

[ejercicios](#) [consultas](#) [resueltos](#) [exam...](#)



JosePalomino



Fundamentos de Bases de Datos



2º Grado en Ingeniería Informática

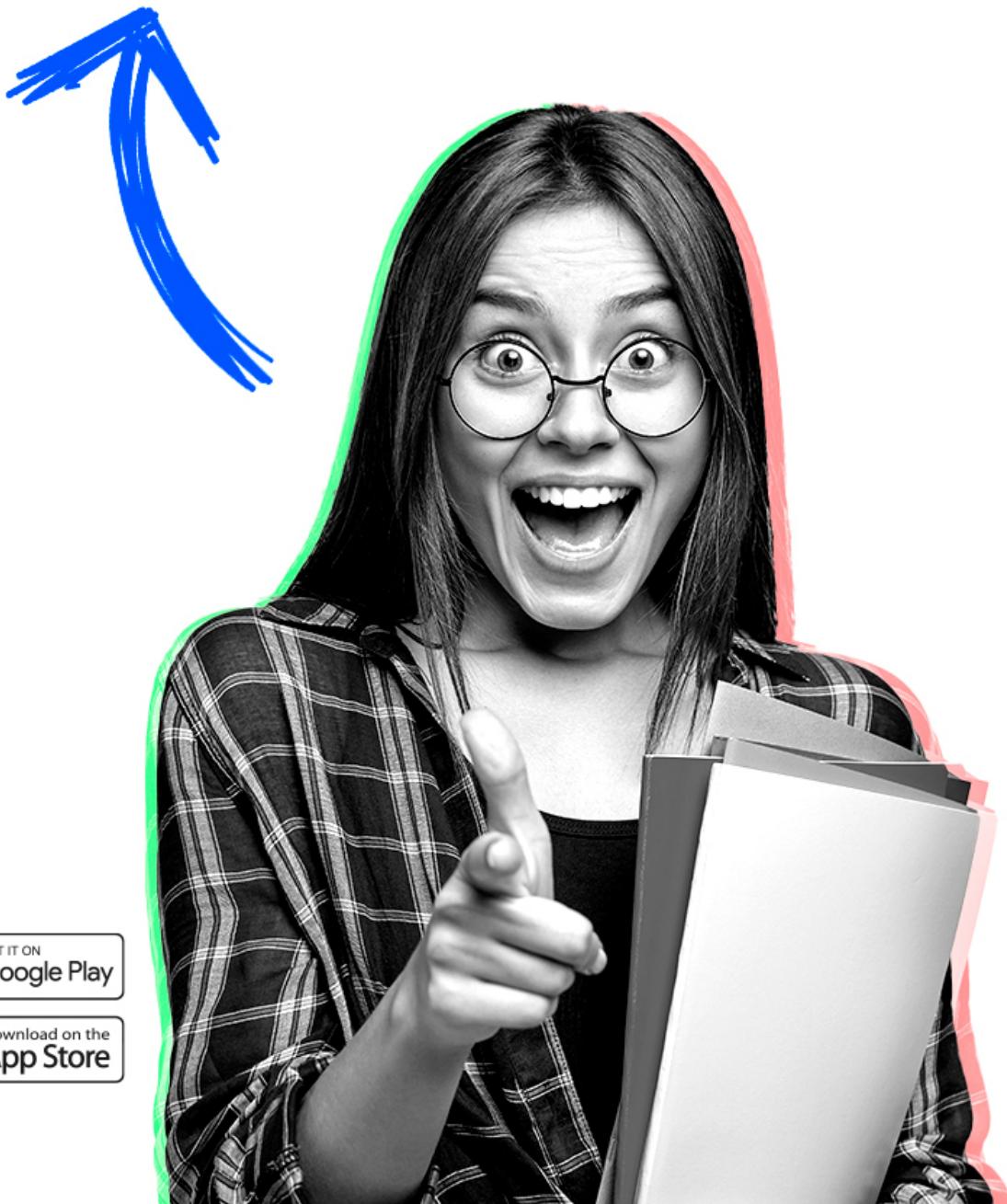


**Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada**

Estudiar sin publi es posible.



