Endlf
Set Session(Queryl_rs) = rs
End Il
%>
<TABLE BORDER=l BGCOLOR=#ffffff
CELLSPACING=O><FONT FACE= ‘Arial
COLOR=#000000><CAPTION>
<THEAD>
<TR>
<TH BGCOLOR=*cOcOcO BORIDERCOLOR=#000000 ><FONT
SIZE=2 FACE=’Arial COLOR=#000000>NomeSocietà
<TH BGCOLOR= BORDERCOLOR=#000000 ><FONT SIZE=2
```

```
FACE=Arial' COLOR=#000000>Indirizzo
<TH BGCOLOR=#cOcOcO BORDERCOLOR=#000000 ><FONT
SIZE=2 FACE='Arial' COLOR=#000000>Ci
</THEAID>
<TBODY>
On Error Resume Next rs.MoveFirst do while Not rs.eof
%>
<TR VALIGN=TOP>
<TD BORDERCOLOR=#cOcOcO ><FONT SIZE=2 FACE=Arìal
COLOR=#000000><%=Server.HTMLEncode(rs.Fields(NomeSocietà
).Value)%><BR> FONT>< /TD>
<TD BORDERCOLOR=#cOcOcO ><FONT SIZE=2 FACE=Arial'
CQLQR= Indirizzo).Value)%><BR>< FONT></TD>
<TD BORDERCOLOR=#cOcOcO ><FONT SIZE=2 FACE=Arial'
COLOR= FONT>< /TD>
rs.MoveNex loop%> </TBODY> <TFOOT>< /TFOOT> </TABLE>
</BO]DY>
</HTML>
```

Nella seconda parte della condizione scritta dopo There nel comando SQL, il valore della città richiesta, proveniente dal form HTML, essendo una stringa. deve essere racchiuso tra una coppia di apici (il simbolo & indica la somma di stringhe):

- sql = 'SELECT Clienti.NomeSocietà, Clie CIie FROM Clienti WHERE Clienti.Città - R [città]) &

In modo analogo a quanto visto per i file IDC HTX, dopo aver trasferito nella sottocartella WebClienti di WWWROOT nel Personal Web Server i file Query2.FJTML e Query2.ASP generati automaticamente da Access, si può eseguire la query dal browser scrivendo nella riga dell'indirizzo:

http://localhost/WebClienti/Query2.HTML

Se si sostituisce il nome localhost con il nome del computer contenente i file, si può eseguire la stessa interrogazione da un computer diverso e connesso alla rete. L'esecuzione della pagina ASP provoca l'attivazione di una sessione di lavoro da parte del Web Server. Durante questa sessione i dati del Recordset, costruito con l'interrogazione SQL, sono conservati nella cache del Web Server. La cache è una memoria di lavoro del server che conserva pagine e immagini già visitate durante le connessioni al server, per rendere più veloci la visualizzazione della pagina Web nel caso di richiesta dello stesso documento.

Questo significa che si può specificare il parametro per la query una sola volta per ogni sessione.

Query parametrica

Per usare la query parametrica più volte nella stessa sessione Occorre introdurre manualmente alcune modifiche ai file creati da Access (essi sono file di testo modificabili con qualsiasi Editor, per esempio Blocco note di Windows):

- rinominare il file Query2.HTML, contenente la richiesta del parametro per la query, in Param2.ASP facendolo diventare un file di tipo ASP;
- all'inizio del file Param2.ASP inserire la seguente riga di comando per chiudere la sessione corrente:

<% Session.Abandon %>

- Alla fine del file Query2.ASP inserire un link per tornare alla pagina di richiesta del parametro; la chiamata della pagina provo-ca infatti la riesecuzione della pagina, con la richiesta del para-metro, da parte del server. Le ultime righe del file Query2.ASP devono essere modificate in questo modo:
- </TBODY>
- <TFOOT>< /TFOOT> </TABLE><P>
- <A HREF=Param2.ASP > alla scelta della città e/A> <I BO LA >

- </HTML>

In alternativa si può usare il pulsante Indietro del browser per tornare alla pagina prece dente e poi il pulsante Aggiorna per far rieseguire al server la pagina con la scelta del parametro.

