# 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:**

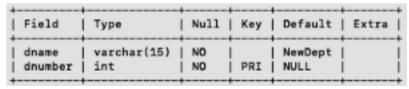
CREATE TABLE department(dname varchar(15) NOT NULL, dnumber INT, Primary key(dnumber));

### PRIMARY KEY & FOREIGN KEY:

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

#### **DEFAULT:**

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



#### **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);

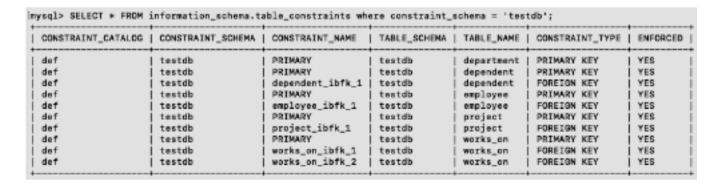
Thus, using ALTER TABLE command the FK has been set.

**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';



## ADD Column from a table:

ALTER TABLE employee ADD Age INT;

### **DROP** (Remove) Column from a table:

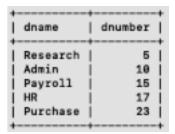
ALTER TABLE employee **DROP** Age;

#### **RENAME** Column of a table:

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

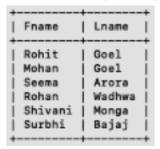
#### **SELECT:**

SELECT \* FROM department;
SELECT \* FROM employee;



## **View Selective columns:**

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



### **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.

### INSERT:

Insert into department values('Research',5);
Insert into employee values('Rohit','K','Goel','1111111111','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='4444444444';

#### **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%";

**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\_\_\_\_";

### **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;

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**;

Fname	Minit	Lname	San	Bdate	Address	Gender	Salary	Dno
Surbhi Mohan	A	Bajaj Goel	666666666	1991-07-05	Chandigarh, Punjab Noida, UP	F	50000.20	15
Rohit	K I	Goel Wadhwa	111111111	1987-07-01 1982-07-01	Noida, UP Pune,Maharashtra	и и	80000.20 120000.20	10
Shivani Seema	M	Monga	55555555 3333333333	1973-07-04	Chandigarh, Punjab Gurgaon, Haryana	F	125000.20	10

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		Salary	Dno
	Goel   222222222   Arora   333333333			М	68000.20 990000.20	5
2 rows in set (θ	.θθ 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;

Fname	Minit	Lname	San	Bdate	Address	Gender	Salary	Dno
Surbhi	A	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15
Shivani	јм ј	Monga	555555555	1973-87-84	Chandigarh, Punjab	F	125000.20	10
Rohan	k	Wadhwa	444444444	1982-87-81	Pune, Maharashtra	M	120000.20	1.0
Rohit	K	Goel	111111111	1987-87-01	Noida, UP	M	80000.20	10
Seema	į N į	Arora	333333333	1977-89-87	Gurgaon, Haryana	F	990000.20	5
Mohan	ім і	Goel	22222222	1992-87-01	Noida, UP	M	68000.20	5

## AND, OR, IN:

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

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Mohan	M	Goel	22222222	1992-87-81	Noida, UP	м	68000.20	5
Seema	l N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	990000.20	6
Rohit	į K į	Goel	111111111	1987-07-01	Noida, UP	į M	80000.20	10
Rohan	į k	Wadhwa	44444444	1982-07-01	Pune, Maharashtra	ім	120000.20	16
Shivani	i M	Monga	555555555	1973-07-04	Chandigarh, Punjab	i F	125000.20	10

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

Fname	Minit	Lname	Sen	Bdate	Address	Gender	Salary	Dno
Mohan	M	Goel	22222222	1992-07-01	Noida, UP	М	68999.29	5
Seema	N	Arora	333333333	1977-09-07	Gurgaon, Haryana	F	998888.28	5
Rohit	į K	Goel	111111111	1987-07-01	Noida, UP	М	80000.20	10
Rohan	j k	Wadhwa	44444444	1982-07-01	Pune, Maharashtra	м	120000.20	10
Shivani	i M	Monga	666666666	1973-07-04	Chandigarh, Punjab	F	125000.20	10

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

mysql> SELECT * F	FROM employee WHERE	Dno=5 AND Gen	der='M';			
Fname   Minit	Lname   Ssn	Bdate	Address	Gender	Salary	Dno
Mohan   M	Goel   222222222	1992-07-01	Noida, UP	м	68000.20	5
1 row in set (θ.6	θθ 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 <u>virtual table</u> 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;

Employee Table (for Reference):

Fname	Minit	Lname	Ssn	Bdate	Address	Gender	Salary	Dno
Rohit   Mohan   Seema   Rohan   Shivani   Surbhi	K   M   N   k   M	Goel Goel Arora Wadhwa Monga Bajaj	111111111 22222222 333333333 444444444 555555555 666666666	1987-07-01 1992-07-01 1977-09-07 1982-07-01 1973-07-04 1991-07-05	Noida, UP Gurgaon,Haryana Pune,Maharashtra Chandigarh, Punjab	M F M F	89008.20 68008.20 990008.20 120008.20 125008.20 50000.20	10   5   5   10   10   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 | 529000.2000000 |
| 10 | 3 | 108333.533333 |
| 15 | 1 | 50000.2000000 |
| 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;

