CHAPTER 11 – INTRODUCTION TO MySQL

MYSQL

It is freely available open source Relational Database Management System (RDBMS) that uses **Structured Query Language(SQL).** In MySQL database , information is stored in Tables. A single MySQL database can contain many tables at once and store thousands of individual records.

FEATURES OF MySQL

1. **Speed :** If the server hardware is optimal, MySQL runs very fast.
2. **Cost :** MySQL is available free of cost.
3. **Portability :** MySQL is portable also, since it can run on many different platforms.
4. **Security :** MySQL offers a privilege and password system that is very flexible and secure.

SQL (Structured Query Language)

SQL is a language that enables you to create and operate on relational databases, which are sets of related information stored in tables.

DIFFERENT DATA MODELS

A **data model refers to a set of concepts to describe the structure of a database**, and certain constraints (restrictions) that the database should obey. The four data model that are used for database management are :

1. **Relational data model** : In this data model, the data is organized into tables (i.e. rows and columns). These tables are called relations.

2. **Hierarchical data model 3. Network data model 4. Object oriented data model**

RELATIONAL MODEL TERMINOLOGY

1. **Relation :** A table storing logically related data is called a Relation.

2. **Tuple :** A **row of a relation** is generally referred to as a tuple.

3. **Attribute :** A **column** of a relation is generally referred to as an attribute.

4. **Degree :** This refers to the **number of attributes** in a relation.

5. **Cardinality :** This refers to the **number of tuples** in a relation.

6. **Primary Key :** This refers to a set of one or more attributes that can uniquely identify tuples within the relation.

7. **Candidate Key :** All attribute combinations inside a relation that can serve as primary key are candidate keys as these are candidates for primary key position.

8. **Alternate Key :** A candidate key that is not primary key, is called an alternate key.

9. **Foreign Key :** A non-key attribute, whose values are derived from the primary key of some other table, is known as foreign key in its current table.

REFERENTIAL INTEGRITY

- A referential integrity is a system of rules that a DBMS uses to ensure that relationships between records in related tables are valid, and that users don't accidentally delete or change related data. This integrity is ensured by foreign key.

CLASSIFICATION OF SQL STATEMENTS

SQL commands can be mainly divided into following categories:

1. **Data Definition Language(DDL) Commands**
   Commands that allow you to perform task, related to data definition e.g;
   - Creating, altering and dropping tables.
   - Granting and revoking privileges and roles.
   - Maintenance commands.

2. **Data Manipulation Language(DML) Commands**
   Commands that allow you to perform data manipulation e.g., retrieval, insertion, deletion and modification of data stored in a database.
   DML are basically of two types : 1. Procedural DML 2. Non-Procedural DML

   PROCEDURAL DML
   It requires a user to specify what data is needed and how to get it.
   NON PROCEDURAL DML
   It require a user to specify what data is needed without specifying how to get it.

3. **Transaction Control Language(TCL) Commands**
   Commands that allow you to manage and control the transactions e.g.,
   - Making changes to database, permanent
   - Undoing changes to database, permanent
   - Creating save points
   - Setting properties for current transactions.

MySQL ELEMENTS

1. Literals 2. Datatypes 3. Nulls 4. Comments

LITERALS

It refer to a fixed data value. This fixed data value may be of character type or numeric type. For example, 'replay' , 'Raj', '8' , '306' are all character literals.
Numbers not enclosed in quotation marks are numeric literals. E.g. 22 , 18 , 1997 are all numeric literals.
Numeric literals can either be integer literals i.e., without any decimal or be real literals i.e. with a decimal point e.g. 17 is an integer literal but 17.0 and 17.5 are real literals.

DATA TYPES

Data types are means to identify the type of data and associated operations for handling it. MySQL data types are divided into three categories:

- Numeric
- Date and time
- String types

Numeric Data Type

1. int – used for number without decimal.
2. Decimal(m,d) – used for floating/real numbers. m denotes the total length of number and d is number of decimal digits.

Date and Time Data Type

1. date – used to store date in YYYY-MM-DD format.

2. time – used to store time in HH:MM:SS format.

String Data Types

1. char(m) – used to store a fixed length string. **m** denotes max. number of characters.
2. varchar(m) – used to store a variable length string. **m** denotes max. no. of characters.

DIFFERENCE BETWEEN CHAR AND VARCHAR DATA TYPE

S.NO.	Char Datatype	Varchar Datatype
1.	It specifies a **fixed length** character String.	It specifies a **variable length** character string.
2.	When a column is given datatype as CHAR(**n**), then MySQL ensures that all values stored in that column have this length i.e. **n** bytes. If a value is shorter than this length **n** then blanks are added, but the size of value remains **n** bytes.	When a column is given datatype as VARCHAR(**n**), then the maximum size a value in this column can have is **n** bytes. Each value that is stored in this column store exactly as you specify it i.e. no blanks are added if the length is shorter than maximum length **n**.

NULL VALUE

If a column in a row has no value, then column is said to be **null** , or to contain a null. **You should use a null value** when the actual value is not known or when a value would not be meaningful.

DATABASE COMMNADS

1. VIEW EXISTING DATABASE

To view existing database names, the command is : **SHOW DATABASES ;**

2. CREATING DATABASE IN MYSQL

For creating the database in MySQL, we write the following command :

CREATE DATABASE <databasename> ;

e.g. In order to create a database Student, command is :

CREATE DATABASE Student ;

3. ACCESSING DATABASE

For accessing already existing database , we write :

USE <databasename> ;

e.g. to access a database named Student , we write command as :

USE Student ;

4. DELETING DATABASE

For deleting any existing database , the command is :

DROP DATABASE <databasename> ;

e.g. to delete a database , say student, we write command as ;

DROP DATABASE Student ;

5. VIEWING TABLE IN DATABASE

In order to view tables present in currently accessed database , command is : **SHOW TABLES ;**

CREATING TABLES IN MYSQL

- Tables are created with the CREATE TABLE command. When a table is created, its columns are named, data types and sizes are supplied for each column.

Syntax of CREATE TABLE command is :

CREATE TABLE <table-name>
(<column name> <data type> ,
<column name> <data type> ,
…… **) ;**

E.g. in order to create table EMPLOYEE given below :

ECODE	ENAME	GENDER	GRADE	GROSS

We write the following command :

```
CREATE TABLE employee
(    ECODE integer ,
     ENAME varchar(20) ,
     GENDER char(1) ,
     GRADE  char(2) ,
     GROSS integer      ) ;
```

INSERTING DATA INTO TABLE

- The rows are added to relations(table) using INSERT command of SQL. Syntax of INSERT is :

```
INSERT INTO <tablename>  [<column list>]
VALUE ( <value1> , <value2> , …..) ;
```

e.g. to enter a row into EMPLOYEE table (created above), we write command as :

```
INSERT INTO employee
VALUES(1001 , 'Ravi' , 'M' , 'E4' , 50000);
```

