

## STRUCTURED QUERY LANGUAGE

### TYPE A : VERY SHORT ANSWER QUESTIONS

<b>1.</b>	<b>What is SQL? What are the various subdivisions of SQL?</b>				
<b>Ans.</b>	SQL means Structured Query Language. It is the set of commands that is recognized by all RDBMS. Data Definition Language (DDL) Data Manipulation Language (DML) Data Control Language (DCL)				
<b>2.</b>	<b>Give examples of some DDL commands and some DML commands.</b>				
<b>Ans.</b>	DDL Commands 1) CREATE 2) ALTER 3) DROP DML Commands 1) INSERT INTO 2) DELETE 3) UPDATE				
<b>3.</b>	<b>What is the difference between column constraint and table constraint? Name some database integrity constrains.</b>				
<b>Ans.</b>	The difference between column constraint and table constraint is that column constraint applies only to individual columns, whereas table constraints apply to groups of one or more columns. Following are the few of database integrity constrains: ➤ Unique constraint ➤ Primary Key constraint ➤ Default constraint ➤ Check constraint				
<b>4.</b>	<b>1. How do following constraint work?</b>				
<b>Ans.</b>	<table style="width: 100%; border: none;"> <tr> <td style="width: 25%;"><b>(i) Unique</b></td> <td style="width: 25%;"><b>(ii) Primary Key</b></td> <td style="width: 25%;"><b>(iii) Default</b></td> <td style="width: 25%;"><b>(iv) Check</b></td> </tr> </table> <p><u>Unique</u>: This constraint ensures that no two rows have the same value in the specified columns. For eg , CREATE TABLE employee (ecode integer NOT NULL UNIQUE, ename char(20),Sex char(2) );</p> <p><u>Primary Key</u>: Primary key does not allow NULL value and Duplicate data in the column which is declared as Primary Key. For eg , CREATE TABLE employee (ecode integer NOT NULL PRIMARY KEY, ename char(20),Sexchar(2) );</p> <p><u>Default</u>: When a user does not enter a value for the column, automatically the defined default value is inserted in field. A column can have only one default value. For eg , CREATE TABLE employee (ecode integer NOT NULL PRIMARY KEY, ename char(20), Sexchar(2), Grade char(2) DEFAULT = 'E1' );</p> <p><u>Check</u>: This constraint limits values that can inserted into a column of table. For eg , CREATE TABLE employee (ecode integer NOT NULL PRIMARY KEY, ename char(20),Sex char(2) , Grade char(2) DEFAULT = 'E1' , Gross decimal CHECK (gross &gt; 2000) );</p>	<b>(i) Unique</b>	<b>(ii) Primary Key</b>	<b>(iii) Default</b>	<b>(iv) Check</b>
<b>(i) Unique</b>	<b>(ii) Primary Key</b>	<b>(iii) Default</b>	<b>(iv) Check</b>		
<b>5.</b>	<b>Compare DISTINCT and ALL keywords when used with SELECT command.</b>				
<b>Ans.</b>	DISTINCT keyword is used to restrict the duplicate rows from the results of a SELECT statement. ALL keyword retains the duplicate rows, by default ALL keyword is use by SELECT statement.				
<b>6.</b>	<b>What is wrong with the following statement? Write the corrected form of this query :</b>				
	<b>SELECT * FROM employee</b> <b>WHERE grade = NULL ;</b>				

**Ans.** IS NULL should be used in place of = NULL. Following is the correct statement :  
 SELECT \* FROM employee  
 WHERE grade IS NULL ;

**7. What is the difference between where and having clause ?**

<u>WHERE CLAUSE</u>	<u>HAVING CLAUSE</u>
Places conditions on individual rows.	Places conditions on groups.
Cannot include aggregate function.	Can include aggregate function.
<b>For eg.</b> SELECT * FROM student WHERE Rno >=10;	<b>For eg.</b> SELECT AVG(marks) FROM student GROUP BY grade HAVING grade = 'B1';

**8. What is difference between working of the following functions?**  
**Count(\*), Count (<column-name>), Count (DISTINCT), Count (ALL)**

**Ans.** Count(\*):- The COUNT(\*) function returns the number of records in a table:  
 SELECT COUNT(\*) FROM student;

Count (<column-name>):- The COUNT(<column-name>) function returns the number of values (NULL values will not be counted) of the specified column:  
 SELECT COUNT(name) FROM student;

Count (DISTINCT):- The COUNT(DISTINCT column\_name) function returns the number of distinct values of the specified column:  
 SELECT COUNT(DISTINCT city) FROM student;

Count (ALL):- to count the number of non-null values in column dept, i.e. counting repetitions too.  
 SELECT COUNT(ALL) FROM student;

**9. 1. What is the difference between SELECT INTO and CREATE VIEW commands?**

SELECT INTO command	CREATE VIEW command
SELECT INTO creates a new table by extracting data from other table.	The CREATE VIEW creates view from a table.
Resides physically in the database.	Not a part of the database's physical representation.
Used to create backup copies of tables.	Not used for backup purpose.
<b>For eg.</b> SELECT Lastname, FirstName INTO Person_Backup FROM Persons;	<b>For eg.</b> CREATE VIEW v_student AS SELECT Roll_no, Name, Class FROM student;

**10. What are views? When can a view be updated?**

**Ans.** A view is, in essence, a virtual table. It does not physically exist. Rather, it is created by a query joining one or more tables.  
 In following condition a view can be updated :

- ✓ If it has only one base table.
- ✓ If select statement used in view contains group by clause then we cant update view.

**11. What is the condition of dropping a table?**

**Ans.** There is a one condition for dropping a table that is a table must be an empty table which we want to drop.

**Note: For the following questions consider the tables EMP given in book.**

**12. Insert a record with suitable data in the table EMP, tabing system date as the Hiredate.**

**Ans.** INSERT INTO emp VALUES  
 (1101, 'ROBIN', 'CLERK', 7902, curdate(), 5000.00, 500.00, 10);

**13. To create a table DEPTO30 to hold the employee numbers, names, jobs and salaries of employee in department with DeptNo = 30.**

