



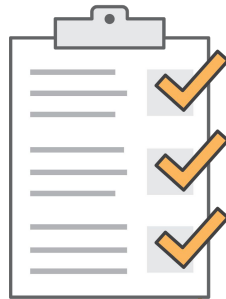
Lenguajes de **gestión** de bases de datos

Structured Query Language

Lenguaje de consulta
estructurada



Estándar SQL



- Descripción del estándar SQL.
- Creación de bases de datos.
- Gestión de registros en tablas.
- Consultas.
- Conversión, generación y manipulación de datos.
- Consultas múltiples.
- Agrupaciones.
- Vistas.
- Funciones avanzadas.

Descripción del estándar

El **SQL** es un lenguaje estándar que sirve para **comunicarse** con fuentes de datos (bases de datos relacionales).



SQL

- El SQL es un lenguaje bastante natural porque se asemeja mucho al lenguaje hablado.
- Se lee de izquierda a derecha
- Está estructurado por secciones (delimitadas por palabras clave como SELECT, FROM, WHERE).
- Las consultas siempre terminan en punto y coma (;) indicando el final de las mismas.

Creación de BD

Data Definition Language o DDL, es el que se encarga de la **modificación de la estructura** de los objetos de la base de datos.



Lenguaje de Definición de Datos DDL

TRUNCATE

DROP

ALTER

CREATE



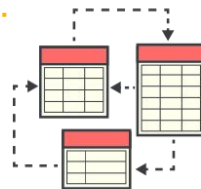
CREATE

1. **CREATE DATABASE** NombreBD;

2. **CREATE TABLE** NombreTabla;

USE NombreBD;

Este comando permite **crear objetos de datos**, como nuevas bases de datos, tablas, vistas y procedimientos almacenados



ALTER

1. **ALTER TABLE** NombreTabla **ADD** NombreCampo datatype;
2. **ALTER TABLE** NombreTabla **DROP** NombreCampo datatype;
3. **ALTER TABLE** NombreTabla **MODIFY** NombreCampo datatype;

Este comando permite **modificar la estructura** de una tabla u objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, etc

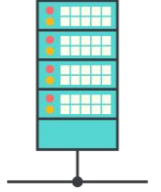


DROP

1. **DROP DATABASE** NombreBD;
2. **DROP TABLE** NombreTabla;

Este comando **elimina** un objeto de la base de datos. Se puede combinar con la sentencia ALTER.

TRUNCATE



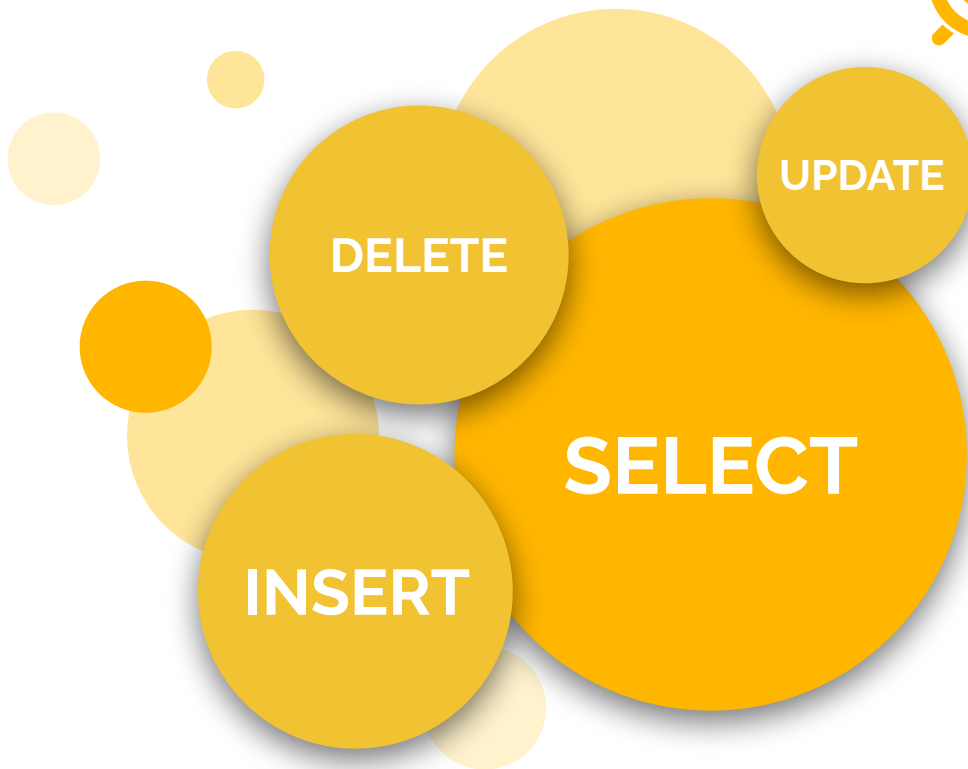
1. TRUNCATE TABLE NombreTabla;

