My SQL Progress

Day 1 – Tuesday

Platform: Microsoft SQL Server

Topics Covered:

- **Database & Table Setup:**
  - `CREATE DATABASE`, `CREATE TABLE`, `INSERT INTO`
- **Data Querying:**
  - `SELECT`, `DISTINCT`, `WHERE`, `ORDER BY`, `ASC`, `DESC`
- **Logical Operators:**
  - `AND`, `OR`, `NOT`, `IN`, `BETWEEN`, `LIKE`
- **Aggregate Functions:**
  - `MAX()`, `MIN()`, `SUM()`, `AVG()`, `COUNT()`
- **Constraints:**
  - `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK`

- **Table Management:**
    - **`ALTER TABLE`, `UPDATE`, column aliases using `AS`**

Practice:
Built a complete student database from scratch and performed filtered queries using conditions, ordering, and aggregations.

Day 2 – Wednesday

Microsoft SQL Server

Topics Covered:

- Stored Procedures: Created reusable SQL blocks using `CREATE PROCEDURE`
- Indexing: Improved performance with `CREATE INDEX`
- SELECT INTO: Cloned data into new tables
- SELECT TOP: Fetched limited records for previews
- Database Backup: Used `BACKUP DATABASE` to protect data
- Views: Created read-only virtual tables
- Drop Table: Removed unwanted tables using `DROP`

- Practice:
  Performed full database backups, created views to hide complexity, and optimized queries using indexes.

- Bulk Insert:
  Added 5 students with names starting with “A” and ages between 18–28.

- Conditional Update:
  Increased the age by 2 for all students under 20.

- Safe Delete:
  Removed students over age 60 using a transaction. Rolled back if deletion count exceeded 2.

Day 4 – Friday

Platforms: SQL & Python Combined Practice

Topics Covered:

- SQL Constraints & Schema Evolution:
    - Added `email` column to `students` with `NOT NULL` & `UNIQUE`
    - Populated existing rows with mock emails like `name@example.com`
- CHECK Constraint Update:
    - Limited `subjects` column to accept only `'Math'`, `'Science'`, `'History'`

Python Practice:

- Created Rock-Paper-Scissors Game
- Built a Rent Calculator using input/output and conditional logic

---

Day 5 – Monday

Topic: SQL JOINs

Types of JOINs Covered:

- INNER JOIN – Matches rows from both tables
- LEFT JOIN (LEFT OUTER JOIN) – All from left + matched from right
- RIGHT JOIN (RIGHT OUTER JOIN) – All from right + matched from left
- FULL OUTER JOIN – All rows from both tables
- CROSS JOIN – Cartesian product of two tables
- SELF JOIN – Join a table with itself
- NATURAL JOIN – Automatically joins columns with the same names (theoretical concept in SQL Server)

Practice:
Joined tables like `Students` and `Subjects` to extract meaningful relationships and apply constraints in relational data structures.