

M3: SQL Lecture & Study Notes (with SQL queries)

=====

- `CREATE DATABASE testdb;`
- `SHOW DATABASES;`
- `USE testdb;`
- `SHOW TABLES;`
- `DROP TABLE <tablename>;`
- `DESC <tablename>;`
- `DROP testdb;`
- Setting FOREIGN KEY checks OFF: `set foreign_key_checks=0;`

CREATE TABLE:

`CREATE TABLE department(dname varchar(15) NOT NULL, dnumber INT, Primary key(dnumber));`

PRIMARY KEY & FOREIGN KEY:

`CREATE TABLE employee (Fname VARCHAR(30) NOT NULL,
Minit VARCHAR(30),
Lname VARCHAR(30),
Ssn CHAR(9),
Bdate DATE,
Address VARCHAR (100),
Gender CHAR(1),
Salary DECIMAL(10,2) NOT NULL DEFAULT 10000.00,
Dno INTEGER,
PRIMARY KEY(Ssn),
FOREIGN KEY(Dno) REFERENCES department(dnumber));`

Field	Type	Null	Key	Default	Extra
Fname	varchar(30)	NO		NULL	
Minit	varchar(30)	YES		NULL	
Lname	varchar(30)	YES		NULL	
Ssn	char(9)	NO	PRI	NULL	
Bdate	date	YES		NULL	
Address	varchar(100)	YES		NULL	
Gender	char(1)	YES		NULL	
Salary	decimal(10,2)	NO		10000.00	
Dno	int	YES	MUL	NULL	

DEFAULT:

CREATE TABLE department(dname varchar(15) NOT NULL **DEFAULT** "NewDept", dnumber INT, Primary key(dnumber));

Field	Type	Null	Key	Default	Extra
dname	varchar(15)	NO		NewDept	
dnumber	int	NO	PRI	NULL	

DROP TABLE:

DROP TABLE employee;

DROP TABLE employee, department;

CHECK:

CREATE TABLE department(dname varchar(15) NOT NULL, dnumber INT **CHECK** (dnumber>0 and dnumber<21), Primary key(dnumber));

Explanation of Circular dependency in Referential Integrity constraint:

CREATE TABLE department(dname varchar(15) NOT NULL, dnumber INT, MGRSSN CHAR(9) NOT NULL, MGRSTARTDATE DATE, Primary key(dnumber));

```
CREATE TABLE employee (Fname VARCHAR(30) NOT NULL,
                        Minit VARCHAR(30),
                        Lname VARCHAR(30),
                        Ssn CHAR(9),
                        Bdate DATE,
                        Address VARCHAR (100),
                        Gender CHAR(1),
                        Salary DECIMAL(10,2) NOT NULL,
                        Dno INTEGER,
                        PRIMARY KEY(Ssn),
                        FOREIGN KEY(Dno) REFERENCES department(dnumber));
```

ALTER TABLE & Add/Drop Constraint:

ALTER TABLE department **ADD CONSTRAINT** DEPTFK FOREIGN KEY (MGRSSN) REFERENCES employee(ssn);

Thus, using ALTER TABLE command the FK has been set.

Note: We can also remove the constraint by altering the table and using drop:

ALTER TABLE department **DROP CONSTRAINT** DEPTFK;

View ALL Constraints:

The SQL command to find out the different constraints for the 'testdb' database is:

SELECT * FROM information_schema.table_constraints where constraint_schema = 'testdb';

```
mysql> SELECT * FROM information_schema.table_constraints where constraint_schema = 'testdb';
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	ENFORCED
def	testdb	PRIMARY	testdb	department	PRIMARY KEY	YES
def	testdb	PRIMARY	testdb	dependent	PRIMARY KEY	YES
def	testdb	dependent_ibfk_1	testdb	dependent	FOREIGN KEY	YES
def	testdb	PRIMARY	testdb	employee	PRIMARY KEY	YES
def	testdb	employee_ibfk_1	testdb	employee	FOREIGN KEY	YES
def	testdb	PRIMARY	testdb	project	PRIMARY KEY	YES
def	testdb	project_ibfk_1	testdb	project	FOREIGN KEY	YES
def	testdb	PRIMARY	testdb	works_on	PRIMARY KEY	YES
def	testdb	works_on_ibfk_1	testdb	works_on	FOREIGN KEY	YES
def	testdb	works_on_ibfk_2	testdb	works_on	FOREIGN KEY	YES

ADD Column from a table:

ALTER TABLE employee **ADD** Age INT;

DROP (Remove) Column from a table:

ALTER TABLE employee **DROP** Age;

RENAME Column of a table:

ALTER TABLE employee **RENAME** column Bdate to DOB;

SELECT:

SELECT * FROM department;

SELECT * FROM employee;

dname	dnumber
Research	5
Admin	10
Payroll	15
HR	17
Purchase	23

View Selective columns:

SELECT Fname,Lname **FROM** employee;

SELECT Fname,Lname,Salary **FROM** employee **WHERE** Salary>70000.00;

Fname	Lname
Rohit	Goel
Mohan	Goel
Seema	Arora
Rohan	Wadhwa
Shivani	Monga
Surbhi	Bajaj

DISTINCT:

Select **DISTINCT** Lname from employee;

Select **DISTINCT** Lname,Fname from employee;

Select **DISTINCT** Lname,Address from employee;

Note: This is used to eliminate duplicate tuples in the query result.

