

SQL

1. Obtén os datos completos dos alumnos que nunca escribiron no foro.

```
select *  
from alumno  
where email not in  
(  
select email  
from men_foro  
where email is not null  
);
```

2. Obtén o nome do profesor que máis mensaxes escribiu.

```
select nome  
from profesor  
where nss=  
(  
select nss  
from men_foro  
group by nss  
having count(nss)=  
(  
select max(count(nss))  
from men_foro  
where nss is not null  
group by nss  
)  
);
```

3. Para cada profesor (nome) mostra o nome do curso, de entre todos os que está capacitado para impartir, que dura máis horas.

```
select distinct p.nome, c.nome  
from curso c join capacitado ca on c.cod_c=ca.cod_c  
join profesor p on ca.nss=p.nss  
where c.horas=
```

```
(
select max(horas)
from curso c1 join capacitado ca1 on c1.cod_c=ca1.cod_c
join profesor p1 on ca1.nss=p1.nss
where p.nome=p1.nome
);
```

4. Indica cantos estudantes se rexistraron en cada edición de cada curso. Mostra o código e nome do curso, o número e ano da edición, e o número de estudantes (0, se non hai ningún).

```
select cu.cod_c, cu.nome, e.numero, count(r.email)
from rexistrase r join edicion e on r.cod_c=e.cod_c and r.numero=e.numero
join curso cu on e.cod_c=cu.cod_c
group by cu.cod_c, cu.nome, e.numero;
```

5. Indica, para cada curso, aquelas edicións caras (con prezo superior a 100 euros). Mostra o código e nome do curso, e a data de inicio da edición. Se un curso non ten edicións caras, debes mostralo igualmente, cos datos da edición en branco.

```
select cu.cod_c, cu.nome, e.prezo, e.data_comezo
from edicion e right join curso cu on e.cod_c=cu.cod_c
where e.prezo>100;
```

6. Indica, para cada curso, a edición que tivo máis estudantes rexistrados. Mostra o código e nome do curso, e o número da edición. (Só cursos con estudantes rexistrados).

```
select cu.nome, e.cod_c, e.numero
from rexistrase r join edicion e on r.cod_c=e.cod_c and r.numero=e.numero
join curso cu on e.cod_c=cu.cod_c
group by e.cod_c, cu.nome, e.numero
having count(e.numero)=
(
select max(count(r1.numero))
from rexistrase r1 join edicion e1 on r1.cod_c=e1.cod_c and
r1.numero=e1.numero
join curso cu1 on e1.cod_c=cu1.cod_c
where e1.cod_c=e.cod_c
```

```
group by r1.numero  
);
```

7. Mostra o código e nome de cada curso impartido, e o nome do método de pagamento máis utilizado para rexistrarse no dito curso. (Só cursos con estudantes rexistrados).

```
select cu.nome, e.cod_c, r.metpago  
from rexistrase r join edicion e on r.cod_c=e.cod_c and r.numero=e.numero  
join curso cu on e.cod_c=cu.cod_c  
group by cu.nome, e.cod_c, r.metpago  
having count(r.metpago)=  
(  
select max(count(r1.metpago))  
from rexistrase r1 join edicion e1 on r1.cod_c=e1.cod_c and  
r1.numero=e1.numero  
join curso cu1 on e1.cod_c=cu1.cod_c  
where cu1.nome=cu.nome  
group by r1.metpago  
);
```

8. Mostrar para cada curso, o seu nome e a edición/s con maior prezo.

```
select cu.nome, e.numero, e.prezo  
from edicion e join curso cu on e.cod_c=cu.cod_c  
where (e.cod_c, e.prezo) in  
(  
select cod_c, max(prezo)  
from edicion e  
group by cod_c  
);
```

9. Mostrar o/s tema/s (o seu identificador e asunto) con menor número de mensaxes sen resposta.

```
select t.id_tema, t.asunto  
from men_foro m join pertence p on m.id_men=p.id_men  
join tema t on t.id_tema=p.id_tema  
where m.id_men not in  
(
```

```

select resposta_de
from men_foro
where resposta_de is not null
)
group by t.id_tema, t.asunto
having count(m.resposta_de)=
(
select min(count(*))
from men_foro m1 join pertenece p1 on m1.id_men=p1.id_men
join tema t1 on t1.id_tema=p1.id_tema
where m1.id_men not in
(
select resposta_de
from men_foro
where resposta_de is not null
)
group by t1.id_tema
);

