

Examen de prueba.pdf



MeryThunder



Bases de Datos



2º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**

BBVA**1/6**

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Ábrete la Cuenta Online de BBVA y llévate 1 año de **Wuolah PRO**

Ventajas Cuenta Online de BBVA

0€

Sin comisión de administración o mantenimiento de cuenta.
(0 % TIN 0 % TAE)

0€

Sin comisión por emisión y mantenimiento de Tarjeta Aqua débito.

0

Sin necesidad de domiciliar nómina o recibos.

Las ventajas de **WUOLAH PRO**



Di adiós a la publi en los apuntes y en la web



Descarga carpetas completas de un tirón



Acumula tickets para los sorteos

cómo??





1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

ventajas

PRO



Di adiós a la publi
en los apuntes y
en la web



Acumula tickets
para los sorteos



Descarga
carpetas
completas

estudia sin publi

WUOLAH PRO

Examen de prueba

Bases de Datos (Curso 2018/19)

--Pregunta 1: Para cada profesor (deben aparecer todos) mostrar id de profesor y número de tramos de investigación que tiene. El número de tramos se encuentra en la tabla INVESTIGADORES. En caso de no tener tramos debe aparecer 0

```
select p.id "PROFESOR", NVL(to_char(i.tramos), '0') "TRAMOS"
from profesores p
join investigadores i on i.id_profesor = p.id;
```

--Pregunta 2: Parejas de profesores que comparten teléfono. Si no tienen teléfono se considera que el teléfono es el mismo y deben emparejarse también. De cada profesor solo queremos el Apellido1. No muestre información duplicada.

```
select pa.apellido1, pb.apellido1, NVL(to_char(pa.telefono), 'No') "TELÉFONO"
from profesores pa, profesores pb
where NVL(to_char(pa.telefono), 'No') = NVL(to_char(pb.telefono), 'No') and pa.id < pb.id;
```

--Pregunta 3: Listar dni y fecha de nacimiento de los tres alumnos de mayor edad que están registrados en el sistema y no están matriculados en ninguna asignatura. Los alumnos sin fecha de nacimiento no se tendrán en cuenta

```
select * from
(select al.dni, al.fecha_nacimiento "FECHA"
from alumnos al
where not exists
(select * from matricular m, alumnos al
where al.dni = m.alumno and m.asignatura is null)
order by al.fecha_nacimiento)
where rownum <= 3;
```

--Pregunta 4: Nombre completo de los profesores que han impartido la asignatura que más alumnos ha tenido matriculados, pero teniendo en cuenta solamente aquellas asignaturas con menos de 25 alumnos matriculados. Si un alumno se ha matriculado varias veces en la asignatura debe contarse sólo una vez.

```
select distinct p.nombre, p.apellido1, p.apellido2
from profesores p
join impartir i on i.profesor = p.id
where exists
(select * from(
select distinct asignatura,count(alumno)
from matricular
group by asignatura
having count(alumno) < 25
order by count(alumno) desc)
where rownum = 1);
```

--Pregunta 5: Queremos obtener la distribución en género de cada asignatura en cada curso académico (el de la tabla matricular). Obtenga el nombre de la asignatura, el curso académico, el número de alumnos y el número de alumnas matriculados

```
select a.nombre, a.curso,
sum(decode(al.genero, 'MASC', 1, 0)) "ALUMNOS",
sum(decode(al.genero, 'FEM', 1, 0)) "ALUMNAS"
from asignaturas a
join matricular m on m.asignatura = a.codigo
join alumnos al on m.alumno = al.dni
group by a.nombre, a.curso;
```

--Pregunta 6: Queremos obtener cuántos alumnos se matricularon por primera vez (FECHA_PRIM_MATRICULA) en cada uno de los meses del año, pero solo de aquellos meses en los que haya más alumnos que la media

```
select
to_char(al.fecha_prim_matricula, 'MM') "Mes",
count(*) "Alumnos"
from alumnos al
group by to_char(al.fecha_prim_matricula, 'MM')
having count(*) > (select avg(count(distinct DNI)) from alumnos
group by to_char(fecha_prim_matricula, 'MM'))
order by to_char(al.fecha_prim_matricula, 'MM');
```