<b>Ans.</b>	CREATE TABLE DEPTP30 AS(SELECT EmpNo, EmpName, Job, Sal FROM EMP WHERE DeptNo=30);																		
<b>14.</b>	<b>Display names all employees whose names include either of the substring "TH" or "LL".</b>																		
<b>Ans.</b>	SELECT empname FROM emp WHERE(empname LIKE '%TH%' OR empname LIKE '%LL%');																		
<b>15.</b>	<b>Display data for all CLERKS who earn between 1000 and 2000.</b>																		
<b>Ans.</b>	SELECT * FROM emp WHERE((job LIKE 'clerk') AND (sal BETWEEN 1000 AND 2000));																		
<b>16.</b>	<b>Display data for all employees sorted by their department, seniority and salary.</b>																		
<b>Ans.</b>	SELECT * FROM emp ORDER BY deptno, hiredate, sal;																		
<b>17.</b>	<b>Write a SQL statement to list EmpNo, EmpName, DeptNo, for all the employees. This information is should be sorted on EmpName.</b>																		
<b>Ans.</b>	SELECT empno, empname,deptno FROM emp ORDER BY empname;																		
<b>18.</b>	<b>Write SQL statement for : Find all the employees who have no manager.</b>																		
<b>Ans.</b>	SELECT empname FROM emp WHERE mgr IS NULL;																		
<b>19.</b>	<b>Write a SQL statement (s) to list all employees in the following format:</b>																		
	<table border="1"> <thead> <tr> <th>EMPLOYEE</th> <th>WORKS IN DEPARTMENT</th> <th>DeptNo</th> </tr> </thead> <tbody> <tr> <td>7369-SMITH</td> <td>WORKS IN DEPARTMENT</td> <td>20</td> </tr> <tr> <td>7300-SUDHIR</td> <td>WORKS IN DEPARTMENT</td> <td>20</td> </tr> <tr> <td>7345-RAJ</td> <td>WORKS IN DEPARTMENT</td> <td>10</td> </tr> <tr> <td>7329-SMITHS</td> <td>WORKS IN DEPARTMENT</td> <td>30</td> </tr> <tr> <td>7234-SANTOSH</td> <td>WORKS IN DEPARTMENT</td> <td>30</td> </tr> </tbody> </table>	EMPLOYEE	WORKS IN DEPARTMENT	DeptNo	7369-SMITH	WORKS IN DEPARTMENT	20	7300-SUDHIR	WORKS IN DEPARTMENT	20	7345-RAJ	WORKS IN DEPARTMENT	10	7329-SMITHS	WORKS IN DEPARTMENT	30	7234-SANTOSH	WORKS IN DEPARTMENT	30
EMPLOYEE	WORKS IN DEPARTMENT	DeptNo																	
7369-SMITH	WORKS IN DEPARTMENT	20																	
7300-SUDHIR	WORKS IN DEPARTMENT	20																	
7345-RAJ	WORKS IN DEPARTMENT	10																	
7329-SMITHS	WORKS IN DEPARTMENT	30																	
7234-SANTOSH	WORKS IN DEPARTMENT	30																	
<b>Ans.</b>	SELECT ename, 'WORKS IN DEPARTMENT', deptno FROM emp;																		
<b>20.</b>	<b>To find all those employees whose job does not start with 'M'.</b>																		
<b>Ans.</b>	SELECT empname FROM emp WHERE job NOT LIKE 'M%';																		
<b>21.</b>	<b>To display all employees who were hired during 1995.</b>																		
<b>Ans.</b>	SELECT ename FROM emp WHERE YEAR(hiredate) = '1995';																		
<b>22.</b>	<b>To display DeptNo, Job, EmpName in reverse order of salary from the EMP table.</b>																		
<b>Ans.</b>	SELECT deptno, job, empname FROM emp ORDER BY sal DESC;																		
<b>23.</b>	<b>List EmpName, Job, Sal for all the employees who have a manager.</b>																		
<b>Ans.</b>	SELECT empname , job, salary from EMP WHERE mgr IS NOT NULL;																		
<b>24.</b>	<b>List the minimum and maximum salary of each job type.</b>																		
<b>Ans.</b>	SELECT job, MIN(sal), MAX(sal) FROM emp GROUP BY job;																		
<b>25.</b>	<b>Show the average salary for all departments with more than 3 people for job.</b>																		
<b>Ans.</b>	SELECT AVG( sal ) FROM emp GROUP BY deptno HAVING COUNT(job)>3;																		
<b>26.</b>	<b>Display only the jobs with maximum salary greater than or equal to 3000.</b>																		
<b>Ans.</b>	SELECT job FROM emp GROUP BY job HAVING MAX(salary)>=3000;																		
<b>27.</b>	<b>Find out number of employee having 'Manager' as job.</b>																		
<b>Ans.</b>	SELECT COUNT(empname) FROM emp WHERE job LIKE 'Manager';																		
<b>28.</b>	<b>Create view Dept20 with EmpName and the Sal of employees for dept 20.</b>																		
<b>Ans.</b>	CREATE VIEW dept20 as SELECT empname, sal FROM emp WHERE deptno=20;																		
<b>29.</b>	<b>Find the average salary and average total remuneration for each job type remember salesman earn commission.</b>																		
<b>Ans.</b>	SELECT AVG( sal ) , AVG( sal + IFNULL( comm, 0 ) ) FROM emp GROUP BY job;																		
<b>30.</b>	<b>What happens if you try to drop a table on which a view exists?</b>																		
<b>Ans.</b>	If we try to drop a table on which a view exist, then the table is dropped but DBMS invalidates these dependent views but does not drop them. We cannot use these views unless we recreate the table or drop and recreate the objects so that they no longer depend on the table.																		
<b>31.</b>	<b>Create a view with one of the columns Salary * 12. Try updating columns of this view.</b>																		
<b>Ans.</b>	CREATE VIEW emp_view (v_empno, v_empname, v_avgsal) AS SELECT empno, empname, salary*12 FROM emp; UPDATE emp_view SET empname = 'MOHAN' WHERE empno=8698;																		
<b>32.</b>	<b>Can you create view of view?</b>																		
<b>Ans.</b>	Yes, We can create view of view.																		

33.	Write a suitable SQL statement to display ALL employees working in New York in the following format : <b>EmpName      Salary      Location</b>
Ans.	SELECT A.empname, A.salary, B.location FROM emp A INNER JOIN dept B on A.deptno=B.deptno WHERE location LIKE 'NewYork';
34.	Write a suitable SQL statement to display employees' name of all the employees of GRADE 3.
Ans.	SELECT empname FROM emp A, salgrade B WHERE grade=3 AND A.empno=B.empno;
35.	Write a suitable SQL statement to find out the total number of employees from EMP table.
Ans.	SELECT count(empname) from EMP;

**TYPE B : SHORT ANSWER QUESTIONS**

1.	Consider the following tables STORE and SUPPLIERS and answer (a) and (b) parts of this question:  Table: STORE																																																
	<table border="1"> <thead> <tr> <th>ItemNo</th> <th>Item</th> <th>Scode</th> <th>Qty</th> <th>Rate</th> <th>LastBuy</th> </tr> </thead> <tbody> <tr> <td>2005</td> <td>Sharpener Classic</td> <td>23</td> <td>60</td> <td>8</td> <td>31-Jun-09</td> </tr> <tr> <td>2003</td> <td>Ball Pen 0.25</td> <td>22</td> <td>50</td> <td>25</td> <td>01-Feb-10</td> </tr> <tr> <td>2002</td> <td>Gel Pen Premium</td> <td>21</td> <td>150</td> <td>12</td> <td>24-Feb-10</td> </tr> <tr> <td>2006</td> <td>Gel Pen Classic</td> <td>21</td> <td>250</td> <td>20</td> <td>11-Mar-09</td> </tr> <tr> <td>2001</td> <td>Eraser Small</td> <td>22</td> <td>220</td> <td>6</td> <td>19-Jan-09</td> </tr> <tr> <td>2004</td> <td>Eraser Big</td> <td>22</td> <td>110</td> <td>8</td> <td>02-Dec-09</td> </tr> <tr> <td>2009</td> <td>Ball Pen 0.5</td> <td>21</td> <td>180</td> <td>18</td> <td>03-Nov-09</td> </tr> </tbody> </table>	ItemNo	Item	Scode	Qty	Rate	LastBuy	2005	Sharpener Classic	23	60	8	31-Jun-09	2003	Ball Pen 0.25	22	50	25	01-Feb-10	2002	Gel Pen Premium	21	150	12	24-Feb-10	2006	Gel Pen Classic	21	250	20	11-Mar-09	2001	Eraser Small	22	220	6	19-Jan-09	2004	Eraser Big	22	110	8	02-Dec-09	2009	Ball Pen 0.5	21	180	18	03-Nov-09
ItemNo	Item	Scode	Qty	Rate	LastBuy																																												
2005	Sharpener Classic	23	60	8	31-Jun-09																																												
2003	Ball Pen 0.25	22	50	25	01-Feb-10																																												
2002	Gel Pen Premium	21	150	12	24-Feb-10																																												
2006	Gel Pen Classic	21	250	20	11-Mar-09																																												
2001	Eraser Small	22	220	6	19-Jan-09																																												
2004	Eraser Big	22	110	8	02-Dec-09																																												
2009	Ball Pen 0.5	21	180	18	03-Nov-09																																												
	Table: SUPPLIERS																																																
	<table border="1"> <thead> <tr> <th>Scode</th> <th>Sname</th> </tr> </thead> <tbody> <tr> <td>21</td> <td>Premium Stationers</td> </tr> <tr> <td>23</td> <td>Soft Plastics</td> </tr> <tr> <td>22</td> <td>Tetra Supply</td> </tr> </tbody> </table>	Scode	Sname	21	Premium Stationers	23	Soft Plastics	22	Tetra Supply																																								
Scode	Sname																																																
21	Premium Stationers																																																
23	Soft Plastics																																																
22	Tetra Supply																																																

