Question bank

Database management system:

1)List any four advantage of DBMS over file processing system.

Reduction in Data redundancy

• Data consistency and integrity

• Data security

• Privacy

• Easy access of data

• Easy recovery • Flexibility

2)Enlist applications of database.

* Railway Reservation System The database is required to keep record of ticket booking, train departure, and arrival status.
* Library Management System There are thousands of books in the library so it is very difficult to keep record of all the books in a copy or register
* Banking
* Universities and colleges
* Credit card transactions
* Social Media Sites
* Telecommunications
* Finance

3)difference between DBMS vs RBMS.

|  |  |
| --- | --- |
| DBMS | RDBMS |
| DBMS stands for "Database Management System | RDBMS stands for "Relational Database Management System”. |
| DBMS technology stores the data in the form of files | RDBMS stores the data in the form of tables. |
| DBMS is designed to handle small amounts of data. | RDBMS is designed to deal with vast amount of data. |
| DBMS provides support only for a single user at a time | RDBMS provides support for multiple users at a time |

4)Define : Data abstraction and Data redundancy.

**Data Abstraction** : Many end users are not computer trained so it is needed to hide complex data structures from them. Hiding complexity of data structures from end user through different levels is known as data abstraction.

It has 3 levels :

a. Physical level

b. logical level

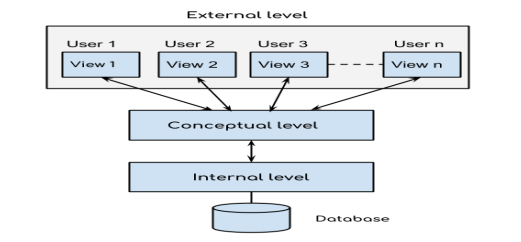
c. view level

**Data redundancy :** The repetition of information is known as redundancy .This redundancy leads to higher storage and access cost. It may lead to data inconsistency, that is different copies of the same data may have different values.

5)Define : Data Integrity.

Data integrity refers to the overall accuracy, completeness, and reliability of data. Data integrity is the validity of the data being entered in the database.

6)Draw three level architecture of dbms.



7)Define attribute. List the different types of attributes.

8)Explain ALTER command.Demonstrate wth any two option(add & modify).

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

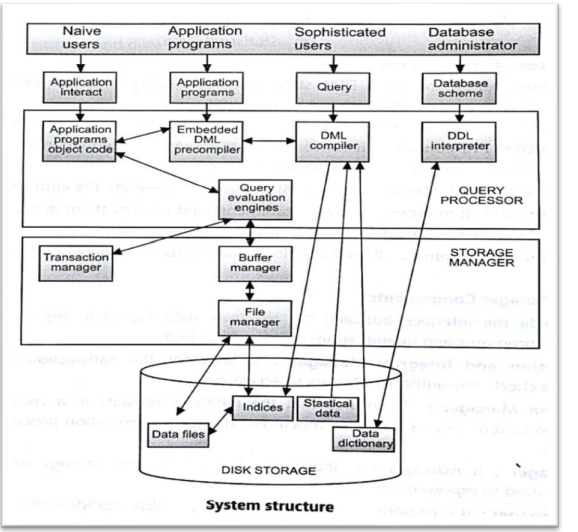
i) To add Columns in a table Syntax:ALTER TABLE table\_name ADD column\_name datatype; Example ALTER TABLE Customers ADD Email varchar2(20);

To modify a column in a table Syntax: ALTER TABLE table\_name MODIFY COLUMN column\_name datatype; Example ALTER TABLE Customers MODIFY COLUMN customeridnumeric(10);

9)Explain overall structure of dbms with the help of diagram.

Components of DBMS structure are classified in 3 categories as:

1. Query processor : Embedded DML pre compiler: It converts DML statements embedded in application. Program to normal procedural calls in host language. DML Compiler: It translates DML statements of high level language into low level instruction that a query evaluation engine understands. DDL interpreter: It interprets DDL statements and records them in a set of tables containing metadata. Query evaluation Engine: It executes low level instructions generated by DML compiler and issued by query processor to select efficient ways to execute query. DDL interpreter. It has following components,
2. Storage Manager Components : Transaction manager: It ensures that the database remains in consistent state despite of the system failure and that concurrent transaction execution proceeds without conflicting. File Manager: It manages the allocation of space on disk storage and data structures used to represent information stored on disk Buffer Manager: It is responsible for fetching data from disk storage into main memory and deciding what data to cache memory.
3. 3. Disk storage : Data files: It stores the database. Data Dictionary: It stores metadata that hold particular values. Indices: Provide fast access to data items that hold particular values.
4. Statistical data: It stores statistical information about the data in the database.



