Unit III: Database Management Class XII Compuer Sc(083) Total Marks 20

● **Database concepts**: introduction to database concepts and its need

● Relational data model: relation, attribute, tuple, domain, degree, cardinality, keys (candidate key, primary key, alternate key, foreign key)

● Structured Query Language: introduction, Data Definition Language and Data Manipulation Language, data type (char(n), varchar(n), int, float, date), constraints (not null, unique, primary key), create database, use database, show databases, drop database, show tables, create table, describe table, alter table (add and remove an attribute, add and remove primary key), drop table, insert, delete, select, operators (mathematical, relational and logical), aliasing, distinct clause, where clause, in, between, order by, meaning of null, is null, is not null, like, update command, delete command, aggregate functions (max, min, avg, sum, count), group by, having clause, joins: cartesian product on two tables, equi-join and natural join

● Interface of python with an SQL database: connecting SQL with Python, performing insert, update, delete queries using cursor, display data by using connect(), cursor(), execute(), commit(), fetchone(), fetchall(), rowcount, creating database connectivity applications, use of %s format specifier or format() to perform queries

---

Database Concepts

Introduction

A **Database** is an organized collection of data that can be easily accessed, managed, and updated.

A database is a collection of interconnected data that is used to efficiently insert, update, delete and retrieve data.

RDBMS stands for Relational Database Management System.

Example of RDBMS:

Postgre SQL, MariaDB, MySQL, Oracle, MS-Access, Libre Base etc.

SQL stands for Structured Query Language

PL/SQL stands for Procedural Language/ Structured Queery Language.

Application Areas of RDBMS

- Railway Reservation Systems
- Social Media Platforms
- Library Management Systems
- Banking Systems
- Education Management Systems
- Human Resource Management Systems
- Credit Card Transaction Systems
- Airline Reservation Systems
- E-commerce Systems
- Hospital Management Systems

A school database stores data about students, teachers, and subjects.

Need for Database

1. **Data Organization** – Keeps data in a structured form.
2. **Avoids Data Redundancy** – Prevents repetition of data.
3. **Ensures Data Consistency** – Same data everywhere.
4. **Data Security** – Access is given only to authorized users.
5. **Easy Data Access** – Quick search, update, and reporting.

Relational Data Model

The **Relational Model**, introduced by **E. F. Codd**, organizes data into **tables (relations)**.

Each **table** consists of:

- **Rows (Tuples)** → Represent individual records.
- **Columns (Attributes)** → Represent data fields of a record.

Example:

Table Student

Roll_No	Adm_no	Name	Class	Marks
1	102	Riya	12-A	85
2	108	Aarav	12-B	90

Basic Terms

1. **Relation:**
   A **table** in the database is called relation. A relation consists of **rows and columns**.
   Example: *Student* table above.
2. **Attribute:**
   A **column** in a table.
   Example: Name, Class, Marks.
3. **Tuple:**
   A **row** in a table.
   Example: (1, Riya, 12-A, 85)
4. **Domain:**
   The **set of possible values** for an attribute. Or The set of allowed values for an attribute.
   Example: For "Class" → {12-A, 12-B, 12-C}

   Roll_no- (1,2,3,4,5,6,7,8,9,10,11,12……)

5. **Degree:**
   The **number of attributes** (columns) in a table.
   Example: Student table has 4 attributes → degree = 4
6. **Cardinality:**
   The **number of tuples** (rows) in a table.
   Example: 2 rows → cardinality = 2

Types of Keys

1. **Candidate Key:**
   A field (or combination of fields) that can uniquely identify each record.
   Example: Roll_No, Adm_no
2. **Primary Key:**
   The **main candidate key** chosen to uniquely identify records.
   Example: Roll_No in Student table.
3. **Alternate Key:**
   Other candidate keys that are **not chosen** as the primary key.
   Example: If both Roll_No and Admission_No can identify a student, and Roll_No is chosen as primary key → Admission_No is an alternate key.
4. **Foreign Key:**
   A field that **links two tables** together.
   It refers to the **primary key** of another table.

Example: *Student_ID* in the *Marks* table can be a foreign key referencing *Student_ID* in the *Student* table.

Structured Query Language(SQL)

Category	Full Form	Main Purpose	Common Commands
DDL	Data Definition Language	Defines/changes database structure	`CREATE`, `ALTER`, `DROP`, `TRUNCATE`, `RENAME`
DML	Data Manipulation Language	Inserts, modifies, and deletes data	`INSERT`, `UPDATE`, `DELETE`
DQL	Data Query Language	Retrieves data from database	`SELECT`
DCL	Data Control Language	Controls user access and permissions	`GRANT`, `REVOKE`
TCL	Transaction Control Language	Manages transactions	`COMMIT`, `ROLLBACK`, `SAVEPOINT`, `SET TRANSACTION`