Compra Wuolah Coins y que nada te distraiga durante el estudio



Estudiar sin publi es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



BASE DE DATOS QUE GESTIONA LOS INVITADOS DE UNA BODA



- Además de las restricciones de integridad especificadas en el dibujo, deben considerarse las siguientes:
- El nombre, la dirección y la ciudad de los invitados son cadenas de caracteres.
- Siempre debe registrarse la dirección de los invitados.
- La ciudad de procedencia de los invitados está restringida a los valores: Málaga, Granada, Jaén y Almería, siendo Granada el valor por defecto.
- El número de personas asistentes por invitado es, por defecto, 1.
- La fecha de reserva de los regalos debe estar entre el 22 de Mayo y el 22 de Junio de 2017.

Número de personas máxima y número mínimo de los invitados

```
SELECT MIN(N_PERSONAS), MAX(N_PERSONAS) FROM INVITADOS
```

Precio medio de los regalos

```
SELECT AVG(Precio) FROM Lista_Boda
```

Descripción del regalo más caro

```
SELECT Descripcion FROM Lista_Boda WHERE Precio=MAX(Precio)
```

Precio de los regalos que han reservado los invitados de Jaén

```
SELECT Precio FROM Lista_Boda, Regalo_Reservado WHERE  
    Lista_Boda.ref=Regalo_Reservado.ref AND  
    Regalo_Reservado.Nombre=Invitados.Nombre AND  
    Ciudad='Jaen'
```

Descripción y precio del regalo que ha reservado el invitado con más personas

```
SELECT Descripcion, Precio FROM Lista_Boda,Regalo_Reservado, Invitados WHERE  
    Lista_Boda.ref=Regalo_Reservado.ref AND  
    Regalo_Reservado.Nombre=Invitados.Nombre AND  
    N_Personas IN (N_personas) FROM ---incompleto, falta la tabla
```

Referencias de regalos de lista de bod que no han sido reservados

opcion 1:

```
SELECT Lista_Boda.Descripcion FROM Lista_Boda WHERE  
    NOT EXISTS(  
        SELECT Regalo_Reservado.Ref FROM Regalo_Reservado WHERE  
            Regalo_Reservado.Ref = Lista_Boda.Ref  
    )
```

opcion2:

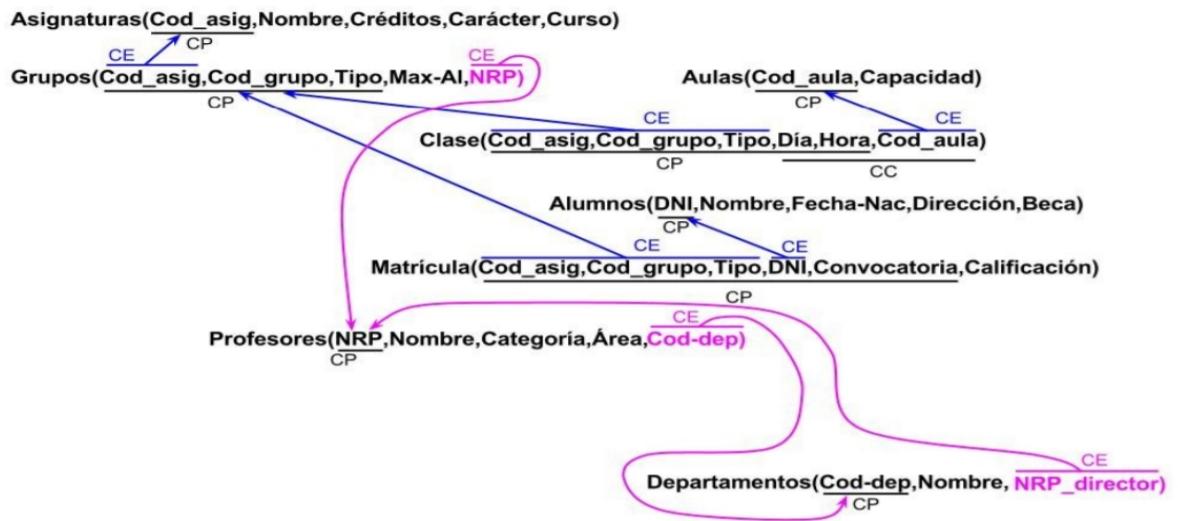
```
SELECT Lista_Boda.Descripcion FROM Lista_Boda WHERE  
    Lista_Boda.Ref NOT IN (  
        SELECT Regalo_Reservado.ref FROM Regalo_Reservado  
    )
```



WUOLAH

Precio más pequeño del regalo que no ha sido reservado

```
SELECT MIN(precio) FROM Lista_Boda WHERE
    NOT EXISTS(
        SELECT Regalo_Reservado.Ref FROM Regalo_Reservado WHERE
            Regalo_Reservado.Ref = Lista_Boda.Ref
    )
```



