

Basi di Dati

Soluzione della prova scritta del 21 Gennaio 2005

Esercizio 1 (2 punti)

Indicare quali delle seguenti affermazioni sono vere:

1. In una relazione del modello relazionale possono esistere tuple con valori uguali per gli attributi che formano la chiave, e valori diversi per gli altri attributi.
2. Un modello logico basato su valori non prevede puntatori a livello logico, anche se i puntatori possono essere presenti a livello fisico
3. Un modello logico basato su valori richiede l'utilizzo di puntatori
4. Si dice che "il modello relazionale è basato su valori" perché le corrispondenze fra tuple di relazioni diverse sono realizzate per mezzo di valori comuni
5. Il modello concettuale di una base di dati rappresenta i dati del sistema informativo tenendo conto del carico applicativo

Soluzione

Sono vere la 2, e la 4.

Schema Relazionale per gli Esercizi 2 e 3

AUTORE(Codice,Cognome,Nome)

EDITORE(Codice,Nome,Città)

LIBRO(Codice,Titolo,Autore,Editore,Anno,NumeroPagine)

Foreign key Autore References AUTORE(Codice)

Foreign key Editore References EDITORE(Codice)

Esercizio 2 (4 punti)

Formulare in SQL la seguente interrogazione: *per ciascun autore trovare il libro (o i libri) con il maggior numero di pagine. Mostrare Nome e Cognome dell'autore, titolo del libro e numero di pagine.*

Soluzione

```
SELECT A.Nome, A.Cognome, L.Titolo, L.NumeroPagine
FROM Autore A, Libro L
WHERE L.Autore = A.Codice
      AND L.NumeroPagine = (SELECT MAX(L1.NumeroPagine)
                           FROM Libro L1
                           WHERE L1.Autore = L.Autore)
```

Esercizio 3 (8 punti)

Formulare in Algebra Relazionale (4 punti) e SQL (4 punti) la seguente interrogazione: *Trovare gli autori che hanno sempre pubblicato i loro libri presso lo stesso editore.* Nella formulazione in algebra relazionale è sufficiente mostrare il codice dell'autore ed il codice dell'editore. Nella formulazione in SQL mostrare nome e cognome dell'autore e nome dell'editore.

Soluzione

Algebra Relazionale

$\pi_{Autore, Editore}(LIBRO) - (\pi_{Autore, Editore}((LIBRO) \Join_{Autore=Autore1 \wedge Editore \neq Editore1} (p_{Codice1, Titolo1, Autore1, Editore1, Anno1, NumeroPagine1} \leftarrow Codice, Titolo, Autore, Editore, Anno, NumeroPagine (LIBRO))))$

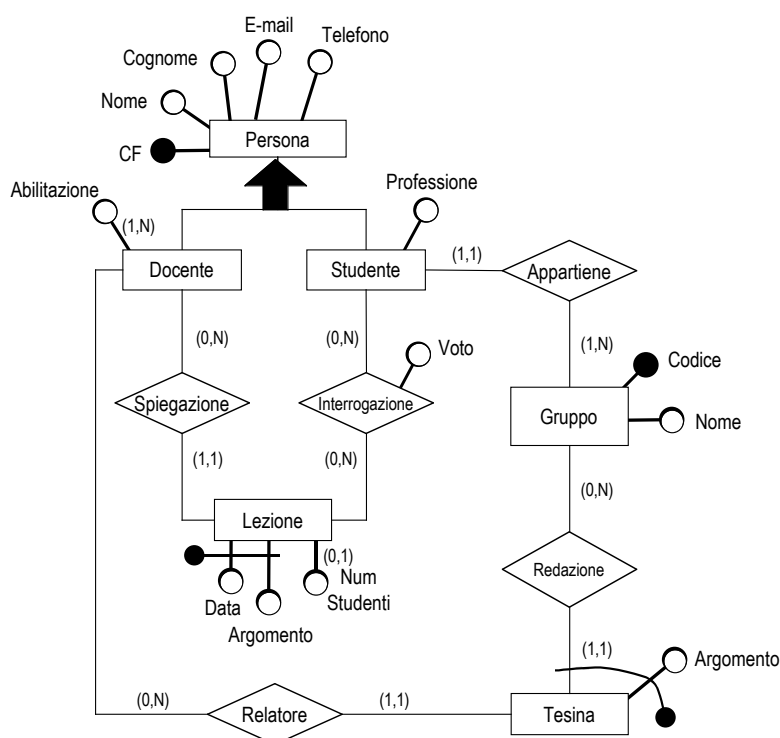
```
SELECT DISTINCT A.Nome, A.Cognome, E.Nome
FROM Autore A, Editore E, Libro L
WHERE A.Codice = L.Autore AND E.Codice = L.Editore
      AND A.Codice NOT IN (SELECT L1.Autore
                           FROM Libro L1
                           WHERE L1.Autore = L.Autore
                           AND L1.Editore <> L.Editore)
```

