

FBD

Examen VII



Los Del DGIIM, losdeldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

FBD

Examen VII

Los Del DGIIM, losdeldgiim.github.io

Granada, 2024-2025

Asignatura Fundamentos de Bases de Datos.

Curso Académico 2024-25.

Grado Doble Grado en Ingeniería Informática y Matemáticas.

Grupo Único.

Profesor Nicolás Marín Ruiz.

Descripción Convocatoria Ordinaria. Parcial Práctico 2 (Seminarios 3-4).

Fecha 10 de enero de 2025.

Duración 90 minutos (durante los cuales también se tenía que hacer la parte de teoría, 20 preguntas tipo test).

Ante el siguiente esquema de la BD:

- Empresa(id_empresa, nombre_empresa, pais)
 - Donde “id_empresa” es la clave primaria.
- Proyecto(id_proyecto, id_empresa, titulo, fecha_creacion)
 - Donde “id_proyecto” es la clave primaria.
 - “id_empresa” es una clave externa que referencia a Empresa(id_empresa).
- Revisor(id_revisor, nombre_revisor, tarifa, categoria)
 - Donde “id_revisor” es la clave primaria.
- Revision(id_revisor, id_proyecto, fecha, puntuacion)
 - Donde “(id_revisor, id_proyecto, fecha)” es la clave primaria.
 - “id_revisor” es una clave externa que referencia a Revisor(id_revisor).
 - “id_proyecto” es una clave externa que referencia a Proyecto(id_proyecto).

SQL

Ejercicio 1. Actualiza la puntuación de las revisiones del revisor con id “1234”, sumándole 10 a cada puntuación realizada por él.

Ejercicio 2. Sobre la tabla **Proyecto**:

- a) Cree un índice de mapa de bits sobre el campo **titulo**.
- b) ¿Es una buena opción? Justifique su respuesta.
- c) Borre el índice.

Ejercicio 3. Mostrar para cada revisor la suma de las puntuaciones de las revisiones hechas en 2024 solo de aquellos proveedores que hayan hecho al menos 5 revisiones en 2023.

Ejercicio 4. Mostrar los revisores que han hecho al menos una revisión a cada uno de los proyectos de la empresa con id “EMP1”.

AR

Ejercicio 5. Haga una lista de **id_revisor**, **fecha** de las revisiones realizadas por los revisores de categoría “SENIOR”.

Ejercicio 6. Mostrar las empresas de los proyectos que han obtenido la mayor puntuación.

Ejercicio 7. Mostrar el id de los revisores cuyo conjunto de puntuaciones otorgadas coincide con el conjunto de puntuaciones que han sido otorgadas por el revisor de id “1234”.

SQL

Ejercicio 1. Actualiza la puntuación de las revisiones del revisor con id “1234”, sumándole 10 a cada puntuación realizada por él.

```
update revision set puntuacion = puntuacion + 10
where id_revisor = '1234';
```

Ejercicio 2. Sobre la tabla Proyecto:

- a) Cree un índice de mapa de bits sobre el campo `titulo`.

```
create bitmap index indice_bmp_titulo
on proyecto(titulo);
```

- b) ¿Es una buena opción? Justifique su respuesta.

No es una opción buena, ya que los índices de mapas de bits son útiles sobre atributos de relaciones cuyo dominio activo cuenta con pocos valores. En este caso, es probable que el atributo “titulo” de la tabla “Proyecto” contenga un título distinto por cada proyecto (aunque también existan proyectos que tengan el mismo título, aunque pocos), así como una gran variedad de títulos en el sistema.

Como contamos una entrada por cada valor y un bit por cada registro, obtendríamos una gran cantidad de enormes mapas de bits en los que cada uno contenga una gran cantidad de ceros.

- c) Borre el índice.

```
drop index indice_bmp_titulo;
```

Ejercicio 3. Mostrar para cada revisor la suma de las puntuaciones de las revisiones hechas en 2024 solo de aquellos proveedores que hayan hecho al menos 5 revisiones en 2023.

```
select id_revisor, sum(puntuacion)
from revision
where to_char(fecha, 'yyyy') = 2024
group by id_revisor
having id_revisor in (
    select id_revisor
    from revision
    where to_char(fecha, 'yyyy') = 2023
    group by id_revisor
    having count(*) >= 5
);
```

Ejercicio 4. Mostrar los revisores que han hecho al menos una revisión a cada uno de los proyectos de la empresa con id “EMP1”.

```
select distinct id_revisor
from revision r
where not exists (
  select * from proyecto p
  where id_empresa = 'EMP1' and not exists (
    select * from revision
    where id_revisor = r.id_revisor
    and id_proyecto = p.id_proyecto
  )
);
```

AR

Ejercicio 5. Haga una lista de `id_revisor`, `fecha` de las revisiones realizadas por los revisores de categoría “SENIOR”.

$$\pi_{id_revisor, fecha}(Revision \bowtie \sigma_{categoria='SENIOR'}(Revisor))$$

Ejercicio 6. Mostrar las empresas de los proyectos que han obtenido la mayor puntuación.

Primero calculamos los proyectos que han obtenido la mayor puntuación en una revisión:

$$R = \rho(R Revision)$$
$$\alpha = \rho(\pi_{id_proyecto}(R) - \pi_{R.id_proyecto}(R \bowtie_{R.puntuacion < Revision.puntuacion} Revision));$$

Finalmente, nos quedamos con las empresas de dichos proyectos:

$$\pi_{id_empresa}(\alpha \bowtie Proyecto)$$

Ejercicio 7. Mostrar el id de los revisores cuyo conjunto de puntuaciones otorgadas coincide con el conjunto de puntuaciones que han sido otorgadas por el revisor de id “1234”.

Conjunto de puntuaciones otorgadas por el revisor de id “1234”:

$$P = \rho(\pi_{puntuacion}(\sigma_{id_revisor='1234'}(Revision)))$$

Revisores que han otorgado puntuaciones que no están en P :

$$\alpha = \rho(\pi_{id_revisor}(Revision \bowtie (\pi_{puntuacion}(Revision) - P)))$$

Revisores que no han otorgado puntuaciones que no están en P (es decir, el conjunto de revisores que solo otorgan puntuaciones de P):

$$\beta = \rho(\pi_{id_revisor}(Revision) - \alpha)$$

Tenemos que asegurarnos de que dichos revisor otorguen **todas** las puntuaciones que hay en P , con lo que habrán otorgado exactamente las puntuaciones que hay en P :

$$\pi_{id_revisor, puntuacion}(\beta \bowtie Revision) \div P$$