

PROGETTAZIONE DI DATABASE Linguaggio SQL

Modello Concettuale (Modellazione del mondo reale)
Modello Logico (Definizione del tipo e del formato dei dati)
Modello Fisico (Implementazione fisica su supporti di massa)

Modello concettuale

Si utilizza il modello Entity/Relationship (Entità/Associazioni)
Ogni oggetto del mondo reale diventa un'entità (grafico rettangolo)
Le associazioni sono i legami tra coppie di entità (grafico rombo)
Le proprietà delle entità e delle associazioni sono dette attributi (grafico pallini)

Esempio Biblioteca
Esempio Farmacia
Esempio Ristorante

Modello Logico

Le entità diventano tabelle
Le associazioni M:M diventano tabelle
Le associazioni 1:M diventano sono espresse tramite campi
Definizione di Chiave Primaria: attributo o serie di attributi che identificano in modo univoco una tupla (riga) della tabella

Tipi di dati:

Numeri interi: smallint 2 byte, integer 4 byte
Numeri reali: single 4 byte, double 8 byte
Stringhe di caratteri: char(n) n byte
Date e orario: date, time 6 byte
Booleani: boolean 1 byte

Operazioni relazionali:

Selezione (opera sulle righe di una tabella)
Proiezione (opera sulle colonne di una tabella)
Congiunzione (effettua il prodotto tra due tabelle)
Il risultato di un'operazione relazionale è una tabella

Linguaggio SQL (Structured Query Language)

Creare e gestire un database:
CREATE DATABASE Ristorante;
USE Ristorante;

Creare una tabella:
CREATE TABLE Tavolo
(id SMALLINT,
numposti SMALLINT,
id_sala SMALLINT,
id_cameriere SMALLINT,
PRIMARY KEY (id),
FOREIGN KEY (id_sala) REFERENCES sala(id),
FOREIGN KEY (id_cameriere) REFERENCES cameriere (id));
Oppure
CREATE TABLE Tavolo

```
( id SMALLINT not null,  
numposti SMALLINT,  
id_sala SMALLINT REFERENCES sala(id),  
id_cameriere SMALLINT REFERENCES cameriere (id) );
```

Cancellare una tabella:
DROP TABLE Tavolo;

Aggiungere un attributo alla tabella
ALTER TABLE Tavolo
ADD descrizione CHAR(20);

Rimuovere un attributo da una tabella
ALTER TABLE Tavolo
DROP numposti;

Inserire una tupla (riga) in una tabella
INSERT INTO Tavolo
(id, numposti, id_sala, id_cameriere)
VALUES (2, 4, 1, 3);

Modificare un dato (un valore di attributo di data una tupla):
UPDATE Tavolo
SET numposti=2
WHERE id=2;

Modificare un dato (un valore di attributo di più tuple):
UPDATE Tavolo
SET numposti=2
WHERE id<=2;

Cancellare una riga:
DELETE FROM Tavolo
WHERE id=1;

Interrogazioni: Comando SELECT

Formato del comando SELECT:
SELECT campo1, campo2, campo3, ...
FROM tabella1, tabella2, ...
WHERE condizione;

La condizione può essere costituita da più condizioni legate tramite gli operatori logici AND OR e NOT.

Esempio: Ristorante
Visualizzare l'elenco di tutti i camerieri:
SELECT * FROM Cameriere;

Elenco dei camerieri che risiedono a Lecce:
SELECT * FROM Cameriere WHERE residenza="Lecce";

Visualizzare il cognome e il nome di tutti i camerieri:
SELECT cognome, nome FROM Cameriere;

Visualizzare il cognome e il nome di tutti i camerieri che risiedono a Lecce:

```
SELECT cognome, nome FROM Cameriere WHERE residenza="Lecce";
```

Visualizzare il nome e il prezzo delle pietanze con più di 100 kcalorie:

```
SELECT nome, prezzo FROM Pietanze WHERE calorie>100;
```

Visualizzare il nome e il prezzo delle pietanze di origine salentina con più di 100 kcalorie:

```
SELECT nome, prezzo FROM Pietanze WHERE (calorie>100 AND origine="Salento");
```

Visualizzare il nome e il prezzo di tutte le pietanze, escluse quelle salentine, con più di 100 kcalorie:

```
SELECT nome, prezzo FROM Pietanze WHERE ((calorie>100) AND (NOT origine="Salento"));
```