Esempio

Esempio: creare una pagina l4Teb per inserire nuovi nomi nel database di una rubrica telefonica.

Si abbia a disposizione, in un database db2.mdb, una tabella Anagrafe con la struttura seguente:

- Nome
- Tipo
- ID
- Cognome
- Nome
- Telefono
- Contatore
- Testo
- Testo
- Testo
- chiave

Si vuole costruire una pagina Web che consenta di inserire nuovi record nella tabella Anagrafe.

L'esempio mostra la costruzione di pagine ASP, senza utilizzare la generazione automatica dell'esportazione da Access. I testi delle pagine possono essere scritte con un qualsiasi Editor di testi e salvati con estensione ASP.

Applicazione

L'applicazione si compone di due file: la pagina Web in formato HTML con un form per acquisire i dati da tastiera Ingresso.H7Ti una pagina ASP per inserire i dati nel database (AdcASP].

Ingresso.HTM

```
<HTNL><HEAD>
<TITLE>Inserimento di record </TITLE> </HEAD>
<BODY>
<H3 ALIGN= center>Inserimento nuovi record
<FORM METHOD= post NAME= formi ACTION— Add.ASP'>
<P>< STRONG>Cognome / STRONG><BR>
<INPUT TYPE- SIZE= 4O NAME=Cognome> <STRONG>Nome< /
STRONG><BR>
<INPUT TYPE='Lext SIZE= 40 NAME= Nome ><BR>
<STRONG>Telefono / STRONG>
<INPUT TYPE='text SIZE='40 NANE= Telefono ><BR>
<P> TYPE=Submit VALUE= Invia' NId4E </FONT> </FORM>
a/TR>
</TABLE>
</BODY></HTML>
```

Pagina Web

La pagina Web contiene un form con tre caselle di testo per i tre campi del record e - pulsante Invia che attiva la pagina ASP. La chiave ID del record non viene richiesta perché, essendo di tipo Contatore, si incre-menta automaticamente ad ogni inserimento:di un nuovo record.

Pagina ASP

La pagina ASP contiene una parte di script racchiusi tra una coppia di <%... %>, che utilizzano parole chiave del linguaggio Visual Basic (VBScript) e un'insieme di tag HTML per i messaggi all'utente. Dopo la registrazione del record, la pagina ASP produce sul video un sommario dei dati inseriti e un link per tornare alla pagina di inserimento dei dati. Per la connessione al database viene creato un oggetto Connection per il collegamento al database e un oggetto Recordset con i dati della tabella Anagrafe contenuta nel database db2.mdb.

La connessione all'origine dei dati è realizzata senza DSN (DSNless), costruendo una stringa di connessione al momento dell'esecuzione: si

deve specificare il tipo di driver [questo caso quello per i file mdb di Access) e il nome fisico del database (db2.mdb).

