```
CREATE TABLE EMPLOYEE (
   EMP_ID INT NOT NULL,
   EMP_NAME VARCHAR(20) NOT NULL,
   DEPT_ID INT,
   PRIMARY KEY (EMP_ID),
   FOREIGN KEY (DEPT_ID) REFERENCES DEPARTMENTS(DEPT_ID)
);


SELECT * FROM EMPLOYEE
INSERT INTO EMPLOYEE (EMP_ID, EMP_NAME, DEPT_ID) VALUES (1, 'JOHN', 10), (2,
'ALICE', 20), (3, 'BOB', NULL), (4, 'CHARLIE', 30), (5, 'DAVID', 10)

CREATE TABLE DEPARTMENTS (
   DEPT_ID INT NOT NULL PRIMARY KEY,
   DEPT_NAME VARCHAR(20) NOT NULL
);


SELECT * FROM DEPARTMENTS

INSERT INTO DEPARTMENTS (DEPT_ID, DEPT_NAME) VALUES (10, 'HR'), (20, 'IT'), (30,
'SALES'), (40, 'MARKETING')
```

1. **INNER JOIN - The INNER JOIN keyword selects records that have matching values in both tables.**

   ```
   SELECT EMP_NAME, DEPT_NAME
   FROM EMPLOYEE
   INNER JOIN DEPARTMENTS ON EMPLOYEE.DEPT_ID = DEPARTMENTS.DEPT_ID
   ```

   Output:

EMP_NAME	DEPT_NAME
JOHN	HR
ALICE	IT
CHARLIE	SALES
DAVID	HR

2. **LEFT JOIN - The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.**

```
SELECT EMP_NAME, DEPT_NAME
FROM EMPLOYEE
LEFT JOIN DEPARTMENTS ON EMPLOYEE.DEPT_ID = DEPARTMENTS.DEPT_ID
```

Output:

EMP_NAME	DEPT_NAME
JOHN	HR
ALICE	IT
BOB	NULL
CHARLIE	SALES
DAVID	HR

3. **RIGHT JOIN - The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.**

```
SELECT EMP_NAME, DEPT_NAME
FROM EMPLOYEE
RIGHT JOIN DEPARTMENTS ON EMPLOYEE.DEPT_ID = DEPARTMENTS.DEPT_ID
```

Output:

EMP_NAME	DEPT_NAME
JOHN	HR
ALICE	IT
CHARLIE	SALES
DAVID	HR
NULL	MARKETING

4. **CROSS JOIN - Returns the Cartesian product of the two tables, i.e., all possible combinations of rows.**

```
SELECT EMP_NAME, DEPT_NAME
FROM EMPLOYEE
CROSS JOIN DEPARTMENTS
```

Output:

EMP_NAME	DEPT_NAME
JOHN	HR
JOHN	IT
JOHN	SALES
JOHN	MARKETING
ALICE	HR
ALICE	IT
ALICE	SALES
ALICE	MARKETING
BOB	HR
BOB	IT
BOB	SALES
BOB	MARKETING
CHARLIE	HR
CHARLIE	IT
CHARLIE	SALES
CHARLIE	MARKETING
DAVID	HR

5. **FULL OUTER JOIN - The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.**
   Tip: FULL OUTER JOIN and FULL JOIN are the same.

   SELECT EMP_NAME, DEPT_NAME
   FROM EMPLOYEE
   FULL OUTER JOIN DEPARTMENTS ON EMPLOYEE.DEPT_ID =
   DEPARTMENTS.DEPT_ID

   Output:

EMP_NAME	DEPT_NAME
JOHN	HR
ALICE	IT
BOB	NULL
CHARLIE	SALES
DAVID	HR
NULL	MARKETING

UNION - Combines the results of two or more SELECT statements

```
SELECT EMP_NAME AS NAME, DEPT_ID
FROM EMPLOYEE
UNION
SELECT DEPT_NAME AS NAME, DEPT_ID
FROM DEPARTMENTS;
```

NAME	DEPT_ID
ALICE	20
BOB	NULL
CHARLIE	30
DAVID	10
HR	10
IT	20
JOHN	10
MARKETING	40
SALES	30

GROUP BY: The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.