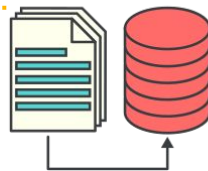
Este comando solo aplica a tablas y su función es **borrar el contenido** completo de la tabla especificada

Gestión de registros

Data Manipulation Language o **DML**, es el que se encarga de la **manipulación de los datos** organizados en la base de datos.

Lenguaje de Manipulación de Datos DML





INSERT

1. **INSERT INTO** NombreTabla (campo1, campo2, ...campoN) **VALUES** (valor1, valor2,...valorN);
2. **INSERT INTO** NombreTabla **VALUES** ('valor1', 'valor2', .valorN);

Agrega uno o más **registros** a una (y sólo una) tabla en una base de datos relacional.



UPDATE

1. **UPDATE** NombreTabla **SET** campo1 = 'valor' **WHERE** campo2 = 'valor2';

Es utilizada para **modificar los valores** de un conjunto de registros existentes en una tabla.



DELETE

1. **DELETE** campo1 **FROM** NombreTabla **WHERE** campo1 = 'valor1';

Borra uno o más **registros** existentes en una tabla



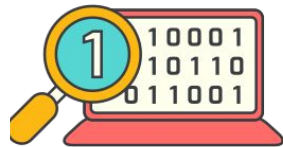
SELECT

1. **SELECT** campo1 **FROM** NombreTabla **WHERE** campo1 = 'valor1';

Selecciona uno o más **registros** existentes en una tabla

Conversión, generación y manipulación de datos

Para convertir un dato de un tipo a otro se cuenta con la **función CAST()**



CAST

1. **CAST**(campo **AS** tipo);

EJEMPLO: **CAST**(2 **AS** CHAR);

Convierte un valor de cualquier tipo a un valor con otro tipo especificado. El tipo de destino puede ser cualquiera de los siguientes tipos: **BINARY, CHAR, DATE, DATETIME, TIME, DECIMAL, SIGNED, UNSIGNED**.

Consultas múltiples

El **SQL** es un lenguaje estándar que sirve para **comunicarse** con fuentes de datos (bases de datos relacionales).



Consultas Multitablas

```
SELECT [ALL | DISTINCT] Tabla1.campo, Tabla2.campo  
FROM Tabla1, Tabla2  
WHERE condición [AND|OR condición]  
ORDER BY Tabla1.campo [ASC | DESC]
```



CROSS JOIN

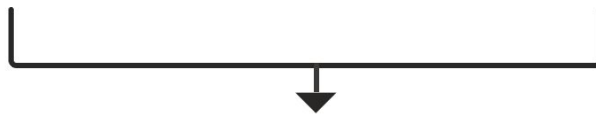
```
SELECT tabla1.campo, tabla2.campo  
FROM tabla1  
CROSS JOIN tabla2;
```

Meals		MealName	DrinkName
Omlet		Omlet	Orange Juice
Fried Egg		Fried Egg	Orange Juice
Sausage		Sausage	Orange Juice
		Omlet	Tea
		Fried Egg	Tea
		Sausage	Tea
		Omlet	Coffee
		Fried Egg	Coffee
		Sausage	Coffee

Realiza la **unión cruzada** de dos tablas, todos los registros de la tabla1 con todos los de la tabla2



Tipos de Combinación



Tipos de Combinación



Solo filas coincidentes de ambas tablas



Solo filas de la tabla izquierda



Solo filas de la tabla derecha



INNER JOIN

```
SELECT t1.campo, t2.campo  
FROM t1  
INNER JOIN t2  
ON t1.fk = t2.id;
```

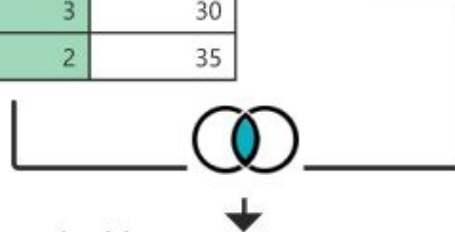
Compara cada fila en la primera t
para encontrar **pares de filas** que

