

Lenguaje de Definición de Datos (DDL)

CREATE →
ALTER →
DROP →
RENAME →

Lenguaje de Control de Transacciones (TCL)

COMMIT →
ROLLBACK →

Lenguaje de Manipulación de Datos (DML)

SELECT →
INSERT →
UPDATE →
DELETE →

Lenguaje de Control de Datos (DCL)

GRANT →
REVOKE →

SELECT

```
SELECT [ALL | DISTINCT] <nombre_campo> [{,<nombre_campo>}]  
FROM    <nombre_tabla>                [{,<nombre_tabla> }]  
[WHERE   <predicado>  
[GROUP BY <nombre_campo>                [{,<nombre_campo>}}]  
[HAVING  <predicado>  
[ORDER BY <nombre_campo>[ASC|DESC] [{,<nombre_campo> [ASC|DESC]}]]
```

UPDATE

```
UPDATE <nombre_tabla>  
SET    <nombre_campo> = <expresión>  
[WHERE <predicado>
```

```
UPDATE instructor  
SET salary = salary*1.05;  
WHERE salary < 70000
```

INSERT

```
INSERT INTO <nombre_tabla> (<nombre_campo> [{,<nombre_campo>}])  
VALUES (<valor> [{,<valor>}])
```

```
INSERT INTO course (course_id, title, dept_name, credits)  
VALUES ('CS-437', 'Bases de datos', 'Informatica', 4)
```

DELETE

```
DELETE FROM <nombre_tabla>  
[WHERE <predicado>
```

```
DELETE FROM instructor  
WHERE salary BETWEEN 13000 AND 15000;
```

CREATE

```
CREATE TABLE <nombre_tabla>  
[( <nombre_campo> <tipo> [{,<nombre_campo> <tipo> } ] )]
```

```
CREATE TABLE departamento
```

ALTER

```
ALTER TABLE <nombre_tabla>  
ADD <nombre_campo> <tipo> | DROP <nombre_campo>
```

```
ALTER TABLE instructor ADD altura INT
```

```
ALTER TABLE instructor DROP altura
```

DROP

```
DROP TABLE <nombre_tabla>
```

```
DROP TABLE instructor
```

1. Encuentra los nombres de todos los instructores del departamento de biología.

```
select name
from instructor
where instructor.dept_name = 'Biology'
```

2. Encuentra los nombres de los cursos en el departamento de informática que tienen 3 créditos.

```
select title
from department natural join course
where department.dept_name = 'Comp. Sci.' and course.credits=3
```

3. Para el estudiante con ID 12345 (o cualquier otro valor), muestre todos los ID de curso y el título de todos los cursos registrados por el estudiante.

```
select course_id, title
from student natural join takes natural join course
where ID = '12345'
```

4. Como se indicó anteriormente, pero muestre el número total de créditos para dichos cursos (tomados por ese estudiante). No muestre el valor de tot_creds en la tabla de estudiantes, debe usar la agregación de SQL en los cursos tomados por el estudiante.

```
select ID, sum(credits)
from student natural join takes natural join course
where ID = '12345'
```

5. Como se indicó anteriormente, pero muestre el total de créditos para cada uno de los estudiantes, junto con la identificación del estudiante; No te preocupes por el nombre del estudiante. (No se preocupe por los estudiantes que no se han inscrito en ningún curso, se pueden omitir)

```
select ID, sum(credits)
from student natural join takes natural join course
group by ID
```

6. Encuentra los nombres de todos los estudiantes que han tomado algún Comp. Sci. curso siempre (no debe haber nombres duplicados)
7. Muestre las ID de todos los instructores que nunca han enseñado una cláusula (Notesad1) Oracle usa la palabra clave menos en lugar de excepto; (2) interpretar "enseñado" como "enseñado o programado para enseñar")
8. Igual que arriba, pero también muestra los nombres de los instructores, no solo los ID.
9. Necesitas crear una base de datos de películas. Cree tres tablas, una para actores (AID, nombre), una para películas (MID, título) y otra para actor_role (MID, AID, rolename). Use los tipos de datos adecuados para cada uno de los atributos y agregue restricciones de clave primaria / foránea apropiadas.
10. Inserte datos en las tablas anteriores (aproximadamente de 3 a 6 filas en cada tabla), incluidos los datos del actor "Charlie Chaplin", y para usted mismo (usando su número de rollo como ID).
11. Escribe una consulta para enumerar todas las películas en las que el actor "Charlie Chaplin" ha actuado, junto con la cantidad de papeles que tuvo en esa película.
12. Escribe una consulta para enumerar a todos los actores que no han actuado en ninguna película.
13. Enumere los nombres de los actores, junto con los títulos de las películas en las que actuaron. Si no actuaron en ninguna película, muestre el título de la película como nulo. (No use la sintaxis de la unión externa SQL aquí, escríbala desde cero).

