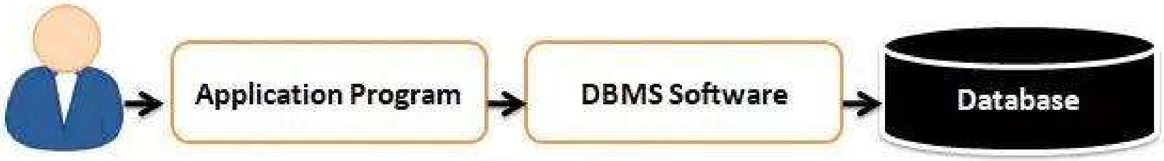
Database Query using SQL Class 12 Notes

Database concepts | Computer Science Class 12 Database Notes

Introduction to Database Concepts

A database is an organized collection of structured data stored electronically on the computer. DBMS (database management system) helps to store, retrieve, update, and manage data efficiently.



Need of Database

Databases are needed because they provide an efficient way for storing, retrieving, and managing large amounts of data.

- A database can manage a large amount of data without repetition (redundancy).
- Databases store data in structured formats like tables, documents, or key-value pairs.
- Query can be used for accessing data efficiently and quickly.
- Databases can provide security like user authentication, access control, etc.
- Multiple users can access and update data at the same time.

What is database?

A data database is an organized collection of structured data stored electronically in a computer and managed by a database management system (DBMS).

Advantages of using DBMS?

Following are the advantages of using a DBMS –

1. **Reduction in Redundancy –** All the data is stored at one place. There is no repetition of the same data. This also reduces the cost of storing data on hard disks or other memory devices.
2. **Improved Consistency –** The chances of data inconsistencies in a database are also reduced as there is a single copy of data that is accessed or updated by all the users.
3. **Improved Availability –** Same information is made available to different users. This helps sharing of information by various users of the database.
4. **Improved Security –** The DBA can protect the database by using passwords and restricting users' database access rights.
5. **User Friendly –** Because of its user-friendly interface, it reduces users' dependence on computer specialists to carry out various data-related actions in a DBMS.

Relational data model

A database is a collection of tables, and data can be organized in a database using one or more tables. In the relational data model, data is stored in structured tables, and relationships can be created between these tables using primary and foreign keys. This connection between two tables is known as a relation.

- **Attribute:** Attributes are the properties of entities or objects stored in a database. In easy words, you can say that it is a column heading. For example, in a “Student” table, attributes could include “Name,” “Age,” or “Grade.”
- **Tuple:** A tuple is also known as a record; it is a single row in a table. For example, Rajesh Kumar, 24, a
- **Domain:** It is a set of values or attributes that share a common meaning or purpose. For example, the “Grade” attribute could be “A,” “B,” “C,” “D,” or “E.”

- **Degree:** Degree refers to the number of columns or attributes in a table. For example, in a student table, there are three attributes: name, age, and grade. It means that table has a degree of 3.
- **Cardinality:** Cardinality refers to the number of rows in a table. For example, if a student table has 10 records, it means that its cardinality is 10.

Keys in database

Keys are important in DBMS; keys help to make each record uniquely in the table. Keys are also important for making relations between tables and for data integrity. There are basically four different types of keys:

- **Primary key:** A primary key or simply a key is a field that uniquely identifies a row in a table.
- **Candidate key:** All the field values that are eligible to be the primary key are the candidate keys for that table.
- **Alternate key:** Out of the candidate keys, one or two are made as primary keys. The others are the alternate keys.
- **Foreign key:** If a field or a combination of fields of one table can be used to uniquely identify records of another table, then that particular field is known as the foreign key.

Structured Query Language

What is SQL?

SQL stands for Structured Query Language. It is a standard programming language which allows the users to perform various tasks on the table, like inserting data, updating data, deleting data and retrieving data from the database.