Cognome e nome dei camerieri che servono ai tavoli con più di due posti:

```
SELECT cognome, nome FROM cameriere, tavolo  
WHERE (posti>2 AND tavolo.id_cameriere=Cameriere.id);
```

Nome e prezzo delle pietanze con categoria="primo":

```
SELECT nome, prezzo FROM Pietanza, Categoria  
WHERE (Categoria.descrizione="primo" AND Pietanza.id_categoria=Categoria.id);
```

Visualizzare il codice id del tavolo e il numero dei posti dei tavoli appartenenti alla sala fumatori:

```
SELECT Tavolo.id, posti FROM Tavolo, Sala  
WHERE (Sala.fumatori="Si" AND id_sala=Sala.id);
```

Nome degli ingredienti che si trovano nella pietanza "Tiramisù":

```
SELECT Ingrediente.nome FROM Pietanza, Ingrediente, Preparato  
WHERE id_pietanza=Pietanza.id AND id_ingrediente=Ingrediente.id AND  
Pietanza.nome="Tiramisù";
```

Visualizzare i piani in cui serve il cameriere dal cognome "Rossi":

```
SELECT Sala.piano FROM Cameriere, Tavolo, Sala  
WHERE Cameriere.cognome="Rossi" AND Tavolo.id_cameriere=Cameriere.id AND  
Tavolo.id_sala=Sala.id;
```

Elenco delle origini delle pietanze:

```
SELECT DISTINCT origine FROM Pietanza;
```

Elenco dei nomi e dei prezzi di tutte le pietanze con prezzo>10.00 euro:

```
SELECT nome, prezzo FROM Pietanza WHERE prezzo<10.00;
```

Nome degli ingredienti con cui si preparano la pietanza dal nome="spaghetti al pomodoro":

```
SELECT Ingrediente.nome FROM Ingrediente, Preparato, Pietanza  
WHERE Pietanza.nome="spaghetti al pomodoro" AND Pietanza.id=id_pietanza AND  
Ingredienti.id=id_ingrediente;
```

Nome e cognome dei camerieri che hanno servito antipasti:

```
SELECT Cameriere.nome, Cameriere.cognome FROM Cameriere, Serve, Pietanza, Categoria  
WHERE descrizione="antipasto" AND Categoria.id=Pietanza.id_categoria AND  
Pietanza.id=Serve.id_pietanza AND Serve.id_cameriere=Cameriere.id;
```

Nome e calorie delle pietanze della regione Puglia:

```
SELECT nome, calorie FROM Pietanza WHERE origine="Puglia";
```

Nome e origine delle pietanze servite dal cameriere "De Pascalis":
SELECT Pietanza.nome, Pietanza.origine FROM Cameriere, Serve, Pietanza
WHERE Cameriere.cognome="De Pascalis" AND Pietanza.id=Serve.id_pietanza AND
Serve.id_cameriere=Cameriere.id;

Nome e origine delle pietanze servite dai camerieri "De Pascalis" e "De Matteis":
SELECT Pietanza.nome, Pietanza.origine FROM Cameriere, Serve, Pietanza
WHERE Pietanza.id=Serve.id_pietanza AND Serve.id_cameriere=Cameriere.id AND
(Cameriere.cognome="De Pascalis" OR Cameriere.cognome="De Matteis");

Funzioni di aggregazione

COUNT (DISTINCT)
SUM
AVG
MIN
MAX

Numero delle pietanze che costano più di 10.00 euro:
SELECT COUNT(*) FROM Pietanza WHERE prezzo>10.00;

Prezzo medio, minimo e massimo di tutte le pietanze:
SELECT AVG(prezzo) AS Media, MIN(prezzo) AS Minimo, MAX(prezzo) AS Massimo
FROM Pietanza;

Numero di province di provenienza dei camerieri:
SELECT COUNT(provincia) FROM Cameriere;

Numero di province differenti di provenienza dei camerieri:
SELECT COUNT (DISTINCT provincia) FROM Cameriere;

Prezzo totale di tutti gli antipasti:
SELECT SUM(prezzo) FROM Pietanza, Categoria WHERE Pietanza.id_categoria=Categoria.id
AND Categoria.descrizione="Antipasto";

