# 9.5. Manipulación de datos y vistas

Las sentencias `UPDATE`, `DELETE` o `INSERT` se pueden utilizar para actualizar el contenido de las tablas subyacentes.

Además las vistas reflejan instantáneamente los cambios producidos en las tablas, de manera que aunque los datos cambien, estos cambios se verán reflejados en la consulta que se haga sobre la vista.

- Para que una vista sea actualizable, debe haber una relación uno a uno entre los registros de la vista y los registros de la tabla subyacente.

# 9.5. Manipulación de datos y vistas

Existen además ciertas restricciones. Para que una vista sea actualizable, no se puede usar:

- Funciones agregadas.
- Cláusulas `DISTINCT`.
- Subconsultas.
- Tablas temporales
- Cláusulas `GROUP BY` ni `HAVING`.
- Uniones ni combinaciones externas.
- Consultas correlacionadas.

# 9.5. Manipulación de datos y vistas

Si se quieren añadir registros mediante sentencias `INSERT`, las columnas de la vista actualizable también tienen que cumplir los siguientes requisitos adicionales:

- No debe haber nombres duplicados entre las columnas de la vista.
- La vista debe contener todas las columnas de la tablas usadas que no tengan indicado un valor por defecto.
- Las columnas de la vista deben ser referencias a columnas simples y no columnas derivadas.
  - Una columna derivada es una que deriva de una expresión.
  - No se pueden insertar registros en una vista que contenga una combinación de columnas simples y derivadas, pero esta vista sí se puede actualizar si se actualizan únicamente las columnas derivadas.

# 9.5. Manipulación de datos y vistas

**Ejemplo.** Si tenemos la tabla t1 de la base test, podemos crear la siguiente vista, consistente en dos campos *a* y un campo derivado *b* cuyo calor es el doble de *a* para cada fila.

```
CREATE VIEW v1 AS SELECT a, 2*a AS b FROM t1;
```

- Entonces podríamos únicamente modificar el campo no derivado *a* con:

```
UPDATE v1 SET a=a+1;
```

- Pero no modificar el campo derivado *b*:

```
UPDATE v1 SET b=b+1;
Error 1348: Column b is not updatable
```

# 9.5. Manipulación de datos y vistas

**Ejemplo**. Obtenemos las vistas v3 y v2 basadas ambas en la vista v1 que, a su vez, se basa en una tabla t1. Sin embargo las sentencias:

```
CREATE VIEW v1 AS SELECT*FROM t1 WHERE a<2 WITH CHECK OPTION;
CREATE VIEW v2 AS SELECT*FROM v1 WHERE a>0 WITH LOCAL CHECK OPTION;
CREATE VIEW v3 AS SELECT*FROM v1 WHERE a>0 WITH CASCADED CHECK OPTION;
```

Si ahora ejecutamos las sentencias:

```
INSERT INTO v2 VALUES (2);
INSERT INTO v3 VALUES (2);
```

La segunda producirá un error, ya que la condición `WHERE` se comprueba en *v3* cascada, es decir, también en la vista *v1* subyacente.

# 9.6. Integridad referencial

- Debemos considerar las reglas de borrado y modificación que impusimos en el diseño a la hora de eliminar o modificar registros.

- Hasta ahora hemos obviado esta parte y, sin embargo, puede ser una gran fuente de errores si no lo tenemos en cuenta.

- Trabajar con integridad referencial puede generar muchos problemas si no observamos cuidadosamente lo anterior.

# 9.6. Integridad referencial

En MySQL/MariaDB tenemos cuatro opciones de integridad referencial:

- CASCADE, Propaga las actualizaciones o borrados a los registros de la tabla hija relacionados con los de la tabla padre. Es el valor más habitual.
- RESTRICT o NO ACTION. Impide el borrado o actualización de un registro en una tabla que tiene registros en otras tablas.
- SET NULL. Permite borrar o actualizar un registro en la tabla padre poniendo a NULL el campo o campos relacionados con la tabla hija (salvo que se hayan definido como NOT NULL).
- SET DEFAULT. No funciona con InnoDB.

# 9.6. Integridad referencial. Inserción

Cuando queremos hacer inserciones de datos masivas, la integridad referencial puede suponer una fuente de errores, especialmente cuando hay relaciones entre campos de dos tablas.

