

BASES DE DATOS
1º DAM

Nombre y apellidos:

Examen 1º evaluación

Importante: para aprobar el examen es necesario sacar un mínimo de 2 puntos en cada una de las partes (teoría y práctica).

TEORÍA (5 PUNTOS)
0.2p cada pregunta

1. ¿Qué es una entidad en el modelo E/R?
 - a) Un atributo que representa un objeto.
 - b) Un conjunto de relaciones entre atributos.
 - c) **Un objeto real o abstracto del que se desea almacenar información.**
 - d) Un grupo de datos no relacionados.
2. ¿Cuál es la función de un atributo en una entidad?
 - a) Representar la relación entre dos entidades.
 - b) **Definir características específicas de la entidad.**
 - c) Agrupar varias entidades.
 - d) Crear una clave primaria.
3. ¿Qué es una relación reflexiva en el modelo E/R?
 - a) Una relación entre tres entidades distintas.
 - b) **Una relación donde una entidad se relaciona consigo misma.**
 - c) Una relación de especialización entre dos entidades.
 - d) Una relación que solo admite una cardinalidad de tipo (1,1).
4. ¿Cuál de los siguientes ejemplos representa una dependencia en identificación?
 - a) Un empleado puede existir independientemente de su departamento.
 - b) **Una línea de pedido necesita el número de pedido para identificarse.**
 - c) Un cliente realiza varias compras en diferentes tiendas.
 - d) Un producto pertenece a múltiples categorías.
5. ¿Qué es el atributo identificador principal en una entidad?
 - a) Un atributo que define una relación.
 - b) Un conjunto de relaciones.
 - c) **Un identificador único para cada instancia de la entidad.**
 - d) Un atributo opcional.

6. En una relación reflexiva, ¿qué representa la cardinalidad (1,n)?
- a) Una instancia de la entidad puede relacionarse con una o varias otras instancias de la misma entidad.
 - b) Cada instancia de la entidad debe relacionarse con solo una instancia de otra entidad.
 - c) La relación es opcional y no es necesaria en ambas direcciones.
 - d) Una entidad no puede relacionarse consigo misma.
7. ¿Qué es una ocurrencia de una entidad?
- a) Un conjunto de atributos.
 - b) Una instancia o registro específico de esa entidad.
 - c) Una relación con otras entidades.
 - d) Un conjunto de claves primarias.
8. ¿Qué es un atributo derivado?
- a) Un atributo que puede tener varios valores.
 - b) Un atributo cuyo valor se calcula a partir de otros atributos.
 - c) Un atributo que identifica una instancia.
 - d) Un atributo que es obligatorio.
9. En el modelo relacional, una tabla también se denomina:
- a) Atributo
 - b) Relación
 - c) Dominio
 - d) Registro
10. ¿Qué es una tupla en el modelo relacional?
- a) Una columna en una tabla
 - b) Una combinación de varias tablas
 - c) Una fila específica en una tabla
 - d) Un tipo de clave primaria
11. ¿Cómo se llama el conjunto de valores que puede tomar un atributo?
- a) Relación
 - b) Esquema
 - c) Dominio
 - d) Registro
12. ¿Qué es una clave primaria en el modelo relacional?
- a) Un atributo que puede tomar múltiples valores
 - b) Una combinación de atributos única para identificar una tupla
 - c) Un atributo opcional en una tabla
 - d) Un campo que almacena los valores duplicados
13. ¿Cuál es la función de una clave ajena (o foránea) en el modelo relacional?
- a) Identificar de forma única cada tupla en una tabla
 - b) Crear un enlace entre dos tablas relacionadas
 - c) Definir el dominio de un atributo
 - d) Almacenar datos duplicados en una tabla
14. ¿Qué tipo de clave puede tener valores nulos en el modelo relacional?
- a) Clave primaria
 - b) Clave candidata
 - c) Clave ajena
 - d) Clave compuesta