OR

```
INSERT INTO employee (ECODE , ENAME , GENDER , GRADE , GROSS)
VALUES(1001 , 'Ravi' , 'M' , 'E4' , 50000);
```

ECODE	**ENAME**	**GENDER**	**GRADE**	**GROSS**
1001	Ravi	M	E4	50000

In order to insert another row in EMPLOYEE table , we write again INSERT command :

```
INSERT INTO employee
VALUES(1002 , 'Akash' , 'M' , 'A1' , 35000);
```

ECODE	**ENAME**	**GENDER**	**GRADE**	**GROSS**
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000

INSERTING NULL VALUES

- To insert value NULL in a specific column, we can type NULL without quotes and NULL will be inserted in that column. E.g. in order to insert NULL value in ENAME column of above table, we write INSERT command as :

```
INSERT INTO EMPLOYEE
VALUES (1004 , NULL , 'M' , 'B2' , 38965 ) ;
```

ECODE	**ENAME**	**GENDER**	**GRADE**	**GROSS**
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000
1004	NULL	M	B2	38965

MODIFYING DATA IN TABLES

you can modify data in tables using UPDATE command of SQL. The UPDATE command specifies the rows to be changed using the WHERE clause, and the new data using the SET keyword. Syntax of update command is :

```
UPDATE <tablename>
SET <columnname>=value , <columnname>=value
 WHERE <condition> ;
```

e.g. to change the salary of employee of those in EMPLOYEE table having employee code 1009 to 55000.

```
UPDATE  EMPLOYEE
SET GROSS = 55000
WHERE ECODE = 1009 ;
```

UPDATING MORE THAN ONE COLUMNS

e.g. to update the salary to 58000 and grade to B2 for those employee whose employee code is 1001.

```
UPDATE  EMPLOYEE
SET GROSS = 58000, GRADE='B2'
WHERE ECODE = 1009 ;
```

OTHER EXAMPLES

e.g.1. Increase the salary of each employee by 1000 in the EMPLOYEE table.

```
UPDATE EMPLOYEE
SET GROSS = GROSS +100 ;
```

e.g.2. Double the salary of employees having grade as 'A1' or 'A2' .

```
UPDATE  EMPLOYEE
SET GROSS = GROSS * 2 ;
WHERE GRADE='A1' OR  GRADE='A2' ;
```

e.g.3. Change the grade to 'A2' for those employees whose employee code is 1004 and name is Neela.

```
UPDATE  EMPLOYEE
SET GRADE='A2'
WHERE ECODE=1004 AND GRADE='NEELA' ;
```

DELETING DATA FROM TABLES

To delete some data from tables, DELETE command is used. **The DELETE command removes rows from a table.** The syntax of DELETE command is :

```
DELETE FROM <tablename>
WHERE <condition> ;
```

For example, to remove the details of those employee from EMPLOYEE table whose grade is A1.

```
DELETE FROM EMPLOYEE
WHERE GRADE ='A1' ;
```

TO DELETE ALL THE CONTENTS FROM A TABLE

```
DELETE FROM EMPLOYEE ;
```

So if we do not specify any condition with WHERE clause, then all the rows of the table will be deleted. Thus above line will delete all rows from employee table.

DROPPING TABLES

The DROP TABLE command lets you drop a table from the database. The **syntax of DROP TABLE** command is :

```
DROP TABLE  <tablename> ;
```

e.g. to drop a table employee, we need to write :

```
DROP TABLE employee ;
```

Once this command is given, the table name is no longer recognized and no more commands can be given on that table. After this command is executed, all the data in the table along with table structure will be deleted.

S.NO.	DELETE COMMAND	DROP TABLE COMMAND
1	It is a DML command.	It is a DDL Command.
2	This command is used to delete only rows of data from a table	This command is used to delete all the data of the table along with the structure of the table. The table is no longer recognized when this command gets executed.
3	Syntax of DELETE command is: **DELETE FROM** <tablename> **WHERE** <condition> ;	Syntax of DROP command is : **DROP TABLE** <tablename> ;

EMPLOYEE

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000
1004	Neela	F	B2	38965
1005	Sunny	M	A2	30000
1006	Ruby	F	A1	45000
1009	Neema	F	A2	52000

SELECTING ALL DATA

- In order to retrieve everything (all columns) from a table, SELECT command is used as :

  SELECT * FROM <tablename> ;

  e.g.

  In order to retrieve everything from **Employee** table, we write SELECT command as :

  SELECT * FROM Employee ;

SELECTING PARTICULAR COLUMNS

- A particular column from a table can be selected by specifying column-names with SELECT command. E.g. in above table, if we want to select ECODE and ENAME column, then command is :

  SELECT ECODE , ENAME
  FROM EMPLOYEE ;

  E.g.2 in order to select only ENAME, GRADE and GROSS column, the command is :

  SELECT ENAME , GRADE , GROSS
  FROM EMPLOYEE ;

SELECTING PARTICULAR ROWS

We can select particular rows from a table by specifying a condition through **WHERE clause** along with SELECT statement. **E.g.** In employee table if we want to select rows where Gender is female, then command is :

SELECT * FROM **EMPLOYEE**
WHERE **GENDER = 'F' ;**

E.g.2. in order to select rows where salary is greater than 48000, then command is :

SELECT * FROM **EMPLOYEE**
WHERE **GROSS > 48000 ;**

RELATIONAL OPERATORS

- To compare two values , a relational operator is used. The result of the comparison is true or false. The SQL recognizes following relational operators:

  = < > <= >= < > (Not equal to).

LOGICAL OPERATORS

- The logical operator OR (||), AND (&&) and NOT (!) are used to connect search conditions in the WHERE clause.e.g.

1. To list the employee details having grade 'A1' or 'A2' from table employee, logical operator OR will be used as :

```
SELECT * FROM EMPLOYEE
WHERE GRADE='A1'  OR  GRADE ='A2' ;
```

2. To list the employee details having grades as A1 and salary greater than 40000, logical operator AND will be used as:

```
SELECT * FROM EMPLOYEE
WHERE GRADE='A1'  AND  GROSS > 40000;
```

3. To list the employee details whose grades are other than 'A1', logical operator NOT will be used as :

```
SELECT * FROM EMPLOYEE
WHERE NOT GRADE="A1" ;
```

ELIMINATING REDUNDANT DATA

The **DISTINCT** keyword eliminates duplicate rows from the results of a SELECT statement. For example ,

SELECT GENDER **FROM** EMPLOYEE ;

GENDER
M
M
F
M
F
F

SELECT DISTINCT(GENDER) **FROM** EMPLOYEE ;

DISTINCT(GENDER)
M
F

VIEWING STRUCTURE OF A TABLE

- If we want to know the structure of a table, we can use DESCRIBE or DESC command, as per following syntax :

  DESCRIBE | **DESC** <tablename> ;

  e.g. to view the structure of table **EMPLOYEE**, command is : **DESCRIBE** EMPLOYEE ; **OR DESC** EMPLOYEE ;

  In the output of this command, we get the column names present in the table along with their data types.

USING COLUMN ALIASES

- The columns that we select in a query can be given a different name, i.e. column alias name for output purpose.

