

20MCA134 ADVANCED DBMS LAB

LAB CYCLE 1

Experiment No: 1

DEVIKA B
ROLL NO : 22

A. Consider the database for a college. Write SQL commands to implement the following:

1. Create a database

```
CREATE DATABASE 24MCA22;
```

2. Select the current database

```
USE 24MCA22;
```



3. Create the following tables

a) Student (roll_no integer, name varchar, dob date, address text, phone_no varchar, blood_grp varchar)

```
create table Student(roll_no integer,name varchar(255),dob DATE,address text,phone_no varchar(255),blood_grp varchar (255));
```

```
mysql> create table student(roll_no integer,name varchar(255),dob DATE,address text(355),phone_no varchar(255),blood_grp varchar(255));  
Query OK, 0 rows affected (0.62 sec)
```

b) Course (Course_id integer, Course_name varchar, course_duration integer)

create table Course(Course_id integer, Course_name
varchar(255), course_duration integer);

```
mysql> Create table course(course_id integer, course_name varchar(255), course_duration integer(255));  
Query OK, 0 rows affected, 1 warning (0.27 sec)
```

```
mysql> show tables;
```

4. List all tables in the current database.

Show tables;

```
mysql> show tables;  
+-----+  
| Tables_in_24mca22 |  
+-----+  
| course            |  
| student           |  
+-----+  
2 rows in set (0.00 sec)
```

5. Display the structure of the Student table.

describe student;

```
mysql> desc student;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| roll_no    | int           | YES  |     | NULL    |       |  
| name       | varchar(255)  | YES  |     | NULL    |       |  
| dob        | date          | YES  |     | NULL    |       |  
| address    | text          | YES  |     | NULL    |       |  
| phone_no   | varchar(255)  | YES  |     | NULL    |       |  
| blood_grp  | varchar(255)  | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
6 rows in set (0.00 sec)
```

6. Drop the column blood_grp from Student table.

`alter table Student drop column blood_grp;`

```
mysql> alter table student drop column blood_grp;
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

7. Add a new column Adar_no with domain number to the table Student.

`alter table Student add column adar_no int(255);`

```
mysql> alter table student add column adar_no int(255);
Query OK, 0 rows affected, 1 warning (0.18 sec)
Records: 0 Duplicates: 0 Warnings: 1
```

8. Change the datatype of phone_no from varchar to int

`alter table student modify column phone_no int;`

```
mysql> alter table student modify column phone_no int;
Query OK, 0 rows affected (1.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

9. Drop the tables.

`DROP TABLE COURSE;`
`DROP TABLE STUDENT;`

10. Delete the database.

`DROP DATABASE 24MCA22;`

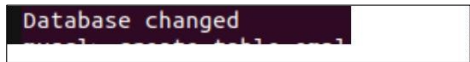
B. Consider the database for an organization. Write SQL commands to implement the following:

1. Create a database

```
CREATE DATABASE 24MCA22;
```

2. Select the current database

```
USE 24MCA22;
```



Database changed

3. Create the following tables:

a) Employee (emp_no varchar, emp_name varchar, dob date, address text, mobile_no integer, dept_no varchar, salary integer)

```
create table employee(emp_no varchar(224),emp_name  
varchar(223),dob date,address text,mobile_no varchar  
(223),salary int);
```



```
mysql> create table employee(emp_no varchar(255),emp_name varchar(300),dob date,address text(255),mobile_no integer(255),dept_no varchar(255),salary integer);  
Query OK, 0 rows affected, 1 warning (0.45 sec)
```

b) Department (dept_no varchar, dept_name varchar, location varchar)

```
create table department(dept_no varchar(224),dept_name  
varchar(223),location varchar(223));
```

```
mysql> create table department(dept_no varchar(255),dept_name varchar(255),location varchar(255));
Query OK, 0 rows affected (0.47 sec)
```

```
mysql> show tables;
```

4. List all tables in the current database.

Show tables;

```
mysql> show tables;
+-----+
| Tables_in_24mca22 |
+-----+
| department        |
| employee          |
+-----+
2 rows in set (0.00 sec)
```