15. Una relación con tipo de correspondencia 1:N en el diagrama E/R, ¿cómo se implementa en el modelo relacional?:
- a) Creando una tabla intermedia
 - b) Mediante un índice
 - c) Propagando la clave primaria de la tabla "uno" a la tabla "varios" como clave ajena**
 - d) Usando solo una clave candidata
16. ¿Qué es el proceso de normalización en bases de datos relacionales?
- a) Descomponer una tabla en varias subtablas para reducir redundancias.**
 - b) Combinar varias tablas en una sola.
 - c) Aumentar el rendimiento de las consultas.
 - d) Crear claves ajenas en todas las tablas.
17. ¿Cuándo se dice que una tabla está en primera forma normal (1FN)?
- a) Cuando sus atributos tienen dependencias parciales.
 - b) Cuando no hay grupos repetitivos.**
 - c) Cuando la clave primaria está compuesta.
 - d) Cuando los valores de la clave primaria no son nulos.
18. ¿Qué garantiza la integridad referencial entre dos tablas relacionadas, como Empleado y Departamento, donde Empleado tiene una clave ajena Cod_Dep que referencia la clave primaria Cod_Dep en Departamento?
- a) Que Cod_Dep en Empleado puede tener cualquier valor.
 - b) Que todos los valores de Cod_Dep en Empleado existen en Departamento o son nulos.**
 - c) Que Empleado puede tener valores duplicados en Cod_Dep.
 - d) Que todos los empleados deben estar asignados a un departamento.
19. ¿Qué significa desnormalizar una base de datos?
- a) Incrementar las dependencias funcionales para mejorar la seguridad.
 - b) Modificar tablas para mejorar el rendimiento, aunque aumente la redundancia.**
 - c) Pasar de un modelo físico a uno conceptual.
 - d) Eliminar las claves ajenas para simplificar el diseño.
20. ¿Cuál es un ejemplo de clave candidata en una tabla de empleados?
- a) Nombre
 - b) DNI o número de identificación**
 - c) Dirección
 - d) Fecha de nacimiento
21. ¿Qué garantiza la integridad de entidad en una base de datos relacional?
- a) Que no haya valores duplicados en las claves ajenas.
 - b) Que ningún valor de la clave primaria sea nulo.**
 - c) Que las tuplas en una tabla estén ordenadas.
 - d) Que las relaciones sean de tipo uno a uno.

22. ¿Cuál es el propósito de los índices en una base de datos relacional?
- a) Almacenar copias de todas las tablas.
 - b) **Mejorar la recuperación rápida de filas en una consulta.**
 - c) Garantizar que las claves ajenas coincidan con las primarias.
 - d) Aumentar el tamaño de las tablas.
23. ¿Qué nivel de la arquitectura de bases de datos describe la estructura física de la base de datos?
- a) Nivel externo.
 - b) Nivel lógico.
 - c) **Nivel interno.**
 - d) Nivel conceptual.
24. ¿Cuál de las siguientes NO es una función de un Sistema Gestor de Bases de Datos (SGBD)?
- a) Garantizar la integridad de los datos.
 - b) Ofrecer un catálogo o diccionario de datos.
 - c) **Proveer procesamiento de datos en tiempo real.**
 - d) Gestionar transacciones y concurrencia.
25. ¿Cuál es una ventaja principal de las bases de datos centralizadas?
- a) Permiten el acceso sin conexión a internet.
 - b) **Reducen la redundancia y mejoran la consistencia de los datos.**
 - c) Almacenan datos solo en formato binario.
 - d) Son exclusivas para sistemas de grandes organizaciones.

