Lecture 8

Database Management System

Er. Shiva Kunwar

Lecturer, GU

Lesson 1: Introduction to DBMS (5hrs)

1. Overview of Database and DBMS
2. Characteristics and Applications
3. Data Abstraction and Independence
4. Database Users and Administrator
5. Application Architecture
6. **Basics of Database Language (DDL, DML, DCL) + Lab**

DDL – Data Definition Language

Definition: DDL is used to define and manage the structure of the database.

Content:

- Creating tables, views, and indexes.
- Modifying the structure of existing tables.
- Defining constraints (e.g., primary keys, foreign keys).
- Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

DDL – Data Definition Language

Here are some tasks that come under DDL:

- **Create**: It is used to create database and its objects in the database.
- **Alter**: It is used to alter the structure of the existing database.
- **Drop**: It is used to delete objects from the database.
- **Truncate**: It is used to remove all records from a table.
- **Rename**: It is used to rename an object.
- **Comment**: It is used to comment on the data dictionary.

These commands are used to update the database schema that's why they come under Data definition language.

Practical DDL

- The general syntax for logging in to MySQL from the command line is:
- *mysql -u <username> -p*

- With root as username,
- *mysql -u root –p*

- If password is blank, -p can be omitted.

Practical DDL

- Use Create command to create a database.
- *create database gu;*
- Use Show command to show the list of all available database in the server.
- *show databases;*
- Use use command to activate the newly created database.
- *use gu;*
- Use show tables command to show list of all tables in the active database.
- *show tables;*

Practical DDL - Create

- CREATE TABLE: This is used to create a new relation (table).
- Syntax: CREATE TABLE <relation_name/table_name >
- (field_1 data_type(size),field_2 data_type(size), .. .)

- *CREATE TABLE Students (StudentID INT PRIMARY KEY,*
- *FirstName VARCHAR(50),*
- *LastName VARCHAR(50),*
- *Age INT ;*

Practical DDL - Alter

- ALTER TABLE ...ADD...: This is used to add some extra fields into existing
- relation.
- Syntax: ALTER TABLE relation_name ADD (new field_1 data_type(size), new field_2
- data_type(size),..);

- *ALTER TABLE Students*
- *ADD COLUMN Email VARCHAR(100);*

Practical DDL - Alter

- ALTER TABLE...MODIFY...: This is used to change the width as well as data
- type of fields of existing relations.
- Syntax: ALTER TABLE relation_name MODIFY (field_1 newdata_type(Size), field_2
- newdata_type(Size),...field_newdata_type(Size))

- *ALTER TABLE Students*
- *ALTER COLUMN GPA DECIMAL(3, 2);*

Practical DDL - Alter

- ALTER TABLE..DROP.... This is used to remove any field of existing relations.
- Syntax: ALTER TABLE relation_name DROP COLUMN (field_name);

- *ALTER TABLE Students*
- *DROP COLUMN Age;*

Practical DDL - Alter

- ALTER TABLE..RENAME...: This is used to change the name of fields in
- existing relations.
- Syntax: ALTER TABLE relation_name RENAME COLUMN (OLD field_name) to (NEW field_name)

- *ALTER TABLE Students*
- *RENAME COLUMN FirstName TO First_Name;*

Practical DDL - Drop

- DROP TABLE: This is used to delete the structure of a relation. It permanently deletes the records in the table.
- Syntax: DROP TABLE relation_name;

- *DROP TABLE Teachers;*

Practical DDL - Rename

- RENAME: It is used to modify the name of the existing database object.
- Syntax: RENAME TABLE old_relation_name TO new_relation_name;

- *RENAME TABLE Students TO AllStudents;*

Practical DDL - Describe

- DESCRIBE: It is useful for exposing details about a table (columns, data types)..
- Syntax: DESCRIBE TABLE relation_name;