### **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 |
| to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to see to s
```

**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;

<u>Using Aliases</u> 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 <u>same</u> results:

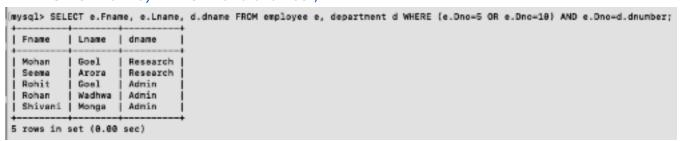
Select \* FROM employee e, department d WHERE e.Dno=d.dnumber;

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

Fnanc	: I	Minit	Lname	Ssr	n	Bdate	Address	1	Gender	1	Salary	Dno	_!	dname	dnumber
Rohit	t i	К	Coel	111	1111111	1987-07-01	Noida, UP	ï	М	ï	80000.20	1	θ	Admin	10
Mohar	ı i	М	Coel	222	2222222	1992-07-01	Noida, UP	i	м	i	68888.28	į į	5 j	Research	5
Seens	ıj	N	Arora	333	3333333	1977-09-07	Gurgaon, Haryana	i	F	i	998888.28	i i	5 j	Research	5
Rohar	ıį	k	Wedhwe	1 444	4444444	1982-07-01	Pune, Maharashtra	i	м	i	120000.20	1	ej	Admin	10
Shive	ani j	М	Monga	j 559	555555 j	1973-87-84	Chandigarh, Punjab	i	F	i	125000.20	1	eί	Admin	19
Surbi	na i	A	Bejej	i 666	6666666 1	1991-87-85	Chandigarh, Punjab	i	F	i	50000.20	1	5 i	Payrol1	15

**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;



**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;

```
| Frame | Lisse | salary | e.salary*1.1 FROM employee e, department d WHERE d.dnamem*Research* AND e.Dnomd.dnumber;
| Frame | Lisse | salary | e.salary*1.1 |
| Mohan | Soel | 68080.20 | 74800.220 |
| Seems | Accrs | 990802.20 | 1889000.220 |
| 2 rows in set (8.00 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;

**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 | 68000.20 | 529000.200000 |
| 1 row in set (0.00 sec)

## **NESTED QUERIES:**

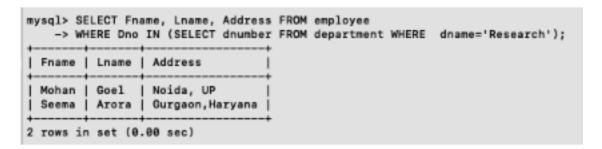
**Q11.** Retrieve the name and address of all employees who work for the 'Research' department.

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;



## **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;

**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 | 529000.2000000 |
| 10 | 3 | 108333.533333 |
| 15 | 1 | 50000.2000000 |
| 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 |