Syntax :

SELECT <columnname> **AS** column alias , <columnname> **AS** column alias …..
FROM <tablename> ;

e.g. In output, suppose we want to display ECODE column as EMPLOYEE_CODE in output , then command is :

SELECT **ECODE AS "EMPLOYEE_CODE"**
FROM **EMPLOYEE ;**

CONDITION BASED ON A RANGE

- The **BETWEEN** operator defines a range of values that the column values must fall in to make the condition true. The range include both lower value and upper value.

e.g. to display ECODE, ENAME and GRADE of those employees whose salary is between 40000 and 50000, command is:

SELECT ECODE , ENAME ,GRADE
FROM **EMPLOYEE**
WHERE GROSS **BETWEEN** 40000 AND 50000 ;

Output will be :

ECODE	ENAME	GRADE
1001	Ravi	E4
1006	Ruby	A1

CONDITION BASED ON A LIST

- To specify a list of values, IN operator is used. The IN operator selects value that match any value in a given list of values. E.g.

SELECT * FROM EMPLOYEE
WHERE GRADE **IN** ('A1' , 'A2');

Output will be :

ECODE	**ENAME**	**GENDER**	**GRADE**	**GROSS**
1002	Akash	M	A1	35000
1006	Ruby	F	A1	45000
1005	Sunny	M	A2	30000
1009	Neema	F	A2	52000

- The **NOT IN** operator finds rows that do not match in the list. E.g.

SELECT * FROM EMPLOYEE
WHERE GRADE **NOT IN** ('A1' , 'A2');

Output will be :

ECODE	**ENAME**	**GENDER**	**GRADE**	**GROSS**
1001	Ravi	M	E4	50000
1004	Neela	F	B2	38965

CONDITION BASED ON PATTERN MATCHES

- LIKE operator is used for pattern matching in SQL. Patterns are described using two special wildcard characters:

1. percent(%) – The % character matches any substring.
2. underscore(_) – The _ character matches any character.

e.g. to display names of employee whose name starts with R in EMPLOYEE table, the command is :

```
SELECT ENAME
FROM EMPLOYEE
WHERE ENAME LIKE 'R%' ;
```

Output will be :

ENAME
Ravi
Ruby

e.g. to display details of employee whose second character in name is 'e'.

```
SELECT *
FROM EMPLOYEE
WHERE ENAME LIKE '_e%' ;
```

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1004	Neela	F	B2	38965
1009	Neema	F	A2	52000

e.g. to display details of employee whose name ends with 'y'.

```
SELECT *
FROM EMPLOYEE
WHERE ENAME LIKE '%y' ;
```

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1005	Sunny	M	A2	30000
1006	Ruby	F	A1	45000

SEARCHING FOR NULL

- The NULL value in a column can be searched for in a table using IS NULL in the WHERE clause. E.g. to list employee details whose salary contain NULL, we use the command :

```
SELECT *
FROM EMPLOYEE
WHERE GROSS IS NULL ;
```

e.g.

STUDENT

Roll_No	Name	Marks
1	ARUN	NULL
2	RAVI	56
4	SANJAY	NULL

to display the names of those students whose marks is NULL, we use the command :

```
SELECT Name
FROM STUDENT
WHERE Marks IS NULL ;
```

Output will be :

Name
ARUN
SANJAY

SORTING RESULTS

Whenever the SELECT query is executed , the resulting rows appear in a predecided order.The **ORDER BY clause** allow sorting of query result. The sorting can be done either in ascending or descending order, the default is ascending.

The **ORDER BY clause is used as :**

SELECT <column name> , <column name>….
FROM <tablename>
WHERE <condition>
ORDER BY <column name> ;

e.g. to display the details of employees in EMPLOYEE table in alphabetical order, we use command :

SELECT *
FROM EMPLOYEE
ORDER BY ENAME ;

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1002	Akash	M	A1	35000
1004	Neela	F	B2	38965
1009	Neema	F	A2	52000
1001	Ravi	M	E4	50000
1006	Ruby	F	A1	45000
1005	Sunny	M	A2	30000

e.g. display list of employee in descending alphabetical order whose salary is greater than 40000.

SELECT ENAME
FROM EMPLOYEE
WHERE GROSS > 40000
ORDER BY ENAME desc ;

Output will be :

ENAME
Ravi
Ruby
Neema

HOW TO PERFORM SIMPLE CALCULATIONS ?

- In MYSQL simple calculations like 4*3 , 5+6 , etc. can also be performed , and the result of such calculations are displayed in a special table called DUAL.
- DUAL is a small dummy table, which has just one row and one column. It is used for obtaining calculation results.

e.g. SELECT 4 * 3 FROM **DUAL** ;

Output will be :

4*3
12

SCALAR EXPRESSIONS WITH FIELDS

- If we want to perform simple numeric computations on the data to put it in a form more appropriate to your needs, SQL allows us to place scalar expressions and constants among selected fields(columns).
  e.g. if in the output , we want to see only 10% of each employee salary, then we write command as:

  SELECT GROSS * 0.10
  FROM **EMPLOYEE** ;

  Output will be :

GROSS * 0.10
5000
3500
3896.5
3000
4500
5200

Note:** The scalar expression specified in above query (GROSS * 0.10) is not going to change actual value in table EMPLOYEE.

QUESTIONS ON TABLE CREATION

Q1. Write an SQL query to create the table 'TEAMS' with the following structure:

Field	Type
TeamCode	Varchar(5)
TeamName	Varchar(5)
TeamLeader	Varchar(5)
NoofMembers	Integer
Team_Symbol	Char(1)

Q2. Write MySQL command to create the table "Toyz" with the following structure :

Column_name	DataType(size)
Toy_no	Int (10)
Toy_name	Varchar(20)
Type	Char(10)
Price	Decimal (8,2)
Colour	Varchar(15)

Q3. Write a MySQL command for creating a table "BANK" whose structure is given below:

Field_Name	DataType	Size
Acct_number	Integer	4
Name	Varchar	3
BirthDate	Date	
Balance	Integer	8

1 MARK QUESTIONS

Q1. Write SQL command to view the structure of EMP table.

Q2. A table 'Customers' in a database has 5 columns and no rows in it. What is its cardinality ? What will be its cardinality if 4 rows are added in the table?

Q3. Sarthya, a student of class XI, created a table "RESULT". Grade is one of the columns of this table. To find the details of students whose Grade have not been entered, he wrote the following MySql query, which did not give the desired result :

```
SELECT * FROM result WHERE grade = NULL;
```

Q4. What is the purpose of DROP TABLE command in MySQL? How is it different from DELETE command?

Q5. Saumya had previously created a table named 'Product' in a database using MySQL. Later on She forgot the table structure. Suggest to her the suitable MySQL command through which she can check the structure of the already created table.

Q6. A table "customer" in a database has 5 columns and no row in it. What is its cardinality? What will be its cardinality if 4 rows are added in the table.

Q7. Observe the table 'Club' given below:

CLUB

Member_id	Member_name	Address	Age	Fees
M001	Sumit	Manipur	20	1000
M002	Nisha	Kolkata	19	2100
M003	Niharika	Nagaland	17	1200

(i) What is the cardinality and degree of the above given table?

(ii) If a new column contact_no has been added and two more members have joined the club then how these changes will affect the degree and cardinality of the given table.