Attribute Name	Data expected to be stored	Data type	Constraint
RollNumber	Numeric value consisting of maximum 3 digits	INT	PRIMARY KEY
SName	Variant length string of maximum 20 characters	VARCHAR(20)	NOT NULL
SDateofBirth	Date value	DATE	NOT NULL
GUID	Numeric value consisting of 12 digits	CHAR (12)	FOREIGN KEY

Table 9.4 Data types and constraints for the attributes of relation GUARDIAN

Attribute Name	Data expected to be stored	Data type	Constraint
GUID	Numeric value consisting of 12 digit Aadhaar number	CHAR (12)	PRIMARY KEY
GName	Variant length string of maximum 20 characters	VARCHAR(20)	NOT NULL
GPhone	Numeric value consisting of 10 digits	CHAR(10)	NULL UNIQUE
GAddress	Variant length String of size 30 characters	VARCHAR(30)	NOT NULL

Table 9.5 Data types and constraints for the attributes of relation ATTENDANCE.

Attribute Name	Data expected to be stored	Data type	Constraint
AttendanceDate	Date value	DATE	PRIMARY KEY*
RollNumber	Numeric value consisting of maximum 3 digits	INT	PRIMARY KEY* FOREIGN KEY
AttendanceStatus	'P' for present and 'A' for absent	CHAR(1)	NOT NULL

means part of composite primary key.

© NCERT not to be republished

DDL (Data Definition Language) commands:

CREATE TABLE STUDENT(-> RollNumber INT, -> SName VARCHAR(20), -> SDateofBirth DATE, -> GUID CHAR (12), -> PRIMARY KEY (RollNumber));

```
CREATE TABLE Student
(
   RollNo INT PRIMARY KEY,
   Name VARCHAR(30) NOT NULL,
   Class VARCHAR(10),
   Marks INT CHECK (Marks >= 0 AND Marks <= 100),
   City VARCHAR(20) DEFAULT 'Prayagraj'
);
```

Describe Table We can view the structure of an already created table using the DESCRIBE statement or DESC statement. Which datatype out of Char and Varchar will you prefer for storing contact number(mobile number)? Discuss. *Syntax:* DESCRIBE tablename; mysql> DESCRIBE STUDENT;

ALTER Table After creating a table, we may realise that we need to add/remove an attribute or to modify the datatype of an existing attribute or to add constraint in attribute.

Add primary key to a relation Let us now alter the tables created in Activity 9.4. The following MySQL statement adds a primary key to the GUARDIAN relation: mysql> ALTER TABLE GUARDIAN ADD PRIMARY KEY (GUID);

Add foreign key to a relation Once primary keys are added, the next step is to add foreign keys to the relation (if any). Following points need to be observed while adding foreign key to a relation: • The referenced relation must be already created. • The referenced attribute(s) must be part of the primary key of the referenced relation. • Data types and size of referenced and referencing attributes must be the same.

Syntax: ALTER TABLE table_name ADD FOREIGN KEY(attribute name) REFERENCES referenced_table_name (attribute name);

Add constraint UNIQUE to an existing attribute In GUARDIAN table, the attribute GPhone has a constraint UNIQUE which means no two values in that column should be the same.

Syntax: ALTER TABLE table_name ADD UNIQUE (attribute name);

Add an attribute to an existing table Sometimes, we may need to add an additional attribute in a table. It can be done using the ADD attribute statement as shown in the following Syntax: ALTER TABLE table_name ADD attribute name DATATYPE;

Modify datatype of an attribute We can change data types of the existing attributes of a table using the following ALTER statement. *Syntax:* ALTER TABLE table_name MODIFY attribute DATATYPE;

Modify constraint of an attribute

Syntax: ALTER TABLE table_name MODIFY attribute DATATYPE NOT NULL;

Add default value to an attribute If we want to specify default value for an attribute, then use the following syntax: ALTER TABLE table_name MODIFY attribute DATATYPE DEFAULT default_value;

Remove an attribute Using ALTER, we can remove attributes from a table, as shown in the following syntax: ALTER TABLE table_name DROP attribute; To remove the attribute income from table GUARDIAN (Table 9.4), write the following MySQL statement:

Remove primary key from the table Sometime there may be a requirement to remove primary key constraint from the table. In that case, Alter table command can be used in the following way: *Syntax:* ALTER TABLE table_name DROP PRIMARY KEY;

DROP Statement Sometimes a table in a database or the database itself needs to be removed. We can use a DROP statement to remove a database or a table permanently from the system. However, one should be very cautious while using this statement as it cannot be undone. *Syntax to drop a table:* DROP TABLE table_name; *Syntax to drop a database:* DROP DATABASE database_name;

SQL FOR DATA MANIPULATION

INSERTION of Records INSERT INTO statement is used to insert new records in a table. Its syntax is: INSERT INTO tablename VALUES(value 1, value 2,....);

