Unidad 5
Data Query Language (III)

DQL

Contenido:

- **Vistas**

Vistas

- Las **vistas** son objetos de la base de datos que ofrecen una **visión parcial** de la base de datos.
- Las vistas tienen asociada una consulta que puede estar utilizando tablas u otras vistas.
- Se utilizan como si fuesen tablas:
  - Se seleccionan sus columnas en la cláusula SELECT, se añaden en el FROM y se les puede aplicar filtros en el WHERE. También se pueden combinar con otras vistas o tablas.
  - Se les asigna un nombre que debe ser único en la base de datos

Vistas

- Existen 2 tipos de vistas:

1. **Lógicas** (normales):
   - La **consulta asociada se ejecuta cada vez** que se utiliza la vista.
   - Vistas estándar que almacenan solo la consulta SQL pero no los datos, **no contienen registros**.
   - Facilitan la reutilización de consultas complejas
   - Se actualizan automáticamente con los cambios en las tablas subyacentes (en las que se basan)
   - Al no almacenar los datos, pueden ser más lentas con las consultas complejas o en las que intervienen un gran volumen de datos.

Vistas

2. **Materializadas**:
   - Los registros de las consultas subyacentes se almacenan físicamente en la base de datos
   - Mejora el rendimiento ya que se evita recalcular los datos en cada consulta.
   - Adecuadas para reportes de datos históricos
   - Ocupan espacio en disco.
   - Pueden quedar desactualizadas, ya que dependen del refresco explícito de las consultas subyacentes.

Vistas

- Sintaxis, en mysql solo se pueden crear vistas lógicas (no materializadas).

```
CREATE [OR REPLACE] [SQL SECURITY { DEFINER | INVOKER }] VIEW nombd.nombre_vista [(columnas)]
AS consulta [WITH [CASCADED | LOCAL] CHECK OPTION]
```

  - **CHECK OPTION** se puede utilizar con vistas actualizables para evitar insertar o actualizar registros que no cumplan con la cláusula WHERE de la consulta en la que se basa la vista.
  - **SQL SECURITY** indica qué privilegios chequear al ejecutar la consulta. Pueden ser los del creador de la vista (definer), o del usuario que invoca la vista (invoker). **Por defecto** se aplica la opción SQL SECURITY **DEFINER**.
  - Columnas: Se puede indicar el nombre que tendrán las columnas de la vista, si no se utiliza esta opción, la vista ofrecerá como nombre de columnas los que corresponda de las tablas originales

Vistas

- Dada la consulta:

```
SELECT a.dni, a.nombre, m.curso, l.nota, aa.Nombre
FROM alumno a
INNER JOIN matricula m ON a.dni = m.dni
INNER JOIN lineamatricula l ON m.codmatr = l.codMatr
INNER JOIN asignatura aa ON l.codasig = aa.codAsig
INNER JOIN ciclo c ON c.codcf = aa.codCF
WHERE c.siglas ='DAM';
```

- Podemos definir la vista **AlumnosDAM** para poder reutilizar esta consulta.
- Será necesario indicar los nombres de las columnas porque en este caso 'nombre' está duplicado.

Vistas

- Añadiendo un alias para una de las columnas con nombre repetido:

```
CREATE OR REPLACE SQL SECURITY INVOKER VIEW ALUMNOSDAM AS
SELECT a.dni, a.nombre, m.curso, l.nota, aa.Nombre as NomAsig
FROM alumno a
INNER JOIN matricula m ON a.dni = m.dni
INNER JOIN lineamatricula l ON m.codmatr = l.codMatr
INNER JOIN asignatura aa ON l.codasig = aa.codAsig
INNER JOIN ciclo c ON c.codcf = aa.codCF
WHERE c.siglas ='DAM';
```

Vistas

- Indicando el nombre que tendrán las columnas de la vista:

```
CREATE OR REPLACE SQL SECURITY INVOKER
VIEW ALUMNOSDAW (dniAlumno, nombreAlumno, curso, notaAsig, Asig) AS
SELECT a.dni, a.nombre, m.curso, l.nota, aa.Nombre
FROM alumno a
INNER JOIN matricula m ON a.dni = m.dni
INNER JOIN lineamatricula l ON m.codmatr = l.codMatr
INNER JOIN asignatura aa ON l.codasig = aa.codAsig
INNER JOIN ciclo c ON c.codcf = aa.codCF
WHERE c.siglas ='DAW';
```

Vistas

- DML con las vistas:
  - Para que una vista sea actualizable manteniendo la integridad de los datos, se debe añadir la opción WITH CHECK OPTION, que garantiza que se realizarán los cambios siempre que cumplan las restricciones de visibilidad y actualización.
  - No se pueden actualizar vistas que utilizan:
    - Funciones agregadas
    - DISTINCT
    - GROUP BY
    - HAVING