Taller 1 - SQL - BDD

1. Configuración del Entorno con Docker

Objetivo: Crear contenedores para PostgreSQL y pgAdmin.

Archivos clave:

- **Dockerfile**:

```
FROM postgres:14.1-alpine
LABEL author="BDD-Fiuba"
COPY *.sql /docker-entrypoint-initdb.d/  # Copia scripts SQL al contene
dor
```

- **docker-compose.yaml**:

```
services:
  postgres:
    image: postgres:14
    environment:
      POSTGRES_DB: bdd_db
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: admin123
    ports: "5433:5432"  # Acceso: localhost:5433
    volumes: ./data/postgres:/var/lib/postgresql/data  # Persistencia de d
atos

  pgadmin:
    image: dpage/pgadmin4:7.5
    environment:
      PGADMIN_DEFAULT_EMAIL: admin@gmail.com
      PGADMIN_DEFAULT_PASSWORD: admin123
    ports: "5050:80"  # Acceso: <http://localhost:5050>
```

Comandos útiles:

- Iniciar contenedores: `docker-compose up -d`

- Detener contenedores: `docker-compose down`
- Ver contenedores activos: `docker ps -a`

Notas:

- Los `volumes` evitan la pérdida de datos al reiniciar contenedores.
- pgAdmin es una interfaz gráfica para administrar bases de datos.

---

2. Creación de la Base de Datos "mundial"

Pasos:

1. Acceder a pgAdmin: http://localhost:5050 (usuario: `admin@gmail.com`, contraseña: `admin123`).
2. Crear una nueva conexión al servidor PostgreSQL:
   - Host: `postgres` (nombre del servicio en Docker) o `localhost`.
   - Puerto: `5432` (interno del contenedor).
3. Crear la base de datos "mundial":

```
CREATE DATABASE mundial;
```

Conceptos clave:

- **Esquemas**: Divisiones lógicas dentro de una base de datos (por defecto: `public`).
- **Bases vs Esquemas**:
  - Una conexión se hace a una **base**, pero puede usar múltiples **esquemas**.
  - Tablas en bases diferentes no se ven entre sí.

---

3. Creación de Tablas

Tablas requeridas:

- **teams**:

```
DROP TABLE IF EXISTS teams;  -- Elimina la tabla si existe
CREATE TABLE teams (
```

```
    team VARCHAR(50) PRIMARY KEY,
    players_used INT,
    avg_age NUMERIC(3,1),
    possession NUMERIC(3,1),
    games INT,
    goals INT,
    assists INT,
    cards_yellow INT,
    cards_red INT
);
```

- **matches**:

```
DROP TABLE IF EXISTS matches;
CREATE TABLE matches (
    team1 VARCHAR(50),
    team2 VARCHAR(50),
    goals_team1 INT,
    goals_team2 INT,
    stage VARCHAR(50)
);
```

Recomendaciones:

- Usar indentación y comentarios (`-`) para mejorar legibilidad.
- Ejecutar scripts desde pgAdmin o la consola de PostgreSQL (`psql`).

---

4. Manipulación de Datos

a) Inserción manual:

```
INSERT INTO teams (team, players_used, avg_age, possession, games, goa
ls, assists, cards_yellow, cards_red)
VALUES ('ARGENTINA', 24, 28.4, 57.4, 7, 15, 8, 17, 0);
```

b) Eliminación:

```
DELETE FROM teams WHERE team = 'ARGENTINA';
```

c) Carga masiva desde CSV:

```
COPY teams FROM '/ruta/teams.csv' DELIMITER ';' CSV HEADER;
COPY matches FROM '/ruta/matches.csv' DELIMITER ';' CSV HEADER;
```

Nota:

- `CSV HEADER` ignora la primera línea (encabezados).
- Asegurar que los archivos CSV estén en la ruta correcta dentro del contenedor.

---

5. Exportación de Datos

a) Exportar a CSV:

```
COPY teams TO '/ruta/teams_backup.csv' DELIMITER ';' CSV HEADER;
```

b) Generar SQL dump (backup completo):

```
docker exec -it bdd_postgres_db pg_dump -U admin -d mundial > mundial_backup.sql
```

---

6. Conceptos Adicionales

Comando COPY:

- Permite importar/exportar datos entre tablas y archivos.
- Sintaxis:

  ```
  COPY tabla FROM/TO 'archivo' [FORMAT] [DELIMITER] [HEADER];
  ```

Participación Total vs Parcial:

- **Total**: Una entidad **debe** estar relacionada (ej: una ronda no existe sin equipos).
- **Parcial**: Una entidad puede existir sin relación (ej: un jugador sin equipo).

---

7. Ejercicio Práctico

Cargar datos del Mundial 2022:

1. Crear tablas `teams` y `matches`.
2. Usar `COPY` para cargar `teams.csv` y `matches.csv`.
3. Verificar datos con `View Data` en pgAdmin.

Errores comunes:

- Identación incorrecta en Docker o SQL.
- Rutas de archivos mal especificadas en `COPY`.

Referencias:

- Documentación de PostgreSQL
- Ejemplos de SQL