



SQL

Come modificare i dati delle
tabelle

COS'È SQL

Linguaggio di
programmazione
dichiarativo

Creare, manipolare
e interrogare
database
relazionali

FUNZIONI DI SQL

DDL

- creare tabelle, vincoli, etc

DML

- inserimento, cancellazione e modifica delle righe

DCL

- permessi di accesso

CARATTERISTICHE SQL

Non è case-sensitive

Nomi di lunghezza massima di 18 caratteri

Nomi iniziano con lettera

– come unico carattere speciale

TIPI DI DATI IN SQL

CHAR(N)

BIT(N)

INT

REAL

DATE

TIME

CREARE UN DATABASE

CREATE DATABASE Negozio

CREARE UNA TABELLA

```
CREATE TABLE <nometabella>
```

```
(  
  <Attributo1> <Tipo1> (<VicoloAttributo1>),  
  <Attributo2> <Tipo2> (<VicoloAttributo2>),  
  ...  
  <Attributo2> <Tipo2> (<VicoloAttributo2>),  
  (<VincoloTabella>)  
);
```

VINCOLI DI ATTRIBUTO

NOT NULL

DEFAULT

CHECK(...)

ESEMPIO CREAZIONE TABELLA

Azienda(CodAzienda, RagioneSociale,
Fatturato, NumDipendenti)

Dipendente(CodDlp, Cognome, Nome,
CodAzienda)

ESEMPIO CREAZIONE TABELLA

CREATE TABLE Azienda

```
(  
  CodAzienda          CHAR(5)          NOT NULL,  
  RagioneSociale      CHAR(30)         NOT NULL,  
  Fatturato           INT              DEFAULT 1000,  
  NumDipendenti       INT              ,  
  CHECK(NumDipendenti BETWEEN 5 AND 200),  
  PRIMARY KEY(CodAzienda)  
);
```

ESEMPIO CREAZIONE TABELLA

CREATE TABLE Dipendente

```
(  
  CodDip          CHAR(5)          NOT NULL,  
  Cognome         CHAR(30)         NOT NULL,  
  Nome           CHAR(30)         NOT NULL,  
  CodAzienda      CHAR(5)          ,  
  PRIMARY KEY(CodDip),  
  FOREIGN KEY(CodAzienda) REFERENCES(Azienda)  
);
```

INSERIRE I VALORI IN UNA TABELLA

```
INSERT INTO <NomeTabella> VALUES (<Valore1>, <Valore2>, ..., <ValoreN>);
```

ESEMPIO INSERIMENTO VALORI

```
INSERT INTO Azienda VALUES ("C001" , "A&B Tessile" , 1500 , 80);
```

```
INSERT INTO Dipendente VALUES ("C74" , "Rossi" , "Mario" , "C001");
```

MODIFICA ED ELIMINAZIONE

UPDATE

**DELETE
FROM**



ESERCIZIO INTERMEZZO



CREARE LE TABELLE DEL SEGUENTE SCHEMA

ATTORE (CodiceAttore, Cognome, Nome)

FILM (CodiceFilm, Titolo, AnnoProduzione, Genere)

CINEMA (CodiceCinema, Nome, Posti, Città)

INTERPRETA (CodiceAttore, CodiceFilm)

PROGRAMMATO (CodiceFilm, CodiceCinema)



OPERAZIONI SULLE TABELLE LOGICHE



SELEZIONARE LA TABELLA

FROM nome tabella1,
nome tabella 2, ...

OPERAZIONI POSSIBILI

Unione

Intersezione

Differenza

Proiezione

Selezione

Join

UNIONE

Fondo due tabelle che
sono uguali nello schema,
ossia hanno gli stessi
attributi

INTERSEZIONE

Ottengo una tabella che
contiene solo i dati in
comune tra due tabelle con
lo stesso schema

DIFFERENZA

Ottengo una tabella che
contiene le istanze della
prima tabella che non sono
presenti nella seconda

PROIEZIONE

**Tengo solo alcune colonne
di una tabella**

COME EFFETTUARE LA PROIEZIONE

```
SELECT nome colonna 1,  
       nome colonna 2, ...
```


SELEZIONE

**Tengo le istanze che
soddisfano un certo criterio
di ricerca**

COME EFFETTUARE LA SELEZIONE

WHERE
attributo1="valore1",
attributo2="valore2", ...