Q8. Write MySql command that will be used to open an already existing database "CONTACTS";

Q9. The Doc_name column of a table Hospital is given below:

Doc_name
Avinash
Hariharan
Vinayak
Deepak
Sanjeev

Based on the information, find the output of the following queries :

(i) Select doc_name from Hospital where doc_name like "%v";

(ii) Select doc_name from Hospital where doc_name like "%e%";

Q10. How is NULL value different from 0(zero) value ?

Q11. Write the UPDATE statement in MySQL to increase commission by 100.00 in the "Commission" column in the "EMP" table.

Q12. When using the LIKE clause, which wildcard symbol represents any sequence of none, one or more characters?

Q13. Rewrite the following SQL statement after correcting error(s). Underline the corrections made.

```
INSERT IN STUDENT(RNO, MARKS) VALUE (5 , 78.5) ;
```

Q14. Write MySQL command to display the list of existing databases.

Q15. Write one similarity and one difference between CHAR and VARCHAR data type.

Q16. What is MySQL ?

Q17. Write two features of MySQL.

Q18. Rows are called _____ in MySQL.

QUESTION BASED ON GIVEN TABLES

Q1. Consider the Table SHOPPE given below. Write command in MySql for (i) to (iv) and output for (v) to (vii).

Table : SHOPPE

Code	Item	Company	Qty	City	Price
102	Biscuit	Hide & Seek	100	Delhi	10.00
103	Jam	Kissan	110	Kolkata	25.00
101	Coffee	Nestle	200	Kolkata	55.00
106	Sauce	Maggi	56	Mumbai	55.00
107	Cake	Britannia	72	Delhi	10.00
104	Maggi	Nestle	150	Mumbai	10.00
105	Chocolate	Cadbury	170	Delhi	25.00

(i) To display names of the items, whose name starts with 'C' in ascending order of Price.
(ii) To display Code, Item name and city of the products whose quantity is less than 100.
(iii) To display distinct Company from the table.
(iv) To insert a new row in the table Shoppe.
110 , 'Pizza' , 'Papa Jones' , 120, 'Kolkata', 50.0
(v) Select Item from Shoppe where Item IN("Jam", "Coffee");
(vi) Select distinct(city) from Shoppe;
(vii) Select Qty from Shoppe where City="Mumbai" ;

Q2. Consider the Table RESULT given below. Write command in MySql for (i) to (iv) and output for (v) to (vii).

Table : RESULT

No	Name	Stipend	Subject	Average	Division
1	Sharon	400	English	38	THIRD
2	Amal	680	Mathematics	72	FIRST
3	Vedant	500	Accounts	67	FIRST
4	Shakeer	200	Informatics	55	SECOND
5	Anandha	400	History	85	FIRST
6	Upasna	550	Geography	45	THIRD

(i) To list the names of those students, who have obtained Division as FIRST in the ascending order of NAME.
(ii) To display a report listing NAME, SUBJECT and Annual Stipend received assuming that the stipend column has monthly stipend.
(iii) To display the details of students, who have either Accounts or Informatics as Subject.
(iv) To insert a new row in the table EXAM :
6 , "Mohan" , 500 , "English" , 73 , "Second"
(v) SELECT Stipend FROM EXAM WHERE Average BETWEEN 60 AND 75 ;
(vi) SELECT DISTINCT Subject FROM EXAM ;
(vii) SELECT Average FROM EXAM WHERE NAME NOT LIKE 'S%' ;
(viii) SELECT NAME FROM EXAM WHERE Division ="THIRD" ;

Q3. Consider the table Projects given below. Write commands in SQL for (i) to (iv) and output for (v) to (viii)

Table : Project

ID	ProjName	ProjSize	StartDate	EndDate	Cost
1	Payroll-MM	Medium	2006-03-17	2006-09-16	60000
2	Payroll-ITC	Large	2008-02-12	2008-01-11	500000
3	IDMgmt-LITE	Large	2008-06-13	2009-05-21	300000
4	Recruit-LITE	Medium	2008-03-18	2008-06-01	50000
5	IDMgmt-MTC	Small	2007-01-15	2007-01-29	20000
6	Recruit-ITC	Medium	2007-03-01	2007-06-28	50000

(i) To display all information about projects of Medium ProjSize.

(ii) To list the Projsize of projects whose ProjName ends with LITE.

(iii) To list ID, Name, and cost of all the projects in descending order of StartDate.

(iv) To display the name of Projects of cost less than 100000.

(v) SELECT ProjName FROM Projects WHERE Cost NOT BETWEEN 50000 AND 60000;

(vi) SELECT Distinct(ProjSize) FROM Projects;

(vii) SELECT * FROM Projects WHERE Cost < 100000 ;

DIFFERENCE BETWEEN TERMS

(i) DELETE Command and DROP TABLE Command

(ii) CHAR and VARCHAR datatype.

LONG ANSWER TYPE QUESTIONS

Q1. What is SQL? What are different categories of commands available in SQL ?

Q2. Difference between DDL and DML command.

Q3. Define the following terms :

(i) Primary Key

(ii) Candidate Key

(iii) Alternate Key

(iv) Foreign Key

(v) Referential Integrity

Q4. What is DBMS ? What are the advantages of DBMS ?

Q5. What are views? Explain with help of example. How are they useful ?

INTEGRITY CONSTRAINTS/CONSTRAINTS

- A constraint is a condition or check applicable on a field(column) or set of fields(columns).
- Common types of constraints include :

S.No.	Constraints	Description
1	NOT NULL	Ensures that a column cannot have NULL value
2	DEFAULT	Provides a default value for a column when none is specified
3	UNIQUE	Ensures that all values in a column are different
4	CHECK	Makes sure that all values in a column satisfy certain criteria
5	PRIMARY KEY	Used to uniquely identify a row in the table
6	FOREIGN KEY	Used to ensure referential integrity of the data

NOT NULL CONSTRAINT

By default, a column can hold NULL. If you not want to allow NULL value in a column, then NOT NULL constraint must be applied on that column. E.g.

```
CREATE TABLE Customer
(       SID integer NOT NULL ,
        Last_Name varchar(30) NOT NULL ,
        First_Name varchar(30)    ) ;
```

Columns **SID** and **Last_Name** cannot include NULL, while **First_Name** can include NULL.

An attempt to execute the following SQL statement,

```
INSERT INTO Customer
VALUES (NULL , 'Kumar' , 'Ajay');
```

will result in an error because this will lead to column SID being NULL, which violates the NOT NULL constraint on that column.

DEFAULT CONSTARINT

The DEFAULT constraint provides a default value to a column when the INSERT INTO statement does not provide a specific value. **E.g.**

```
CREATE TABLE Student
(  Student_ID integer ,
   Name varchar(30) ,
   Score integer DEFAULT 80);
```

When following SQL statement is executed on table created above:

```
INSERT INTO Student
VALUES (10 , 'Ravi' );
```

no value has been provided for score field.