Data Definition Language and Data Manipulation Language

SQL commands are categorised in two types:

DDL (Data Defination Language)	DML (Data Manipulation Language)
Defines database structure	Manages and manipulates data inside the table.
CREATE, ALTER, DROP and TRUNCATE	SELECT, INSERT, UPDATE and DELETE
Does not handle actual data; only works on structure.	Directly changes or retrieves data
Creating a new table, adding a column, etc.	Adding a new record, updating a value, etc.

What is Data type?

A data type refers to the type of data that will be stored in that particular field. The memory size of a field varies according to its data type.

Commonly used data types in MySQL

Data type	Description
char(n)	CHAR is a fixed-length character data type. means it always reserves space for the specified number of characters (0-255).
varchar(n)	VARCHAR does not have a fixed length; it stores up to the specified length. For example, if you have defined varchar(30) and store “MUMBAI”, then it will use space for only 6 characters.
int	INT specifies an integer value. Each INT value occupies 4 bytes of storage.

Data type	Description
float	Holds numbers with decimal points. Each FLOAT value occupies 4 bytes.
date	The DATE type is used for dates in ‘YYYY-MM-DD’ format.

Difference between char and varchar

Char	Varchar
Fixed length character string.	Variable length character string.
It uses static memory location.	It uses dynamic memory location.
Faster to access.	Takes less disk space.

What is Constraints?

Constraints are the certain types of restrictions on the data values that an attribute can have.

Constraint	Description
NOT NULL	Ensures that a column cannot have NULL values where NULL means missing/ unknown/ not applicable value.
UNIQUE	Ensures that all the values in a column are distinct/unique
DEFAULT	A default value specified for the column if no value is provided
PRIMARY KEY	The column which can uniquely identify each row/record in a table.
FOREIGN KEY	The column which refers to value of an attribute defined as primary key in another table

Create database

To create a database, we use the **CREATE DATABASE** statement as shown in the following syntax:

```
CREATE DATABASE databasename;
```

Use database

The use statement selects a specific data type in the database.

```
USE database_name;
```

Show databases

SHOW DATABASES will display a list of databases in the MySQL server.

```
SHOW DATABASES;
```

Drop database

The DROP DATABASE command is used to drop an existing database.

```
DROP DATABASE databasename;
```

Show table

The SHOW TABLE command is used to display the table in the database.

```
SHOW TABLES;
```

Create table

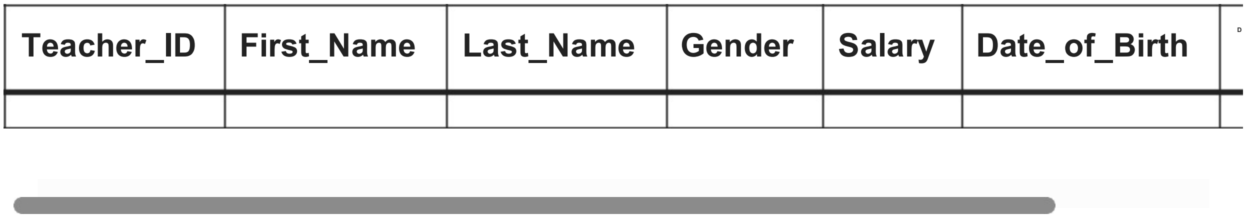
The CREATE TABLE is used for creating a table in a database. In the CREATE TABLE command, the column name and data type must be specified.

```
CREATE TABLE table_name (
   column1 datatype,
   column2 datatype,
   ....
);
```

Question> Write a Query to Create a new table where the field will be Teacher_ID, First_Name, Last_Name, Gender, Date_of_Birth, Salary, Dept_No.

```
CREATE TABLE Teacher
(
Teacher_ID INTEGER,
First_Name VARCHAR(20),
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2),
Date_of_Birth DATE,
Dept_No INTEGER
);
```

Output –



Teacher_ID	First_Name	Last_Name	Gender	Salary	Date_of_Birth	D

Describe table

The DESCRIBE TABLE is commonly written as DESC TABLE or DESCRIBE. This command is used to retrieve metadata about a table, like field, data type, null, key constraints, etc.

```
DESCRIBE table_name;
```

Alter table (add and remove an attribute, add and remove primary key)