- *DESCRIBE TABLE Students;*

Practical DDL - Truncate

- TRUNCATE: It is useful for removing all records from a table while keeping the table structure.
- Syntax: TRUNCATE TABLE relation_name;

- *TRUNCATE TABLE Teachers;*

Practical DDL - Comment

- COMMENT: It is useful for adding comments or descriptions to database objects.
- Syntax: COMMENT ON TABLE relation_name IS ‘comments to add’;

- *COMMENT ON TABLE Students*
- *IS 'Contains information about students';*

Practical DDL - Assignment

- *Create a table EMPLOYEE with following schema:*
- *(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary)*
- *Add a new column; HIREDATE to the existing relation.*
- *Change the datatype of JOB_ID from char to varchar2.*
- *Change the name of column/field Emp_no to E_no.*
- *Modify the column width of the job field of emp table.*

DML – Data Manipulation Language

Definition: DML is responsible for manipulating data stored in the database.

- It is used for accessing and manipulating data in a database.
- It handles user requests.
- It includes operations like SELECT, INSERT, UPDATE, and DELETE.
- DML establishes communication between user and database.

DML – Data Manipulation Language

Here are some tasks that come under DML

- **Select**: It is used to retrieve data from a database.
- **Insert**: It is used to insert data into a table.
- **Update**: It is used to update existing data within a table.
- **Delete**: It is used to delete all records from a table.
- **Merge**: It performs UPSERT operation, i.e., insert or update operations.
- **Call**: It is used to call a structured query language or a Java subprogram.
- **Explain Plan**: It has the parameter of explaining data or data access path.
- **Lock Table**: It controls concurrency.

Practical DML - Select

- SELECT: Used to retrieve data from one or more tables in a database.
- Syntax: SELECT column1, column2, ...
- FROM table_name
- WHERE condition;

- *SELECT FirstName, LastName*
- *FROM Students*
- *WHERE GPA > 3.0;*
- *Note: * can be used to select all columns.*

Practical DML - Insert

- INSERT: Used to insert new records into a table.
- Syntax: INSERT INTO table_name (column1, column2, ...)
- VALUES (value1, value2, ...);

- *INSERT INTO Students (StudentID, FirstName, LastName, GPA)*
- *VALUES (1, 'John', 'Doe', 3.5);*

Practical DML - Update

- UPDATE: Used to modify existing data in a table.
- Syntax: UPDATE table_name
- SET column1 = value1, column2 = value2, ...
- WHERE condition;

- *UPDATE Students*
- *SET GPA = 3.8*
- *WHERE StudentID = 1;*

Practical DML - Delete

- DELETE: Used to delete records from a table based on specified conditions.
- Syntax: DELETE FROM table_name
- WHERE condition;

- *DELETE FROM Students*
- *WHERE GPA < 2.0;*

Practical DML - Merge

- MERGE: Performs an operation that either inserts new rows or updates existing rows, based on a specified condition.
- Generally used for UPSERT operations (INSERT or UPDATE).
- *MERGE INTO target_table USING source_table*
- *ON (condition)*
- *WHEN MATCHED THEN*
- *UPDATE SET column1 = value1, ...*
- *WHEN NOT MATCHED THEN*
- *INSERT (column1, column2, ...) VALUES (value1, value2, ...);*

Practical DDL - Assignment

- *Create a table EMPLOYEE with following schema:*
  (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary)
- *Insert at least 5 rows in the table.*
- *Display all the information of EMP table.*
- *Display the record of each employee who works in department D10.*
- *Update the city of Emp_no-12 with current city as Nagpur.*
- *Display the details of Employee who works in department MECH.*
- *Delete the email_id of employee James.*
- *Display the complete record of employees working in SALES Department.*

END OF LECTURE 8

- SHIVA.KUNWAR@HOTMAIL.COM
- +977-9819123654

Google classroom code : drbzdcf

PREVIEW FOR LECTURE 9

RELATIONAL MODEL