SQL

Introducción básica

Lima, Perú | 19 Agosto 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.	

01

INTRODUCCION A SQL

PostgreSQL y el lenguaje estructurado de consulta

Contenido

I

Bases de datos

¿Qué son las bases de datos?

II

Modelo Relacional

Definición, tablas, relaciones.

III

SQL

Definición y sistemas de gestión de bases de datos.

IV

PosgreSQL

Descripción e instalación

I. Bases de datos

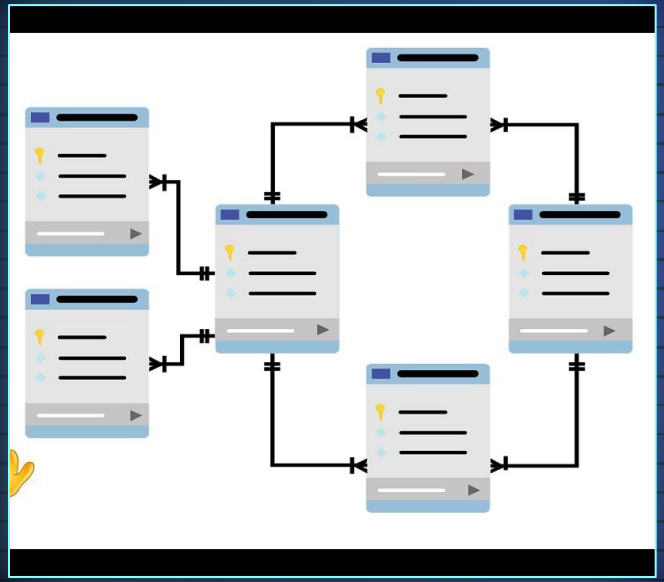
¿Qué es?

Son conjuntos de datos pertenecientes a un mismo contexto y almacenados para un uso posterior de manera organizada.

Pueden ser manipuladas a través de un Sistema Gestor de Bases de Datos (SGBD).

Existen muchos tipos, en este caso nos centraremos en las bases de datos relacionales.

II. Modelo Relacional



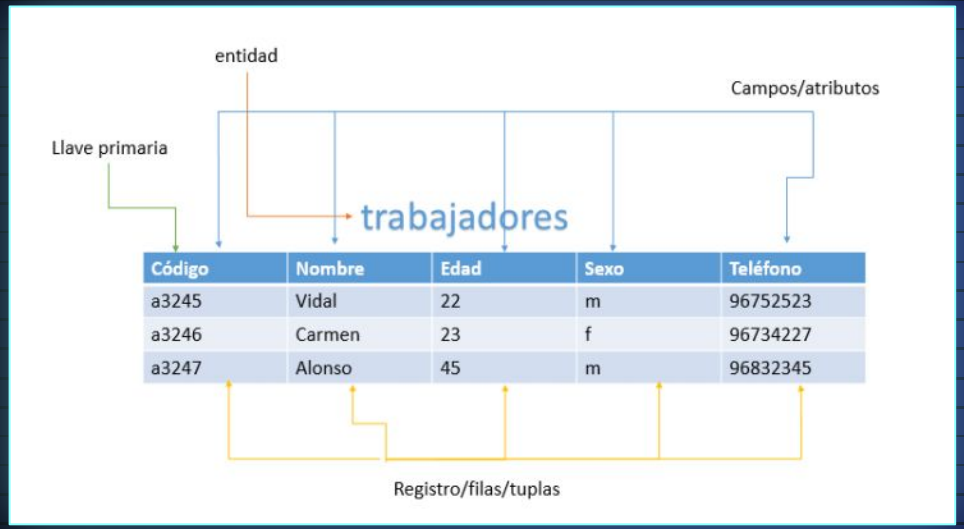
¿Qué es?

Es un enfoque para organizar y gestionar datos en bases de datos.

Elementos clave:

- Tablas
- Relaciones.

II. Modelo Relacional



Tablas

Cada tabla representa una entidad: objetos, eventos, transacciones sobre los que se quiere almacenar datos.