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	2	35

Right Table

ID	Country
3	Panama
4	Spain



Merged Table

Date	CountryID	Units	Country
1/3/2020	3	30	Panama

LEFT JOIN

```
SELECT t1.campo, t2.campo  
FROM t1  
LEFT JOIN t2  
ON t1.fk = t2.id;
```

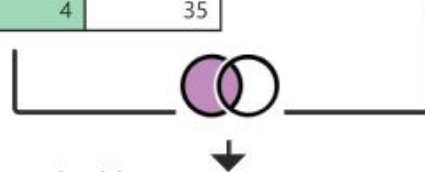
Al unir dos tablas utilizando a LEFT JOIN, se muestra toda la fila de la izquierda y los comunes con la derecha.

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	4	35

Right Table

ID	Country
1	USA
2	Canada
3	Panama



Merged Table

Date	CountryID	Units	Country
1/1/2020	1	40	USA
1/2/2020	1	25	USA
1/3/2020	3	30	Panama
1/4/2020	4	35	null





RIGHT JOIN

```
SELECT t1.campo, t2.campo  
FROM t1  
RIGHT JOIN t2  
ON t1.fk = t2.id;
```

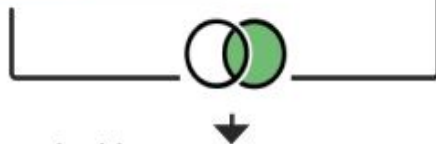
Con RIGHT JOIN, todas las filas de aparecerán en el conjunto de resul

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	4	35

Right Table

ID	Country
3	Panama



Merged Table

Date	CountryID	Units	Country
1/3/2020	3	30	Panama



FULL OUTER JOIN

```
SELECT t1.campo, t2.campo  
FROM t1  
FULL OUTER JOIN t2  
ON t1.fk = t2.id;
```

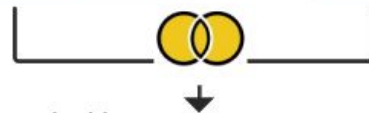
Con **FULL OUTER JOIN**, todas las filas d
aparecerán en el conjunto de resultados

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	2	35

Right Table

ID	Country
1	USA
2	Canada
3	Panama
4	Spain



Merged Table

Date	CountryID	Units	Country
1/1/2020	1	40	USA
1/2/2020	1	25	USA
1/4/2020	2	35	Canada
1/3/2020	3	30	Panama
null	null	null	Spain



LEFT OUTER JOIN

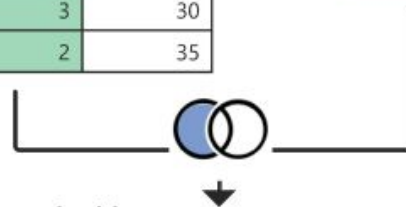
```
SELECT t1.campo, t2.campo
FROM t1
LEFT OUTER JOIN t2
ON t1.fk = t2.id
WHERE t2.id is null
```

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	2	35

Right Table

ID	Country
3	Panama
4	Spain



Merged Table

Date	CountryID	Units	Country
1/1/2020	1	40	null
1/2/2020	1	25	null
1/4/2020	2	35	null



RIGHT OUTER JOIN

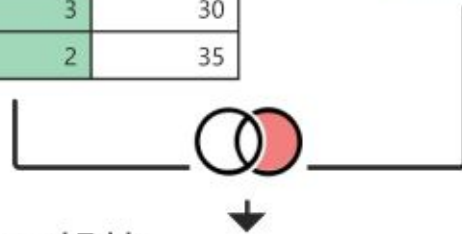
```
SELECT t1.campo, t2.campo  
FROM t1  
RIGHT OUTER JOIN t2  
ON t1.fk = t2.id  
WHERE t1.fk is null
```

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	2	35

Right Table

ID	Country
3	Panama
4	Spain



Merged Table

Date	CountryID	Units	Country
null	null	null	Spain

Agrupaciones

Agrupar consiste en realizar una operación extra sobre los datos consultados para **contar duplicados**, calcular medias, mínimos, máximos, etc.



Consultas **SELECT**

SELECT **[ALL | DISTINCT]** nombre_campo , nombre_campo

FROM nombre_tabla, nombre_tabla

WHERE condición **[AND|OR]** condición]

GROUP BY nombre_campo

HAVING condición **[AND|OR]** condición]

ORDER BY nombre_campo **[ASC | DESC]**

Vistas

Consisten en **consultas de SQL** almacenadas en el SGBD que pueden ser utilizadas como si fueran tablas.



