

EJERCICIOS DE REPASO

TEMA 5. EDICIÓN DE LOS DATOS

BD LIGA TERCERA

- Consulta 1: Realiza una consulta para obtener cuantos partidos ha jugado cada equipo como local. Para hacerlo hay que obtener en cuantos partidos aparece cada equipo como local teniendo golesloc a no nulo.

BD LIGA TERCERA

- Consulta 2: Realiza una consulta para obtener cuantos partidos ha jugado cada equipo como local. Para hacerlo hay que obtener en cuantos partidos aparece cada equipo como local teniendo golesloc a no nulo.

```
SELECT eqloc, count(*) AS jugadoslocal  
FROM partidos  
WHERE golesloc IS NOT NULL  
GROUP BY eqloc;
```

BD LIGA TERCERA

- Consulta 2: Establece que en la tabla **clasificación**, cada equipo tenga en la columna de partidos jugados el número de partidos que ha jugado como local.

BD LIGA TERCERA

- Consulta 2: Establece que en la tabla **clasificación**, cada equipo tenga en la columna de partidos jugados el número de partidos que ha jugado como local.

```
UPDATE clasificacion
SET pj=(SELECT count(*)
        FROM partidos
        WHERE eqloc=codeq
        AND golesloc IS NOT NULL);
```

BD LIGA TERCERA

- Explicación de la modificación.
- En primer lugar hemos hecho una consulta con un group by.
- Cuando queremos utilizar una consulta de group by para modificar CADA campo de una tabla, tenemos que establecer la relación entre la tabla a modificar con la tabla de donde sacamos la información

```
UPDATE clasificacion
SET pj=(SELECT count(*)
        FROM partidos
        WHERE eqloc=codeq
        AND golesloc IS NOT NULL);
```

Cada **equipo** a modificar en **clasificación**, se relaciona con cada **equipo que jugo en local** en **partidos**

BD EMPLOYEES

- Elimina los empleados del departamento **Human Resources** cuyo salario actual sea superior a la media de los salarios actuales en ese departamento.
- Para ello, primero calculamos la media de los salarios actuales del departamento Human Resources.

BD EMPLOYEES

- Si quisiéramos obtener la media de los salarios actuales de los empleados de **Human Resources**, haríamos:

```
SELECT avg(salary)
FROM salaries
INNER JOIN employees USING(emp_no)
INNER JOIN dept_emp USING(emp_no)
INNER JOIN departments USING(dept_no)
WHERE salaries.to_date='9999-01-01'
AND dept_name='human resources';
```

- Ahora hacemos la instrucción de borrado...

BD EMPLOYEES

- La instrucción completa de borrado sería:

```
DELETE employees
FROM employees
INNER JOIN salaries USING (emp_no)
INNER JOIN dept_emp USING (emp_no)
INNER JOIN departments USING (dept_no)
WHERE dept_name='human resources' AND salaries.to_date='9999-01-01'
AND salary>(select * from(SELECT avg(salary)
                        FROM salaries
                        INNER JOIN employees USING(emp_no)
                        INNER JOIN dept_emp USING(emp_no)
                        INNER JOIN departments USING(dept_no)
                        WHERE salaries.to_date='9999-01-01' AND dept_name='human resources')as a);
```

REPASO DE INSERT

- En el INSERT se pueden utilizar el Values con subconsultas, o directamente insertar el resultado de una consulta concreta.
- Lo vemos con un ejemplo de los apuntes
- **Ejemplo 1:** queremos insertar mediante una sola instrucción dos nuevos contratos realizados en la fecha actual por el cliente de DNI 11223344M para los automóviles de matrícula 5031JHL y 4738JBJ. En kilómetros iniciales del contrato, se deben cargar los kilómetros que hay registrados en los automóviles de esas matrículas.
- Lo intentamos utilizando values.

REPASO DE INSERT

Cuando se utilizan subconsultas dentro de un Values, esta consulta solo puede devolver un único valor.