```
[mysql> Select DISTINCT Lname from employee;
+-----+
| Lname |
+-----+
| Goel  |
| Arora |
| Wadhwa |
| Monga |
| Bajaj |
+-----+
5 rows in set (0.00 sec)
```

```
[mysql> Select DISTINCT Lname,Fname from employee;
+-----+-----+
| Lname | Fname |
+-----+-----+
| Goel  | Rohit |
| Goel  | Mohan |
| Arora | Seema |
| Wadhwa | Rohan |
| Monga | Shivani |
| Bajaj | Surbhi |
+-----+-----+
```

INSERT:

Insert into department values('Research',5);

Insert into employee values('Rohit','K','Goel','11111111','1987-07-01','Noida, UP', 'M', 80000.20, 5);

Selective INSERT:

Insert into employee (Fname,Ssn,Gender,Salary,Dno)

values('Rohit','444444444','M',80000.20,5);

DELETE:

DELETE FROM employee WHERE Ssn='444444444';

UPDATE:

UPDATE employee **SET** Dno=10 WHERE Ssn=222222222;

UPDATE employee **SET** Lname="Kumar" WHERE Fname="Mohan";

UPDATE employee **SET** Lname="Basu" WHERE Dno=10;

Note: Note the data types (int, varchar, char, etc.) while using the UPDATE operation.

LIKE:

Q1: Retrieve names of all employees who stay in Noida:

Here, we used % before and end of 'noida' to make sure anything with noida will be sorted out.

```
SELECT Fname,Lname,Address FROM employee WHERE Address LIKE "%noida%";
```

```
[mysql> SELECT Fname,Lname,Address FROM employee WHERE Address LIKE "%noida%";
```

Fname	Lname	Address
Rohit	Goel	Noida, UP
Mohan	Goel	Noida, UP

```
2 rows in set (0.00 sec)
```

Q2: Retrieve names of all employees who DOB are in their 80's:

Here, we replaced YY8Y-MM-DD with underscore(_) to make sure any digits are captured.

```
SELECT Fname,Lname,Bdate FROM employee WHERE Bdate LIKE "__8_____";
```

```
[mysql> SELECT Fname,Lname,Bdate FROM employee WHERE Bdate LIKE "__8_____";
```

Fname	Lname	Bdate
Rohit	Goel	1987-07-01
Rohan	Wadhwa	1982-07-01

```
2 rows in set (0.00 sec)
```

BETWEEN:

Q3: Retrieve all employees in department 5 whose salary is between 30,000 and 40,000.

```
SELECT Fname,Lname,Salary FROM employee WHERE Dno=5 AND (salary BETWEEN 50000 AND 80000);
```

Or,

```
SELECT Fname,Lname,Salary FROM employee WHERE (salary BETWEEN 50000 AND 80000) AND Dno=5;
```

```
[mysql> SELECT Fname,Lname,Salary FROM employee WHERE Dno=5 AND (salary BETWEEN 50000 AND 80000);
```

Fname	Lname	Salary
Mohan	Goel	68000.20