5. Display the structure of the Employee table and Department table.

desc employee;

```
mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_no     | varchar(255)  | YES  |     | NULL    |       |
| emp_name   | varchar(300)  | YES  |     | NULL    |       |
| dob        | date          | YES  |     | NULL    |       |
| address    | text          | YES  |     | NULL    |       |
| mobile_no  | int           | YES  |     | NULL    |       |
| dept_no    | varchar(255)  | YES  |     | NULL    |       |
| salary     | int           | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

desc department;

```
mysql> desc department;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| dept_no    | varchar(255)  | YES  |     | NULL    |       |
| dept_name  | varchar(255)  | YES  |     | NULL    |       |
| location   | varchar(255)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

6. Add a new column 'Designation' to the table Employee.

alter table employee add column designation varchar (456);

```
mysql> alter table employee add column Designation varchar(200);
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

7. Drop the column 'location' from Department table.

alter table department drop column location;

```
mysql> alter table department drop column location;
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Experiment No: 2

Familiarization of SQL Constraints.

1. Create new table Persons with attributes PersonID (integer, PRIMARY KEY), Name (varchar , NOT NULL), Aadhar (Number, NOT NULL, UNIQUE), Age (integer , CHECK>18).

```
create table persons(PersonID integer PRIMARY KEY,name varchar(567) NOT NULL,aadhar integer NOT NULL UNIQUE,age integer CHECK(age>18));
```

```
mysql> alter table employee add constraint FK_departmentnumbers foreign key(dept_no)references department(dept_no) on delete cascade;
Query OK, 0 rows affected (1.23 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

2. CREATE TABLE Orders with attributes OrderID (PRIMARY KEY), OrderNumber(NOT NULL) and PersonID(set FOREIGN KEY on attribute PersonID referencing the column PersonId of Person table)

```
Create table orders(orderID int PRIMARY KEY,ordernumber int NOT NULL,personID int,FOREIGN KEY(personID) references persons(prsonID));
```

```
mysql> create table orders(orderID int PRIMARY KEY,OrderNumber int NOT NULL,personID int,FOREIGN KEY(personID) REFERENCES persons(personID));
Query OK, 0 rows affected (0.32 sec)
```

3. Display the structure of Persons tables.

desc persons;

```
mysql> desc persons;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| personID | int | NO | PRI | NULL | |
| name | varchar(255) | NO | | NULL | |
| Aadhar | int | NO | UNI | NULL | |
| age | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

4. Display the structure of Orders tables.

desc orders;

```
mysql> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| orderID | int | NO | PRI | NULL | |
| OrderNumber | int | NO | | NULL | |
| personID | int | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

5. Add emp_no as the primary key of the table Employee.

alter table employee modify emp_no varchar (255) PRIMARY KEY;

```
mysql> alter table employee modify emp_no varchar(255)PRIMARY KEY;
Query OK, 0 rows affected (0.62 sec)
Records: 0 Duplicates: 0 Warnings: 0
```


6. Add dept_no as the primary key of the table Department.

`alter table department modify dept_no varchar (567) PRIMARY KEY;`

```
mysql> alter table department modify dept_no varchar(255)PRIMARY KEY;  
Query OK, 0 rows affected (0.55 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

7. Add dept_no in Employee table as the foreign key reference to the table Department with on delete cascade.

`alter table employee add constraint FK_departmentnumbers foreign key(dept_no) references department(dept_no) on delete cascade;`

```
mysql> alter table employee add constraint FK_departmentnumbers foreign key(dept_no)references department(dept_no) on delete cascade;  
Query OK, 0 rows affected (1.23 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

8. Drop the primary key of the table Orders.

`Alter table orders drop primary key;`

```
mysql> alter table orders drop primary key;  
Query OK, 0 rows affected (0.78 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> █
```

Experiment No: 3 - Familiarization of DML Commands

1. Add at least 10 rows into the table Employee and Department.

