Experiment 1: Student should decide on a case study, analyse and then formulate the problem statement by populating object (entities) and their role.

Example: Entities:

1. BUS
2. Ticket
3. Passenger
4. Reservation
5. Cancellation

PRIMARY KEY ATTRIBUTES:

1. Bus_No (Bus Entity)
2. Ticket ID (Ticket Entity)
3. Passenger ID (Passenger Entity)

Example: Entities:

1. BUS
2. Ticket
3. Passenger

The Following are the entities and its attributes

Bus

Bus_No : varchar(10) **(primary key)**

Source : varchar(20)

Destination : varchar(20)

Passenger

PNR_No : Number(9) **(primary key)**

Ticket_No : Number(9)

Name : varchar(15)

Age : integer(4)

Sex : char(10) ; Male/Female

PPNO : varchar(15)

Reservation

PNR_No : number(9)

Journey_date : date

No_of_seats : integer(8)

Address : varchar(50)

Contact_No : Number(9)

Status : Char(2)

Cancellation

PNR_No : number(9)

Journey_date : date

No_of_seats : integer(8)

Address : varchar(50)

Contact_No : Number(9)

Status : Char(2)

Ticket

Ticket_No : number(9)**(primary key)**

Journey_date : date

Age : int(4)

Sex : Char(10)

Source : varchar

Destination :varchar ,Dep_time : varchar

VIVA Questions:

1.What is an Entity?

An entity is an object that exists. It doesn't have to do anything; it just has to exist. In database administration, an **entity** can be a single thing, person, place, or object.

2.What is an Entity set?

It is a set of entities of same entity type. so a set of one or more entities of Student Entity type is an Entity Set.

3.What is an attribute?

It is a property of an entity. For example, in table STUDENT id,name and Age are properties of an entity of entity type student. Hence these are attributes.

4.**What is Relationship?**

The association among entities is called a relationship.

5.What is Weak Entity set?

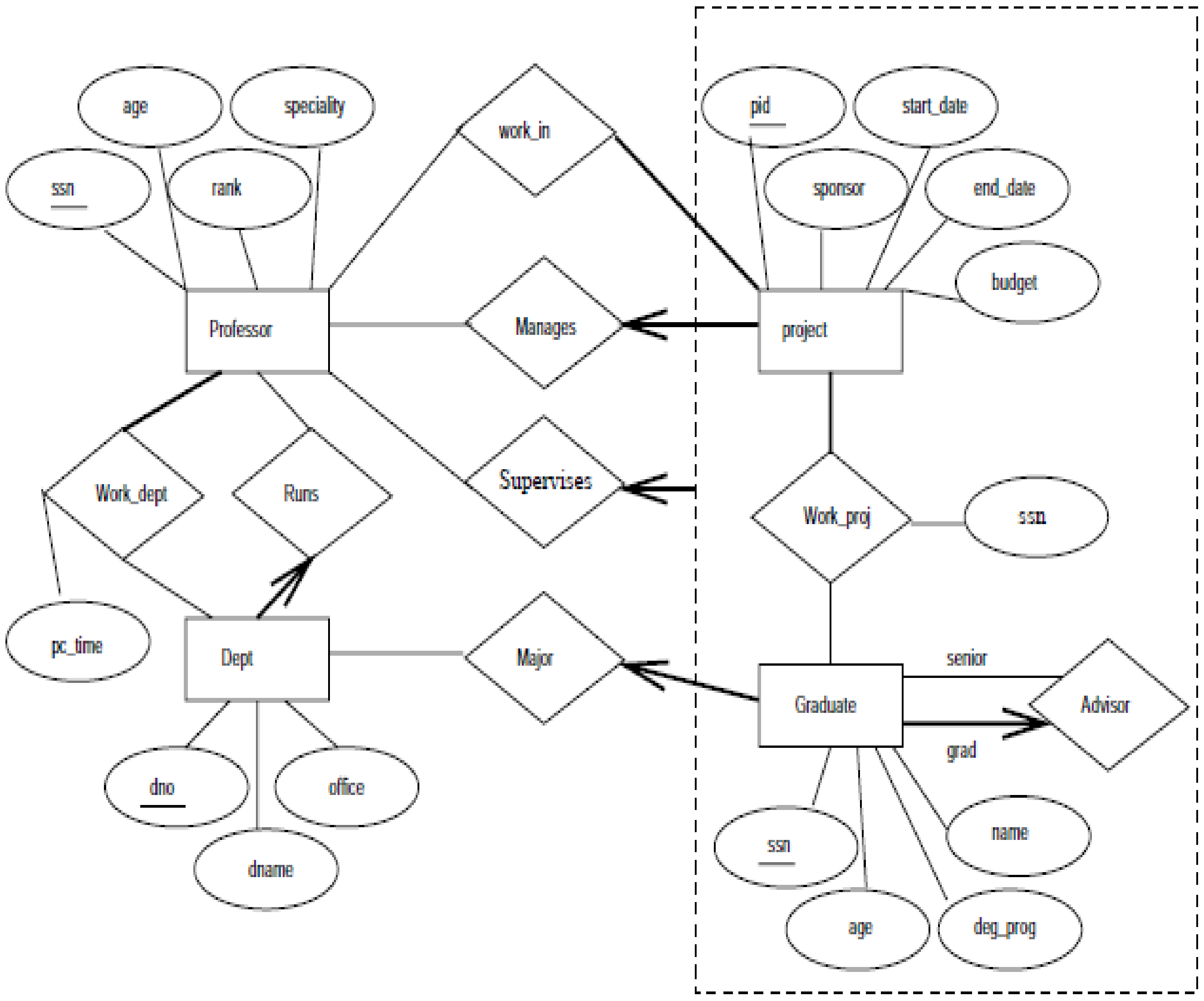
In a relational database, a **weak entity** is an entity that cannot be uniquely identified by its attributes alone; therefore, it must use a foreign key in conjunction with its attributes to create a primary key.

Experiment 2: Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.) Note: Student is required to submit a document by drawing ER Diagram to the Lab teacher.

Consider the following information about a university database:

- Professors have an SSN, a name, an age, a rank, and a research specialty.
- Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget.
- Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.).
- Each project is managed by one professor (known as the project's principal investigator).
- Each project is worked on by one or more professors (known as the project's co-investigators).
- Professors can manage and/or work on multiple projects.
- Each project is worked on by one or more graduate students (known as the project's research assistants).
- When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
- Departments have a department number, a department name, and a main office.
- Departments have a professor (known as the chairman) who runs the department.
- Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.
- Graduate students have one major department in which they are working on their degree.
- Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take.

Design and draw an ER diagram that captures the information about the university. Use only the basic ER model here; that is, entities, relationships, and attributes. Besure to indicate any key and participation constraints.


age
speciality
ssn
rank
Professor
work_in
Manages
Supervises
Work_dept
Runs
pc_time
Dept
dno
office
dname
Major
pid
start_date
sponsor
end_date
budget
project
Work_proj
ssn
Graduate
senior
grad
Advisor
ssn
name
age
deg_prog

VIVA Questions:

1. Define the term derived attribute?

Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database.

2. Define multivalued attribute?

Multi-value attributes may contain more than one values.

3. List different types of cardinalities?

Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

- One to one
- One to many
- Many to one
- Many to many

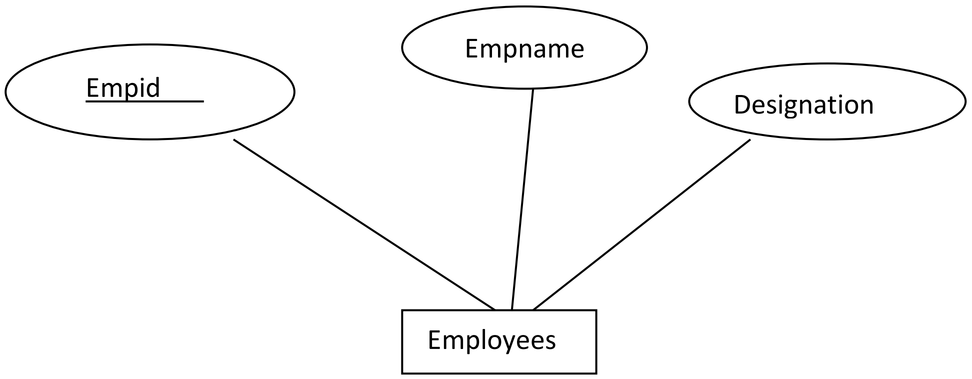
4. What is generalization?

Generalization is the process of extracting common properties from a set of entities and creates a generalized entity from it.

5. What is specialization?

Specialization is the process of extracting values from a set of entities and creates properties from the entity. It is top-down approach.

Experiment 3: Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys) Note: Student is required to submit a document showing the database tables created from ER Model.

