

## DMS Practical Exam Question bank

1.1. Create the following database schema:

PATIENT (Patient\_id, name, gender, doc\_id, date, age)

DOCTOR (Doc\_id, doc\_name, drug\_id, experience)

DRUGS (Drug\_id, drug\_name, price, expiry\_date)

2. List all drugs in ascending order of expiry date.

3. List information of the patients' who have visited in the year 2017.

4. Create a view displaying information of the doctor, patient and the drug that the doctor prescribes based on the patients' age.

5. Display the information of the doctor which has highest experience and lowest experience in terms of years.

ANS:

```
mysql> CREATE DATABASE P1;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> USE P1;
```

```
Database changed
```

```
mysql> CREATE TABLE PATIENT (Patient_id INT PRIMARY KEY, name varchar(20), gender CHAR, doc_id INT, date DATE, age INT);
```

```
Query OK, 0 rows affected (0.27 sec)
```

```
mysql> CREATE TABLE DOCTOR (Doc_id INT PRIMARY KEY, doc_name varchar(20), drug_id INT, experience varchar(20));
```

```
Query OK, 0 rows affected (0.30 sec)
```

```
mysql> CREATE TABLE DRUGS (Drug_id INT PRIMARY KEY, drug_name varchar(20), price INT, expiry_date DATE);
```

```
Query OK, 0 rows affected (0.28 sec)
```

```
mysql> INSERT INTO PATIENT VALUES (01, 'Rajesh', 'M', 1452, '2017-02-02', 45);
```

```
Query OK, 1 row affected (0.11 sec)
```

```
mysql> INSERT INTO PATIENT VALUES (02, 'Suresh', 'M', 1451, '2017-03-02', 49);
```

```
Query OK, 1 row affected (0.11 sec)
```

```
mysql> INSERT INTO PATIENT VALUES (03, 'Seeta', 'F', 1450, '2017-04-02', 50);
```

```
Query OK, 1 row affected (0.11 sec)
```

```
mysql> INSERT INTO PATIENT VALUES (04, 'Geeta', 'F', 1453, '2017-04-03', 48);
```

```
Query OK, 1 row affected (0.14 sec)
```

```
mysql> INSERT INTO DOCTOR VALUES (1450, 'Allan', 123456, '3 Years');
```

```
Query OK, 1 row affected (0.11 sec)
```

```
mysql> INSERT INTO DOCTOR VALUES (1451, 'Leslie', 516546, '1 Years');
```

```
Query OK, 1 row affected (0.11 sec)
```

```
mysql> INSERT INTO DOCTOR VALUES (1452, 'Joseph', 897945, '2 Years');
```

```
Query OK, 1 row affected (0.08 sec)
```

```
mysql> INSERT INTO DOCTOR VALUES (1453, 'Diesel', 798778, '4 Years');
```

```
Query OK, 1 row affected (0.13 sec)
```

```
mysql> INSERT INTO DRUGS VALUES (123456, 'Crocine Advance', 50, '2017-11-02');
```

```
Query OK, 1 row affected (0.48 sec)
```

```
mysql> INSERT INTO DRUGS VALUES (516546, 'Bournvita', 100, '2017-12-12');
```

```
Query OK, 1 row affected (0.09 sec)
```

```
mysql> INSERT INTO DRUGS VALUES (897945, 'Glycogel', 150, '2017-12-31');
```

```
Query OK, 1 row affected (0.09 sec)
```

```
mysql> INSERT INTO DRUGS VALUES (798778, 'Supergel', 160, '2018-01-11');
```

```
Query OK, 1 row affected (0.11 sec)
```

```
mysql> SELECT drug_name, expiry_date
```

```
-> FROM DRUGS
```

```
-> ORDER BY expiry_date;
```

```
+-----+-----+
| drug_name | expiry_date |
+-----+-----+
| Crocin Advance | 2017-11-02 |
| Bournvita | 2017-12-12 |
| Glycogel | 2017-12-31 |
| Supergel | 2018-01-11 |
+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> SELECT *
-> FROM PATIENT
-> WHERE date between '2017-01-01' AND '2017-12-31';
```

Patient_id	name	gender	doc_id	date	age
1	Rajesh	M	1452	2017-02-02	45
2	Suresh	M	1451	2017-03-02	49
3	Seeta	F	1450	2017-04-02	50
4	Geeta	F	1453	2017-04-03	48