```
1 row in set (0.01 sec)
```

CASCADE operation: CREATE TABLE employee (Fname VARCHAR(30) NOT NULL,
Minit VARCHAR(30),
Lname VARCHAR(30),
Ssn CHAR(9),
Bdate DATE,
Address VARCHAR (100),
Gender CHAR(1),
Salary DECIMAL(10,2) NOT NULL,
Dno INTEGER,
PRIMARY KEY(Ssn),
FOREIGN KEY(Dno) REFERENCES department(dnumber) **ON UPDATE cascade**);

Note: This helps to update all FK in the emp table when the dependent table PK gets updated.

ORDER BY and **DESC**:

SELECT * FROM employee **ORDER BY** Salary;

SELECT * FROM employee ORDER BY Salary **DESC**;

```
mysql> SELECT * FROM employee ORDER BY Salary;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Surbhi	A	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5

6 rows in set (0.00 sec)

SELECT * FROM employee WHERE Dno = 5 **ORDER BY** Lname;

SELECT * FROM employee WHERE Dno = 5 ORDER BY Lname **DESC**; (#Just in reverse order)

```
mysql> SELECT * FROM employee WHERE Dno = 5 ORDER BY Lname DESC;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5

2 rows in set (0.00 sec)

SELECT * FROM employee ORDER BY Dno,Salary DESC;

SELECT * FROM employee ORDER BY Dno DESC,Salary;

SELECT * FROM employee ORDER BY Dno DESC,Salary DESC;

```
mysql> SELECT * FROM employee ORDER BY Dno DESC,Salary DESC;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Surbhi	A	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5

6 rows in set (0.00 sec)

AND, OR, IN:

SELECT * FROM employee WHERE Dno=5 OR Dno=10;

```
mysql> SELECT * FROM employee WHERE Dno=5 OR Dno=10;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10

5 rows in set (0.00 sec)

SELECT * FROM employee WHERE Dno IN (5,10);

```
mysql> SELECT * FROM employee WHERE Dno IN (5,10);
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10

5 rows in set (0.00 sec)

SELECT * FROM employee WHERE Dno=5 AND Gender='M';

```
mysql> SELECT * FROM employee WHERE Dno=5 AND Gender='M';
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5

1 row in set (0.00 sec)

Aggregate Functions: (MAX, MIN, SUM, AVG, COUNT):

SELECT MAX(salary) FROM employee;

SELECT MIN(Ssn) FROM employee;

SELECT AVG(Salary) FROM employee;

SELECT SUM(Salary) FROM employee;

SELECT COUNT(SSN) FROM employee;

CREATE VIEW & DROP VIEW:

This creates a **virtual table** where the table from which it gets created is called as the base table or, defining tables (in case it is created by joining multiple tables).

CREATE VIEW EMP_DETAILS AS SELECT Fname, Lname FROM employee;

CREATE VIEW WORKS_ON_VIEW AS SELECT Fname, Lname, pname, working_hours FROM employee, project, works_on WHERE Ssn=Essn AND pnum=pnumber;

DROP VIEW EMP_DETAILS; \\This disposes the view table.

GROUP BY:

The GROUP BY works well with aggregate functions, where we are grouping data of some attributes and finding some statistical summaries.

`SELECT dno, count(*) FROM employee GROUP BY dno;`

`SELECT Lname, count(ssn) FROM employee GROUP BY Lname;`

```
[mysql> SELECT dno, count(*) FROM employee GROUP BY dno;
+-----+-----+
| dno | count(*) |
+-----+-----+
| 5   | 2       |
| 10  | 3       |
| 15  | 1       |
+-----+-----+
3 rows in set (0.00 sec)
```

```
[mysql> SELECT Lname, count(ssn) FROM employee GROUP BY Lname;
+-----+-----+
| Lname | count(ssn) |
+-----+-----+
| Goel  | 2          |
| Arora | 1          |
| Wadhwa | 1         |
| Monga | 1          |
| Bajaj | 1          |
+-----+-----+
5 rows in set (0.00 sec)
```

Employee Table (for Reference):

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10
Surbhi	A	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15

Q4: Retrieve average salary of the employees based on department numbers:

`SELECT dno, count(ssn), AVG(salary) FROM employee GROUP BY dno;`

```
[mysql> SELECT dno, count(ssn), AVG(salary) FROM employee GROUP BY dno;
+-----+-----+-----+
| dno | count(ssn) | AVG(salary) |
+-----+-----+-----+
| 5   | 2         | 529000.200000 |
| 10  | 3         | 108333.533333 |
| 15  | 1         | 50000.200000 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```


HAVING: This comes only after GROUP BY. This is used to filter the Groups. Just, the way Select is used to filter individual records.