```
INSERT INTO Employee (emp_no, emp_name, dob, address, mobile_no, dept_no, salary, designation) VALUES ('E001', 'John Doe', '1985-05-12', 'New York', 9876543210, 'D01', 50000, 'Manager'), ('E002', 'Alice Smith', '1990-07-22', 'California', 8765432109, 'D02', 60000, 'Software Engineer'), ('E003', 'Bob Brown', '1988-11-15', 'Texas', 7654321098, 'D03', 40000, 'HR'), ('E004', 'David Miller', '1995-03-30', 'Florida', 6543210987, 'D01', 45000, 'Manager'), ('E005', 'Emma Wilson', '1992-12-20', 'Nevada', 5432109876, 'D02', 30000, 'Computer Assistant'), ('E006', 'Johnson', '1987-08-10', 'Arizona', 4321098765, 'D03', 25000, 'Sales Executive'), ('E007', 'Sophia Garcia', '1993-06-25', 'Illinois', 3210987654, 'D04', 75000, 'Senior Engineer'), ('E008', 'Daniel Lee', '1989-02-14', 'Michigan', 2109876543, 'D04', 50000, 'Manager'), ('E009', 'Olivia Martinez', '1991-09-05', 'Ohio', 1098765432, 'D05', 28000, 'Technician'), ('E010', 'James Anderson', '1996-04-18', 'Georgia', 1987654321, 'D05', 20000, 'Support Staff');
```

```
mysql> INSERT INTO Employee (emp_no, emp_name, dob, address, mobile_no, dept_no, salary, designation) VALUES
-> ('E001', 'John Doe', '1985-05-12', 'New York', 9876543210, 'D01', 50000, 'Manager'),
-> ('E002', 'Alice Smith', '1990-07-22', 'California', 8765432109, 'D02', 60000, 'Software Engineer'),
-> ('E003', 'Bob Brown', '1988-11-15', 'Texas', 7654321098, 'D03', 40000, 'HR'),
-> ('E004', 'David Miller', '1995-03-30', 'Florida', 6543210987, 'D01', 45000, 'Manager'),
-> ('E005', 'Emma Wilson', '1992-12-20', 'Nevada', 5432109876, 'D02', 30000, 'Computer Assistant'),
-> ('E006', 'Michael Johnson', '1987-08-10', 'Arizona', 4321098765, 'D03', 25000, 'Sales Executive'),
-> ('E007', 'Sophia Garcia', '1993-06-25', 'Illinois', 3210987654, 'D04', 75000, 'Senior Engineer'),
-> ('E008', 'Daniel Lee', '1989-02-14', 'Michigan', 2109876543, 'D04', 50000, 'Manager'),
-> ('E009', 'Olivia Martinez', '1991-09-05', 'Ohio', 1098765432, 'D05', 28000, 'Technician'),
-> ('E010', 'James Anderson', '1996-04-18', 'Georgia', 1987654321, 'D05', 20000, 'Support Staff');
Query OK, 10 rows affected (0.05 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
INSERT INTO Department(dept_no,dept_name)VALUES
('D01','HR'),
('D02','Finance'),
('D03','IT'),
('D04','Marketing'),
('D05','Admin');
```

```
mysql> INSERT INTO Department(dept_no,dept_name)VALUES
-> ('D01','HR'),
-> ('D02','Finance'),('D03','IT'),('D04','Marketing'),('D05','Admin');
Query OK, 5 rows affected (0.06 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