Le frasi con l'apice iniziale sono righe di commento (come nella sintassi di Visual Basic) e servono a documentare le diverse parti della pagina ASP

Add.ASP

```
<%
Dichiarazione delle variabili locali
dim con dim rs
dim strlD
dio strconn
'Stringa di connessione a origine dati
strconn = 'DRIVER=Microsoft Access Driver (*.mdb);DBQ= &
Server.MapPath( db2.mdb')
Oggetto Connection
set conn = server.createobject(ADODB.connection
conn.open strconn
Oggetto Recordset
set rs - server.createobject( ADODB.recordset Apre la tabella
rs,open Anagrafe, cono, 2, 2
Metodo AddNew per i oggetto Recordset per aggiungere un record
rs. AddNew
Inserisce nei campi del record i dati acquisiti con il form
rs( Cognome) – request ( Cognome)
rs( Nome ) - request('Nome)
rs( Telefono ) = request(
rs Update
Metodo MoveLast per ottenere il valore di ID generato automaticamente rs.MoveLast
str = rs(
%>
<HTML>
```

<HEAD>
<TITLE>Riassunto dei dati inseriti</TITLE> </HEAD>
<BODY>
<CENTER> assunto dei dati inseriti <P>Numero Record <% - strlD %>
<P>Cognome <% = request(%> <P>Nome <% = request(Nome') %> <P>Telefono <% = request("Telefono %>
<P><A HREF= alla pagina di inserimento</A> </BODY> </HTML>
'Rilascia gli oggetti
set rs - nothing
set cono = nothing

Si noti, anche in questo testo di pagina ASP, l'uso della sintassi e delle parole chiave di Visual Basic per la scrittura degli script. Sono presenti anche i metodi AddNew, Update. MoveLast applicati a un oggetto di tipo Recordset con le stesse modalità operative già viste nella programmazione visuale presentata nell'unità didattica precedente. Un terzo esempio mostra come si possa realizzare lo stesso risultato dell'esercizio precedente per l'inserimento di nuovi record nella tabella Anagrafe, ma attraverso un comando SQL.

Esempio web con SQL

Esempio Creare una pagina Web per inserire nuovi nomi nel database di una rubrica telefonica. utilizzando un comando SQL.
Il database, la tabella e il tracciato record sono gli stessi dell'esempio precedente.
Anche in questo caso l'applicazione viene creata senza la generazione automatica di Access ed è formata da due file: la pagina Web in formato HTML per l'inserimento dei dati [una pagina ASP per la registrazione dei dati nel database.
Ingresso2.HTM
<HTML><HEAD>
<TITLE>Inserimento di record con SQL</TITLE> </HEAD>
<BODY>

<H3 ALIGN= nuovi record</H3> <FORM METHOD-post" NAME= forml ACTION='AddSql.ASP> <P><STRONG>Cognome< / STRONG>
<INPUT TYPE="text SIZE="40" NAME="Cognome>
 <STRONG>Nome< / STRONG>
<INPUT TYPE='text' SIZE= 40' NAME="Nome <STRONG>Te le fono< I STRONG>
<INPUT TYPE='text" SIZE= 40 NAME= >
 <P><INPUT TYPE- VALUE= "Invia NANE" bl'> </FO 1> </FORM>
</TABLE>
</BODY></HTML>

La pagina Web è del tutto simile alla precedente ed è stata riscritta solo per comodità di lettura: l'unica differenza consiste nel nome del file ASP che viene attivato quando l'utente fa un clic sul pulsante Invia (in questo caso AddSql.ASP].
Il testo della pagina ASP è riportato sotto.

ASP

AddSqLASP
Dichiarazione delle variabili locali dim conn
dim strconn dim strSQL strsql =stringa di connessione al database con driver ODBC
strconn - "DRIVER=Microsoft Access Driver (*mdb).DBQ. &
Server.MapPath().
costruzione del comando SQL con gli input del form strSQL - INSERT INTO Anagrafe(Cognome, Nome, Telefono) strSQL = strSQL & 'VALUES ('

SLQ

strSQL = strSQL & ''' & request(& '',' strSQL = strSQL & "' & request) & "', strSQL = strSQL & " & request(& strSQL = strSQL &
Oggetto connection
set conn = server.createobject(
conn.open strconn

Metodo Execute dell'oggetto Connection per inserire il record conn. execute (strSQL)
conn. dose set conn = nothing
%>
<HTML>
<HEAD>
<TITLE>Esempio di inserimento d± record con SQL</TITLE> </HEAD>
<BODY><H2>
= "Il record è stato registrato %></H2>
<P><A HREF='Ingresso2.HTM'>Torna alla pagina di inserimento</A> </BODY>
</HTML>

In questo secondo caso si ha una connessione all'origine dei dati senza specificazione del DSN come prima, ma non viene usato alcun oggetto Recordset, in quanto la registrazione di un nuovo record nel database si realizza applicando all'oggetto Connection il metodo Execute, il quale lancia l'esecuzione del comando SQL con INSERT INTO...