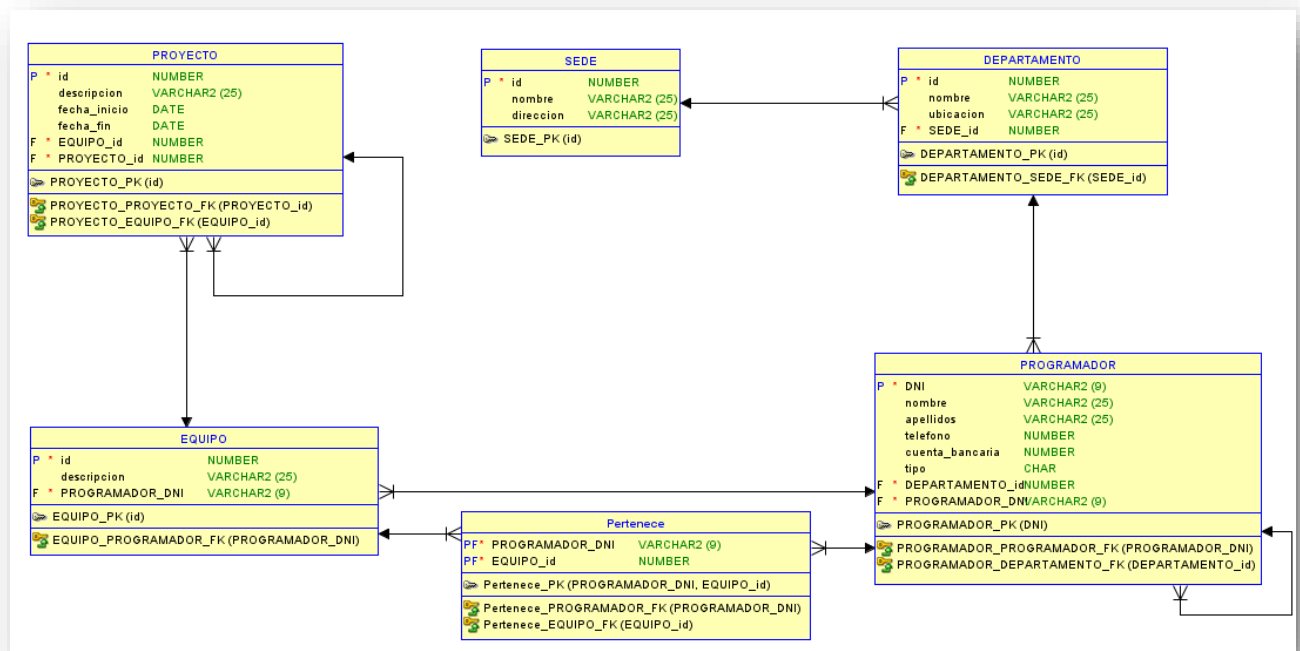
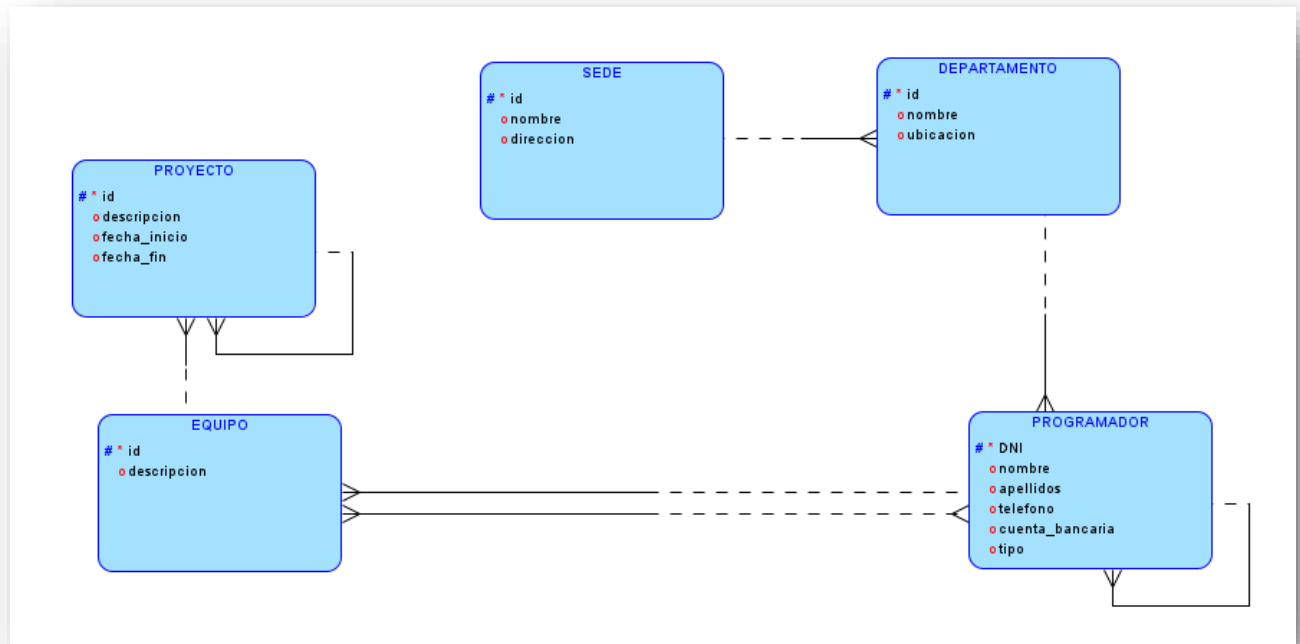
PRÁCTICA (5 PUNTOS)