2. Display all the records from the above tables.

SELECT * FROM Employee;

```
mysql> SELECT * FROM Employee;
+-----+-----+-----+-----+-----+-----+-----+-----+
| emp_no | emp_name | dob | address | mobile_no | dept_no | salary | Designation |
+-----+-----+-----+-----+-----+-----+-----+
| E001 | John Doe | 1985-05-12 | New York | 9876543210 | D01 | 50000 | Manager |
| E002 | Alice Smith | 1990-07-22 | California | 8765432109 | D02 | 60000 | Software Engineer |
| E003 | Bob Brown | 1988-11-15 | Texas | 7654321098 | D03 | 40000 | HR |
| E004 | David Miller | 1995-03-30 | Florida | 6543210987 | D01 | 45000 | Manager |
| E005 | Emma Wilson | 1992-12-20 | Nevada | 5432109876 | D02 | 30000 | Computer Assistant |
| E006 | Michael Johnson | 1987-08-10 | Arizona | 4321098765 | D03 | 25000 | Sales Executive |
| E007 | Sophia Garcia | 1993-06-25 | Illinois | 3210987654 | D04 | 75000 | Senior Engineer |
| E008 | Daniel Lee | 1989-02-14 | Michigan | 2109876543 | D04 | 50000 | Manager |
| E009 | Olivia Martinez | 1991-09-05 | Ohio | 1098765432 | D05 | 28000 | Technician |
| E010 | James Anderson | 1996-04-18 | Georgia | 1987654321 | D05 | 20000 | Support Staff |
+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

SELECT * FROM Department;

```
mysql> SELECT * FROM Department;
+-----+-----+
| dept_no | dept_name |
+-----+-----+
| D01 | HR |
| D02 | Finance |
| D03 | IT |
| D04 | Marketing |
| D05 | Admin |
+-----+-----+
5 rows in set (0.00 sec)
```

3. Display the emp_no and name of employees from department no 'D02'.

```
SELECT emp_no, emp_name FROM Employee WHERE dept_no = 'D02';
```

```
mysql> SELECT emp_no, emp_name FROM Employee WHERE dept_no = 'D02';
+-----+-----+
| emp_no | emp_name |
+-----+-----+
| E002   | Alice Smith |
| E005   | Emma Wilson |
+-----+-----+
2 rows in set (0.00 sec)
```

4. Display emp_no, emp_name, designation, dept_no, and salary of employees in the descending order of salary.

```
SELECT emp_no, emp_name, designation, dept_no, salary FROM Employee
ORDER BY salary DESC;
```

```
mysql> SELECT emp_no, emp_name, designation, dept_no, salary FROM Employee ORDER BY salary DESC;
+-----+-----+-----+-----+-----+
| emp_no | emp_name | designation | dept_no | salary |
+-----+-----+-----+-----+-----+
| E007   | Sophia Garcia | Senior Engineer | D04 | 75000 |
| E002   | Alice Smith | Software Engineer | D02 | 60000 |
| E001   | John Doe | Manager | D01 | 50000 |
| E008   | Daniel Lee | Manager | D04 | 50000 |
| E004   | David Miller | Manager | D01 | 45000 |
| E003   | Bob Brown | HR | D03 | 40000 |
| E005   | Emma Wilson | Computer Assistant | D02 | 30000 |
| E009   | Olivia Martinez | Technician | D05 | 28000 |
| E006   | Michael Johnson | Sales Executive | D03 | 25000 |
| E010   | James Anderson | Support Staff | D05 | 20000 |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

5. Display the emp_no and name of employees whose salary is between 2000 and 5000.

```
SELECT emp_no, emp_name FROM Employee WHERE salary BETWEEN 2000 AND 5000;
```

```
mysql> SELECT emp_no, emp_name FROM Employee WHERE salary BETWEEN 2000 AND 5000;  
Empty set (0.00 sec)
```

6. Display the designations without duplicate values.

```
SELECT DISTINCT designation FROM Employee;
```

```
mysql> SELECT DISTINCT designation FROM Employee;  
+-----+  
| designation |  
+-----+  
| Manager    |  
| Software Engineer |  
| HR         |  
| Computer Assistant |  
| Sales Executive |  
| Senior Engineer |  
| Technician  |  
| Support Staff |  
+-----+  
8 rows in set (0.01 sec)
```

7. Change the salary of employees to 45000 whose designation is 'Manager'.

UPDATE Employee SET salary = 45000 WHERE designation = 'Manager';

```
mysql> UPDATE Employee SET salary = 45000 WHERE designation = 'Manager';  
Query OK, 2 rows affected (0.04 sec)  
Rows matched: 3  Changed: 2  Warnings: 0
```