Then table **Student** looks like the following:

Student_ID	Name	Score
10	Ravi	**80**

score field has got the default value

UNIQUE CONSTRAINT

- The UNIQUE constraint ensures that all values in a column are distinct. In other words, no two rows can hold the same value for a column with UNIQUE constraint.

e.g.

```
CREATE TABLE Customer
(       SID integer Unique ,
        Last_Name varchar(30) ,
        First_Name varchar(30) ) ;
```

Column SID has a unique constraint, and hence cannot include duplicate values. So, if the table already contains the following rows :

SID	Last_Name	First_Name
1	Kumar	Ravi
2	Sharma	Ajay
3	Devi	Raj

The executing the following SQL statement,

```
INSERT INTO Customer
VALUES (3 , 'Cyrus' , 'Grace') ;
```

will result in an error because the value 3 already exist in the SID column, thus trying to insert another row with that value violates the UNIQUE constraint.

CHECK CONSTRAINT

- The CHECK constraint ensures that all values in a column satisfy certain conditions. Once defined, the table will only insert a new row or update an existing row if the new value satisfies the CHECK constraint.

e.g.

```
CREATE TABLE Customer
(       SID integer CHECK (SID > 0),
        Last_Name varchar(30) ,
        First_Name varchar(30) ) ;
```

So, attempting to execute the following statement :

```
INSERT INTO Customer
VALUES (-2 , 'Kapoor' , 'Raj');
```

will result in an error because the values for SID must be greater than 0.

PRIMARY KEY CONSTRAINT

- A primary key is used to uniquely identify each row in a table. A primary key can consist of one or more fields(column) on a table. When multiple fields are used as a primary key, they are called a **composite key.**

- You can define a primary key in CREATE TABLE command through keywords PRIMARY KEY. e.g.

```
CREATE TABLE Customer
(       SID integer PRIMARY KEY,
        Last_Name varchar(30) ,
        First_Name varchar(30) ) ;
```

Or

```
CREATE TABLE Customer
(       SID integer,
        Last_Name varchar(30) ,
        First_Name varchar(30),
        PRIMARY KEY (SID) ) ;
```

- The latter way is useful if you want to specify a composite primary key, **e.g.**

```
CREATE TABLE Customer
(       Branch integer NOT NULL,
        SID integer NOT NULL ,
        Last_Name varchar(30) ,
        First_Name varchar(30),
        PRIMARY KEY (Branch , SID) ) ;
```

FOREIGN KEY CONSTRAINT

- Foreign key is a non key column of a table (**child table**) that draws its values from **primary key** of another table(**parent table).**
- The table in which a foreign key is defined is called a **referencing table or child table.** A table to which a foreign key points is called **referenced table or parent table.**

e.g.

TABLE: STUDENT ← **Parent Table**

ROLL_NO ← **Primary key**	NAME	CLASS
1	ABC	XI
2	DEF	XII
3	XYZ	XI

Child Table

TABLE: SCORE

ROLL_NO	MARKS
1	55
2	83
3	90

Here column ***Roll_No is a foreign key in table SCORE(Child Table)*** and it is deriving its values from Primary key (ROLL_NO) of STUDENT table.(Parent Key).

```
CREATE TABLE STUDENT
(   ROLL_NO integer NOT NULL PRIMARY KEY ,
    NAME VARCHAR(30) ,
    CLASS VARCHAR(3) );
```

```
CREATE TABLE SCORE
(   ROLL_NO integer ,
    MARKS integer ,
    FOREIGN KEY(ROLL_NO) REFERNCES STUDENT(ROLL_NO) ) ;
```

**** Foreign key is always defined in the child table.***

Syntax for using foreign key

FOREIGN KEY (column name) REFERENCES Parent_Table (PK of Parent Table);

REFERENCING ACTIONS

Referencing action with ON DELETE clause determines what to do in case of a DELETE occur in the parent table.
Referencing action with ON UPDATE clause determines what to do in case of a UPDATE occur in the parent table.

Actions:

1. **CASCADE :** This action states that if a DELETE or UPDATE operation affects a row from the parent table, then automatically delete or update the matching rows in the child table i.e., cascade the action to child table.
2. **SET NULL :** This action states that if a DELETE or UPDATE operation affects a row from the parent table, then set the foreign key column in the child table to NULL.
3. **NO ACTION :** Any attempt for DELETE or UPDATE in parent table is not allowed.
4. **RESTRICT :** This action rejects the DELETE or UPDATE operation for the parent table.

Q: Create two tables

Customer(customer_id, name)
Customer_sales(transaction_id, amount , **customer_id**)

Underlined columns indicate primary keys and bold column names indicate foreign key.
Make sure that no action should take place in case of a DELETE or UPDATE in the parent table.

Sol :
```
CREATE TABLE Customer (
    customer_id  int Not Null Primary Key ,
    name varchar(30) ) ;

CREATE TABLE Customer_sales (
    transaction_id Not Null Primary Key ,
    amount int ,
    customer_id int ,
    FOREIGN KEY(customer_id) REFERENCES Customer (customer_id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION   );
```

Q: Distinguish between a Primary Key and a Unique key in a table.

S.NO.	PRIMARY KEY	UNIQUE KEY
1.	Column having Primary key can not contain NULL value	Column having Unique Key can contain NULL value
2.	There can be only one primary key in Table.	Many columns can be defined as Unique key

ALTER TABLE COMMAND

The ALTER TABLE command is used to change definitions of existing tables.(adding columns,deleting columns etc.). The ALTER TABLE command is used for :

1. adding columns to a table
2. Modifying column-definitions of a table.
3. Deleting columns of a table.
4. Adding constraints to table.
5. Enabling/Disabling constraints.

ADDING COLUMNS TO TABLE

To add a column to a table, ALTER TABLE command can be used as per following syntax:

ALTER TABLE <tablename>
ADD <Column name> <datatype> <constraint> ;

e.g. to add a new column ADDRESS to the EMPLOYEE table, we can write command as :

ALTER TABLE EMPLOYEE
ADD ADDRESS VARCHAR(50);

A new column by the name ADDRESS will be added to the table, where each row will contain NULL value for the new column.

ECODE	ENAME	GENDER	GRADE	GROSS	ADDRESS
1001	Ravi	M	E4	50000	NULL
1002	Akash	M	A1	35000	NULL
1004	Neela	F	B2	38965	NULL
1005	Sunny	M	A2	30000	NULL
1006	Ruby	F	A1	45000	NULL
1009	Neema	F	A2	52000	NULL

However **if you specify NOT NULL constraint while adding a new column**, MySQL adds the new column with the default value of that datatype e.g. for INT type it will add 0 , for CHAR types, it will add a space, and so on.

e.g. Given a table namely Testt with the following data in it.

Col1	Col2
1	A
2	G

Now following commands are given for the table. Predict the table contents after each of the following statements:

(i) ALTER TABLE testt ADD col3 INT ;
(ii) ALTER TABLE testt ADD col4 INT NOT NULL ;
(iii) ALTER TABLE testt ADD col5 CHAR(3) NOT NULL ;
(iv) ALTER TABLE testt ADD col6 VARCHAR(3);

MODIFYING COLUMNS

Column name and data type of column can be changed as per following syntax :

ALTER TABLE <table name>
CHANGE <old column name> <new column name> <new datatype>;

If **Only data type of column need to be changed**, then

ALTER TABLE <table name>
MODIFY <column name> <new datatype>;

e.g.1. In table EMPLOYEE, change the column GROSS to SALARY.

