

FBDORD2021PRACTICAS.pdf



danielsp10



Fundamentos de Bases de Datos



2º Grado en Ingeniería Informática



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



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



WUOLAH

Belén será lo que quiera.

#NoTeApuntesAWuolah

¿Tu sueño es estudiar Medicina
pero no sabes si podrás conseguirlo?
Pues deja de soñar y vive tus sueños.



LITHUANIAN UNIVERSITY
OF HEALTH SCIENCES

ACCEDER A MEDICINA EN LITUANIA
sin nota de corte y 100% en inglés



WEBINAR GRATUITO

Daniel Pérez Ruiz

PRÁCTICAS FBD [CONSULTAS SQL Y AR] - ORDINARIO 2021

Esquema de colores para interpretación de las tablas.

- **NOMBRE DE TABLA**
- **clave_primaria**
- **clave_externa**

MODELO 1. Considere el siguiente esquema:

```
ASEGURADORA(id_aseguradora, nombre_aseguradora, país);  
  
PACIENTE(id_paciente, nombre_paciente, id_aseguradora, fecha_nacimiento);  
id_aseguradora apunta a : ASEGURADORA.  
  
MEDICO(id_medico, nombre_medico, especialidad, sueldo);  
  
CONSULTA(id_paciente, id_medico, fecha, precio_facturado, numero_horas);  
id_paciente apunta a : PACIENTE.  
id_medico apunta a : MEDICO.
```

Cada paciente tiene asignada una aseguradora que cubre sus gastos médicos. La tabla CONSULTA almacena citas en las que un médico ha atendido a un paciente en una fecha dada.

SQL 1: Encontrar, considerando sólo los médicos que han realizado consultas en el 2014, el sueldo del médico que ha realizado menos consultas durante dicho año. Puede haber más de un médico que cumpla esta condición.

```
SELECT sueldo FROM medico NATURAL JOIN  
(SELECT id_medico FROM consulta WHERE TO_CHAR(fecha, 'YYYY')='2014'  
GROUP BY id_medico  
HAVING COUNT(*) <= (SELECT MIN(COUNT(*)) FROM consulta WHERE  
TO_CHAR(fecha, 'YYYY')='2014' GROUP BY id_medico));
```

SQL 2: Encontrar, entre los pacientes que han tenido consultas en el 2011, el id de la aseguradora de aquel paciente que ha gastado en total menos dinero sumando sus consultas en dicho año. Puede haber más de un paciente que cumpla esta condición.

```
SELECT id_aseguradora FROM paciente NATURAL JOIN  
(SELECT id_paciente FROM consulta WHERE TO_CHAR(fecha, 'YYYY')='2011'  
GROUP BY id_paciente  
HAVING SUM(precio_facturado) <= (SELECT MIN(SUM(precio_facturado)) FROM consulta  
WHERE TO_CHAR(fecha, 'YYYY')='2011' GROUP BY id_paciente));
```

EDUCOWAY
EUROPEAN EXCELLENCE EDUCATION



Escoge entre:
• MEDICINA · ODONTOLOGÍA
• ENFERMERÍA · VETERINARIA

Lunes, 14
de junio
de 2021,
a las 18h.

WUOLAH

SQL 3: Encontrar el nombre y sueldo de los médicos con especialidad VASCULAR que, para cada aseguradora que hay en la base de datos, han hecho al menos una consulta con un precio superior a 100 a un paciente de esa aseguradora.

```
SELECT nombre_medico,sueldo FROM medico WHERE especialidad='VASCULAR'
AND id_medico IN(
    SELECT DISTINCT id_medico FROM medico WHERE NOT EXISTS(
        SELECT id_aseguradora FROM aseguradora
        MINUS
        SELECT id_aseguradora FROM consulta NATURAL JOIN (SELECT
            id_paciente,id_aseguradora FROM paciente)
        WHERE precio_facturado > 100 AND consulta.id_medico=medico.id_medico
    )
);
```

SQL 4: Encontrar el id_paciente y el id_aseguradora de los pacientes que, para cada especialidad de los médicos que tienen un sueldo superior a 4000, han sido atendidos al menos una vez durante menos de una hora por un médico con esa especialidad (gane lo que gane ese médico).

```
SELECT id_paciente,id_aseguradora FROM paciente WHERE id_paciente IN(
    SELECT DISTINCT id_paciente FROM paciente WHERE NOT EXISTS(
        SELECT especialidad FROM medico WHERE sueldo > 4000
        MINUS
        SELECT especialidad FROM consulta NATURAL JOIN (SELECT
            id_medico,especialidad FROM medico)
        WHERE numero_horas < 1 AND consulta.id_paciente=paciente.id_paciente
    )
);
```

SQL 5: Encontrar, considerando sólo los médicos que han realizado consultas en el año 2017, el nombre del médico que ha facturado en total más dinero sumando sus consultas durante dicho año. Puede haber más de un médico que cumpla esta condición.

```
SELECT nombre_medico FROM medico NATURAL JOIN
(SELECT id_medico FROM consulta WHERE TO_CHAR(fecha,'YYYY')='2017'
GROUP BY id_medico
HAVING SUM(precio_facturado) >= (SELECT MAX(SUM(precio_facturado)) FROM
consulta WHERE TO_CHAR(fecha,'YYYY')='2017' GROUP BY id_medico));
```

ÁLGEBRA RELACIONAL 1: Elaborar un listado con el id de los pacientes atendidos en consulta para los que todas sus consultas son siempre del mismo médico.