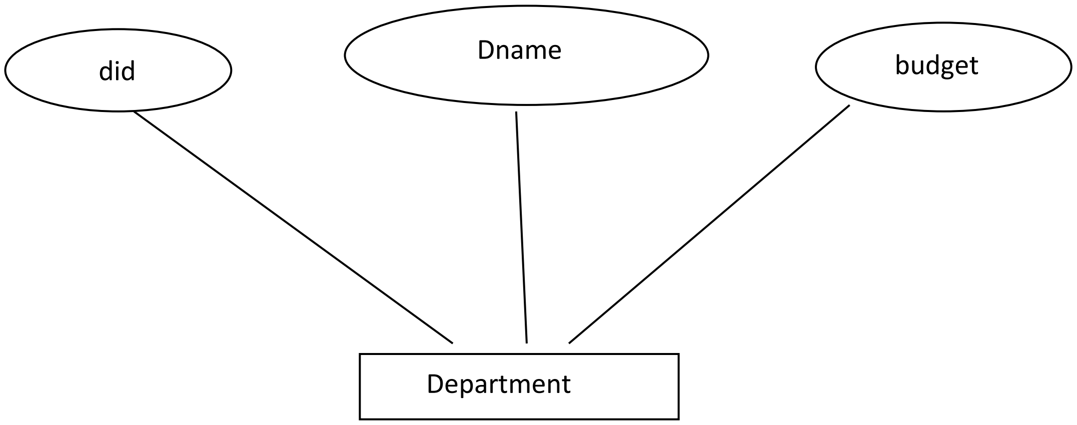


Create table Employees1 (Empid varchar2(10), Empnamechar (20), Designation char (10), primary key(EMPID));

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
EMPID                                     NOT NULL VARCHAR2(10)
EMPNAME                                            CHAR(20)
DESIGNATION                                        CHAR(10)
```

Relational model for Employees1

Empid	Empname	Designation

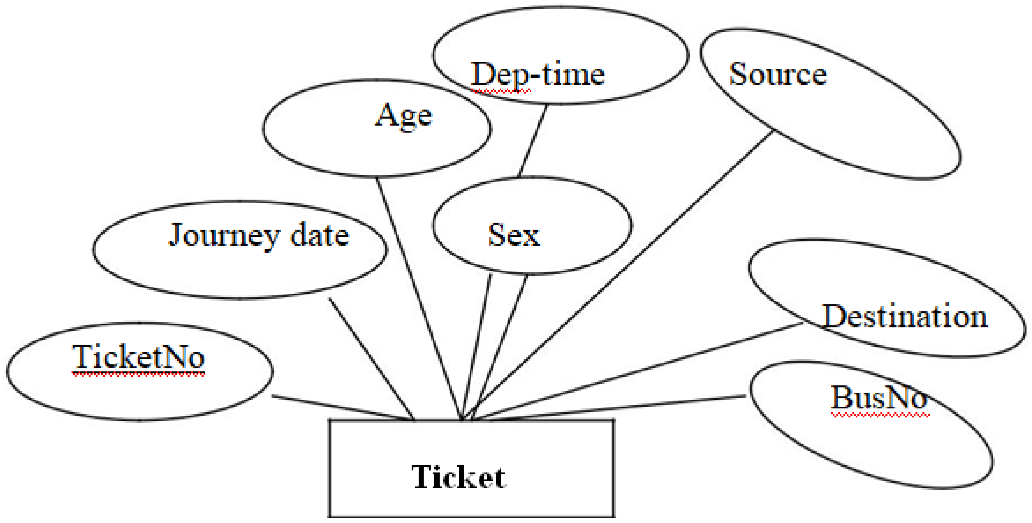


Create table department (did number(5),dname char(10),budget real);

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
DID                                                NUMBER(5)
DNAME                                              CHAR(10)
BUDGET                                             FLOAT(63)
```

Relation model for Department

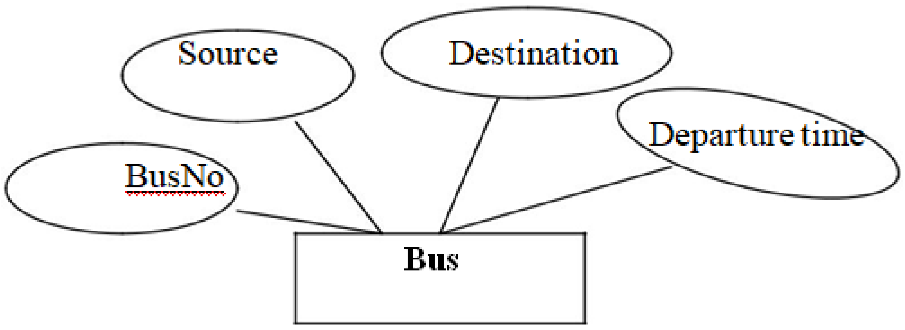
DID	Dname	Budget



create table ticket(ticketno number(10) primary key,journeydate varchar(10),age integer,dept_time varchar(10),sex char(10),source varchar(10),destination varchar(10),busno varchar(10));

Table created.

ticketno	journeydate	age	dept_time	sex	source	destination	busno



SQL> create table bus(busno varchar(10) primary key,source varchar2(10),destination varchar2(10),departuretime varchar(10));

Table created.

Busno	source	destination	departuretime

VIVA Questions:

1.What is Relational Model?

The **relational model** (**RM**) for database management is an approach to managing data using a structure and language consistent with first-order predicate logic, first described in 1969 by Edgar F. Codd, where all data is represented in terms of tuples, grouped into relations. A database organized in terms of the relational model is a relational database.

2.How primary key is represented in a ER model ?

Primary key is represented by underline the attribute name.

3. List different types of constraints?

- Domain integrity
- Entity integrity
- Referal Integrity

4.What is Primary Key Constraint? Explain its uses

A primary key avoids duplication of rows and does not allow null values. It can be defined on one or more columns in a table and is used to uniquely identify each row in a table. These values should never be changed and should never be null.

5. What is foreign key constraint?

A foreign key establishes a relationship between tables by referencing the primary key of another table.

Experiment 4: Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Practicing DDL Commands-Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables

Theory:

OVERVIEW OF SQL:

STRUCTURED QUERY LANGUAGE (SQL) is a commercial relational data base language developed by IBM Corporation at San Jose Research Laboratory, U.S.A. The original name of SQL is SEQUEL. It is a combination of Relational Algebra and Relational calculus. SQL is also referred as Query Language. By using SQL, We can CREATE tables, INSERT values, Maintaining the Database and Providing Security.

RULES for constructing SQL statements:

1) SQL statement must start with a Verb.
2) The Attribute and Data type must be separated by Space.
3) The Attributes must be separated by Comma (,).
4) The SQL statement must be terminated with a semicolon (;).
5) The SQL command must be typed at prompt only i.e., SQL>

SQL DATA TYPES: The commonly used data types in SQL are,

1) **Char (size):** It is used to store character string data [Alpha Numeric]. It is a Fixed Length character string.The size in brackets can determine the maximum number of characters that can hold. The Char data type can hold maximum of 256 characters (i.e. -128 to +128).

2) Varchar (Size) (or) Varchar2 (Size): It is also used to store character string data i.e., alphanumeric data. It is a Variable length character string. Varchar stands for “Character Varying” or “Varied Character”. Varchar data type can hold maximum of 2000 characters/ Bytes. The Maximum length of Varchar2 is 4000 bytes.

3) Int (or) Integer*:* Integer data type stores number data or Numeric data. It does not contain any size in to it. The default size of INT data type is 5.

4) Number (P, S): It is used to store numbers (both Fixed point and Floating point). It stores Zero, Positive, Negative fixed and floating point numbers. Here P means Precision and S means Scale. The Precision determines the maximum length of data i.e., total number of digits in integer part and decimal part. The Scale determines the number of positions in the in the floating part. If the Scale is omitted; it takes the default value as zero. The number data type can store Maximum of 38 digits precision.

5) Date: This data type is used to store (or) represent Date and Time. The ORACLE format of Date is DD_MON_YY. While entering date, we must keep value in single inverted comma (‘ ‘).

6) Long: It is used to store long text character strings of size up to 2GB.

7) Raw/ Long Raw: It is used to store binary data or byte strings i.e. digitized picture (or) image. The maximum length of RAW is 2000 bytes. And the maximum length of long RAW 2000 GB.

8) Real (or) Float (Size): It is used to store floating point number with user specified precision (decimal point) of at least n digits.

Constraint: constraint is a CONDITION. DBMS specifies some conditions. When these conditions are satisfied then only the data is inserted in to database.

Key Constraint: It is a condition that can be applied on a field of the table, that holds UNIQUE values for all its records.

Ex: Reg. Number

Candidate Key: It is a set of fields that uniquely identifies a row/tuple in a relation.

Ex: Reg. Number & E-mail of the student in student table.

Super Key: It is a super set of candidate key. It is a set of fields that each contains a candidate key.

Ex: The set of attributes {reg.number, E-mail} is a super key.

Primary Key: It is a combination of NOT NULL and UNIQUE.