10) Define Instance and Schema of database.

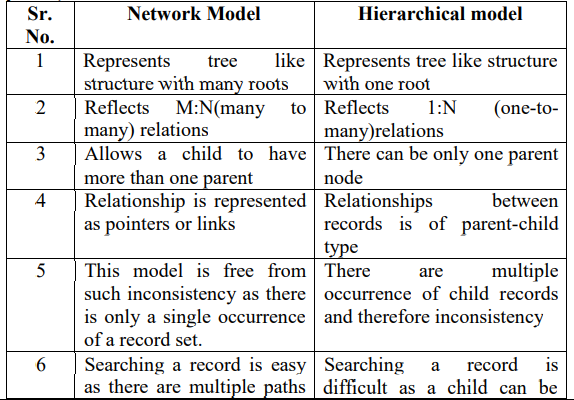
**Instance:** The data stored in database at a particular moment of time is called instance of database. **Schema:** Design of a database is called the schema. Schema is of three types: Physical schema, logical schema and view schema.

11)List various data models.

12)Describe basic concepts of relational model.

13) What is data model? Explain data network and hierarchical model.

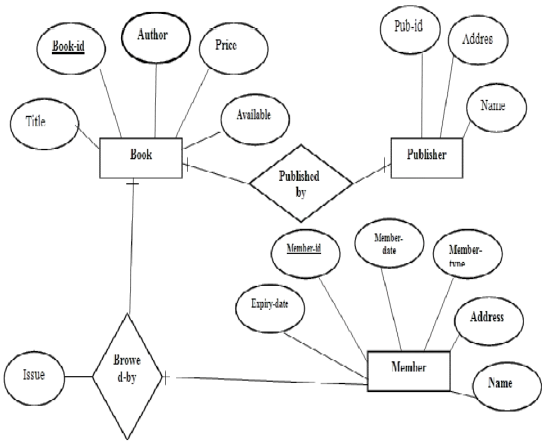
14)hierarchical model vs Network model.



15)Relational model vs hierarchical model.

16)List and draw any four symbols used in ER model.

17)Draw an ER diagram of library management system (Use books,publisher and member entities).



18)Explain strong and weak entity.

**Strong entity set:** An entity set that has sufficient attributes to form a primary key is called as Strong entity set. Example: Employee is a Strong entity with attributes as empid, name, address, salary, birthdate among which empid can be considered as primary key.

**Weak entity set:** The entity set which does not have sufficient attributes to form a primary key is called as Weak entity set. A weak entity is an entity that cannot be uniquely identified by its attributes alone; therefore, it must use a foreign key in conjunction with its attributes to create a primary key. The foreign key is typically a primary key of an entity it is related to. Example: Employee has "dependents" with name, birthdate, and relationship to employee and it can be related to employee with the help of empid, so "dependents" is a weak entity which depends on strong entity "Employee".

19)Define entity . Differenciate between strong and weak entity set with example.

20)Define recursive relationship.

21)Define tuple,field,cardinality,degree.

22)Define Table and Field.

Table: A table is a collection of related data held in table format. It is a set of data elements using a model of vertical columns and horizontal rows.

Field: Each table contains field which is a data structure, used to hold the data. It can also be termed as attribute.

23)State any 2 E.F Codd’s rule for RDBMS.

**1. The Information rule:** All information in an RDBMS is represented logically in just one way - by values in tables.

**2. The View Updating rule**: All views of the data which are theoretically updatable must be updatable in practice by the DBMS.

**3. The Guaranteed Access rule:** Each item of data in an RDBMS is guaranteed to be logically accessible by resorting to a combination of table name, primary key value, and column name.

**4. The Systematic Treatment of Null Values rule**: Null values (distinct from an empty character string or a string of blank characters and distinct from zero or any other number) are supported in a fully relational DBMS for representing missing

24)Define Normalization.List its type.

**Normalization**: Normalization can be defined as process of decomposition/division of database tables to avoid the data redundancy.