Insert into

La sintassi del comando INSERT INTO, che è stato presentato nell'unità didattica sui linguaggio SQL, è la seguente:

- INSERT INTO Anagrafe(Cognome, Nome, Telefono)
- VALUES)Rossi, Giovanni, 06/728190)
- Utilizzando i valori provenienti dal form della pagina HTML con i dati inseriti dall'utente, anziché valori costanti, si deve costruire la frase SQL come è stato fatto nel file ASP precedente:
- costruzione del comando SQL con gli input del form
- strSQL = INSERT INTO Anagrafe(Cognome, Nome, Telefono)
- strSQL = strSQL & 'VALUES

- strSQL = strSQL & & request (Cognome') & '
- strSQL = strSQL & & request("Nome") & '
- strSQL = strSQL & ''' & request)Telefono) &
- strSQL = strSQL & ')

7.4 – Le pagine di accesso ai dati

Un'altra modalità per pubblicare sul Web Server i dati provenienti da un database di Access è costituita dalla pagina di accesso ai dati (Data Access Page). Essa è semplice mente una pagina Web connessa a un database. In una pagina di accesso ai dati è possibile visualizzare i dati memorizzati nel database, aggiungere altri dati o modificare e manico-lare quelli esistenti, utilizzando un'interfaccia simile a quella delle maschere e dei report di Access.

Data access page

Per creare una pagina di accesso ai dati:

1. dalla lista degli oggetti di Access aprire l'elenco Pagine e fare clic sul pulsante Nuovo;
2. si apre la finestra di dialogo Nuova pagina di accesso ai dati, scegliere l'opzione Creazione guidata pagina e specificare la tabella o a query che si vuole utilizzare;
3. nella finestra di dialogo successiva fare clic sul pulsante per spostare tutti campi della tabella nella casella campi selezionati e di seguito clic su Avanti;
4. nella seconda e terza finestra di dialogo per la definizione della pagina, si possono fissare campi di raggruppamento dei record e criteri di ordinamento dei dati; in assenza di richieste specifiche viene confermato quanto suggerito da Access;
5. specificare un nome per indicare il titolo della pagina e fare clic sul comando Aprire per concludere la procedura di creazione.

Viene creato un file HTML che può essere pubblicato in Internet. Le pagine di accesso ai dati sono supportate dal browser Internet

Explorer. con versione 5 o successiva. Se la pagina è stata salvata con il nome Clientil.htm nella sottocartella WebClienti del Web Server, essa può essere visualizzata dal browser specificandone l'indirizzo:

http://localhost/WebClienti Clienti.htm

Si noti che la pagina creata con la procedura precedente presenta nella parte inferiore una barra di pulsanti di spostamento analoga a quella utilizzata nelle maschere di Access.

Record tabella

Con questi pulsanti è possibile visualizzare tutti i record della tabella origine ed effettua re operazioni di manipolazione sui dati contenuti nelle righe.

Navigazione sui record	Manipolazione Filtri e ordinamenti
Primo record Successivo	Rimuove il filtro
Precedente record	Ultimo Filtro
Numero del record corrente	Aggiunge un record Ordinamento decrescente
Elimina	Ordinamento crescente
Salva	Annulla l'ultima modifica

A differenza degli altri oggetti di Access (tabelle, query, maschere, ecc.), i file delle pagine non sono memorizzati all'interno del database, ma nelle cartelle del desktop di Windows, salvo diversa specificazione da parte dell'utente; il database contiene solo dei collegamenti a tali pagine.

Le pagine di accesso ai dati, pur essendo oggetti esterni al database di Access, permettono di visualizzare, modificare, aggiornare i Recordset contenuti nel database.

OLE DB

Per le pagine di accesso ai dati non è necessario specificare un DSN per ODBC, perché si utilizza la tecnologia OLE DB, ma è necessario collocare il database di Access su un server Internet o nelle cartelle condivise del Personal Web Server.