JOIN

**Combino due relazioni
aventi uno o più attributi in
comune, generando una
nuova relazione**

JOIN

**Le colonne delle tabelle
vengono concatenate in
orizzontale, senza
duplicazioni**

JOIN

**Le righe delle tabelle
vengono concatenate in
verticale, senza
duplicazione**

ESEMPIO JOIN

CLIENTE			
CodiceCli	NomeCli	IndirizzoCli	CodAG
C006	Bianchi	Via Po 23	A0052
C002	Neri	Via Roma 12	A0016
C005	Rossi	Via Moro 2	A0052

AGENTE		
CodAG	NomeAg	TelAgente
A0016	Polis	3461234567
A0052	Rinaldi	3331234567

ESEMPIO JOIN

JOIN CLIENTE E AGENTE					
CodiceCli	NomeCli	IndirizzoCli	CodAG	NomeAg	TelAgente
C006	Bianchi	Via Po 23	A0052	Rinaldi	3331234567
C002	Neri	Via Roma 12	A0016	Polis	3461234567
C005	Rossi	Via Moro 2	A0052	Rinaldi	3331234567

COME SCRIVERE IL JOIN

WHERE
tabella1.nomeAttributo =
tabella2.nomeAttributo



ESEMPI SQL

VISUALIZZARE IL NOME DI TUTTI I CINEMA
PRESENTI NELLA CITTÀ DI LECCE

```
SELECT Cinema.nome  
FROM Cinema  
WHERE Cinema.Città = "Lecce"
```

VISUALIZZARE IL NOME DI TUTTI GLI ATTORI IL
CUI COGNOME E' LEVITT

```
SELECT Attore.Nome, Attore.Cognome  
FROM Attore  
WHERE Attore.Cognome = "Levitt"
```

VISUALIZZARE I CINEMA PRESENTI NELLA CITTÀ DI BARI CON PIÙ DI 500 POSTI IN SALA

```
SELECT Cinema.Nome
```

```
FROM Cinema
```

```
WHERE Cinema.Città = "Bari" AND Cinema.Posti > 500
```

VISUALIZZARE I FILM DRAMMATICI USCITI DOPO IL 1970

SELECT Film.Titolo

FROM Film

WHERE Film.Genere = "Drammatico"

AND Film.AnnoProduzione > 1970

**VISUALIZZARE IL COGNOME DEGLI ATTORI CHE
HANNO INTERPRETATO NEI FILM ROMANTICI**

```
SELECT Attore.Cognome  
FROM Attore JOIN Interpreta ON CodiceAttore  
      AND Film JOIN Interpreta ON CodiceFilm  
WHERE Film.Genere = "Romantici"
```



ESERCIZIO

SCHEMA RELAZIONALE

BICICLETTA (CodBici, Marca, Modello, Prezzo, CodTipo)

TIPOBICI (CodTipo, Descrizione)

SCRIVI SQL PER

1. Creare le tabelle BICICLETTA e TIPOBICI
2. Inserire una nuova bicicletta
3. Elencare tutte le biciclette di marca "Velox" e di tipo "Mountain Bike"
4. Elenca tutte le biciclette economiche (prezzo <100 euro) della marca "BMX"
5. Elencare le biciclette di prezzo inferiore a 500 euro, compresa la descrizione del tipo di bicicletta

SOLUZIONE

CREATE TABLE Bicicletta

```
(  
  CodBici          CHAR(5)          NOT NULL,  
  Marca            CHAR(30)         ,  
  Modello          CHAR(30)         ,  
  Prezzo           REAL(5)          ,  
  CodTipo          CHAR(5)          NOT NULL,  
  PRIMARY KEY(CodBici),  
  FOREIGN KEY(CodTipo) REFERENCES(Tipobici)  
)
```