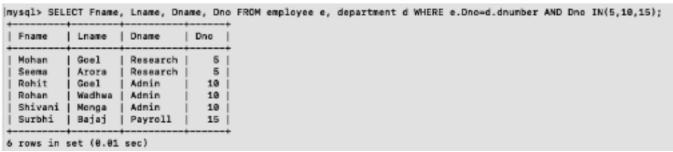
| 5 | Research | 2 | 529800.200000 |

| 15 | Payroll | 1 | 50000.200000 |

| 3 rows in set (0.00 sec)
```

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

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



#### **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;

Fname	!	Minit	Lname	!	San	Bdate	Address	Gender	Salary	Dno	dname	dnumber
Rohit	ï	K	Coel	ï	111111111	1987-07-01	Noida, UP	I M	80000.20	10	Admin	1
Mohan	i	М	Goel	i	222222222	1992-07-01	Noida, UP	jм	68999.29	j 5	Research	1
Seema	i	N	Arora	i	333333333	1977-89-87	Curgeon, Heryana	F	990000.20	j 5	Research	
Rohan	i	k	Madhwa	٠i	444444444	1982-07-01	Pune, Maharashtra	ј м	120000.20	10	Admin	1
Shivani	ιi	м	Monga	i	55555555	1973-07-04	Chandigarh, Punjab	F	125000.20	10	Admin	1
Surbhi	i	Α.	Bajaj	i	666666666	1991-87-85	Chandigarh, Punjab	i E	50000.20	15	Payroll	1

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;

Fname	!	Minit	ļ	Lname	ļ	Sen	ļ	Bdate		Address	Gender	1	Salary	Dno		dnane	dnumbe
Mohan	ĭ	н	ï	Soel	ĭ	22222222	ĭ	1992-07-01	ï	Noida, UP	м	Ť	68888.28		5	Research	
Seema	ì	N	Ĺ	Arora	Ĺ	333333333	İ	1977-89-87	ĺ	Gurgaon, Haryana	F	ı	998888.28		5	Research	
Rohit	1	K	Ĺ	Goel	Ĺ	111111111	Ĺ	1987-07-01	ĺ	Noida, UP	M	Ĺ	80000.20	1	.0	Admin	1
Rohan	i	k	Ĺ	Wadhwa	Ĺ	444444444	İ	1982-07-01	İ	Pune, Maharashtra	М	ı	120000.20	1	.0	Admin	1
Shivani	i	H	i.	Monga	Ĺ	55555555	İ	1973-07-04	İ	Chandigarh, Punjab	F	ı	125000.20	1	.0	Admin	1
Surbhi	i	A	Ĺ	Bajaj	Ĺ	66666666	Ĺ	1991-07-05	İ	Chandigarh, Punjab	F	-i	50000.20	1	5	Payroll	1
NULL	i	MULL	i.	NULL	İ.	NULL	İ	NULL	i	NULL	MULL	ı	NULL	NUU	L	HR	1
NULL	i	MULL	i.	NULL	i	NULL	i	NULL	i	NULL	MULL	-i	NULL	NUU	L	Purchase	2

# **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;

rysql> Selec mployee e c			d RIGHT	OUTER JO:	IN employee o	e on e.Dno=d.o	dnumber UNION Select	FROM dep	partment d Li	EFT OUT	ER :
dname	dnumber	Fname	Minit	Lname	San	Bdate	Address	Gender	Salary	Dno	
Admin Research Research Admin Admin Payroll HR Purchase	18 5 5 18 18 15 17 23	Rohit   Mahan   Seema   Rohan   Shivani   Surbhi   NULL   NULL	K M M K M A MULL	Goel Goel Arore Wadhwa Monga Bajaj NULL NULL	111111111 22222222 333333333 44444444 55555555 66666566 NULL NULL	1987-87-01 1992-87-01 1977-89-07 1982-87-01 1973-87-04 1991-87-05 MULL MULL	Moida, UP Moida, UP Gurgeon, Haryene Pune, Maharashtra Chandigarh, Punjab Chandigarh, Punjab MULL	M M F M F F MULL MULL	88000.28 58000.20 990000.20 120000.20 125000.28 50000.28 MULL MULL	18 5 19 18 15 MULL MULL	
rows in se	t (0.01 s	ec)			+	<b>+</b>	·	·	+	+	٠

## **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);

## 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 |
| dnumber |
| 10 |
| 15 |
| 17 |
| 23 |
| ------+
| 4 rows in set (0.00 sec)
```

**Q15:** Make a list of all project numbers for <u>projects</u> 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;

# **Snaps of the created DB tables (Note: Below queries not verified):**

## **Department:**

++	
dname   dnumber	
++	
Research   5	++
Admin   10	Field   Type   Null   Key   Default   Extra
Payroll   15	+
HR   17	dname   varchar(15)   NO     NewDept
Purchase   23	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	Sen	Bdate	Address	Gender	Salary	Dno
Rohit     Mohan     Seema	K M N	Goel Goel Arora	111111111   222222222   33333333333	1987-07-01 1992-07-01 1977-09-07	Noida, UP   Noida, UP   Gurgaon,Harvana	H	80000.20 68000.20 990000.20	10
Rohan	k M	Wadhwa Monga	44444444 55555555	1982-07-01 1982-07-01 1973-07-04	Pune, Maharashtra   Chandigarh, Punjab	M F	120000.20	19
Surbhi	Α	Bajaj	666666666	1991-07-05	Chandigarh, Punjab	F	50000.20	15

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

Insert into employee values('Rohit','k','goel','1111111111','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','33333333','1977-09-07','Gurgaon,Haryana', 'F', 990000.20,5);
Insert into employee values('Rohan','k','Wadhwa','4444444444','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
200	ProjectX ProjectY ProjectZ	Noida	5   10   15

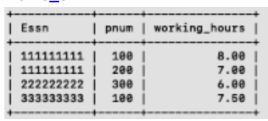
Field	Type	Null	Key	Default	Extra
pnumber   pname   plocation     dnum	int varchar(30) varchar(50) int	NO YES YES YES	PRI	NULL NULL NULL NULL	

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:



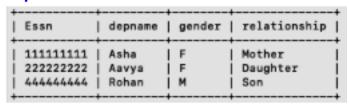
Field	Туре	Null	Key	Default	Extra	+
Essn   pnum   working_hours	char(9) int decimal(5,2)	NO NO YES	PRI PRI	NULL NULL NULL		+       +

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('1111111111',100, 8); Insert into works\_on values('111111111',200, 7); Insert into works\_on values('22222222',300, 6); Insert into works\_on values('33333333',100, 7.5);

# **Dependent:**



Field	Type	Null	Key	Default	Extra
Essn   depname   gender   relationship	char(9)   varchar(30)   char(1)   varchar(30)	NO NO YES YES	PRI PRI	NULL NULL NULL 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('22222222', 'Aavya', 'F', 'Daughter');
Insert into dependent values('444444444', 'Rohan', 'M', 'Son');
```