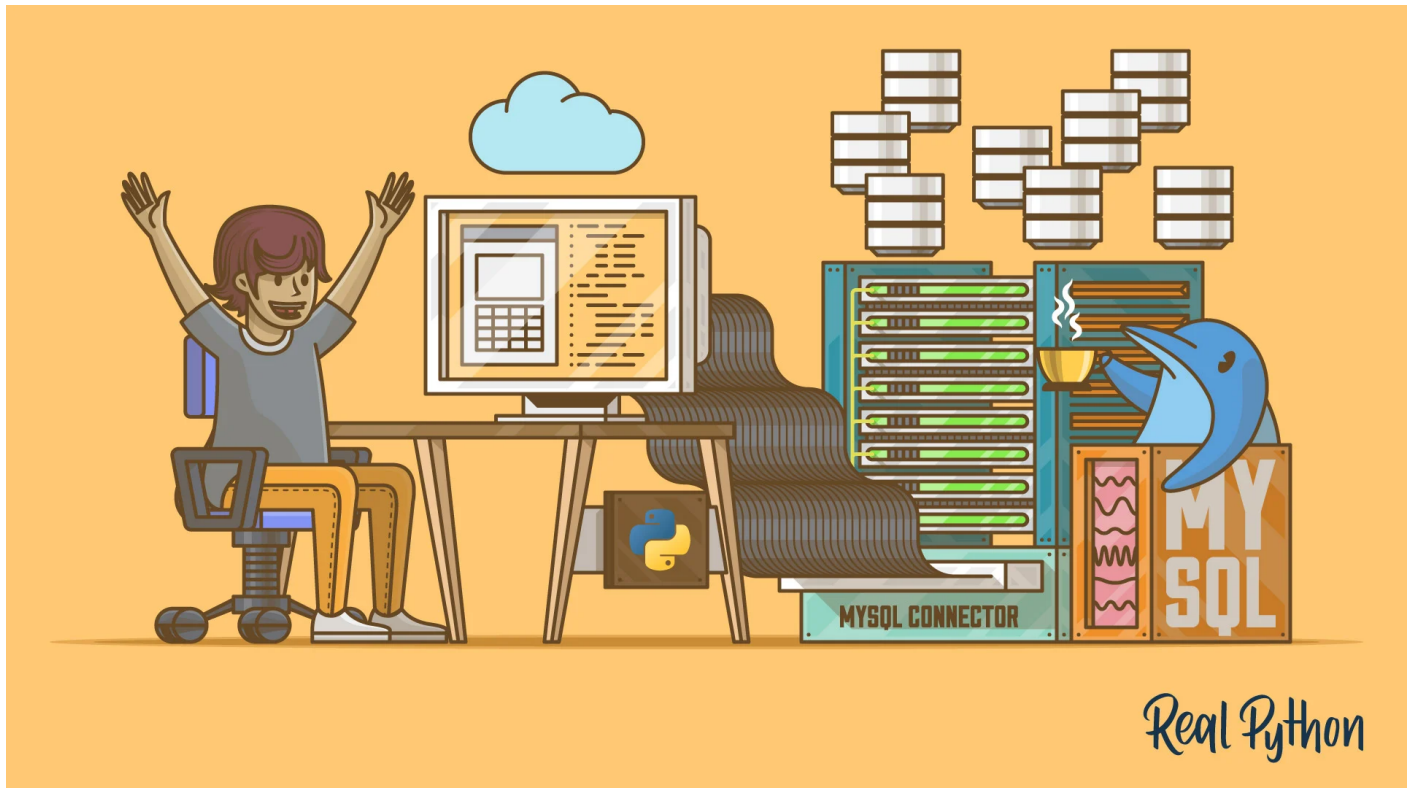


# TP 1 - Diplomado Data Engineer



En esta entrega, se presenta la primera práctica del Diplomado en Ingeniería de Datos. El objetivo principal es utilizar el lenguaje de consultas SQL para resolver los ejercicios propuestos. Durante la clase, creamos una base de datos e insertamos los registros necesarios. El repositorio con la base de datos y las consultas utilizadas para resolver esta práctica se encuentra disponible en el siguiente enlace:

[https://github.com/Fabrilennart5/diplomado\\_ingenieria\\_datos\\_practica\\_1/tree/main](https://github.com/Fabrilennart5/diplomado_ingenieria_datos_practica_1/tree/main)



En este proyecto, demostré varias habilidades. Utilicé SQL para escribir consultas eficientes que analizaran datos en bases de datos relacionales, siguiendo las buenas prácticas de formato sugeridas por Simon Holywell. Además, mostré mi comprensión de las relaciones entre tablas. También documenté el script de la base de datos de manera clara y concisa, facilitando su comprensión para otros usuarios. Además, demostré mi capacidad para trabajar con Docker al levantar un servidor MySQL utilizando una imagen de Docker para poner en práctica el proyecto. A continuación se muestran las preguntas y el código usado para responderlas:

---

Pregunta 1: ¿Podrías obtener la lista de juegos de la clase "Aventura" con su descripción y nivel? 🎮 🔍

```
SELECT
    g.name,
    c.description,
    g.id_level,
    l.description
FROM
    GAME AS g
LEFT JOIN
    CLASS AS c ON g.id_level = c.id_level
LEFT JOIN
    LEVEL_GAME AS l ON g.id_level = l.id_level
WHERE
    c.description = "Adventure";
```

Pregunta 2: ¿Puedes encontrar los nombres de los usuarios que han completado el juego con ID 5? 😬

```
SELECT
    s.first_name,
    s.last_name,
    s.id_system_user,
    p.completed
FROM
    SYSTEM_USER AS s
LEFT JOIN
    PLAY AS p
ON
    s.id_system_user = p.id_system_user
WHERE
    s.id_system_user = 5
    AND p.completed = 1;
```

Pregunta 3: ¿Puedes mostrar los comentarios realizados después del 1 de enero de 2024, ordenados por fecha ascendente? 📅 ⬆️ "TOP"

```
SELECT
    comment_date,
    commentary
FROM
    COMMENTARY
WHERE
    comment_date > '2024-01-01'
ORDER BY
    comment_date DESC;
```

"Pregunta 4: ¿Puedes calcular la cantidad de juegos en cada clase? 🎮 1234

```
SELECT
    c.id_class,
    COUNT(g.id_game) AS cantidad
FROM
    GAME AS g
LEFT JOIN
    CLASS AS c ON g.id_class = c.id_class
GROUP BY
    c.id_class
ORDER BY
    cantidad DESC;
```

Pregunta 5: ¿Puedes obtener el nombre del juego con la calificación más alta? 🎮 ☀️

```
SELECT
    g.id_game,
    g.name,
    v.value
FROM
    GAME AS g
LEFT JOIN
    VOTE AS v
    ON g.id_game = v.id_game
ORDER BY
    v.value DESC
```

```
LIMIT
```

```
1;
```

---

Pregunta 6: ¿Puedes encontrar los usuarios que han jugado más de 3 juegos diferentes? 🎮 👤

```
SELECT
```

```
    p.id_system_user,
```

```
    s.first_name,
```

```
    COUNT(p.completed) AS total_juegos
```

```
FROM
```

```
    PLAY AS p
```

```
INNER JOIN
```

```
    SYSTEM_USER AS s ON p.id_system_user = s.id_system_user
```

```
GROUP BY
```

```
    p.id_system_user,
```

```
    s.first_name
```

```
HAVING
```

```
    total_juegos > 3;
```

---

Pregunta 7: ¿Puedes mostrar la lista de juegos con una calificación de 4 o 5, junto con el nombre del usuario que los sugirió (si existe)? 🎮 👤

```
SELECT
```

```
    g.name,
```

```
    v.value,
```

```
    s.first_name,
```

```
    v.id_system_user
```

```
FROM
```

```
    GAME AS g
```

```
INNER JOIN
```

```
    VOTE AS v
```

```
ON
```

```
    g.id_game = v.id_game
```

```
INNER JOIN
```

```
    SYSTEM_USER AS s
```

```
ON
```

```
v.id_system_user = s.id_system_user
WHERE
v.value = 4 OR v.value = 5;
```

Pregunta 8: ¿Podrías calcular el promedio de las calificaciones de los juegos de la clase 'Motion Graphics'? 📁 ✨

```
SELECT
    g.id_class,
    c.description,
    ROUND(AVG(v.value)) AS promedio_calificacion
FROM
    GAME AS g
LEFT JOIN
    VOTE AS v ON g.id_game = v.id_game
LEFT JOIN
    CLASS AS c ON g.id_class = c.id_class
GROUP BY
    g.id_class,
    c.description
HAVING
    g.id_class = 153;
```

PREGUNTA 9: ¿Puedes encontrar los usuarios que no han realizado ningún comentario? 🤖

```
SELECT
    s.first_name,
    s.last_name,
    c.commentary
FROM
    SYSTEM_USER AS s
LEFT JOIN
    COMMENTARY AS c
ON
```

```
s.id_system_user = c.id_system_user  
WHERE  
c.commentary IS NULL;
```

---

Pregunta 10: ¿Puedes mostrar la lista de juegos, junto con la cantidad de usuarios que los han jugado? 🎮 👤

```
SELECT  
    g.name AS juego,  
    COUNT(p.id_system_user) AS cantidad_usuarios  
FROM  
    GAME AS g  
LEFT JOIN  
    PLAY AS p ON g.id_game = p.id_game  
GROUP BY  
    g.name;
```