ALIAS(consulta) = C1 ; **ALIAS**(consulta) = C2

PROYECCIÓN_{C1.id_paciente}(C1) -

PROYECCIÓN_{C1.id_paciente}(SELECCIÓN_{C1.id_paciente = C2.id_paciente ^
C1.id_medico != C2.id_medico}(C1 x C2))

ÁLGEBRA RELACIONAL 2: Elaborar un listado de países con una única aseguradora.

ALIAS(aseguradora) = A1 ; **ALIAS**(aseguradora) = A2

PROYECCIÓN_{A1.pais}(A1) - **PROYECCIÓN**_{A1.pais}(**SELECCIÓN**<sub>{A1.pais = A2.pais
^ C1.id_aseguradora != C2.id_aseguradora}</sub>(A1 x A2))

/******

MODELO 2. Considere el siguiente esquema:

```
EMPRESA(id_empresa, nombre_empresa, país);

PROYECTO(id_proyecto, titulo, id_empresa, fecha_elaboracion);
    id_empresa apunta a : EMPRESA.

REVISOR(id_revisor, nombre_revisor, categoria, tarifa);

REVISION(id_revisor, id_proyecto, fecha, puntuacion);
    id_revisor apunta a : REVISOR.
    id_proyecto apunta a : PROYECTO.
```

Cada proyecto tiene asignada la empresa que lo elabora. La tabla REVISION registra revisiones indicando que revisor revisa qué proyecto en qué fecha.

SQL 1: Encontrar, considerando sólo los revisores que han hecho revisiones en el 2012, el nombre del revisor que ha otorgado mayor total de puntos sumando todas sus revisiones de ese año. Puede haber más de un revisor que cumpla esta condición.

```
SELECT nombre_revisor FROM revisor NATURAL JOIN
(SELECT id_revisor FROM revision WHERE TO_CHAR(fecha,'YYYY')='2012'
GROUP BY id_revisor
HAVING SUM(puntuacion) >= (SELECT MAX(SUM(puntuacion)) FROM revision WHERE
TO_CHAR(fecha,'YYYY')='2012' GROUP BY id_revisor));
```

SQL 2: Encontrar el nombre y la tarifa de los revisores con categoría SENIOR que, para cada empresa que hay en la base de datos, han hecho al menos una revisión con una puntuación mayor de 75 puntos a un proyecto de esa empresa.

```
SELECT nombre_revisor, tarifa FROM revisor WHERE categoria='SENIOR'
AND id_revisor IN(
    SELECT DISTINCT id_revisor FROM revisor WHERE NOT EXISTS(
        SELECT id_empresa FROM empresa
        MINUS
        SELECT id_empresa FROM revision NATURAL JOIN (SELECT
            id_proyecto, id_empresa FROM proyecto)
        WHERE puntuacion > 75 AND revision.id_revisor=revisor.id_revisor
    ));
```

¿Tu sueño es estudiar Medicina
pero no sabes si podrás conseguirlo?
Pues deja de soñar y vive tus sueños.



LITHUANIAN UNIVERSITY
OF HEALTH SCIENCES

ACCEDE A MEDICINA EN LITUANIA
sin nota de corte y 100% en inglés



WEBINAR GRATUITO

Daniel Pérez Ruiz

SQL 3: Encontrar, considerando sólo proyectos que han tenido alguna revisión durante el año 2015, la fecha de elaboración del proyecto que ha tenido más revisiones en dicho año. Puede haber más de un proyecto que cumpla esta condición.

```
SELECT fecha_elaboracion FROM proyecto NATURAL JOIN
(SELECT id_proyecto FROM revision WHERE TO_CHAR(fecha,'YYYY')='2015'
GROUP BY id_proyecto
HAVING COUNT(*) >= (SELECT MAX(COUNT(*)) FROM revision WHERE
TO_CHAR(fecha,'YYYY')='2015' GROUP BY id_proyecto));
```

/*****

MODELO 3. Considere el siguiente esquema:

```
MODELO(id_modelo, marca, descripcion);

VEHICULO(matricula, id_modelo, fecha_matriculacion);
id_modelo apunta a : MODELO.

MECANICO(id_mecanico, nombre_mecanico, cargo, salario);

REPARA(id_mecanico, matricula, fecha, numero_horas);
id_mecanico apunta a : MECANICO.
matricula apunta a : VEHICULO.
```

Cada vehículo tiene asignado un modelo de una marca determinada. La tabla REPARA registra reparaciones indicando qué mecánico repara qué vehículo en qué fecha.