Esercizio 4 (7 punti)

Si vogliono memorizzare dati sulle lezioni organizzate da un ente di formazione secondo le seguenti specifiche:

- Per gli studenti e i docenti, memorizziamo codice fiscale, Nome, Cognome, indirizzo e-mail e un numero di telefono. Per gli studenti memorizziamo la professione, mentre per i docenti l'area (o le aree) disciplinari cui sono abilitati.
- Ogni lezione è tenuta da un docente e riguarda un argomento
- In una certa data un docente può tenere una ed una sola lezione ed un argomento può essere trattato in una ed una sola lezione.
- Per ogni lezione deve essere riportato il numero totale degli studenti presenti e le interrogazioni fatte agli studenti con il relativo voto. Durante una lezione uno studente può essere interrogato al massimo una volta
- Gli studenti sono organizzati in gruppi di lavoro. Un gruppo di lavoro è descritto da un codice e da un nome ed è costituito da esattamente tre studenti. Uno studente appartiene ad uno ed un solo gruppo.
- Per un dato argomento, un gruppo può effettuare una ed una sola tesina ed è seguito in questa attività da un unico docente.

Soluzione



Esercizio 5 (5 punti)

Tradurre lo schema concettuale ottenuto al punto precedente in uno schema logico del modello relazionale.

Soluzione

STUDENTE(CF, Nome, Cognome, E-mail, Telefono, Professione, CodiceGruppo)

Foreign Key CodiceGruppo References GRUPPO(Codice)

DOCENTE(CF, Nome, Cognome, E-mail, Telefono)

ABILITAZIONI(AreaDisciplinare, CFDocente)

Foreign Key CFDocente References DOCENTE(CF)

LEZIONE(Data, Argomento, NumStudenti, CFDocente)

Foreign Key CFDocente References DOCENTE(CF)

INTERROGAZIONE(Data, Argomento, CFStudente, Voto)

Foreign Key Data, Argomento References LEZIONE(Data, Argomento)

Foreign Key CFStudente References STUDENTE(CF)

GRUPPO(Codice, Nome)

TESINA(CodiceGruppo, Argomento, Docente)

Foreign Key CodiceGruppo References GRUPPO(Codice)

Foreign Key Docente References DOCENTE(CF)

Esercizio 6 (4 punti)

Si consideri il seguente schema di relazione

VOLO-AEREO(CodiceVolo,AeroportoPartenza,CittàPartenza,OraPartenza,AeroportoArrivo,CittàArrivo,OraArrivo)

Si supponga che ogni aeroporto serva una sola città. Trovare le dipendenze funzionali definite sullo schema (2 punti). Trovare una decomposizione in BCNF motivando la risposta (2 punti)

Soluzione

Le dipendenze funzionali sono

CodiceVolo → AeroportoPartenza,CittàPartenza,OraPartenza,AeroportoArrivo,CittàArrivo,OraArrivo

AeroportoPartenza → CittàPartenza

AeroportoArrivo → CittàArrivo

La chiave è CodiceVolo e lo schema non è in BCNF.

Se decomponiamo rispetto alle dipendenze funzionali otteniamo

VOLO(CodiceVolo,AeroportoPartenza ,OraPartenza,AeroportoArrivo ,OraArrivo)

AEROPORTO(Nome, Città)

Esercizio 7 (2 punti)

Commentare il codice riportato di seguito con particolare riferimento ai tag <?php e ?> e al ruolo della variabile globale \$_POST (si supponga che il codice di seguito venga richiamato successivamente alla compilazione di un *form*).

```
<HTML>
<HEAD>
<TITLE>Test Apache + PHP + MySQL</TITLE>
</HEAD>
<BODY>
<?php
// parametri del database
$db_host = "localhost";
$db_user = "root";
$db_password = "";
$db_name = "testphp";
$db = mysql_connect($db_host, $db_user, $db_password);

if ($db == FALSE) die ("Errore 1!");

mysql_select_db($db_name, $db) or die ("Errore 2!");

$query = "INSERT INTO tabella1 VALUES($_POST[id], '$_POST[nome]',
'$_POST[cognome]')";
$result = mysql_query($query) or die("Errore 3!");

mysql_close($db);

?>
<h1>Dati inseriti correttamente!</h1>
<a href="insert-form.html">insert new data!</a>
</BODY>
</HTML>
```