RDBMS Class 11 Notes

RDBMS Class 11 Notes

Database Management System (DBMS)

A database management system, often known as a DBMS, is required to develop and maintain databases on computers. Users can develop and administer databases using software called a database management system. MySQL, PostgreSQL, Microsoft Access, Oracle, Microsoft SQL Server, DB2, and Sybase are some of the well-known DBMSs.

DBMS allow us to:

- store data in a structured way.
- query the database (that is, ask questions about the data)
- sort and manipulate the data in the database
- validate the data entered and check for inconsistencies
- produce flexible reports, both on screen and on paper, that make it easy to comprehend theninformation stored in the database.

Relational Database

The term “relational database” refers to a database that contains data in several tables that are connected by a common column is known as Relational Database.

- Redundancy can be controlled
- Inconsistence can be avoided
- Data can be shared
- Security restrictions can be applied.

RDBMS Terminology

Domain

It is pool of values or the collection (set) of possible values from which the value for a column is derived.

Tables or Relation in a Database

Relational Databases store data or information in tables. A table is similar to a spreadsheet where data is stored in rows and columns. We can interlink or we can make the relation between the table using a common field.

1. **Record –** A record in a database is a collection of fields within a table that are relevant to one particular entity. This record is also called row or tuple.
2. **Field –** The term field is also refer as a column. Each column has a unique name and the content within it must be of the same type. Field is also known as Column and Attribute.
3. **MySQL –** It is an Open Source RDBMS Software. It is available free of cost.
4. **Key –** A key is an attribute or group of attributes that aids in uniquely identifying a row (or tuple) in a table (or relation). When we want to create connections between the various columns and tables in a relational database, we also employ a key.

Types of Keys in DBMS

There is a 7 different types of keys, but in your syllabus only 4 keys are given we are going to discusses based on these four key only.

1. **Primary Key –** The group of one or more columns used to uniquely identify each row of a relation is called its Primary Key.

2. **Candidate Key –** A column or a group of columns which can be used as the primary key of a relation is called a Candidate key because it is one of the candidates available to be the primary key of the relation.
3. **Alternate Key –** A candidate key of a table which is not selected as the primary key is called its Alternate Key.
4. **Foreign Key –** A primary key of a base table when used in some other table is called as Foriegn Key.

Characteristics of MySQL-

a. It requires no cost or payment for its usage.
b. MySQL has superior speed, is easy to use and is reliable.
c. MySQL uses a standard form of the well-known ANSI-SQL standards.
d. MySQL is a platform independent application which works on many operating systems like Windows, UNIX, LINUX etc. and has compatibility with many languages including JAVA , C++, PHP, PERL, etc.
e. MySQL is an easy to install RDBMS and is capable of handling large data sets.

Categories of SQL Commands

SQL commands can be classified into the following categories:

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)

Data Definition Language (DDL)

The DDL part of SQL permits database tables to be created or deleted. It also defines indices (keys), specifies links between tables, and imposes constraints on tables.

Examples of DDL commands in SQL are:

- **CREATE DATABASE –** creates a new database
- **CREATE TABLE –** creates a new table
- **ALTER TABLE –** modifies a table
- **DROP TABLE –** deletes a table

Data Manipulation Language (DML)

The query and update commands form the DML part of SQL: Examples of DDL commands are:

- **SELECT –** extracts data from a table
- **UPDATE –** updates data in a table
- **DELETE –** deletes data from a table
- **INSERT INTO –** inserts new data into a table

MySQL Data Types

Data types can be broadly classified into following categories:-

- Numeric Types
- Alphanumeric Types
- Date Time

Numeric Types

They are used for describing numeric values like mobile number, age, etc.

- **Decimal –** Decimal Data type represent decimal number and store the decimal values in database.
- **Integer –** Integer Data type is used for storing integer values.

Alphanumeric Types

Alphanumeric data types are used to store character values in the database like student name, address etc.

- **Char(size) –** A fixed-length string from 1 to 255 characters in length right-padded with spaces to the specified length when stored. Values must be enclosed in single quotes or double quotes
- **Varchar(size) –** A variable-length string from 1 to 255 characters in length; i.e. VARCHAR(25). Values must be enclosed in single quotes or double quotes.

Date Time Types

When specifying date and time values for a column used in a database table, date time data types are utilised. Information like a person’s date of birth, admission date, and so on might be stored on it.

- Date (Stores month, day and year information)
- Time (Store hour , minute and second information)

Adding constraints in table

Keeping a manual check on the data entering the tables is frequently not possible. The entered info might not be accurate. Constraints, which MySQL offers, are a set of guidelines that assist us somewhat ensure the accuracy of the data. These limitations are:

1. **Primary Key –** Sets a column or a group of columns as a primary key of the table. Therefore, NULLs and Duplicate values in this column are not accepted.

2. **NOT NULL –** Makes sure that NULLs are not accepted in the specified column.
3. **FOREIGN KEY –** Data will be accepted in this column, if the same data value exists in a column in another related table. This other related table name and column name are specified while creating the foreign key constraint.
4. **UNIQUE –** Make sure that duplicate values in the specified column are not accepted.

SQL Commands

1. Creating a Database

Syntax – CREATE DATABASE <table name>;

Example – CREATE DATABASE School;

2. Use Database

A database called School is established on the system once the above statement has been executed. You can choose any database name to work on the database.