Types of Normalization: 1. 1NF 2. 2NF 3. 3NF 4. BCNF

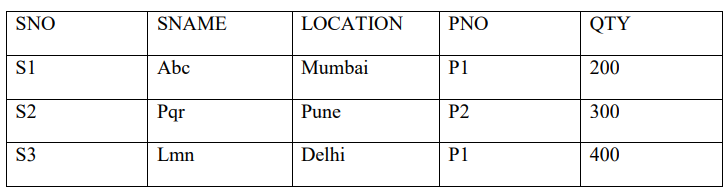
25)Lists any two needs of normalization.

26)State and explain 1NF and 2NF with examples.

First Normal Form (1NF): A relation R is said to be in first normal form (1NF) if the domain of all attributes of R are atomic. OR

A table is in the first normal form if it contains no repeating elements groups.

Example: Supplier(sno,sname,location,pno,qty)



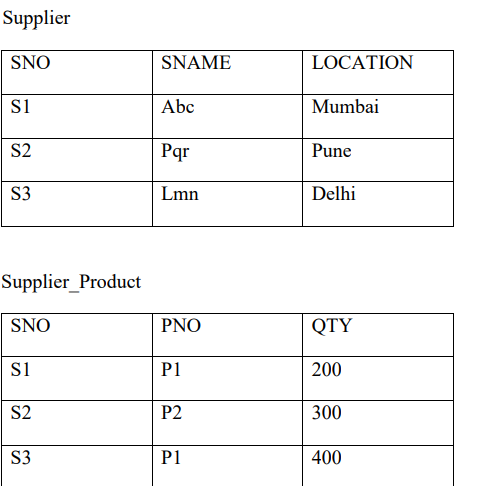
the above relation is in 1NF as all the domains are having atomic value. But it is not in 2NF.

Second Normal Form (2NF): A relation is said to be in the second normal form if it is in first normal form and all the non key attributes are fully functionally dependent on the primary key.

Example:

In the above relation NAME, LOCATION depends on SNO and QTY on (SNO, PNO) so the table can be split up into two tables as Supplier(SNO,SNAME,LOCATION) and SP(SNO,PNO,QTY) and now both the tables are in second normal form.

Supplier



27) State and explain 3NF with examples.

3NF: An entity is said to be in the third normal form when,

1) It satisfies the criteria to be in the second normal form.

2) There exists no transitive functional dependency. (Transitive functional dependency can be explained with the relationship link between three tables. If table A is functionally dependent on B, and B is functionally dependent on C then C is transitively dependent on A).

Let us consider the Schema given: (Supplier\_no,SupplierName,Supplier\_city,Order\_no,Order\_quantity, Order\_amount,Product\_code,Product name,rate) Step 1.To convert it into 2NF, We have to decompose the given table into two tables with fully functional dependencies and establishing a referential integrity constraint relationship among the two tables.

Table2: Supplier Details (Supplier\_no,Supplier\_name,Supplier\_city)

Table 3:Order Details (Order\_no,Order\_ quantity,Order\_amount,Supplier\_no Product\_code, product\_name,rate) Now the above two tables are in 2NF Step 2: To convert the above tables in 3NF, we have to decomposehem in three tables satisfying the transitive dependencies property.

Table 4: Supplier Details (Supplier\_no,Supplier\_name,Supplier\_city) Table 5: Product Details: (Product\_code, product\_name,rate) Table 6: Order Details (or Transaction Details) ((Order\_no,Supplier\_no,Product\_code,Order\_quantity,Order\_amount ) Hence the above three tables are satisfying Transitive dependencies. Thus they are in 3NF

28)list any four DDL commands with syntax.

**1.Create Syntax** : create table (Column\_name1 datatype1, column\_name2 Datatype2,…Column\_nameN DatatypeN);

2. Drop

Syntax: drop table ;

3. Desc Syntax: describe ; OR Desc

4. Truncate Syntax: truncate table ;

5. Alter Syntax: Alter table add Column\_name Datatype (size);

29)Define Candidate key, primary key, alternate key, foreign key.

**Primary key** is an attribute or set of attributes used to identify an entity from an entity set. All the values of a primary key should be unique and null values are not allowed.

**Foreign key** is an attribute of an entity which is the primary key of another entity. It is used to show relation between entities. The table containing foreign key is called the child table.

**Candidate key:** In a relation, there may be a primary key or may not, but there may be a key or combination of keys which uniquely identify the record. Such a key is called as Candidate key.

OR

A candidate key is a column, or set of columns, in a table that can uniquely identify any database record without referring to any other data. The candidate key can be simple (having only one attribute) or composite as well. For Example, {STUD\_NO, COURSE\_NO} is a composite candidate key for relation STUDENT\_COURSE.

