DATABASE MANAGEMENT SYSTEMS LAB

B.Tech. II Year II Sem.

COURSE OBJECTIVE:
This lab enables the students to practice the concepts learnt in the subject DBMS by developing a database for an example company named "Roadway Trave s" whose description is as follows. The student is expected to practice the designing, deve oping and querying a database in the context of example database "Roadway travels". Students a e expected to use "Mysql" database.

COURSE OUTCOMES:
- Ability to design and implement a database schema for given pr blem.
- Apply the normalization techniques for development f applicati n s ftware to realistic problems.
- Ability to formulate queries using SQL DML/DDL/DCL commands.

Roadway Travels: "Roadway Travels" is in business since 1997 with several buses connecting different places in India. Its main office is located in Hyderabad.

The company wants to *computerize its operations* in the following areas:
- Reservations and Ticketing
- Cancellations

Reservations & Cancellation: Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family). Cancellations are also directly handed at the booking office.

In the process of *computerization* of **Roadway Travels** you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query's using SQL to retrieve the data from the database.

The above process involves many steps like 1. Analyzing the problem and identifying the Entities and Relationships, 2. E-R Model 3. Relational Model 4. Normalization 5. Creating the database 6. Querying. *Students are supposed to work on these steps week wise and fina y create a complete "Database System" to Roadway Travels.* Examples are given at every experiment for guidance to students.

Experiment 1: E-R Model
  nalyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.

Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example:

Entities: 1. BUS   2. Ticket   3. Passenger

Relationships: 1. Reservation   2. Cancellation

Primary Key Attributes: 1. Ticket ID (Ticket Entity)   2. Passport ID (Passenger Entity)

   3. Bus_NO (Bus Entity)

Apart from the above mentioned entities you can identify more. The above mentioned are few.

Note: *The student is required to submit a document by writing the Entities and Keys to the lab teacher.*

Experiment 2: Concept design with E-R Model

zRelate the entities appropriately. Apply cardinalities for each relati nship. Identify strong entities and weak entities (if any). Indicate the type of relationships (t tal / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

Example: E-R diagram for bus
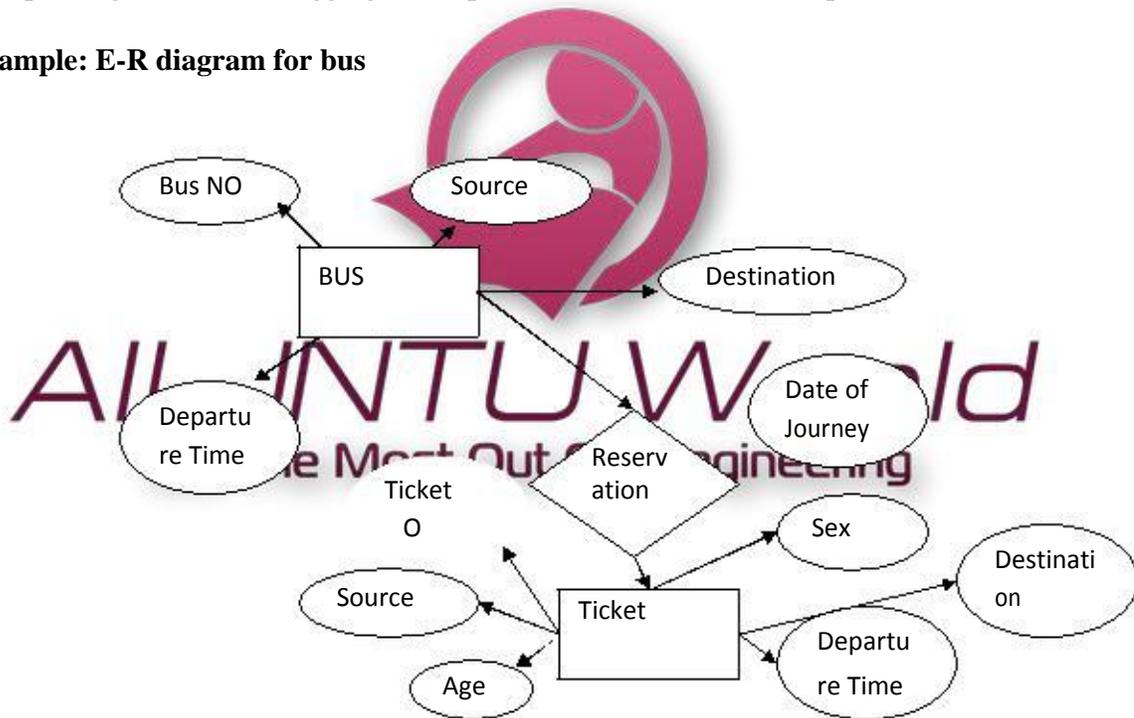
Note: *The student is required to submit a document by drawing the* E-R Diagram *to the* lab teacher.

Experiment 3: Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement.

Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

Example: The passenger tables look as below. This is an example. You can add more attributes based on your E-R model. This is not a normalized table. Passenger

Name	Age	Sex	Address	Passport ID
				Ticket _id

Note: *The student is required to submit a document by Represent relati nships in a tabular fashion to the lab teacher.*

Experiment 4: Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

For the above table in the First normalization we can remove the multi valued attribute Ticket_id and place it in another table along with the primary key of passenger.