- 1ª versión, con Values:

```
INSERT INTO contratos (matricula,dnicliente,fini,kini) VALUES  
('5031JHL' ,'11223344M',curdate(),(SELECT kilometros FROM  
automoviles WHERE matricula='5031JHL')),  
('4738JBJ', '11223344M',curdate(),(SELECT kilometros FROM  
automoviles WHERE matricula='4738JBJ'));
```

- Ahora lo intentamos con una subconsulta:

REPASO DE INSERT

- 2º versión: inserta el resultado de una consulta directamente

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)  
SELECT matricula,'11223344M',curdate(),kilometros  
FROM automoviles  
WHERE matricula='5031JHL' OR matricula='4738JBJ';
```

Cuando se inserta el resultado de una select sin el values, esta consulta si que puede devolver varias filas de datos, no como en la diapositiva anterior.

- La matricula y los kilómetros lo saca de la tabla automóviles para los coches que son de las matriculas 5031JHL o 4738JBJ.

REPASO DE INSERT

- **Ejemplo 2:** Añadir un nuevo contrato con fecha de hoy realizado por Sandra Flores Jorje sobre el automóvil de matrícula '2058JGF' poniendo los kilómetros iniciales a los kilómetros del automóvil.
- Versión con VALUES, utilizando subconsultas

REPASO DE INSERT

- Versión con VALUES, utilizando subconsultas

```
INSERT INTO contratos (matricula, dnicliente, fini, kini)
```

```
VALUES ('2058JGF' ,
```

```
(SELECT dni FROM clientes WHERE nombre='Sandra' AND  
apellidos='flores jorje'),
```

```
curdate()),
```

```
(SELECT kilometros FROM automoviles WHERE matricula='2058JGF'));
```

- Ahora inténtalo como resultado de una select

REPASO DE INSERT

- 2º versión: insertando el resultado de una select. En este caso habría que hacer producto cartesiano

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
SELECT '2058JGF' , dni, curdate(), kilometros
FROM clientes, automoviles
WHERE matricula='2058JGF'
AND nombre='Sandra'
AND apellidos='flores jorje';
```

Se hace producto cartesiano porque no hay relación entre el que alquila el coche con el coche en cuestión. Entonces, se hacen todas las combinaciones entre clientes y coches, pero solo te quedas con la matricula y el cliente del enunciado.

REPASO DE INSERT

- **Ejemplo 3:** Añadir un nuevo contrato con fecha de hoy realizado por Anais Rodriguez sobre el automóvil más barato de los que no tienen un contrato sin finalizar actualmente. En kilómetros iniciales pondremos el valor cero.
- Lo primero vamos a sacar el automóvil más barato de los que no están contratados actualmente (según el campo alquilado de la misma tabla automoviles), haríamos una select.

REPASO DE INSERT

- Hay que tener en cuenta que para sacar el automóvil más barato de los que no están contratados actualmente (según el campo alquilado de la misma tabla automóviles), haríamos:

```
SELECT matricula  
FROM automoviles  
WHERE alquilado=false  
ORDER BY precio LIMIT 1;
```

- Ahora hacemos la instrucción completa del insert.

REPASO DE INSERT

- La instrucción final sería:

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
```

```
VALUES (
```

```
(SELECT matricula FROM automoviles WHERE alquilado=false ORDER  
BY precio LIMIT 1),
```

```
(SELECT dni FROM clientes WHERE nombre='Anais' AND  
apellidos='Rodriguez'),
```

```
curdate(),0);
```

REPASO DE INSERT

- **Ejemplo 4:** Añadir un nuevo contrato con fecha de hoy realizado por la cliente de dni ' 11223344M ' sobre los 3 automóviles más baratos. En kilómetros iniciales pondremos el valor cero.
- Esto no lo podemos hacer con una subconsulta de esta forma (estaríamos tratando de insertar 3 matrículas en un mismo VALUES), por tanto, en un mismo contrato:

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
```

```
VALUES ((SELECT matricula FROM automoviles ORDER BY precio LIMIT 3),
```

```
' 11223344M ',curdate(),0);
```

- Lo hacéis con Select

Esto da error, porque la select que hay dentro de values va a devolver más de un resultado, con lo que no se puede hacer de este modo.

