

Esercitazione SQL

DDL e DML

DDL

Domanda

- Dato il seguente schema di basi di dati (le chiavi sono scritte in maiuscolo)

Cliente(CODCLIENTE, Nome, Indirizzo, Citta)

Ordine(NUMERO, Codcliente, Data, Importo)

PartiOrdine(NROORDINE, CODPRODOTTO, Quantita, PrezzoUnitario)

Prodotto(CODICE, Descrizione, QtaMagazzino)

Domanda

- Si scriva l'istruzione SQL che è in grado di creare la tabella Cliente sapendo che
 - Cod cliente e chiave primaria
- Si scriva l'istruzione SQL che è in grado di creare la tabella Ordine sapendo che:
 - Numero è di tipo intero ed è chiave primaria
 - CodCliente è di tipo intero ed è una chiave esterna di Client
 - Data è di tipo DATE e deve sempre avere un valore

```
CREATE TABLE CLIENTE (  
    CODCLIENTE INT PRIMARY KEY,  
    NOME VARCHAR(20) NOT NULL,  
    CONOME VARCHAR(20) NOT NULL,  
    INDIRIZZO VARCHAR(100),  
    CITTA    VARCHAR(100)  
)  
  
CREATE TABLE ORDINE(  
    NUMERO INT PRIMARY KEY,  
    CODICECLIENTE INT REFERENCES CLIENTE,  
    DATA DATE NOT NULL,  
    IMPORTO INT  
)
```

DML

Domanda

- Considerare una base di dati relativa a studenti ed esami da essi superati:
- Studenti(Matricola, Cognome, Nome)
- Esami(Studente, Materia, Voto, Data)
- con vincolo di integrità referenziale fra l'attributo Studente di Esami e la relazione Studenti.
- Formulare in algebra relazionale e sql l'interrogazione che trova matricola, cognome e nome degli studenti che hanno preso almeno un 30:

Risposta

$\pi_{Matricola, Cognome, Nome}(Studenti \bowtie_{Matricola=Studente} \sigma_{Voto=30}(Esami))$

```
SELECT Matricola, Cognome, Nome
FROM Studente JOIN Esami ON
  Matricola=Studente
WHERE Voto=30
```

Domanda

- Formulare in algebra relazionale e sql l'interrogazione trova gli studenti (mostrare il numero di matricola) che hanno superato almeno due esami dopo il 1/1/2000.

Risposta

$$\pi_{Studente}(\sigma_{Data > 1/1/2000 \wedge Data' > 1/1/2000 \wedge Materia <> Materia'}(Esami \bowtie_{Studente=Studente'}(\rho_{Studente',Materia',Data',Voto'} \leftarrow Studente,Materia,Data,Voto(Esami))))$$

```
SELECT Studente
FROM Esame E1 JOIN Esame E2 ON
  E1.Studente=E2.Studente
WHERE E1.Data>"1/1/2000" AND
      E2.Data>"1/1/2000" AND
      E1.Materia<>E2.Materia
```

Domanda

- Formulare in SQL l'interrogazione che mostra per ogni materia il voto medio degli esami svolti da studenti il cui cognome inizia per L

Risposta

```
SELECT Materia, AVG(Voto) as VotoMedio  
FROM Esami JOIN Studente ON  
    Matricola=Studente  
WHERE Cognome LIKE "L%"  
GROUP BY Materia
```

Domanda

- Formulare in SQL l'interrogazione che mostra l'elenco ordinato per numero di voti e voto medio gli studenti che hanno sostenuto almeno un esame di basi di dati.

Risposta

```
SELECT Nome, Cognome, Matricola,  
       Count(Materia) as NumeroEsami,  
       AVG(Voto) as Voto_Medio  
FROM Esami JOIN Studente ON  
       Matricola=Studente  
WHERE Materia LIKE "%Basi di dati%"  
GROUP BY Matricola  
Order by COUNT(Materia) DESC,  
       AVG(Voto) DESC
```

Domanda

- Formulare in SQL l'interrogazione che mostra l'elenco degli studenti che hanno superato l'ultimo appello di Elementi di Basi di Dati con una votazione pari a 30. (Usare una interrogazione nidificata)

Risposta

```
SELECT Nome, Cognome, Matricola
FROM Esami JOIN Studente ON
    Matricola=Studente
WHERE Materia = "Elementi di Basi di dati"
    AND Voto =30 AND
    Data IN (SELECT Max(Data)
        FROM Esami
    WHERE Materia="Elementi di Basi di
    Dati")
```