(1,5 puntos) Pregunta 1. Realizar en SQL Developer Data Modeler (diagrama lógico y relacional) el siguiente supuesto:

"Equipo de programadores"

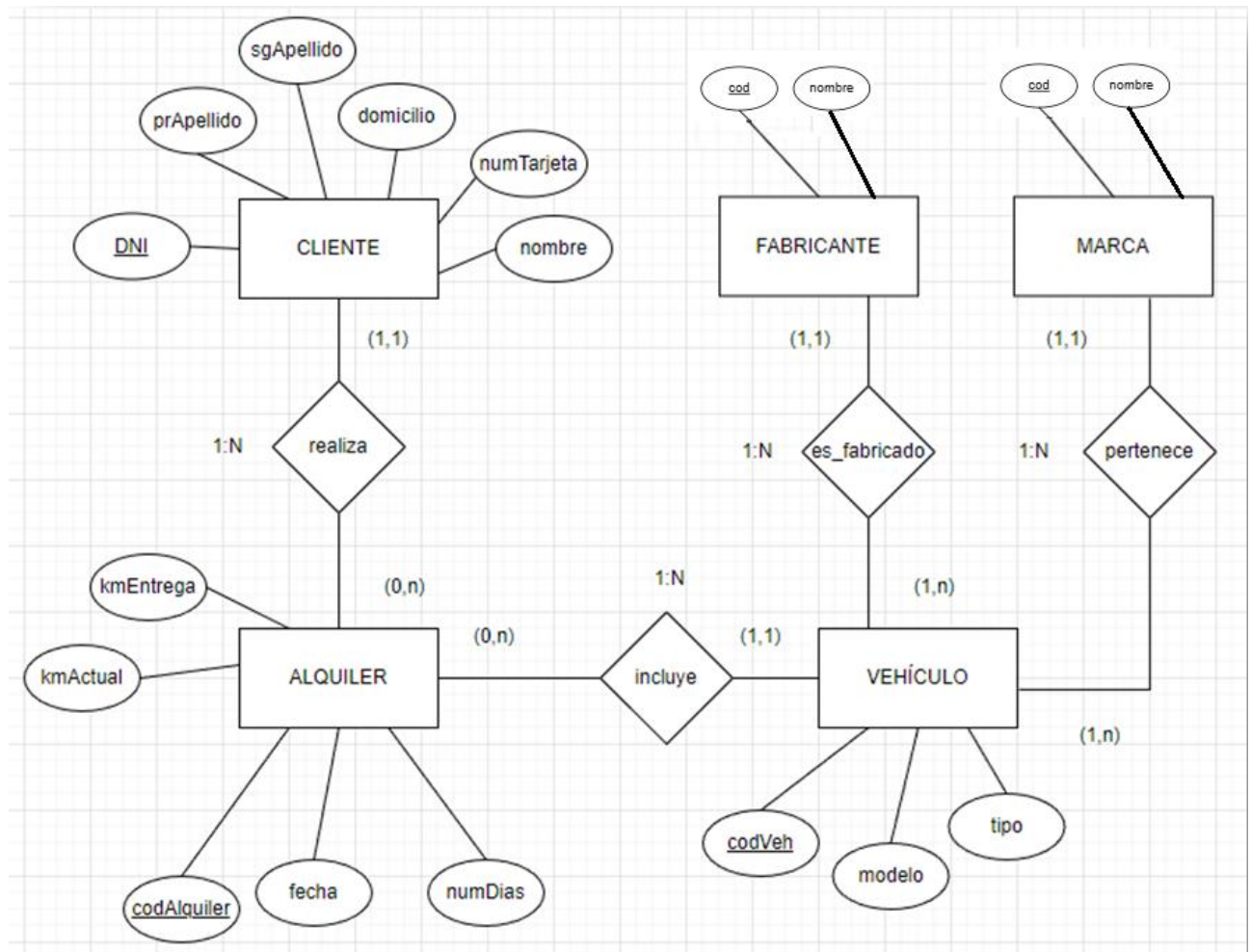
Supónganse los siguientes requisitos de información para una empresa que requiere implantar una base de datos para organizar los proyectos en los que sus programadores intervienen organizándose a través de equipos de desarrollo.