The ALTER TABLE is used to add, delete or modify columns in an existing table.

```
ALTER TABLE table_name ADD column_name datatype;
```

Add and remove an attribute

Question> Write an SQL command for adding an address column in the Teacher table.

```
ALTER TABLE Teacher ADD address varchar(50);
```

Question> Write an SQL command for removing the address column in the Teacher table.

```
ALTER TABLE Teacher DROP COLUMN address;
```

Add and remove primary key

Question> Write an SQL command for adding a primary key on the first column of the teacher table.

```
ALTER TABLE Teacher ADD PRIMARY KEY (Teacher_ID);
```

Question> Write an SQL command for removing a primary key from the teacher table.

```
ALTER TABLE Teacher DROP PRIMARY KEY;
```

Drop table

The DROP TABLE command is used to remove a table permanently from a database.

```
DROP TABLE table_name;
```

INSERT INTO Statement

To insert a new record in a table, the INSERT INTO statement can be used. There are two different ways to insert a value in the table.

```
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...);
```

```
INSERT INTO table_name
VALUES (value1, value2, ...);
```

DELETE Statement

The SQL DELETE statement is used to remove one or more rows from a table based on the condition using the WHERE clause.

```
DELETE FROM table_name WHERE condition;
```

SELECT Statement

The SELECT statement is used to display data from a table.

To display all records from the table teacher.

```
SELECT * FROM teacher;
```

The WHERE clause is used to filter records based on condition.

```
SELECT * FROM teacher WHERE Teacher_ID='T100';
```

SQL Aliases

Aliases are used as a temporary name of the column; it helps to make column names more readable. An alias is created with the AS keyword during the execution of the query.

```
SELECT Teacher_ID AS ID, First_Name AS Teacher_Name FROM Teacher;
```

Distinct clause

The DISTINCT statement is used to retrieve only unique values from the table; it eliminates duplicate records from the table.

```
SELECT DISTINCT Department FROM Teacher;
```

Where clause

The WHERE clause is used to filter records from the table based on the specific condition.

```
SELECT * FROM Teacher WHERE Salary>50000;
```

a. The SQL IN Operator

The IN operator allows you to filter multiple values in a WHERE clause.

```
SELECT * FROM Teacher WHERE Department IN ('IT', 'ENGLISH', 'MATHS');
```

b. The SQL BETWEEN Operators

The BETWEEN operator select values within a given range.

```
SELECT * FROM Teacher WHERE Salary BETWEEN 50000 AND 75000;
```

c. SQL like Statement

The LIKE operator is used in a WHERE clause to search for a specific value from the column. There are two wildcards used in conjunction with the LIKE operator:

- The percent sign % represents zero, one, or multiple characters.

- The underscore sign _ represents one single character.

```
SELECT * FROM Teacher
WHERE First_Name LIKE 'a%';
```

The SQL order by

The ORDER BY keyword is used to sort the data in ascending or descending order.

```
SELECT * FROM Teacher ORDER BY Teacher_ID;
```

Meaning of null

NULL indicates that the value is unknown or no value; this NULL is different from zero or empty.

is null: IS NULL value is used to check whether the column is NULL or not.

```
SELECT column_names
FROM table_name
WHERE column_name IS NULL;
```

is not null: IS NOT NULL Value is used when you want to check whether a column’s value is not NULL.

```
SELECT column_names
FROM table_name
WHERE column_name IS NOT NULL;
```

Aggregate functions (max, min, avg, sum, count)

max(): The MAX() function returns the largest value from the table.

```
SELECT MAX(Salary) FROM Teacher;
```

min(): The MIN() function returns the smallest value from the table.

```
SELECT MIN(Salary) FROM Teacher;
```

avg(): The AVG() function returns the average value of the given number in the table.

```
SELECT AVG(Salary) FROM Teacher;
```

sum(): The SUM() function is used to find the sum of numbers in the table.

```
SELECT SUM(Salary) FROM Teacher;
```

count(): The COUNT() function is used to count the number of rows present in the table.

```
SELECT COUNT(Teacher_ID) FROM Teacher;
```