

SQL – Esercizi DML – Blocco 3

Studenti(Matricola, NomeS, CognomeS, CittàRes, Sesso, NumTelefono)

Corsi(CodCorso, NomeC, NumCrediti, MatricolaDocente)

EsamiSuperati(Matricola, CodCorso, Voto)

Docenti(MatricolaD, NomeD, CognomeD, Stipendio, Dipartimento)

Iscritti_Esami (Data, CodCorso, Matricola, Voto)

- 1) Selezionare per ogni studente matricola e numero di esami superati

```
Select matricola,count(*) as esamiuperati
from esamiuperati
group by matricola;
```

- 2) Selezionare matricola studente, codice corso e numero di volte che lo studente ha sostenuto l'esame del corso per tutti gli studenti e i relativi corsi.

```
select Matricola, CodCorso, COUNT(*)
from Iscritti_Esami
group by Matricola, CodCorso;
```

- 3) Selezionare la matricola degli studenti che hanno almeno un corso per cui hanno sostenuto almeno 5 volte l'esame

```
select DISTINCT Matricola
from Iscritti_Esami
group by Matricola, CodCorso
having count(*)>=5;
```

- 4) Selezionare la matricola degli studenti che hanno preso il voto più alto all'esame identificato dal codice corso R25.

```
select matricola
from EsamiSuperati
where CodCorso='R25'
and Voto = (select max(voto)
            from EsamiSuperati
            where CodCorso='R25');
```

Viste

- 5) Selezionare la matricola degli studenti che hanno superato il maggior numero d'esami

```
create view StudentiNumeroEsami
AS
Select matricola,count(*) as numesamisuperati
from esamiuperati
group by matricola;
```

```

select matricola
from StudentiNumeroEsami
where numesamisuperati=(select max(numesamisuperati)
                        from StudentiNumeroEsami
                        );

```

- 6) Selezionare matricola, nome e cognome dei docenti che hanno uno stipendio maggiore rispetto allo stipendio medio dei propri colleghi di dipartimento (docenti dello stesso dipartimento)

```

create view StipendioMedioPerDipartimento

AS
Select dipartimento,avg(stipendio) as StipendioMedio
from docenti
group by dipartimento;

select matricolaD, nomed, cognomeD
from docenti D, StipendioMedioPerDipartimento S
where D.dipartimento=S.dipartimento
and D.Stipendio>S.StipendioMedio;

```

Query annidate

- 7) Selezionare la matricola degli studenti che hanno superato l'esame di tutti i corsi possibili (tutti i corsi presenti nella tabella CORSI). Si supponga l'esistenza di almeno un corso (esiste almeno una tupla nell'istanza della tabella CORSI)

```

Select matricola
from esamisuperati
group by matricola
having count(*)=(select count(*)
                 from corsi);

```

Interrogazioni binarie/insiemiche

UNION

- 8) Selezionare la matricola degli studenti che hanno superato l'esame di informatica con 30 o quello di fisica con 25.
- a. Select matricola
 from esamisuperati e, corsi c
 where e.CodCorso=c.CodCorso
 and NomeC='Informatica' and Voto=30
 UNION
 Select matricola
 from esamisuperati e, corsi c

where e.CodCorso=c.CodCorso
and NomeC='Fisica' and Voto=25

- b. Select matricola
from studenti
where matricola in (Select matricola
from esamisuperati e, corsi c
where e.CodCorso=c.CodCorso
and NomeC='Informatica' and Voto=30)
or matricola in (Select matricola
from esamisuperati e, corsi c
where e.CodCorso=c.CodCorso
and NomeC='Fisica' and Voto=25);

INTERSECT

- 9) Selezionare la matricola degli studenti che hanno superato l'esame di informatica con 30 e quello di fisica con 25.

- a. Select matricola
from esamisuperati e, corsi c
where e.CodCorso=c.CodCorso
and NomeC='Informatica' and Voto=30

INTERSECT

Select matricola
from esamisuperati e, corsi c
where e.CodCorso=c.CodCorso
and NomeC='Fisica' and Voto=25

- b. Select matricola
from studenti
where matricola in (Select matricola
from esamisuperati e, corsi c
where e.CodCorso=c.CodCorso
and NomeC='Informatica' and Voto=30)
and matricola in (Select matricola
from esamisuperati e, corsi c
where e.CodCorso=c.CodCorso
and NomeC='Fisica' and Voto=25);

EXCEPT

- 10) Selezionare la matricola degli studenti che non hanno mai preso 30

- a. select matricola
from student
except
select matricola
from esamisuperati
where Voto=30;

Esercizi tratti da temi d'esame

Si consideri il seguente schema di base di dati che vuole tenere traccia delle informazioni riguardanti i campionati mondiali di atletica.

CAMPIONATO (EdizioneCamp, Anno, DataInizio, DataFine, CittàOspitante, StatoOspitante)

GARA (EdizioneCamp, NumGara, Disciplina, Data, Oralnizio)

ATLETA (CodiceAtleta, Nome, Cognome, Sesso, DataNascita, Nazionalità)

CLASSIFICA_GARA (EdizioneCamp, NumGara, CodiceAtleta, Posizione)

Dove il campo Posizione è un intero.

Rispondere alle seguenti interrogazioni utilizzando il linguaggio SQL

1. *Trovare nome e cognome degli atleti italiani che hanno gareggiato in almeno 4 diversi campionati mondiali di atletica, ma che non si sono mai classificati in prima posizione nelle gare cui hanno partecipato.*
2. *Trovare codice, nome e cognome degli atleti che hanno vinto il maggior numero di gare nel campionato mondiale dell'anno 2005.*

Soluzione

1.

```
SELECT Nome, Cognome
FROM ATLETA A, CLASSIFICA_GARA CG
WHERE A.CodiceAtleta=CG.CodiceAtleta
      AND Nazionalità='Italiana'
      AND CodiceAtleta NOT IN ( SELECT CodiceAtleta
                                FROM CLASSIFICA_GARA
                                WHERE Posizione=1
                              )
GROUP BY A.CodiceAtleta, Nome, Cognome
HAVING COUNT(DISTINCT EdizioneCamp)>=4;
```
2.

```
CREATE VIEW GareVinteAnno2005 (CodiceAtleta, NumGareVinte)
AS
SELECT CodiceAtleta, COUNT(*) AS NumGareVinte
FROM CLASSIFICA_GARA CG, CAMPIONATO C
WHERE CG.EdizioneCamp=C.EdizioneCamp
      AND Anno=2005
      AND Posizione=1
GROUP BY CodiceAtleta;

SELECT A.CodiceAtleta, Nome, Cognome
FROM ATLETA A, GareVinteAnno2005 GV
WHERE A.CodiceAtleta=GV.CodiceAtleta
      AND GV.NumGareVinte= (SELECT MAX(NumGareVinte)
                            FROM GareVinteAnno2005
                          );
```