- R1. La empresa tiene diferentes sedes repartidas por el país identificadas por un número y se quiere almacenar el nombre de la sede y su dirección.
- R2. En cada sede, existen diferentes departamentos, de los que se quiere almacenar el nombre y la ubicación en el edificio.
- R3. Los programadores pertenecen a un único departamento y se necesita almacenar sus datos personales (dni, nombre, apellidos) y teléfono, su cuenta bancaria y el tipo de programador (A1, A2, S1, S2, M1, M2 y D).
- R4. Los programadores experimentados son mentores de otros programadores hasta que dejan de serlo.
- R5. Los programadores se organizan en equipos de desarrollo y pueden pertenecer a tantos como se requiera.
- R6. Los equipos tienen un código y una descripción, y se necesita almacenar el programador que es jefe de dicho equipo. Un programador puede ser jefe de más de un equipo.
- R7. A los equipos de les asignan diferentes proyectos, pero un proyecto solo es desarrollado por un único equipo de desarrollo.
- R8. Los proyectos tienen un código único, una descripción y una fecha de inicio y de fin. Los proyectos pueden estar compuestos por proyectos más pequeños llamados subproyectos, pero tienen las mismas propiedades que los proyectos (descripción, fecha de inicio y fecha de finalización).





(1,5 puntos) Pregunta 2. Obtener el script DDL (incluyendo comentarios) para el siguiente diagrama E/R.



```
--*****  
--  
--  
--  
--*****
```

```
DROP TABLE Cliente CASCADE CONSTRAINTS;  
DROP TABLE Fabricante CASCADE CONSTRAINTS;  
DROP TABLE Marca CASCADE CONSTRAINTS;  
DROP TABLE Vehiculo CASCADE CONSTRAINTS;  
DROP TABLE Alquiler CASCADE CONSTRAINTS;
```

```
CREATE TABLE Cliente(  
dni NUMBER(8),  
Nombre VARCHAR2(20) NOT NULL,  
prApellido VARCHAR2(20) NOT NULL,  
sgApellido VARCHAR2(20) NOT NULL,  
Domicilio VARCHAR2(50) NOT NULL,  
numTarjeta NUMBER(20) NOT NULL,  
CONSTRAINT pk_cliente PRIMARY KEY(dni));
```

```
CREATE TABLE Fabricante(  