REPASO DE INSERT

- Solución:

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
SELECT matricula, '11223344M', curdate(), kilometros
FROM automoviles
WHERE matricula NOT IN (SELECT matricula
                        FROM contratos
                        WHERE ffin IS NULL)
ORDER BY precio LIMIT 3;
```

Inserta en contratos el resultado de la consulta, con la condición que su contrato esté finalizado, y solo lo hace de los res más baratos.

REPASO DE INSERT

- **Ejemplo 5:** Añadir un nuevo contrato con fecha de hoy realizado por Anais Rodriguez sobre el automóvil más barato de los que no tienen un contrato sin finalizar actualmente. En kilómetros iniciales pondremos el valor cero.
- Hay que tener en cuenta que para sacar el automóvil más barato de los que no están contratados actualmente (según la fecha final de contratos), hacemos esta select.

REPASO DE INSERT

- Automovil más barato que tiene contrato finalizado segun su fecha final de contrato.

```
SELECT matricula  
FROM automoviles  
WHERE matricula NOT IN (SELECT matricula  
                        FROM contratos WHERE ffin IS NULL)  
ORDER BY precio LIMIT 1;
```

- Ahora, hacemos la inserción completa que se pedía.

REPASO DE INSERT

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)
VALUES (
(SELECT matricula FROM automoviles
WHERE matricula NOT IN (SELECT matricula
                        FROM contratos WHERE ffin IS NULL)
ORDER BY precio LIMIT 1),
(SELECT dni FROM clientes
WHERE nombre= 'Anais' AND apellidos= 'Rodriguez '),
curdate(), 0);
```

Problema de esta solución:
no se puede usar como
subconsulta la misma tabla
en la que quieres insertar.

REPASO DE INSERT

- Solución: HACER UN RENOMBRADO A TABLA DE UNA CONSULTA SOBRE CONTRATOS.

```
INSERT INTO contratos (matricula,dnicliente,fini,kini)  
VALUES (
```

```
(SELECT matricula FROM automoviles
```

```
WHERE matricula NOT IN (SELECT a.matricula
```

```
FROM (SELECT matricula
```

```
FROM contratos WHERE ffin IS NULL ) AS a)
```

```
ORDER BY precio LIMIT 1),
```

```
(SELECT dni FROM clientes WHERE nombre= 'Anais' AND apellidos= 'Rodriguez'),  
curdate(), 0);
```

Envolvemos la subconsulta en un select * from () as alias. De este modo ya se puede usar la misma tabla en la subconsulta y en la inserción.

REPASO DE UPDATE

- **Ejemplo 1:** Modificar la columna alquilado de la tabla automóviles para que todos los automóviles que están actualmente contratados queden marcados como no disponibles y que los que no están contratados figuren como disponibles.
- Primero hacemos un update para poner todos a false
- Luego hacemos un update para poner a true los que si están alquilados.

REPASO DE UPDATE

- **Ejemplo 1:** Modificar la columna alquilado de la tabla automóviles para que todos los automóviles que están actualmente contratados queden marcados como no disponibles y que los que no están contratados figuren como disponibles.

UPDATE automoviles SET alquilado=false;

UPDATE automoviles

INNER JOIN contratos ON contratos.matricula=automoviles.matricula

SET alquilado=true WHERE ffin IS NULL;

Si quiero modificar una tabla pero la condición está en otra tabla, hacemos inner join.

REPASO DE UPDATE

- Para los update, se pueden hacer las instrucciones con subconsultas, o utilizando inner join
- **Ejemplo 4:** Modificar las fechas de inicio y de finalización de los contratos realizados por Mariano Dorado para que tengan una fecha un día superior a la que tienen.
- Hacemos este ejemplo con subconsultas.

REPASO DE UPDATE

- **Ejemplo 4:** Modificar las fechas de inicio y de finalización de los contratos realizados por Mariano Dorado para que tengan una fecha un día superior a la que tienen.

