Unidad de Trabajo 4
Realización de consultas

# El lenguaje SQL

IES Palomeras Vallecas

Curso 2020/2021

Profesor: Alberto Ruiz

# Historia de SQL

- 1974: un grupo de trabajo en los laboratorios de investigación de IBM buscaba un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional.
- Este lenguaje se llamaba **SEQUEL** (*Structured English Query Language*)
- Entre 1976 y 1977: revisión del lenguaje (**SEQUEL/2**)
- A partir de ese momento cambió de nombre por motivos legales, convirtiéndose en **SQL** *(Structured Query Language).*

# Historia de SQL

- A lo largo de los años ochenta, numerosas compañías (por ejemplo Oracle y Sybase) comercializaron productos basados en SQL
- SQL se convierte en el **estándar** industrial de hecho (*de facto)* en lo que respecta a las bases de datos relacionales.

# Historia de SQL

- En 1986, el ANSI adoptó SQL como estándar para los lenguajes relacionales.
- En 1987 se transformó en estándar ISO.
  - Esta versión del estándar va con el nombre de **SQL/86**.
- En los años siguientes, éste ha sufrido diversas revisiones:
  - La primera: **SQL:89**
  - La última a fecha de 2021: **SQL:2016**

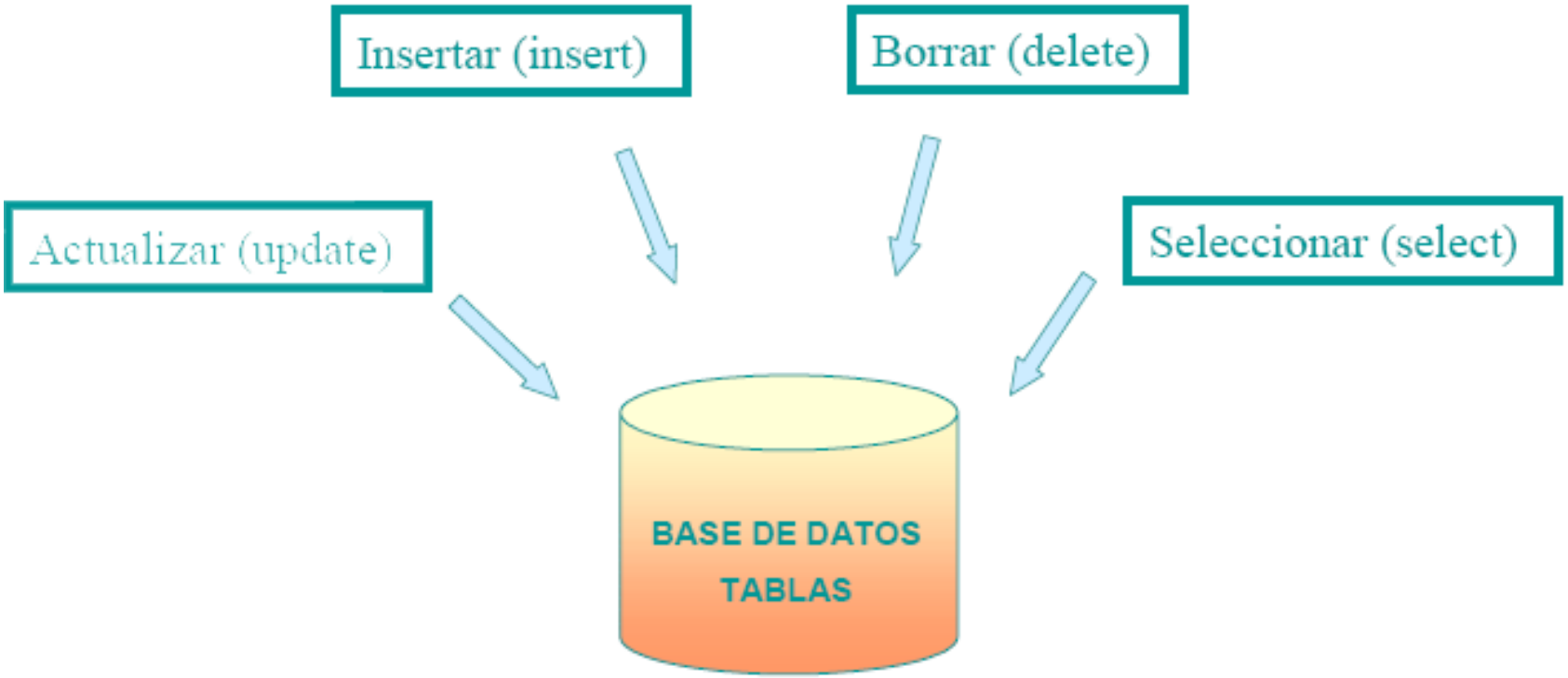
# Funcionamiento de SQL

- Consulta SQL para acceder a la BD:

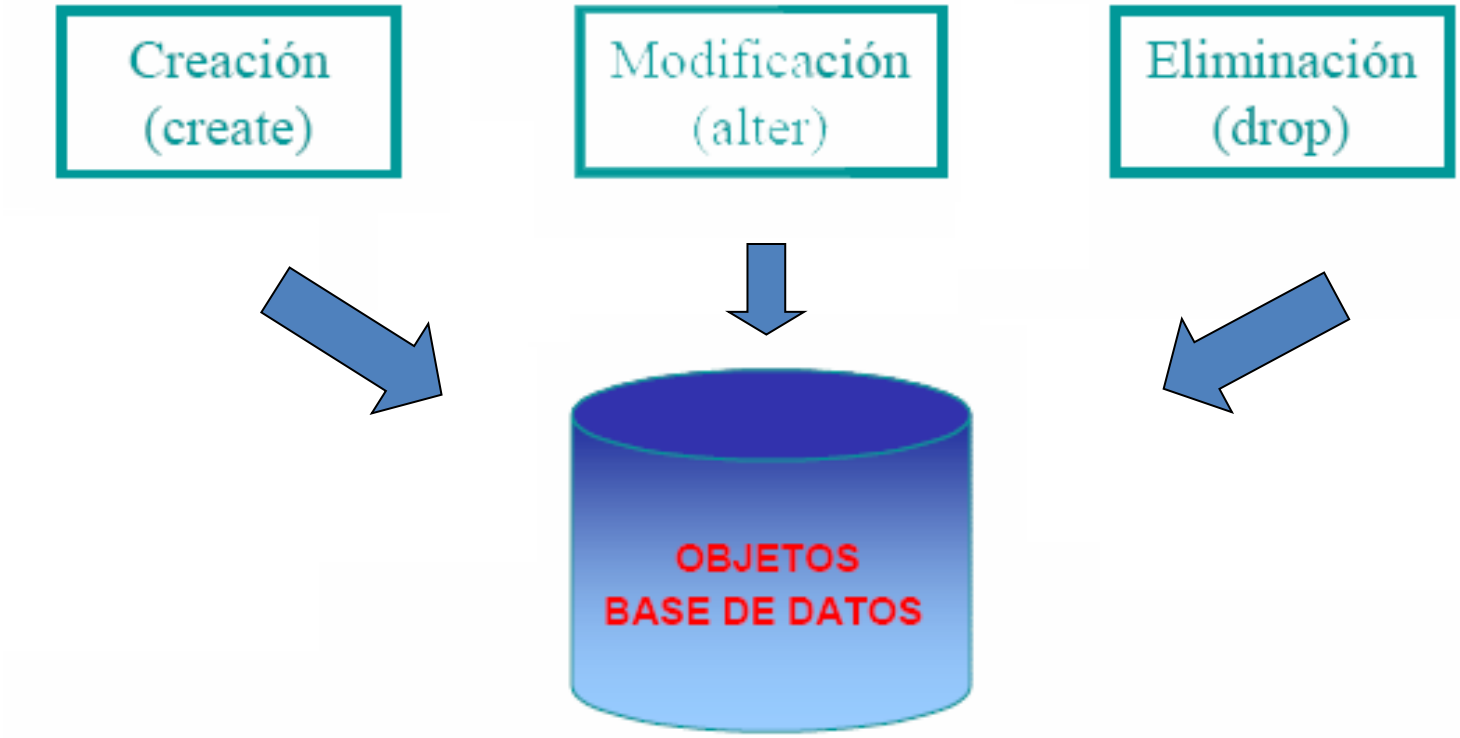
# Las tres caras de SQL

- **DDL** (Lenguaje de Definición de Datos)
  - Definición de dominios, tablas, vistas y restricciones de integridad
- **DML** (Lenguaje de Manipulación de Datos)
  - Realizar consultas, insertar, borrar o modificar datos
- **DCL** (Lenguaje de Control de Datos)
  - Gestionar privilegios, permisos, etc