DDL:

It is used to communicate with database. DDL is used to:

- Create an object
- Alter the structure of an object
- To drop the object created.

The commands used are: Create, Alter, Drop, Truncate.

CREATE TABLE

It is used to create a table

Syntax: Create table tablename (column_name1 data_ type constraints, column_name2 data_type constraints …)

Rules:

1. Oracle reserved words cannot be used.
2. Underscore, numerals, letters are allowed but not blank space.
3. Maximum length for the table name is 30 characters.
4. Different tables should not have same name.
5. We should specify a unique column name.
6. We should specify proper data type along with width.
7. We can include “not null” condition when needed. By default it is ‘null’.

Q. Create a table called EMP with the following structure.

Name	Type
EMPNO	NUMBER(6)
ENAME	VARCHAR2(20)
JOB	VARCHAR2(10)
DEPTNO	NUMBER(3)
SAL	NUMBER(7,2)

Allow NULL for all columns except ename and job.

Create table Emp (EmpNo number(6), EName VarChar(20),Job VarChar(10), Deptno number(3),Sal number(7,2));

Q: Create dept table with the following structure.

Name	Type
DEPTNO	NUMBER(2)
DNAME	VARCHAR2(10)
LOC	VARCHAR2(10)

Deptno as the primarykey

Create table dept (deptno number(2) primary key,dname varchar(10),Loc varchar(20));

ALTER TABLE

Alter command is used to:

1. Add a new column.
2. Modify the existing column definition.
3. To include or drop integrity constraint.

Syntax:

alter table tablename add/modify (attribute datatype(size));

Q: Add a column experience to the emp table.

```
SQL> alter table emp1 add(exp varchar(10));

Table altered.

SQL> desc emp1;
 Name                                      Null?    Type
 ----------------------------------------- -------- --------------------
 EMPNO                                              NUMBER(6)
 ENAME                                              VARCHAR2(20)
 JOB                                                VARCHAR2(20)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
 EXP                                                VARCHAR2(10)
```

Q: Modify the column width of the job field of emp table.

```
SQL> alter table emp1 modify(job varchar(30));


Table altered.
SQL> desc emp1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 EMPNO                                              NUMBER(6)
 ENAME                                              VARCHAR2(20)
 JOB                                                VARCHAR2(30)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
 EXP                                                VARCHAR2(10)
```

Q: create the emp1 table with ename and empno, add constraints to check the empno value while entering (i.e) empno > 100.

```
create table emp3(empno number(10),ename char(10),check(empno>100));
```

```
SQL> insert into emp3 values(101,'sravya');

1 row created.

insert into emp3 values(96,'prasanna');

insert into emp3 values(96,'prasanna')

*

ERROR at line 1:

ORA-02290: check constraint (PRASANNA.SYS_C0035541) violated
```

DROP TABLE

It will delete the table structure provided the table should be empty.

Syntax:

```
drop table tablename;
```

Q: Drop any column in the emp table.

```
SQL> alter table emp1 drop column job;

Table altered.

SQL> desc emp1;
 Name                                      Null?    Type
 ----------------------------------------- -------- --------------------
 EMPNO                                              NUMBER(6)
 ENAME                                              VARCHAR2(20)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
 EXP                                                VARCHAR2(10)
```

TRUNCATE TABLE

If there is no further use of records stored in a table and the structure has to be retained then the records alone can be deleted.

Syntax:

TRUNCATE TABLE <TABLE NAME>;

Q7: Truncate the emp table

```
SQL> truncate table emp1;

Table truncated.

SQL> desc emp1;

 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------
 EMPNO                                              NUMBER(6)
 ENAME                                              VARCHAR2(20)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
 EXP                                                VARCHAR2(10)
```

DESC

This is used to view the structure of the table.

Syntax: desc tablename;

Example:

desc emp;

Q1: Write a query to describe the structure department table.

```
SQL> DESC DEPT;
 Name                          Null?    Type
      ----------------------------------------- -------- ----
      DEPTNO    NOT NULL NUMBER(6)
      DNAME                             CHAR(6)
```

VIVA Questions:

1. Define the term DDL?

Data base schema is specified by a set of definitions expressed by a special language called a data definition language.

2. What are the categories of SQL command?

SQL commands are divided in to the following categories:
Data Delimitation language
Data manipulation language
Data control language
Transaction Control Language

3. What is difference between Truncate and Delete?

i) TRUNCATE TABLE removes the data by deallocating the data pages used to store the table data and records only the page deallocations in the transaction log.
ii) The DELETE command is used to remove rows from a table

4. Listout the datatypes of SQL?

- char(size)
- varchar(size)
- integer
- number(p,s)
- date
- long
- raw/long raw
- real (or) float(size)

5.What is difference between Char and Varchar2? Explain its uses

The length is fixed and indicates the number of characters declared when a table is created. It can be any value from **0** to **255 bytes**.The length is variable, but the maximum is specified when creating a table. Maximum lengths can range from **0** to **255 bytes** (before MySQL 5.0.3) or from 0 to **65,535 bytes** in later versions.

Experiment 5: Practicing DML commands-Insert, Select, Update, delete of tables.

Theory:

DML COMMAND

DML commands are the most frequently used SQL commands and is used to query and manipulate the existing database objects. Some of the commands are Insert, Select, Update, Delete

Insert Command

This is used to add one or more rows to a table. The values are separated by commas and the data types char and date are enclosed in apostrophes. The values must be entered in the same order as they are defined.

Inserting a single row into a table:
Syntax: insert into <table name> values (value list)

Inserting more than one record using a single insert commands:
Syntax: insert into <table name> values (&col1, &col2, ….)

Skipping the fields while inserting:
Syntax:Insert into <tablename(coln names to which data to be inserted)> values (list of values).

Q1: Insert a single record into dept table.

```
SQL> insert into dept values (1,'IT','Tholudur');
```

Q2: Insert more than a record into emp table using a single insert command.

```
SQL> insert into emp values(&empno,'&ename','&job',&deptno,&sal);

Enter value for empno: 1

Enter value for ename: Mathi

Enter value for job: AP


Enter value for deptno: 1

Enter value for sal: 10000

old  1: insert into emp values(&empno,'&ename','&job',&deptno,&sal)
```

```
new  1: insert into emp values(1,'Mathi','AP',1,10000)
1 row created.

SQL> /

Enter value for empno: 2

Enter value for ename: Arjun

Enter value for job: ASP

Enter value for deptno: 2

Enter value for sal: 12000

old  1: insert into emp values(&empno,'&ename','&job',&deptno,&sal)

new  1: insert into emp values(2,'Arjun','ASP',2,12000)

1 row created.

SQL> /

Enter value for empno: 3

Enter value for ename: Gugan

Enter value for job: ASP

Enter value for deptno: 1

Enter value for sal: 12000

old  1: insert into emp values(&empno,'&ename','&job',&deptno,&sal)

new  1: insert into emp values(3,'Gugan','ASP',1,12000)

1 row created
```

Select command

It is used to retrieve information from the table. it is generally referred to as querying the table.we can either display all columns in the table or only specify the column in the table.

Selects all rows from the table

Syntax: Select * from tablename;

The retrieval of specific columns from a table:

It retrieves the specified columns from the table

Syntax: Select column_name1, …..,column_namen from table name;

Elimination of duplicates from the select clause:

It prevents retriving the duplicated values .Distinct keyword is to be used.

Syntax: Select DISTINCT col1, col2 from table name;

Select command with where clause:

To select specific rows from a table we include 'where' clause in the select command. It can appear only after the 'from' clause.

Syntax: Select column_name1, …..,column _name from table name where condition;

Select command with order by clause:

Syntax: Select column_name1, …..,column _name from table name where condition
order by colmn_name;

Select command to create a table:

Syntax: create table table_name as select * from existing_tablename;

Select command to insert records:

Syntax: insert into table name (select columns from existing_tablename);

Elimination of duplicates from the select clause:
It prevents retrieving the duplicated values.Distinct keyword is to be used.

Syntax: Select DISTINCT col1, col2 from table name;

Q: select employee name, job from the emp table

```
SQL> select ename, job from emp;
ENAME              JOB
-------------------- --------------------
ashok               manager
kumar              clerk
anil                asst manager
```

Q: Create a pseudo table employee with the same structure as the table emp and insert rows into the table using select clauses.

```
SQL>  create table emp12  as select * from emp1where 1=2;
Table created.
SQL> insert into emp12(select * from emp1);
5 rows created.
```

Q: Display only those employees whose deptno is 30.

```
SQL>select * from emp where deptno=30;
EMPNO ENAME   DEPTNO  SAL   JOB
---------- -------------------- -------------------- ----------
11  ashok   30       50000  manager
```

Q: Display deptno from the table employee avoiding the duplicated values.

```
SQL> select distinct deptno from emp;

DEPTNO
------
  30
 102
 103
```

Q: List the records in the emp table order by salary in descending order.

```
SQL> select * from emp order by salary desc;

EMPNO ENAME  DEPTNO   SAL  JOB
----- ------------------- ------------------- ---------- ----------
  11     ashok      30    50000    manager
  13     anil      103    30000    asstmanager
  12    kumar      102    15000    clerk
```

Q: List the records in the emp table order by salary in ascending order.

```
SQL> select * from emp order by salary;

EMPNO ENAME  DEPTNO    SAL     JOB
----- ------------------- ------------------- ----------
  12      kumar     102       15000   clerk
  13      anil       103       30000   asst manager
  11      ashok      30        50000   manager
```

UPDATE COMMAND

It is used to alter the column values in a table.A single column may be updated or more than one columns can be updated.