Q5: Retrieve the average salary of the employees based on the department with more than equal to two employees. And also, avg sal. of emps based on dept. with a salary more than 60K.

SELECT dno, count(ssn),AVG(salary) FROM employee GROUP BY dno HAVING count(ssn)>=2;
SELECT dno, count(ssn),AVG(salary) FROM employee GROUP BY dno HAVING
AVG(salary)>60000;

```
[mysql> SELECT dno, count(ssn),AVG(salary) FROM employee GROUP BY dno HAVING AVG(salary)>60000;
+-----+-----+-----+
| dno | count(ssn) | AVG(salary) |
+-----+-----+-----+
| 5   |          2 | 529000.200000 |
| 10  |          3 | 108333.533333 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

JOIN Queries:

We can retrieve data from multiple tables using JOIN queries.

Q6: Suppose we need to find the names of employees working in a particular department. This is a simple join but we have not used a JOIN query, yet.

SELECT Fname,Lname,dname FROM employee,department WHERE dname='Research'
and Dno=dnumber;

```
[mysql> SELECT Fname,Lname,dname FROM employee,department WHERE dname='Research' and Dno=dnumber;
+-----+-----+-----+
| Fname | Lname | dname |
+-----+-----+-----+
| Mohan | Goel  | Research |
| Seema | Arora | Research |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

Pre-fixing the names of the tables with their attributes to better specify the attribute/column:

SELECT employee.Fname, employee.Lname, department.dname FROM
employee,department WHERE department.dname='Research' and
employee.Dno=department.dnumber;

Using Aliases instead of using the names of the tables multiple times. **This is a GOOD practice:**

SELECT e.Fname, e.Lname, d.dname FROM employee e, department d WHERE
d.dnumber='Research' and e.Dno=d.dnumber;

NATURAL JOIN:

`Select * FROM employee NATURAL JOIN department;`

Note: In, Natural join we don't have to explicitly specify the Join condition. Here, natural join will happen only if, in both the tables (employee & department) the column name is same (i.e, 'Dno').

CARTESIAN Product:

`Select * FROM employee, department;`

The following two will give same results:

`Select * FROM employee e, department d WHERE e.Dno=d.dnumber;`

`Select * FROM employee e JOIN department d ON e.Dno=d.dnumber;`

```
mysql> Select * FROM employee e JOIN department d ON e.Dno=d.dnumber;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno	dname	dnumber
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10	Admin	10
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	60000.20	5	Research	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5	Research	5
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10	Admin	10
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10	Admin	10
Surbhi	A	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15	Payroll	15

6 rows in set (0.00 sec)

Q7: Retrieve Fname, Lname, dname of all those employees who work for department number 5 or, 10.

`SELECT e.Fname, e.Lname, d.dname FROM employee e, department d WHERE (e.Dno=5 OR e.Dno=10) AND e.Dno=d.dnumber;`

```
mysql> SELECT e.Fname, e.Lname, d.dname FROM employee e, department d WHERE (e.Dno=5 OR e.Dno=10) AND e.Dno=d.dnumber;
```

Fname	Lname	dname
Mohan	Goel	Research
Seema	Arora	Research
Rohit	Goel	Admin
Rohan	Wadhwa	Admin
Shivani	Monga	Admin

5 rows in set (0.00 sec)

Q8: Retrieve Fname, Lname, current Salary, increased Salary of 'Research' department employees. Note, all employees of the Research department are given a 10% raise.

`SELECT e.Fname, e.Lname, e.salary, e.salary*1.1 FROM employee e, department d WHERE d.dname="Research" AND e.Dno=d.dnumber;`

```
mysql> SELECT e.Fname, e.Lname, e.salary, e.salary*1.1 FROM employee e, department d WHERE d.dname="Research" AND e.Dno=d.dnumber;
```

Fname	Lname	salary	e.salary*1.1
Mohan	Goel	60000.20	74000.220
Seema	Arora	990000.20	1089000.220

2 rows in set (0.00 sec)

Q9: We can also change the names of the columns which will be displayed:

```
SELECT e.Fname, e.Lname, e.salary as current_sal, e.salary*1.1 as increased_sal  
FROM employee e, department d WHERE d.dname="Research" AND e.Dno=d.dnumber;
```

```
mysql> SELECT e.Fname, e.Lname, e.salary as current_sal, e.salary*1.1 as increased_sal FROM employee e, department d WHERE d.dname="Research" AND e.Dno=d.dnumber;
```

Fname	Lname	current_sal	increased_sal
Mohan	Goel	60000.20	74000.220
Seema	Arora	990000.20	1089000.220

2 rows in set (0.00 sec)

Q10: Retrieve Max, Min & Average Salary of 'Research' department employees.