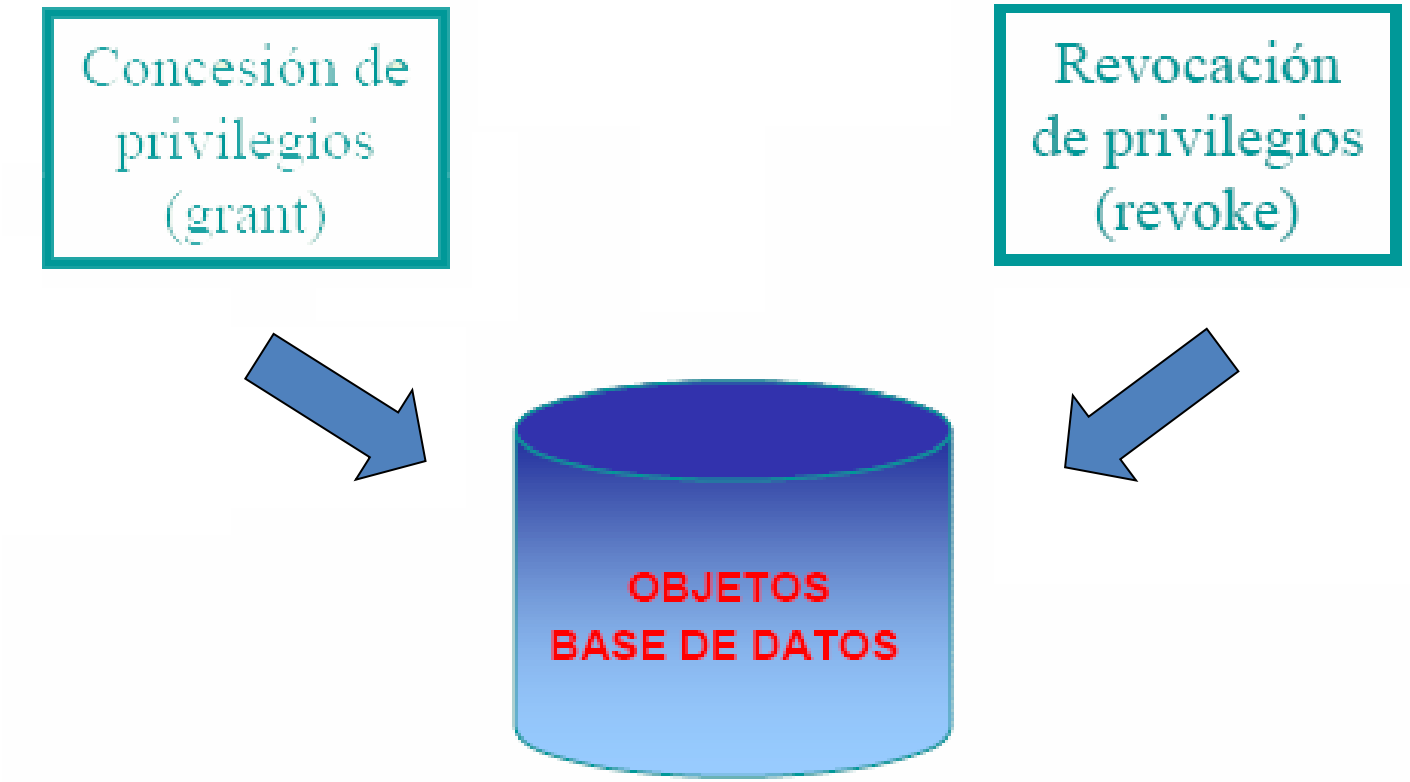
# SQL como DML


Insertar (insert)
Borrar (delete)
Actualizar (update)
Seleccionar (select)
BASE DE DATOS
TABLAS

# SQL como DDL


Creación
(create)
Modificación
(alter)
Eliminación
(drop)
OBJETOS
BASE DE DATOS

# SQL como DCL


Concesión de privilegios (grant)
Revocación de privilegios (revoke)
OBJETOS
BASE DE DATOS

# Elementos de SQL

- Sentencia SQL: es una frase en la que expresamos qué deseamos obtener, y dónde buscarlo. Formada por comando, cláusulas y operadores

```
SELECT [DISTINCT] [Campo1, Campo2, ... | *]
        FROM [Tabla1, Tabla2, ...]
        [WHERE condición]
        [ORDER BY Campo1 [DESC|ASC] [, Campo2[DESC|ASC]...] ;
```