Número de registro personal de los profesores que imparten la asignatura FBD

```
SELECT DISTINCT NRP FROM Asignaturas, Grupos WHERE
```

```
    Asignaturas.Cod_Asig=Grupos.Cod_Asig AND
    Nombre='FBD'
```

Calificación mínima que tiene un alumno en FBD impartida por 'Perico'

```
SELECT MIN(Caliifcacion) FROM Matricula, Grupos, Asignaturas, Profesor WHERE
```

```
    Asignatura.Cod_Asig=Grupos.Cod_Asig AND
    Grupos.Cod_Asig=Matricula.Cod_Asig AND
    Grupos.Cod_Asig=MAtricula.Cod_grupo AND
    Grupos.Tipo= Matricula.Tipo AND Grupos.NRP=Profesores.NRP AND
    Asignaturas.Nombre='FBD' AND
    Profesores.Nombre='Perico'
```

Alumnos matriculados en el curso 18/19 que hayan sacado esa nota

```
SELECT Alumnos.nombre FROM Alumnos , Matricula , Grupos , Asignaturas WHERE
```

```
    Alumno.DNI=Matricula.DNI AND
    Asignatura.Cod_Asig=Grupos.Cod_Asig AND
    Grupos.Cod_Asig=Matricula.Cod_Asig AND
    Grupos.Cod_Asig=MAtricula.Cod_grupo AND
    Grupos.Tipo= Matricula.Tipo AND Grupos.NRP=Profesores.NRP AND
    Asignaturas.Nombre='FBD' AND
    Convocatoria='18/19' AND Calificacion IN(
```

```
        SELECT MIN(Caliifcacion) FROM Matricula, Grupos, Asignaturas, Profesor WHERE
```

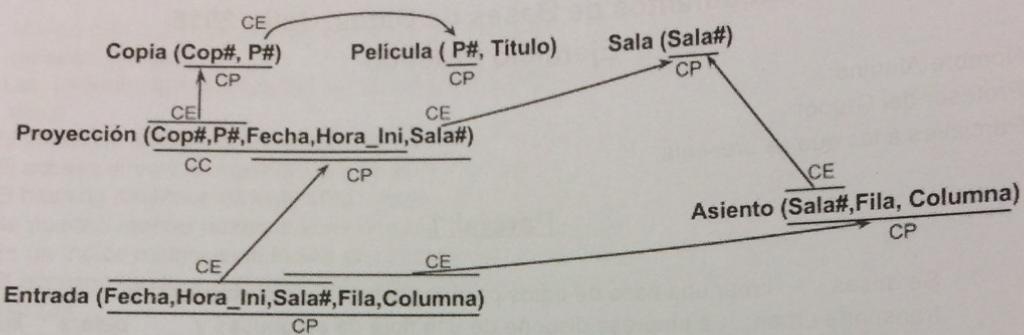
Asignatura.Cod_Asig=Grupos.Cod_Asig AND
Grupos.Cod_Asig=Matricula.Cod_Asig AND
Grupos.Cod_Asig=MAtricula.Cod_grupo AND
Grupos.Tipo= Matricula.Tipo AND Grupos.NRP=Profesores.NRP AND
Asignaturas.Nombre='FBD' AND
Profesores.Nombre='Perico'
)

Alumnos matriculados en ALG del curso 18/19 que tienen beca

```
SELECT DISTINCT Nombre FROM Alumnos, Matricula, Grupos, Asignatura WHERE  
Asignatura.Cod_asig=Grupos.Cod_Asig AND  
Matricula.Cod_asig=Grupos.Cod_Asig AND  
Matricula.Cod_grupo=Grupos.Cod_grupo AND  
Matricula.Tipo=Grupos.Tipo AND  
Matricula.DNI=Alumnos.DNI AND  
Asignaturas.Nombre='ALG'  
AND Convocatoria='18/19'  
AND Beca='SI'  
ORDER BY Nombre
```

Parcial 2

Disponemos de la siguiente BD que gestiona la actividad de un cine, donde la tabla Entrada registra las entradas que se han vendido para cada proyección.