```

```
cod NUMBER(3),
nombre VARCHAR2(30),
CONSTRAINT pk_fabricante PRIMARY KEY(cod));

CREATE TABLE Marca(
cod NUMBER(3),
nombre VARCHAR2(30),
CONSTRAINT pk_marca PRIMARY KEY(cod));

CREATE TABLE Vehiculo(
CodVeh NUMBER(4),
Modelo VARCHAR2(30) NOT NULL,
Tipo CHAR(1),
codFabricante NUMBER(3) NOT NULL,
codMarca NUMBER(3) NOT NULL,
CONSTRAINT pk_vehiculo PRIMARY KEY(codVeh),
CONSTRAINT fk_vehiculo_fab FOREIGN KEY(Fabricante) REFERENCES
Fabricante(Cod),
CONSTRAINT fk_vehicul_marca FOREIGN KEY(Marca) REFERENCES Marca(Cod));

CREATE TABLE Alquiler(
codAlquiler NUMBER(3),
Fecha DATE NOT NULL,
NumDias NUMBER(3) NOT NULL,
KmActual NUMBER(6) NOT NULL,
KmEntrega NUMBER(6) NOT NULL,
Dni NUMBER(8) NOT NULL,
CodVeh NUMBER(4) NOT NULL,
CONSTRAINT pk_Alquiler PRIMARY KEY(codAlquiler),
CONSTRAINT fk_alquiler_cliente FOREIGN KEY(dni) REFERENCES Cliente(dni),
CONSTRAINT fk_alquiler_vehiculo FOREIGN KEY(codVeh) REFERENCES
Vehiculo(codVeh));
```


(1 punto) Pregunta 3. Partiendo del script OFICINA, encontrar las sentencias SQL para:

a) Mostrar las ciudades en las que existen oficinas, cuando el objetivo se encuentre entre 400000 y 500000.

```
SELECT DISTINCT CIUDAD  
FROM OFICINA  
WHERE OBJETIVO BETWEEN 400000 AND 500000;
```

b) Mostrar el valor mínimo y máximo de ventas cuando el oficio sea "Marketing" y el número de oficina sea superior a 20.

```
SELECT MIN(VENTAS), MAX(VENTAS)  
FROM VENDEDOR  
WHERE OFICIO = 'Marketing' AND NUMOFI>20;
```

c) Mostrar el número de vendedores por oficina, cuando dicho número sea superior a 1. Utiliza ALIAS de columna.

```
SELECT NUMOFI, COUNT(*) AS NÚMERO  
FROM VENDEDOR  
GROUP BY NUMOFI  
HAVING COUNT(*)>1;
```

d) Mostrar la media de ventas por oficio, ordenado por dicha media en orden descendente.

```
SELECT OFICIO, AVG(VENTAS)  
FROM VENDEDOR  
GROUP BY OFICIO  
ORDER BY AVG(VENTAS) DESC;
```

e) Mostrar los nombres de los vendedores jefe.

```
SELECT NOMBRE  
FROM VENDEDOR  
WHERE JEFE IS NULL;
```

f) Mostrar el número de oficina, ciudad, nombre de empleado y ventas, cuando el nombre de la ciudad comience por el carácter 'M'. La información debe aparecer ordenada por CIUDAD

```
SELECT NUMOFI, CIUDAD, NOMBRE, VENTAS  
FROM OFICINA JOIN VENDEDOR USING (NUMOFI)  
WHERE CIUDAD LIKE 'M%'  
ORDER BY CIUDAD;
```

g) Mostrar el nombre y número de años de antigüedad de los empleados que no son jefes. Utilizar un alias para el número de años.

```
SELECT NOMBRE, ROUND((SYSDATE-FINGRE)/365,0) AS "NUMERO DE AÑOS"  
FROM VENDEDOR JOIN OFICINA USING (NUMOFI)  
WHERE JEFE IS NOT NULL;
```