# Elementos de SQL

- Comandos: son verbos que indican la acción a realizar
  - Comandos DDL: CREATE, DROP, ALTER
  - Comandos DML: SELECT, INSERT, UPDATE, DELETE

**SELECT [DISTINCT] [*Campo1, Campo2, ... | **]**
**FROM [*Tabla1, Tabla2, ...*]**
**[WHERE *condición*]**
**[ORDER BY *Campo1* [DESC|ASC] [, *Campo2*[DESC|ASC]...] ;**

# Elementos de SQL

- Cláusulas: son condiciones de modificación utilizadas para definir qué datos queremos seleccionar o modificar
  - FROM, WHERE, ORDER BY, *GROUP BY, HAVING*

**SELECT [DISTINCT] [*Campo1, Campo2, ...* | *]**
**FROM [*Tabla1, Tabla2, ...*]**
**[WHERE *condición*]**
**[ORDER BY *Campo1* [DESC|ASC] [, *Campo2*[DESC|ASC]...] ;**

# Elementos de SQL

- Operadores: son elementos utilizados en las sentencias SQL para conectar las distintas condiciones existentes
  - Operadores de comparación: <, >, =, !=, BETWEEN, LIKE, IN)
  - Operadores lógicos: OR, NOT, AND

# Consultas básicas

## Selección de todos los datos de una tabla

- Selección de todos los campos y registros:
  - SELECT * FROM coches;

*Muestra información de todas las columnas y filas de la tabla coches*

# Consultas básicas

## Selección de campos concretos

- Selección de columnas o campos
  - SELECT modelo, precio FROM coches;

*Muestra el contenido de las columnas indicadas (modelo y precio) para todas las filas de la tabla coches*

# Consultas básicas

## Campos con espacios en blanco

- Selección de columnas o campos
  - SELECT **`Modelo de coche`**, precio FROM coches;

*Si el nombre del campo contiene espacios en blanco, es necesario utilizar el símbolo acento (`).*

*No confundir con comillas dobles o simples " '*

# Consultas básicas

## Selección de valores diferentes

- Selección de columnas o campos diferentes
  - SELECT DISTINCT nombre FROM empleados;

*Aunque haya varios empleados que se llamen igual, sólo se mostrará una vez cada nombre*

# Consultas básicas

## Cambio de nombre

- Cambio de nombre en los resultados:
  - SELECT nombre AS NombreAlumno FROM alumnos;

*Pedimos que el resultado se presente con un nombre diferente al que tenía el campo de la tabla*

- *Como en los créditos de las películas:*
  - *Daniel Radcliffe AS Harry Potter:*
    - *"En esta película vemos a Daniel COMO Harry Potter"*

# Consultas básicas

## Ordenación de resultados

- Si queremos ordenar la selección de registros:
  - SELECT * FROM empleados ORDER BY nombre, apellido DESC;

*Utilizamos la cláusula ORDER BY para ordenar por uno o varios campos (el orden importa).*

*Después de cada campo puedes indicar si debe ordenarse de forma ascendente o descendente:*

- *ASC es el orden por defecto si no pones nada.*
- *DESC indica orden de la Z a la A y del 9 al 0.*

# Consultas básicas

## Límite de número de resultados

- Mostrar un número concreto de resultados
  - SELECT * FROM productos LIMIT 10;

*Mostramos los 10 primeros productos*

  - SELECT nombre FROM jugadores LIMIT 5, 10;

*Mostramos 10 jugadores, a partir del sexto*

*- El primero sería el número 0*

*Combina bien con ORDER BY ("obtener l@s 10 alumn@s con la nota más alta")*

# Consultas básicas

## Condiciones

- Selección de determinados registros con una condición:
  - SELECT modelo, precio FROM coches WHERE precio < 12000;

*Muestra información de las columnas indicadas (modelo y precio) sólo para las filas de la tabla coches que verifican la condición*

# Consultas básicas

## Condiciones

- Selección de determinados registros con más de una condición
  - SELECT * FROM empleados WHERE edad>18 AND edad<45;

*Muestra los empleados cuya edad está comprendida entre 18 y 45 años*

# Consultas básicas

## Operadores (1)

- SQL admite diferentes operadores:
  - Aritméticos (+ - * / )

```
SELECT Salario * PorcentajeBonificación
FROM Empleados
WHERE YEAR(CURDATE()) – YEAR(Fecha_Contratación) > 10;
```

  - De comparación (= < <= > >= <> !=)

```
WHERE nombre="Alberto“
WHERE precio<5000
```

