



# TechSaksham Training - Day 8 Notes

---

## ◆ SQL Joins and Their Usage

Joins in SQL allow us to combine rows from two or more tables based on a related column.

### 1 Left Join

- Returns all records from the left table and the matching records from the right table.
- If no match is found, NULL values are returned for the right table.

#### Example Query:

```
SELECT * FROM product  
LEFT JOIN payment ON product.pid = payment.pid;
```

### 2 Right Join

- Returns all records from the right table and the matching records from the left table.
- If no match is found, NULL values are returned for the left table.

#### Example Query:

```
SELECT * FROM product  
RIGHT JOIN payment ON product.pid = payment.pid;
```

### 3 Cross Join

- Produces a Cartesian product of two tables, meaning every row from the first table is paired with every row from the second table.

#### Example Query:

```
SELECT * FROM users CROSS JOIN product;
```

### 4 Self Join

- A table is joined with itself. Useful for hierarchical relationships, like employees and their managers.

### Creating a Table for Self Join:

```
CREATE TABLE emp (  
    eid INT PRIMARY KEY,  
    ename VARCHAR(20),  
    designation VARCHAR(200),  
    bid INT  
);
```

### Inserting Values:

```
INSERT INTO emp VALUES  
(1,'John','Developer',6),  
(2,'Abc','Developer',6),  
(3,'Rocky','Developer',6),  
(4,'Bhai','Developer',6),  
(5,'Rahul','Tester',8),  
(6,'Xyz','D-Manager',9),  
(7,'Shelar','Tester',8),  
(8,'Tom','T-Manager',9),  
(9,'Akshay','CEO',NULL);
```

### Performing Self Join:

```
SELECT a.ename AS Employee, a.designation, b.ename AS Manager, b.designation  
FROM emp AS a, emp AS b  
WHERE a.bid = b.eid;
```

## 5 Natural Join

- Automatically joins two tables based on columns with the same name and data type.
- Unlike INNER JOIN, the join condition is not explicitly mentioned.

### Example Query:

```
SELECT * FROM table1  
NATURAL JOIN table2;
```

---

## ◆ Creating and Managing Users in MySQL

### Creating a New User

```
CREATE USER 'user12'@'localhost' IDENTIFIED BY 'user12';
```

## Granting Privileges to the User

```
GRANT SELECT ON database_name.table_name TO 'user12'@'localhost';
```

## Revoking Privileges from the User

```
REVOKE SELECT ON database_name.table_name FROM 'user12'@'localhost';
```

---

## ◆ Transaction Control Language (TCL)

- **START TRANSACTION** - Begins a transaction.
  - **SAVEPOINT** - Creates a savepoint within the transaction.
  - **ROLLBACK TO SAVEPOINT** - Rolls back to a previous savepoint.
  - **COMMIT** - Saves all changes made in the transaction.
- 

## ◆ Case Study: Hospital Management System

### Database Schema

#### Creating Tables

```
CREATE TABLE patient (  
    pid INT PRIMARY KEY,  
    pname VARCHAR(100) NOT NULL,  
    paddress VARCHAR(255),  
    disease VARCHAR(100)  
);
```

```
CREATE TABLE hospital (  
    hid INT PRIMARY KEY,  
    hname VARCHAR(100) NOT NULL,  
    location VARCHAR(255)  
);
```

```
CREATE TABLE payments (  
    payid INT PRIMARY KEY,  
    amount DECIMAL(10,2) NOT NULL,  
    pid INT,
```

```

hid INT,
FOREIGN KEY (pid) REFERENCES patient(pid),
FOREIGN KEY (hid) REFERENCES hospital(hid)
);

CREATE TABLE admit (
  aid INT PRIMARY KEY,
  pid INT,
  hid INT,
  payid INT,
  FOREIGN KEY (pid) REFERENCES patient(pid),
  FOREIGN KEY (hid) REFERENCES hospital(hid),
  FOREIGN KEY (payid) REFERENCES payments(payid)
);

```

---

## ◆ Set Operators in MySQL

Operator	Description	Duplicates Removed?
UNION	Combines results and removes duplicates	✓ Yes
UNION ALL	Combines results and keeps duplicates	✗ No
INTERSECT	Returns common records	✓ Yes
EXCEPT	Returns records from the first query not in the second	✓ Yes

### Example Queries

```

SELECT * FROM employees1 UNION SELECT * FROM employees2;
SELECT * FROM employees1 UNION ALL SELECT * FROM employees2;
SELECT * FROM employees1 EXCEPT SELECT * FROM employees2;

```

---

## ◆ Connecting MySQL with Python

### Installing Required Libraries

```

pip install mysql-connector-python
pip install pymysql

```

## Connecting MySQL with Python

```
import pymysql
myconn = pymysql.connect(
    host='localhost',
    user='root',
    password='your_password',
    database='pywit'
)
cur = myconn.cursor()
cur.execute('SHOW TABLES')
for i in cur:
    print(i)
```

---

 Looking forward to applying these concepts in real-world applications! 