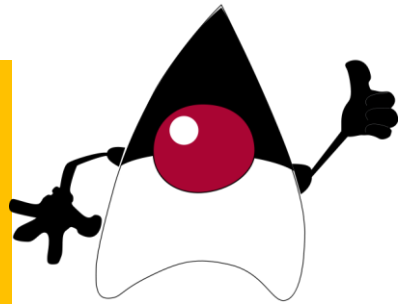
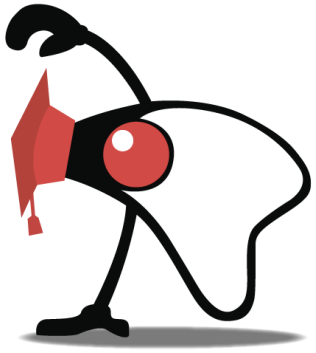
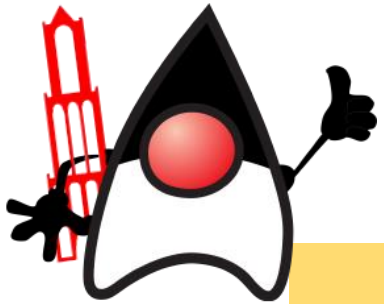
Curso FullStack Python

Codo a Codo 4.0





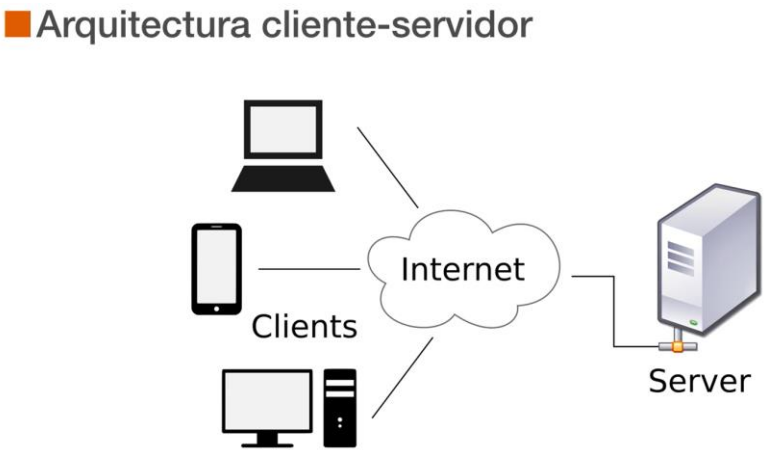


MySQL

Parte 2

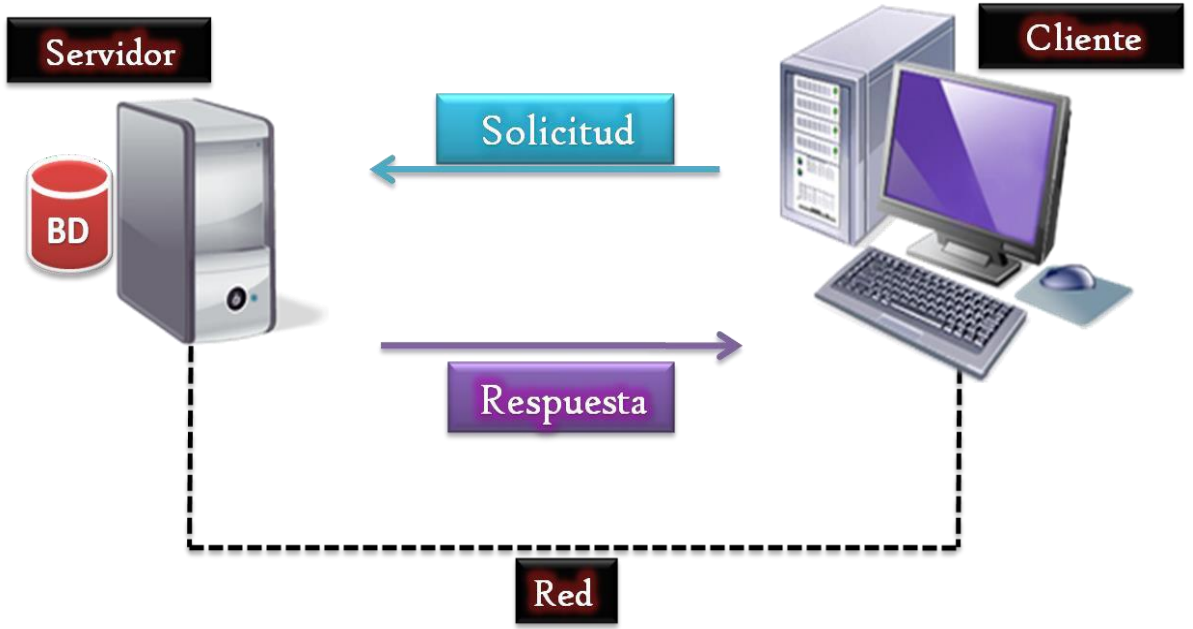
Arquitectura Cliente-Servidor (repaso)