8. Change the mobile number of employees named John.

UPDATE Employee SET mobile_no = 9999999999 WHERE emp_name = 'John Doe';

```
mysql> UPDATE Employee SET mobile_no = 9999999999 WHERE emp_name = 'John Doe';  
Query OK, 1 row affected (0.07 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

9. Delete all employees whose salary is equal to Rs.7000.

DELETE FROM Employee WHERE salary = 7000;

```
mysql> DELETE FROM Employee WHERE salary = 7000;  
Query OK, 0 rows affected (0.00 sec)
```

10. Retrieve the name and mobile number of all employees whose name starts with "A".

SELECT emp_name, mobile_no FROM Employee WHERE emp_name LIKE 'A%';

```
mysql> SELECT emp_name, mobile_no FROM Employee WHERE emp_name LIKE 'A%';
+-----+-----+
| emp_name | mobile_no |
+-----+-----+
| Alice Smith | 8765432109 |
+-----+-----+
1 row in set (0.00 sec)
```

11. Display the details of the employee whose name has at least three characters and salary greater than 20000.

SELECT * FROM Employee WHERE LENGTH(emp_name) >= 3 AND salary > 20000;

```
mysql> SELECT * FROM Employee WHERE LENGTH(emp_name) >= 3 AND salary > 20000;
+-----+-----+-----+-----+-----+-----+-----+-----+
| emp_no | emp_name | dob | address | mobile_no | dept_no | salary | Designation |
+-----+-----+-----+-----+-----+-----+-----+-----+
| E001 | John Doe | 1985-05-12 | New York | 9999999999 | D01 | 45000 | Manager |
| E002 | Alice Smith | 1990-07-22 | California | 8765432109 | D02 | 60000 | Software Engineer |
| E003 | Bob Brown | 1988-11-15 | Texas | 7654321098 | D03 | 40000 | HR |
| E004 | David Miller | 1995-03-30 | Florida | 6543210987 | D01 | 45000 | Manager |
| E005 | Emma Wilson | 1992-12-20 | Nevada | 5432109876 | D02 | 30000 | Computer Assistant |
| E006 | Michael Johnson | 1987-08-10 | Arizona | 4321098765 | D03 | 25000 | Sales Executive |
| E007 | Sophia Garcia | 1993-06-25 | Illinois | 3210987654 | D04 | 75000 | Senior Engineer |
| E008 | Daniel Lee | 1989-02-14 | Michigan | 2109876543 | D04 | 45000 | Manager |
| E009 | Olivia Martinez | 1991-09-05 | Ohio | 1098765432 | D05 | 28000 | Technician |
+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```


12. Display the details of employees with empid 'emp1', 'emp2', and 'emp6'.

`SELECT * FROM Employee WHERE emp_no IN ('E001', 'E002', 'E006');`

```
mysql> SELECT * FROM Employee WHERE emp_no IN ('E001', 'E002', 'E006');
+-----+-----+-----+-----+-----+-----+-----+-----+
| emp_no | emp_name | dob       | address | mobile_no | dept_no | salary | Designation |
+-----+-----+-----+-----+-----+-----+-----+-----+
| E001   | John Doe | 1985-05-12 | New York | 9999999999 | D01     | 45000  | Manager     |
| E002   | Alice Smith | 1990-07-22 | California | 8765432109 | D02     | 60000  | Software Engineer |
| E006   | Michael Johnson | 1987-09-10 | Arizona | 4321098765 | D03     | 25000  | Sales Executive |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

13. Display employee name and employee id of those who have salary between 120000 and 300000.

`SELECT emp_no, emp_name FROM Employee WHERE salary BETWEEN 120000 AND 300000;`

```
mysql> SELECT emp_no, emp_name FROM Employee WHERE salary BETWEEN 120000 AND 300000;
Empty set (0.01 sec)
```

14. Display the details of employees whose designation is 'Manager' or 'Computer Assistant'.

SELECT * FROM Employee WHERE designation IN ('Manager', 'Computer Assistant');