h) Mostrar el número de vendedores por ciudad. La información debe aparecer ordenada por CIUDAD

```
SELECT CIUDAD, COUNT(*)  
FROM OFICINA JOIN VENDEDOR USING (NUMOFI)  
GROUP BY CIUDAD  
ORDER BY CIUDAD;
```

i) Mostrar el nombre del jefe junto al nombre de cada vendedor.

```
SELECT V1.NOMBRE AS JEFE, V2.NOMBRE AS VENDEDOR  
FROM VENDEDOR V1 JOIN VENDEDOR V2 ON V1.NUMVEN=V2.JEFE  
ORDER BY V1.NOMBRE, V2.NOMBRE;
```

j) Mostrar por ciudad y objetivo, la suma de ventas, siempre que el objetivo sea superior al doble de la suma de ventas.

```
SELECT CIUDAD, OBJETIVO, SUM(VENTAS)  
FROM OFICINA JOIN VENDEDOR USING (NUMOFI)  
GROUP BY CIUDAD, OBJETIVO  
HAVING OBJETIVO>SUM(VENTAS)*2  
ORDER BY CIUDAD;
```

(1 punto) Pregunta 4. Encuentra el error (si existe) en las siguientes sentencias SQL:

```
1 --
DROP TABLE TUTOR CASCADE CONSTRAINTS;
CREATE TABLE TUTOR (
    DNI VARCHAR2(9),
    NOMBRE VARCHAR2(20) NOT NULL,
    DIRECCION VARCHAR2(20)
    SALARIO NUMBER (4) NOT NULL,
    POBLACION VARCHAR2(30),
    CONSTRAINT PK_TUTOR PRIMARY KEY(DNI));

DROP TABLE CURSO CASCADE CONSTRAINTS;
CREATE TABLE CURSO (
    CODIGO VARCHAR2(4),
    NOMBRE VARCHAR2(15) NOT NULL,
    HORAS NUMBER (3),
    DNI VARCHAR2(9),
    CONSTRAINT PK_CURSO PRIMARY KEY(CODIGO),
    CONSTRAINT FK_CURSO_TUTOR FOREIGN KEY(DNI) REFERENCES TUTOR(DNI));
```

Hay que incluir una coma al final del atributo Dirección de la tabla TUTOR.

```
2 --
INSERT INTO TUTOR VALUES ('123456789', 'PEDRO', 1000);
```

Es preciso indicar los campos que vamos a dar valor en el insert.

```
3 --
UPDATE TUTOR
SET DIRECCION='C/ PRINCIPAL, 2' POBLACION='VALLADOLID'
WHERE DNI='123456789';
```

Es necesario separar los campos a actualizar con una coma.

```
4 --
DELETE TUTOR
WHERE SALARIO >=1300 AND (DNI='987654321' OR DIRECCION='c/ Primera, 1');
```

No hay error.

```
5 --
SELECT DNI, COUNT(*)
FROM CURSO;
```

Falta el GROUP BY DNI

```
6 --
SELECT NOMBRE, NOMBRE
FROM TUTOR JOIN CURSO USING (DNI);
```

Hay que utilizar alias de columna con AS



```
7 --  
SELECT AGV(HORAS)  
FROM CURSO;
```

Hay que utilizar AVG

```
8 --  
SELECT SALARIO, COUNT(*)  
FROM CURSO  
GROUP BY DNI  
HAVING COUNT(*)>2;
```

La tabla CURSO no tiene SALARIO. Si se agrupa por DNI, hay que poner DNI en la SELECT

```
9 --  
SELECT *  
FROM CURSO  
WHERE HORAS IN 2,4;
```

Faltan los paréntesis del IN

```
10 --  
SELECT COUNT(*)  
FROM TUTOR  
WHERE SALARIO BETWEEN (1000 AND 2000);
```

Sobran los paréntesis del BETWEEN
