

Experiment No. 2

Group: - 1 from T4 Batch

0077 – Aakash Joshi

2039 – Akanksha Lokhande

Aim: -

SQL Queries:

- a. Design and Develop SQLDDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym, different constraints etc.
- b. Write at least 10 SQL queries on a suitable database application using SQL DML statements.

Note: Instructor will design the queries which demonstrate the use of concepts like Insert, Select, Update, Delete with operators, functions, and set operator etc.

Date: - 18/8/2023

Theory: -

Structured Query Language (SQL) is a powerful and ubiquitous programming language designed for managing, querying, and manipulating relational databases. SQL serves as a bridge between humans and databases, allowing users to interact with data using straightforward, declarative commands. SQL commands are categorized into several key operations:

1. Data Query: SQL enables the retrieval of specific data from a database using SELECT statements. This operation forms the basis for generating reports, extracting insights, and fetching information from tables.
2. Data Manipulation: SQL provides the means to insert, update, and delete data using INSERT, UPDATE, and DELETE statements. This allows for efficient data modification while maintaining data integrity.
3. Data Definition: SQL helps define the structure of a database, including creating tables, modifying their attributes, and establishing relationships between tables using CREATE, ALTER, and DROP statements.
4. Data Control: SQL allows the management of user access, privileges, and permissions to ensure data security and confidentiality.

SQL's simplicity, versatility, and widespread adoption make it an essential tool for data professionals, developers, and analysts in effectively working with relational databases.

SQL Data Definition Language (DDL) statements are instrumental in defining and managing SQL objects within a relational database. These statements create, modify, and control various objects, including:

1. Table: CREATE TABLE defines the structure for data storage.
2. View: CREATE VIEW creates a virtual table, offering a customized perspective on the data.
3. Index: CREATE INDEX enhances query performance by optimizing data retrieval.
4. Sequence: CREATE SEQUENCE generates unique numeric values, often used for primary keys.
5. Synonym: CREATE SYNONYM simplifies complex object names for ease of use.
6. Constraints: Constraints like UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, and NOT NULL maintain data integrity and enforce specific rules within the database.

DDL statements are fundamental in shaping the database schema and ensuring data accuracy and consistency.

Conclusion: -

This SQL exercise provides a comprehensive understanding of SQL objects and data manipulation. Key takeaways include mastering SQLDDL statements for object creation, proficiency in data manipulation with DML statements, and practical application in real-world scenarios. Learners also gain problem-solving skills, data integrity knowledge, and the ability to optimize queries. This exercise equips individuals for effective data-driven decision-making and database administration.

Code: -

```
-- Create the Twitch database
```

```
CREATE DATABASE TwitchDB1;
```

```
-- Switch to the newly created database
```

```
USE TwitchDB1;
```

```
-- Create the User table
```

```
CREATE TABLE User (  
    uid INT PRIMARY KEY,  
    username VARCHAR(255) NOT NULL,  
    streamer_id INT,  
    chat_id INT,  
    video_id INT,  
    stream_preference VARCHAR(255),  
    email VARCHAR(255) NOT NULL,  
    mobile_number VARCHAR(20)  
);
```

```
-- Create the Stream table
```

```
CREATE TABLE Stream (  
    stream_id INT PRIMARY KEY,  
    title VARCHAR(255) NOT NULL,  
    stream_category VARCHAR(50),  
    start_time DATETIME NOT NULL,  
    viewer_count INT,  
    end_time DATETIME,  
    quality VARCHAR(20)  
);
```

-- Create the Chat table

```
CREATE TABLE Chat (  
    chat_id INT PRIMARY KEY,  
    timestamp DATETIME NOT NULL,  
    message_content TEXT  
);
```

-- Create the Video table

```
CREATE TABLE Video (  
    video_id INT PRIMARY KEY,  
    title VARCHAR(255) NOT NULL,  
    duration INT,  
    upload_date DATE,  
    upload_time TIME,  
    viewer_count INT,  
    quality VARCHAR(20)  
);
```

-- Add foreign key constraints

```
ALTER TABLE User  
ADD FOREIGN KEY (streamer_id) REFERENCES Stream(stream_id);
```

```
ALTER TABLE User  
ADD FOREIGN KEY (chat_id) REFERENCES Chat(chat_id);
```

```
ALTER TABLE User
```

```
ADD FOREIGN KEY (video_id) REFERENCES Video(video_id);
```

```
-- Insert data into the Stream table
```

```
INSERT INTO Stream (stream_id, title, stream_category, start_time, viewer_count, end_time, quality)
```

```
VALUES
```

```
(1, 'Gaming Stream 1', 'Gaming', '2023-10-25 12:00:00', 1000, '2023-10-25 14:00:00', 'HD'),
```

```
(2, 'Cooking Show 1', 'Cooking', '2023-10-25 18:00:00', 800, '2023-10-25 20:00:00', 'SD'),
```

```
(3, 'Music Jam Session 1', 'Music', '2023-10-25 15:00:00', 500, '2023-10-25 17:00:00', 'HD');
```

```
-- Insert data into the Chat table
```

```
INSERT INTO Chat (chat_id, timestamp, message_content)
```

```
VALUES
```

```
(1, '2023-10-25 12:30:00', 'Hello, everyone!'),
```

```
(2, '2023-10-25 18:15:00', 'Whats cooking today?'),
```

```
(3, '2023-10-25 15:30:00', 'Play your favorite song!');
```

```
-- Insert data into the Video table
```

```
INSERT INTO Video (video_id, title, duration, upload_date, upload_time, viewer_count, quality)
```

```
VALUES
```

```
(1, 'Gaming Highlights', 120, '2023-10-25', '14:30:00', 5000, 'HD'),
```

```
(2, 'Cooking Masterclass', 60, '2023-10-25', '20:30:00', 3500, 'SD'),
```

```
(3, 'Music Concert Replay', 180, '2023-10-25', '18:30:00', 7000, 'HD');
```

```
-- Insert data into the User table
```

```
INSERT INTO User (uid, username, streamer_id, chat_id, video_id, stream_preference, email,
mobile_number)
```

```
VALUES
```

```
(1, 'user1', 1, 1, 1, 'Gaming', 'user1@email.com', '123-456-7890'),
```

```
(2, 'user2', 2, 2, 2, 'Cooking', 'user2@email.com', '987-654-3210'),
```

```
(3, 'user3', 3, 3, 3, 'Music', 'user3@email.com', '555-555-5555');
```

```
select * from chat;
```

```
select * from stream;
```

```
select * from user;
```

```
select * from video;
```

```
-- Update the viewer count for a stream with stream_id = 1
```

```
UPDATE Stream
```

```
SET viewer_count = 1200
```

```
WHERE stream_id = 1;
```

```
select * from stream;
```

Output: -

	chat_id	timestamp	message_content
▶	1	2023-10-25 12:30:00	Hello, everyone!
	2	2023-10-25 18:15:00	Whats cooking today?
	3	2023-10-25 15:30:00	Play your favorite song!
•	NULL	NULL	NULL