30)State any four PL/SQL data types.

1. NUMBER or NUMBER(P,S)

2. PLS\_INTEGER

3. CHAR

4. RAW

5. ROWID

6. VARCHAR2

7. DATE

31)List data integrity constraints.

32)Describe CREATE & ALTER command with syntax and example.

The SQL CREATE TABLE statement is used to create a new table.

Syntax The basic syntax of the CREATE TABLE statement is as follows –

CREATE TABLE table\_name ( column1 datatype (size), column2 datatype(size), column3 datatype(size), .... ); Example: CREATE TABLE Persons ( PersonIDnumber(10), LastNamevarchar2(20), FirstNamevarchar2(20), Address varchar2(20), City varchar2(20) );

2)The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

i) To add Columns in a table Syntax:ALTER TABLE table\_name ADD column\_name datatype; Example ALTER TABLE Customers ADD Email varchar2(20);

ii) To delete a column in a table ALTER TABLE table\_name DROP COLUMN column\_name; Example ALTER TABLE Customers DROP COLUMN Email;

iii) To modify a column in a table Syntax: ALTER TABLE table\_name MODIFY COLUMN column\_name datatype; Example ALTER TABLE Customers MODIFY COLUMN customeridnumeric(10);

iv) To add Constraints in A table Syntax: ALTER TABLE table\_name ADD constraint constraintname (column\_name); Example: ALTER TABLE Customers ADD constraint primary key(CustomerID);

33)

|  |  |  |
| --- | --- | --- |
| **Comparison Basis** | **Primary Key** | **Foreign Key** |
| **Basic** | It is used to identify each record into the database table uniquely. | It is used to links two tables together. It means the foreign key in one table refers to the primary key of another table. |
| **NULL** | The primary key column value can never be NULL. | The foreign key column can accept a NULL value. |
| **Count** | A table can have only one primary key. | A table can have more than one foreign key. |
| **Duplication** | The primary key is a unique attribute; therefore, it cannot stores duplicate values in relation. | We can store duplicate values in the foreign key column. |
| **Indexing** | The primary key is a clustered index by default, which means it is indexed automatically. | A foreign key is not a clustered index by default. We can make clustered indexes manually. |
| **Deletion** | The primary key value can't be removed from the table. If you want to delete it, then make sure the referencing foreign key does not contain its value. | The foreign key value can be removed from the table without bothering that it refers to the primary key of another table. |
| **Insertion** | We can insert the values into the primary key column without any limitation, either it present in a foreign key or not. | The value that is not present in the column of a primary key cannot be inserted into the referencing foreign key. |
| **Temporary table** | The primary key constraint can be defined on the temporary tables. | A foreign key constraint cannot be defined on the temporary tables. |
| **Relationship** | It cannot create a parent-child relationship in a table. | It can make a parent-child relationship in a table. |

34) Scalar data types - Scalar data types haven't internal components.

* Composite data types - Composite data types have internal components to manipulate data easily.
* Reference data types - This data types work like a pointer to hold some value.
* LOB data types - Stores large objects such as images, graphics, video.
* Unknown Column types - Identify columns when not know the type of column
* 35) Domain Integrity Constraint
* Entity Integrity Constraint
* Referential Integrity Constraint
* Key Constraints

36) The SQL **ALTER TABLE** command is used to add, delete or modify columns in an existing table. You should also use the ALTER TABLE command to add and drop various constraints on an existing table.

The basic syntax of an ALTER TABLE command to add a **New Column** in an existing table is as follows.

ALTER TABLE table\_name ADD column\_name datatype;

Eg.

ALTER TABLE CUSTOMERS ADD SEX char(1);

41) The examples of DML in the Database Management System (DBMS) are as follows −

* SELECT − Retrieve data from the database.
* INSERT − Insert data into a table.
* UPDATE − Update existing data within a table.
* DELETE − Delete records from a database table.

42) DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

* List of DCL commands:
* • GRANT: This command gives users access privileges to the database.
* • REVOKE: This command withdraws the user’s access privileges given by using the GRANT command.