First Normal Form: The above table can be divided into two tables as shown below. Passenger

Name	Age	Sex	Address	Passport ID

Passport ID	Ticket_id

You can do the second and third normal forms if required. Any how Normalized tables are given at the end.

Experiment 5: Installation of Mysql and practicing DDL commands
Installation of MySql. In this week you will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. You will also try truncate, rename commands etc.

Example for creation of a normalized "Passenger" table.
CREATE TABLE Passenger (
   Passport_id   INTEGER PRIMARY KEY,
   Name VARCHAR (50) Not NULL,
   Age   Integer Not NULL,
   Sex Char,
   Address VARCHAR (50) Not NULL);

Similarly create all other tables.

Note: Detailed creation of tables is given at the end.

Experiment 6: Practicing DML commands
DML commands are used to for managing data within schema objects. Some examples:
- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

Inserting values into "Bus" table:
Insert into Bus values (1234,'hyderabad', 'tirupathi');
Insert into Bus values (2345,'hyderabd','Banglore');
Insert into Bus values (23,'hyderabd','Kolkata');
Insert into Bus values (45,'Tirupathi,'Banglore');
Insert into Bus values (34,'hyderabd','Chennai');

Inserting values into "Passenger" table:
Insert into Passenger values (1, 45,'ramesh', 45,'M','abc123');
Insert into Passenger values (2, 78,'geetha', 36,'F','abc124');
Insert into Passenger values (45, 90,'ram', 30,'M','abc12');
Insert into Passenger values (67, 89,'ravi', 50,'M','abc14');
Insert into Passenger values (56, 22,'seetha', 32,'F','abc55');

Few more Examples of DML commands:
Select * from Bus; (selects all the attributes and display)
UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;

Experiment 7: Querying

In this week you are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Practice the following Queries:
1. Display unique PNR_no of all passengers.
2. Display all the names of male passengers.
3. Display the ticket numbers and names of all the passengers.
4. Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'.
5. Find the names of passengers whose age is between 30 and 45.
6. Display all the passengers names beginning with 'A'
7. Display the sorted list of passengers names

Experiment 8 and Experiment 9: Querying (continued…)

You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping f Views.
1. Write a Query to display the Information present in the Passenger and cancellation tables. **Hint:** Use UNION Operator.
2. Display the number of days in a week on which the 9 01 bus is available.
3. Find number of tickets booked for each PNR_ no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1. **Hint:** Use   GRO P BY, WHERE and HAVING CLAUSES.
6. Find the total number of cancelled seats.

Experiment 10: Triggers

In this week you are going to work on riggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

Eg: **CREATE TRIGGER updcheck BEFORE UPDATE ON passenger**
   FOR EACH ROW
   BEGIN
   IF NEW.Tickent O > 60 THEN
     SET New.Tickent no = Ticket no;
   ELSE
     SET New.Ticketno = 0;
   END IF;
   END;

Experiment 11: Procedures

In this session you are going to learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

Eg: CREATE PROCEDURE myProc()
BEGIN
SELECT COUNT (Tickets) FROM Ticket WHERE age>=40;
End;

Experiment 12: Cursors

In this week you need to do the following: Declare a cursor that defines a result set.
Open the cursor to establish the result set. Fetch the data into local variables as needed from
the cursor, one row at a time. Close the cursor when done

```
CREATE PROCEDURE myProc(in_customer_id INT)
 BEGIN
 DECLARE v_id  INT;
 DECLARE v_name VARCHAR (30);
 DECLARE c1 CURSOR FOR SELECT stdId,stdFirstname FROM students
    WHERE stdId=in_customer_id;
OPEN c1;
FETCH c1 into v_id, v_name;
Close c1;
END;
```

Tables

BUS

Bus No: Varchar: PK (public key)

Source : Varchar

Destination : Varchar

Passenger

PPNO: Varchar (15)) : PK

Name: Varchar (15)

Age : int (4)

Sex:Char (10) : Male / Female

Address: VarChar (20)

Passenger Tickets

PPNO: Varchar (15)) : PK

Ticket_No: umeric (9)

Reservation

PNR_ No: Numeric (9) : FK

Journey_ date : datetime (8)

No_ of_ seats : int (8)

Address: Varchar (50)

Contact_ No: Numeric (9) --> Should not be less than 9 and Should not accept any other
    character other than Integer

Status: Char (2) : Yes / No

Cancellation

PNR_No: Numeric(9) : FK

Journey_date : datetime(8)

No_of_seats : int (8)

Address : Varchar (50)

Contact_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer

Status: Char (2) : Yes / No

Ticket

Ticket_No: Numeric (9): PK

Journey_date : datetime(8)

Age : int (4)

Sex:Char(10) : Male / Female

Source : Varchar

Destination : Varchar

Dep_time : Varchar

REFERENCE BOOKS:

1. Introduction to SQL, Rick F. Vander Lans, Pearson educati n.
2. Oracle PL/SQL, B. Rosenzweig and E. Silvestrova, Pearson education.
3. SQL & PL/SQL for Oracle 10 g, Black Book, Dr. P. S. Deshpande, Dream Tech.
4. Oracle Database 11 g PL/SQL Programming, M. Mc Laughlin, TMH.