(a)	Write SQL commands for the following statements:
(i)	To display details of all the items in the Store table in ascending order of LastBuy.
Ans.	SELECT * FROM STORE ORDER BY LastBuy;
(ii)	To display ItemNo and Item name of those items from Store table, whose Rate is more than 15 Rupees.
Ans.	SELECT ItemNo, Item FROM STORE WHERE Rate >15;
(iii)	To display the details of those items whose Supplier code (Scode) is 22 or Quantity in Store (Qty) is more than 110 from the table Store.
Ans.	SELECT * FROM STORE WHERE Scode = 22 OR Qty >110;
(iv)	To display Minimum Rate of items for each Supplier individually as per Scode from the table Store.
Ans.	SELECT Scode, MIN(Rate) FROM STORE GROUP BY Scode;

<p>(b) (i) Ans.  (ii) Ans.  (iii) Ans.  (iv) Ans.</p>	<p><b>Give the output of the following SQL queries:</b></p> <p><b>SELECT COUNT(DISTINCT Scode) FROM Store;</b>  <u>COUNT(DISTINCT Scode)</u>  3</p> <p><b>SELECT Rate*Qty FROM Store WHERE ItemNo=2004;</b>  <u>RATE*QTY</u>  880</p> <p><b>SELECT Item,Sname FROM Store S, Suppliers P WHERE S.Score=P.Score AND ItemNo=2006;</b>  <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">ITEM</th> <th style="text-align: left;">SNAME</th> </tr> </thead> <tbody> <tr> <td>Gel Pen Classic</td> <td>Premium Stationers</td> </tr> </tbody> </table> </p> <p><b>SELECT MAX(LastBuy) FROM Store;</b>  <u>MAX (LASTBUY)</u>  24-Feb-10</p>	ITEM	SNAME	Gel Pen Classic	Premium Stationers																																												
ITEM	SNAME																																																
Gel Pen Classic	Premium Stationers																																																
<p>2.</p>	<p><b>Consider the following table Item and Customer. Write SQL commands for the statement (i) to (iv) and give outputs for SQL queries (v) to (viii).</b></p> <p style="text-align: center;"><b>Table : ITEM</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>i_ID</th> <th>ItemName</th> <th>Manufacturer</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>PC01</td> <td>Personal Computer</td> <td>ABC</td> <td>35000</td> </tr> <tr> <td>LC05</td> <td>Laptop</td> <td>ABC</td> <td>55000</td> </tr> <tr> <td>Pc03</td> <td>Personal Computer</td> <td>XYZ</td> <td>32000</td> </tr> <tr> <td>Pc06</td> <td>Personal Computer</td> <td>COMP</td> <td>37000</td> </tr> <tr> <td>Lc03</td> <td>Laptop</td> <td>PQR</td> <td>57000</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Table : CUSTOMER</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>C_ID</th> <th>CustomerName</th> <th>City</th> <th>I_ID</th> </tr> </thead> <tbody> <tr> <td>01</td> <td>N Roy</td> <td>Delhi</td> <td>LC03</td> </tr> <tr> <td>06</td> <td>H Singh</td> <td>Mumbai</td> <td>PC03</td> </tr> <tr> <td>12</td> <td>R Pandey</td> <td>Delhi</td> <td>PC06</td> </tr> <tr> <td>15</td> <td>C Sharma</td> <td>Delhi</td> <td>LC03</td> </tr> <tr> <td>16</td> <td>K Agarwal</td> <td>Banglore</td> <td>PC01</td> </tr> </tbody> </table>	i_ID	ItemName	Manufacturer	Price	PC01	Personal Computer	ABC	35000	LC05	Laptop	ABC	55000	Pc03	Personal Computer	XYZ	32000	Pc06	Personal Computer	COMP	37000	Lc03	Laptop	PQR	57000	C_ID	CustomerName	City	I_ID	01	N Roy	Delhi	LC03	06	H Singh	Mumbai	PC03	12	R Pandey	Delhi	PC06	15	C Sharma	Delhi	LC03	16	K Agarwal	Banglore	PC01
i_ID	ItemName	Manufacturer	Price																																														
PC01	Personal Computer	ABC	35000																																														
LC05	Laptop	ABC	55000																																														
Pc03	Personal Computer	XYZ	32000																																														
Pc06	Personal Computer	COMP	37000																																														
Lc03	Laptop	PQR	57000																																														
C_ID	CustomerName	City	I_ID																																														
01	N Roy	Delhi	LC03																																														
06	H Singh	Mumbai	PC03																																														
12	R Pandey	Delhi	PC06																																														
15	C Sharma	Delhi	LC03																																														
16	K Agarwal	Banglore	PC01																																														
<p>(i) Ans.  (ii) Ans.  (iii) Ans.  (iv) Ans.  (v) Ans.</p>	<p><b>To display the details of those Customer whose City is Delhi.</b>  SELECT * FROM CUSTOMER WHERE CITY = 'DELHI';</p> <p><b>To display the details of Item whose Price is in the range of 3500 to 55000 (Both values included).</b>  SELECT * FROM ITEM WHERE PRICE BETWEEN 35000 AND 55000;</p> <p><b>To displa the customerName, City from table Customer, and ItemName and Price from table Item, with their corresponding matching I_ID.</b>  SELECT CUSTOMERNAME,CITY,ITEMNAME,PRICE FROM CUSTOMER A INNER JOIN ITEM B WHERE A.I_ID=B.I_ID;</p> <p><b>To increase the Price of all Items by 1000 in the table Item.</b>  UPDATE ITEM SET PRICE=PRICE+1000;</p> <p><b>SELECT DISTINCT City FROM Customer;</b>  <u>City</u>  Delhi  Mumbai  Banglore</p>																																																

(vi) **SELECT ItemName, Max(Price), Count(\*) FROM Item GROUP BY ItemName;**

Ans. 

<u>Name</u>	<u>Max(Price)</u>	<u>Count(*)</u>
Laptop	58000	2
Personal Computer	38000	3

(vii) **SELECT CustomerName, Manufacturer From Item, Customer WHERE Item.I\_Id=Customer.I\_Id;**

Ans. 

<u>Cname</u>	<u>Manufacturer</u>
N Roy	PQR
H Singh	XYZ
R Pandey	COMP
C Sharma	PQR
K Agarwal	ABC

(viii) **SELECT ItemName, Price \* 100 FROM Item WHERE Manufacturer = 'ABC';**

Ans. 

<u>Name</u>	<u>Price*100</u>
Personal Computer	3600000
Laptop	5600000

3. Consider the following tables. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).

**TABLE : SENDER**

SenderID	SenderName	SenderName	SenderCity
ND01	R Jain	2,ABC Appts	New Delhi
MU02	H Sinha	12, Newtown	Mumbai
MU15	S Jha	27/A, Park Street	Mumbai
ND50	T Prasad	122-K, SDA	New Delhi

**TABLE : RECIPIENT**

RecID	SenderID	RecName	RecAddress	ReCity
KO05	ND01	R Bajpayee	5, Central Avenue	Kolkata
ND08	MU02	S Mohan	116, A vihar	New Delhi
MU19	ND01	H singh	2a, Andheri east	Mumbai
MU32	MU15	P K Swamy	B5, c S Terminus	Mumbai
ND48	ND50	S Tirupathi	13, B1 d, Mayur vihar	New Delhi

(i) **To display the names of all Senders from Mumbai**

Ans. `SELECT SENDERNAME FROM SENDER WHERE SENDERCITY='MUMBAI' ;`

(ii) **To display the RecID, SenderName, SenderAddress, RecName, RecAddress for every Recipient**

Ans. `SELECT RECID, SENDERNAME, SENDERADDRESS, RECNAME, RECADDRESS FROM RECIPIENT A INNER JOIN SENDER B ON A.SENDERID=B.SENDERID;`

(iii) **To display Recipient detail in asending order of RecName**

Ans. `SELECT * FROM RECIPIENT ORDER BY RECNAME;`

(iv) **To display number of Recipients from each city**

Ans. `SELECT RECCITY, COUNT(RECNAME) FROM RECIPIENT GROUP BY RECCITY;`

(v) **SELECT DISTINCT Sendercity FROM Sender;**

Ans. 