- Es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**.
  - Un cliente realiza peticiones a otro programa.
  - El servidor es quien le da respuesta.



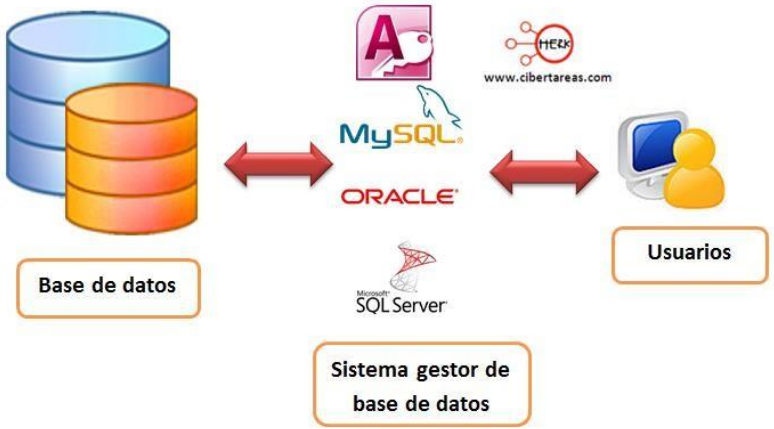
Cliente-Servidor en Bases de Datos

- Las bases de datos en general utilizan la arquitectura Cliente-Servidor para proveer servicios de almacenamiento de información a determinados usuarios (Clientes).



¿Cómo un cliente se conecta a un servidor de BD?

- El software intermediario, entre un usuario y el servidor que provee el servicio de almacenamiento en bases de datos, es conocido como **SGBD** (Sistema Gestor de Bases de Datos).
- A través de los SGBD, los usuarios pueden hacer **CONSULTAS** en lenguaje **SQL** (Lenguaje de Consulta Estructurada, *Structured Query Language* en inglés) para así realizar, por ejemplo, operaciones de lectura de datos.



¿Cómo armar un Servidor de BD? (repaso)

- Para armar un servidor de base de datos se puede utilizar diferentes softwares, entre ellos, distribuciones de Linux, sistemas operativos especializados para bases de datos, servidores virtuales, servidores online, servidores para páginas web, etc.
- De forma experimental, el software que podremos utilizar para armar un servidor de BD es el **XAMPP SERVER**, visto en la presentación anterior. Para más detalles ver tutorial **Instalar XAMPP y MySQL Workbench** en los videos del Aula Virtual.
- El Sistema Gestor de Bases de Datos que utilizaremos será MySQL. Uno de los más utilizados a nivel mundial.

Sentencias DML: Lenguaje de Manipulación de Datos

- La **manipulación** de los datos consiste en la realización de operaciones de *inserción, borrado, modificación y consulta* de la información almacenada en la base de datos. La inserción y el borrado son el resultado de añadir nueva información a la que ya se encontraba almacenada o eliminarla de nuestra base de datos, tomando en cuenta las restricciones marcadas por el DDL y las relaciones entre la nueva información y la antigua. La modificación nos permite alterar esta información, y la consulta nos permite el acceso a la información almacenada en la base de datos siguiendo criterios específicos.
- Las sentencias de lenguaje de manipulación de datos (DML) son utilizadas para gestionar datos dentro de los schemas. Algunos ejemplos:
  - **SELECT:** para obtener datos de una base de datos.
  - **INSERT:** para insertar datos a una tabla.
  - **UPDATE:** para modificar datos existentes dentro de una tabla.
  - **DELETE:** elimina todos los registros de la tabla; no borra los espacios asignados a los registros.
- Son estas sentencias las que nos permitirán más adelante realizar los sistemas denominados CRUD.