43)

|  |  |  |
| --- | --- | --- |
| **Comparison Basis** | **DELETE** | **TRUNCATE** |
| **Definition** | The delete statement is used to remove single or multiple records from an existing table depending on the specified condition. | The truncate command removes the complete data from an existing table but not the table itself. It preserves the table structure or schema. |
| **Language** | It is a DML (Data Manipulation Language) command. | It is a DDL (Data Definition Language) command. |
| **WHERE** | It can use the WHERE clause to filter any specific row or data from the table. | It does not use the WHERE clause to filter records from the table. |
| **Permission** | We need to have DELETE permission to use this command. | We need to have ALTER permission to use this command. |
| **Working** | This command eliminates records one by one. | This command deletes the entire data page containing the records. |
| **Lock** | It will lock the row before deletion. | It will lock the data page before deletion. |
| **Table Identity** | This command does not reset the table identity because it only deletes the data. | It always resets the table identity. |
| **Transaction** | It maintains transaction logs for each deleted record. | It does not maintain transaction logs for each deleted data page. |
| **Speed** | Its speed is slow because it maintained the log. | Its execution is fast because it deleted entire data at a time without maintaining transaction logs. |
| **Trigger** | This command can also activate the trigger applied on the table and causes them to fire. | This command does not activate the triggers applied on the table to fire. |
| **Restore** | It allows us to restore the deleted data by using the COMMIT or ROLLBACK statement. | We cannot restore the deleted data after using executing this command. |
| **Indexed view** | It can be used with indexed views. | It cannot be used with indexed views. |
| **Space** | The DELETE statement occupies more transaction space than truncate because it maintains a log for each deleted row. | The TRUNCATE statement occupies less transaction space because it maintains a transaction log for the entire data page instead of each row. |

**44) DROP is used to delete a whole database or just a table.The DROP statement destroys the objects like an existing database, table, index, or view.**

**A DROP statement in SQL removes a component from a relational database management system (RDBMS).**

**Syntax:DROP object object\_name**

**Examples:**

**DROP TABLE table\_name;**

**table\_name**: Name of the table to be deleted.

**DROP DATABASE database\_name;**

**database\_name**: Name of the database to be deleted.

TRUNCATE statement is a Data Definition Language (DDL) operation that is used to mark the extents of a table for deallocation (empty for reuse). The result of this operation quickly removes all data from a table, typically bypassing a number of integrity enforcing mechanisms. It was officially introduced in the SQL:2008 standard.

The TRUNCATE TABLE mytable statement is logically (though not physically) equivalent to the DELETE FROM mytable statement (without a WHERE clause).

Syntax:

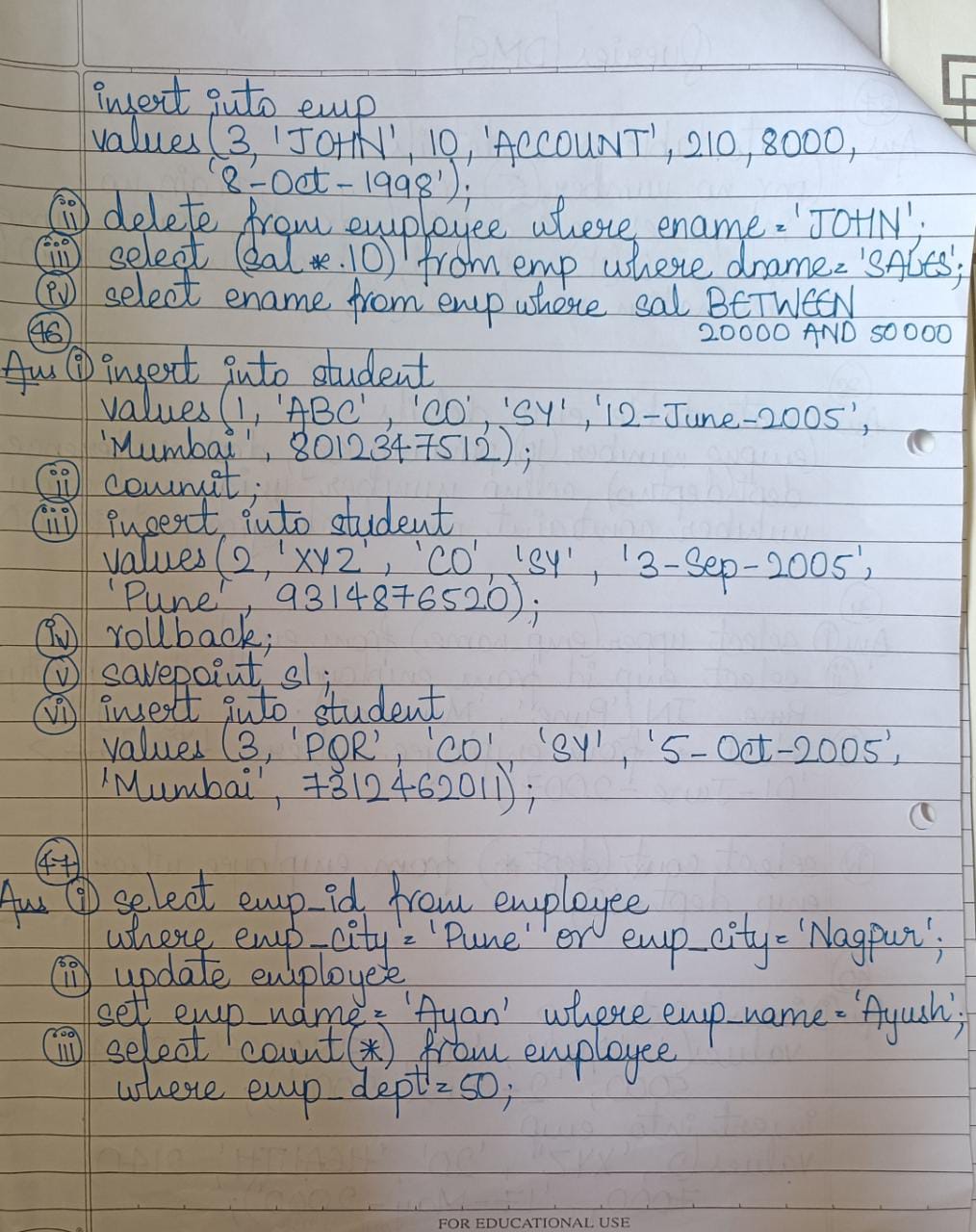
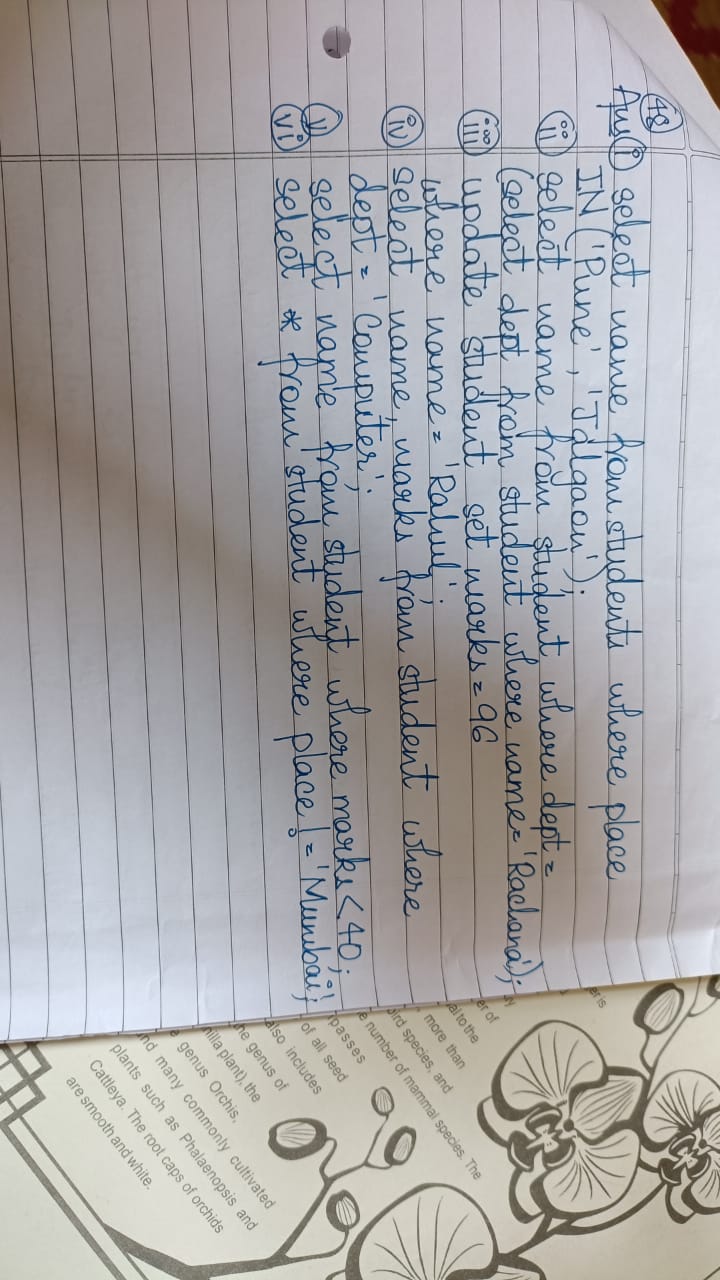
TRUNCATE TABLE table\_name;

table\_name: Name of the table to be truncated.