Syntax:updatetable_name set field=values where condition

Q: Update the emp table to set the salary of all employees to Rs15000/- who are working as manager.

```
SQL> update emp set sal=15000 where job='manager';

EMPNO ENAME  DEPTNO  SAL   JOB
---------- -------------------- -------------   -------
    11        ashok         30      15000   manager
    12        kumar        102     15000   clerk
    13        anil           103     30000   asst manager
```

DELETE COMMAND

After inserting a row in the table we can also delete row in the table if required. The delete command consists of a from clause followed by an optional where clause.

Syntax: Delete from table where conditions;

Q: Delete only those who are working as lecturer

```
SQL> delete from emp where job='lecturer';
1row deleted
SQL> select * from emp;

EMPID     ENAME  DEPTID  JOB          SALARY
----------   -------------  -----------  --------      ---------------

566          Bavya       1234        Teacher         35000
568          Navya      1237Accountant    60000
```

VIVA:

1. What is DML?

DML commands are the most frequently used SQL commands and is used to query and manipulate the existing database objects.

2. What are DML commands? Give the general form of SQL Queries?

Some of the commands are Insert, Select, Update, Delete.

Select A1, A2…………, An From R,1R2……………,
R m Where P

3. What is the use of rename operation?
Rename operation is used to rename both relations and an attributes. It uses the as clause, taking the form: Old-name as new-name.

4. Define tuple variable?

Tuple variables are used for comparing two tuples in the same relation. The tuple variables are defined in the from clause by way of the as clause.

5. Write the syntax to retrieve specific columns from a table?

Syntax: Select column_name1, …..,column_namen from table name;

Experiment 6: Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, EXCEPT, CONSTRAINTS etc

Sailors(sid:number,sname:char,rating:number,age:real)

Boats (bid: number.bname: char,color:char)

Reserves (Sid: number, bid: number, day: date)

Sailors Table

Sid	Sname	Rating	Age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
85	Art	3	25.5
95	Bob	3	63.5

Reserves Table

Sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Boats

Bid	Bname	color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	red

Create the above tables using DDL and DML commands and perform the following Queries

1)Find sailors whose rating is better than some sailor called Horatio. (Any)

```
SQL>select s.sid from sailors s where s.rating >any(select s2,rating from sailors s2 where
 s2.sname='horatio');
```

Output:

```
SID
-------
58
71

74

31

32
```

2)Find the sailors with the highest rating. (All)

```
SQL>select s.sid from sailors s where s.rating >=all(select s2.rating from sailors s2);
```

Output:

```
SID
----
58
71
```

3) Find the names of sailors who have reserved both a red and a green boat. (In)

```
SQL>select s.name from sailors s, reserves r, boats b, where s.sid=r.sid and r.bid=b.bid and

b.colour='red' and s.sname in(select s2.sname from sailors s2,boats b2,reserves r2,where
 s2.sid=r2.sid and r2.bid=b2.bid and b2.color='green';
```

Output:

```
SNAME
----------
dustin

dustin

lubber

lubber

horatio
```

4) Find the names of sailors who have reserved boat 103.(Exists)

```
SQL>select s.name from sailors s where exists(select * from reserves r where r.bid=103 and
r.sid=s.sid);
```

Output:

```
SNAME
----------
Dustin

Lubber

horatio
```

5) Find the names of sailors who have reserved all boats. (Not exists)

```
SQL>select s.sname from sailors s where not exits(select b.bid from boats b where not
exist(select r.bid from reserves r where r.bid=b.bid and r.sid=s.sid));
```

Output:

```
SNAME:
----------
Dustin
```

6) Find the names of sailors who have reserved a redand a green boat. (Union)

```
SQL>select s.sname from sailors s, reserves r ,boats b where s.sid=r.sid and r.bid=b.bid and
b.color='red'

Union

select s2.sname from sailors s2, reserves r2 ,boats b2 where s2.sid=r2.sid and r2.bid=b2.bid and
b2.color='green' ;
```

Output:

```
SNAME:
----------
Dustin

Horatio

Lubber
```

7)Find the sids of all sailors who have reserved red boats but not green boats.(Except)

```
SQL>select r.sid from reserves r ,boats b where r.bid=b.bid and  and b.color='red'

except

select r2.sid from reserves r2 ,boats b2 where  r2.bid=b2.bid and b2.color='green' ;
```

Output:

```
SID:
-----
22

31

64
```

8) Find the names of sailors who have reserved botha redand a green boat. (Intersect)

```
SQL>select s.sname from sailors s, reserves r ,boats b where s.sid=r.sid and r.bid=b.bid and
 b.color='red'

Intersect

Select  s2.sname from sailors s2, reserves r2 ,boats b2 where s2.sid=r2.sid and r2.bid=b2.bid and
 b2.color='green' ;
```

Output:

```
SNAME:
----------
Dustin

Horatio

Lubber
```

9) Find the names of sailors who have reserved a red and a green boat. (Union All)

```
SQL>select s.sname from sailors s,reserves r,boats b where s.sid=r.sid and r.bid=b.bid and
 b.color=’red’

Union all

Select s2.sname from sailors s2,boats b2,reserves r2 where s2.sid=r2.sid and r2.bid=b2.bid and
 b2.color=’green’;
```

Output:

```
SNAME:
---------
Dustin

Dustin

Lubber

Lubber

Horatio

Dustin

Lubber

Horatio
```

VIVA Questions:

1. What is difference between union and union all?

UNION combines the result set of two or more queries into a single result set. This result set includes all the rows that belong to all queries in the UNION. UNION ALL is very similar to UNION. It also includes duplicate rows in the result set.

2. What are SET operators and list out all?

Different types of SET operations, along with example:

1. UNION
2. UNION ALL
3. INTERSECT
4. MINUS

3. What is the use of Existsoperation?

The EXISTS operator is used to test for the existence of any record in a subquery.The EXISTS operator returns true if the subquery returns one or more records.

4. Definethe condition for union operations on two tables?

- The same number of columns selected
- The same number of column expressions
- The same data type and
- Have them in the same order

5.Define the use of using Any and All operators on queries?

The ALL operator is used to compare a value to all values in another value set. The ANY operator is used to compare a value to any applicable value in the list as per the condition.

Experiment 7: Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).

Sailors(sid:number,sname:char,rating:number,age:real)

Boats (bid: number.bname: char, color:char)

Reserves (Sid: number, bid: number, day: date)

Sailors Table

Sid	Sname	Rating	Age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorbas	10	16.0
85	Art	3	25.5
95	Bob	3	63.5

Reserves Table

Sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Boats

Bid	Bname	color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	red

Create the above tables using DDL and DML commands and perform the following Queries

1)Find the names of sailors who have reserved red boat (Nested query)

```
SQL>select s.sname from sailors s where s.sid IN(select r.sid from reserves r where r.bid  IN(select
b.bid from boats b where b.color='red');
```

Output:

```
SNAME:
----------
Dustin

Lubber

Horatio
```

2)Find the names of sailors who have reserved boat number 103(Correlated Nested query)

SQL>select s.sname from sailors s where exist(select * from reserves r where r.bid=103 and r.sid=s.sid**);**

Output:

```
SNAME:
----------
Dustin
Lubber
horatio
```

The SQL **Joins** clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each.

CUSTOMER TABLE

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

ORDER TABLE

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

There are different types of joins available in SQL

- INNER JOIN − returns rows when there is a match in both tables.
- LEFT JOIN − returns all rows from the left table, even if there are no matches in the right table.
- RIGHT JOIN − returns all rows from the right table, even if there are no matches in the left table.
- FULL JOIN − returns rows when there is a match in one of the tables.
- SELF JOIN − is used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.
- CARTESIAN JOIN − returns the Cartesian product of the sets of records from the two or more joined tables.

Inner Join

SELECT columns FROM table1 INNER JOIN table2 ON table1.column = table2.column;

Example:Write a query to perform inner join between any two tables

SQL>select id,name,amount,day from customer inner join order1 on customer.id=order1.customer_id;

```
        ID NAME           AMOUNT DAY
---------- ---------- ---------- ---------
         2 khilan           1560 20-NOV-09
         3 kaushik          1500 08-OCT-09
         3 kaushik          3000 08-OCT-09
         4 chaitali         2060 20-MAY-08
```

Left Outer Join

SELECT columns FROM table1 LEFT [OUTER] JOIN table2 ON table1.column = table2.column;

Example:Write a query to perform Left outer join between any two tables.