Se pide:

- Escribe las instrucciones en SQL para la creación de la tabla Proyección y la inserción de una tupla en dicha tabla. Para la creación de la tabla se presuponen creadas el resto de tablas. Además de las restricciones de integridad especificadas en el esquema, deben considerarse las siguientes:
 - Fecha es de tipo date, Hora_ini es de tipo numérico, se almacena en formato 24h, o sea de 0 a 23h y sólo se permiten valores enteros desde las 10h a las 01h de la madrugada, el resto de los campos son de tipo alfanumérico con un máximo de 10 caracteres. (1.25 pts.).
- Realiza las siguientes consultas::
 - "Muestra proyecciones para las que no se han vendido ninguna entrada".(en AR 1.25 pts. y CRT 1.25 pts.).
 - "Muestra el título de la primera película que se proyectó en ese cine". (en AR 1.25pts. y SQL 1.25 pts.)
 - "Muestra el título de las películas que se han proyectado en todas las salas". (en CRT 1.25 pts. y SQL 1.25 pts.).
- Crea una vista que muestre la cantidad de entradas vendidas por cada proyección para aquellas proyecciones para las que se hayan vendido más de 40 entradas. (1.25 pt.).

Tiempo de realización: Un Parcial 1.5h, dos parciales 3h

Estudiar sin publi es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



A

```
CREATE TABLE Proyeccion(
    Cop VARCHAR(10) NOT NULL, //se le pone not null por que son claves candidatas
    P   VARCHAR(10) NOT NUL,
    FECHA DATE      NOT NULL,
    HORA_INI INT      CHECK(HORA_INI BETWEEN 0 AND 1
                           OR HORA_INI BETWEEN 10 AND 23),
    SALA VARCHAR(10) BETWEEN SALA.C.SALA,
    FOREIGN KEY(COP, P) REFERENCES COPIA(COP, P),
    UNIQUE(COP ,P, FECHA, HORA_INI),
    PRIMARY KEY (FECHA, HORA_INI, C.SALA)
);
INSERT INTO PROYECCION VALUES('1', '2', SYSDATE, 12,'J');
```

Ba(pero en SQL):

```
SELECT * FROM Proyeccion WHERE NOT EXIST(
    SELECT * FOM Entrada WHERE Entrado.Fecha=Proyeccion.Fecha AND
                                Entrado.Hora_Ini=Proyeccion. Hora_Ini AND
                                Entrado.C_Sala=Proyeccion. C_Sala
)
```

Bb(pero en SQL):

```
SELECT Titulo FROM Pelicula WHERE
    Pelicula.P=Copia.P AND
    Copia.Cop=Proyeccion.Cop AND Copia.P=Proyeccion.P AND
    Proyeccion.Fecha IN (
        SELECT MIN(P2.Fecha) FROM Proyeccion P2
    )
```



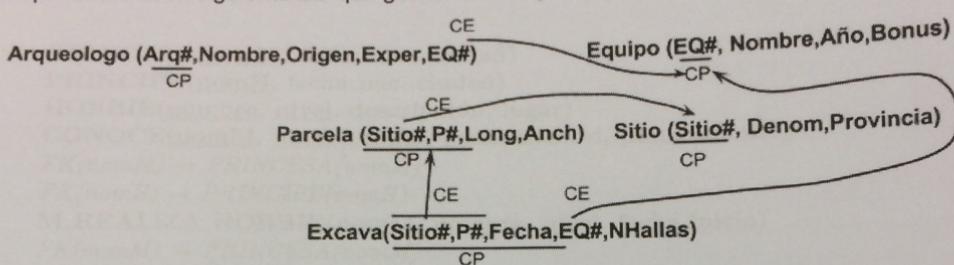
Fundamentos de Bases de Datos. Grupo C

Parcial 2. Ejercicio Práctico

Nombre Alumno:

Profesor del Grupo:

Disponemos de la siguiente BD que gestiona unos yacimientos arqueológicos:



Se pide:

- Escribe las instrucciones en SQL para la creación de la tabla **Excava** y la inserción de una tupla en dicha tabla. Para la creación de la tabla se presuponen creadas el resto de tablas. Además de las restricciones de integridad especificadas en el esquema, deben considerarse las siguientes:
 - En la tabla **Excava**, **Nhallas** es un número que cuenta los hallazgos encontrados en esa intervención, el contador comienza en 0 y siempre será positivo.
 - El resto de los campos son de tipo alfanumérico con un máximo de 10 caracteres, salvo **Fecha** que es de tipo **date**.
 - No se puede dejar incompleto ningún campo. **(1.5 pts.)**.
- Realiza las siguientes consultas::
 - "Muestra los sitios en los que queda alguna parcela sin excavar". (en **AR 1.25pts.** y **CRT 1.5pts.**).
 - "Muestra el nombre del equipo que hizo la excavación más antigua" (en **AR 1.5pts.**).
 - "Muestra la denominación de los sitios donde hayan excavado todos los equipos" (en **CRT 1.25pts.** y **SQL 1.5 pts.**)
- Escribe la instrucción en SQL para incrementar en 5 el bonus de los equipos con más de 30 hallazgos. **(1.5 pt.)**.