```
SELECT Max(Salary), MIN(Salary), AVG(Salary) FROM employee, department WHERE  
dname="Research" AND Dno=dnumber;
```

```
mysql> SELECT Max(Salary), MIN(Salary), AVG(Salary) FROM employee, department WHERE dname="Research" AND Dno=dnumber;
```

Max(Salary)	MIN(Salary)	AVG(Salary)
990000.20	60000.20	529000.200000

1 row in set (0.00 sec)

NESTED QUERIES:

Q11. Retrieve the name and address of all employees who work for the 'Research' department.

```
SELECT Fname, Lname, Address FROM employee
```

```
WHERE Dno IN (SELECT dnumber FROM department WHERE dname='Research');
```

OR, we can also use this join query:

```
SELECT Fname,Lname,Address,dname FROM employee, department WHERE dname="Research"  
AND Dno=dnumber;
```

```
mysql> SELECT Fname, Lname, Address FROM employee  
-> WHERE Dno IN (SELECT dnumber FROM department WHERE dname='Research');
```

Fname	Lname	Address
Mohan	Goel	Noida, UP
Seema	Arora	Gurgaon, Haryana

2 rows in set (0.00 sec)

CORRELATED NESTED QUERIES:

Q12. Retrieve the name of each employee who has a dependent with the same first name as the employee.

SELECT e.Fname, e.Lname FROM employee e WHERE e.Ssn IN
(SELECT Essn FROM dependent WHERE Essn=e.Ssn AND e.Fname=depname);

OR, we can also use this join query:

SELECT e.Fname, e.Lname, d.depname, d.relationship FROM employee e, dependent d WHERE
d.Essn=e.Ssn AND e.Fname=d.depname;

```
mysql> SELECT e.Fname, e.Lname FROM employee e WHERE e.Ssn IN  
-> (SELECT Essn FROM dependent WHERE Essn=e.Ssn AND e.Fname=depname);
```

Fname	Lname
Rohan	Wadhwa

1 row in set (0.00 sec)

Q13: For each department, retrieve the department no., the no. of emp in the department, and their avg salary.

SELECT Dno, count(ssn), AVG(salary) FROM employee GROUP BY dno;

```
mysql> SELECT Dno, count(ssn), AVG(salary) FROM employee GROUP BY dno;
```

Dno	count(ssn)	AVG(salary)
5	2	529000.200000
10	3	108333.533333
15	1	50000.200000

3 rows in set (0.00 sec)

Now, if the department name is also required; we have to use join query:

SELECT Dno, dname, count(ssn), AVG(Salary) FROM employee, department WHERE Dno=dnumber
GROUP BY dno;

```
mysql> SELECT Dno, dname, count(ssn), AVG(Salary) FROM employee, department WHERE Dno=dnumber GROUP BY dno;
```

Dno	dname	count(ssn)	AVG(Salary)
10	Admin	3	108333.533333
5	Research	2	529000.200000
15	Payroll	1	50000.200000

3 rows in set (0.00 sec)

Q14: Retrieve fname, lname and dname of all employees who works for department number 5 or 10 or 15.

SELECT Fname, Lname, Dname, Dno FROM employee e, department d WHERE e.Dno=d.dnumber AND Dno IN(5,10,15);

```
mysql> SELECT Fname, Lname, Dname, Dno FROM employee e, department d WHERE e.Dno=d.dnumber AND Dno IN(5,10,15);
```

Fname	Lname	Dname	Dno
Mohan	Goel	Research	5
Seema	Arora	Research	5
Rohit	Goel	Admin	10
Rohan	Wadhwa	Admin	10
Shivani	Monga	Admin	10
Surbhi	Bajaj	Payroll	15

6 rows in set (0.01 sec)

OUTER JOIN:

LEFT OUTER JOIN & RIGHT OUTER JOIN:

Select * FROM employee e LEFT OUTER JOIN department d on e.Dno=d.dnumber;

Gives same output, as:

Select * FROM department d RIGHT OUTER JOIN employee e on e.Dno=d.dnumber;