Consultas

View

Las vistas permiten simplificar el realizar **consultas frecuentes** y el manejo de permisos entre los usuarios de la base de datos.

EJEMPLO VIEW

CREATE VIEW Ventas_mes **AS**

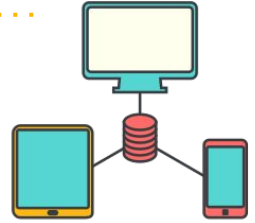
SELECT

Id, SUM(cantidad * precio)

FROM

pedidos

ORDER BY nombre;



Funciones Avanzadas

Las **funciones MySQL más utilizadas**, son las funciones agregadas, funciones de cadenas, funciones de fecha y funciones de flujo de control, etc.



FUNCIONES DE AGREGADO

Realizan un cálculo en un conjunto de valores y devuelven **un solo valor**

FUNCIONES DE AGREGADO



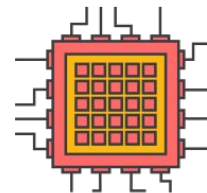
AVG (expresión)	calcula el valor promedio de un conjunto de valores o expresión.
COUNT (expresión)	cuenta el número de filas en una tabla.
INSTR (str, substr)	devuelve la posición de una subcadena dentro de una cadena
SUM (expresión)	calcula la suma de un conjunto de valores o expresión.
MIN (expresión)	encuentra el valor mínimo en un conjunto de valores
MAX (expresión)	encuentra el valor máximo en un conjunto de valores



FUNCIONES DE CADENA

CONCAT (expr1, expr2)	combina dos o más cadenas en una sola cadena.
LENGTH (str)	obtiene la longitud de las cadenas en bytes y en caracteres.
LEFT (str,length);	obtiene la parte izquierda de una cadena con una longitud específica (en número).
REPLACE (campo,old_string,new_string)	busca y reemplaza una subcadena en una cadena.
TRIM (expresión)	elimina los caracteres no deseados de una cadena.

Funciones Matemáticas



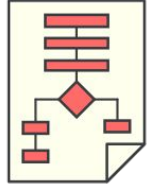
- **ABS:** devuelve el valor absoluto de un número
- **CEIL:** devuelve el valor entero aproximando por debajo
- **FLOOR:** devuelve el valor entero aproximando por arriba
- **MOD:** devuelve el resto
- **ROUND:** redondea un número con decimales
- **TRUNCATE:** trunca un número a un número específico de lugares decimales

FUNCIONES DE **CONTROL DE** FLUJO

- **IF:** devuelve un valor basado en una condición dada.
- **CASE:** devuelve el resultado correspondiente si cumple la condición



EJEMPLO IF



SELECT

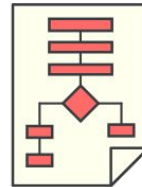
nombre **IF**(edad <= 20, 'Joven', 'Adulto')

FROM

clientes

ORDER BY edad;

EJEMPLO CASE



SELECT

nombre, provincia, comunidad

FROM

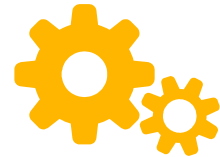
clientes

ORDER BY (CASE

WHEN provincia **IS NULL THEN** comunidad

ELSE provincia

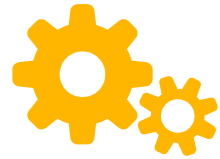
END);



TRIGGERS **DISPARADORES**

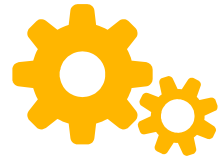
Un disparador (TRIGGER, en inglés) es un **objeto asociado a una tabla** de base de datos que se ejecuta cuando ocurre un evento sobre esa tabla





EJEMPLO DE **DISPARADORES**

```
CREATE TRIGGER nombre_trigger BEFORE INSERT  
ON nombre_tabla  
FOR EACH ROW  
BEGIN  
Instrucción SQL  
END;
```