CRUD: acrónimo de “Crear, Leer, Actualizar y Borrar” (en inglés *Create, Read, Update and Delete*), que se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

Sentencias SQL

- Las consultas SQL son los “diálogos” o “preguntas” que se generan entre el usuario y el sistema gestor de bases de datos donde se encuentran almacenados ciertos datos.
- Existen **diferentes cláusulas** dentro de las consultas SQL. Las más conocidas son:
- **DE LECTURA**
  - **SELECT:** cláusula utilizada para especificar qué atributo (dato) se pretende obtener.
  - **FROM:** es utilizada en conjunto con el SELECT para especificar desde qué tabla (entidad) se pretende traer el dato.
  - **WHERE:** cláusula para proponer una condición específica que deberá cumplir el dato que se pretende traer (**cláusula no obligatoria**).
- **DE ORDEN Y/O AGRUPAMIENTO**
  - **ORDER BY:** es utilizada para especificar por qué criterio se pretende **ordenar** los “registros” de una tabla.
  - **GROUP BY:** es utilizada para especificar por qué criterio se deben **agrupar** los registros de una tabla.

Ejemplos sentencias SQL

Ej 1 y 2

- **DE LECTURA**
  - Supongamos que tenemos una tabla de empleados y queremos traer el nombre, apellido y fecha de nacimiento de todos aquellos que hayan nacido después del año 1970 inclusive.

EMPLEADO

id_empleado	nombre	apellido	sexo	fecha_nacimiento	salario	puesto
1	Juan	Perez	M	22-09-1960	5000	administrador
2	Mario	Gimenez	M	10-02-1980	3000	secretario
3	Susana	Malcorra	F	11-03-1980	3000	secretaria
4	María	Casan	F	01-02-1965	6000	administrador

```
SELECT nombre, apellido, fecha_nacimiento
FROM empleado
WHERE empleado.fecha_nacimiento >= 01-01-1970;
```

¿Qué atributo/s quiero traer?
¿De dónde lo/s traigo?
¿Qué condición tiene/n que cumplir?

El resultado de esta consulta será:

Mario Gimenez 10-02-1980
Susana Malcorra 11-03-1980

Ejemplos sentencias SQL

- **DE ORDENAMIENTO**
  - Supongamos que tenemos una tabla de empleados y que queremos obtener todos sus elementos y ordenarlos por apellido.

```
SELECT *
FROM empleado
ORDER BY apellido;
```

  - El resultado de esta consulta será traer TODOS los empleados, pero en lugar de estar ordenados por id (identificación del empleado) van a estar ordenados por apellido.
  - El ***** significa que deberá traer TODOS los campos sin distinción.

Ejemplos sentencias SQL

- **DE ORDENAMIENTO: ejemplos**

Orden por una columna (por defecto ascendente)

```
1 • SELECT *
2   FROM escuelas.alumnos
3   ORDER BY nombre;
```

id	id_escuela	legajo	nombre	nota	grado	email
4	1	101	Juan Perez	10	3	
7	1	106	Martín Bossio	10	3	
9	4	1234	Pedro Gómez	6	2	
5	1	105	Pedro González	9	3	

Orden por más de una columna

```
1 • SELECT *
2   FROM escuelas.alumnos
3   ORDER BY id_escuela, nombre;
```

id	id_escuela	legajo	nombre	nota	grado	email
4	1	101	Juan Perez	10	3	
7	1	106	Martín Bo...	10	3	
5	1	105	Pedro Go...	9	3	
6	1	190	Roberto L...	8	3	roberto...
1	2	1000	Ramón M...	8	1	rmesa...

Orden descendente de una columna

```
1 • SELECT *
2   FROM escuelas.alumnos
3   ORDER BY id_escuela, nombre DESC;
```

id	id_escuela	legajo	nombre	nota	grado	email
6	1	190	Roberto L...	8	3	roberto...
5	1	105	Pedro Go...	9	3	
7	1	106	Martín Bo...	10	3	
4	1	101	Juan Perez	10	3	
2	2	1002	Tomás Smith	8	1	

SELECT: LIMIT

Ej 22

- La cláusula SELECT LIMIT se usa para especificar el número de registros a devolver.
- Sintaxis:

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;
```

Operadores de Comparación

Ej 3 a 7

- También conocidos como operadores relacionales, son utilizados en MySQL para comparar igualdades y desigualdades.
- Los operadores de comparación se utilizan con la cláusula WHERE para determinar qué registros seleccionar.

Operador	Descripción
=	Igual
<>	Diferente
!=	Diferente
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que

Operador	Descripción
LIKE	Define un patrón de búsqueda y utiliza % y _
NOT LIKE	Negación de LIKE
IS NULL	Verifica si el Valor es NULL
IS NOT NULL	Verifica si el Valor es diferente de NULL
IN ()	Valores que coinciden en una lista
BETWEEN	Valores en un Rango (incluye los extremos)

Operador LIKE

Ej 8 a 12

- Otra cláusula es la que se utiliza para **comparaciones** con campos de tipo de **cadenas de texto**.
- Esta sentencia se podría utilizar para consultar cuáles son los clientes que viven en una calle que contiene el texto “San Martín”.
- Al colocar el % al comienzo y al final estamos representando un texto que no nos preocupa cómo comienza ni cómo termina, siempre y cuando contenga la/s palabra/s que nos interesa.

```
SELECT * FROM clientes c
    WHERE calle LIKE '%San Martín%'
```

IS NULL / IS NOT NULL

Ej 17 y 18

- Permiten seleccionar registros cuyo valor en un campo sea **null** o **no sea null (not null).** No debemos confundir null con campo en blanco, es un campo que no tiene dato.

```
SELECT *
FROM escuelas.alumnos
WHERE nota IS NULL;
```

id	id_escuela	legajo	nombre	nota	grado	email
7	0	106	Martín Bo...	NULL	3	
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT *
FROM escuelas.alumnos
WHERE nota IS NOT NULL;
```

id	id_escuela	legajo	nombre	nota	grado	email
1	2	1000	Ramón M...	8	1	rmesa...
2	2	1002	Tomás Smith	8	1	
4	1	101	Juan Perez	10	3	
5	1	105	Pedro Go...	9	3	
6	5	190	Roberto L...	8	3	roberto...
8	4	100	Ramiro Es...	3	1	mail@m...
9	4	1234	Pedro Gó...	6	2	

SELECT: uso de ALIAS

- Es recurrente en el desarrollo de consultas o sentencias SQL extensas el uso de ALIAS.
- Esta propiedad es extensible tanto para tablas como para campos y permite renombrar los nombres originales de tablas o campos de manera temporal.
- El uso de ALIAS presenta algunas ventajas:
  - Permite acelerar la escritura de código SQL
  - Mejorar la legibilidad de las sentencias
  - Ocultar/Renombrar los nombres reales de las tablas o campos a usuarios
  - Permite asignar un nombre a una expresión, fórmula o campo calculado
- Ejemplo: renombrar tablas y atributos calculados

```
SELECT V.precio , V.fecha, (V.precio * 1.21) AS precio_con_iva
FROM ventas AS V
```

Operador IN

- Si tenemos una lista larga de posibilidades, escribir todas cláusulas OR encadenadas sería muy tedioso, entonces usamos la **sentencia IN** que funciona de manera equivalente:

```
SELECT codigo FROM productos
    WHERE descripción IN ('Harina' ,'Azúcar' ,'Leche')
```

SELECT: DISTINCT

- La instrucción SELECT DISTINCT se usa para devolver solo valores distintos (diferentes).
- Dentro de una tabla, una columna a menudo contiene muchos valores duplicados; y a veces solo quieres enumerar los diferentes valores (distintos).
- La instrucción SELECT DISTINCT se usa para devolver solo valores distintos (diferentes).
- Sintaxis:

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

MATERIAL COMPLEMENTARIO

- **Resumen SQL.pdf:** resumen con las sentencias SQL básicas más utilizadas.
- **Guía práctica de SQL:** guía de ejercicios con los que podrá poner en práctica los conocimientos de esta Unidad.

  Nota: la guía **NO ES OBLIGATORIA** pero les dará la práctica necesaria para poder trabajar sin problemas con las bases de datos.
- **world.sql:** script para generar la base de datos que deberá utilizar para resolver la guía práctica.
- **der-bd-world.jpg:** DER de la base de datos anteriormente mencionada.
- **W3SCHOOLS – SQL Tutorial:** https://www.w3schools.com/sql/
- **Página Oficial MYSQL:** https://dev.mysql.com/