<u>Sendercity</u>
New Delhi
Mumbai

<b>(vi)</b>	<b>SELECT A.SenderName, B.RecName FROM Sender A, Recipient B WHERE A.SenderID=B.SenderID AND B.RecCity='Mumbai';</b>						
<b>Ans.</b>	<table border="1"> <tr> <td><u>SenderName</u></td> <td><u>RecName</u></td> </tr> <tr> <td>R Jain</td> <td>H Singh</td> </tr> <tr> <td>S Jha</td> <td>P K Swamy</td> </tr> </table>	<u>SenderName</u>	<u>RecName</u>	R Jain	H Singh	S Jha	P K Swamy
<u>SenderName</u>	<u>RecName</u>						
R Jain	H Singh						
S Jha	P K Swamy						
<b>(vii)</b>	<b>SELECT RecName, RecAddress FROM Recipient WHERE recCity NOT IN('Mumbai', 'Kolkata');</b>						
<b>Ans.</b>	<table border="1"> <tr> <td><u>RecName</u></td> <td><u>RecAddress</u></td> </tr> <tr> <td>S Mahajan</td> <td>116, A Vihar1</td> </tr> <tr> <td>S Tirupati</td> <td>13, Bl D, Mayur Vihar</td> </tr> </table>	<u>RecName</u>	<u>RecAddress</u>	S Mahajan	116, A Vihar1	S Tirupati	13, Bl D, Mayur Vihar
<u>RecName</u>	<u>RecAddress</u>						
S Mahajan	116, A Vihar1						
S Tirupati	13, Bl D, Mayur Vihar						
<b>(viii)</b>	<b>SELECT RecID, RecName FROM Recipient WHERE SenderID='MU02' OR SenderID='ND50';</b>						
<b>Ans.</b>	<table border="1"> <tr> <td><u>RecID</u></td> <td><u>RecName</u></td> </tr> <tr> <td>ND08</td> <td>S Mahajan</td> </tr> <tr> <td>ND48</td> <td>S Tirupati</td> </tr> </table>	<u>RecID</u>	<u>RecName</u>	ND08	S Mahajan	ND48	S Tirupati
<u>RecID</u>	<u>RecName</u>						
ND08	S Mahajan						
ND48	S Tirupati						

**4. (a) What happens if you drop a table on which a view exists?**

**Ans.** If we try to drop a table on which a view exist, then the DBMS like Oracle invalidates these dependent views but does not drop them. We cannot use these views unless we recreate the table or drop and recreate the objects so that they no longer depend on the table.

Note: Write the SQL commands for (b) to (g) and write outputs for SQL commands given in (h) on the basis of table MOV

**Table : MOV**

No.	Title	Type	Rating	Stars	Qty	Price
1	Gone with the Wind	Drama	G	Gable	4	39.95
2	Friday the 13 <sup>th</sup>	Horror	R	Jason	2	69.95
3	Top Gun	Drama	PG	Cruise	7	49.95
4	Splash	Comedy	PG13	Hanks	3	29.95
5	Independence Day	Drama	R	Turner	3	19.95
6	Risky Business	Comedy	R	Cruise	2	44.95
7	Cocoon	Scifi	PG	Ameche	2	31.95
8	Crocodile Dundee	Comedy	PG13	Harris	2	69.95
9	101 Dalmatians	Comedy	G		3	59.95
10	Tootsie	Comedy	PG	Hoffman	1	29.95

**(b) Find the total value of the movie cassettes available in the library.**

**Ans.** SELECT COUNT(TITLE) FROM MOV;

**(c) Display a list of all movies with Price over 20 and sorted by Price.**

**Ans.** SELECT \* FROM MOV WHERE PRICE>20 ORDER BY PRICE;

**(d) Display all the movies sorted by Qty in decreasing order.**

**Ans.** SELECT \* FROM MOV ORDER BY QTY DESC;

**(e) Display a report listing a movie number, current value and replacement value for each movie in the above table. Calculate the replacement value for all movies as**

$$QTY * Price * 1.15$$

**Ans.** SELECT NO,PRICE AS 'CURRENT VALUE', (QTY\*PRICE\*1.15) AS 'REPLACEMENT VALUE' FROM MOV;

**(f) Count the number of movies where Rating is not "G".**

**Ans.** SELECT COUNT(TITLE) FROM MOV WHERE RATING<>'G';

<b>(g)</b>	<b>Insert a new movie in MOV table. Fill all the columns with values.</b>
<b>Ans.</b>	INSERT IN TO MOV VALUES(11,'Republic Day','Drama','R','Turner',3,38.95);
<b>(h)</b>	<b>Give the output of following SQL commands on the basis of table MOV.</b>
<b>(i)</b>	<b>Select AVG(Price) from MOV where Price &lt; 30;</b>
<b>Ans.</b>	<u>AVG(Price)</u> 19.95
<b>(ii)</b>	<b>Select MAX(Price) from MOV where price &gt; 30;</b>
<b>Ans.</b>	<u>MAX(Price)</u> 79.95
<b>(iii)</b>	<b>Select SUM(Price * QTY) from MOV where QTY &lt; 4;</b>
<b>Ans.</b>	<u>SUM(Price*QTY)</u> 791.75
<b>(iv)</b>	<b>Select COUNT(DISTINCT TYPE) from MOV;</b>
<b>Ans.</b>	<u>COUNT(DISTINCT TYPE)</u> 4
<b>5.</b>	<b>Write SQL statement to create EMPLOYEE relation which contains EmpNo, Name, Skill, PayRate.</b>
<b>Ans.</b>	CREATE TABLE EMPLOYEE(EmpNo VARCHAR(10), Name CHAR(20), Skill CHAR(20), PayRate DECIMAL);