Ordinamenti e raggruppamenti:
Ordinamento in ordine alfabetico dei camerieri:
SELECT nome, cognome, data_nascita FROM Cameriere ORDER BY cognome, nome;

Elenco delle pietanze, con descrizione e prezzo, della regione Puglia, in ordine decrescente di prezzo:
SELECT descrizione, prezzo FROM Pietanza WHERE origine="Puglia" ORDER BY prezzo DESC;

Numero di pietanze prezzo medio raggruppate per origine:
SELECT origine, COUNT(*) AS numero, AVG(prezzo) AS media
FROM Pietanza GROUP BY origine;

Condizioni di ricerca

BETWEEN (usato per cercare un valore tra un intervallo)
IN (usato per definire alcuni valori di un campo)
LIKE (equivalente al simbolo =, ma usato per campi di tipo testo o caratteri jolly)

Elenco dei camerieri appartenenti alle sole province di Lecce, Brindisi e Taranto:
SELECT cognome, provincia FROM cameriere WHERE provincia IN ("LE","BR","TA");

Elenco dei camerieri il cui cognome inizia per "Ro":
SELECT Cognome, nome FROM cameriere WHERE cognome LIKE "Ro%";

Un campo che non contiene dati si definisce NULL
Elenco dei camerieri di cui non è nota la provincia:
SELECT Cognome, nome FROM cameriere WHERE provincia IS NULL;

Elenco dei camerieri di cui è nota la provincia:
SELECT Cognome, nome FROM cameriere WHERE provincia IS NOT NULL;

Query nidificate (sub query)
Elenco dei pacchetti (descrizione e prezzo) il cui prezzo sia maggiore del prezzo medio:
SELECT descrizione, prezzo FROM pacchetto
WHERE prezzo > (SELECT AVG(prezzo) FROM Pacchetto);

Gestione della sicurezza dei dati
Definizione di profili utente e assegnamento (o revoca) dei privilegi:

```
CREATE USER utente1;  
CREATE USER utente2;  
GRANT UPDATE ON Pacchetto TO utente1;  
REVOKE UPDATE ON Pacchetto TO utente1;
```

Privilegi possibili:
UPDATE (aggiornare dati su ogni singola tupla della tabella)
DELETE (cancellare tuple)
SELECT (query)
ALTER (modificare la struttura della tabella)
INSERT (aggiungere nuove tuple in una tabella)
ALL (tutti i privilegi)

Esempio database - Agenzia di viaggi

```
LINGUAGGIO SQL (MYSQL)  
CREATE DATABASE Agenzia_Viaggi;  
USE Agenzia_Viaggi;  
SHOW DATABASES;  
SHOW TABLES;  
DESCRIBE nome_tabella;  
CREATE TABLE Tour_Operator (Id CHAR(2), Nome CHAR(30), Tel CHAR(16), PRIMARY  
KEY(Id));  
CREATE TABLE Pacchetto (Id CHAR(2), Descrizione CHAR(30), Prezzo FLOAT, Data_inizio DATE,  
Data_fine DATE, PRIMARY KEY(Id));  
ALTER TABLE Pacchetto ADD Id_Tour_Operator CHAR(2) REFERENCES Tour_Operator(Id);  
CREATE TABLE Cliente (CF CHAR(16), Cognome CHAR(30), Nome CHAR(30), Indirizzo  
CHAR(60), Tel CHAR(16), PRIMARY KEY(CF));  
CREATE TABLE Località (Id CHAR(16), Descrizione CHAR(30), Luogo CHAR(30), Clima  
CHAR(30), Moneta CHAR(30), PRIMARY KEY(Id));  
CREATE TABLE Composto (Id_pacchetto CHAR(2) REFERENCES Pacchetto(Id), Id_località  
CHAR(2) REFERENCES Località(Id), PRIMARY KEY(Id_pacchetto, Id_località));
```

```
CREATE TABLE Acquista (Id_cliente CHAR(16) REFERENCES Cliente(CF), Id_pacchetto CHAR(2)
REFERENCES Pacchetto(Id), PRIMARY KEY(Id_cliente, Id_pacchetto));
INSERT INTO Tout_Operator VALUES('T1','Salento Viaggi','0832/337622');
INSERT INTO Pacchetto('P1','Weekend a Parigi',840.00,'2005-12-2','2005-12-5');
```