ALTER TABLE EMPLOYEE
CHANGE GROSS SALARY INTEGER;

e.g.2. In table EMPLOYEE , change the column ENAME to EM_NAME and data type from VARCHAR(20) to VARCHAR(30).

ALTER TABLE EMPLOYEE
CHANGE ENAME EM_NAME VARCHAR(30);

e.g.3. In table EMPLOYEE , change the datatype of GRADE column from CHAR(2) to VARCHAR(2).

ALTER TABLE EMPLOYEE
MODIFY GRADE VARCHAR(2);

DELETING COLUMNS

To delete a column from a table, the ALTER TABLE command takes the following form :

ALTER TABLE <table name>
DROP <column name>;

e.g. to delete column GRADE from table EMPLOYEE, we will write :

ALTER TABLE EMPLOYEE
DROP GRADE ;

ADDING/REMOVING CONSTRAINTS TO A TABLE

ALTER TABLE statement can be used to add constraints to your existing table by using it in following manner:

- **TO ADD PRIMARY KEY CONSTRAINT**

  ALTER TABLE <table name>
  ADD PRIMARY KEY (Column name);

e.g. to add PRIMARY KEY constraint on column ECODE of table EMPLOYEE , the command is :

ALTER TABLE EMPLOYEE
ADD PRIMARY KEY (ECODE) ;

- **TO ADD FOREIGN KEY CONSTRAINT**

  ALTER TABLE <table name>
  ADD FOREIGN KEY (Column name) REFERENCES Parent Table (Primary key of Parent Table);

REMOVING CONSTRAINTS

- To remove primary key constraint from a table, we use ALTER TABLE command as :

  ALTER TABLE <table name>
  DROP PRIMARY KEY ;

- To remove foreign key constraint from a table, we use ALTER TABLE command as :

  ALTER TABLE <table name>
  DROP FOREIGN KEY ;

MySQL FUNCTIONS

A function is a special type of predefined command set that performs some operation and returns a single value.

Types of MySQL functions : String Functions , Maths Functions and Date & Time Functions.

Table : EMPL

EMPNO	ENAME	JOB	SAL	DEPTNO
8369	SMITH	CLERK	2985	10
8499	ANYA	SALESMAN	9870	20
8566	AMIR	SALESMAN	8760	30
8698	BINA	MANAGER	5643	20
8912	SUR	NULL	3000	10

STRING FUNCTIONS

1. CONCAT( ) - Returns the Concatenated/joined String.

Syntax : CONCAT(Column1 , Column2 , Column3, …….)

e.g. SELECT CONCAT(EMPNO , ENAME) FROM EMPL WHERE DEPTNO=10;

Output

CONCAT(EMPNO , ENAME)
8369SMITH
8912SUR

2. LOWER( ) / LCASE( ) - Returns the argument in lowercase.

Syntax : LOWER(Column name)

e.g.

SELECT LOWER(ENAME) FROM EMPL ;

Output

LOWER(ENAME)
smith
anya
amir
bina
sur

3. UPPER( ) / UCASE( ) - Returns the argument in uppercase.

Syntax : UPPER(Column name)

e.g.

SELECT UPPER(ENAME) FROM EMPL ;

Output

UPPER(ENAME)
SMITH
ANYA
AMIR
BINA
SUR

4. SUBSTRING( ) / SUBSTR( ) – Returns the substring as specified.

Syntax : SUBSTR(Column name, m , n), where **m specifies starting index** and **n specifies number of characters from the starting index m.**

e.g.

SELECT SUBSTR(ENAME,2,2) FROM EMPL WHERE DEPTNO=20;

Output

SUBSTR(ENAME,2,2)
NY
IN

SELECT SUBSTR(JOB,-4,2) FROM EMPL WHERE DEPTNO=20;

Output

SUBSTR(JOB,-4,2)
SM
AG

5. LTRIM( ) – Removes leading spaces.

e.g. SELECT LTRIM(' RDBMS MySQL') ;

Output

LTRIM(' RDBMS MySQL')
RDBMS MySQL

6. RTRIM( ) – Removes trailing spaces.

e.g. SELECT RTRIM('RDBMS MySQL ') ;

Output

RTRIM('RDBMS MySQL ')
RDBMS MySQL

7. TRIM( ) – Removes trailing and leading spaces.

e.g. SELECT TRIM(' RDBMS MySQL ') ;

Output

TRIM(' RDBMS MySQL ')
RDBMS MySQL

8. LENGTH( ) – Returns the length of a string. e.g.

SELECT LENGTH("CANDID") ;

Output

LENGTH("CANDID")
6

e.g.2.

SELECT LENGTH(ENAME) FROM EMPL;

Output

LENGTH(ENAME)
5
4
4
4
3

9. LEFT( ) – Returns the leftmost number of characters as specified.

e.g. SELECT LEFT('CORPORATE FLOOR' , 3) ;

Output

LEFT('CORPORATE FLOOR', 3)
COR

10. RIGHT( ) – Returns the rightmost number of characters as specified.

e.g. SELECT RIGHT('CORPORATE FLOOR' , 3) ;

Output

RIGHT('CORPORATE FLOOR', 3)
OOR

11. MID( ) – This function is same as SUBSTRING( ) / SUBSTR( ) function. E.g.

SELECT MID("ABCDEF" , 2 , 4) ;

Output

MID("ABCDEF" , 2 , 4)
BCDE

NUMERIC FUNCTIONS

These functions accept numeric values and after performing the operation, return numeric value.

1. MOD( ) – Returns the remainder of given two numbers. e.g. SELECT MOD(11 , 4) ;

Output

MOD(11, 4)
3

2. POW( ) / POWER( ) - This function returns m^n i.e , a number m raised to the n^{th} power.

e.g. SELECT POWER(3,2) ;

Output

POWER(3, 2)
9

3. ROUND( ) – This function returns a number rounded off as per given specifications.

e.g. ROUND(15.193 , 1) ;

Output

ROUND(15.193 , 1)
15.2

e.g. 2. SELECT ROUND(15.193 , -1); - This will convert the number to nearest ten's .

Output

ROUND(15.193 , -1)
20

4. SIGN( ) – This function returns sign of a given number.

If number is negative, the function returns -1.

If number is positive, the function returns 1.

If number is zero, the function returns 0.

e.g. SELECT SIGN(-15) ;

Output

SIGN(-15)
-1

e.g.2 SELECT SIGN(20) ;

Output

SIGN(20)
1

5. SQRT( ) – This function returns the square root of a given number. E.g.

SELECT SQRT(25) ;

Output

SQRT(25)
5

6. TRUNCATE( ) – This function returns a number with some digits truncated. E.g.

SELECT TRUNCATE(15.79 , 1) ;

Output

TRUNCATE(15.79 , 1)
15.7

E.g. 2. SELECT TRUNCATE(15.79 , -1); - This command truncate value 15.79 to nearest ten's place.