<b>6.</b>	<b>Create a table with the under mentioned structures</b>																																																						
	<table border="1"> <thead> <tr> <th colspan="2"><u>Table : EMP</u></th> <th colspan="2"><u>Table : PROJECT</u></th> <th colspan="2"><u>Table : DEPT</u></th> </tr> </thead> <tbody> <tr> <td>EmpNo</td> <td>NUMBER(4)</td> <td>ProjId</td> <td>NUMBER(4)</td> <td>DeptNo</td> <td>NUMBER(2)</td> </tr> <tr> <td>DeptNo</td> <td>NUMBER(2)</td> <td>ProjDesig</td> <td>CHAR(20)</td> <td>DeptName</td> <td>CHAR(12)</td> </tr> <tr> <td>EmpName</td> <td>CHAR(10)</td> <td>ProjStartDT</td> <td>DATE</td> <td>Location</td> <td>CHAR(12)</td> </tr> <tr> <td>Job</td> <td>CHAR(10)</td> <td>ProjEndDT</td> <td>DATE</td> <td></td> <td></td> </tr> <tr> <td>Manager</td> <td>NUMBER(4)</td> <td>BudgetAmount</td> <td>NUMBER(7)</td> <td></td> <td></td> </tr> <tr> <td>Hiredate</td> <td>DATE</td> <td>MaxNoStaff</td> <td>NUMBER(2)</td> <td></td> <td></td> </tr> <tr> <td>Salary</td> <td>NUMBER(7,2)</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Commission</td> <td>NUMBER(7,2)</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	<u>Table : EMP</u>		<u>Table : PROJECT</u>		<u>Table : DEPT</u>		EmpNo	NUMBER(4)	ProjId	NUMBER(4)	DeptNo	NUMBER(2)	DeptNo	NUMBER(2)	ProjDesig	CHAR(20)	DeptName	CHAR(12)	EmpName	CHAR(10)	ProjStartDT	DATE	Location	CHAR(12)	Job	CHAR(10)	ProjEndDT	DATE			Manager	NUMBER(4)	BudgetAmount	NUMBER(7)			Hiredate	DATE	MaxNoStaff	NUMBER(2)			Salary	NUMBER(7,2)					Commission	NUMBER(7,2)				
<u>Table : EMP</u>		<u>Table : PROJECT</u>		<u>Table : DEPT</u>																																																			
EmpNo	NUMBER(4)	ProjId	NUMBER(4)	DeptNo	NUMBER(2)																																																		
DeptNo	NUMBER(2)	ProjDesig	CHAR(20)	DeptName	CHAR(12)																																																		
EmpName	CHAR(10)	ProjStartDT	DATE	Location	CHAR(12)																																																		
Job	CHAR(10)	ProjEndDT	DATE																																																				
Manager	NUMBER(4)	BudgetAmount	NUMBER(7)																																																				
Hiredate	DATE	MaxNoStaff	NUMBER(2)																																																				
Salary	NUMBER(7,2)																																																						
Commission	NUMBER(7,2)																																																						
<b>Ans.</b>	<p>CREATE TABLE <b>EMP</b>(EmpNo INTEGER(4), DeptNo INTEGER(2), EmpName CHAR(10), Job CHAR(10), Manager INTEGER(4), HireDate DATE, Salary DECIMAL(7,2), Commission DECIMAL(7,2));</p> <p>CREATE TABLE <b>PROJECT</b>(ProjId INTEGER(4), ProjDesign CHAR(20), ProjStartDT DATE, ProjEndDT DATE, BudgetAmount INTEGER(7), MaxNoStaff INTEGER(2));</p> <p>CREATE TABLE <b>DEPT</b>(DeptNo INTEGER(2), DeptName CHAR(12), Location CHAR(12));</p>																																																						

<b>7.</b>	<b>Create a table called SALEGRADE with the columns specified below :</b>						
	<table> <tr> <td><b>LowSal</b></td> <td>NUMBER(7,2)</td> </tr> <tr> <td><b>HighSal</b></td> <td>NUMBER(7,2)</td> </tr> <tr> <td><b>Grade</b></td> <td>NUMBER(2)</td> </tr> </table>	<b>LowSal</b>	NUMBER(7,2)	<b>HighSal</b>	NUMBER(7,2)	<b>Grade</b>	NUMBER(2)
<b>LowSal</b>	NUMBER(7,2)						
<b>HighSal</b>	NUMBER(7,2)						
<b>Grade</b>	NUMBER(2)						
<b>Ans.</b>	CREATE TABLE SALEGRADE(LowSal DECIMAL(7,2) CHECK(LowSal>=1000.00), HighSal DECIMAL(7,2) CHECK(HighSal<=10000.00), Grade INTEGER);						

<b>8.</b>	<b>Write SQL commands for (a) to (f) and write the outputs for (g) on the basis of tables FURNITURE and ARRIVALS:</b>																														
	<b>TABLE : FURNITURE</b>																														
	<table border="1"> <thead> <tr> <th>NO</th> <th>ITEMNAME</th> <th>TYPE</th> <th>DATEOFSTOCK</th> <th>PRICE</th> <th>DISCOUNT</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>White lotus</td> <td>Double Bed</td> <td>23/02/02</td> <td>30000</td> <td>25</td> </tr> <tr> <td>2</td> <td>Pink feather</td> <td>Baby cot</td> <td>20/01/02</td> <td>7000</td> <td>20</td> </tr> <tr> <td>3</td> <td>Dolphin</td> <td>Baby cot</td> <td>19/02/02</td> <td>9500</td> <td>20</td> </tr> <tr> <td>4</td> <td>Decent</td> <td>Office Table</td> <td>01/01/02</td> <td>25000</td> <td>30</td> </tr> </tbody> </table>	NO	ITEMNAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT	1	White lotus	Double Bed	23/02/02	30000	25	2	Pink feather	Baby cot	20/01/02	7000	20	3	Dolphin	Baby cot	19/02/02	9500	20	4	Decent	Office Table	01/01/02	25000	30
NO	ITEMNAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT																										
1	White lotus	Double Bed	23/02/02	30000	25																										
2	Pink feather	Baby cot	20/01/02	7000	20																										
3	Dolphin	Baby cot	19/02/02	9500	20																										
4	Decent	Office Table	01/01/02	25000	30																										



5	Comfort zone	Double Bed	12/01/02	25000	25
6	Donald	Baby cot	24/02/02	6500	15
7	Royal Finish	Office Table	20/02/02	18000	30
8	Royal tiger	Sofa	22/02/02	31000	30
9	Econo sitting	Sofa	13/12/01	9500	25
10	Eating Paradise	Dining Table	19/02/02	11500	25

**TABLE : ARRIVALS**

NO	ITEMNAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT
11	Wood Comfort	Double Bed	23/03/03	25000	25
12	Old Fox	Sofa	20/02/03	17000	20
13	Micky	Baby cot	21/02/03	7500	15

(a) **To show all information about the Baby cots from the FURNITURE table.**

Ans. `SELECT * FROM FURNITURE WHERE TYPE='Baby cot' ;`

(b) **To list the ITEMNAME which are priced at more than 15000 from the FURNITURE table.**

Ans. `SELECT ITEMNAME FROM FURNITURE WHERE PRICE>15000 ;`

(c) **To list ITEMNAME and TYPE of those items, in which DATEOFSTOCK is before 22/01/02 from the FURNITURE table in descending order of ITEMNAME.**

Ans. `SELECT ITEMNAME, TYPE FROM FURNITURE WHERE DATEOFSTOCK<{22/01/02} ORDER BY ITEMNAME DESC ;`

(d) **To display ITEMNAME and DATEOFSTOCK of those items, in which the DISCOUNT percentage is more than 25 form FURNITURE table.**

Ans. `SELECT ITEMNAME,DATEOFSTOCK FROM FURNITURE WHERE DISCOUNT>25 ;`

(e) **To count the number of items, whose TYPE is "Sofa" from FURNITURE table.**

Ans. `SELECT COUNT(TYPE) FROM FURNITURE WHERE TYPE='SOFA' ;`

(f) **To insert a new row in the ARRIVALS table with the following data:**

Ans. **14, 'Velvet touch', 'Double bed', {25/03/03}, 25000,30**

`INSERT INTO ARRIVALS VALUES(14, 'Velvet touch', 'Double bed', {25/03/03}, 25000,30) ;`

(g) **Give the output of following SQL statement:**

*NOTE : Outputs of the below mentioned queries should be based on original data given in both the tables, i.e., without considering the insertion done in (f) part of this question :*

(i) **Select COUNT (distinct TYPE) from FURNITURE;**

Ans. `COUNT(distinct TYPE)`

5

(ii) **Select MAX(DISCOUT) form FURNITURE, ARRIVALS;**

Ans. `MAX(DISCOUNT)`

30,25

(iii) **Select AVG(DISCOUT) form FURNITURE where TYPE = 'Baby cot';**

Ans. `AVG(DISCOUT)`

15

(iv) **Select SUM(PRICE) from FURNITURE where DATEOFSTOCK<{12/02/02};**

Ans. `SUM(PRICE)`

66500

9. **Differentiate between SQL commands DROP TABLE and DROP VIEW.**

**Ans.** DROP TABLE:- DROP TABLE statement is used to delete the table and all its data from the database entirely. The syntax for DROP TABLE is DROP TABLE ;

DROP VIEW:- Removes an existing view from a database. DROP VIEW statement is used to remove a view or an object view from the database. The syntax for DROP VIEW is DROP VIEW ;