Cada tabla posee:

- Atributos (columnas)
- Instancias o registros (filas)

Una de las columnas de tabla actuará como un identificador único e irrepetible para cada instancia. A esta se le denominará “Llave primaria”

II. Modelo Relacional

Relaciones

Las relaciones son el nexo que une a las tablas y les da sentido lógico.

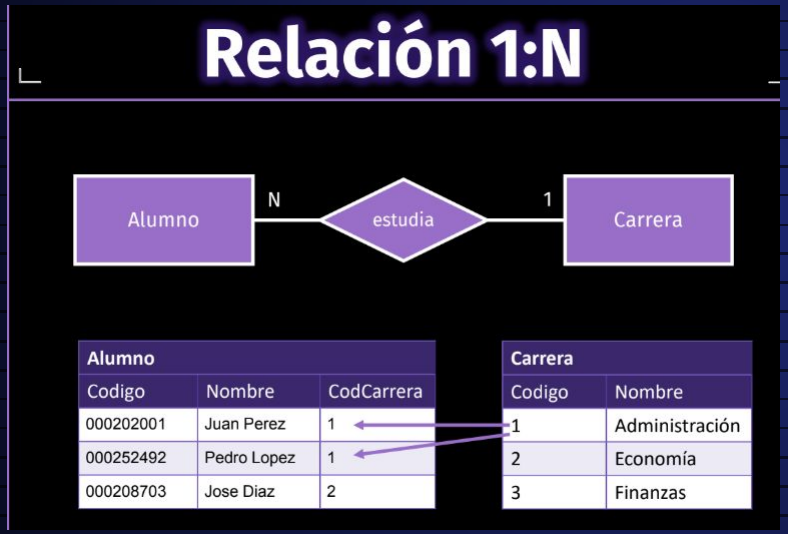
Pueden ser:

- **Uno a uno (1:1).**
- Uno a muchos (1:N)
- Muchos a muchos (N:M)

Las relaciones se establecen a partir de la inclusión de una columna que proviene de otra tabla: la llave foránea.

II. Modelo Relacional

Relaciones



Alumno		
Codigo	Nombre	CodCarrera
000202001	Juan Perez	1
000252492	Pedro Lopez	1
000208703	Jose Diaz	2

Carrera	
Codigo	Nombre
1	Administración
2	Economía
3	Finanzas

Las relaciones son el nexo que une a las tablas y les da sentido lógico.

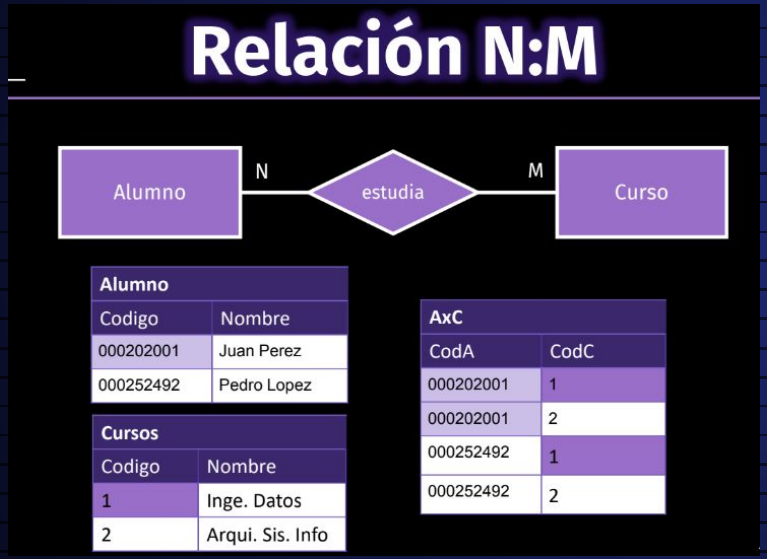
Pueden ser:

- Uno a uno (1:1).
- **Uno a muchos (1:N)**
- Muchos a muchos (N:M)

Las relaciones se establecen a partir de la inclusión de una columna que proviene de otra tabla: la llave foránea.