È opportuno collocare il database nella cartella prima di creare la

pagina; infatti se si sposta il database a cui è connessa la pagina dopo averla creata, si deve, anche, modificare la connessione dell'origine dati della pagina.

Per modificare la connessione al database di una pagina di accesso ai dati:

1. fare clic sulla pagina di accesso ai dati;
2. scegliere la visualizzazione Struttura;
3. fare clic sull'icona Elenco Campi nella barra degli strumenti di Access;
4. fare clic con il pulsante destro del mouse sull'icona del database, che si trova all'interno della finestra che si apre;
5. scegliere Connessione nel menu di scelta rapida;
6. specificare il percorso del database al quale la pagina deve essere connessa.

Se i dati devono essere condivisi tra più utenti di una rete, è opportuno non utilizzare percorsi che fanno riferimento a unità presenti sul computer, ma piuttosto si indichino le denominazioni di rete per i file, per rendere i percorsi indipendenti dai dischi della macchina.

Per esempio: ***Esempio***

\\pc 1\Esercizi\db i.mdb

essendo Esercizi una cartella condivisa del computer avente il come nome identificativo nella rete.

Queste denominazioni di file in Windows si chiamano UNC (Universal Naming Conventitori).

Le pagine di accesso ai dati utilizzano la tecnologia XML (eXtensible Markup Language) per la pubblicazione dei database sul Web. XML è una versione avanzata del linguaggio HTML, utilizzato per scrivere le pagine visualizzabili con un browser: le funzioni di XML consentono la manipolazione dei database e in generale di qualsiasi altro

documento, in modo interattivo sul Web.
Nelle versioni più recenti di Office per Windows, è il linguaggio che consente di salvare un documento Word in formato Web e viceversa di tornare al formato Word.

7.5 – Esempi di pagine di accesso ai dati

Esempio Costruire una pagina Web per visualizzare gli ordini dei clienti ricevuti in un mese dell'anno, raggruppandoli per giorno.

Pagina Web

Questo esempio mostra come la pagina di accesso ai dati possa essere usata per consolidare e raggruppare le informazioni presenti nel database e per pubblicare un sommario dei dati.
I dati sono contenuti nella tabella Ordini del database dbl.mdb.
La sottocartella del Web Server si chiama Web Clienti.
Gli ordini si riferiscono al mese di giugno; nella maschera di visualizzazione della pagine Web vengono considerati, per ciascun ordine, i seguenti campi: Numero, Data e Importo (complessivo dell'ordine).
La prima operazione consiste nel creare in Access una query di nome Ordini, che entra dalla tabella Ordini i record aventi la data dell'ordine nel mese di giugno 2000.
Nella finestra della struttura della query, scegliamo dalla tabella Ordini i campi Numero Data e Importo e impostiamo, nella riga dei criteri, la condizione per la data: Between 01/06/00 and 3 0/06/00.
Salviamo quindi la query con il nome Ordini.
La seconda parte del lavoro riguarda la creazione di una nuova pagina di accesso ai da associata alla query Ordini.
Scegliamo la creazione guidata di una nuova pagina di accesso ai dati e specifichiamo l'oggetto query Ordini come riferimento per la pagina.
Nella finestra di richiesta dei campi da inserire nella pagina, facciamo clic sul pulsante » per trasferire tutti e tre i campi della query nella pagina.

Nella finestra successiva, alla domanda Aggiungere livelli di raggruppamento, scegliamo il campo Data come campo di raggruppamento, Nella stessa finestra facciamo poi clic sul pulsante Opzioni di raggruppamento e nella casella Intervalli di raggruppamento scegliamo Giorno.

Nell'ultima finestra indichiamo anche due criteri di ordinamento: il primo sulla data e il secondo sul numero dell'ordine. Alla fine salviamo la pagina con il nome Ordini.htm nella sottocartella del Web Server.