  - De campos vacíos (IS NULL, IS NOT NULL)

```
WHERE observaciones IS NOT NULL
```

  - Operadores lógicos (AND &&, OR ||, NOT !)

```
WHERE precio <5000 AND potencia >=100
```

(en rojo alternativas válidas)

# Consultas básicas

## Operadores (2)

- Específicos de cadenas de caracteres:
  - LIKE: valores aproximados según un patrón
    - El comodín % indica cualquier número de caracteres
    - El comodín _ indica un carácter

      WHERE nombre LIKE "al%" *(que empiece por al)*

      WHERE nombre LIKE "_a%“ *(que tenga una “a” en la segunda posición)*

      WHERE nombre LIKE "_____a“ *(que tenga seis caracteres, el último“a”)*

NOTAS:

- No se distinguen mayúsculas y minúsculas
- Puedes usar comillas simples ' ' o dobles " "
- Cuidado si copias de Office: usa comillas “bonitas” pero ‘no válidas’.
- Además de LIKE puedes usar los operadores ya estudiados de comparación:

  WHERE nombre="Alberto"

  WHERE nombre>"j%"

# Consultas básicas

## Operadores (3)

- **Operador AND**
  - Las condiciones deben ser verdaderas
    - WHERE NotaExamen>5 AND NotaPráctica>5
- **Operador OR**
  - Basta con que una de las condiciones sea verdadera
    - WHERE NotaExamenOrdinario>5 OR NotaExamenRecuperación>5
- **Operador NOT**
  - La condición no debe cumplirse
    - WHERE Nombre NOT LIKE "a%"

# Consultas básicas

## Operadores (4)

- **Precedencia de operadores**
  - El operador * (multiplicación) tiene más prioridad que + (suma)
    - 2 + 5 * 3 = 17
    - Si quiero romper esa precedencia utilizo paréntesis:
    - (2 + 5) * 3 = 21
    - 2*3 + 5*3 = 21
  - El operador AND tiene más prioridad que OR
    - "Obtener los módulos que tengan menos de 4 horas semanales o más de 6, y además sean de primer curso
    - SELECT * FROM Módulo
      WHERE HorasSemanales <4 OR HorasSemanales>6 AND Curso=1
      - Módulos de menos de 4 horas, o módulos de primer curso y más de 6 horas
    - Soluciones correctas rompiendo la precedencia:
    - SELECT * FROM Módulo
      WHERE (HorasSemanales <4 OR HorasSemanales>6) AND Curso=1
    - SELECT * FROM Módulo
      WHERE HorasSemanales<4 AND Curso=1 OR HorasSemanales>6 AND Curso=1

# Consultas básicas

## Operadores (5)

- De conjuntos de valores (IN, BETWEEN)
  - IN se usa para buscar valores concretos entre un conjunto de posibles valores

```
WHERE género IN ('comedia', 'fantasía');
WHERE tamaño_rueda IN (26, 29);
WHERE tamaño_rueda=26 OR tamaño_rueda=29; (equivalente)
WHERE año NOT IN (2000, 2010); (se puede usar con NOT)
WHERE año <>2000 AND año <>2010; (equivalente)
```

  - BETWEEN (“entre”) se usa para buscar en rangos

```
WHERE precio BETWEEN 5000 AND 6000;
WHERE precio >= 5000 AND precio <= 6000; (equivalente, ojo, sin = no)
WHERE nombre BETWEEN "c%" AND "j%"; (funciona con caracteres)
WHERE edad NOT BETWEEN 30 AND 35; (se puede usar con NOT)
```

# Funciones y operadores

Al plantear las consultas nos irá surgiendo la necesidad de usar diferentes funciones, por ejemplo:

```
SELECT CONCAT(Nombre, " ", Apellido)
FROM Empleados
WHERE YEAR(CURDATE()) – YEAR(Fecha_Contratación) > 10;
```

Lo mejor es consultar la referencia oficial de MySQL, en inglés:

https://dev.mysql.com/doc/refman/8.0/en/sql-function-reference.html

Alternativa: referencia de W3Schools (palabras clave y funciones):

https://www.w3schools.com/sql/sql_ref_keywords.asp

https://www.w3schools.com/sql/sql_ref_mysql.asp

*Sólo se te pedirá conocer los operadores y funciones que aparezcan en los apuntes y en las prácticas. Si lo deseas puedes preparar tu propia guía-resumen, que podrás consultar durante los exámenes.*