Tiempo total de realización (Teoría y práctico): 2h.

A

```
CREATE TABLE EXCAVA(
    SITIO VARCHAR(10),
    P      VARCHAR(10),
    FECHA DATE,
    EQ    VARCHAR(10) NOT NULL REFERENCES EQUIPO(EQ),
    NHALLAS INT DEFAULT 0 NOT NULL CHECK (NHALLAS>0)
    FOREIGN KEY (SITIO, P) REFERENCES PARCELA(SITIO,P),
    PRIMARY KEY (SITIO,P, FECHA)
);
INSERT INTO EXCAVA VALUES ('S1', 'P1', TO_DATE('1/1/2001'), 'F1', 3)
```

Ba(pero en SQL):

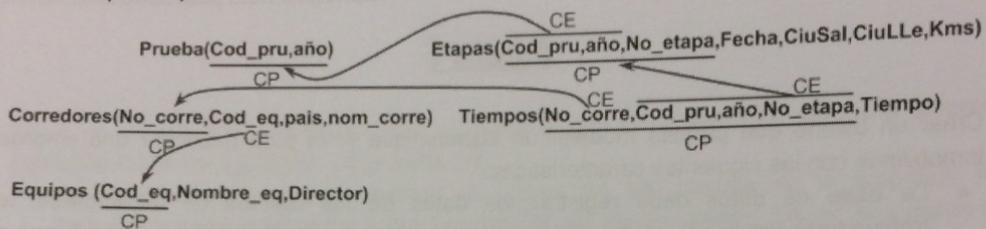
```
SELECT C_SITIO FROM Parcela WHERE NOT EXISTS(
    SELECT * FROM Excava WHERE Excava.c_sitio=Parcela.c_sitio AND Excava.P=Parcela.P
)
```

Bb(pero en SQL):

```
SELECT Nombre FROM Equipo WHERE Equipo.eq IN (
    SELECT eq FROM Excava WHERE Fecha <= ALL (
        SELECT MIN(Fecha) FROM Excava E2
    )
)
```

Parcial 2

Disponemos de la siguiente BD que gestiona la información relativa a competiciones ciclísticas por etapas.



Se pide:

- Escribe las instrucciones en SQL para la creación de la tabla Tiempos y la inserción de una tupla en dicha tabla, suponiendo que el campo Tiempo es de tipo numérico entero positivo y que almacena los segundos invertidos por un corredor en una etapa. Para la creación de la tabla se presuponen creadas el resto de tablas. Además de las restricciones de integridad especificadas en el esquema, deben considerarse las siguientes:
 - Los campos No_corre y No_etapa son de tipo entero positivo mayores que 0 y con un valor máximo de 9999 para el primero y de 30 para el segundo. El campo Tiempo siempre debe contener un valor. El campo año toma un valor entero positivo de cuatro dígitos mayor o igual que 1900 y el campo Cod_pru toma valores alfanuméricicos con un máximo de 30 caracteres. (1.25 pts.).
- Realiza las siguientes consultas:
 - "Muestra la etapa más corta".(en AR 1.25 pts. y CRT 1.25 pts.).
 - "Muestra el nombre de los corredores que han realizado todas las etapas de la prueba 'Giro de Italia' del 2015". (en AR 1.25pts. y SQL 1.25 pts.)
 - "Muestra los equipos cuyos corredores son todos del mismo país". (en CRT 1.25 pts. y SQL 1.25 pts.).
- Crea una vista que muestre la clasificación general individual del 'Tour de Francia' del 2014 (nombre del corredor y tiempo total invertido, ordenada de forma creciente con respecto al tiempo total empleado). (1.25 pt.).

Tiempo de realización: **Un Parcial 1.5h, dos parciales 3h.**

Estudiar sin publi es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



Ba(pero en SOL):

```
SELECT * FROM Etapas WHERE kms <= ALL(  
    SELECT E2.kms FROM Etapas E2  
)
```

Bb(pero en SQL):

```
SELECT No_Corre FROM Corredores WHERE NOT EXISTS(  
    SELECT * FROM Corredores C2 WHERE C2.Cod_eq=Corredores.Cod_eq AND  
    Corredores .Pais != C2.Pais  
)
```