Output

TRUNCATE(15.79 , -1)
10

DATE AND TIME FUNCTIONS

Date functions operate on values of the DATE datatype.

1. CURDATE( ) / CURRENT_DATE( ) – This function returns the current date. E.g.

SELECT CURDATE( ) ;

Output

CURDATE( )
2016-12-13

2. DATE( ) – This function extracts the date part from a date. E.g.

SELECT DATE('2016-02-09') ;

Output

DATE('2016-02-09')
09

3. MONTH( ) – This function returns the month from the date passed. E.g.

SELECT MONTH('2016-02-09') ;

Output

MONTH('2016-02-09')
02

4. YEAR( ) – This function returns the year part of a date. E.g.

SELECT YEAR('2016-02-09') ;

Output

YEAR('2016-02-09')
2016

5. DAYNAME( ) – This function returns the name of weekday. E.g.

SELECT DAYNAME('2016-02-09') ;

Output

DAYNAME('2016-12-14')
Wednesday

6. DAYOFMONTH( ) – This function returns the day of month. Returns value in range of 1 to 31.E.g.

SELECT DAYOFMONTH('2016-02-09') ;

Output

DAYOFMONTH('2016-02-09')
09

7. DAYOFWEEK( ) – This function returns the day of week. Return the weekday index for date. (1=Sunday, 2=Monday,……., 7=Saturday)

SELECT DAYOFWEEK('2016-12-14') ;

Output

DAYOFWEEK('2016-12-14')
4

8. DAYOFYEAR( ) – This function returns the day of the year. Returns the value between 1 and 366. E.g.

SELECT DAYOFYEAR('2016-02-04) ;

Output

DAYOFYEAR('2016-02-04')
35

9. NOW( ) – This function returns the current date and time.

It returns a constant time that indicates the time at which the statement began to execute.

e.g. SELECT NOW( );

10. SYSDATE( ) – It also returns the current date but it return the time at which SYSDATE( ) executes. It differs from the behavior for NOW( ), which returns a constant time that indicates the time at which the statement began to execute.

e.g. SELECT SYSDATE( ) ;

AGGREGATE / GROUP FUNCTIONS

Aggregate / Group functions work upon groups of rows , rather than on single row, and return one single output.
Different aggregate functions are : COUNT( ) , AVG( ) , MIN( ) , MAX( ) , SUM ( )

Table : EMPL

EMPNO	ENAME	JOB	SAL	DEPTNO
8369	SMITH	CLERK	2985	10
8499	ANYA	SALESMAN	9870	20
8566	AMIR	SALESMAN	8760	30
8698	BINA	MANAGER	5643	20
8912	SUR	NULL	3000	10

1. **AVG( )**

This function computes the average of given data.

e.g. SELECT AVG(SAL)
FROM EMPL ;

Output

AVG(SAL)
6051.6

2. **COUNT( )**

This function counts the number of rows in a given column.

If you specify the COLUMN name in parenthesis of function, then this function returns rows where COLUMN is not null.

If you specify the asterisk (*), this function returns all rows, including duplicates and nulls.

e.g. SELECT COUNT(*)
FROM EMPL ;

Output

COUNT(*)
5

e.g.2 SELECT COUNT(JOB)
FROM EMPL ;

Output

COUNT(JOB)
4

3. **MAX( )**

This function returns the maximum value from a given column or expression.

e.g. SELECT MAX(SAL)
FROM EMPL ;

Output

MAX(SAL)
9870

4. **MIN( )**

This function returns the minimum value from a given column or expression.

e.g. SELECT MIN(SAL)
FROM EMPL ;

Output

MIN(SAL)
2985

5. **SUM( )**

This function returns the sum of values in given column or expression.

e.g. SELECT SUM(SAL)
FROM EMPL ;

Output

SUM(SAL)
30258

Q1. Write an SQL query to create the table 'TEAMS' with the following structure:

Field	Type	Constraints
TeamCode	Varchar(5)	Primary Key
TeamName	Varchar(5)	
TeamLeader	Varchar(5)	
NoofMembers	Integer	
Team_Symbol	Char(1)	Not Null

Q2. Write MySQL command to create the table "Toyz" with the following structure and constraint :

Column_name	DataType(size)	Constraints
Toy_no	Int (10)	Primary Key
Toy_name	Varchar(20)	Not Null
Type	Char(10)	
Price	Decimal (8,2)	Price should be greater than 0.
Colour	Varchar(15)	

Q3. Write a MySQL command for creating a table "BANK" whose structure is given below:

Field_Name	DataType	Size	Constarint
Acct_number	Integer	4	Not Null
Name	Varchar	3	
BirthDate	Date		
Balance	Integer	8	Default value should be 0.

Q4. Write an SQL query to create the table 'Item' with the following structure :

Field	Type	Constraint
ItemCode	Varchar(20)	Primary Key
ItemName	Varchar(20)	
Category	Varchar(20)	
Price	Decimal(5,2)	Not Null

Q5. Write an SQL query to create the table 'SALESPERSON' with the following structure :

Table : SALESPERSON

Field	Type	Size	Constraint
SCODE	Decimal	6	Primary Key
FIRSTNAME	Varchar	30	Not Null
LASTNAME	Varchar	30	Not Null
CITY	Varchar	30	
SALES	Decimal	8	

Q6. Write a MySQL command for creating a table "PAYMENT" whose structure is given below:

Table : PAYMENT

Field	DataType	Size	Constraint
Loan_number	Integer	6	Primary Key

Payment_number	Varchar	30	Not Null
Payment_date	Date	30	
Payment_amount	Integer	30	Not Null

CHAPTER 15- IT APPLICATIONS

E-GOVERNANCE

E-governance refers to the application of electronic means in governance with an aim of fulfilling the requirements of common man at affordable costs and in fastest possible time.

OBJECTIVES OF E-GOVERNANCE

- To provide citizens access to information and knowledge about the political process, about services and about choices available.
- To satisfactorily fulfill the public's needs and expectations on the front office side, by simplifying their interaction with various online services.
- In the back-office side, to facilitate a speedy, transparent, accountable, efficient and effective process for performing government administration activities.

MAJOR E-GOVERNANCE PROJECTS IN INDIA

1. **Consular Passport and VISA Division (Indian Passport Office)**
   - The Consular Passport and VISA Division, a division of Ministry of External Affairs, is the Indian Passport office for people of India. It is responsible for issuing Indian Passports.

   (**URL**: http://passport.gov.in)
2. **Income Tax Portal**
   - Income tax portal include: preparation and filling individual Income Tax returns and TDS returns by tax deductore and filling and tracking of PAN/TAN applications.

   (**URL**: http://www.incometaxindia.gov.in)
3. **DRDO**
   - It is a network of more than 50 laboratories which are deeply engaged in developing defence technologies.

   (**URL**: drdo.nic.in)
4. **Supreme Court of India**
   - It is the highest judicial body in India. It has also its web-presence in the form of a website that can be used to know about Supreme Court Judgments.

   (**URL**: http://supremecourtofindia.nic.in)
5. **RTI Portal**

   Right to Information Act 2005 mandates timely response to citizen requests for government information.

