1.YEAR/SEM : II YEAR II SEM

2.LAB CODE/NAME: CS407ES

3.LAB INCHARGE(Lab staff/Teacher): Ms. K.RADHIKA, Mr. G. RAKESH REDDY,

Mr.K. RAKHESH

4.Experiment list:

Experiment 1: E-R Model

Analyze 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

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 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.

Note: *The student is required to submit a document by Represent relationships 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 of 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 9W01 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 GROUP 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 Triggers. 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.TickentNO > 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: Numeric (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

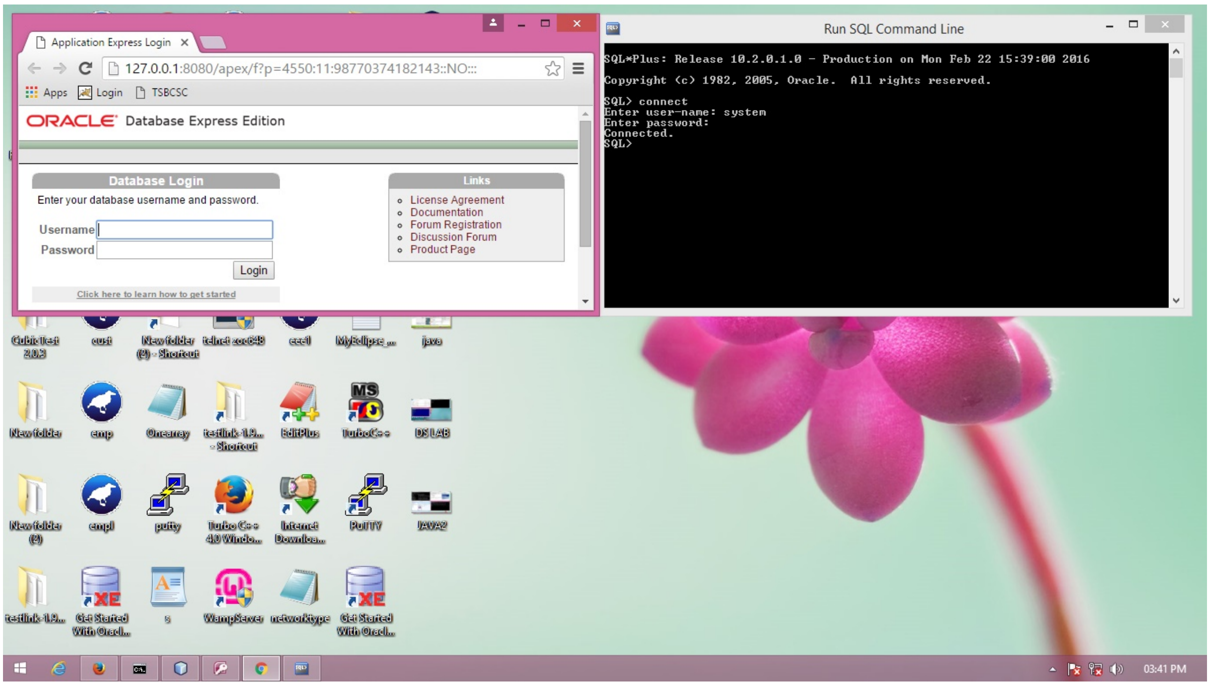
Ticket

Ticket_No: Numeric (9): PK
Journey_date : datetime(8)
Age : int (4)
Sex:Char(10) : Male / Female
Source : Varchar
Destination : Varchar
Dep_time : Varchar

5.

EQUIPMENT	SPECIFICATIONS	PHOTOS
COMPUTER SYSTEM	RAM: 2GB HARD DISK: 500 GB PROCESSOR: DUALCORE @ 2.7 GHZ MOTHER BOARD: INTEL MONITOR: ACER – LED-19 INCHES KEYBOARD: ACER MOUSE: ACER (OPTICAL)	

PHOTOS:



Application Express Login

127.0.0.1:8080/apex/f?p=4550:11:98770374182143::NO:::

ORACLE Database Express Edition

Database Login

Enter your database username and password.

Username

Password

Login

Click here to learn how to get started

Links

- License Agreement
- Documentation
- Forum Registration
- Discussion Forum
- Product Page

Run SQL Command Line

```
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Feb 22 15:39:00 2016

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL>
```