DATABASE name - student\_data

45) The ***rollback*** command is used to get back to the previous permanent status of the table, which is saved by the commit command.

If you want to save all the commands which are executed in a transaction, then just after completing the transaction, you have to execute the **commit** command. This command will save all the commands which are executed on a table. All these changes made to the table will be saved to the disk permanently



49)

• Arithmetic operator

• Comparison operator

• Logical operator

Arithmetic operators:

We can use various arithmetic operators on the data stored in the tables. Arithmetic Operators are:

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | The addition is used to perform an addition operation on the data values. |
| – | This operator is used for the subtraction of the data values. |

/

This operator works with the ‘ALL’ keyword and it calculates division operations.

\*

This operator is used for multiply data values.

%

Modulus is used to get the remainder when data is divided by another.

Comparison operators:

Another important operator in SQL is a comparison operator, which is used to compare one expression’s value to other expressions. SQL supports different types of the comparison operator, which is described below:

Operator Description

= Equal to.

> Greater than.

< Less than.

>= Greater than equal to.

<= Less than equal to.

<> Not equal to.

Logical operators:

The Logical operators are those that are true or false. They return true or false values to combine one or more true or false values.

Operator Description

AND

Logical AND compares between two Booleans as expressions and returns true when both expressions are true.

OR

Logical OR compares between two Booleans as expressions and returns true when one of the expressions is true.

NOT

Not takes a single Boolean as an argument and changes its value from false to true or from true to false.

Special operators:

Operat or Description

ALL

ALL is used to select all records of a SELECT STATEMENT. It compares a value to every value in a list ofof results from a query. The ALL must be preceded by the comparison operators and evaluates to TRUE if the query returns no rows.

ANY

ANY compares a value to each value in a list of results from a query and evaluates to true if the result of an inner query contains at least one row.

BETWEEN

The SQL BETWEEN operator tests an expression against a range. The range consists of a beginning, followed by an AND keyword and an end expression.

IN

The IN operator checks a value within a set of values separated by commas and retrieves the rows from the table which are matching.

EXISTS

The EXISTS checks the existence of a result of a subquery. The EXISTS subquery tests whether a subquery fetches at least one row. When no data is returned then this operator returns ‘FALSE’.

SOME SOME operator evaluates the condition between the outer and inner tables and evaluates to true if the final result returns any one row. If not, then it evaluates to false.

UNIQUE UNIQUE operator searches every unique row of a specified table.

## 50) **UNION:**

* UNION will be used to combine the result of two select statements.
* Duplicate rows will be eliminated from the results obtained after performing the UNION operation.

## **UNION ALL**

* This operator combines all the records from both the queries.
* Duplicate rows will be not be eliminated from the results obtained after performing the UNION ALL operation.

## **INTERSECT:**

* It is used to combine two SELECT statements, but it only returns the records which are common from both SELECT statements.

**MINUS**

* It displays the rows which are present in the first query but absent in the second query with no duplicates.

## 51) **Different Types of SQL JOINs**

Here are the different types of the JOINs in SQL:

* (INNER) JOIN: Returns records that have matching values in both tables
* LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
* RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
* FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

52) JOINS are used to retrieve data from multiple tables.**Outer joins are used to return all rows from at least one of the tables**. All rows are retrieved from the left table referenced with a left outer join, and all rows from the right table referenced in a right outer join.

53)

## **AVG Function**

This function returns the average value of the numeric column that is supplied as a parameter.

Select COUNT(\*) from Employee where Salary > 20000;

## **COUNT Function**

The count function returns the number of rows in the result. It does not count the null values.

Select COUNT(\*) from Employee where Salary > 20000;

## **MAX Function**

The MAX function is used to find maximum value in the column that is supplied as a parameter. It can be used on any type of DATA

Select MAX(salary) from Employee

## **SUM Function**

This function sums up the values in the column supplied as a parameter.

Select SUM(salary) from Employee

## **STDDEV Function**

The STDDEV function is used to find standard deviation of the column specified as argument.

Select STDDEV(salary) from Employee

## **VARIANCE Function**

The VARIANCE Function is used to find variance of the column specified as argument.

Select VARIANCE(salary) from Employee

54) SQL String functions are the predefined functions that allow the database users for string manipulation. These functions only accept, process, and give results of the string data type.

