

1. Mostrare nome e descrizione di tutti i moduli da 9 CFU

*

1. Mostrare nome e descrizione di tutti i moduli da 9 CFU

*\

```
SELECT nome, descrizione
FROM modulo
WHERE cfu = 9
```

proj(nome, descrizione) (Sel cfu=9 (modulo))

2. Mostrare matricola, nome e cognome dei docenti che hanno più di 60 anni

proj(matricola, nome, cognome) (sel data_nascita < 1960-01-01 (docente))

*

2. Mostrare matricola, nome e cognome dei docenti che hanno più di 60 anni

*\

```
SELECT matricola, nome, cognome
FROM docente
WHERE data_nascita < '1960-01-01'
```

3. Mostrare nome, cognome e nome del dipartimento di ogni docente, ordinati dal più giovane al più anziano.

proj(nome, cognome, dipartimento).....non posso ordinare

*

3. Mostrare nome, cognome e nome del dipartimento di ogni docente, ordinati dal più giovane al più anziano.*\

```
SELECT nome, cognome, dipartimento
FROM docente
ORDER BY data_nascita
```

4. Mostrare città e indirizzo di ogni sede del dipartimento di codice "uniba"
prog(città, indirizzo) (sede) (sel codice='uniba' (sede))

```
\*  
4. Mostrare città e indirizzo di ogni sede del  
dipartimento di codice "uniba"  
.*\  
SELECT città, indirizzo  
FROM sede  
WHERE codice='uniba'
```

5. Mostrare nome del dipartimento, città e indirizzo di ogni sede di ogni dipartimento, ordinate alfabeticamente prima per nome dipartimento, poi per nome città e infine per indirizzo.

Non ho order by in algebra relazionale.

```
\*  
5. Mostrare nome del dipartimento, città e indirizzo di  
ogni sede di ogni dipartimento, ordinate alfabeticamente  
prima per nome dipartimento, poi per nome città e infine  
per indirizzo.  
.*\  

```

```
select D.nome, S.città, S.indirizzo  
from sede as S join sede_dipartimento as SD on  
S.codice=SD.codice_sede JOIN dipartimento as D on  
SD.codice_dipartimento=D.codice  
ORDER BY D.nome;
```

```
select D.nome, S.città, S.indirizzo  
from sede as S join sede_dipartimento as SD on  
S.codice=SD.codice_sede JOIN dipartimento as D on  
SD.codice_dipartimento=D.codice  
ORDER BY S.città;
```

```
select D.nome, S.città, S.indirizzo  
from sede as S join sede_dipartimento as SD on  
S.codice=SD.codice_sede JOIN dipartimento as D on  
SD.codice_dipartimento=D.codice  
ORDER BY S.indirizzo;
```

2. 6. Mostrare il nome di ogni dipartimento che ha una sede a Bari.

```
Proj( (D.nome) sede as S (SEL sede='bari')
JOIN(S.codice=SD.codice_sede) sede_dipartimento as SD
JOIN(SD.codice=D.codice) dipartimento as D )
```

*

6. Mostrare il nome di ogni dipartimento che ha una sede a Bari.

.*\

```
SELECT D.nome
FROM sede as S JOIN sede_dipartimento as SD on S.codice =
SD.codice_dipartimento JOIN dipartimento as D on
SD.codice_dipartimento=D.codice
WHERE S.citta = 'bari'
```

3. 7. Mostrare il nome di ogni dipartimento che non ha sede a bari.

```
Proj( (D.nome) sede as S JOIN(S.codice=SD.codice_sede)
sede_dipartimento as SD JOIN(SD.codice=D.codice) dipartimento
as D )
EXCEPT
Proj( (D.nome) sede as S(SEL citta='bari')
JOIN(S.codice=SD.codice_sede) sede_dipartimento as SD
JOIN(SD.codice=D.codice) dipartimento as D )
```

```
SELECT citta
FROM sede
WHERE not exists (SELECT *
                  FROM sede
                  WHERE citta='brindisi')
```

*

7. Mostrare il nome di ogni dipartimento che non ha sede a bari.*\

```
select D.nome from sede as S JOIN sede_dipartimento as SD on
S.codice = SD.codice_sede JOIN dipartimento as D on
SD.codice_dipartimento = D.codice where not exists (SELECT codice
FROM sede WHERE citta='bari');
```

1. 8. Mostrare media, numero esami sostenuti e totale CFU acquisiti dello studente con matricola 123456.

In algebra relazionale non ho avg e count

*

8. Mostrare media, numero esami sostenuti e totale CFU acquisiti dello studente con matricola 123456.

.*\

```
select avg(voto), count(matricola_studente), sum(M.cfu) from esame  
JOIN modulo as M on codice_modulo=M.codice where  
matricola_studente='123456';
```

1. 9. Mostrare nome, cognome, nome del corso di laurea, media e numero esami sostenuti dello studente con matricola 123456.

*

9. Mostrare nome, cognome, nome del corso di laurea, media e numero esami sostenuti dello studente con matricola 123456.

.*\

```
select S.nome, S.cognome, avg(voto),  
count(matricola_studente), CL.nome as corso_di_laurea from studente  
as S JOIN corso_laurea as CL on S.corso_laurea=CL.codice JOIN  
esame on S.matricola=matricola_studente JOIN modulo as M on  
codice_modulo=M.codice where matricola_studente='123456';
```

1. 10. Mostrare codice, nome e voto medio di ogni modulo, ordinati dalla media più alta alla più bassa.

*

10. Mostrare codice, nome e voto medio di ogni modulo, ordinati dalla media più alta alla più bassa.

.*\

```
select E.codice_modulo, avg(E.voto) as voto_medio, M.nome as  
modulo FROM esame AS E JOIN modulo as M on  
E.codice_modulo=M.codice GROUP BY E.codice_modulo, M.nome ORDER BY  
avg(E.voto) DESC;
```

- 1.
- 2.
- 3.
- 4.
- 5.