4 rows in set (0.03 sec)

```
mysql> CREATE VIEW INFORMATION1 AS
```

```
-> SELECT Doctor.Doc_id,Doctor.doc_name,Patient.Patient_id,Patient.name,Drugs.drug_name,Patient.age
-> FROM PATIENT,DOCTOR,DRUGS
-> WHERE Doctor.Doc_id=Patient.doc_id
-> AND
-> Drugs Drug_id=Doctor.drug_id
-> GROUP BY age;
```

Query OK, 0 rows affected (0.34 sec)

```
mysql> select * from Information1;
```

Doc_id	doc_name	Patient_id	name	drug_name	age
1452	Joseph	1	Rajesh	Glycogel	45
1453	Diesel	4	Geeta	Supergel	48
1451	Leslie	2	Suresh	Bournvita	49
1450	Allan	3	Seeta	Crocine Advance	50

4 rows in set (0.08 sec)

```
mysql> SELECT Doc_id,doc_name,experience as Highest_Experience FROM DOCTOR WHERE experience=(SELECT
MAX(experience) FROM DOCTOR);
```

Doc_id	doc_name	Highest_Experience
1453	Diesel	4 Years

1 row in set (0.00 sec)

```
mysql> SELECT Doc_id,doc_name,experience as Lowest_Experience FROM DOCTOR WHERE experience=(SELECT
MIN(experience) FROM DOCTOR);
```

Doc_id	doc_name	Lowest_Experience
1451	Leslie	1 Years

1 row in set (0.00 sec)

2.1. Create the following database schema:

MOVIES (title, movie\_id, year, length, type, studio\_name, star\_name)

STUDIO (studio\_name, city)

STARS (star\_name, address)

2. Find all movies that were shot at 'Filmcity' studio.

3. Find the information of all studios that are located in the same city where studio 'Rajhans' is located.

4. List how many stars live in 'Bangalore' city.

5. Find the 'comedy' movies that were released in the year 2011.

```
mysql> CREATE DATABASE P1;
Query OK, 1 row affected (0.00 sec)
mysql> use p2;
Database changed
mysql> INSERT INTO MOVIES VALUES('PK',01,2011,'1 HR','comedy','Filmcity','Aamir Khan');
Query OK, 1 row affected (0.08 sec)
mysql> INSERT INTO MOVIES VALUES('DANGAL',02,2012,'2 HR','comedy','Filmcity','Aamir Khan');
Query OK, 1 row affected (0.11 sec)
mysql> INSERT INTO MOVIES VALUES('BAHIBALI 1',03,2011,'2 HR','action','Rajhans','Prabhas');
Query OK, 1 row affected (0.17 sec)
mysql> INSERT INTO MOVIES VALUES('BAHIBALI 2',04,2012,'2 HR','action','Rajhans','Prabhas');
Query OK, 1 row affected (0.09 sec)
mysql> INSERT INTO STUDIO VALUES('Rajhans','Banglore');
Query OK, 1 row affected (0.08 sec)
mysql> INSERT INTO STUDIO VALUES('Filmcity','Mumbai');
Query OK, 1 row affected (0.08 sec)
mysql> INSERT INTO STARS VALUES('Aamir Khan','Mumbai');
Query OK, 1 row affected (0.08 sec)
mysql> INSERT INTO STARS VALUES('Prabhas','Banglore');
Query OK, 1 row affected (0.05 sec)
```

```
mysql> SELECT title
-> FROM MOVIES
-> WHERE studio_name='Filmcity';
```

```
+-----+
| title |
+-----+
| PK    |
| DANGAL|
+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT studio_name
-> FROM STUDIO
-> WHERE city=(SELECT city FROM STUDIO WHERE studio_name='Rajhans');
```

```
+-----+
| studio_name |
+-----+
| Rajhans    |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT star_name
-> FROM STARS
-> WHERE address='Banglore';
```

```
+-----+
| star_name |
+-----+
| Prabhas   |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT title
-> FROM MOVIES
-> WHERE year=2011
```

-> AND

-> type='comedy';

+-----+

| title |

+-----+

| PK |

+-----+

1 row in set (0.00 sec)

3.1. Create the following database schema:

EMPLOYEE (name, address, salary, emp\_id, plant\_no)

MACHINE (machine\_no, type, plant\_no)