Per visualizzare il risultato del lavoro apriamo il programma browser e nella casella dell'indirizzo scriviamo:

http://localhost/WebClienti/Ordini.htm

La maschera presenta due barre di navigazione tra i record: una per i giorni e l'altra per gli ordini all'interno dello stesso giorno. Accanto al giorno compare anche un pulsante di espansione con il segno +, che diventa un pulsante di riduzione con il segno - quando nella maschera si aprono le caselle con i valori dei campi.

La pagina di accesso ai dati può anche essere costruita (o modificata successivamente alla sua creazione] attraverso la visualizzazione Struttura, come avviene per gli altri oggetti di Access (tabelle, query, maschere, report). Con questa modalità di lavoro è possibile personalizzare la pagina aggiungendo controlli dalla Casella degli strumenti per arricchire l'interfaccia grafica.

Casella di strumenti excel

La Casella degli strumenti contiene alcuni controlli già visti nell'ambiente di programmazione Visual Basic, oltre ad altri controlli per inserire fogli e grafici di Excel. Ci sono poi alcuni controlli specifici per la costruzione delle pagine di accesso ai dati:

- controllo HTML Associato. È possibile associare a questo controllo il campo di tipo Testo di una tabella del database, contenente codice e tag HTM anziché stringhe di caratteri. Nella visualizzazione della pagina di accesso ai dati. il codice

viene interpretato come avviene con un browser;

- controllo testo scorrevole (Marque). Il controllo che visualizza testo scorrevole viene utilizzato per attirare l'attenzione del l'utente, come un titolo o un annuncio importante. È possibile personalizzare il controllo impastando opzioni quali la direzione di spostamento, la velocità e il tipo di movimento.
- controllo spostamento tra record. Il controllo aggiunge una barra degli strumenti per lo spostamento tra record nella sezione principale o in quella di un livello di gruppo. Il controllo collegamento ipertestuale associato serve per associare un collegamento ipertestuale (contenuto in un campo di tipo Collegamento Ipertestuale) diverso per ciascun record di una tabella. Spostandosi da un record a un altro, è possibile fare clic collegamento ipertestuale e accedere a una pagina Web diversa per ogni record.
- controllo immagine area sensibile: il controllo aggiunge nella pagina di accesso ai dati un'immagine che rappresenta collegamento ipertestuale a un file o a una pagina Web. Si deve specificare il nome file immagine e l'indirizzo del collegamento ipertestuale. Quando la pagina visualizzata dai browser, passando il cursore sull'immagine, questo assume l'aspetto: una mano indicando in tal modo che l'immagine è un link a un'altra pagina Web. È possibile definire il testo alternativo per l'immagine (quello che compare nel rettangolo::giallo quando si passa sopra l'immagine con il puntatore del mouse).

Esempio

Costruire una pagina Web per la visualizzazione dei dati e per le operazioni di inserimento, e variazione dei prodotti disponibili a listino.

L'esempio serve a mostrare la modalità di costruzione di una pagina di accesso a senza la creazione guidata, ma in modalità di visualiz-

zazione Struttura, con utili alcuni dei controlli della Casella degli strumenti.

La tabella di riferimento si chiama Prodotti, che contiene, tra gli altri, i campi (campo chiave), Descrizione e Prezzo unitario.

In Access facciamo clic sulla tabella Prodotti e poi clic sull'icona Nuovo oggetto barra degli strumenti di Access. Per questo esempio non usiamo la creazione gv - quindi scegliamo Visualizzazione Struttura per creare una nuova pagina di acce vuota. Sul video si apre anche la Casella degli strumenti e la finestra Elenco dei clienti disponiamo queste due finestre sui desktop in modo da poter lavorare sulla nuova.

La pagina comprende il titolo, una zona dove inserire testo descrittivo e la zona inserire i controlli, detta Sezione non associata. Per inserire un nuovo titolo, fare clic sul messaggio che compare e scrivere i Prodotti.

Il titolo può essere modificato nel font, nella dimensione, nel colore o nell'allineamento.