```
mysql> SELECT * FROM Employee WHERE designation IN ('Manager', 'Computer Assistant');
+-----+-----+-----+-----+-----+-----+-----+-----+
| emp_no | emp_name | dob       | address | mobile_no | dept_no | salary | Designation |
+-----+-----+-----+-----+-----+-----+-----+-----+
| E001   | John Doe | 1985-05-12 | New York | 9999999999 | D01     | 45000 | Manager     |
| E004   | David Miller | 1995-03-30 | Florida | 6543210987 | D01     | 45000 | Manager     |
| E005   | Emma Wilson | 1992-12-20 | Nevada | 5432109876 | D02     | 30000 | Computer Assistant |
| E008   | Daniel Lee | 1989-02-14 | Michigan | 2109876543 | D04     | 45000 | Manager     |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

15. Display how many employees work for each department.

SELECT dept_no, COUNT(*) AS total_employees FROM Employee GROUP BY dept_no;

```
mysql> SELECT dept_no, COUNT(*) AS total_employees FROM Employee GROUP BY dept_no;
+-----+-----+
| dept_no | total_employees |
+-----+-----+
| D01     | 2               |
| D02     | 2               |
| D03     | 2               |
| D04     | 2               |
| D05     | 2               |
+-----+-----+
5 rows in set (0.00 sec)
```

16. Display the average salary of employees in each department.

SELECT dept_no, AVG(salary) AS average_salary FROM Employee GROUP BY dept_no;

```
mysql> SELECT dept_no, AVG(salary) AS average_salary FROM Employee GROUP BY dept_no;
```

dept_no	average_salary
D01	45000.0000
D02	45000.0000
D03	32500.0000
D04	60000.0000
D05	24000.0000

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

17. Display the total salary of employees in each department.

SELECT dept_no, SUM(salary) AS total_salary FROM Employee GROUP BY dept_no;

```
mysql> SELECT dept_no, SUM(salary) AS total_salary FROM Employee GROUP BY dept_no;
```

dept_no	total_salary
D01	90000
D02	90000
D03	65000
D04	120000
D05	48000

```
5 rows in set (0.01 sec)
```

18. Display the highest and lowest salary of employees in each department.

SELECT dept_no, MAX(salary) AS highest_salary, MIN(salary) AS lowest_salary FROM Employee GROUP BY dept_no;

```
mysql> SELECT dept_no, MAX(salary) AS highest_salary, MIN(salary) AS lowest_salary FROM Employee GROUP BY
-> dept_no;
```

dept_no	highest_salary	lowest_salary
D01	45000	45000
D02	60000	30000
D03	40000	25000
D04	75000	45000
D05	28000	20000

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

19. Display the average salary of employees in all departments except department 'D05'.

`SELECT dept_no, AVG(salary) AS average_salary FROM Employee WHERE dept_no <> 'D05' GROUP BY dept_no;`

```
mysql> SELECT dept_no, AVG(salary) AS average_salary FROM Employee WHERE dept_no <> 'D05' GROUP BY dept_no;
```

dept_no	average_salary
D01	45000.0000
D02	45000.0000
D03	32500.0000
D04	60000.0000

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

20. Display the average salary of employees in all departments except department 'D01' and show those where the average salary is greater than 20000 in ascending order.

`SELECT dept_no, AVG(salary) AS average_salary FROM Employee WHERE dept_no <> 'D01' GROUP BY dept_no HAVING AVG(salary) > 20000 ORDER BY average_salary ASC;`

```
mysql> SELECT dept_no, AVG(salary) AS average_salary FROM Employee WHERE dept_no <> 'D01' GROUP BY dept_no  
-> HAVING AVG(salary) > 20000 ORDER BY average_salary ASC;
```

dept_no	average_salary
D05	24000.0000
D03	32500.0000
D02	45000.0000
D04	60000.0000

4 rows in set (0.01 sec)

```
mysql>
```