



# SQL

## Introducción básica

Segunda sesión

Lima, Perú | 02 septiembre 2023

# AGENDA SQL BASICO



Contenido	Semana
1. Introducción a SQL: Bases de datos, Modelo Relacional, SQL, PostgreSQL, Instalación de PostgreSQL.	1era semana
2. Lenguaje de Definición de Datos (DDL): DDL, creación de BD, eliminación de BD, Creación de una Tabla, Restricciones, Tipos de datos, Modificar Tabla, Integridad referencial, Primeras consultas	
3. Lenguaje de Modificación de Datos (DML) monotabla: DML, Comando SELECT, Operadores, Sentencias UPDATE-DELETE, Otros Comandos, Funciones de Agregado, GROUP BY, HAVING.	2da semana
4. DML multitabla: Operaciones entre tablas (intersección, unión, diferencia, INNER JOIN, OUTER, LEFT, RIGHT JOIN, Subconsultas.	



03

>>>>> LENGUAJE DE MODIFICACIÓN DE  
DATOS (DLM) MONOTABLA

PostgreSQL



# Contenido

I

## DML

Lenguaje de  
modificación de datos

II

## Comando SELECT

Recuperar información

III

## Operadores

Operadores de comparación, operador  
LIKE, operadores lógicos,  
BETWEEN, IN, NOT, NULL

IV

## Otros Comandos

DISTINCT, ORDER BY,  
TOP

# Contenido

V

## Funciones de agregado

SUM, AVG, MIN, MAX,  
COUNT.

VI

## GROUP BY

Agrupar las filas.

VII

## HAVING

Condicional de  
funciones agregadas.

# I. LENGUAJE DE MODIFICACIÓN DE DATOS - DML



## ¿Qué es?

Conjunto de instrucciones que permiten realizar operaciones como la inserción, actualización y eliminación de datos en tablas específicas dentro de una base de datos relacional.

Principales comandos: SELECT, INSERT, UPDATE, DELETE.

## II. Comando SELECT

### Utilidad

Permite presentar información (datos) total o parcial de la base de datos.

### Sintaxis

Colocar el nombre de las columnas que desea obtener como resultado. Si son varias columnas se separan con coma (,). Si se desean todas las columnas solo se coloca (\*).

Query	Query History
1	
2	<b>SELECT</b> <nombrecolumnas> <b>FROM</b> <nombretabla>
3	<b>WHERE</b> <condiciones>;
4	

Se pueden agregar condiciones para filtrar solo aquellas filas que las cumplen.

Colocar el nombre de la tabla a la cual pertenecen las columnas mencionadas anteriormente.

# III. Operadores

## Operadores de Comparación

## Sintaxis

Colocar el nombre de las columnas que desea obtener como resultado. Si son varias columnas se separan con coma (,). Si se desean todas las columnas solo se coloca (\*).

Query Query History

```
1  
2 SELECT <nombrecolumnas> FROM <nombretabla>  
3 WHERE <condiciones>;  
4
```

Se pueden agregar condiciones para filtrar solo aquellas filas que las cumplen.

Colocar el nombre de la tabla a la cual pertenecen las columnas mencionadas anteriormente.



# III. Operadores

La sentencia WHERE nos permite utilizar estos operadores para utilizar filtros.

## Operadores de comparación

Podemos utilizar =, <, >, <>

## Operador LIKE

Es un operador de comparación de cadenas/formato texto (LIKE).

## Operador Between

Es un operador de comparación de cadenas/formato texto (LIKE).

## IN, NOT, NULL

Es un operador de comparación de cadenas/formato texto (LIKE).

## IV. Otros Comandos

**DISTINCT, ORDER  
BY, TOP**



# 04

## DLM MULTITABLA

Juntando mi informacion

# Conceptualización

Seguimos lógica de conjuntos

`SELECT * FROM a  
INNER JOIN b ON a.key = b.key`



`SELECT * FROM a  
LEFT JOIN b ON a.key = b.key`



`SELECT * FROM a  
RIGHT JOIN b ON a.key = b.key`



POSTGRESQL  
JOINS



`SELECT * FROM a  
LEFT JOIN b ON a.key = b.key  
WHERE b.key IS NULL`



`SELECT * FROM a  
RIGHT JOIN b ON a.key = b.key  
WHERE a.key IS NULL`



`SELECT * FROM a  
FULL JOIN b ON a.key = b.key`



`SELECT * FROM a  
FULL JOIN b ON a.key = b.key  
WHERE a.key IS NULL OR b.key IS NULL`



# Práctica

Lista de empleados que están debajo de la media de faltas por enfermedad.

**JOIN**

Verificar si todos los empleados se encuentran en la lista de contactos

LEFT Y OUTER

# Práctica

## JOIN

```
SELECT asset.SickLeaveHours, asset.MaritalStatus, pay.Rate
FROM humanresources.Employee as asset
JOIN humanresources.EmployeePayHistory as pay
ON pay.businessentityid = asset.businessentityid
WHERE asset.SickLeaveHours > 10
ORDER BY pay.Rate ASC
```

```
SELECT asset.SickLeaveHours, asset.MaritalStatus, pay.rate
FROM humanresources.Employee as asset
JOIN humanresources.EmployeePayHistory as pay
ON pay.businessentityid = asset.businessentityid
WHERE asset.SickLeaveHours > (
    SELECT AVG(SickLeaveHours)
    FROM humanresources.Employee
);
```

# Práctica

## LEFT & OUTER

```
SELECT *  
FROM person.Person as contact  
FULL JOIN HumanResources.Employee as employee  
ON contact.businessentityid = employee.businessentityid  
WHERE contact.businessentityid IS NULL or  
employee.businessentityid IS NULL
```



Gracias