INSERT INTO GUARDIAN -> VALUES (444444444444, 'Amit Ahuja', -> 5711492685, 'G-35,Ashok vihar, Delhi');

SQL FOR DATA QUERY

SELECT Statement

The SQL statement SELECT is used to retrieve data from the tables in a database and the output is also displayed in tabular form. *Syntax:* SELECT attribute1, attribute2, ... FROM table_name WHERE condition;

Example 9.2 The following query retrieves the name and date of birth of student with roll number 1: mysql> SELECT SName, SDateofBirth -> FROM STUDENT -> WHERE RollNumber = 1;

(A) Retrieve selected columns The following query selects employee numbers of all the employees: mysql> SELECT EmpNo FROM EMPLOYEE;

(B) Renaming of columns In case we want to rename any column while displaying the output, it can be done by using the alias 'AS'. The following query selects Employee name as Name in the output for all the employees:
mysql> SELECT EName as Name FROM EMPLOYEE;

Distinct Clause By default, SQL shows all the data retrieved through query as output. However, there can be duplicate values. The SELECT statement when combined with DISTINCT clause, returns records without repetition (distinct records). For example, while retrieving a department number from employee relation, there can be duplicate values as many employees are assigned to the same department. To select unique department number for all the employees, we use DISTINCT as shown below:
mysql> SELECT DISTINCT DeptId FROM EMPLOYEE;

WHERE Clause The WHERE clause is used to retrieve data that meet some specified conditions. In the OFFICE database, more than one employee can have the same salary. Following query gives distinct salaries of the employees working in the department number D01:
mysql> SELECT DISTINCT Salary -> FROM EMPLOYEE -> WHERE Deptid='D01';

Example 9.4 Display all the details of those employees of D04 department who earn more than 5000.
Example 9.5 The following query selects records of all the employees except Aaliya.
mysql> SELECT * FROM EMPLOYEE -> WHERE NOT Ename = 'Aaliya';

Example 9.6 The following query selects the name and department number of all those employees who are earning salary between 20000 and 50000 (both values inclusive).
mysql> SELECT Ename, DeptId -> FROM EMPLOYEE -> WHERE Salary>=20000 AND Salary<=50000;

SELECT * FROM EMPLOYEE WHERE Salary > 5000 OR DeptId= 20;
mysql> SELECT Ename, DeptId -> FROM EMPLOYEE -> WHERE Salary BETWEEN 20000 AND 50000;

Example 9.7 The following query selects details of all the employees who work in the departments having deptid D01, D02 or D04.

Membership operator IN The IN operator compares a value with a set of values and returns true if the value belongs to that set. The above query can be rewritten using IN operator as shown below:

mysql> SELECT * FROM EMPLOYEE -> WHERE DeptId IN ('D01', 'D02' , 'D04');

Example 9.8 The following query selects details of all the employees except those working in department number D01 or D02.

mysql> SELECT * FROM EMPLOYEE -> WHERE DeptId NOT IN('D01', 'D02');

(F) ORDER BY Clause

ORDER BY clause is used to display data in an ordered form with respect to a specified column. By default, ORDER BY displays records in ascending order of the specified column's values. To display the records in descending order, the DESC (means descending) keyword needs to be written with that column.

Example 9.9 The following query selects details of all the employees in ascending order of their salaries**. mysql> SELECT * FROM EMPLOYEE -> ORDER BY Salary DESC;**

(G) Handling NULL Values SQL supports a special value called NULL to represent a missing or unknown value. For example, the Gphone column in the GUARDIAN table can have missing value for certain records. Hence, NULL is used to represent such unknown values. It is important to note that NULL
is different from 0 (zero). Also, any arithmetic operation performed with NULL value gives NULL. For example: 5 + NULL = NULL because NULL is unknown hence the result is also unknown. In order to check for NULL value in a column, we use IS NULL operator. *Example 9.11* The following query selects details of all those employees who have not been given a bonus. This implies that the bonus column will be blank.

mysql> SELECT * FROM EMPLOYEE -> WHERE Bonus IS NULL;

Example 9.12 The following query selects names of all employees who have been given a bonus (i.e., Bonus is not null) and works in the department D01.

mysql> SELECT EName FROM EMPLOYEE -> WHERE Bonus IS NOT NULL -> AND DeptID = 'D01';

H) Substring pattern matching

Many a times we come across situations where we do not want to query by matching exact text or value. Rather, we are interested to find matching of only a few characters or values in column values. For example, to find out names starting with "T" or to find out

pin codes starting with ‘60’. This is called substring pattern matching. We cannot match such patterns using = operator as we are not looking for an exact match. SQL provides a LIKE operator that can be used with the WHERE clause to search for a specified pattern in a column. The LIKE operator makes use of the following two wild card characters: • % (per cent)- used to represent zero, one, or multiple characters • _ (underscore)- used to represent exactly a single character

Example 9.13 The following query selects details of all those employees whose name starts with 'K'. mysql> SELECT * FROM EMPLOYEE -> WHERE Ename like 'K%';

