BASES DE DATOS

Clase 13 - Lenguaje de consulta SQL

Profesora: Erika Gutiérrez Beltrán

Tema 2: Bases de datos relacionales

Objetivo

Transformar los modelos conceptuales a lógicos. Construir bases de datos para datos estructurados

Temario

- Utilización de los modelos en el mundo real ✅
- Transformación del modelado a la base de datos ✅
- Clases de entidades ✅
- Modelo relacional, restricciones de integridad y normalización ✅
- Lenguaje de consulta formal
- Lenguaje de consulta SQL ✅
- Arquitectura de los sistemas gestores de Bases de datos, OLTP, OLAP

Logros

Capturar datos desde una pantalla de usuario real y almacenarlos en una base de datos. Manipulación de estos datos.

Comandos básicos aprendidos hasta el momento

- Crear base de datos **(Recuerda reemplazar siempre los valores, lo que aquí se muestran son de guía o ejemplo)**

```
CREATE DATABASE "EafitMovie";
```

```
CREATE DATABASE "EafitMovie"
WITH
  ENCODING = 'UTF8'
  OWNER = postgres
  CONNECTION LIMIT = 100;
```

- Crear un esquema (un espacio de nombre que contiene objetos de base de datos)

```
CREATE SCHEMA estudiantes;
```

Comandos básicos aprendidos hasta el momento

- Definir esquema de trabajo

```
SET search_path TO estudiantes, public;
```

-Dar privilegios a un usuario sobre un esquema

```
GRANT CREATE ON SCHEMA estudiantes TO postgres;
```

Comandos básicos aprendidos hasta el momento

- Crear roles (usuarios)

```
CREATE USER pepe WITH ENCRYPTED PASSWORD 'admin123';
```

```
CREATE ROLE prueba
SUPERUSER
LOGIN
PASSWORD 'admin123';
```

- Asignar permisos a los usuarios

```
GRANT ALL
ON ALL TABLES
IN SCHEMA "public"
TO prueba;
```

Comandos básicos aprendidos hasta el momento

- \du para listar todos los usuarios y sus roles (los roles otorgan privilegios a los usuarios)

```
                                   List of roles
 Role name |                         Attributes                                   | Member of
-----------+------------------------------------------------------------------------+-----------
 erika     | Superuser                                                              | {}
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS             | {}
```

- Ver información de los usuarios creados en la base de datos

```sql
SELECT * FROM pg_catalog.pg_user;
SELECT rolname, rolpassword FROM pg_authid;
SHOW password_encryption;
```

Comandos básicos aprendidos hasta el momento

- Crear tablas con validación de si ya existe o ha sido creada previamente

```
CREATE TABLE IF NOT EXISTS "PeliculasRentadas" (
    id_usuario SERIAL,
    id_pelicula SERIAL,
    fecha_prestamo DATE,
    fecha_devolucion DATE,
    estado BOOLEAN
);
```

Comandos básicos aprendidos hasta el momento

- Crear tablas

```sql
CREATE TABLE "Usuario" (
    id SERIAL PRIMARY KEY,
    nombre_completo VARCHAR(20),
    edad INT
);
```

- Alterar tablas (agregar columnas)

```sql
ALTER TABLE "Usuario" ADD COLUMN fecha_nacimiento DATE;
```

Comando básicos aprendidos hasta el momento

- Alterar tablas (eliminar columnas)

```sql
ALTER TABLE "Usuario" DROP COLUMN edad;
```

- Alterar tablas (renombrar columnas)

```sql
ALTER TABLE "Usuario" RENAME COLUMN nombre_completo TO nombre;
```

Comandos básicos aprendidos hasta el momento

- Alterar tablas (agregar validaciones)

```
ALTER TABLE "Usuario" ADD CHECK (edad >= 18);
```

- Alterar tablas (añadir clave primaria)

```
ALTER TABLE "Usuario" ADD PRIMARY KEY (id);
```

Comandos básicos aprendidos hasta el momento

- Alterar tablas (agregar llave foránea)

```
ALTER TABLE "Usuario" ADD CONSTRAINT "fk_direccion" FOREIGN KEY (id_direccion) REFERENCES "Direccion" (id);
```

- Alterar tablas (cambiar nombre tabla)

```
ALTER TABLE "Usuario" RENAME TO nuevo_usuario;
```

Comandos básicos aprendidos hasta el momento

- Agregar constraint:

UNIQUE

```
CREATE UNIQUE INDEX CONCURRENTLY correo_unique
ON "Usuario" (correo_electronico);

ALTER TABLE "Usuario" ADD CONSTRAINT correo_unique_id UNIQUE USING INDEX correo_unique;
```