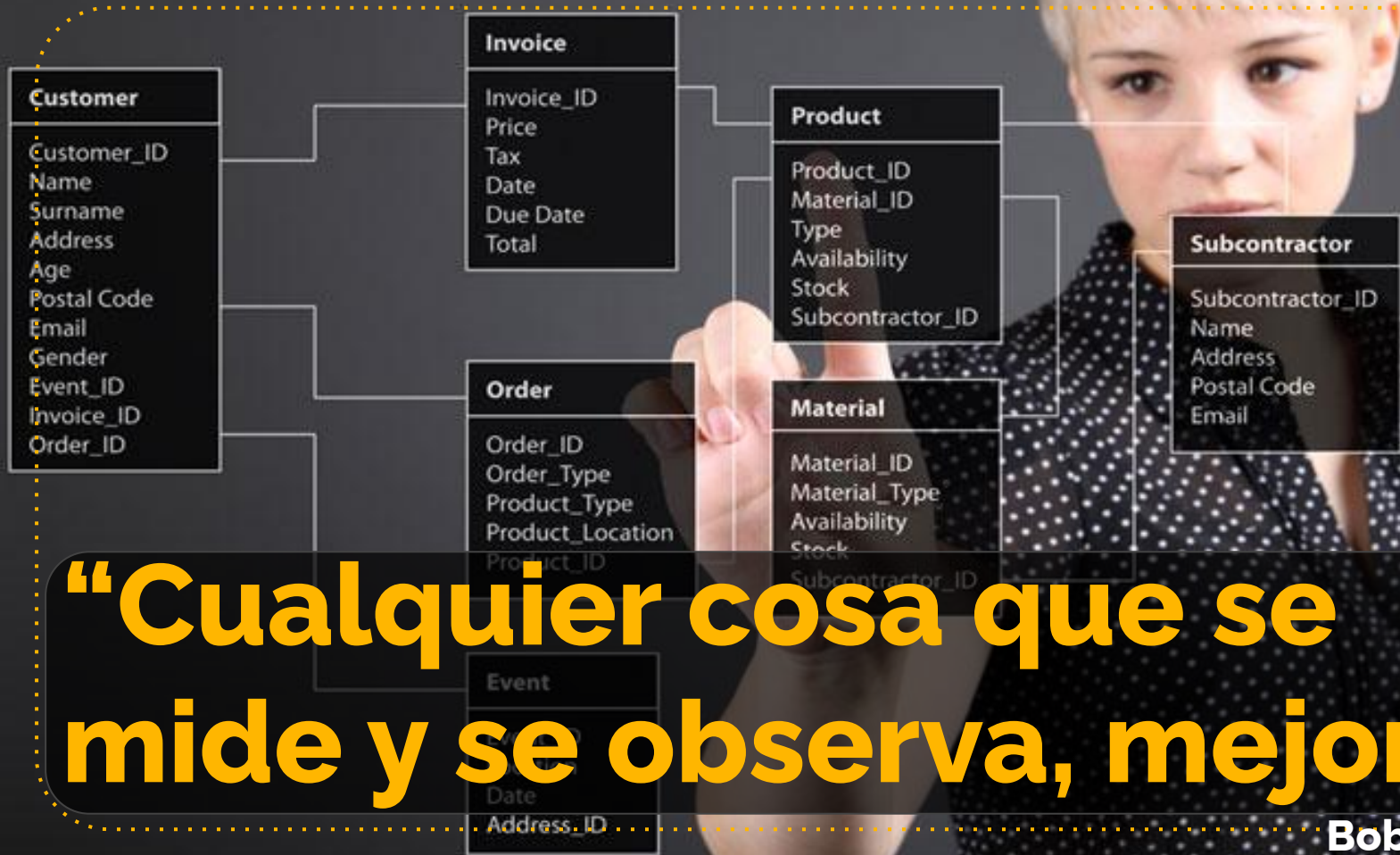


PROCEDURE PROCEDIMIENTOS

Un procedimiento almacenado es un conjunto de comandos **SQL** que pueden almacenarse en el servidor. Utilidades:

- Múltiples aplicaciones cliente
- Más seguros

```
CREATE PROCEDURE nombre_procedimento()  
BEGIN  
    SELECT...;  
END;
```



“Cualquier cosa que se mide y se observa, mejora.”

Bob Parsons



Creditos

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Unsplash](#)



Presentation **design**

This presentation uses the following typographies:

- Titles: Raleway ExtraBold
- Body copy: Raleway Light

You can download the fonts at

<https://www.fontsquirrel.com/fonts/raleway>

You don't need to keep this slide in your presentation. It's only here to serve you as a design guide if you need to create new slides or download the fonts to edit the presentation in PowerPoint®



SlidesCarnival icons are editable shapes.

This means that you can:

- Resize them without losing quality.
- Change fill color and opacity.
- Change line color, width and style.

Isn't that nice? :)

Examples:

