



# INDIAN SCHOOL MUSCAT



## CLASS XII

# INFORMATION TECHNOLOGY(802)

## UNIT-1

# DATABASE CONCEPTS

**POWER CAPSULE**



# RDBMS



## **BASIC DATABASE CONCEPTS**

### **Definitions:**

**Data :- Raw facts and figures which are useful to an organization. We cannot take decisions on the basis of data.**

**Information:-Well processed data is called information. We can take decisions on the basis of information**



# RDBMS



**Field: Set of characters that represents specific data element.**

**Record: Collection of fields is called a record. A record can have fields of different data types.**



# RDBMS



**Database:** Collection of logically related data along with its description is termed as database. In relational model the data's are stored as tables. A table is also known as a relation.

## **Why we need a database?**

**Database** helps in keeping the files in a systematic manner. It helps in managing large amount of information in small time. A **database** is an organized collection of data. A relational **database**, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements.





# RDBMS



## Definitions:

**Tuple:** A row in a relation or table is called a tuple.

**Attribute:** A column in a relation or table is called an attribute. It is also termed as field or data item.



# RDBMS



**Degree:** Number of attributes in a relation or number of columns in a table is called degree of a relation or table.

**Cardinality:** Number of tuples in a relation or number of rows in a table is called cardinality of a relation or table.



# RDBMS



**Primary Key:** Primary key is a key that can uniquely identifies the records/tuples in a relation or a table. This key can never be duplicated and NULL.

**Foreign Key:** Foreign Key is a key that is defined as a primary key in some other relation or table. This key is used to enforce referential integrity in RDBMS.



# RDBMS



**Candidate Key:** Set of all attributes which can serve as a primary key in a relation or a table.

**Alternate Key:** All the candidate keys other than the primary keys of a relation are alternate keys for a relation or a table.





# RDBMS



## **Advantages of Database Management System:**

**Reducing Data Redundancy:** The file based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems.

**Sharing of Data.**

**Data Integrity.**

**Data Security.**

**Privacy.**

**Backup and Recovery.**

**Data Consistency.**



# RDBMS



**DBA:** Data Base Administrator is a person (manager) that is responsible for defining the data base schema, setting security features in database, ensuring proper functioning of the data bases etc.



# STRUCTURED QUERY LANGUAGE( MySQL )

## SQL Commands



CREATE A TABLE EMPLOYEE WITH THE FOLLOWING DETAILS

Name	Data type	size	constraints
-----	-----	-----	-----
Ecode	integer		Primary Key
Ename	character	25	Not Null
Sex	character	1	Not Null
Dept	character	25	Not null
Desig	character	25	
DOJ	date		Not null
Salary	decimal(8,2)		Salary > 500



# STRUCTURED QUERY LANGUAGE( MySQL )

## SQL Commands



### CREATING TABLE EMPLOYEE IN MySQL:

```
CREATE TABLE EMPLOYEE (  
Ecode      integer          not null primary key,  
Ename      varchar(25)      not null,  
Sex        char(1)         not null,  
Dept       varchar(25)      not null,  
Desig      varchar(25),  
DOJ        date,  
Salary     decimal(8,2) check(Salary > 500) );
```





# STRUCTURED QUERY LANGUAGE( MySQL )

## SQL Commands



INSERT THE FOLLOWING DATA TO TABLE EMPLOYEE

Ecode	ENAME	Sex	Dept	Desig	DOJ	Salary
1101	Brian	M	Accounts	Manager	2015-02-03	2000
1102	Mrinal	M	Logistics	Clerk	2012-03-02	650
1103	Sunita	F	Production	Asst. Manager	2010-04-01	1900
1104	Vansh	M	Accounts	Clerk	2009-05-07	700
1105	Samira	F	Logistics	Manager	2011-07-11	2200



# **STRUCTURED QUERY LANGUAGE( MySQL )**

## **SQL Commands**



### **INSERTING ROWS TO TABLE EMPLOYEE**

**INSERT INTO EMPLOYEE VALUES(1101, 'Brian', 'M', 'Accounts',  
'Manager', '2015-02-03', 2000),  
(1102, 'Mrinal', 'M', 'Logistics', 'Clerk', '2012-03-02', 650),  
(1103, 'Sunita', 'F', 'Production', 'Asst. Manager', '2010-04-01', 1900),  
(1104, 'Vansh', 'M', 'Accounts', 'Clerk', '2009-05-07', 700),  
(1105, 'Samira', 'F', 'Logistics', 'Manager', '2011-07-11', 2200) ;**



# **STRUCTURED QUERY LANGUAGE( MySQL )**

## **SQL Commands**



**Write MySQL commands to the following questions**

- 1) To describe the structure of the table employee**
- 2) To display all records from employee table**
- 3) To display name and designation of all employees**
- 4) To display the records in the descending order of their salaries**
- 5) To display name, department and DOJ of all employees whose salary is between 1500 and 2000**
- 6) To display the details of all managers**
- 7) To display the number of employees department wise**
- 8) To display all records in the ascending order of their DOJ**
- 9) To display the name of employees starting with the letter 'S'**
- 10) To display the name of employees ending with letter 'a'**
- 11) To display the name of employees whose name has exactly 5 letters**
- 12) To display the details of male employees**





# STRUCTURED QUERY LANGUAGE( MySQL )

## SQL Commands



### ANSWERS

- 1) **mysql> DESC Employee ;**
- 2) **mysql> SELECT \* FROM EMPLOYEE ;**
- 3) **mysql> SELECT Ename, Desig FROM EMPLOYEE ;**
- 4) **mysql> SELECT \* FROM EMPLOYEE ORDER BY SALARY DESC ;**
- 5) **mysql> SELECT Ename, Dept, DOJ FROM EMPLOYEE WHERE SALARY BETWEEN 1500 AND 2000 ;**
- 6) **mysql> SELECT \* FROM EMPLOYEE WHERE Desig ='Manager' ;**
- 7) **mysql> SELECT Dept ,count(Dept) FROM EMPLOYEE GROUP BY Dept ;**
- 8) **mysql> SELECT \* FROM EMPLOYEE ORDER BY DOJ ASC ;**
- 9) **mysql> SELECT Ename FROM EMPLOYEE WHERE Ename like 'S%';**
- 10) **mysql> SELECT Ename FROM EMPLOYEE WHERE Ename like '%a';**
- 11) **mysql> SELECT Ename FROM EMPLOYEE WHERE Ename like '\_\_\_\_\_' ;**
- 12) **mysql> SELECT \* FROM EMPLOYEE WHERE SEX ='M' ;**





# STRUCTURED QUERY LANGUAGE( MySQL )

TABLE: INTERIORS

No.	ITEMNAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT
1	Red rose	Double Bed	23/02/02	32000	15
2	Soft touch	Baby cot	20/01/02	9000	10
3	Jerry's home	Baby cot	19/02/02	8500	10
4	Rough wood	Office Table	01/01/02	20000	20
5	Comfort zone	Double Bed	12/01/02	15000	20
6	Jerry look	Baby cot	24/02/02	7000	19
7	Lion king	Office Table	20/02/02	16000	20
8	Royal tiger	Sofa	22/02/02	30000	25
9	Park sitting	Sofa	13/12/01	9000	15
10	Dine Paradise	Dining Table	19/02/02	11000	15
11	White Wood	Double Bed	23/03/03	20000	20
12	James 007	Sofa	20/02/03	15000	15
13	Tom look	Baby cot	21/02/03	7000	10



# STRUCTURED QUERY LANGUAGE( MySQL )



**Write the SQL commands for (a) to (f) and write the outputs for SQL queries (g) parts (i) to (iii) on the basis of the table INTERIORS.**

- (a) To show all information about the Sofa from the INTERIORS table.
- (b) To list the ITEM NAME, which are priced at more than 10000 from the INTERIORS table.
- (c) To list ITEM NAME and TYPE of those items, in which DATEOFSTOCK is before 22/01/02 from the INTERIORS table in descending order of ITEM NAME.
- (d) To display ITEM NAME and DATEOFSTOCK of those items, in which the discount percentage is more than 15 from INTERIORS table.

# STRUCTURED QUERY LANGUAGE( MySQL )



(e) To count the number of items, whose type is Double Bed from INTERIORS table.

(f) To insert a new row in the INTERIORS table with the following data

{114,'True Indian','Office Table',' 28/03/03',15000,20}

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

(i) `SELECT COUNT ( DISTINCT TYPE) FROM INTERIORS; .`

(ii) `SELECT AVG (DISCOUNT) FROM INTERIORS WHERE TYPE = 'Baby cot' ;`

(iii) `SELECT SUM( PRICE) FROM INTERIORS WHERE DATE0FST0CK<'12/02/02' ;`

# STRUCTURED QUERY LANGUAGE( MySQL )



- (a) **SELECT \* FROM INTERIORS WHERE TYPE = 'Sofa';**
- (b) **SELECT ITEMNAME FROM INTERIORS WHERE PRICE > 10000;**
- (c) **SELECT ITEMNAME, TYPE FROM INTERIORS WHERE DATEOFSTOCK < '22/01/02'**  
**ORDER BY ITEMNAME DESC;**
- (d) **SELECT ITEMNAME , DATEOFSTOCK FROM INTERIORS WHERE DISCOUNT > 15 ;**
- (e) **SELECT COUNT(\*) FROM INTERIORS WHERE TYPE = 'Double Bed';**
- (f) **INSERT INTO INTERIORS VALUES (114, ' True Indian', 'Office Table', '28/03/03',15000,20);**



# STRUCTURED QUERY LANGUAGE( MySQL )



g) The outputs is given after excluding the row given in part (f).

(i)	COUNT (DISTINCT TYPE)
-----	-----------------------

5
---

(ii)	AVG(DISCOUNT)
------	---------------

12.25
-------

(iii)	SUM(PRICE)
-------	------------

53000
-------

# STRUCTURED QUERY LANGUAGE( MySQL )



TABLE: HOSPITAL

No.	Name	Age	Department	Dateofadm	Charges	Sex
1	Sandeep	65	Surgery	23/02/98	300	M
2	Ravina	24	Orthopaedic	20/01/98	200	F
3	Karan	45	Orthopaedic	19/02/98	200	M
4	Tarun	12	Surgery	01/01/98	300	M
5	Zubin	36	ENT	12/01/98	250	M
6	Ketaki	16	ENT	24/02/98	300	F
7	Ankita	29	Cardiology	20/02/98	800	F
8	Zareen	45	Gynaecology	22/02/98	300	F
9	Kush	19	Cardiology	13/01/98	800	M
10	Shailya	31	Nuclear Medicine	19/02/98	400	M



# STRUCTURED QUERY LANGUAGE( MySQL )



**Write the SQL commands for (a) to (f) and write the outputs for SQL queries (g) parts (i) to (iv) on the basis of the table HOSPITAL.**

- (a) To show all information about the patients of Cardiology Department.
- (b) To list the name of female patients, who are in Orthopaedic Department.
- (c) To list names of all patients with their date of admission in ascending order.
- (d) To display Patient's Name, Charges, Age for male patients only.
- (e) To count the number of patients with age>20.
- (f) To insert a new row in the HOSPITAL table with the following data:  
{11, 'Mustafa', 37, 'ENT', '25/02/98', 250, 'M' }

# STRUCTURED QUERY LANGUAGE( MySQL )



(g) Give the output of following SQL statements:

(i) `SELECT COUNT(DISTINCT Charges) FROM HOSPITAL;`

(ii) `SELECT MIN(Age) FROM HOSPITAL WHERE Sex = 'M' ;`

(iii) `SELECT SUM(Charges) FROM HOSPITAL WHERE Sex = ' F' ;`

(iv) `SELECT AVG(Charges) FROM HOSPITAL WHERE Dateofadm < '12/02/98' ;`



# STRUCTURED QUERY LANGUAGE( MySQL )



- (a) `SELECT * FROM HOSPITAL WHERE Department = 'Cardiology';`
- (b) `SELECT Name FROM HOSPITAL WHERE Department = 'Orthopaedic' AND Sex = ' F' ;`
- (c) `SELECT Name FROM HOSPITAL ORDER BY Dateofadm;`
- (d) `SELECT Name, Charges, Age FROM HOSPITAL WHERE Sex = 'M' ;`
- (e) `SELECT COUNT(*) FROM HOSPITAL WHERE Age > 20;`
- (f) `INSERT INTO HOSPITAL VALUES (11, 'Mustafa', 37, 'ENT', '25/02/98', 250, 'M');`

# STRUCTURED QUERY LANGUAGE( MySQL )



(g) The output is given after excluding the row given in part (f).

(i) COUNT (DISTINCT Charges)

5

(ii) MIN(Age)

12

(iii) SUM(Charges)

1600

(iv) AVG(Charges)

387.50



# STRUCTURED QUERY LANGUAGE( MySQL )



TABLE: FAMILY

No.	Name	FemaleMembers	MaleMembers	Income	Occupation
1	Mishra	3	2	7000	Service
2	Gupta	4	1	50000	Business
3	Khan	6	3	8000	Mixed
4	Chaddha	2	2	25000	Business
5	Yadav	7	2	20000	Mixed
6	Joshi	3	2	14000	Service
7	Maurya	6	3	5000	Farming
8	Rao	5	2	10000	Service

# STRUCTURED QUERY LANGUAGE( MySQL )



Write SQL commands for the following statements:

- (a) To select all the information of family, whose Occupation is Service.
- (b) To list the name of family, where female members are more than 3.
- (c) To list all names of family with income in ascending order.
- (d) To display family's name, male members and occupation of business family.
- (e) To count the number of family, whose income is less than 10,000.
- (f) To insert a new record in the FAMILY table with the following data  

```
9, "D'souza", 2, 1, 15000, "Service"
```



# STRUCTURED QUERY LANGUAGE( MySQL )



Give the output of the following SQL commands .

```
(i) SELECT MIN(DISTINCT Income) FROM FAMILY;  
(ii) SELECT MIN(FemaleMembers) FROM FAMILY WHERE Occupation = 'Mixed';  
(iii) SELECT SUM(Income) FROM FAMILY WHERE Occupation = 'Service';  
(iv) SELECT AVG(Income) FROM FAMILY;
```

# STRUCTURED QUERY LANGUAGE( MySQL )



- (a) `SELECT * FROM FAMILY WHERE Occupation = 'Service';`
- (b) `SELECT Name FROM FAMILY WHERE FemaleMembers > 3;`
- (c) `SELECT Name, Income FROM FAMILY ORDER BY Income;`
- (d) `SELECT Name, MaleMembers , Occupation  
FROM FAMILY WHERE Occupation = ' Business ' ;`
- (e) `SELECT COUNT(*) FROM FAMILY WHERE Income < 10000;`
- (f) `INSERT INTO FAMILY VALUES (9, 'D'souza' ,2,1,15000, 'Service') ;`

# STRUCTURED QUERY LANGUAGE( MySQL )



## OUTPUT QUESTION ANSWERS:

(i)	<b>MIN(DISTINCT Income)</b>
-----	-----------------------------

5000
------

(ii)	<b>MIN(FemaleMembers)</b>
------	---------------------------

6
---

(iii)	<b>SUM(Income)</b>
-------	--------------------

31000
-------

(iv)	<b>AVG(Income)</b>
------	--------------------

17375
-------

# STRUCTURED QUERY LANGUAGE( MySQL )



TABLE: SPORTS

STUDENTNO	CLASS	NAME	GAME1	GRADE	GAME2	GRADE1
10	7	Sameer	Cricket	B	Swimming	A
11	8	Sujit	Tennis	A	Skating	C
12	7	Kamal	Swimming	B	Football	B
13	7	Veena	Tennis	C	Tennis	A
14	9	Archana	Basketball	A	Cricket	A
15	10	Arpit	Cricket	A	Athletics	C



# STRUCTURED QUERY LANGUAGE( MySQL )



## Write MySQL commands for the following questions

- (a) Display the names of the students, who have grade 'C' in either GAME1 or GAME2 or both.
- (b) Display the number of students getting grade A in Cricket.
- (c) Display the names of the students who have same game for both GAME1 and GAME2.
- (d) Display the games taken up by the students, whose name starts with A.
- (e) Add a new column named MARKS.
- (f) Assign a value 200 for Marks for all those, who are getting grade 'A' or grade 'B' in both GAME1 and GAME2.
- (g) Arrange the whole table in the alphabetical order of Name.

# STRUCTURED QUERY LANGUAGE( MySQL )



Write MySQL commands for the following questions

- (a) `SELECT NAME FROM SPORTS WHERE GRADE = 'C' OR GRADE1 = 'C' ;`
- (b) `SELECT COUNT(*) FROM SPORTS WHERE (GAME1 = 'Cricket' AND GRADE = 'A' ) OR (GAME2 = 'Cricket' AND GRADE1 = 'A') ;`
- (c) `SELECT NAME FROM SPORTS WHERE GAME1 = GAME2 ;`
- (d) `SELECT GAME1, GAME2 FROM SPORTS WHERE NAME LIKE ' A% ' ;`
- (e) `ALTER TABLE SPORTS ADD MARKS integer ;`
- (f) `UPDATE SPORTS SET MARKS = 200 WHERE GRADE = 'A' OR GRADE = 'B' OR GRADE1 = 'A' OR GRADE1 = 'B' ;`
- (g) `SELECT * FROM SPORTS ORDER BY NAME ;`

# STRUCTURED QUERY LANGUAGE( MySQL )



TABLE: COLLEGE

No	Name	Age	Department	DateofJoin	Basic	Sex
1	Shalaz	45	Biology	13/02/88	10500	M
2	Sameera	54	Biology	10/01/90	9500	F
3	Yagyen	43	Physics	27/02/98	8500	M
4	Pratyush	34	Chemistry	11/01/93	7500	M
5	Aren	51	Mathematics	22/01/91	8500	M
6	Reeta	27	Chemistry	14/02/94	9000	F
7	Urvashi	29	Biology	10/02/93	8500	F
8	Teena	35	Mathematics	02/02/89	10500	F
9	Viren	49	Mathematics	03/01/88	9000	M
10	Prakash	22	Physics	17/02/92	8000	M

# STRUCTURED QUERY LANGUAGE( MySQL )



## Write MySQL commands for the following questions

- (a) To change the Basic salary to 10500 of all those teachers from COLLEGE, who joined the COLLEGE after 01/02/89 and are above the age of 50.
- (b) To display -Name, Age and Basic of all those from COLLEGE, who belong to Physics and Chemistry department only.
- (c) To display all the department names from COLLEGE, with no duplication.
- (d) To list names of all teachers from COLLEGE with their date of joining in ascending order within their Basic salaries in ascending order.
- (e) To display maximum salary amongst the female teachers and also amongst the male teachers from COLLEGE. (Give a single command)
- (f) To insert a new row in the table COLLEGE with the following data:  
15, "ATIN", 27, "Physics", '15/05/02', 8500, "M"
- (g) To delete a row from COLLEGE in which name is VIREN.



# STRUCTURED QUERY LANGUAGE( MySQL )



## Write MySQL commands for the following questions

- (a) UPDATE COLLEGE SET Basic = 10500 WHERE  
DateofJoin>'01/02/89 ' AND Age>50;
- (b) SELECT Name, Age, Basic FROM COLLEGE WHERE  
Department='Physics' OR Department ='Chemistry' ;
- (c) SELECT DISTINCT Department FROM COLLEGE;
- (d) SELECT Name,DateofJoin,Basic FROM COLLEGE ORDER BY  
Basic,DateofJoin;
- (e) SELECT Sex,MAX(Basic) FROM COLLEGE GROUP BY Sex;
- (f) INSERT INTO COLLEGE VALUES (15, 'ATIN', 27 'Physics',  
'15/05/02', 8500, 'M');
- (g) DELETE FROM COLLEGE WHERE Name = 'VIREN';

# STRUCTURED QUERY LANGUAGE( MySQL )



TABLE: FLIGHTS

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

# STRUCTURED QUERY LANGUAGE( MySQL )



TABLE: FARES

FL_NO	AIRLINES	FARE	TAX%
IC701	INDIAN AIRLINES	6500	10
MU499	SAHARA	9400	5
AM501	JET AIRWAYS	13450	8
IC899	INDIAN AIRLINES	8300	4
IC302	INDIAN AIRLINES	4300	10
IC799	INDIAN AIRLINES	10500	10
MC101	DECCAN AIRLINES	3500	4

# STRUCTURED QUERY LANGUAGE( MySQL )



## Write MySQL commands for the following questions

- (a) Display FL\_NO and NO\_FLIGHT from KANPUR to BENGALURU from the table FLIGHTS.
- (b) Arrange the contents of the table FLIGHTS in the ascending order of FL\_NO.
- (c) 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 =  $\text{FARE} + \text{FARE} * \text{TAX} / 100$ .
- (d) Display the minimum fare INDIAN AIRLINES is offering from the table FARES.



# STRUCTURED QUERY LANGUAGE( MySQL )



(e) `SELECT FL_NO, NO_FLIGHT, AIRLINES FROM FLIGHTS,  
FARES WHERE STARTING = 'DELHI' AND  
FLIGHTS.FL_NO = FARES.FL_NO;`

(f) `SELECT COUNT (DISTINCT ENDING) FROM FLIGHTS;`

# STRUCTURED QUERY LANGUAGE( MySQL )



- (a) `SELECT FL_NO, NO_FLIGHT FROM FLIGHTS WHERE  
STARTING = 'KANPUR' AND ENDING = 'BENGALURU';`
- (b) `SELECT * FROM FLIGHTS ORDER BY FL_NO;`
- (c) `SELECT FL_NO, FARE + FARE * TAX/100 FROM FARES  
WHERE FL_NO = (SELECT FL_NO FROM FLIGHTS WHERE  
STARTING = 'DELHI' AND ENDING = 'MUMBAI');`
- (d) `SELECT MIN(FARE) FROM FARES GROUP BY AIRLINES  
HAVING AIRLINES = 'INDIAN AIRLINES';`

# STRUCTURED QUERY LANGUAGE( MySQL )



(e)

FL_No	NO_FLIGHT	AIRLINES
IC302	8	INDIAN AIRLINES
AM501	1	JET AIRWAYS
IC701	4	INDIAN AIRLINES

(f)

COUNT (DISTINCT ENDING)
7

# STRUCTURED QUERY LANGUAGE( MySQL )



TABLE: SHOPPE

Id	SName	Area
S001	ABC Computeronics	CP
S002	All Infotech Media	GK II
S003	Tech Shoppe	CP
S004	Geeks Tecno Soft	Nehru Place
S005	Hitech Tech Store	Nehru Place



# STRUCTURED QUERY LANGUAGE( MySQL )



**TABLE: ACCESSORIES**

No	Name	Price	Id
A01	Mother Board	12000	S01
A02	Hard Disk	5000	S01
A03	Keyboard	500	S02
A04	Mouse	300	S01
A05	Mother Board	13000	S02
A06	Keyboard	400	S03
A07	LCD	6000	S04
T08	LCD	5500	S05
T09	Mouse	350	S05
T10	Hard Disk	4500	S03

# STRUCTURED QUERY LANGUAGE( MySQL )



**Write the SQL queries:**

- (i) To display Name and Price of all the Accessories in ascending order of their Price.
- (ii) To display Id and SName of all Shoppe located in Nehru Place.
- (iii) To display Minimum and Maximum Price of each Name of Accessories.
- (iv) To display Name, Price of all Accessories and their respective SName, where they are available.

# STRUCTURED QUERY LANGUAGE( MySQL )



Write the output of the following SQL commands:

(i) `SELECT DISTINCT (NAME) FROM ACCESSORIES WHERE PRICE >= 5000;`

(ii) `SELECT AREA, COUNT (*) FROM SHOPPE GROUP BY AREA;`

(iii) `SELECT COUNT (DISTINCT AREA) FROM SHOPPE;`

# STRUCTURED QUERY LANGUAGE( MySQL )



## SQL queries answers:

- (i) `SELECT Name, Price FROM ACCESSORIES ORDER BY Price;`
- (ii) `SELECT Id, SName FROM SHOPPE WHERE Area ='Nehru Place';`
- (iii) `SELECT MIN(Price) as "Minimum Price", MAX(Price) as "Maximum Price", Name FROM ACCESSORIES GROUP BY Name;`
- (iv) `SELECT Name, Price, SName FROM ACCESSORIES A, SHOPPE S  
WHERE A.Id = S.Id;`



# STRUCTURED QUERY LANGUAGE( MySQL )



## SQL queries answers:

(b) (i)

NAME
Mother Board
Hard Disk
LCD

(ii)

AREA	COUNT(*)
GK II	1
Nehru Place	2
CP	2

(iii)

COUNT (DISTINCTAREA)
3