1. Obtener los nombres de todos los profesores junto con los nombres de sus departamentos y el nombre del edificio donde se encuentra el departamento

```
select name, profesor.dept_name, building
from profesor, departamento
where profesor.dept_name=departamento.dept_name
```

2. Encontrar los nombres de los profesores y los identificadores de asignatura de los profesores del departamento de Informática

```
select name, course_id
from profesor, enseña
where profesor.ID=enseña.ID and profesor.dept_name='Comp. Sci.'
```

3. Para todos los profesores de la universidad que hayan enseñado alguna asignatura, encontrar sus nombres y course_id de todas las asignaturas que hayan enseñado

```
select profesor.name, enseña.course_id
from profesor, enseña
where profesor.ID=enseña.ID'
```

4. Encontrar los nombres de todos los profesores cuyo sueldo sea mayor que al menos un profesor del departamento de Biología

```
select distinct T.name
from profesor as T, profesor as S
where T.salary>S.salary and S.dept_name='Biology'
```

5. Encontrar los nombres de todos los departamentos cuyo nombre de edificio incluye la subcadena 'Watson'

```
select dept_name
from departamento
where building like '%Watson%'
```

6. Todos los profesores del departamento de física ordenados por orden alfabético

```
select name
from profesor
where dept_name='Physics'
order by name;
```

7. Listar todos los profesores en orden descendente de sueldo, y si varios profesores tienen el mismo sueldo, se ordenan ascendente de nombre

```
select *
from profesor
order by salary desc, name asc
```

8. Encontrar los nombres de los profesores con sueldos entre 90000 y 100000€

```
select name
from profesor
where salary between 90000 and 100000
```

9. Encontrar los nombres de los profesores y las asignaturas que enseñan para todos los profesores del departamento de Biología que hayan enseñado alguna asignatura

```
select name, course_id
from profesor, enseña
where profesor.id = enseña.ID and dept_name='Biology'
```

10. El conjunto de todas las asignaturas que se enseñan en el semestre de otoño de 2017

```
select course_id
from seccion
where semester = 'Fall' and year=2017
```

11. Encontrar todas las asignaturas que se enseñaron en el otoño de 2017 o en la primavera de 2018, o en ambas.

```
select course_id
from seccion
where semester = 'Fall' and year =2017
union
select course_id
from seccion
where semester='Spring' and year=2018
```

12. Encontrar todas las asignaturas que se enseñan en el otoño de 2017, así como en la primavera de 2018.

```
select course_id
from seccion
where semester = 'Fall' and year =2017
intersect
select course_id
from seccion
where semester='Spring' and year=2018
```

13. Encontrar el sueldo medio de los profesores del departamento de Informática.

```
select avg(salary)
from profesor
where dept_name='Comp. Sci.'
```

14. Encontrar el número total de profesores que enseñaron alguna asignatura en el semestre de primavera de 2018

```
select count (distinct ID)
from enseña
where semester = 'Spring' and year=2018
```

15. Encontrar el sueldo medio de cada departamento

```
select dept_name, avg(salary) as med_sueldo
from profesor
group by dept_name
```

16. Encontrar el número de profesores de cada departamento que enseñó una asignatura en el semestre de cada departamento que enseñó una asignatura en el semestre de primavera de 2018

```
select dept_name, count(distinct ID) as prof_cuenta  
from profesor natural join enseña  
where semester = 'Spring' and year=2018  
group by dept_name
```

17. Encontrar el sueldo medio en cada departamento en los que el sueldo medio de los profesores es superior a 42000 €

```
select dept_name, avg(salary) as med_sueldo  
from instructor  
group by dept_name  
having avg(salary)>42000
```

INTERMEDIO

1. Encuentre la inscripción máxima y mínima en todas las secciones, considerando solo las secciones que tuvieron alguna inscripción, no se preocupe por aquellos que no tuvieron estudiantes que tomaron esa sección

```
select ID, course_id, sec_id, semester, year, count(*)  
from matricula natural estudiante  
group by sec_id, semester, year
```

2. Encuentre todas las secciones que tenían la inscripción máxima (junto con la inscripción), usando una subconsulta.
3. Como en la Q1, pero ahora también incluye secciones sin alumnos que las tomen; la inscripción para dichas secciones debe tratarse como 0. Haga esto de dos maneras diferentes (y cree los datos requeridos para la prueba)
 - a. Usando una subconsulta escalar
 - b. Uso de la agregación en una combinación externa izquierda (use la sintaxis natural de la combinación externa izquierda de SQL)
4. Encuentre todos los cursos cuyo identificador comience con la cadena "CS-1"
5. Encuentra instructores que hayan enseñado todos los cursos anteriores.
 - a. Usando la estructura "no existe ... excepto ..."
 - b. Usando la comparación de los conteos que cubrimos en clase (;no olvide la cláusula distinta!).
- 6.
7. Inserte cada instructor como estudiante, con tot_creds = 0, en el mismo departamento
8. Ahora borre todos los "estudiantes" recién agregados arriba (nota: los estudiantes ya existentes que tuvieron tot_creds = 0 no deberían ser eliminados)
9. Algunos de ustedes pueden haber notado que el valor de tot_creds para los estudiantes no coincide con los créditos de los cursos que han tomado. Escriba y ejecute la consulta para actualizar tot_creds en función de los créditos aprobados, para que la base de datos vuelva a la coherencia. (Esta consulta se proporciona en el libro / diapositivas).
10. Actualice el salario de cada instructor a 10000 veces el número de secciones del curso que han enseñado.
11. Cree su propia consulta: defina lo que quiere hacer en inglés, luego escriba la consulta en SQL. Hazlo tan difícil como desees, cuanto más difícil, mejor.