UPDATE contratos

SET fini=adddate(fini,INTERVAL 1 DAY),ffin=adddate(ffin,INTERVAL 1 DAY)

WHERE dnicliente= (SELECT dni
FROM clientes
WHERE nombre='Mariano' AND apellidos='Dorado');

Dentro de UPDATE, podemos usar subconsultas, tanto para obtener los valores que se asignan con SET como para establecer las condiciones WHERE.

- Esta consulta también se puede hacer con inner join

REPASO DE UPDATE

- **Ejemplo 4** con inner join

UPDATE **contratos**

INNER JOIN **clientes** ON dni=dncliente

SET **fini**=adddate(**fini**,INTERVAL 1 DAY),**ffin**=adddate(**ffin**,INTERVAL 1 DAY)

WHERE **nombre**='Mariano' AND **apellidos**='Dorado';

Se modifica la tabla contratos,
pero las condiciones son de la
tabla de clientes, por eso hay
que hacer inner join

REPASO DE UPDATE

- **Ejemplo 5:** Establecer que el contrato 25 fue realizado por Antonio Diaz Vera.
- La hacemos con subconsultas

REPASO DE UPDATE

- **Ejemplo 5:** Establecer que el contrato 25 fue realizado por Antonio Diaz Vera.

UPDATE contratos

SET dnicliente=(SELECT dni FROM clientes WHERE nombre='Antonio'
AND apellidos='Diaz vera')

WHERE numcontrato=25;

REPASO DE UPDATE

- **Ejemplo 6:** Establecer que en los kilómetros de los automóviles se carguen los kilómetros finales del último contrato finalizado de cada automóvil.
- Es el mismo caso que vimos en la diapositiva 6. Hay que actualizar CADA automóvil en función de su contrato.

REPASO DE UPDATE

- **Ejemplo 6:** Establecer que en los kilómetros de los automóviles se carguen los kilómetros finales del último contrato finalizado de cada automóvil.

Como hay que relacionar los contratos de cada automóvil con el máximo de km que tiene ese automóvil en contratos, la condición junta el campo de la tabla a modificar con la tabla donde establecemos la condición

```
UPDATE automoviles
SET kilometros=(SELECT max(kfin)
                FROM contratos
                WHERE kfin IS NOT NULL
                AND contratos.matricula=automoviles.matricula);
```

REPASO DE UPDATE

- Como puede verse, en la subconsulta, se compara la matricula de cada contrato con la matricula del automóvil que se está modificando. La ejecución de esta instrucción supone que por cada automóvil, se ejecuta la subconsulta para obtener el máximo valor de kilómetros finales para ese automóvil.
- Si un automóvil no tuviera contratos, se cargaría el valor NULL en kilómetros, lo cual no sería muy adecuado.

REPASO DE UPDATE

- **Ejemplo 7:** Para que no ocurra lo anterior, modifica los kilómetros sólo de automóviles con contratos finalizados para que contengan los kilómetros finales de cada uno de los últimos contratos finalizados.
- La solución es haciendo subconsultas, una para el SET y otra para el WHERE.

REPASO DE UPDATE

- **Ejemplo 7:** Para que no ocurra lo anterior, modifica los kilómetros sólo de automóviles con contratos finalizados para que contengan los kilómetros finales de cada uno de los últimos contratos finalizados.

- Una solución es:

UPDATE automoviles

SET kilometros=(SELECT max(kfin)

FROM contratos

WHERE kfin IS NOT NULL

AND contratos.matricula=automoviles.matricula)

WHERE matricula IN (SELECT matricula

FROM contratos

WHERE ffin IS NOT NULL);

REPASO DE UPDATE

- **Ejemplo 8:** Establecer que el contrato número 26 ha sido realizado por el mismo cliente del contrato número 4.
- Si hacemos, como puede suponerse:

```
UPDATE contratos
SET dnicliente= (SELECT dnicliente
                FROM contratos
                WHERE numcontrato=4)
WHERE numcontrato=26;
```

Da error, por utilizar en la subconsulta la misma tabla que quieres modificar.