II. Modelo Relacional

Relaciones



Alumno

Codigo	Nombre
000202001	Juan Perez
000252492	Pedro Lopez

Cursos

Codigo	Nombre
1	Inge. Datos
2	Arqui. Sis. Info

AxC

CodA	CodC
000202001	1
000202001	2
000252492	1
000252492	2

Las relaciones son el nexo que une a las tablas y les da sentido lógico.

Pueden ser:

- Uno a uno (1:1).
- Uno a muchos (1:N)
- Muchos a muchos (N:M)

Las relaciones se establecen a partir de la inclusión de una columna que proviene de otra tabla: la llave foránea.

III. SQL

¿Qué es?

SQL (Structured Query Language) es el lenguaje utilizado para interactuar con bases de datos relacionales.

Con SQL, puedes realizar operaciones como consultar, insertar, actualizar y eliminar datos en las tablas.

Las consultas SQL te permiten recuperar información específica, realizar cálculos, combinar datos de diferentes tablas y mucho más.

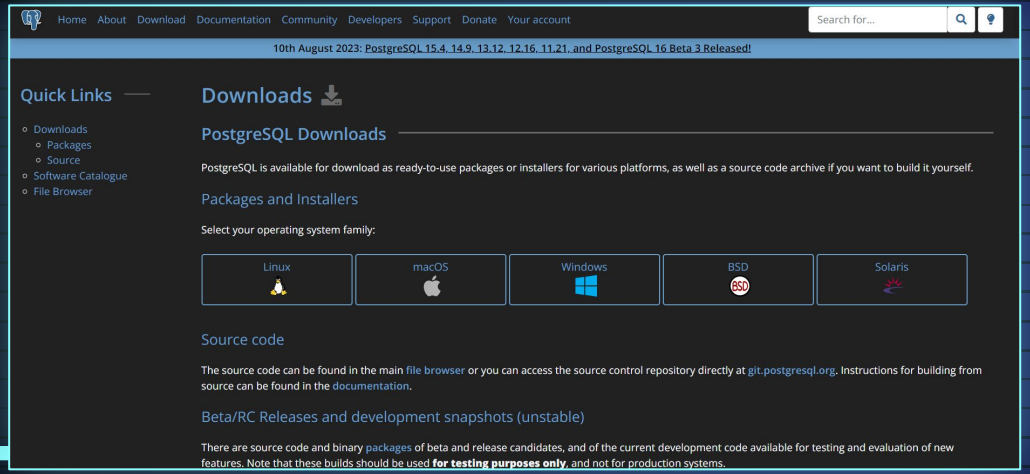
III. PosgreSQL

¿Qué es?

PostgreSQL es un sistema de gestión de bases de datos relacionales (SGBDR) de código abierto.