**10.** Study the following tables DOCTOR and SALARY and write SQL commands for the questions (i) to (iv) and give outputs for SQL queries (v) to (vi) :

**TABLE : DOCTOR**

ID	NAME	DEPT	SEX	EXPERIENCE
101	John	ENT	M	12
104	Smith	ORTHOPEdic	M	5
107	George	CARDIOLOGY	M	10
114	Lara	SKIN	F	3
109	K George	MEDICINE	F	9
105	Johnson	ORTHOPEdic	M	10
117	Lucy	ENT	F	3
111	Bill	MEDICINE	F	12
130	Morphy	ORTHOPEdic	M	15

**TABLE : SALARY**

ID	BASIC	ALLOWANCE	CONSULTATION
101	12000	1000	300
104	23000	2300	500
107	32000	4000	500
114	12000	5200	100
109	42000	1700	200
105	18900	1690	300
130	21700	2600	300

**(i)** Display NAME of all doctors who are in "MEDICINE" having more than 10 year experience from the table DOCTOR.

**Ans.** SELECT NAME FROM DOCTOR WHERE DEPT='MEDICINE' AND EXPERIENCE>10;

**(ii)** Display the average salary of all doctors working in "ENT" department using the tables DOCTOR and SALARY. Salary=BASIC + ALLOWANCE.

**Ans.** SELECT AVG(BASIC+ALLOWANCE) FROM DOCTOR A, SALARY B WHERE DEPT='ENT' AND A.ID=B.ID;

**(iii)** Display the minimum ALLOWANCE of female doctors.

**Ans.** SELECT MIN(ALLOWANCE) FROM DOCTOR A AND SALARY B WHERE SEX='F' AND A.ID=B.ID;

**(iv.)** Display the highest consultation fee among all male doctor.

**Ans.** SELECT MAX(CONST) FROM DOCTOR A, SALARY B WHERE SEX='M' AND A.ID=B.ID;

**(v)** SELECT count(\*) from DOCTOR where SEX="F".

**Ans.** count (\*)  
4

**(vi)** SELECT NAME, DEPT, BASIC from DOCTOR Salary WHERE DEPT="ENT" AND DOCTORID=SALARY.ID

**Ans.**

NAME	DEPT	BASIC
John	ENT	12000

**11. What are DDL and DML commands?**

**(a)** DDL is short form of Data Definition Language statements are used to build and modify the structure of database, tables and other objects in the database. When you execute a DDL statement, it takes effect immediately. Some of the commands comprising DDL are CREATE TABLE, DROP TABLE and CREATE INDEX.

**Ans.** DML is abbreviation of Data Manipulation Language. It is used to retrieve, store, modify, delete, insert and update data in database. Examples: SELECT, UPDATE, INSERT statements.

**(b)** Study the following tables **FLIGHTS** and **FARES** and write SQL commands for the questions (i) to (iv) and give outputs for SQL queries (v) to (vi).

**Table : FLIGHTS**

FL_NO	STARTING	ENDING	NO_FLIGHTS	NO_STOPS
IC301	MUMBAI	DELHI	8	0
IC799	BANGALORE	DELHI	2	1
MC101	INDORE	MUMBAI	3	0
IC302	DELHI	MUMBAI	8	0
AM812	KANPUR	BANGALORE	3	1
IC899	MUMBAI	KOCHI	1	4
AM501	DELHI	TRIVANDRUM	1	5
MU499	MUMBAI	MADRAS	3	3
IC701	DELHI	AHMEDABAD	4	0

**Table : FARES**

FL_NO	AIRLINES	FARE	TAX%
IC701	Indian Airlines	6500	10
MU499	Sahara	9400	5
AM501	Jet Airways	13450	8
IC899	India Airlines	8300	4
IC302	Indian Airlines	4300	10
IC799	Indian Airlines	10500	10
MC101	Deccan Airlines	3500	4

**(i)** Display FL\_NO and NO\_FLIGHTS from “KANPUR” to “BANGALORE” from the table FLIGHTS.

**Ans.** SELECT FL\_NO,NO\_FLIGHTS FROM FLIGHTS WHERE 'STARTING' LIKE 'KANPUR' AND ENDING LIKE 'BANGALORE' ;

**(ii)** Arrange the contents of the table FLIGHTS in the ascending order of FL\_NO.

**Ans.** SELECT \* FROM FLIGHTS ORDER BY FL\_NO

**(iii)** Display the FL\_NO and fare to be paid for the flights from DELHI to MUMBAI using the tables FLIGHTS and FARES, where the fare to be paid=FARE + FARE\*TAX%/100.

**Ans.** SELECT `FLIGHTS`.`FL\_NO`, (`FARE`+`FARE`\*(`TAX%`/100)) FROM `FLIGHTS`,`FARES` WHERE `STARTING` LIKE 'DELHI' AND `ENDING` LIKE 'MUMBAI' AND `FLIGHTS`.`FL\_NO` = `FARES`.`FL\_NO` ;

**(iv)** Display the minimum fare “Indian Airlines” is offering from the table FARES.

**Ans.** SELECT MIN(FARE) FROM FARES ;

**(v)** SELECT FL\_NO, NO\_FLIGHTS, AIRLINES from FLIGHTS, FARES WHERE STARTING = “DELHI” AND FLIGHTS.FL\_NO=FARES.FL\_NO.

**Ans.** ERROR - Column 'FL\_NO' in field list is ambiguous

**Correct Code and Ans.**

SELECT FLIGHTS.FL\_NO,NO\_FLIGHTS,FARES.AIRLINES FROM FLIGHTS,FARES  
WHERE FLIGHTS.STARTING LIKE 'DELHI' AND FLIGHTS.FL\_NO = FARES.FL\_NO;

FL_NO	NO_FLIGHTS	AIRLINES
IC302	8	Indian Airlines
AM501	1	Jet Airways
IC701	4	Indian Airlines

(vi) **SELECT count(distinct ENDING) from FLIGHTS.**

Ans. count(distinct ENDING)

7

12. Consider the following tables WORKERS and DESIG. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).

**Table : Workers**

W_ID	FIRSTNAME	LASTNAME	ADDRESS	CITY
102	Sam	Tones	33 Elm St.	Paris
105	Sarah	Ackerman	440 U. S. 110	New York
144	Manila	Sengupta	24 Friends Street	New Delhi
210	George	Smith	83 First Street	Howard
255	Mary	Jones	842 Vine Ave.	Losantiville
300	Robert	Samuel	9 Fifth Cross	Washington
335	Henry	Williams	12 Moore Street	Boston
403	Ronny	Lee	121 Harrison St.	New York
451	Pat	Thompson	11 Red Road	Paris

**Table : DESIG**

W_ID	SALARY	BENEFITS	DESIGNATION
102	75000	15000	Manager
105	85000	25000	Director
144	70000	15000	Manager
210	75000	12500	Manager
255	50000	12000	Clerk
300	45000	10000	Clerk
335	40000	10000	Clerk
400	32000	7500	Salesman
451	28000	7500	Salesman

(i) **To display W\_ID Firstname, Address and City of all employees living in New York from the table WORKERS.**

Ans. SELECT W\_ID, FIRSTNAME, ADDRESS, CITY FROM WORKERS WHERE CITY='NEW YORK' ;

(ii) **To display the content of WORKERS table in ascending order of LASTNAME.**

Ans. SELECT \* FROM WORKERS ORDER BY LASTNAME ;

(iii) **To display the Firstname, Lastname, and Total Salary of all clerk from the tables WORKERS and DESIG, where Total Salary is calculated as Salary + Benefits.**

Ans. SELECT FIRSTNAME, LASTNAME, SALARY+BENEFITS AS 'TOTAL SALARY' FROM WORKERS, DESIG WHERE WORKERS.W\_ID=DESIG.W\_ID ;

(iv) **To display the Minimum salary among Managers and Clerks from the table DESIG.**

Ans. SELECT MIN(SALARY) FROM DESIG WHERE DESIGNATION IN('MANAGER', 'CLERK') ;

(v) **SELECT FIRSTNAME, SALARY FROM WORKERS, DESIG WHERE DESIGNATION = 'Manager' AND WORKERS.W\_ID=DESIG.W\_ID;**

<b>Ans.</b>	<u>FIRSTNAME</u> <u>SALARY</u> Sam                75000 Manila            70000 George            75000
<b>(vi)</b>	<b>SELECT COUNT(DISTINCT DESIGNATION) FROM DESIG;</b>
<b>Ans.</b>	<u>COUNT(DISTINCT DESIGNATION)</u> 4
<b>(vii)</b>	<b>SELECT DESIGNATION, SUM(SALARY) FROM DESIG GROUP BY DESIGNATION HAVING COUNT(*)&lt;3;</b>
<b>Ans.</b>	<u>DESIGNATION</u> <u>SUM(SALARY)</u> Director                        85000 Salesman                        60000
<b>(viii)</b>	<b>SELECT SUM(BENEFITS) FROM DESIG WHERE DESIGNATION='Salesman';</b>
<b>Ans.</b>	<u>SUM(BENEFITS)</u> 15000

<b>13.</b>	<p><b>Consider the following tables GARMENT and FABRIC. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).</b></p> <p><b>Table: GARMENT</b></p> <table border="1"> <thead> <tr> <th>GCODE</th> <th>DESCRIPTION</th> <th>PRICE</th> <th>FCODE</th> <th>READYDATE</th> </tr> </thead> <tbody> <tr><td>10023</td><td>PENCIL SKIRT</td><td>1150</td><td>F03</td><td>19-DEC-08</td></tr> <tr><td>10001</td><td>FORMAL SHIRT</td><td>1250</td><td>F01</td><td>12-JAN-08</td></tr> <tr><td>10012</td><td>INFORMAL SHIRT</td><td>1550</td><td>F02</td><td>06-JAN-08</td></tr> <tr><td>10024</td><td>BABY TOP</td><td>750</td><td>F03</td><td>07-APR-07</td></tr> <tr><td>10090</td><td>TULIP SKIRT</td><td>850</td><td>F02</td><td>31-MAR-07</td></tr> <tr><td>10019</td><td>EVENING GOWN</td><td>850</td><td>F03</td><td>06-JUN-08</td></tr> <tr><td>10009</td><td>INFORMAL PANT</td><td>1500</td><td>F02</td><td>20-OCT-08</td></tr> <tr><td>10007</td><td>FORMAL PANT</td><td>1350</td><td>F01</td><td>09-MAR-08</td></tr> <tr><td>10020</td><td>FROCK</td><td>850</td><td>F04</td><td>09-SEP-07</td></tr> <tr><td>10089</td><td>SLACKS</td><td>750</td><td>F03</td><td>20-OCT-08</td></tr> </tbody> </table> <p><b>Table: FABRIC</b></p> <table border="1"> <thead> <tr> <th>FCODE</th> <th>TYPE</th> </tr> </thead> <tbody> <tr><td>F04</td><td>POLYSTER</td></tr> <tr><td>F02</td><td>COTTON</td></tr> <tr><td>F03</td><td>SILK</td></tr> <tr><td>F01</td><td>TERELENE</td></tr> </tbody> </table> <p><b>(i)</b>    <b>To display GCODE and DESCRIPTION of each GARMENT in descending order of GCODE</b>  <b>Ans.</b>    <code>SELECT GCODE, DESCRIPTION FROM GARMENT ORDER BY GCODE DESC;</code></p> <p><b>(ii)</b>    <b>To display the details of all the GARMENTs, which have READYDATE in between 08-DEC-07 and 16-JUN-08 (inclusive of both the dates).</b>  <b>Ans.</b>    <code>SELECT * FROM GARMENT WHERE READYDATE BETWEEN '08-DEC-07' AND '16-JUN-08';</code></p> <p><b>(iii)</b>    <b>To display the average PRICE of all the GARMENTs, which are made up of FABRIC with FCODE as F03.</b>  <b>Ans.</b>    <code>SELECT AVG(PRICE) FROM GARMENT WHERE FCODE = 'F03';</code></p> <p><b>(iv)</b>    <b>To display FABRICwise highest and lowest price of GARMENTs from GARMENT table. (Display FCODE of each GARMENT along with highest and lowest price).</b>  <b>Ans.</b>    <code>SELECT FCODE, MAX(PRICE), MIN(PRICE) FROM GARMENT GROUP BY FCODE;</code></p>	GCODE	DESCRIPTION	PRICE	FCODE	READYDATE	10023	PENCIL SKIRT	1150	F03	19-DEC-08	10001	FORMAL SHIRT	1250	F01	12-JAN-08	10012	INFORMAL SHIRT	1550	F02	06-JAN-08	10024	BABY TOP	750	F03	07-APR-07	10090	TULIP SKIRT	850	F02	31-MAR-07	10019	EVENING GOWN	850	F03	06-JUN-08	10009	INFORMAL PANT	1500	F02	20-OCT-08	10007	FORMAL PANT	1350	F01	09-MAR-08	10020	FROCK	850	F04	09-SEP-07	10089	SLACKS	750	F03	20-OCT-08	FCODE	TYPE	F04	POLYSTER	F02	COTTON	F03	SILK	F01	TERELENE
GCODE	DESCRIPTION	PRICE	FCODE	READYDATE																																																														
10023	PENCIL SKIRT	1150	F03	19-DEC-08																																																														
10001	FORMAL SHIRT	1250	F01	12-JAN-08																																																														
10012	INFORMAL SHIRT	1550	F02	06-JAN-08																																																														
10024	BABY TOP	750	F03	07-APR-07																																																														
10090	TULIP SKIRT	850	F02	31-MAR-07																																																														
10019	EVENING GOWN	850	F03	06-JUN-08																																																														
10009	INFORMAL PANT	1500	F02	20-OCT-08																																																														
10007	FORMAL PANT	1350	F01	09-MAR-08																																																														
10020	FROCK	850	F04	09-SEP-07																																																														
10089	SLACKS	750	F03	20-OCT-08																																																														
FCODE	TYPE																																																																	
F04	POLYSTER																																																																	
F02	COTTON																																																																	
F03	SILK																																																																	
F01	TERELENE																																																																	

(v) Ans.	<b>SELECT SUM(PRICE) FROM GARMENT WHERE FCODE='F01';</b> <u>SUM(PRICE)</u> 2600
(vi) Ans.	<b>SELECT DESCRIPTION, TYPE FROM GARMENT, FABRIC WHERE GARMENT.FCODE =FABRIC.FCODE AND GARMENT.PRICE &gt; = 1260;</b> <u>DESCRIPTION TYPE</u> INFORMAL SHIRT COTTON INFORMAL PANT COTTON FORMAL PANT TERELENE
(vii) Ans.	<b>SELECT MAX(FCODE) FROM FABRIC;</b> <u>MAX(FCODE)</u> F04
(viii) Ans.	<b>SELECT COUNT (DISTINCT PRICE) FROM GARMENT;</b> <u>COUNT(DISTINCT PRICE)</u> 7