SQL 1: Encontrar, de entre los coches que han sufrido reparaciones en el año 2017, la fecha de matriculación del coche al que le han realizado más reparaciones en dicho año. Puede haber más de un coche que cumpla esta condición.

```
SELECT fecha_matriculacion FROM vehiculo NATURAL JOIN
(SELECT matricula FROM repara WHERE TO_CHAR(fecha,'YYYY')='2017'
GROUP BY matricula
HAVING COUNT(*) >= (SELECT MAX(COUNT(*)) FROM repara WHERE
TO_CHAR(fecha,'YYYY')='2017' GROUP BY matricula));
```

EDUCOWAY
EUROPEAN EXCELLENCE EDUCATION



Escoge entre:
• MEDICINA · ODONTOLOGÍA
• ENFERMERÍA · VETERINARIA

Lunes, 14
de junio
de 2021,
a las 18h.

WUOLAH

SQL 2: Encontrar la matrícula y la fecha de matriculación de los vehículos con id_modelo igual a AURIS que, por cada cargo de mecánico que han en la BD, han sido atendidos al menos una vez durante más de una hora por un mecánico con ese cargo.

```
SELECT matricula, fecha_matriculacion FROM vehiculo WHERE id_modelo='AURIS'
AND matricula IN(
    SELECT DISTINCT matricula FROM vehiculo WHERE NOT EXISTS(
        SELECT cargo FROM mecanico
        MINUS
        SELECT cargo FROM repara NATURAL JOIN (SELECT
            id_mecanico, cargo FROM mecanico)
        WHERE numero_horas > 1 AND repara.matricula=vehiculo.matricula
    )
);
```

/*****

MODELO 4. Considere el siguiente esquema:

```
GENERO(id_genero, nombre_genero, pais_origen);

GRUPO(id_grupo, id_genero, nombre_grupo, fecha_creacion);
    id_genero apunta a : GENERO.

POBLACION(id_poblacion, nombre_poblacion, numero, habitantes);

ACTUACION(id_grupo, id_poblacion, fecha, duracion);
    id_grupo apunta a : GRUPO.
    id_poblacion apunta a : POBLACION.
```

Cada grupo musical tiene asignado un género musical. La tabla ACTUACION registra actuaciones indicando qué grupo actúa en qué población y cuándo.

ÁLGEBRA RELACIONAL 1: Elaborar un listado de los países que han dado origen a un solo género musical.

ALIAS(genero) = G1 ; **ALIAS**(genero) = G2

PROYECCIÓN_{G1.pais_origen}(G1) -
PROYECCIÓN_{G1.pais_origen}(**SELECCIÓN**<sub>{G1.pais_origen = G2.pais_origen ^
G1.id_genero != G2.id_genero}</sub>(G1 x G2))

/*****

EJERCICIOS ÁLGEBRA RELACIONAL - ORDINARIO 2021

AR1: Considere las tablas $A(\underline{F}, B)$, $D(\underline{E}, B)$ y $C(\underline{F}, \underline{E})$, donde el atributo F de la tabla C es una **clave externa** que apunta al atributo F de la tabla A y el atributo E de la tabla C es una **clave externa** que apunta al atributo E de la tabla D . La **clave primaria** de A es F , la de D es E y la de C está formada por **sus dos atributos**.

Encontrar los valores del atributo F de la tabla A que cumplen la condición de que aparecen relacionados en la tabla C con todos los valores de E que tienen asociado en la tabla D el mismo valor del atributo B que tiene asociado el valor de F en la tabla A . Es decir, un valor x debería aparecer en el resultado si existe una tupla (x,y) en A tal que, para cada tupla (z,y) de D , existe una tupla (x,z) en C (siendo x un valor de F , z un valor de E e y un valor de B).

/*****/

AR2: Considere las tablas $B(\underline{E}, A)$, $D(\underline{F}, A)$ y $C(\underline{E}, \underline{F})$, donde el atributo E de la tabla C es una **clave externa** que apunta al atributo E de la tabla B y el atributo F de la tabla C es una **clave externa** que apunta al atributo F de la tabla D . La **clave primaria** de B es E , la de D es F y la de C está formada por **sus dos atributos**.

Encontrar los valores del atributo E de la tabla B que cumplen la condición de que aparecen relacionados en la tabla C con todos los valores de F que tienen asociado en la tabla D el mismo valor del atributo A que tiene asociado el valor de E en la tabla B . Es decir, un valor “ x ” debería a aparecer en el resultado si existe una tupla (x,y) en B tal que, para cada tupla (z,y) de D , existe una tupla (x,z) en C (siendo “ x ” un valor de E , “ z ” un valor de F e “ y ” un valor de A).

/*****/