III. PosgreSQL

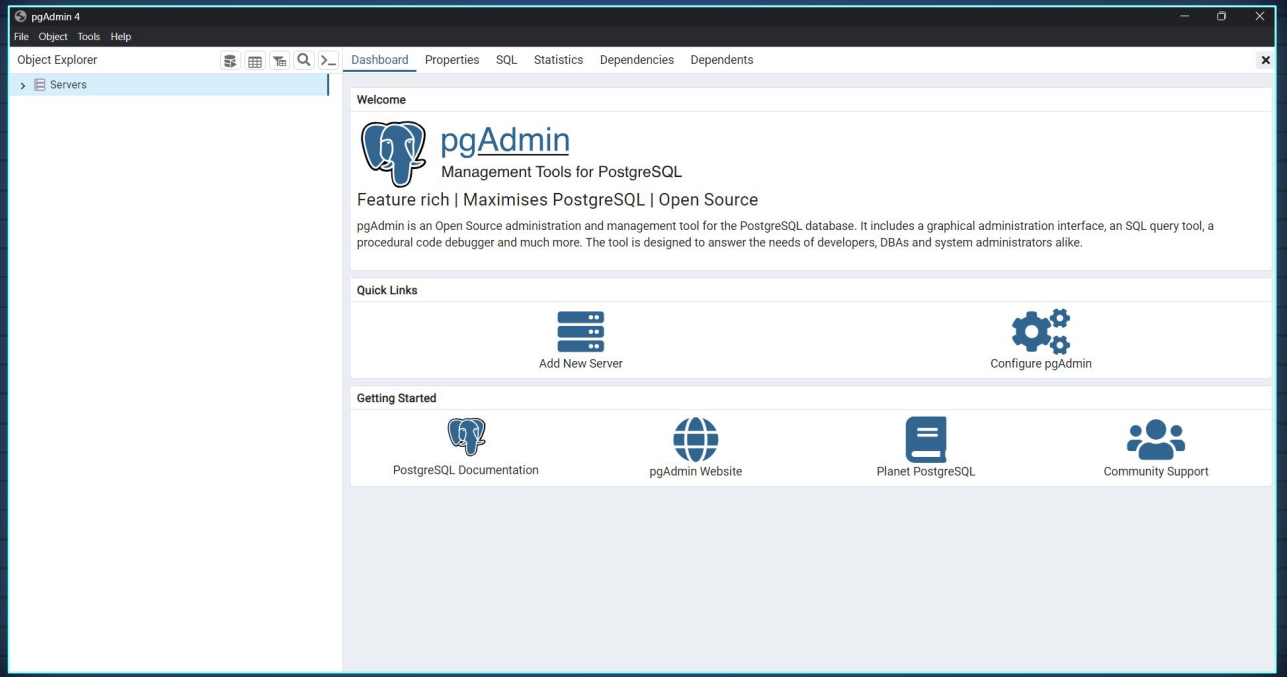
Tutorial



Instalar PostgreSQL

https://www.youtube.com/watch?v=jxIEDKzGrOs

III. PosgreSQL



02

ESTRUCTURA RELACIONAL

Bases de datos y tablas en el DDL

OUT OF THE BOX

Ahora que?

PostgreSQL esta instalado, pero ... Donde pongo mis datos?

Database

Utilizamos el espacio para consultas por primera vez.

Usuarios

Solo yo tendré acceso? Debemos verificar

Tablas

Hora de crear nuestro modelo!

TIPOS DE DATOS

De manera similar a otros lenguajes digitales, SQL utiliza tipos de datos para definir la memoria que usaran cada uno de nuestros casos dentro de las bases de datos.

Compatibility

The following types (or spellings thereof) are specified by SQL: `bigint`, `bit`, `bit varying`, `boolean`, `char`, `character varying`, `character`, `varchar`, `date`, `double precision`, `integer`, `interval`, `numeric`, `decimal`, `real`, `smallint`, `time` (with or without time zone), `timestamp` (with or without time zone), `xml`.

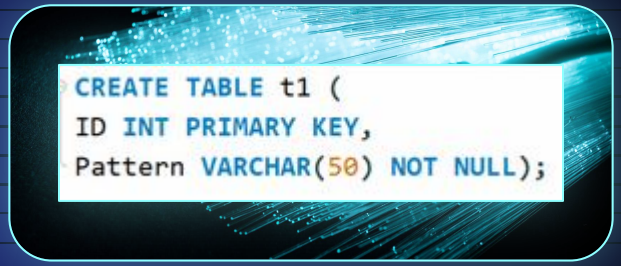
Escoger nuestros tipos ideales?

- Podemos forzar la inclusión de datos en tipos que no parecen “correctos”, siempre y cuando funcionen para nuestras operaciones

En lugar de perder tiempo escogiendo el tipo perfecto, a veces basta con escoger uno “suficiente”

MANEJANDO NUESTRA BASE

CREATE en SQL sirve para generar Usuarios, Tablas e incluso Bases de datos.