SQL>select id,name,amount,day from customer left join order1 on customer.id=order1.customer_id

```
 ID NAME           AMOUNT DAY
--- ---------- ---------- ---------
  3 kaushik          3000 08-OCT-09
  3 kaushik          1500 08-OCT-09
  2 khilan           1560 20-NOV-09
  4 chaitali         2060 20-MAY-08
  5 hardik
  1 ramesh
  6 komal
  7 muffy
```

Right Outer Join

SELECT columns FROM table1 RIGHT [OUTER] JOIN table2 ON table1.column = table2.column;

Example:Write a query to perform Right outer join between any two tables.

```
SQL>select id,name,amount,day from customer right join order1 on
customer.id=order1.customer_id

  ID NAME           AMOUNT DAY
---- ---------- ---------- ---------
   2 khilan           1560 20-NOV-09
   3 kaushik          1500 08-OCT-09
   3 kaushik          3000 08-OCT-09
   4 chaitali         2060 20-MAY-08
```

Equijoin
SELECT column_list FROM table1, table2.... WHERE table1.column_name = table2.column_name;

Example:Write a query to perform Equijoin between any two tables.

```
SQL>select id,name,amount,day from customer, order1 where customer.id=order1.customer_id

  ID NAME           AMOUNT DAY
---- ---------- ---------- ---------
   2 khilan           1560 20-NOV-09
   3 kaushik          1500 08-OCT-09
   3 kaushik          3000 08-OCT-09
   4 chaitali         2060 20-MAY-08
```

VIVA Questions:

1. What is Nested Query?

Nesting of queries one within another is known as a nestedqueries.

2. What are Joins? Give the general form of joining two tables?

The purpose of a join concept is to combine data spread across tables. A join isactually performed by the ‘where’ clause which combines specified rows of tables.

3. What is the difference between left outer join and right outer join?

It extends the result of a simple join. An outer join returns all the rowsreturned by simple join as well as those rows from one table that do not match anyrow from the table. The symbol (+) represents outer join.Inner join returns the matching rows from the tables that are beingjoined.

4. Define Correlated Nested query?

A sub query is evaluated once for the entire parent statement whereas a correlated Sub query is evaluated once per row processed by the parent statement.

5. Write the syntax for Equijoin of two tables?

Select column_list from table1,table2…. Where

table1.column_name=table2.column_name2

Experiment 8: Practice Queries using Aggregate Operators - COUNT, SUM, AVG, MAX, MIN. GROUP BY, HAVING, VIEWS Creation and Dropping.

Sailors(sid:number,sname:char,rating:number,age:real)

Boats (bid: number.bname: char, color:char)

Reserves (Sid: number, bid: number, day: date)

Sailors Table

Sid	Sname	Rating	Age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorbas	10	16.0
85	Art	3	25.5
95	Bob	3	63.5

Reserves Table

Sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Boats

Bid	Bname	color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	red

1) Find the average age of sailors with a rating of 10?

```
SQL>select avg( s.age) from sailors s where s.rating=10;

AVG(AGE)

----------

     25.5
```

2)Find the name and age of the oldest sailor?

```
SQL> select s.sname,s.age from sailors s where s.age=(select max(s.age) from sailors s);



SNAME          AGE

----------         ----------

bobby          63.5
```

3) Write the query to Count the number of different sailor names?

```
SQL>select count (distinct s.sname) from sailors s;

COUNT (DISTINCTS.SNAME)

---------------------------------

              9
```

4)Find the age of the youngest sailor for each rating level?

```
SQL> select s.rating,min(s.age) from sailors s group by s.rating;


 RATING MIN(S.AGE)

----------    --------------

      1        33

      8       25.5

      7        35

      3       25.5

     10        16

      9        35
```

5)Find the sum of ages of sailors whose rating is above 10?

```
SQL> select sum(age) from sailors where rating>10;

 SUM(AGE)

----------

 no rows selected
```

because greater than 10 no row data in the database

6)Find the average age of sailors for each rating level that has at least two sailors?(group by and Having)

```
SQL> select s.rating,avg(s.age) as average from sailors s group by s.rating  having count(*)>1;


   RATING    AVERAGE
-----------------------
        8      40.5
        7        40
        3     44.5
       10      25.5
```

A VIEW in SQL is a logical subset of data from one or more tables. View is used to restrict data access.

To create a view the syntax is**:**

CREATE or REPLACE VIEW view_name AS SELECT column_name(s) FROM table_name WHERE condition

Example:Write a query to create a view on a table

```
SQL>create view v11 as select sid,age from sailors where rating>7;
View created.
```

```
SQL> select * from v11;

      SID       AGE
---------- ----------
        31       55.5
        32       25.5
        58         35
        71         16
```

```
    74      35
```

To drop a view Syntaxis**: Drop View View_ name**

Example: Write a query to drop the existing view

```
SQL> drop view v11;

      View dropped.
```

VIVA Questions:

1. What is view?

A view is a logical table based on a table or another view. A view contains no data of its own but is like a window through which data from tables can be viewed or changed.

2. What are aggregate operators and list out all?

- Count
- Max
- Min
- Sum
- avg

3. What is the use of count() and count(*)?

The SQL COUNT() function returns the number of rows in a table satisfying the criteria specified in the WHERE clause. It sets the number of rows or non NULL column values.

4. Define the use of group by and Having clause?

The **HAVING Clause** enables you to specify conditions that filter which group results appear in the results. Expressions that are not **encapsulated** within an aggregate function and must be included in the GROUP BY Clause at the end of the SQL statement.

5. Write the syntax to retrieve specific columns from a table using aggregate operators?

```
SELECT aggregateoperator(column_name)

FROM table_name

WHERE condition;
```

Experiment 9: Practicing on Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger

AIM: Creation of Triggers, and perform insert, update, delete triggers.

Definition: Triggers are stored programs, which are automatically executed or fired when some events occur.

Triggers are occurred when we perform any of the following events:

1.A database manipulation (DML) statements (DELETE, INSERT, UPDATE).
2.A database definition (DDL) statements (CREATE, ALTER, And DROP).
3.A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers could be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers:

Triggers can be written for the following purposes:

1. Generating some derived column values automatically.
2. Enforcing referential integrity.
3. Event logging and storing information on table access.
4. Auditing.
5. Synchronous replication of tables.
6. Imposing security authorizations.
7. Preventing invalid transactions.

Syntax:

Create or replace trigger <trigger name> (before/after/instead of} {insert/delete/update} on <table name> [for each statement/row] [when <condition>]

Declare

<Declarations>

Begin

<Executable statements>

End;

SQL>set serverOutput on;

Example:

```
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
```

Write a PL/SQL code for creation of trigger to insert data into a table.

```
SQL>create or replace trigger t1
before
insert on sailors
for each row
begin
:new.sname:=upper(:new.sname);
end;
/Trigger created
```

Output:

```
insert into sailors values(22,’dustin’);
select * from sailors;



SID      SNAME
----    ---------
22       DUSTIN
```

Write a PL/SQL code for creation of trigger to update data into a table.

```
SQL>create trigger t2
```

```
after update of sid on sailors

for each row

begin

if(:new.sid<80) then

raise_application_error(-20017,'cant update');

end if;

end;

/

Trigger created
```

Output:

```
SQL> update sailors set sid=79 where rating=10;

update sailors set sid=95 where rating=10;
       *
ERROR at line 1:
ORA-20018: you cant update  this row
ORA-06512: at "PRASANNA.TRG3", line 3
ORA-04088: error during execution of trigger
 'PRASANNA.TRG3'
```

Write a PL/SQL code for creation of trigger to delete data from a table.

```
SQL>create trigger t6

after

delete on sailors

for each row
```

```
begin
if :old.sid=22 then
raise_application_error
(-20019,'you can't delete this row');
end if;
end;
/
Trigger created
```

Output:

```
SQL> delete from sailors where sid=22;
delete from sailors where sid=22
            *
ERROR at line 1:
ORA-20010: you cant delete this row
ORA-06512: at "PRASANNA.TRG2", line 3
ORA-04088: error during execution of trigger
 'PRASANNA.TRG2'
```

VIVA Questions:

1. What is trigger?

Triggers are statements that are executed automatically by the system as the side effect of a modification to the database. The triggers can be initiated before the event or after the event.

2. What kind of operations can be performed using trigger?

BEFORE, INSTEAD OF, AFTER, and CONFLICT

3. What is the difference between before update and after update?

All the code written in the "before update" triggers, executes BEFORE that DML is committed. After update trigger generally works when you want to update any other object.

4.How many triggers are possible per table?

12

5.When multiple after triggers are attached to sql table, how to control the order of execution?

Using SV_SET trigger order procedure is used for the execution of multiple triggers.

Experiment 10: Procedures- Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure.