SOLUZIONE

```
CREATE TABLE Tipobici
```

```
(
```

```
  CodTipo          CHAR(5)      NOT NULL,
```

```
  Descrizione      CHAR(100)    ,
```

```
  PRIMARY KEY(CodTipo)
```

```
)
```

SOLUZIONE

```
INSERT INTO Bicicletta VALUES(«B003», «Honda», «BMX», 150, «T098»)
```

```
SELECT *
```

```
FROM Bicicletta
```

```
WHERE Bicicletta.Marca=«Velox» AND Bicicletta.Tipo=«Mountain Bike»
```

SOLUZIONE

```
SELECT *  
FROM Bicicletta  
WHERE Bicicletta.Prezzo < 100 AND Bicicletta.Marca = «BMX»
```

```
SELECT *  
FROM Bicicletta JOIN Tipobici ON CodTipo  
WHERE Bicicletta.Prezzo < 500
```

SCHEMA RELAZIONALE

STUDENTE (CodStud, Nome, Cognome)

INTERROGATO (CodStud, Data, Voto, CodMateria)

MATERIA (CodMateria, NomeMateria)

SCRIVI SQL PER

1. Crea le tabelle ed inserisci un valore in Studente
2. Elencare il nome degli studenti che hanno come cognome "Rossi"
3. Elencare il voto delle interrogazioni dello studente "Mario Verdi"
4. Elencare il nome delle materie delle interrogazioni di "Mario Verdi"
5. Elencare nome, data, voto e nome della materia delle interrogazioni degli studenti

SOLUZIONE

```
CREATE TABLE Studente
```

```
(
```

```
  CodStud      CHAR(5)      NOT NULL,
```

```
  Nome         CHAR(50)     ,
```

```
  Cognome      CHAR(50)     ,
```

```
  PRIMARY KEY(CodStud)
```

```
)
```


SOLUZIONE

```
CREATE TABLE Interrogato
(
  CodStud      CHAR(5) NOT NULL,
  Data         DATE      ,
  Voto         REAL(2)   ,
  CodMateria   CHAR(5) NOT NULL,
  PRIMARY KEY(CodStud, CodMateria),
  FOREIGN KEY(CodStud) REFERENCES (Studente)
  FOREIGN KEY(CodMateria) REFERENCES (Materia)
)
```

SOLUZIONE

```
CREATE TABLE Materia
```

```
(
```

```
  CodMateria          CHAR(5)          NOT NULL,
```

```
  NomeMateria         CHAR(20)         ,
```

```
  PRIMARY KEY(CodMateria)
```

```
)
```

SOLUZIONE

```
INSERT INTO Studente VALUES(«S076», «Mario», «Rossi»)
```

```
SELECT Studente.Nome
```

```
FROM Studente
```

```
WHERE Studente.Cognome=«Rossi»
```

SOLUZIONE

```
SELECT Interrogato.Voto
FROM Studente JOIN Interrogato ON CodStud
WHERE Studente.Nome=«Mario» AND Studente.Cognome=«Verdi»
```

```
SELECT Materia.NomeMateria
FROM Interrogato JOIN Materia ON CodMateria
      AND Interrogato JOIN Studentr ON CodStud
WHERE Studente.Nome=«Mario» AND Studentr.Cognome=«Verdi»
```

SOLUZIONE

```
SELECT Studente.Nome, Interrogato.Data, Interrogato.Voto, Materia.NomeMateria  
FROM Interrogato JOIN Materia ON CodMateria  
      AND Interrogato JOIN Studentr ON CodStud
```

A man in a dark suit and tie is smiling and waving his right hand. He is positioned in the center of a hypnotic spiral background that consists of concentric circles in shades of orange and red, creating a tunnel-like effect. The text "That's all Folks!" is written in a white, cursive script across the lower half of the image, partially overlapping the man and the spiral.

That's all Folks!