```
mysql> Select * FROM employee e LEFT OUTER JOIN department d on e.Dno=d.dnumber;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno	dname	dnumber
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10	Admin	10
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	60000.20	5	Research	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5	Research	5
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10	Admin	10
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10	Admin	10
Surbhi	A	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15	Payroll	15

6 rows in set (0.00 sec)

Select * FROM department d LEFT OUTER JOIN employee e on e.Dno=d.dnumber;

Gives same output, as:

Select * FROM employee e RIGHT OUTER JOIN department d on e.Dno=d.dnumber;

```
mysql> Select * FROM employee e RIGHT OUTER JOIN department d on e.Dno=d.dnumber;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno	dname	dnumber
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	60000.20	5	Research	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5	Research	5
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10	Admin	10
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10	Admin	10
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10	Admin	10
Surbhi	A	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15	Payroll	15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	17	HR	17
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	23	Purchase	23

8 rows in set (0.00 sec)

FULL Outer Join implementation in MySQL (using, UNION):

Select * FROM employee e LEFT OUTER JOIN department d on e.Dno=d.dnumber **UNION** Select * FROM employee e RIGHT OUTER JOIN department d on e.Dno=d.dnumber;

Select * FROM department d RIGHT OUTER JOIN employee e on e.Dno=d.dnumber **UNION** Select * FROM department d LEFT OUTER JOIN employee e on e.Dno=d.dnumber;

```
mysql> Select * FROM department d RIGHT OUTER JOIN employee e on e.Dno=d.dnumber UNION Select * FROM department d LEFT OUTER JOIN employee e on e.Dno=d.dnumber;
```

dname	dnumber	Fname	Minit	Lname	Sex	Bdate	Address	Gender	Salary	Dno
Admin	10	Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	88000.20	10
Research	5	Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5
Research	5	Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	99000.20	5
Admin	10	Rohan	K	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10
Admin	10	Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10
Payroll	15	Surbhi	A	Beja	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15
HR	17	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Purchase	23	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

8 rows in set (0.01 sec)

IN clause and ANY clause: Difference

Using, IN clause we can only make = comparison,

Whereas, with ANY operator we can do =, <, >, <=, >= etc.

Query using IN:

SELECT dnumber FROM department WHERE dnumber IN (SELECT dnumber FROM department);

```
[mysql> SELECT dnumber FROM department WHERE dnumber IN (SELECT dnumber FROM department);
```

dnumber
5
10
15
17
23

5 rows in set (0.00 sec)

Query using ANY:

SELECT dnumber FROM department WHERE dnumber > ANY (SELECT dnumber FROM department);

```
[mysql> SELECT dnumber FROM department WHERE dnumber > ANY (SELECT dnumber FROM department);
```

dnumber
10
15
17
23

4 rows in set (0.00 sec)

Q15: Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.
<need to verify>

```
SELECT DISTINCT pnumber
FROM project, department, employee
WHERE Dnum=dnumber AND Mgr_ssn=Ssn
AND Lname='smith'
```

UNION

```
SELECT DISTINCT Pnumber
FROM PROJECT, WORKS_ON, EMPLOYEE
WHERE Pnumber=Pno AND Essn=Ssn
AND Lname='smith'
```

Q16: Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, then first name.

```
SELECT d.Dname, e.Lname, e.Fname, p.pname
FROM department d, employee e, works_on w, project p
WHERE d.dnumber= e.Dno AND e.Ssn= w.Essn AND w.pnum= p.pnumber
ORDER BY d.dname, e.Lname, e.Fname;
```