AIM: Creation of Stored procedure, and execution of Procedure and Modification of Procedure.

<u>Definition:</u> A Procedure is a module that performs one or More Actions. It does not return any values.

Syntax:

Create or replace procedure <procedure name> [(parameter1, …] AS

<Local Declarations>

Begin

<Executable statements>

End;

Write a PL/SQL code for creation of procedure to view some specified columns from a table.

```
SQL>create or replace procedure p_sail(sid1 in number)
is
v_sname varchar(10);
v_age number(10);
begin
select sname,age into v_sname,v_age from sailors where sid=sid1;
dbms_output.put_line('sname:'||v_sname);
dbms_output.put_line('age:'||v_age);
end;
/
procedure created
```

Output:

```
sql>execute p_sail(22);
sname:dustin
```

```
age:45

procedure executed successfully
```

Write a PL/SQL code for modification of a procedure on specified columns from a table.

```
SQL>create or replace procedure p_sailors2(
v_sid1 in sailors.sid%type,
v_sname in sailors.sname%type,
v_age in sailors.age%type)is
  begin
  update set sname=v_sname,age=v_age where sid=v_sid;
commit;
    end;
SQL> /
Procedure created.
SQL> execute p_sailors(22,balu,28);
PL/SQL procedure successfully completed.
```

VIVA Questions:

1. What is a stored procedure?

Stored procedures in SQL Server can accept input parameters and return multiple values of output parameters; in SQL Server, stored procedures program statements to perform operations in the database and return a status value to a calling procedure or batch.

2. What are the advantages of using stored procedures?

- **Maintainability**
- **Testing**
- **Speed / Optimization**

3.How you will execute stored procedures as a different user?

Execute as user='special user'
Execute procedure name

4.Can you return the Null values using stored procedures?

No

5. Where the stored procedures are stored in database?
A stored procedure is sub routine available to application accessing a relational database system.

It is actually stored in the database data dictionary.

Experiment 11: Cursors- Declaring Cursor, Opening Cursor, Fetching the data, closing the cursor.

A cursor is a temporary work area created in the system memory when a SQL statement is executed. A cursor contains information on a select statement and the rows of data accessed by it.

This temporary work area is used to store the data retrieved from the database, and manipulate this data. A cursor can hold more than one row, but can process only one row at a time. The set of rows the cursor holds is called the *active* set.

There are two types of cursors in PL/SQL:

Implicit cursors

These are created by default when DML statements like, INSERT, UPDATE, and DELETE statements are executed. They are also created when a SELECT statement that returns just one row is executed.

Explicit cursors

They must be created when you are executing a SELECT statement that returns more than one row. Even though the cursor stores multiple records, only one record can be processed at a time, which is called as current row. When you fetch a row the current row position moves to next row.

Both implicit and explicit cursors have the same functionality, but they differ in the way they are accessed.

Cursor Operations

Create cursor

Eg:cursor Emp is select empid,name,salary from employee where salary>15000;

Open cursor

Execute the query and put the pointer at the first tuple.

Eg:open Emp;

Fetch next Tuple

Pointer moves automatically when a tuple is fetched

Eg:Fetch Emp into <variable>;

Close Cursor

Open cursor is closed

Eg:Close Emp;

Attributes used in the cursor programs:

C1%ROWCOUNT: The number of tuples in C1(C1 is cursor name)

C1%FOUND: True if the last fetch was successful

C1%NOTFOUND: True if the last fetch was not successful

C1%ISOPEN: True if C1 is open

Example of cursor program

```
Declare c is select * from emp_information where emp_no<=2;
tmp emp_information%rowtype;
Begin
Open c;
for tmp in c loop
Fetch c into tmp;
Dbms_output.put_line(‘EMP_no:    ‘||tmp.emp_no);
Dbms_output.put_line(‘EMP_name:    ‘||tmp.emp_name);
Dbms_output.put_line(‘EMP_dept:    ‘||tmp.emp_dept);
Dbms_output.put_line(‘EMP_salary:    ‘||tmp.emp_salary);
End loop;
Close c;
End;
/
```

Output:

```
EMP_NO      EMP_NAME    EMP_DEPT     EMP_SALARY
------  ---------   ----------    ---------
1       RAMU         SE           25000
2       JOHN        ME           28000
```

AIM: Write a PL/SQL program that uses all cursor operation on any data base .

```
SQL>declare
   v_sname sailors.sname%type;
 v_age sailors.age%type;
  cursor c2 is select sname,age from sailors;
 begin
 open c2;
  loop
 fetch c2 into v_sname,v_age;
  exit
 when c2%rowcount>3;
  dbms_output.put_line(v_sname|| ' '||v_age);
 end loop;
 close c2;
  end;
/
```

Output:

```
SNAME    AGE

Xyz 45

Brutus33

lubber 55.5
```

VIVA:

1. Define the concept of cursor?

A cursor is a temporary work area created in the system memory when a SQL statement is executed. A cursor contains information on a select statement and the rows of data accessed by it.

2. What are the different types of cursors?

- Implicit
- Explicit

3. Why does %ISOPEN return false for an implicit cursor?

Implicit cursors: SQL%ISOPEN always returns FALSE, indicating that the implicit cursor has been closed.

4. What are the differences between Implicit and Explicit Cursors?

An implicit cursor is one created "automatically" for you by Oracle when you execute a query. An explicit cursor is one you create yourself. It takes more code, but gives more control

5.State the differences between cursor and Procedures**?**

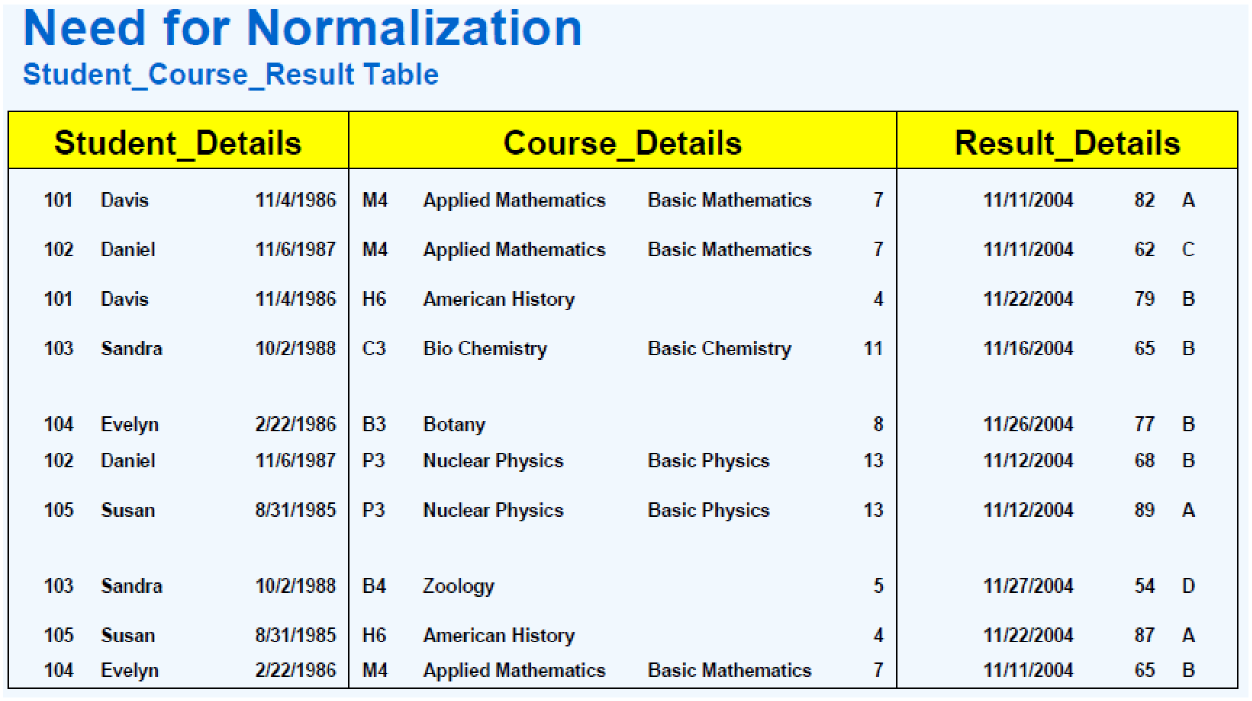
A cursor basically is a place to hold the results from a query. A cursor allows you to transverse the result set row by row. A stored procedured is a named bit of saved code that can be run from the database.

Experiment 12: Normalization -To remove the redundancies and anomalies in the above relational tables, Normalize upto Third Normal Form.

First Normal form:1NF

- A relation schema is in 1NF:
- If and only if all the attributes of the relation R are atomic in nature.
- **Atomic:** The smallest level to which data may be broken down and remain meaningful

Generate the table in to First Normal Form.