Sotto il titolo inseriamo un testo scorrevole con una scritta che serve a richiamare l'attenzione dell'utente. Per far questo occorre scegliere con un clic il controllo Test scorrevole nella Casella degli strumenti e fissare la posizione e le dimensioni dirette nella pagina trascinando il mouse. Con un clic al suo interno, si può comunque: scrivere la frase del testo scorrevole. Elenco dei prodotti disponibili a listino proprietà di questo controllo possono essere cambiate per ottenere font, colori c sioni preferiti.

Per modificare le proprietà di un controllo

1. selezionare il controllo con un clic su di esso;
2. fare clic sull'icona Proprietà nella barra degli strumenti di Access.

In alternativa fare clic con il pulsante destro sul controllo e scegliere Proprietà dal menu di scelta rapida che si apre.

Si tratta ora di inserire i campi della tabella nella pagina: dalla finestra Elenco dei campi scegliere i campi desiderati, uno per volta, e trascinarli nella pagina, disponendoli in modo ordinato. Anche le descrizioni e i contenuti delle caselle di testo possono essere modificate nel font, nelle dimensioni o nel colore. Inseriamo i campi per il codice pro dotto, la descrizione e il prezzo unitario.

Il programma aggiunge automaticamente il nome della tabella nell'intestazione della zona dei controlli e la barra di navigazione tra i record nella parte bassa della pagina (Sezione spostamento).

Per completare la pagina inseriamo anche un'immagine in basso che deve diventare un link a un sito Web. Scegliamo quindi, con un clic nella Casella degli strumenti, il controllo Immagine area sensibile e specifichiamo il nome del file contenente l'immagine oltre all'indirizzo Internet del sito che deve essere aperto quando l'utente fa un clic sopra l'immagine.

Definiamo anche la descrizione che deve comparire sullo schermo quando il puntatore del mouse passa sopra l'immagine.

La progettazione della pagina è conclusa e si può vederne un'anteprima facendo clic sull'icona di Visualizza, in alto a sinistra nella barra degli strumenti di Access.

Salviamo la pagina con il nome Prodotti.htm nella sottocartella del Web Server.

Per validare il funzionamento della pagina in ambiente Web, chiudiamo Access e apriamo la pagina con il browser, scrivendo nella casella dell'indirizzo:

http://localhost/WebClienti/Prodotti.html

La pagina può essere usata per visualizzare i record, per inserire un nuovo record o per modificare il contenuto di un record esistente, utilizzando le icone della barra di navigazione nella parte bassa della pagina. Dopo queste operazioni occorre fare clic sull'icona Salva della barra di navigazione dei record per rendere effettive le

modifiche nel database.

Per verificare che le operazioni di manipolazione siano state effettivamente registrate nella tabella del database, si può tornare in Access e aprire la tabella Prodotti per controllare il contenuto dopo le modifiche.

Test

1. Che cosa indica il termine localhost?
   a) Il server Web sul computer locale.
   b) Il computer locale dell'utente.
   c) Un'applicazione locale sul disco C:
   d) Un browser locale.

2. Quali tra le seguenti affermazioni riferite alle pagine ASP.NET sono vere e quali falde ?
   a) La pagine ASP:NET sono identificate dall'estensione. Asp
   b) La tecnologia ASP:NET è caratterizzata da una bassa scalabilità delle applicazioni.
   c) ASP.NET utilizza il linguaggio Visual Basic, non la versione VBScript
   d) Le pagine ASP.NET contengono codice compilato.

3. Completa le frasi seguenti utilizzando una tra le parole indicate alla fine domanda

   L'oggetto per stabilire una connessione al database è…….

   L'istruzione per aprire una connessione con n è…….

   L'oggetto per gestire un Recordset è…….

   L'istruzione per aprire un Recordset rs è…….

   Request.form, Response.Write, Server.CreateObject, ADODB.Con-nection, ADODB:Recordset, Conn.Open, rs.open,conn.close, rs.close.