Example 9.14 The following query selects details of all those employees whose name ends with 'a', and gets a salary more than 45000.
mysql> SELECT * FROM EMPLOYEE -> WHERE Ename like '%a' -> AND Salary > 45000;
Example 9.15 The following query selects details of all those employees whose name consists of exactly 5 letters and starts with any letter but has ‘ANYA’ after that. mysql> SELECT * FROM EMPLOYEE -> WHERE Ename like '_ANYA';

Example 9.16 The following query selects names of all employees containing 'se' as a substring in name.
mysql> SELECT Ename FROM EMPLOYEE -> WHERE Ename like '%se%';

Example 9.17 The following query selects names of all employees containing 'a' as the second character. mysql> SELECT EName FROM EMPLOYEE -> WHERE Ename like '_a%';

DATA UPDATION AND DELETION
The UPDATE statement is used to make such modifications in existing data. ***Syntax:*** UPDATE table_name SET attribute1 = value1, attribute2 = value2, ... WHERE condition;

Data Deletion DELETE statement is used to delete/remove one or more records from a table. *Syntax:* DELETE FROM table_name WHERE condition;
Aggregate Functions
MAX(column)
mysql> SELECT MAX(Price) FROM INVENTORY;

MIN(Column)
mysql> SELECT MIN(Price) FROM INVENTORY;
AVG(column)
mysql> SELECT AVG(Price) FROM INVENTORY;
SUM(column)
mysql> SELECT SUM(Price) FROM INVENTORY;

COUNT(*)
mysql> SELECT COUNT(*) from MANAGER;

COUNT(column)
mysql> SELECT * from MANAGER;

Example 9.22 a) Display the total number of records from table INVENTORY having a model as VXI. mysql> SELECT COUNT(*) FROM INVENTORY WHERE Model=”VXI”;
Display the average price of all the cars with Model LXI from table INVENTORY.
mysql> SELECT AVG(Price) FROM INVENTORY WHERE Model="LXI";

GROUP BY CLAUSE IN SQL At times we need to fetch a group of rows on the basis of common values in a column. This can be done using a group by clause. It groups the rows together that contains the same values in a specified column. We can use the aggregate functions (COUNT, MAX, MIN, AVG and SUM) to work on the grouped values. HAVING Clause in SQL is used to specify conditions on the rows with Group By clause.
Example 9.23 **a) Display** the number of Cars purchased by each Customer from SALE table. mysql> SELECT CustID, COUNT(*) "Number of Cars" FROM SALE GROUP BY CustID;

Display the Customer Id and number of cars purchased if the customer purchased more than 1 car from SALE table. mysql> SELECT CustID, COUNT(*) FROM SALE GROUP BY CustID HAVING Count(*)>1;

Display the number of people in each category of payment mode from the table SALE.
mysql> SELECT PaymentMode, COUNT(PaymentMode) FROM SALE GROUP BY Paymentmode ORDER BY Paymentmode;

Display the PaymentMode and number of payments made using that mode more than once. mysql> SELECT PaymentMode, Count(PaymentMode) FROM SALE GROUP BY Paymentmode HAVING COUNT(*)>1 ORDER BY Paymentmode;
Database is a collection of related tables. MySQL is a 'relational' DBMS.
• DDL (Data Definition Language) includes SQL statements such as, Create table, Alter table and Drop table.
• DML (Data Manipulation Language) includes SQL statements such as, insert, select, update and delete. • A table is a collection of rows and columns, where each row is a record and columns describe the feature of records.
• ALTER TABLE statement is used to make changes in the structure of a table like adding, removing or changing datatype of column(s).
• UPDATE statement is used to modify existing data in a table.
• WHERE clause in SQL query is used to enforce condition(s)
. • DISTINCT clause is used to eliminate repetition and display the values only once.

The BETWEEN operator defines the range of values inclusive of boundary values.
• The IN operator selects values that match any value in the given list of values.
• NULL values can be tested using IS NULL and IS NOT NULL.
• ORDER BY clause is used to display the result of a SQL query in ascending or descending order with respect to specified attribute values. By default, the order is ascending.
• LIKE operator is used for pattern matching. % and _ are two wild card characters. The per cent (%) symbol is used to represent zero or more characters. The underscore (_) symbol is used to represent a single character.
• A Function is used to perform a particular task and return a value as a result.
• Single Row functions work on a single row of the table and return a single value.
• Multiple Row functions work on a set of records as a whole and return a single value. Examples include COUNT, MAX, MIN, AVG and SUM.
• GROUP BY function is used to group rows of a table that contain the same values in a specified column.
• Join is an operation which is used to combine rows from two or more tables based on one or more common fields between them.

SQL Queries:

Q1. Consider the following tables Product and Client. Write SQL commands for the statement (i) to (iv) and give outputs for SQL queries (v) to (viii).

Table: **PRODUCT**

P_ID	Product Name	Manufacturer	Price
TP01	TalcomPowder	LAK	40
FW05	Face Wash	ABC	45
BS01	Bath Soap	ABC	55
SH06	Shampoo	XYZ	120
FW12	Face Wash	XYZ	95

Table: CLIENT

C_ID	Client Name	City	P_ID
01	TalcomPowder	Delhi	FW05
06	Face Wash	Mumbai	BS01
12	Bath Soap	Delhi	SH06
15	Shampoo	Delhi	FW12
16	Face Wash	Banglore	TP01

(i) To display the details of those Clients whose city is Delhi.

Ans: Select* from Client where City=”Delhi”;

(ii) To display the details of Products whose Price is in the range of 50 to 100(Both values included).

Ans: Select* from product where Price between 50 and 100;

(iii) To display the ClientName, City from table Client, and ProductName and Price from table Product, with their corresponding matching P_ID.

Ans: Select ClientName,City,ProductName, Price from Product,Client where Product.P_ID=Client.P_ID;

(iv) To increase the Price of all Products by 10

Ans: Update Product Set Price=Price +10;

(v) SELECT DISTINCT Address FROM Client.

Ans:

City
Delhi
Mumbai
Bangalore

(vi) SELECT Manufacturer, MAX(Price), Min(Price), Count(*) FROM Product GROUP BY Manufacturer;

Ans:

Manufacturer	**Max(Price)**	**Min(Price)**	**Count(*)**
LAK	40	40	1
ABC	55	45	2
XYZ	120	95	2

(vii) SELECT ClientName, ManufacturerName FROM Product, Client WHERE Client.Prod_Id=Product.P_Id;

Ans:

ClientName	**ManufacturerName**
CosmeticShop	ABC
TotalHealth	ABC
LiveLife	XYZ
PrettyWoman	XYZ
Dreams	LAK

(viii) SELECT ProductName, Price * 4 FROM Product.

Ans:

ProductName	Price*4
TalcomPoweder	160
FaceWash	180
BathSoap	220
Shampoo	480
Face Wash	380

Q2. Consider the following tables Item and Customer. Write SQL commands for the statement (i) to (iv) and give outputs for SQL queries (v) to (viii). **Table: ITEM**

C_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

Table: CUSTOMER

C_ID	CustomerName	City	P_ID
01	N.Roy	Delhi	LC03
06	H.Singh	Mumbai	PC03
12	R.Pandey	Delhi	PC06
15	C.Sharma	Delhi	LC03
16	K.Agarwalh	Banglore	PC01

(i) To display the details of those Customers whose city is Delhi.

Ans: Select* from Customer Where City=”Delhi”;

(ii) To display the details of Item whose Price is in the range of 35000 to 55000 (Both values included).

Ans: Select* from Item Where Price>=35000 and Price <=55000;

(iii) To display the CustomerName, City from table Customer, and ItemName and Price from table Item, with their corresponding matching I_ID.

Ans: Select CustomerName,City,ItemName, Price from Item,Customer where Item.I_ID=Customer.I_ID;

(iv) To increase the Price of all Items by 1000 in the table Item.

Ans: Update Item set Price=Price+1000;

(v) SELECT DISTINCT City FROM Customer.

Ans:

City
Delhi
Mumbai
Bangalore

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

Ans:

ItemName	Max(Price)	Count(*)
PersonalComputer	37000	3
Laptop	57000	2

(vii) SELECT CustomerName, Manufacturer FROM Item, Customer WHERE Item.Item_Id=Customer.Item_Id;

Ans:

CustomerName	Manufacturer Name
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:

ItemName	Price*100

PersonalComputer	3500000
Laptop	5500000

Q3. Consider the following tables Consignor and Consignee. Write SQL command for the statements(i)to(iv) And give outputs for the SQL quries (v) to (viii).

TABLE : CONSIGNOR

CnorID	**CnorName**	**CnorAddress**	**City**
ND01	R singhal	24,ABC Enclave	New Delhi
ND02	AmitKumar	123,Palm Avenue	New Delhi
MU15	R Kohil	5/A,South,Street	Mumbai
MU50	S Kaur	7-K,Westend	Mumbai

TABLE : CONSIGNEE

CneeID	**CnorID**	**CneeName**	**CneeAddress**	**CneeCity**
MU05	ND01	RahulKishore	5,Park Avenue	Mumbai
ND08	ND02	P Dhingr a	16/j,Moore Enclave	New Delhi
KO19	MU15	A P Roy	2A,Central/ avenue	Kolkata
MU32	ND0 2	S mittal	P 245, AB Colony	Mumbai
ND48	MU5 0	B P jain	13,Block d,a,viha	New Delhi

(i) To display the names of all consignors from Mumbai.

Ans: Select CnorName from Consignor where city=”Mumbai”;

(ii) To display the cneeID, cnorName, cnorAddress, CneeName, CneeAddress for every Consignee.

Ans: Select CneeId, CnorName, CnorAddress, CneeName, CneeAddress from Consignor,Consignee where Consignor.CnorId=Consignee.CnorId;

(iii) To display the consignee details in ascending order of CneeName.

Ans: Select * from Consignee Orderby CneeName Asc;

(iv) To display number of consignors from each city.

Ans: Select city, count(*) from Consignors group by city;

(v) SELECT DISTINCT City FROM CONSIGNEE;

Ans:

CneeCity
Mumbai
New Delhi
Kolkata

(vi) SELECT A.CnorName, B.CneeName FROM Consignor A, Consignee B WHERE A.CnorID=B.CnorID AND B.CneeCity='Mumbai';

Ans:

CnorName	**CneeName**

(vii) SELECT CneeName,CneeAddress FROM Consignee WHERE CneeCity Not IN ('Mumbai', 'Kolkata');

Ans:

CneeName	**CneeAddress**
P Dhingr a	16/j,Moore Enclave
B P jain	13,Block d,a,viha

(viii) SELECT CneeID, CneeName FROM Consignee WHERE CnorID = 'MU15' OR CnorID = 'ND01';

senderCity
New Delhi
Mumbai

Ans:

CneeID	**CneeName**
MU05	Rahul
KO19 A P Roy	Kishore

Q4. Consider the following tables. Write SQL command for the statements (i)to(iv)and give outputs for the SQL quries (v) to (viii).

TABLE : SENDER

SenderID	SenderName	Sender Address	Sender City
ND01	R jain	2,ABC Appts	New Delhi
MU02	H sinha	12, Newton	Mumbai
MU1 5	S haj	27/ A,Park Street	New Delhi
ND5 0	T Prasad	122-K,SDA	Mumbai

TABLE :RECIPIENT

RecID	SenderID	ReCName	RecAddress	ReCCity
KO05	ND01	RBajpayee	5,Central Avenue	Kolkata
ND08	MU0 2	S Mahajan	116, A Vihar	NewDelhi
MU19	ND01	H sing	2A,Andheri East	Mumbai
MU32	MU1 5	PK Swamy	B5, CS erminus	Mumbai
ND48	ND50	S Tripathi	13, B1 D,Mayur Vihar	NewDelhi

(i) To display the names of all senders from Mumbai.

Ans: Select * from Sender where SenderCity ='Mumbai';

(ii) To display the recID, senderName, senderAddress, RecName, RecAddress for every recipt.

Ans: Select recID, SenderName, SenderAddress, RecName, RecAddress from Sender, Recipient where Sender.Senderid=Recipient.RenderId;

(iii) To display the sender details in ascending order of SenderName.

Ans: Select * from Sender order by SenderName;

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

Ans: Select RecCity,Count(*) from Recipient group by RecCity;

(v) SELECT DISTINCT SenderCity FROM Sender;

Sender

Ans:

City
New Delhi
Mumbai

(vi) SELECT A.SenderName A, B.RecName FROM Sender A, Recipient B WHERE A.SenderID=B. SenderID AND B.RecCity='Mumbai';

Ans:

SenderName	RecName
R.Jain	H.Singh
S.Jha	P.K.Swamy

(vii) SELECT RecName,RecAddress FROMRecipient WHERE RecCity Not IN ('Mumbai',Kolkata');

Ans:

RecName	RecAddressS
Mahajan	116,A Vihar
S Tripati	13, B1 D, Mayur Vihar

(viii) SELECT RecID, RecName FROM Recipient WHERE SenderID = 'MU02' OR SenderID = 'ND50';

Ans:

RecID	RecName
ND08	S Mahajan
ND48	S Tripathi

Q5. Study the following tables FLIGHTS and FARES and write SQL commands for the questions (i) to (iv).

TABLE: FLIGHTS

FL_NO	STARTING	ENDING	NO_ FLGHTS	NO_ STOPS
IC301	MUMBAI	DELHI	8	0
IC799	BANGALORE	DELHI	2	1
MC101	INDORE	MUMBAI	3	0

IC302	DELHI	MUMBAI	8	0
AM812	KANPUR	BANGLORE	3	1
IC899	MUMBAI	KOCHI	1	4
AM501	DELHI	TRIVENDRUM	1	5
MU499	MUMBAI	MADRAS	3	3
IC701	DELHI	AHMEDABAD	4	0

TABLE:FLIGHTS

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	1050	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=”KANPUR” AND ENDING=”BANGALORE”;

(ii) Arrange the contents of the table FLIGHTS in the ascending order of FL_NO.

Ans: SELECT * From FLIGHTS ORDER BY FL_NO ASC;

(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 paid = FARE+FARE+TAX%/100.

Ans: Select FL_NO, FARE+FARE+(TAX%/100) from FLIGHTS, FARES where Starting=”DELHI” AND Ending=”MUMBAI”;

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

Ans: Select min(FARE) from FARES Where AIRLINES=”Indian Airlines”;

Q6. 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	Johan	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	Murphy	ORTHOPEDIC	M	15

TABLE: SALARY

ID	BASIC	ALLOWANCE	CONSULTAION
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 years 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. DOCTORS and SALARY Salary =BASIC+ALLOWANCE.

Ans: Select avg(basic+allowance) from Doctor,Salary where Dept=”Ent” and Doctor.Id=Salary.Id;

(iii) Display the minimum ALLOWANCE of female doctors.

Ans: Select min(Allowance) from Doctro,Salary where Sex=”F” and Doctor.Id=Salary.Id;

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

Ans: Select max(Consulation) from Doctor,Salary where Sex=”M” and Doctor.Id=Salary.Id;

(v) SELECT count (*) from DOCTOR where SEX = “F”

Ans: 4

(vi) SELECT NAME, DEPT , BASIC from DOCTOR, SALRY Where DEPT = “ENT” AND DOCTOR.ID = SALARY.ID

Ans:

Name	Dept	Basic
Jonah	Ent	12000

Q7. Consider the following tables EMPLOYEES and EMPSALARY. write SQL commands for the Statements (i) to (iv) and give outputs for SQL quires (v) to (viii).

EMPID	FIRSTNAME	LASTNAME	ADDRESS	CITY
010	GEORGE	Smith	83 First Street	Howard
105	MARY	Jones	842VineAve	Losantiville
152	SAM	Tones	33 Elm st	Paris
215	SARAH	Ackerman	440 U.S.110	Upton
244	MANILA	Sengupta	24 FriendsStreet	New Delhi
300	ROBERT	Samuel	9 Fifth Cross	Washington
335	HENRY	Williams	12 Moore Street	Boston
400	RACHEL	Lee	121 Harrison	New York
441	PETER	Thompson	11 Red road	Paris

EMPSALRAY

EMPID	SALARY	BENEFITS	DESIGNATION

010	75000	15000	Manager
105	65000	15000	Manager
152	80000	25000	Director
215	75000	12500	Manager
244	50000	12000	Clerk
300	45000	10000	Clerk
335	40000	10000	Clerk
400	32000	7500	Salesman
441	28000	7500	Salesman

(i) To display Firstname, Lastname, Address and City of all employees living in Paris from the table EMPLOYEES.

Ans: Select Firstname,Lastname,Address,City from Employees where City=”Paris”;

(ii) To display the content of EMPLOYEES table in descending order of FIRSTNAME.

Ans: Select * from Employees Order By Firstname Desc;

(iii) To display the Firstname, Lastname, and Total Salary of all managers from the tables, where Total Salary is calculated as Salary+Benifts.

Ans: Select Firstname,Lastname,Salary+Benefits from Employees, Empsalary where Designation=”Manager” and Employees.EmpId=EmpSalary.EmpId;

(iv) To display the Maximum salary among Managers and Clerks from the table EMPSALARY.

Ans: Select Designation,max(Salary) from EmpSalary where Designation=”Manager” or Designation=”Clerk”;

(v) SELECT FIRSTNAME,SALARY FROM EMPLOYEES,EMPSALARY WHERE DESTINATION =’Salesman’AND EMPOLYEES.EMPID=EMPSALARY.EMPID ;

Ans:

Firstname	**Salary**
Rachel	32000
Peter	28000

(vi) SELECT COUNT (DISTINT DESIGNATION) FROM EMPSALARY

Ans: 4

(vii) SELECT DESIGNATION , SUM(SALARY) FROM EMPSALARY GROUP BY DESIGNATION HAVING COUNT(*)>2;

Ans:

Designation	Sum(Salary)
Manager	215000
Clerk	135000

(viii) SELECT SUM (BENEFITS) FROM EMPSALARY WHERE DESIGNATION='Clerk';

Ans: 32000

Q8. 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).

WORKERS

W_ID	FIRSTNAME	LASTNAME	ADDRESS	CITY
102	Sam	Tones	33 Elm St.	Paris
105	Sarah	Ackerman	44 U.S.110	NewYork
144	Manila	Sengup ta	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	12Moore Street	Boston
403	Ronny	Lee	121 Harrison St.	New York
451	Pat	Thomps on	11 Red Road	Paris

DESIG

W_ID	SALARY	BENEFITS	DESIGINATION
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 andCity of all employees living in New York fromthe 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 Worker Order By lastname Asc;

(iii) To display the FIRSTNAME, LASTNAME and Total Salary of all Clerks from the tables WORKERS And DESIG, where Total salary is calculated as Salary + benifts.

Ans: Select firstname, lastname, salary+benefits where worker.w_id=desg.w_id and Designation=”Clerk”;

(iv) To display the minimum salary among managers and Clerks from the tables DESIG.