Need for Normalization

Student_Course_Result Table

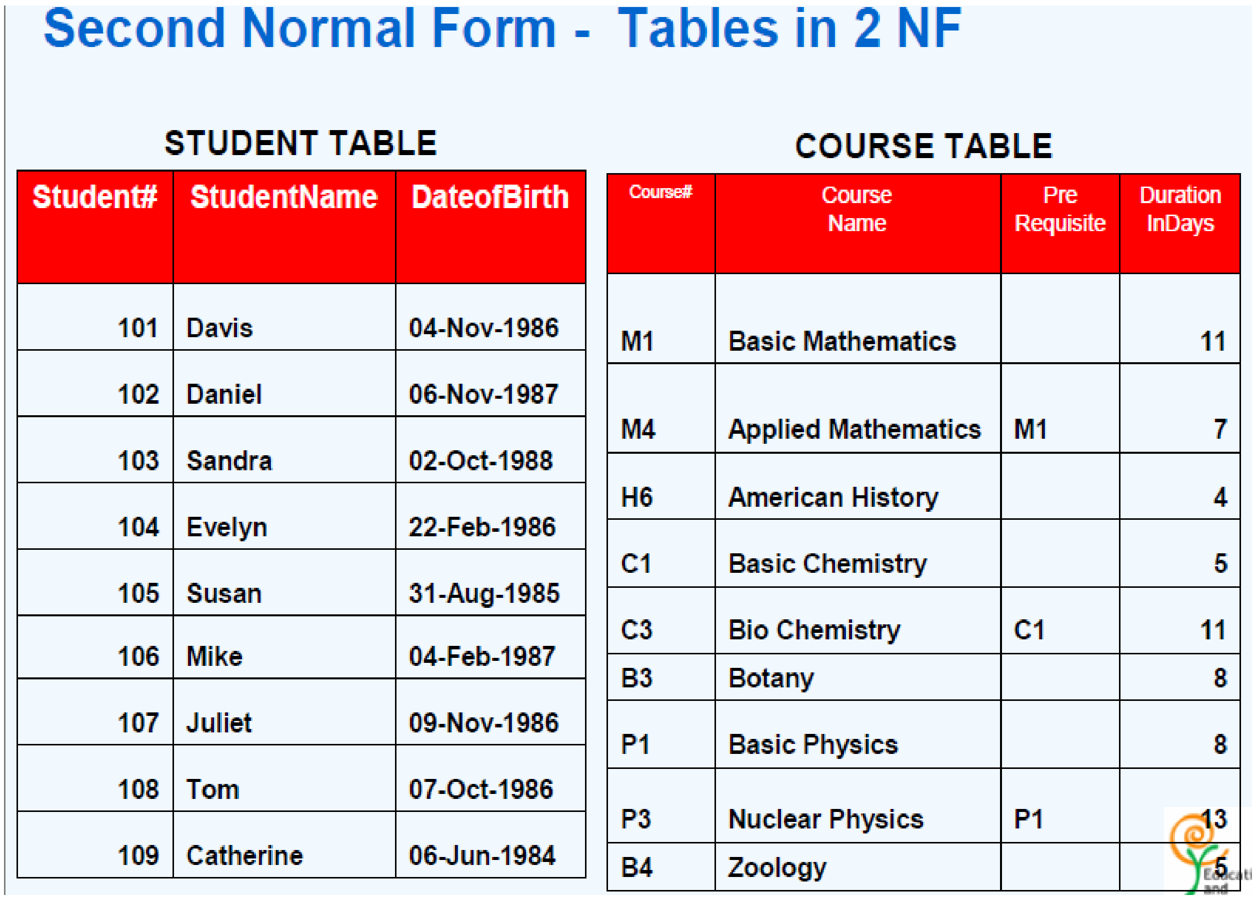
Student_Details			Course_Details				Result_Details		
101	Davis	11/4/1986	M4	Applied Mathematics	Basic Mathematics	7	11/11/2004	82	A
102	Daniel	11/6/1987	M4	Applied Mathematics	Basic Mathematics	7	11/11/2004	62	C
101	Davis	11/4/1986	H6	American History		4	11/22/2004	79	B
103	Sandra	10/2/1988	C3	Bio Chemistry	Basic Chemistry	11	11/16/2004	65	B
104	Evelyn	2/22/1986	B3	Botany		8	11/26/2004	77	B
102	Daniel	11/6/1987	P3	Nuclear Physics	Basic Physics	13	11/12/2004	68	B
105	Susan	8/31/1985	P3	Nuclear Physics	Basic Physics	13	11/12/2004	89	A
103	Sandra	10/2/1988	B4	Zoology		5	11/27/2004	54	D
105	Susan	8/31/1985	H6	American History		4	11/22/2004	87	A
104	Evelyn	2/22/1986	M4	Applied Mathematics	Basic Mathematics	7	11/11/2004	65	B

Second normal form:2NF

- A Relation is said to be in second Normal Form if and only if:
- It is in the First normal form, and

- No Partial dependency exists between non-key attributes and key attributes.

Generate the First Normal form table in to Second Normal Form



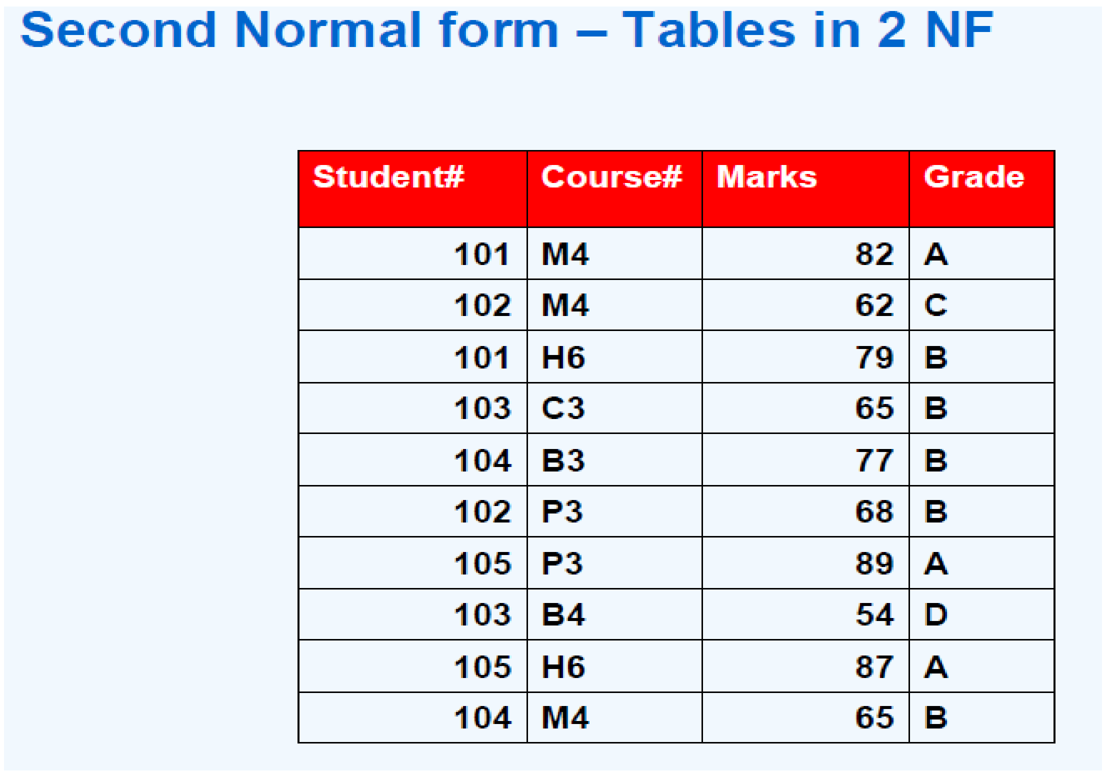
Second Normal Form - Tables in 2 NF

STUDENT TABLE

Student#	StudentName	DateofBirth
101	Davis	04-Nov-1986
102	Daniel	06-Nov-1987
103	Sandra	02-Oct-1988
104	Evelyn	22-Feb-1986
105	Susan	31-Aug-1985
106	Mike	04-Feb-1987
107	Juliet	09-Nov-1986
108	Tom	07-Oct-1986
109	Catherine	06-Jun-1984

