SQL Work Life Tracker Project

This project simulates a real-world work life tracking system using SQL and DB Browser for SQLite.

It includes database creation, table design, record insertion, and various SQL queries from beginner to intermediate level.

---

📚 Project Overview

The database contains three main tables:

- \*\*Projects\*\*: List of projects with IDs and names

- \*\*Tasks\*\*: List of tasks assigned under projects with statuses

- \*\*Leaves\*\*: Record of employee leave requests and approval status

---

🔥 Skills Demonstrated

- Database design and table creation

- SQL querying: `SELECT`, `WHERE`, `ORDER BY`, `JOIN`, `GROUP BY`

- Writing business-relevant queries

- Real-life scenarios mapped into a database model

---

🚀 Project Files

| File | Description |

| `Work\_Life\_Tracker\_Queries.sql` | SQL queries (Beginner + Intermediate) |

| `Work\_Life\_Tracker.db` | Database file created in SQLite |

---

🧠 Learnings from Project

- How to design a basic relational database

- How to query data to solve real-world work life problems

- Basic data summarization and reporting using SQL

---

📍 Tools Used

- DB Browser for SQLite

- SQL

**SQL Developer Best Practices (for Life!)**

👉 Always Design Before You Code:

* Sketch tables and relationships first on paper or mindmap.
* Think like "What story am I trying to tell with my data?"

👉 Always Name Clearly:

* Table names = plural (Tasks, Projects, Leaves)
* Column names = meaningful (task\_name, status, leave\_date)

👉 Always Indent Neatly:

* Good indentation in queries makes debugging 100x easier.

👉 Always Comment Complex Queries:

* Add -- this query fetches pending leaves kind of comments if queries are bigger later.

👉 Always Think in Stories:

* SQL is not just code, it’s **"talking to the database"** to solve **someone’s real-world question**.

👉 Always Save in Versions:

* If you update your project later (example, add "Work Hours Table"), version it like v2, v3 etc.