Ans: Selet DESIGNATION,min(SALARY) From DESIG Group By DESIGNATION Having DESIGNATION In(“Clerk”,”Manager”);

(v) SELECT FIRSTNAME, SALARY FROM WORKERS, DESIG WHERE DESIGINATION = “MANAGER” AND WORKERS.W_ID = DESIGN.W_ID

Ans:

FIRSTNAME	**SALARY**
Sam	75000
Manila	70000
George	75000

(vi)SELECT COUNT(DISTINCT DESIGNATION) FROM DESIGN ;

Ans: 4

(vii) SELECT DESIGNATION, SUM(SALARY) FROM DESIG GROUP BY DESIGNATION HAVING COUNT (*) < 3;

Ans:

Designation	Sum(Salary)
Director	85000
Salesman	60000

(viii) SELECT SUM(BENIFTS) FROM DESIG WHERE DESIGINATION =”salesman”;

Ans: 15000

Q9. Give the following table for database a LIBRARY.**TABLE : BOOKS**

BOOK_ID	BOOK_NAME	AUTHONAME	PUBLISHER	PRICE	TYPE	QUANTITY
F0001	The Tears	William Hopkins	First Publ	750	Fiction	10
F0002	Thund erbolts	Anna Roberts	First Publ.	700	Fiction	5
T0001	My first PYTHON	Brains & Brooke	EPB	250	Text	10
T0002	PYTHON Brain works	A.W.Ros saine	TDH	325	Text	5
C001	Fast Cook	Lata Kapoore	EPB	350	Cookery	8

TABLE:ISSUED

BOOK_ID	QUANTITY_ISSUED
F0001	3
T0001	1
C0001	5

Write SQL queries from b to g.

(i) To show Book name, Author name and Price of books of EPB publisher.

Ans: select Book_name,Author_name, price from books where Publisher =”EPB”;

(ii) To list the names of the books of FICTIONS type.

Ans: Select Book_name from books where type=”FICTION”;

(iii) To display the names and prices of the books in descending order of their price.

Ans: Select Book_name, price from books order by price desc;

(iv) To increase the price of all books of First Pub.by 50.

Ans: update books set price= price+50 where publishers = “First Publ”;

(v) To Display the Book_ID, Book_Name and Quantity Issued for all books Which have been issued.

Ans:Select Book_ID, Book_Name, Quantity_Issued from Books,Issued where Books.BookId=Issued.BookId;

(vi) To insert a new row in the table Issued having the following data: “F0002”,4

Ans: insert into Issued values(“F0002”,4);

(vii) Give the output of the following queries on the above tables:

(1) Select Count(Distinct Publishers) From Books;

Ans: 3

(2) Select Sum(Price) From Books Where Quantity>5;

Ans: 1350.

(3) Select Book_Name,Author_Name From Books Where Price<500;

Ans: Book_Name Author_Name My First PYTHON Brian & Brooks PYTHON Brainworks A.W.Rossaine Fast Cook Lata Kapoor.

(4) Select Count(*) From Books;

Ans: 5

Q10. Write SQL commands for (b) to (g) and write the outputs for (h) on the basis of tables TNTERIORS and NEWONES.

TABLE: INTERIORS

NO	ITEM	TYPE	DATEOFSTOCK	PRICE	DISCOUNT

	NAME				
1	Red rose	DoubleBed	23/02/02	32000	15
2	Soft touch	Baby cot	20/01/02	9000	10
3	Jerry'shome	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	DinningTable	19/02/02	11000	15

TABLE:**NEWONES**

NO	ITEMNAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT
11	White wood	Doublebed	23/03/03	20000	20
12	James007	Sofa	20/02/03	15000	15
13	Tom look	Baby cot	21/02/03	7000	10

(i) To show all information about the sofas from the INTERIORS table.

Ans: Select * from INTERIORS where type= "sofa";

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

Ans: Select Itemname,Type From Interiors Where Dateofstock<{22/01/02} order by Itemname;

(iii) To display ITEMNAME and DATEOFSTOCK of those items in which the Discount percentage is more than 15 from INTERIORS.

Ans: Select Itemname,Dateofstock from Interiors Where Discount>15;

(iv) To count the number of items whose type is "Double bed";

Ans: Select Count(*) from Interiors Where Type="Double Bed";

(v) To insert new row in the NEWONES table with the following data:14, "True Indian ", "Office Table ", {28/03/03},15000,20.

Ans: Insert into Newones values(14,”True Indian”,”Office Table”,”{28/03/03},15000,20);

(vi) Give the outputs for the following SQL statements.

(1) Select COUNT (distinct TYPE) from INTERIORS;

Ans: 5

(2) Select AVG(DISCOUNT)from INTERIORS where TYPE =”Baby cot”;

Ans: 13

(3) Select SUM(price)from INTERIORS where DATEOFSTOCK<{12/02/02};

Ans: 53000