```

10. Nome de curso e edición que teña o menor número de alumnos rexistrados.

```

select cu.cod_c, cu.nome, e.numero, count(r.email)
from curso cu join edicion e on cu.cod_c=e.cod_c
left join rexistrase r on e.cod_c=r.cod_c and e.numero=r.numero
group by cu.cod_c, cu.nome, e.numero
having count(r.email)=
(
select min(count(r1.email))
from curso cu1 join edicion e1 on cu1.cod_c=e1.cod_c
left join rexistrase r1 on e1.cod_c=r1.cod_c and e1.numero=r1.numero
group by cu1.cod_c, cu1.nome, e1.numero
);

```

11. Nomes dos alumnos que escribiron en temas de cursos nos que xamais estiveron rexistrados.

```

select distinct a.nome
from alumno a join men_foro m on m.email = a.email
join pertenece p on p.id_men = m.id_men
join tema t on t.id_tema = p.id_tema

```

```

where a.email not in
(
select email
from rexistrase
where cod_c =t.cod_c
);

```

12. Para cada tema mostrar os profesores que máis mensaxes escribiron relacionados con el.

```

select t.id_tema, m.nss
from men_foro m join pertenece p on m.id_men=p.id_men
join tema t on p.id_tema=t.id_tema
group by t.id_tema, m.nss
having count(*)=
(
select max(count(*))
from men_foro m1 join pertenece p1 on m1.id_men=p1.id_men
join tema t1 on p1.id_tema=t1.id_tema
where m1.nss is not null
group by m1.nss
);

```

13. Numero medio de mensaxes ao día.

```

select avg(count(id_men))
from men_foro
group by data;

```

23. Mostra aqueles docentes (nss, nome) capacitados para impartir TODOS os cursos rexistrados na BD.

```

select p.nss, p.nome
from profesor p join capacitado c on p.nss=c.nss
group by p.nss, p.nome
having count(c.cod_c)=
(
select count(*)

```

```
from curso
);
```

24. Mostra aqueles docentes (nss, nome) que NON ESTÁN capacitados para impartir TODOS os cursos rexistrados na BD.

```
select p.nss, p.nome
from profesor p join capacitado c on p.nss=c.nss
group by p.nss, p.nome
having count(c.cod_c) !=
(
select count(*)
from curso
);
```

27. Para cada edición mostrar o número de temas e alumnos rexistrados. No caso de que non teñan temas ou alumnos ten que aparecer un 0. Debe aparecer o nome do curso.

```
select e.cod_c, e.numero, cu.nome, count(distinct r.email), count(distinct
id_tema)
from rexistrase r right join edicion e on r.cod_c=e.cod_c and
r.numero=e.numero
left join curso cu on e.cod_c=cu.cod_c
left join tema t on e.cod_c=t.cod_c and e.numero=t.numero
group by e.cod_c, e.numero, cu.nome;
```

32. Para cada edición de curso cantos mensaxes de alumnos foron respondidos polos profesores. Mostrar nome do curso e número de edición.

```
select c.nome, e.numero, count(distinct men_al.id_men)
from curso c join edicion e on c.cod_c = e.cod_c
join tema t on e.cod_c = t.cod_c and e.numero = t.numero
join pertenece p on p.id_tema = t.id_tema
join men_foro men_al on men_al.id_men = p.id_men and men_al.nss
is null
join men_foro resp_p on men_al.id_men = resp_p.resposta_de and
resp_p.nss is not null
group by c.cod_c, c.nome, e.numero;
```

37. Para cada alumno amosa o importe medio das súas matrículas e a diferenza co importe medio de todas as matrículas de todos alumnos.

```
select a.nome, avg(e.prezo), abs(avg(e.prezo)-
(
select avg(e1.prezo)
from alumno a1 join rexistrase r1 on a1.email=r1.email
join edicion e1 on r1.cod_c=e1.cod_c and r1.numero=e1.numero
))
from alumno a join rexistrase r on a.email=r.email
join edicion e on r.cod_c=e.cod_c and r.numero=e.numero
group by a.nome;
```