- Este tipo de problemas se pueden evitar insertando primero los datos y añadiendo después las restricciones de integridad.

# 9.6. Integridad referencial. Actualización

Las modificaciones de datos no suelen generar problemas, ya que se la integridad referencial de actualización se suele definir de tipo CASCADE.

- Si la restricción es de tipo SET NULL, la integridad referencial será una fuente de valores nulos en los campos relacionados
  - Además habrá que asegurar que los campos de las claves ajenas se hayan definido como no obligatorios.

# 9.6. Integridad referencial. Borrado

Cuando borramos registros debemos tener en cuenta el tipo de integridad referencial definida en el borrado.

- Si se definió integridad referencial de tipo CASCADE, los datos de registros relacionados se borrarán automáticamente en todas las tablas.
  - Está operación debe hacerse cuando se está muy seguro de sus implicaciones.
- Para el caso de borrados masivos de todos los datos ocurre a la inversa que en el caso de la inserción.
  - Añadiendo primero las restricciones de integridad e insertando después los datos.

# Actividad 3

Realiza consultas en la base de datos *liga*:

1. Crea una vista con los nombres y equipo de los jugadores del CAI Zaragoza. Crea otra vista basada en la anterior con los nombres de jugadores del CAI Zaragoza. Pon los nombres en mayúscula. Comprueba el resultado en la tabla base.
2. Crea una nueva vista con los datos de los jugadores del Valencia. Inserta en esta vista un nuevo jugador con los siguientes datos: ID 30, nombre John Smith, base, capitán 2, alta el 2021-01-01, salario 4000, equipo 2, altura 2, puntos 9. Comprueba si se ve en la vista y en la tabla base. ¿Qué ocurre? ¿Cómo se soluciona?
3. Crea dos vistas basadas en la tabla *jugador*, una con los campos *nombre*, *id_jugador*, *equipo* y *altura* y otra con *id_jugador*, *equipo* y *nombre*. ¿Se pueden insertar jugadores nuevos en ambas? ¿Por qué?

# Actividad 4

Realiza consultas en la base de datos *liga*:

1. Modifica lo que consideres necesario para poder aumentar el *id_equipo* de la tabla *equipos* en diez unidades para todos los registros.
2. Modifica lo que consideres necesario para poder borrar los registros de *equipos* que no hayan jugado *partidos.*

# Actividad 5

Realiza consultas en la base de datos *hospital*:

1. Crea una tabla que contenga los campos código de inscripción, NSS, apellido y fecha de nacimiento. Inserta a los enfermos que tengan sexo masculino.
2. Sube el sueldo un 10% a los empleados de plantilla que trabajan en las salas 2 y 3 del hospital con código 15 (La Paz).
3. Mueve a todos los enfermos nacidos en 1960 al Hospital Guadalajara.
4. Crea una tabla para enfermeros con las mismas columnas que la tabla plantilla. Inserta a todos los enfermeros y enfermeras
5. Borra a los enfermos de la tabla creada en el apartado 1 que tengan código de inscripción par.
6. Crea una vista con los empleados de la tabla plantilla que tienen el turno de noche.

# Actividad 6

Realiza consultas en la base de datos *educación*:

1. Cambia de centro a todos los *Conserjes*. Asígnalos al centro I.E.S. El quijote.
2. Crea una tabla nueva *profesores* con las *especialidades* de los profesores según su *DNI*. Asegúrate de que al borrar un profesor de *personal* se borra también de esta tabla. Las especialidades son: 12345678A: INFORMÁTICA, 41230050B: MATEMÁTICAS, 41220256C: MATEMÁTICAS, 98009908F: LENGUA, 11123457D: DIBUJO, 86609909A: LENGUA, 76500005A: MATEMÁTICAS, 43526789V: INFORMÁTICA.
3. Crea una vista llamada *especialistas* que muestre el *nombre* del centro, *apellidos*, *salario* y *especialidad* de los profesores. ¿Por qué no se puede añadir un nuevo profesor a esta vista?
4. Borra a los *profesores* de MATEMÁTICAS que están en el I.E.S. El quijote.