

SQL

- Install RDBMS, DBeaver and connectors
- Create Database and table
- Insert data and query table
- SQL Filter
- Database operations from Python

Installation

- Postgres: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>
- DBeaver: <https://dbeaver.io/download/>
- `pip install psycopg2-binary / pymysql` if MySQL is pre-installed
- Open DBeaver and add connection for your database

Database

- Holds collection of the tables. You can think Database as Excel File and Tables as each sheet in an Excel file
- Table stores data in tabular form similar to the excel. It has rows and columns.
- You can create a new database with command : **CREATE DATABASE {database_name};**
- For creating table, you need to connect to that database and run command to create it

SQL Commands

- DDL (Data Definition Language): CREATE, ALTER, DROP
- DML (Data Manipulation Language): INSERT, UPDATE, DELETE
- DCL (Data Control Language): GRATE, REVOKE, CREATE USER
- DQL (Data Query Language): SELECT, JOIN, GROUP, HAVING
- TCL (Transaction Control Language): COMMIT, ROLLBACK

Create Table

```
CREATE TABLE {table_name} (  
    {column_name} {data_type}, -- Constraints  
    ....,  
    ....);
```

Common Data Types

- Int, serial
- Float, date
- Varchar(255)
- text

Create Table

Create Table with Schema as below:

transaction						
id	txn_date	customer_id	product_id	product_qty	total_price	cashier_id
1	2025-07-07	1	1	2	15	1
2	2025-07-20	1	2	1	180	2
customer						
id	registered_date	name	phone	address	is_vip	
1	2025-07-01	Hari	6565	Nepal	FALSE	
employee						
id	name	joined_date	department	salary	is_active	manager_id
1	Ram	2024-01-01	Cashier	2000	TRUE	NULL
2	Shyam	2024-01-01	Cashier	3000	TRUE	1
product						
id	name	stock	price			
1	Noodles	20	25			
2	Juice	200	50			

Constraint

- Helps to enforce data quality
- If data is not within a constraint, it is rejected
- Constraints:
 - NOT NULL
 - UNIQUE
 - DEFAULT
 - CHECK
 - KEY (PRIMARY and FOREIGN)

Create Constraint

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    email TEXT UNIQUE,  
    username TEXT,  
    country varchar(255),  
    CONSTRAINT unique_username UNIQUE (username)  
);
```

```
ALTER TABLE users ADD CONSTRAINT unq_em_nam  
UNIQUE(username, email);
```

- DEFAULT {default_value}
- CHECK (check_condition)

DROP

- DROP TABLE {table_name};
- DROP DATABASE {database_name};
- TRUNCATE TABLE {table_name};

Question

Drop tables created earlier and create new one with constrains as below:

1. Transaction Table

- id serial primary key
- customer_id not null foreign key to customer table
- product_id not null foreign key to product table
- Total price not null
- cashier id foreign key to employee table

2. Customer Table

- id serial and primary key
- registered date default value as now
- name not null
- phone unique and check length = 10
- address default as 'nepal'

3. Employee

- id serial primary key
- name not null
- joined_date default now
- salary check > 0

ALTER

- Used to modify existing column or constraint
- We can add, modify or remove column
- Modify includes:
 - Changing Column Name
 - Changing Data Type
 - Adding New constraint

ALTER Example

- ALTER TABLE {table_name}
ADD COLUMN {col_name} {data_type} {constraint};
- ALTER TABLE {table_name}
ALTER COLUMN {col_name} TYPE {new_datatype};
- ALTER TABLE {table_name}
RENAME COLUMN {column} TO {new_name};
- ALTER TABLE {table_name}
DROP COLUMN {col_name};
- ALTER TABLE {table_name} RENAME TO {new_name};

Questions

1. Remove column total_price of the table transaction
2. Add new column discount_amount to the table transaction
3. Rename column cashier_id to employee_id of table transaction
4. Add field mobile_number to table employee (make it unique)

Add Data

```
INSERT INTO table_name(col_1, col_2) VALUES  
( 'a', '1' ), ( 'b', '2' );
```

```
SELECT * FROM table_name;
```

```
SELECT col_1, col_2 FROM table_name;
```

```
SELECT * FROM table_name LIMIT 2;
```

```
SELECT col_1 FROM table_name LIMIT 2;
```

UPDATE

- Used to update existing record of table using filter condition
- We can combine filter with **AND** / **OR**

```
UPDATE table  
SET col_1 = val_1  
WHERE col_1 = 'a'  
AND col_2 = '1';
```

DELETE

- Used to remove record from table
- We can combine filter with **AND** / **OR**

```
DELETE FROM table  
WHERE col_1 = 'a'  
AND col_2 = '1';
```

Restore Data

- Download data for DDL and DML
- Understand the code on it and run via DBeaver
- Navigate via GUI & list constraint name on it
- Draw a relationship graph between tables
- Using SQL, identify columns in each table

DQL

- To fetch result from database based on condition
- We can use where filter, projection & LIMIT
- We can use DISTINCT to fetch unique results.
- Example:
 - `SELECT * FROM table;`
 - `SELECT col_1 AS c FROM table;`
 - `SELECT col_1 FROM table WHERE address = 'KTM' LIMIT 5`
 - `SELECT distinct name from student_info;`

Filter

- Give only the result that matches the filter provided by user
- We can combine filter with **AND** / **OR**

```
SELECT
```

```
    *
```

```
FROM student_info
```

```
WHERE address = 'California'
```

```
    AND student_name = 'Alex';
```

SQL Filter

The most common filter used in SQL are:

- >, <, >=, <=, <>, =
- IS NULL, IS NOT NULL
- IN, BETWEEN
- LIKE

Order By

- We can use order by to sort the result set
- Default sorting is ASC but we can order in DESC as well
- We sort by multiple columns as well.
- In multi column sort result is sorted in order of column
- Example:
 - `SELECT a, b, c FROM table ORDER BY a;`
 - `SELECT a, b, c FROM table ORDER BY b, a DESC;`
 - `SELECT a, b, c FROM table ORDER BY a DESC, b;`

Code Blueprint

SELECT

{fields} **AS** {aliasing}

FROM {table} **AS** {aliasing}

WHERE {filter} {filter}

GROUP BY {grouping fields}

HAVING {filter}

ORDER BY {ordering}

LIMIT {limiting}



Exercise