NOT NULL

```
ALTER TABLE "Usuario"
ALTER COLUMN correo_electronico SET NOT NULL;
```

Comandos básicos aprendidos hasta el momento

- Insertar datos a la tabla

```
INSERT INTO "Usuario" (nombre, fecha_nacimiento, edad, correo_electronico)
VALUES ('Pepe Perez', '2000/06/26', 22, 'erika@hotmail.com')
```

- Actualizar un registro de la tabla

```
UPDATE "Usuario"
SET correo_electronico = 'pepeperez@hotmail.com'
WHERE id = 1;
```

Comandos básicos aprendidos hasta el momento

- Seleccionar los registros de la tabla

```sql
SELECT nombre, fecha_nacimiento, edad FROM "Usuario" WHERE id = 1;
```

- Eliminar un registro de una tabla

```sql
DELETE FROM "Usuario" WHERE id = 1;
```

Propiedades ACID

A: Atomicidad, los cambios sobre los datos se realizan en su totalidad o los datos no son modificados si hay algún error. Esta propiedad garantiza que todas las transacciones se confirmen de manera exitosa.

C: Consistencia, cualquier cambio realizado sobre un registro o dato debe conducir a un estado válido de acuerdo a las restricciones.

I: Aislamiento – isolation. Un acambio sobre un dato o registro no debe afectar a otros.

D: Durabilidad, una vez ejecutado el cambio debe mantenerse en el tiempo.

Modelos semánticos de BD

Representación de datos de una manera lógica específica, lo cual permite dar contexto y sentido.

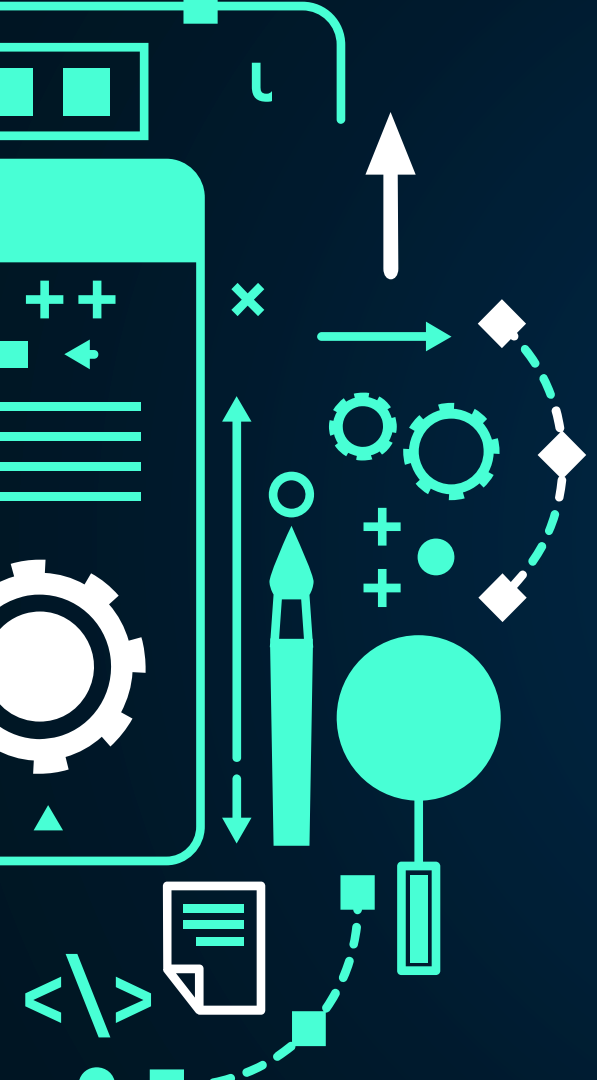
El modelo semántico tiene la siguiente estructura:

- Clasificación
- Agregación
- Generalización.

Los modelos semánticos dan significado a los datos independientemente de su implementación

REFERENCIAS

- El lenguaje SQL: https://www.dataprix.com/files/UOC_OpenSource_El_lenguaje_SQL.pdf
- Libro: Fundamentos de base de datos 5 edición, Silberschatz
- Tutorial de PostgreSQL: https://www.postgresqltutorial.com/
- Roles de bases de datos: https://www.ibm.com/docs/es/data-studio/4.1.1?topic=management-database-roles
- Propiedades de las transacciones: https://www.ibm.com/docs/es/iis/11.5?topic=transactions-transaction-properties



Gracias!