   (**URL**: rti.gov.in)

BENEFITS OF E-GOVERNANCE TO COMMON MAN

1. The numbers of trips to government offices reduced by nearly 11% to 27%.
2. Waiting time is reduced significantly.
3. Bribes reduced by 50% to 90%.
4. All above factors resulted in reduced cost of availing the service.

POSITIVE IMPACTS OF E-GOVERNANCE

1. E-governance programs have improved the efficiency of administration and service delivery.
2. E-governance programs have resulted in reduced waiting time before the work is done.
3. People have also benefitted from e-governance in the form of reduced cost of availing the service.
4. E-governance has proved successful in keeping a tab on corruption to some extent.
5. E-governance programs have resulted in increased public participation.

NEGATIVE IMPACTS OF E-GOVERNANCE

1. People living in rural and remote areas could not benefit from the e-governance initiatives because of lack of computerization in these areas.
2. Lack of awareness about the e-governance programs has prevented people to benefit from it.
3. Not all services are part of e-governance, so manual methods cannot be avoided.
4. Incompatibility of software and hardware has prevented people to benefit from it.
5. Some people find it inconvenient to make payments online using credit cards.

E-BUSINESS

- Electronic Commerce (**EC or e-commerce**) describes the process of buying, selling, transferring, or exchange products, services and information via computer networks, including the Internet.
- E-Business refer to a broader definition of EC, not just the buying and selling of goods and services, but also servicing customers, collaborating with business partners, conducting e-learning, and conducting electronic transactions within an organization.

MAJOR E- BUSINESS PORTALS

1. **IRCTC Portal**
   - The **I**ndian **R**ailways **C**atering and **T**ourism **C**orporation (IRCTC) is a subsidiary of the Indian Railways.

   (**URL**: www.irctc.co.in)
2. **Online reservation site Yatra.com**
   - Yatra.com provides airline reservation, flight ticket booking service online.

   (**URL:** www.yatra.com)
3. **E- Banking Site of State Bank of India**
   - This portal serves corporate banking services, individual banking services, loan lending and many more.

   (**URL:** http://www.statebankofindia.com)
4. **Online store Amazon.com**
   - The amazon.com was launched online in 1995. It is an online store which sells wide variety of products.

   (**URL:** www.amazon.com)

BENEFITS OF E-BUSINESS TO CUSTOMERS

1. Improved speed of response
2. Cost savings
3. Improved efficiency and productivity
4. Improved customer service

BENEFITS OF E-BUSINESS TO SELLER

1. Offers opportunity to increase sales
2. Offers opportunity to access new markets across the globe

3. Provides convenience and comfort for customers.
4. Allow 24 x 7 access to the firm's products and services.

POSITIVE IMPACTS OF E-BUSINESS/E-COMMERCE

1. **INCREASE IN THE INTERNET USERS**
   - A significant segment of society is now using the internet for purchasing products online. Most of them are college going students and young persons.
2. **MIDDLE CLASS ATTRACTED TOWARDS LOW COST FLIGHTS**
   - With the help of certain e-business sites, people are easily finding low cost flights.

3. **CHANGE IN ONLINE SHOPPING HABITS**
   - Online offers are given at attractive discounts and prices. This convenience coupled with better bargains have brought changes in the online shopping habits of buyers.

4. **INCREASE IN ONLINE PAYMENTS**
   - Security of transactions online has been a major barrier to the growth of the E-business. However, with secure payment interface being provided by the websites, the Internet users are fast overcoming their apprehensions. According to major players, 65%-90% of their customers pay through credit cards.

NEGATIVE IMPACTS OF E-BUSINESS/ E-COMMERCE

1. **POOR TELECOM AND INFRASTRUCTURE FOR RELIABLE CONNECTIVITY**
   - Even after a telecom boom, Internet connectivity is still slow, access costs are high and connections are unreliable. All this has prevented users to rely on e-business.
2. **MULTIPLE ISSUES OF TRUST**
   - There have been cases where the online buyers have not received the goods matching up to their expectations.
   - Sometimes, the goods are faulty, or sometimes, the sizes or looks differ from what they perceived. This resulted into violation of their trust on e-businesses.

E-LEARNING

- It is a flexible term used to describe a means of teaching through technology such as a network, browser, CDROM or DVD multimedia platforms.

MAJOR E-LEANING SITES

1. **w3schools.com**
   - The w3schools.com hosts thousands of **online web tutorials** pertaining the domain of web building.

   (**URL** : www.w3schools.com)
2. **exe project**
   - It is developed as a freely available open source authoring application to **assist teachers and academics** in the publishing of web content without the need to become proficient in HTML or XML.

   (**URL** : exelearning.org)
3. **Xerte Project**
   - It is open source e-learning developer tool. The Xerte Project provides a **full suite of tools for e-learning developer and content authors.**

   (**URL**: www.nottingham.ac.uk/xerte)

BENEFITS OF E-LEARNING

1. **AVAILABLITY OF SAME COURSE TO MILLIONS**
   - As e-learning course are available without the boundaries of geography, same courses are available to students from all across the globe.
   - A student from remote interior of India can enroll herself to a high rank university of States without actually going there and enhance her skills.
   - This people from areas where the physical reach of education is limited are actually benefitting from it.
2. **BOON FOR WORKING CLASS**
   - The flexibility offered by e-learning in terms of time and learning pace has proved a boon for the working people.
   - Now thousands of working people are enhancing their skills at their time, at their own pace.

3. **SELF PACED LEARNING**
   - It's self-paced. Most e-learning programs can be taken when needed.
4. **SELF MOTIVATED INTERACTIVITY**
   - The self-motivation engage users, pushing them rather than pulling them through training.

NEGATIVE IMPACTS/DISADVANTAGES OF E-LEARNING

1. **HIGH DROPOUT RATE**
   - It has been observed that while e-learning has been popular with working class and retired people, a percentage of beginners often drop out the course midway. The reason for this are tat e-learning course lack interactivity and follow up.
2. **Technology issues** of the learners are most commonly technophobia and unavailability of required technologies.
3. **Portability** of materials has become a strength of e-learning, but still does not rival that of printed workbooks or reference material.
4. **Reduced social and cultural interaction** can be a drawback.
5. **Inappropriate content** for e-learning may exist according to some experts, though are limited in number.

BENEFITS OF e-LEARNING TO THE LEARNER

1. **ON DEMAND AVAILABILITY** – It can work from any location and any time. E-learners can go through training sessions from anywhere, usually at any time.
2. **SELF PACED LEARNING** – It's self-paced. Most e-learning programs can be taken when needed.
3. **SELF MOTIVATED INTERACTIVITY** – The self-motivation engages users, pushing them rather than pulling them through training.
4. **CONFIDENCE** – The fact that refreshers or quick reference materials are available.

PREVIOUS YEAR QUESTIONS

Q1. How does e-business improve customer satisfaction? Write one point.

Q2. How has our society benefited from e-governance?

Q3. How is e-learning beneficial to students? Write one point.

Q4. What benefits does an e-business offer to the customers?

Q5. How does e-business improve customer satisfaction? Write one point.

Q6. How has our society benefited from e-governance? Write 2 points.

****