```
mysql> SELECT d.Dname, e.Lname, e.Fname, p.pname
-> FROM department d, employee e, works_on w, project p
-> WHERE d.dnumber= e.Dno AND e.Ssn= w.Essn AND w.pnum= p.pnumber
-> ORDER BY d.dname, e.Lname, e.Fname;
```

Dname	Lname	Fname	pname
Admin	Goel	Rohit	ProjectX
Admin	Goel	Rohit	ProjectY
Research	Arora	Seema	ProjectX
Research	Goel	Mohan	ProjectZ

4 rows in set (0.00 sec)

Snap of the created DB tables (Note: Below queries not verified):

Department:

dname	dnumber
Research	5
Admin	10
Payroll	15
HR	17
Purchase	23

Field	Type	Null	Key	Default	Extra
dname	varchar(15)	NO		NewDept	
dnumber	int	NO	PRI	NULL	

Insert into department values('Research',5);
Insert into department values('Admin',10);
Insert into department values('Payroll',15);
Insert into department values('HR',17);
Insert into department values('Purchase',23);

Employee:

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Rohit	K	Goel	111111111	1987-07-01	Noida, UP	M	80000.20	10
Mohan	M	Goel	222222222	1992-07-01	Noida, UP	M	68000.20	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	5
Rohan	k	Wadhwa	444444444	1982-07-01	Pune, Maharashtra	M	120000.20	10
Shivani	M	Monga	555555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10
Surbhi	A	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15

Field	Type	Null	Key	Default	Extra
Fname	varchar(30)	NO		NULL	
Minit	varchar(30)	YES		NULL	
Lname	varchar(30)	YES		NULL	
Ssn	char(9)	NO	PRI	NULL	
Bdate	date	YES		NULL	
Address	varchar(100)	YES		NULL	
Gender	char(1)	YES		NULL	
Salary	decimal(10,2)	NO		10000.00	
Dno	int	YES	MUL	NULL	

Insert into employee values('Rohit','k','goel','111111111','1987-07-01','Noida, UP', 'M', 80000.20,5);
Insert into employee values('Mohan','M','goel','222222222','1992-07-01','Noida, UP', 'M', 68000.20,5);
Insert into employee values('Seema','N','Arora','333333333','1977-09-07','Gurgaon,Haryana', 'F', 990000.20,5);
Insert into employee values('Rohan','k','Wadhwa','444444444','1982-07-01','Pune,Maharashtra', 'M', 120000.20,10);
Insert into employee values('Shivani','M','Monga','555555555','1973-07-04','Chandigarh, Punjab', 'F', 125000.20,10);
Insert into employee values('Surbhi','A','Bajaj','666666666','1991-07-05','Chandigarh, Punjab', 'F', 50000.20,15);
Insert into employee values('Palak','K','Arora','000000000','1993-07-09','Jaipur, Rajasthan', 'M', 60000.20,15);

Project:

pnumber	pname	plocation	dnum
100	ProjectX	Noida	5
200	ProjectY	Noida	10
300	ProjectZ	Gurgaon	15

Field	Type	Null	Key	Default	Extra
pnumber	int	NO	PRI	NULL	
pname	varchar(30)	YES		NULL	
plocation	varchar(50)	YES		NULL	
dnum	int	YES	MUL	NULL	

Create table project (Pnumber int primary key,
Pname varchar(30),
plocation varchar(50),
Dnum int,
foreign key(Dnum) references department(dnumber));

Insert into project values(100, 'ProjectX', 'Noida', 5);
Insert into project values(200, 'ProjectY', 'Noida', 10);
Insert into project values(300, 'ProjectZ', 'Gurgaon', 15);

Works_on:

Essn	pnum	working_hours
111111111	100	8.00
111111111	200	7.00
222222222	300	6.00
333333333	100	7.50

Field	Type	Null	Key	Default	Extra
Essn	char(9)	NO	PRI	NULL	
pnum	int	NO	PRI	NULL	
working_hours	decimal(5,2)	YES		NULL	

create table works_on (Essn char(9),
Pnum int,
working_hours decimal(5,2),
primary key(Essn,pnum),
foreign key(ESSN) references employee(ssn),
foreign key(pnum) references project(pnumber));

Insert into works_on values('111111111',100, 8);
Insert into works_on values('111111111',200, 7);
Insert into works_on values('222222222',300, 6);
Insert into works_on values('333333333',100, 7.5);

Dependent:

Essn	depname	gender	relationship
111111111	Asha	F	Mother
222222222	Aavya	F	Daughter
444444444	Rohan	M	Son

Field	Type	Null	Key	Default	Extra
Essn	char(9)	NO	PRI	NULL	
depname	varchar(30)	NO	PRI	NULL	
gender	char(1)	YES		NULL	
relationship	varchar(30)	YES		NULL	

```
create table dependent(ESSN char(9),  
                        Depname varchar(30),  
                        gender char(1),  
                        relationship varchar(30),  
                        primary key(ESSN,Depname),  
                        foreign key(ESSN) references Employee(ssn));
```

```
Insert into dependent values('111111111', 'Asha', 'F', 'Mother');  
Insert into dependent values('222222222', 'Aavya', 'F', 'Daughter');  
Insert into dependent values('444444444', 'Rohan', 'M', 'Son');
```