MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2430573 DATABASE MANAGEMENT SYSTEMS LAB

B. Tech. II Year-I Sem	**L / T / P / C**
	0 / 0 / 2 / 1

COURSE OUTCOMES - CO'S

- Demonstrate database creation and manipulation concepts with the help of SQL queries
- Make use of inbuilt functions of SQL queries to perform data aggregations, subqueries, embedded queries and views
- Apply key constraints on database for maintaining integrity and quality of data
- Demonstrate normalization by using referential key constraint.
- Implement PL/SQL programs on procedures, cursors and triggers for enhancing the features of database system to handle exceptions.

LIST OF EXPERIMENTS:

Problem statement:

"Roadway Travels" is in business since 1997 with several buses connecting different places in india. Its main office is located in Hydarabad.

The company wants to computerize its operations in the following areas:

- Reservation and Ticketing
- Cancellations

Reservation & 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 develop query's using SQL to retrieve the data from 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 finally create a complete "Database System" to Roadway Travels. Examples are given at every experiment for guidance to students.

Experiment 1: E-R Model

Analyze the 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 others keys like candidate keys, partial keys, if any.

Example: Entities:

1. BUS
2. Ticket
3. Passenger

Relationships:

1. Reservation
2. Cancellation

PRIMARY KEY ATTRIBUTES:

Ticket ID (Ticket Entity)

Passport ID (Passenger Entity)

Bus_No (Bus Entity)

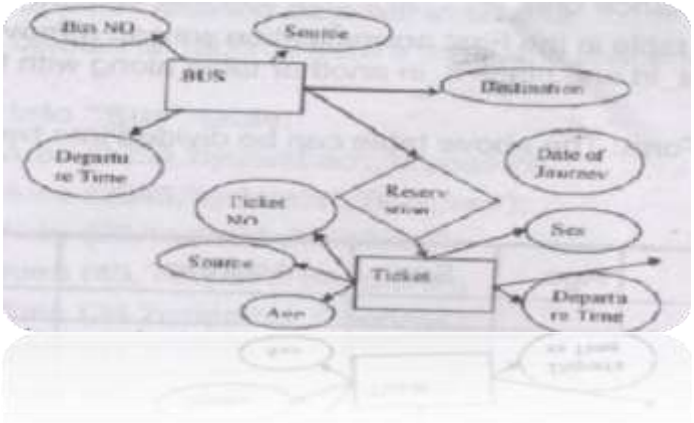
A part from the above mentioned entities you can identify more. The above mentioned are few.

Note: The students is required to submit a document by writing the Entities and keys to the lab teacher.

Experiment 2: Concept design with E-R Model

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

Example: E-R diagram for bus



Note: The students is required to submit a document by drawing the E-R Diagram.

Experiment 3: Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. Thereare different ways of representing relationships as tables based on the requirement. Different types ofattributes (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 E-Rmodel. This is not a normalized table.

Passenger

Name	Age	Sex	Address	Ticket_id	Passport ID

Note: The students is required to submit a document by Represent relationships in a tabular fashion to the labteacher.

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 multiple valued attribute Ticket_id and placeit in another table along with the primary key of passenger.

First Normal Form: The above table can 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 practice DDL commands

Installation of MySQL. In this week you will learn Creating databases, How to create tables, altering thedatabase, droping 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 commandsare used for managing data within schema objects. Some examples:

- SELECT - retrieve data from the 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

insert values into "Bus" table:

```
insert into Bus values (1234, 'hyderabad', 'tirupathi');
```

insert values into "Passenger" table:

```
insert into Passenger values(1, 45, 'ramesh', 45, 'M', 'abc123');
insert into Passenger values(56, 22, 'seetha', 32, 'F', 'abc55');
```

Few more Examples of DML commands:

```
SELECT * FROM Bus; (selects all the attributes and displays)
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, NOTEXIST, UNION, INTERSECT, Constraints etc.

Practice the following Queries:

Display unique PNR_No of all passengers.

Display all the names of male passengers.

Display the ticket numbers and names of all the passengers.

Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'.

Find the names of passengers whose age is between 30 and 45.

Display all the passengers names beginning with 'A'

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, MAX,and MIN),GROUP BY, HAVING and Creation and dropping of VIEWS.

Write a Query to display the information present in the Passenger and cancellation tables. Hint: Use UNION Operator.

Display the number of days in a week on which the 9W01 bus is available.

Find number of tickets booked for each PNR_No using GROUP BY CLAUSE. Hint: Use GROUP BYon PNR_No.

Find the distinct PNR numbers that are present.

Find the number of tickets booked by a passenger where the number of seats is greater than 1. Hint: Use GROUP BY, WHERE and HAVING CLAUSES.

Find the total number of cancelled seats.

Experiment 10: Triggers

. Creation of insert trigger, delete trigger, update trigger.Practice triggers using the above database.

E.g:

```
CREATE TRIGGER updatecheck BEFORE UPDATE ON passenger FOR EACH ROW BEGIN
IF NEW.TickentNO> 60 THEN
SET New.TickentNO = TicketNo; ELSE
SET New.TicketNo = 0;END IF;
 END
```

Experiment 11; Procedures

Creation of stored procedure, Execution of procedure and modificationof procedure. Practice procedures using the database.

E.g:

```
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_nameVARCHAR(30);


   DECLARE c1 CURSOR FOR

   SELECT stdid, stdFirstname FROM studentsss WHERE stdid - in_customer_id;


   OPEN c1;

   FETCH c1 INTO v_id, v_name; CLOSE c1;

   END;
```

Tables:

BUS

Bus No: VARCAHR : PK(primary key) Source: VARCHAR

Destination: VARCHAR

Passenger

PPNO: VARCHAR(15) : PK Name: VARCHAR(15)

Age: INT(4)

Sex: CHAR(10) : Male/FemaleAddress: VARCHAR(20) **Passenger_Tickets**

PPNO: VARCHAR(15) : PK

Ticket_No: NUMERIC(9)

Reservation

PNR_No: NUMERIC(9) : FK

Journey_date: DATETIME(8) No_of_seats: INT(8) Address: VARCHRA(50)

Contact_No: NUMERIC(9) --> Should not less than 9 and Should not accept any other character other than interger

STATUS: CHAR(2) : Yes/No

Cancellation

PNR_No: NUMERIC(9) : FK

Journey_date: DATETIME(8) No_of_seats: INT(8) Address: VARCHRA(50)

Contact_No: NUMERIC(9) --> Should not less than 9 and Should not accept any other character other than interger

STATUS: CHAR(2) : Yes/No

Ticket

Ticket_No: NUMERIC(9) : FK Journey_date: DATETIME(8) Age: INT(4)

Sex: CHAR(10) : Male/Female Source: VARCHAR Destination: VARCHAR Dep_time: VARCHAR