14.	<p>Consider the following tables DRESS and MATERIAL. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).</p> <p><b>Table : DRESS</b></p> <table border="1"> <thead> <tr> <th>DCODE</th> <th>DESCRIPTION</th> <th>PRICE</th> <th>MCODE</th> <th>LAUNCHDATE</th> </tr> </thead> <tbody> <tr><td>10001</td><td>FORMAL SHIRT</td><td>1250</td><td>M001</td><td>12-JAN-08</td></tr> <tr><td>10020</td><td>FROCK</td><td>750</td><td>M004</td><td>09-SEP-07</td></tr> <tr><td>10012</td><td>ONFORMAL SHIRT</td><td>1450</td><td>M002</td><td>06-JUN-08</td></tr> <tr><td>10019</td><td>EVENING GOWN</td><td>850</td><td>M003</td><td>06-JUN-08</td></tr> <tr><td>10090</td><td>TULIP SKIRT</td><td>850</td><td>M002</td><td>31-MAR-07</td></tr> <tr><td>10023</td><td>PENCIL SKIRT</td><td>1250</td><td>M003</td><td>19-DEC-08</td></tr> <tr><td>10089</td><td>SLACKS</td><td>850</td><td>M003</td><td>20-OCT-08</td></tr> <tr><td>10007</td><td>FORMAL PANT</td><td>1450</td><td>M001</td><td>09-MAR-08</td></tr> <tr><td>10009</td><td>INFORMAL PANT</td><td>1400</td><td>M002</td><td>20-OCT-08</td></tr> <tr><td>10024</td><td>BABY TOP</td><td>650</td><td>M003</td><td>07-APR-07</td></tr> </tbody> </table> <p><b>Table : MATERIAL</b></p> <table border="1"> <thead> <tr> <th>MCODE</th> <th>TYPE</th> </tr> </thead> <tbody> <tr><td>M001</td><td>TERELENE</td></tr> <tr><td>M002</td><td>COTTON</td></tr> <tr><td>M004</td><td>POLYESTER</td></tr> <tr><td>M003</td><td>SILK</td></tr> </tbody> </table>	DCODE	DESCRIPTION	PRICE	MCODE	LAUNCHDATE	10001	FORMAL SHIRT	1250	M001	12-JAN-08	10020	FROCK	750	M004	09-SEP-07	10012	ONFORMAL SHIRT	1450	M002	06-JUN-08	10019	EVENING GOWN	850	M003	06-JUN-08	10090	TULIP SKIRT	850	M002	31-MAR-07	10023	PENCIL SKIRT	1250	M003	19-DEC-08	10089	SLACKS	850	M003	20-OCT-08	10007	FORMAL PANT	1450	M001	09-MAR-08	10009	INFORMAL PANT	1400	M002	20-OCT-08	10024	BABY TOP	650	M003	07-APR-07	MCODE	TYPE	M001	TERELENE	M002	COTTON	M004	POLYESTER	M003	SILK
DCODE	DESCRIPTION	PRICE	MCODE	LAUNCHDATE																																																														
10001	FORMAL SHIRT	1250	M001	12-JAN-08																																																														
10020	FROCK	750	M004	09-SEP-07																																																														
10012	ONFORMAL SHIRT	1450	M002	06-JUN-08																																																														
10019	EVENING GOWN	850	M003	06-JUN-08																																																														
10090	TULIP SKIRT	850	M002	31-MAR-07																																																														
10023	PENCIL SKIRT	1250	M003	19-DEC-08																																																														
10089	SLACKS	850	M003	20-OCT-08																																																														
10007	FORMAL PANT	1450	M001	09-MAR-08																																																														
10009	INFORMAL PANT	1400	M002	20-OCT-08																																																														
10024	BABY TOP	650	M003	07-APR-07																																																														
MCODE	TYPE																																																																	
M001	TERELENE																																																																	
M002	COTTON																																																																	
M004	POLYESTER																																																																	
M003	SILK																																																																	
(i) Ans.	<b>To display DCODE and DISCRIPTION of each dress in ascending order of DCODE.</b> SELECT DCODE,DESCRIPTION FROM DRESS ORDER BY DCODE;																																																																	
(ii) Ans.	<b>To display the details of all the dresses which have LAUNCHDATE in between 05-DEC-07 AND 20-JUN-08 (inclusive of both the dates).</b> SELECT * FROM DRESS WHERE LAUNCHDATE BETWEEN '05-DEC-07' AND '20-JUN-08';																																																																	
(iii) Ans.	<b>To display the average PRICE of all the dresses which are made up of material with MCODE as M003.</b> SELECT AVG(PRICE) FROM DRESS WHERE MCODE='M003';																																																																	
(iv) Ans.	<b>To display materialwie highest and lowest price of dresses from DRESS table. (Display MCODE of each dress along with highest and lowest price)</b> SELECT B.MCODE,TYPE,MAX(PRICE) AS "HIGHEST",MIN(PRICE) AS "LOWEST" FROM DRESS A, MATERIAL B WHERE A.MCODE=B.MCODE GROUP BY TYPE;																																																																	

(v) **SELECT SUM(PRICE) FROM DRESS WHERE MCODE = 'M001';**

Ans. SUM(PRICE)  
2700

(vi) **SELECT DESCRIPTION, TYPE FROM DRESS, MATERIAL WHERE DRESS.MCODE=MATERIAL.MCODE AND DRESS.PRICE >= 1250;**

Ans. 

DESCRIPTION	TYPE
FORMAL SHIRT	TERELENE
INFORMAL SHIRT	COTTON
PENCIL SKIRT	SILK
FORMAL PANT	TERELENE
INFORMAL PANT	COTTON

(vii) **SELECT MAX(MCODE) FROM MATERIAL;**

Ans. MAX(MCODE)  
M004

(viii) **SELECT COUNT(DISTINCT PRICE) FROM DRESS;**

Ans. COUNT(DISTINCT PRICE)  
6

15. Consider the following tables Stationery and Consumer. Write SQL commands for the statement (i) to (iv) and give output for SQL queries (v) to (viii).

**Table : STATIONERY**

S_ID	StationeryName	Company	Price
DP01	Dot Pen	ABC	10
PL02	Pencil	XYZ	6
ER05	Eraser	XYZ	7
PL01	Pencil	CAM	5
GP02	Gel Pen	ABC	15

**Table : CONSUMER**

C_ID	ConsumerName	Address	S_ID
01	Good Lerner	Delhi	PL01
06	Write Well	Mumbai	GP02
12	Topper	Delhi	DP01
15	Write & Draw	Delhi	PL02
16	Motivation	Banglore	PL01

(i) **To display the details of those Consumers whose Address is Delhi.**

Ans. `SELECT * FROM CONSUMER WHERE ADDRESS="DELHI";`

(ii) **To display the details of Stationery whose Price is in the range of 8 to 15 (Both value included)**

Ans. `SELECT * FROM STATIONERY WHERE PRICE BETWEEN 8 AND 15;`

(iii) **To display the ConsumerName, Address from Tble Consumer, and Company and Price from table Stationery, with their corresponding matching S\_ID**

Ans. `SELECT CONSUMERNAME, ADDRESS, COMPANY, PRICE FROM CONSUMER, STATIONERY WHERE CONSUMER.S_ID=STATIONERY.S_ID;`

(iv) **To increase the Price of all stationery by 2**

Ans. `UPDATE STATIONERY SET PRICE=PRICE+2;`

(v) **SELECT DISTINCT Address FROM Consumer;**

**Ans.** Address  
Delhi  
Mumbai  
Banglore

**(vi)** **SELECT Company, MAX(Price),Min(Price),Count(\*) FROM Stationery GROUP BY Company;**

**Ans.**

<u>Company</u>	<u>MAX(Price)</u>	<u>Min(Price)</u>	<u>Count(*)</u>
ABC	17	12	2
CAM	7	7	1
XYZ	9	8	2

**(vii)** **SELECT Consumer.CnsumerName, stationery.stationeryName, Stationery.Price FROM Stationery, Consumer WHERE Consumer.S\_Id=Stationery.S\_Id**

**Ans.**

<u>ConsumerName</u>	<u>StationeryName</u>	<u>Price</u>
Good Lerner	Pencil	7
Write Well	Gel Pen	17
Topper	Dot Pen	12
Write & Drow	Pencil	8
Motivation	Pencil	7

**(viii)** **SELECT StationeryName, Price \* 3 FROM Stationery**

**Ans.**

<u>StationeryName</u>	<u>Price*3</u>
Dot Pen	36
Pencil	24
Eraser	27
Pencil	21
Gel Pen	51