CREATE

```
UPDATE Customers
SET ContactName='Juan'
WHERE Country='Mexico';
```

UPDATE

ALTER y UPDATE son declaraciones utilizada para modificar los datos o atributos de Usuarios, Tablas y Bases de datos .

```
ALTER TABLE tableName

ADD columnName columnDefinition;
```

MANEJANDO NUESTRA BASE

Si queremos eliminar algún valor, usuario, tabla o base de datos usamos DELETE

```
DELETE FROM Customers
WHERE CustomerName='Alfreds Futterkiste';
```

DELETE

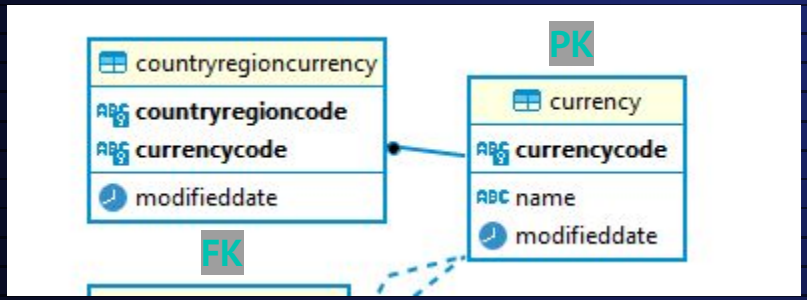
```
UNIQUE,
PRIMARY KEY,
ON DELETE RESTRICT
ON DELETE CASCADE
FOREIGN KEY (b, c) REFERENCES other_table (c1, c2)
```

RESTRICCIONES

Cualquiera de estos “statements” o declaraciones están enmarcados dentro de definidas restricciones.

RESTRICCIONES: INTEGRIDAD REFERENCIAL

La base de datos funciona a partir de “referencias” entre unas y otras. Estas son clave para entender las ventajas y desventajas de SQL y constituyen el origen de la integridad referencial.



Primary Key

Valor único para cada caso dentro de una tabla, identifica la “fila”

Foreign Key

Referencia a una llave primaria en otro lugar, creando una referencia entre tablas

EJERCICIO: Verificando la integridad

Usemos la base AdventureWorks (https://github.com/NorfolkDataSci/adventure-works-postgres) para nuestro ejercicio.

1. **Crear** la base de datos usando CREATE <database>;
2. **Crear** un usuario para la base de datos usando CREATE <user>;
3. **Actualizar** el usuario con permisos de administrador;
4. **Importar** la estructura de la BD*;

Pregunta para la casa: Al utilizar psql para importar la estructura de Adventureworks se generan esquemas incompletos (hr, pe, pr, etc.). Cual es la causa de esto? Cómo solucionamos el problema?

03

PRIMERAS CONSULTAS

DML y statements principales

Estructura básica de la consulta

Los pasos lógicos de cualquier “query” o consulta en SQL son siempre los mismos

SELECT

Qué es lo que quiero?
Debo definir el **atributo** que quiero extraer.

FROM

Dónde está lo que quiero?
Debo definir la tabla donde esta mi consulta

WHERE

Como es lo que quiero?
Puedo definir los **casos** que quiero extraer.

SELECT: El corazón de la consulta

Se emplea para seleccionar los datos de la base de datos relacional.

SELECT

Qué es lo que quiero?

- Quiero las columnas que aparecen en mis resultados de consulta

Pero, al definir el **atributo** que quiero extraer también me doy cuenta que existen en **una tabla**.

- Por ende, SELECT funciona solo en conjunto con FROM

FROM: El origen de nuestra tabla

Nuestra consulta se origina de una tabla en particular...

FROM

Dónde está lo que quiero?

- Nuestro SELECT puede contener muchas tablas, debemos escoger una cómo la central para colocarla en la declaración FROM

La tabla **central** es (usualmente) aquella de la que surgen la mayor cantidad de relaciones de nuestra consulta..

WHERE: El origen de nuestra tabla

Nuestra consulta se origina de una tabla en particular...

WHERE

Como es lo que quiero?
Puedo definir los **casos** que quiero extraer.

Dónde está lo que quiero?

- Nuestro SELECT puede contener muchas tablas, debemos escoger una cómo la central para colocarla en la declaración FROM

La tabla **central** es (usualmente) aquella de la que surgen la mayor cantidad de relaciones de nuestra consulta..

HASTA LA PROXIMA