6. 11. Mostrare nome e cognome del docente, nome e descrizione del modulo per ogni docente ed ogni modulo di cui quel docente abbia tenuto almeno un esame; il risultato deve includere anche i docenti che non abbiano tenuto alcun esame, in quel caso rappresentati con un'unica tupla in cui nome e descrizione del modulo avranno valore NULL.

TOP-DOWN

```
proj(nome,cognome)(docente)
proj(nome_modulo, descrizione)(docente)
Ren nome_modulo<-nome
docente left join esame(matricola=matricola_docente)
Left join (codice_modulo=codice) modulo
```

*

11. Mostrare nome e cognome del docente, nome e descrizione del modulo per ogni docente ed ogni modulo di cui quel docente abbia tenuto almeno un esame; il risultato deve includere anche i docenti che non abbiano tenuto alcun esame, in quel caso rappresentati con un'unica tupla in cui nome e descrizione del modulo avranno valore NULL.

.*\

```
SELECT D.nome, D.cognome, M.nome as nome_modulo, M.descrizione
FROM docente as D LEFT OUTER JOIN esame as E on
D.matricola=E.matricola_docente LEFT OUTER JOIN modulo as M on
E.codice_modulo = M.codice;
```

1. 12. Mostrare matricola, nome, cognome, data di nascita, media e numero esami sostenuti di ogni studente.

*

12. Mostrare matricola, nome, cognome, data di nascita, media e numero esami sostenuti di ogni studente.

.*\

```
select S.nome, S.cognome, S.matricola, S.data_nascita, avg(E.voto)
as media_studente, count(E.matricola_studente) as esami_sostenuti
from studente as S JOIN esame as E on
S.matricola=E.matricola_studente GROUP BY
S.nome,S.cognome,S.matricola;
```

13. Mostrare matricola, nome, cognome, data di nascita, media e numero esami sostenuti di ogni studente del corso di laurea di codice "ICD" che abbia media maggiore di 27.

*

13. Mostrare matricola, nome, cognome, data di nascita, media e numero esami sostenuti di ogni studente.

.*\

```
SELECT S.nome, S.cognome, S.matricola, S.data_nascita, avg(E.voto)
as media_studente, count(E.matricola_studente) as esami_sostenuti
FROM studente as S JOIN esame as E on
S.matricola=E.matricola_studente WHERE S.corso_laurea='ICD' GROUP
BY S.nome,S.cognome,S.matricola HAVING avg(E.voto) > 27;
```

1. 14. Mostrare nome, cognome e data di nascita di tutti gli studenti che ancora non hanno superato nessun esame.

```
proj(nome, cognome, data_nascita)
Studente left join(S.matricola=E.matricola)
sel(codice_modulo IS NULL)
```

*

14. Mostrare nome, cognome e data di nascita di tutti gli studenti che ancora non hanno superato nessun esame.

.*\

```
select distinct S.nome, S.cognome, S.data_nascita from studente as
S LEFT OUTER JOIN esame as E on S.matricola=E.matricola_studente
WHERE E.codice_modulo IS NULL;
```

1. 15. Mostrare la matricola di tutti gli studenti che hanno superato almeno un esame e che hanno preso sempre voti maggiori di 26.

*

15. Mostrare la matricola di tutti gli studenti che hanno superato almeno un esame e che hanno preso sempre voti maggiori di 26.

.*\

```
select distinct S.nome, S.cognome, S.data_nascita from studente as
S LEFT OUTER JOIN esame as E on S.matricola=E.matricola_studente
WHERE E.codice_modulo IS NOT NULL GROUP BY S.nome,S.cognome,
S.data_nascita HAVING avg(E.voto) > 26;
```

16. Mostrare, per ogni modulo, il numero degli studenti che hanno preso tra 18 e 21, quelli che hanno preso tra 22 e 26 e quelli che hanno preso tra 27 e 30 (con un'unica interrogazione).

— — — —

17. Mostrare matricola, nome, cognome e voto di ogni studente che ha preso un voto maggiore della media nel modulo "BDD"

```
SELECT S.matricola, S.nome, S.cognome, E.voto
FROM studente as S JOIN esame as E on
S.matricola=E.matricola_studente
JOIN modulo as M on E.codice_modulo=M.codice
WHERE M.nome='database';
```

```
SELECT S.matricola, S.nome, S.cognome, E.voto FROM studente as S
JOIN esame as E on S.matricola=E.matricola_studente JOIN modulo
as M on E.codice_modulo=M.codice WHERE M.nome='database';
```

```
SELECT S.matricola, S.nome, S.cognome, E.voto
FROM studente as S JOIN esame as E on
S.matricola=E.matricola_studente
JOIN modulo as M on E.codice_modulo=M.codice
WHERE M.nome='database'
GROUP BY S.matricola, S.nome, S.cognome, E.voto;
```

```
SELECT S.matricola, S.nome, S.cognome, E.voto FROM studente as S
JOIN esame as E on S.matricola=E.matricola_studente JOIN modulo
as M on E.codice_modulo=M.codice WHERE M.nome='database' GROUP BY
S.matricola, S.nome, S.cognome, E.voto;
```

```
SELECT S.matricola, S.nome, S.cognome, E.voto
FROM studente as S JOIN esame as E on
S.matricola=E.matricola_studente
JOIN modulo as M on E.codice_modulo=M.codice
WHERE M.nome='database'
HAVING avg(E.voto) > 20
GROUP BY S.matricola, S.nome, S.cognome, E.voto;
```