- Para solucionarlo hay que usar un renombrado de subconsulta a tabla

REPASO DE UPDATE

- Para solucionarlo hay que usar un renombrado de subconsulta a tabla:

UPDATE contratos

SET dnicliente=(SELECT a.dnicliente
FROM (SELECT *
FROM contratos
WHERE numcontrato=4) AS a)

WHERE numcontrato=26;

Envolvemos la subconsulta
haciendo un renombrado.
Ahora ya no da error.

REPASO DE DELETE

- **Ejemplo 1:** Eliminar todos los clientes que no hayan realizado contratos.
- Hacemos el Delete con un left join

REPASO DE DELETE

- **Ejemplo 1:** Eliminar todos los clientes que no hayan realizado contratos.

- Esto esta hecho con la versión de left join

DELETE clientes

FROM clientes LEFT JOIN contratos ON dni=dnicliente

WHERE numcontrato IS NULL;

- Se puede hacer con subconsultas

REPASO DE DELETE

- **Ejemplo 1:** Eliminar todos los clientes que no hayan realizado contratos.
- Con subconsultas

DELETE FROM clientes

WHERE dni NOT IN (SELECT DISTINCT dnicliente FROM contratos);

REPASO DE DELETE

- **Ejemplo 2:** Eliminar los contratos realizados por Mariano Dorado.
- Como queremos borrar de una tabla (contratos) pero las condiciones están en otra tabla (Clientes), hay que hacer un inner join en el delete

REPASO DE DELETE

- **Ejemplo 2:** Eliminar los contratos realizados por Mariano Dorado.

```
DELETE contratos
FROM clientes INNER JOIN contratos ON dni=dnicliente
WHERE nombre='Mariano' AND apellidos='Dorado';
```

Queremos borrar de una tabla pero las condiciones para borrar están en otra tabla, se hace con inner join.

REPASO DE DELETE

- **Ejemplo 6:** Eliminar todos los contratos realizados el mismo día que el día de inicio del último contrato del cliente con dni 03549358G.

```
DELETE FROM contratos
```

```
WHERE fini=(SELECT fini
```

```
FROM contratos
```

```
WHERE dnicliente= '03549358G'
```

```
ORDER BY numcontrato DESC LIMIT 1);
```

Esto da error porque no se puede utilizar la tabla de la que quieres borrar en la subconsulta

- Corrige el error.

REPASO DE DELETE

- **Ejemplo 6:** Solución.

DELETE FROM contratos

WHERE fini=(SELECT a.fini
FROM (SELECT * FROM contratos) AS a
WHERE a.dnicliente= '03549358G'
ORDER BY a.numcontrato DESC LIMIT 1);

REPASO DE DELETE

- **Ejemplo 7:** Eliminar de la tabla contratos todos los contratos realizados en el año anterior al actual y anteriores al primer contrato realizado ese año por la clienta Reyes Sanz Lopez.

El borrado tiene dos condiciones. El primero es que el año sea anterior al actual, y además, que sea anterior al contrato realizado por una cliente. Como la condición implica la misma tabla de la que quieres borrar, hay que hacer un renombrado para que funcione

REPASO DE DELETE

- **Ejemplo 7:** Eliminar de la tabla contratos todos los contratos realizados en el año anterior al actual y anteriores al primer contrato realizado ese año por la clienta Reyes Sanz Lopez.

```
DELETE FROM contratos
WHERE year(fini)=year(curdate())-1
AND fini< (SELECT a.fini FROM
           (SELECT *
            FROM contratos
            INNER JOIN clientes ON dnicliente=dni) AS a
           WHERE a.nombre='Reyes' AND a.apellidos='Sanz Lopez'
           ORDER BY numcontrato LIMIT 1);
```

El borrado tiene dos condiciones. El primero es que el año sea anterior al actual, y además, que sea anterior al contrato realizado por una cliente. Como la condición implica la misma tabla de la que quieres borrar, hay que hacer un renombrado para que funcione