Following are the most important string functions in Structured Query Language:

1. ASCII()
2. CHAR\_LENGTH()
3. CHARACTER\_LENGTH()
4. CONCAT()
5. CONCAT\_WS()
6. FIND\_IN\_SET()
7. FORMAT()
8. INSERT()
9. INSTR()
10. LCASE()

55) The Group by clause is often used to**arrange identical duplicate data into groups with a select statement to group the result-set by one or more columns**. This clause works with the select specific list of items, and we can use HAVING, and ORDER BY clauses. Group by clause always works with an aggregate function like MAX, MIN, SUM, AVG, COUNT.

The SELECT statement used in the GROUP BY clause can only be used contain column names, aggregate functions, constants and expressions. The HAVING clause is used to restrict the results returned by the GROUP BY clause.

56) Arithmetic operators can perform arithmetical operations on numeric operands involved. Arithmetic operators are addition(+), subtraction(-), multiplication(\*) and division(/). The + and - operators can also be used in date arithmetic.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Meaning** | **Operates on** |
| + (Add) | Addition | Numeric value |
| - (Subtract) | Subtraction | Numeric value |
| \* (Multiply) | Multiplication | Numeric value |
| / (Divide) | Division | Numeric value |
| % (Modulo) | Returns the integer remainder of a division. For example, 17 % 5 = 2 because the remainder of 17 divided by 5 is 2. | Numeric value |

**Syntax:**

SELECT <Expression>[arithmetic operator]<expression>...

FROM [table\_name]

WHERE [expression];

57) In SQL, dates are complicated for newbies, since while working with database, the format of the date in table must be matched with the input date in order to insert. In various scenarios instead of date, datetime (time is also involved with date) is used.

In MySql the default date functions are:

NOW(): Returns the current date and time. Example:

SELECT NOW();

Output:

2017-01-13 08:03:52

CURDATE(): Returns the current date. Example:

SELECT CURDATE();

Output:

2017-01-13

CURTIME(): Returns the current time. Example:

SELECT CURTIME();

Output:

08:05:15

DATE(): Extracts the date part of a date or date/time expression.

SELECT Name, DATE(BirthTime) AS BirthDate FROM Test;

EXTRACT(): Returns a single part of a date/time. Syntax:

EXTRACT(unit FROM date);

There are several units that can be considered but only some are used such as:

MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR, etc.

And ‘date’ is a valid date expression.

58) In SQL, operators are used to perform arithmetic or string operation on variables. There are three types of operators in SQL: Arithmetic Operators, String Operators and Logical Operators. Operators are used in SQL to perform specific mathematical, logical or comparison operations on the records in a database. The comparison operators in SQL are equal to, less than, greater than, less than or equal to, greater than or equal to, not equal to and NULL. The mathematical operators are addition, subtraction, multiplication and division. The logical operators are AND and OR.

59) SQL Equal Operator (=)

1. SQL Not Equal Operator (!=)
2. SQL Greater Than Equals to Operator (>=)
3. SQL Less Than Operator (<)
4. SQL Greater Than Operator (>)
5. SQL Less Than Equals to Operator (<=)

| **Operator** | **Description** |
| --- | --- |
| AND | The **AND** operator in SQL is used to compare data with more than one condition. If all the conditions return **TRUE**, then only it will display records. |
| OR | The **OR** operator in SQL compares data with more than one condition. If either of the condition is **TRUE**, it will return data. |
| ALL | The **ALL** operator in SQL returns true when the value matches all values in a single column. It’s like **AND operator**; it will compare the value against all values in a column. |
| ANY | The **Any** operator in SQL returns true when the value matches any value in a single column set of values. It’s like an **OR operator,** and it will compare the value against any value in the column. |
| LIKE | The **LIKE** operator in SQL searches for a character string with the specified pattern using wildcards in a column. |
| IN | The **IN** operator in SQL is used to search for a specified value that matches any value in the set of multiple values. |
| BETWEEN | The **BETWEEN** operator in SQL is used to get values within a range. |
| EXISTS | The **EXISTS** operator in SQL is used to show the result if the subquery returns data. |
| NOT | The **NOT** operator in SQL is a negate operator, which means it will show data for the opposite of conditions that we mentioned in the SQL statement. |
| SOME | The **SOME** operator in SQL compares values with a single column set of values returned by the subquery. **SOME** must match at least one value in a subquery, and that value must be preceded by comparison operators. |