35256 – RELATIONAL DATABASE MANAGEMENT SYSTEM PRACTICAL

DETAILED SYLLABUS

OBJECTIVES

On completion subject, the students must be able to

- Define data , database , database Management systems and data base models.
- Compare file processing and database system.
- Study about architecture of DBMS.
- Understand the concept of Data warehousing , Big Data and client/Server Technology
- State CODD’s Rules.
- Explain normalization and explain different types of Normal Forms.
- Create Normalized Database structure files .
- Perform all database DDL, DML, DCL, and all related commands.
- Write Logical and Conditional statement for Database Query.
- Write procedures and functions .
- Create and use Triggers.
- Understanding Data warehousing & Introduction to Big data and NoSQL

LAB EXERCISES

PART - A

1. Install, configure and connect to MySQL server and MySQL workbench in Windows. Create a database, backup and restore the database.
2. Create a simple database for Social Networking Platform with the following entities.
   a) users - table
      id - auto increment, primary key field
      username - varchar (60)
      email - varchar(255)
      address - varchar(150)
      dob - timestamp
      is_active - TINY INT
      registered_on - timestamp
      last_logged_on - timestamp
   b) friends - table_name

id - auto increment, primary key field

user_id - unsigned INT, NOT NULL

friend_name - varchar(60)

c) users profiles

id -

user_id

location

Perform the following operations on above entities.

a. Create table with fields of appropriate datatypes.
b. Verify the table created using DESCRIBE command
c. Insert 10 users and some friendship data in friends table
d. Add a ‘gender’ field of type CHAR(1). Allow NULL values for this field.
e. Rename friends table to users friends
f. Modify the dob field type to date of birth.
g. Remove the field is active
h. Drop the table users profiles

3. Perform the following operations on database created in Ex.no.2 using SELECT command.

i. Fetch the most recent 5 registered users.
ii. Fetch all the friends of user_id user x
iii. Fetch all the users who are above 21 years old.
iv. Find the count of users who signed-up with gmail Id. (ie. users' email ends with @gmail.com)
v. Fetch all the users who registered last month.
vi. Fetch all users of ‘Chennai’ location .
vii. Find actively monthly and weekly users count. ie. Count of users who have logged-in in the last 15 days.
viii. Find how many users who have not mentioned their gender.

4. a) Create a database ' Polytechnic College ' Create 2 users namely ‘Staff’ and ‘student’.

- Grant all privileges to the user 'Staff ‘and grant only ‘create’ privilege to ‘student’ user and verify the same .
- Revoke all privileges to the 2 users and verify the same.

b) Implement the following transaction control statements

i. Commit

ii. Rollback

iii. Save point

5. Create a table 'author' with the following structure

author_id

author name

address

mobile

book title

pages

published on

a) Insert 4 books published by 3 authors each. (12 records)

b) Fetch all the rows and observe how the data duplicated.

c) Apply 1st and 2nd normal forms to fix it.

6. Create table, "mail" with the following fields

t DATETIME, # when message was sent

srcuser VARCHAR (8), # sender (source user and host)

srchost VARCHAR (20),

dstuser VARCHAR (8), # recipient (destination user and host)

dsthost VARCHAR (20),

size BIGINT, # message size in bytes

a) Sort the mail with the largest mail being first.

b) List the mails that is over 25 MB

c) Remove the duplicate rows from result set.

d) Execute a 'SELECT' query and store its result in a user defined variable. Use another 'SELECT' to display the value of the variable.

7. Create two tables with the following structure.

a. Requests table

request_id - UNSIGNED, INT, AUTO INCREMENT, PRIMARY KEY

from_id - INT

to_id - INT

b. requests_log table

request_id - FOREIGN KEY refers to request_id field of requests table request_status - enum("PENDING", "APPROVED", "REJECTED")

8. Create a view combining both tables to display all the requests along with their most recent status for the requests.

9. Create a library Table with proper fields. Create another table called Library1 and insert rows from Library table.

   Hint:

   CREATE TABLE new_table LIKE original_table;

   INSERT INTO new_table SELECT * FROM original_table;

PART – B

1. 9) Create a table to store the details of a customer in a Bank. Do some transactions like withdrawal, deposit. Find the Balance amount(Credit Limit). Based on customer's credit limit, write a program using IF or CASE flow control statements to find the customer levels namely SILVER, GOLD or PLATINUM.
   If the Credit limit is
   - greater than 50K, then the customer level is PLATINUM
   - less than 50K and greater than 10K, then the customer level is GOLD
   - less than 10K, then the customer level is SILVER
2. Create two tables with the following structure.
   a) users - table name
      user_id - UNSIGNED, INT, AUTO INCREMENT, PRIMARY KEY
      username - VARCHAR (60)
      password - VARCHAR (128)
      email - VARCHAR (255)
   b) users_profiles
      user_id - FOREIGN KEY refers to user_id field of user table
      first_name - VARCHAR(60)
      last_name - VARCHAR(60)
      mobile - VARCHAR(15)

   i. SELECT all the users along with their profile details. (Hint: Use INNER JOIN)
   ii. SELECT the users who do not have profiles (Hint: USE LEFT JOIN and exclude the rows generated with NULL values from joining table)
3. Create an employee database and create a stored procedure that accepts employee Id as input and returns complete details of employee as output.
4. Create two tables with the following structure

   Authors
      author_id - INT
      name VARCHAR (60)

titles_count INT -- holds the total number numbers of titles authored

Titles

author_id – INT

Name VARCHAR (512) -- name of the title

a. Create a trigger to update the titles count field of respective row in authors table each time a title gets inserted into titles table.
b. Create a log table with the following structure

author_id - INT

Name VARCHAR (512) -- name of the title

Status VARCHAR(25) --- ADDITION,DELETION,UPDATION and insert an entry in that table each time the tile is added, deleted or updated. Use a trigger to accomplish this.

5. Create a table containing phone number, user name, address of the phone user. Write a function to search the address using phone number.
6. Create a table to store the salary details of the employees in a company. Declare the cursor id to contain employee number, employee name and net salary. Use cursor to update the employee.
7. Create a table 'stock' to contains the itemcode, itemname, current stock, date of last purchase. Write a stored procedure to seek for an item using itemcode and delete it, if the date of last purchase is before one year from the current date. If not, update the current stock.