2. Find the employee that has maximum salary.

3. Display all employees plant wise.

4. Display information of the machine which belongs to the same plant to which employee E104 belongs.

5. Display machine information where type of machine is 'Mill'.

```
mysql> INSERT INTO EMPLOYEE VALUES('Ashok','Mumbai',60000,'E100',01);
```

Query OK, 1 row affected (0.36 sec)

```
mysql> INSERT INTO EMPLOYEE VALUES('Raman','Mumbai',65000,'E101',02);
```

Query OK, 1 row affected (0.08 sec)

```
mysql> INSERT INTO EMPLOYEE VALUES('Rajan','Banglore',45000,'E102',03);
```

Query OK, 1 row affected (0.34 sec)

```
mysql> INSERT INTO EMPLOYEE VALUES('Ramesh','Hyderabad',75000,'E104',04);
```

Query OK, 1 row affected (0.08 sec)

```
mysql> INSERT INTO MACHINE VALUES(1230,'Mill',01);
```

Query OK, 1 row affected (0.08 sec)

```
mysql> INSERT INTO MACHINE VALUES(1231,'Industrial',02);
```

Query OK, 1 row affected (0.08 sec)

```
mysql> INSERT INTO MACHINE VALUES(1232,'Mill',03);
```

Query OK, 1 row affected (0.06 sec)

```
mysql> INSERT INTO MACHINE VALUES(1233,'Industrial',04);
```

Query OK, 1 row affected (0.08 sec)

```
mysql> SELECT emp_id,name,salary as Maximum_Salary
-> FROM EMPLOYEE
-> WHERE salary=(SELECT MAX(salary) FROM EMPLOYEE);
```

```
+-----+-----+
| emp_id | name | Maximum_Salary |
+-----+-----+
| E104   | Ramesh | 75000 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT emp_id,name,plant_no
-> FROM EMPLOYEE
-> GROUP BY plant_no;
```

```
+-----+-----+
| emp_id | name | plant_no |
+-----+-----+
| E100   | Ashok | 1 |
| E101   | Raman | 2 |
| E102   | Rajan | 3 |
| E104   | Ramesh | 4 |
+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> SELECT machine_no,type
-> FROM MACHINE
-> WHERE plant_no=(SELECT plant_no FROM EMPLOYEE WHERE emp_id='E104');
```

```
+-----+-----+
| machine_no | type |
+-----+-----+
| 1233       | Industrial |
+-----+-----+
```

1 row in set (0.00 sec)

```
mysql> SELECT machine_no,type  
-> FROM MACHINE  
-> WHERE type='Mill';
```

machine_no	type
1230	Mill
1232	Mill

2 rows in set (0.00 sec)

- 4.1. Create the following database schema:  
 BOOK (book\_id, author, topic, price)  
 BOOKSTORE (store\_id, city, qtysold)  
 STOCK (store\_id, book\_id, qty\_in\_stock)
2. Give the details of the book with stock quantity greater than 35.
3. List the book information in descending order of price.
4. Find the book whose authorname starts with letter 'S'.
5. Implement the constraint that the quantity in stock never falls below 5.

```
mysql> INSERT INTO BOOK VALUES(01,'Swati','Database',350);
Query OK, 1 row affected (0.08 sec)
```

```
mysql> INSERT INTO BOOK VALUES(02,'Reema','Structure',300);
Query OK, 1 row affected (0.09 sec)
```

```
mysql> INSERT INTO BOOK VALUES(03,'Albert','Java',250);
Query OK, 1 row affected (0.05 sec)
```

```
mysql> INSERT INTO BOOK VALUES(04,'MSBSHSE','English',200);
Query OK, 1 row affected (0.08 sec)
```

```
mysql> INSERT INTO BOOKSTORE VALUES(01,'Mumbai',35);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> INSERT INTO BOOKSTORE VALUES(02,'Aurangabad',50);
Query OK, 1 row affected (0.11 sec)
```

```
mysql> INSERT INTO BOOKSTORE VALUES(03,'Hyderabad',60);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> INSERT INTO BOOKSTORE VALUES(04,'Vizak',70);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> INSERT INTO STOCK VALUES(01,01,36);
Query OK, 1 row affected (0.06 sec)
```

```
mysql> INSERT INTO STOCK VALUES(02,02,30);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> INSERT INTO STOCK VALUES(03,03,40);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> INSERT INTO STOCK VALUES(04,04,10);
Query OK, 1 row affected (0.05 sec)
```

```
mysql> SELECT BOOK.book_id,BOOK.author,BOOK.topic,BOOK.price
-> FROM BOOK,STOCK
-> WHERE BOOK.book_id=STOCK.book_id
-> AND
-> qty_in_stock>35;
```

```
+-----+-----+-----+-----+
| book_id | author | topic | price |
+-----+-----+-----+-----+
|    1    | Swati  | Database | 350 |
|    3    | Albert | Java   | 250 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT book_id,author,topic,price
-> FROM BOOK
-> ORDER BY price DESC;
```

```
+-----+-----+-----+-----+
| book_id | author | topic | price |
+-----+-----+-----+-----+
```

1	Swati	Database	350
2	Reema	Structure	300
3	Albert	Java	250
4	MSBSHSE	English	200

-----+

4 rows in set (0.00 sec)

mysql> SELECT book\_id,author,topic,price

-> FROM BOOK

-> WHERE author REGEXP '[S].\*\$';

-----+

book_id	author	topic	price
1	Swati	Database	350

-----+

1 row in set (0.00 sec)

-----+

1 row in set (0.00 sec)

mysql> ALTER TABLE STOCK ADD CHECK(qty\_in\_stock>=5);

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0



5.1. Create the following database schema:

CLIENT (client\_no, name, city, balance\_due)

PRODUCT (prod\_no, profit\_percentage, qty\_on\_hand, selling\_price)

PURCHASE (client\_no, prod\_no, qty)

2. Find the clients who have purchased a product with quantity greater than 15.
3. Find all clients who live in 'Panvel'.
4. Find all products where profit percentage is greater than 50.
5. Create a view that displays client name and number of products purchased by him.

ANS:

```
>SELECT CLIENT.client_no,name,city,balance_due,PURCHASE.prod_no
>FROM CLIENT,PURCHASE
>WHERE CLIENT.client_no=PURCHASE.client_no
>AND
>qty>50;
```

```
>SELECT client_no,name,city
>FROM CLIENT
>WHERE city='Panvel';
```

```
>SELECT prod_no,profit_percentage,selling_price,qty_on_hand
>FROM PRODUCT
>WHERE profit_percentage>50;
```

```
>CREATE VIEW PURCHASED AS
>SELECT CLIENT.client_no,CLIENT.name,PURCHASE.qty AS No_of_Products
>FROM CLIENT,PURCHASE
>WHERE CLIENT.client_no=PURCHASE.client_no;
```

6.1. Create the following database schema:

STUDENT (stu\_no, name, date\_of\_birth)

SUBJECT (sub\_code, sub\_name)

ENROLLMENT (stu\_no, sub\_code, marks)

2. Find the name of the student with highest marks.

3. Display names of the students according to their subjects.

4. Create a view to display student name and subject name where marks obtained are above 75.

5. Display records of students where date of birth is null.

```
>SELECT STUDENT.stu_no,STUDENT.name,STUDENT.marks as Highest_Marks
>FROM STUDENT,ENROLLMENT
>WHERE STUDENT.stu_no=ENROLLMENT.stu_no
>AND
>STUDENT.marks=(SELECT MAX(marks) FROM ENROLLMENT);
```

```
>SELECT STUDENT.stu_no,STUDENT.name,SUBJECT.sub_name
>FROM STUDENT,SUBJECT,ENROLLMENT
>WHERE STUDENT.stu_no=ENROLLMENT.stu_no
>AND
>SUBJECT.sub_code=ENROLLMENT.sub_code
>GROUP BY sub_name;
```

```
>CREATE VIEW DETAILS AS
>SELECT STUDENT.name,SUBJECT.sub_name,ENROLLMENT.marks
>FROM STUDENT,SUBJECT,ENROLLMENT
>WHERE STUDENT.stu_no=ENROLLMENT.stu_no
>AND
>SUBJECT.sub_code=ENROLLMENT.sub_code
>AND
>ENROLLMENT.marks>75;
```

```
>SELECT stu_no,name,date_of_birth
>FROM STUDENT
>WHERE date_of_birth=null;
```

7.1. Create the following database schema:

SUPPLIER (supplier\_id, name, city)

PART (part\_no, part\_name, supplier\_id, color, cost)