Syntax – USE <table name>;

Example – USE School;

3. Display the current database

We must know which database we are currently working in, to see the name of the current database we use SELECT command.
Syntax – SELECT DATABASE();

4. Creating a Table

After creating a database, the next step is creation of tables in the database. For this CREATE
TABLE statement is used.

Syntax – CREATE TABLE table_name(column1 datatype, column2 datatype, column3 datatype, ….. columnN datatype);

Example – CREATE TABLE Students(Rollno Integer, Name Varchar(25));

5. Create a Table with constraints

Example – CREATE TABLE Shoes (Code CHAR(4) PRIMARY KEY, Name VARCHAR(20), type VARCHAR(10), size INT(2), cost DECIMAL(6,2), margin DECIMAL(4,2), Qty INT(4));

6. Viewing the tables in the database

To see a list of tables present in the current database we will use SHOW TABLES.
Syntax – SHOW TABLES;
Example – SHOW TABLES;

7. Viewing the structure of the table

The DESCRIBE statement can be used to see the structure of a table as indicated in the Create Statement. It displays the Column names, their data types, whether Column must contain data, whether the Column is a Primary key.

Syntax: DESCRIBE <table name>;

OR

DESC <table name>;

Example – DESCRIBE Student;

8. Modifying the structure of table

When we create a table we define its structure. We can also change its structure i.e. dd, remove or change its column(s) using the ALTER TABLE statement. ALTER TABLE is used to add a constraint, to remove a constraint, to remove a column from a table, to modify a table column.

Syntax – ALTER TABLE <table name> ADD/DROP <column name> [datatype];

ALTER TABLE <table> MODIFY <column> <new definition> ;

If we want to add a column named Games in the student table .

Example –

```
ALTER TABLE Student ADD Games VARCHAR(20);
```

9. Deleting the structure of the table

Sometimes there is a requirement to remove a table from the database. In such cases we don’t want merely to delete the data from the table, but we want to delete the table itself. DROP TABLE command is used for this purpose.

Syntax –

```
DROP TABLE <table name> ;
```

10. Insert Command

After creating database and relations, we can add data in the relations. INSERT INTO command
is used to enter values in a table. Syntax of insert command is as follows :

Syntax –

```
INSERT INTO <TABLENAME >(COLUMN NAMES>) VALUES (value1, value2, value3,……)
```

Example –

```
 INSERT INTO STUDENT(RNO,GENDER, LNAME,FNAME)
VALUES(8,'F','SHANU', 'DEEPAKSHI');
```

11. Select Command

SELECT command is used to view the data from a relation in a database. It returns a result set of data/ records from one or more relations.

Syntax –

```
SELECT <column name> FROM <table name>;
```

12. DISTINCT (Displaying values without repetition)

If a column contains repeated values then the select statement gives the result set with repeated values like if a command is given to display DISTINCT keyword is used to eliminate repeated values
Consider the query:
Example – select marks from student;

13. WHERE Clause

Where clause is used to fetch data based on a criteria/ condition. Criteria can be given using an expression.

Syntax:

```
SELECT <column name1> [,<column name> ,….] FROM <table name> WHERE <condition>;
```

Operators

Arithmetic Operators

- +
- –
- *
- /
- %

Relational Operators

- =
- <
- >
- <=
- >=
- != or <>

Logical Operator

- AND
- OR
- NOT

HANDLING NULL Values

IS operator is used to match NULL value in the expression.
For example, following command is to display the records with NULL values in column edesig from table employee
Example – select empname, zone from employee where zone IS NULL;

IS operator

IS operator is used to compare equality with NULL whereas IS NOT may be used for comparing the values not equal to NULL;
Example – select empname, zone from employee where zone IS NOT NULL;

Alias Name to columns

To make the output more user friendly we can give customized heading to column name in the select command. Keyword AS is used to give column alias in a set of

single quotes as shown in the example below:
For example :
Example – select rno AS ‘Roll Number’ , fname AS ‘First Name’ , marks from Student;

Inserting Text within the Query SQL

To make the output more user friendly SQL supports inserting text within the output display. The text is displayed only in the output and no changes are made in the table.
For example :
Example – select fname ,’has roll number ‘, Rno AS ‘Roll Number’ from student;

IN operator

IN operator is very useful when we wish to fetch selected records which match a certain set of values. Suppose we wish to display list of students who have score marks 88,90,92,95 , then IN operator can be used as follows :
Example – select fname, marks from student WHERE marks IN (88, 92, 95);

LIKE keyword

LIKE is used for pattern matching and is very useful. Following characters used for applying pattern matching:

- **% –** percent symbol is used to match none or more characters **_ –**
- underscore character is used to match occurrence of one character in the string

For example : To search for records having first name starting with letter ‘R’;
Example – select * from student where fname like ‘A%’;

ORDER BY

The result set fetched by the query can be displayed in sorted order. The sorting can be on the basis of any particular column from the table using ORDER BY clause.

Syntax is :

```
SELECT <column name>, [<column name >…]
FROM <table name>
[ WHERE <condition> ]
ORDER BY < column name> [DESC];
```

UPDATE

UPDATE command is used to modify data of records within the table. It is a type of

DML and is used to make changes in the values entered in the table :

Syntax: `

```
UPDATE <table name> SET <column name> = <value>, [<column name> =
<value> , …] [WHERE <condition> ];
```

DELETE

DELETE Command is used to remove records from the table.

Syntax of the command is :

```
DELETE from <table name> [Where <condition>];
```