Elenco dei pacchetti con prezzo minore di 2000 euro, in ordine di prezzo:

```
SELECT * FROM Pacchetto WHERE prezzo<2000 ORDER BY prezzo;
```

Elenco dei pacchetti, descrizione e prezzo, nella località "Parigi":

```
SELECT Pacchetto.descrizione, prezzo FROM Pacchetto, Località
WHERE Località.descrizione="Parigi" AND Località.id_località=Composto.id_località AND
Pacchetto.id_pacchetto=Composto.id_pacchetto;
```

Cognome, nome e indirizzo dei clienti che hanno fatto un viaggio a Parigi:

```
SELECT Cliente.cognome, Cliente.nome, Cliente.indirizzo FROM Località, Pacchetto, Cliente,
Composto, Acquista WHERE Cliente.cf=Acquista.id_cliente AND
Pacchetto.id_pacchetto=Acquista.id_pacchetto AND
Pacchetto.id_pacchetto=Composto.id_pacchetto AND Composto.id_località=Località.id_località
AND Località.descrizione="Parigi";
```

Numero dei pacchetti offerti dalla organizzazioe turistica CTS:

```
SELECT COUNT(Pacchetto.id_pacchetto) FROM Tour_operator, Pacchetto
WHERE Pacchetto.organizzazione=Tour_operator.id AND Tour_operator.nome="CTS";
```

Nomi dei pacchetti la cui data di inizio è compresa nelle date 29/10/2005 e 31/10/2005:

```
SELECT descrizione, data_inizio FROM Pacchetto
WHERE data_inizio BETWEEN 29/10/2005 AND 31/10/2005;
```

Questa query equivale a:

```
SELECT descrizione, data_inizio FROM Pacchetto
WHERE data_inizio>=29/10/2005 AND data_inizio<=31/10/2005;
```

Elenco dei Tour Operator, nome e telefono, che hanno almeno 5 pacchetti in archivio:

```
SELECT Tour_operator.nome, Tour_operator.telefono, COUNT(Pacchetto.id)
FROM Tour_operator, Pacchetto WHERE Pacchetto.organizzazione=Tour_operator.id GROUP BY
Tour_operator.nomeHAVING COUNT(Pacchetto.id)>=5;
```

Elenco delle località con numero dei pacchetti che si riferiscono ad esse:

```
SELECT Località.descrizione COUNT(Pacchetto.id) FROM Località, Composto, Pacchetto
WHERE Pacchetto.id=Composto.id_pacchetto AND Composto.id_località=Località.id
GROUP BY Località.descrizione;
```

Query nidificate:

Elenco dei Tour Operator che organizzano pacchetti il cui prezzo sia minore del prezzo massimo dei pacchetti riferiti alla località "Salento":

```
SELECT nome FROM Tour_operator, Pacchetto WHERE Pacchetto.id_to=Tour_operator.id AND
prezzo<(SELECT MAX(prezzo) FROM Pacchetto, Composto, Località WHERE
Località.descrizione="Salento" AND Località.id=Composto.id_località AND
Composto.id_pacchetto=Pacchetto.id);
```

Elenco dei Tour Operator che organizzano pacchetti la cui descrizione sia quella dei pacchetti composti dalle località con moneta "euro":

```
SELECT nome FROM Tour_operator, Pacchetto WHERE Pacchetto.id_to=Tour_operator.id AND
Pacchetto.descrizione IN (SELECT descrizione FROM Pacchetto, Località, Composto WHERE
```

Località.moneta="euro" AND Località.id=Composto.id_località AND
Composto.id_pacchetto=Pacchetto.id);

Esercizio proposto:

In un database si vogliono organizzare le informazioni sulle prove sostenute dagli studenti di una scuola nelle diverse materie e con i diversi docenti: si devono quindi conservare i dati degli studenti, dei docenti, delle materie e delle prove in modo da rendere possibili le seguenti interrogazioni:

- L'elenco delle prove di uno studente
- L'elenco delle prove effettuate da un docente in una classe;
- I voti di uno studente in una materia;
- I voti di una classe nella prova svolta in una materia in un determinato giorno;
- Le date e il tipo delle prove effettuate da un docente in una materia;
- Le date e il tipo di prove effettuate da un docente in una classe;