2. Find the suppliers who live in the same city.
3. Display details of the part that has maximum cost.
4. Display all part details belonging to particular supplier.
5. Write a stored procedure to display part number when supplier id is given.

```
>SELECT supplier_id,name,city  
>FROM SUPPLIER s1, SUPPLIER s2  
>WHERE s1.city=s2.city;
```

```
>SELECT part_no,part_name,color,cost  
>FROM PART  
>WHERE cost=(SELECT MAX(cost) FROM PART);
```

```
>SELECT SUPPLIER.name,PART.part_no,PART.part_name,PART.supplier_id,PART.color,PART.cost  
>FROM SUPPLIER,PART  
>WHERE SUPPLIER.supplier_id=PART.supplier_id  
>GROUP BY supplier_id;
```

```
Create PROCEDURE GetPartNumber(  
@supplier_id INT)  
AS  
BEGIN  
SELECT part_no  
FROM PART  
WHERE supplier_id=@supplier_id  
END
```

8.1. Create the following database schema:

MEMBERS (mem\_id, name, age, gender)

BOOKS (book\_id, title, author, price, publisher)

RESERVES (mem\_id, book\_id, date)

2. Find the members who have reserved books between 01 July 2013 and 15 July 2013.
3. Find number of female members.
4. Display details of books reserved by members having age above 55.
5. Perform left outer join on BOOKS and MEMBERS.

```
>SELECT MEMBERS.mem_id,name,age,gender
>FROM MEMBERS,RESERVES
>WHERE MEMBERS.mem_id=RESERVES.mem_id
>AND
>date between '2013-07-01' AND '2013-07-15';
```

```
>SELECT COUNT(gender)
>FROM MEMBERS
>WHERE gender='female';
```

```
>SELECT MEMBERS.mem_id,name,BOOKS.book_id,title,author,price,publisher
>FROM MEMBERS,BOOKS,RESERVES
>WHERE MEMBERS.mem_id=RESERVES.mem_id
>AND
>BOOKS.book_id=RESERVES.book_id
>AND
>age>55;
```

```
>SELECT MEMBERS.mem_id,MEMBERS.name,BOOKS.title
>FROM BOOKS LEFT JOIN MEMBERS ON
>MEMBERS.mem_id=BOOKS.book_id
>ORDER BY MEMBERS.name;
```

9.1. Create the following database schema:

STUDENT (stu\_code, name, subject, prof\_code)

PROFESSOR (prof\_code, prof\_name, qualification, experience)

2. Display details of professors who are either ME or MTech and with at least 8 years of experience.
3. Count number of students appearing for the subject 'DBMS'.
4. Find the professor who teaches course 'Computer Programming'.
5. Write a procedure to display stu\_code given prof\_code.

```
>SELECT prof_code,prof_name,qualification,experience  
>FROM PROFESSORS  
>WHERE experience>=8 AND qualification='ME' OR qualification='MTech';
```

```
>SELECT COUNT(stu_code)  
>FROM STUDENT  
>WHERE subject='DBMS';
```

```
>SELECT PROFESSOR.prof_code,prof_name  
>FROM PROFESSOR,STUDENT  
>WHERE PROFESSOR.prof_code=STUDENT.prof_code  
>AND  
>subject='Computer Programming';
```

```
Create PROCEDURE GetStuCode(  
@prof_code INT)  
AS  
BEGIN  
SELECT stu_code  
FROM STUDENT  
WHERE prof_code=@prof_code  
END
```

10.1. Create the following database schema:

MATCH (match\_id, team1, team2, date, ground, winner)

PLAYER (player\_id, name, country, first\_test\_date)

BATTING (match\_id, player\_id, num\_of\_runs, fours, six)

2. Find the date of the matches played between 'India' and 'Australia'.
3. Find the number of runs, fours and sixes scored by 'Sachin' in M20.
4. Create a view that displays matches played on 'Lords' along with the winner.
5. Find the country of the player who has scored highest number of runs.

```
>SELECT date
>FROM MATCH
>WHERE team1='India' OR team1='Australia'
>AND
>team2='India' OR team2='Australia';
```

```
>SELECT PLAYER.player_id,name,num_of_runs,fours,six
>FROM PLAYER,BATTING
>WHERE PLAYER.player_id=BATTING.player_id
>AND
>BATTING.match_id='M20';
```

```
>CREATE VIEW LORDS AS
>SELECT match_id,team1,team2,date,ground,winner
>FROM MATCH
>WHERE ground='Lords';
```

```
>SELECT name,country
>FROM BATTING,PLAYER
>WHERE PLAYER.player_id=BATTING.player_id
>AND
>no_of_runs=(SELECT MAX(no_of_runs) FROM BATTING);
```