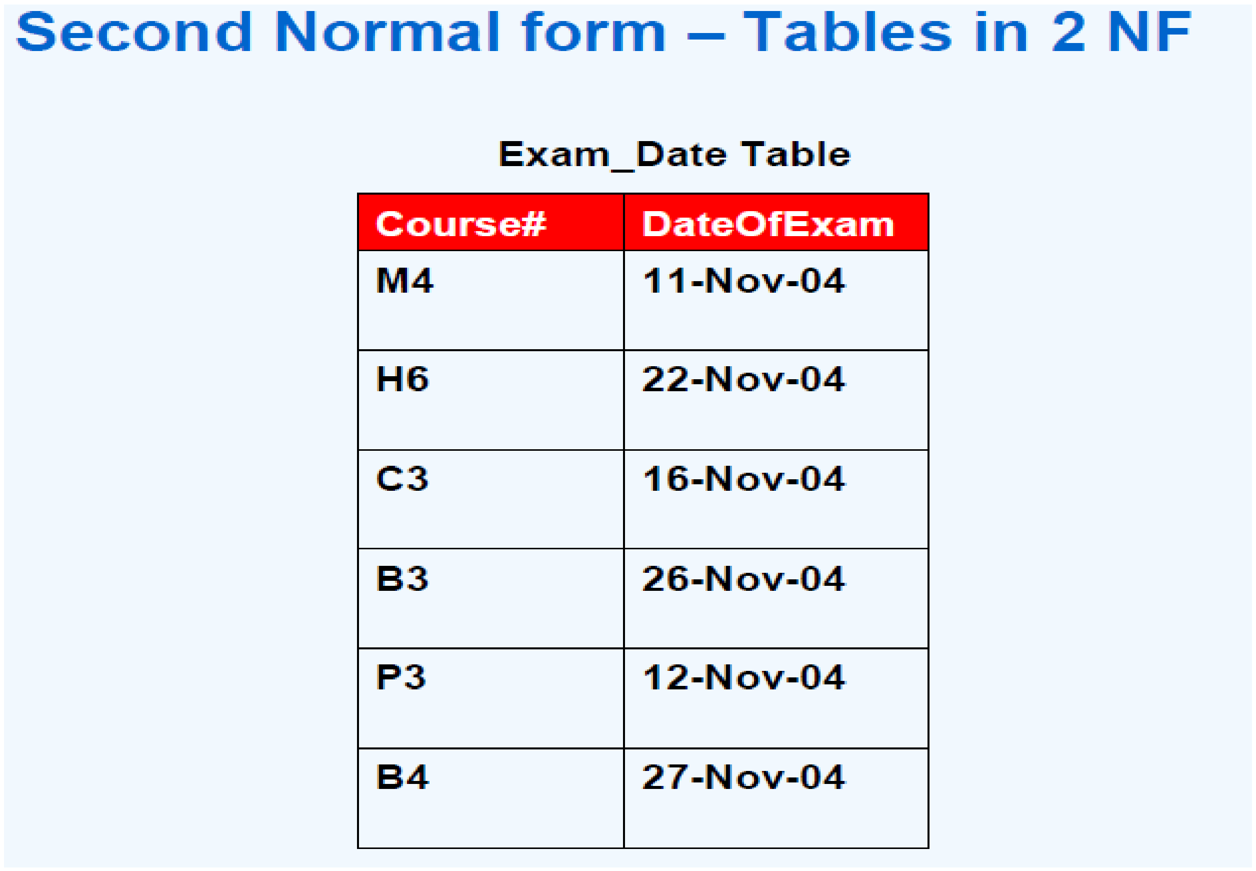
COURSE TABLE

Course#	Course Name	Pre Requisite	Duration InDays
M1	Basic Mathematics		11
M4	Applied Mathematics	M1	7
H6	American History		4
C1	Basic Chemistry		5
C3	Bio Chemistry	C1	11
B3	Botany		8
P1	Basic Physics		8
P3	Nuclear Physics	P1	13
B4	Zoology		5



Second Normal form – Tables in 2 NF

Student#	Course#	Marks	Grade
101	M4	82	A
102	M4	62	C
101	H6	79	B
103	C3	65	B
104	B3	77	B
102	P3	68	B
105	P3	89	A
103	B4	54	D
105	H6	87	A
104	M4	65	B



Second Normal form – Tables in 2 NF

Exam_Date Table

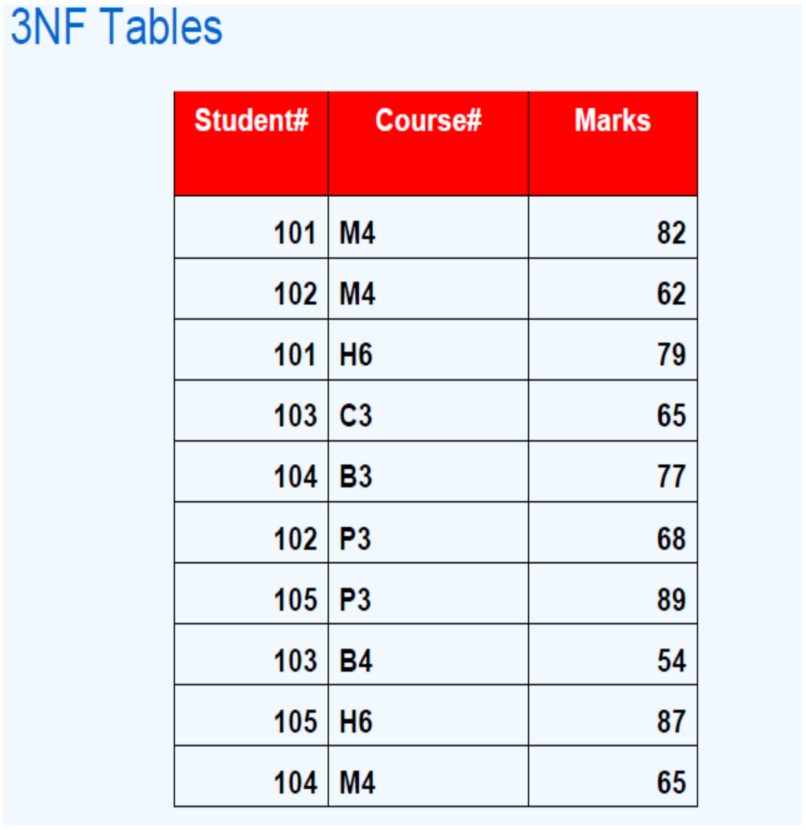
Course#	DateOfExam
M4	11-Nov-04
H6	22-Nov-04
C3	16-Nov-04
B3	26-Nov-04
P3	12-Nov-04
B4	27-Nov-04

Third normal form:3NF

A relation R is said to be in the third normal form(3NF) if and only if

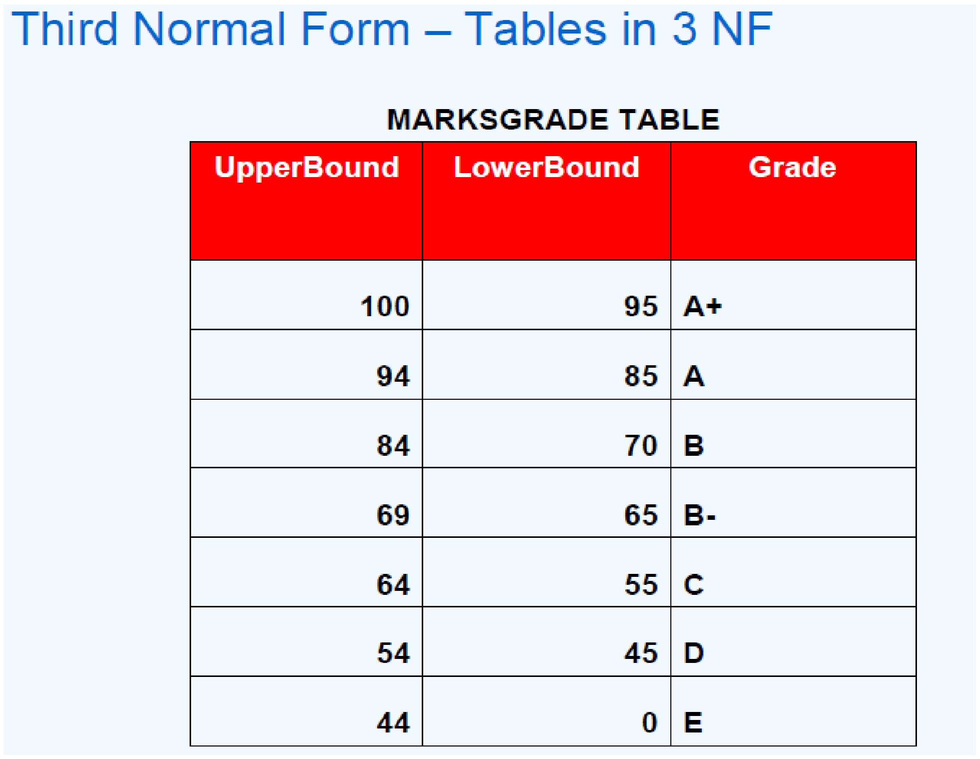
- It is in 2NF and
- No transitive dependency exists between non –key attributes and key attributes.

Generate the Second Normalform table in to Third Normal Form



3NF Tables

Student#	Course#	Marks
101	M4	82
102	M4	62
101	H6	79
103	C3	65
104	B3	77
102	P3	68
105	P3	89
103	B4	54
105	H6	87
104	M4	65



Third Normal Form – Tables in 3 NF

MARKSGRADE TABLE

UpperBound	LowerBound	Grade
100	95	A+
94	85	A
84	70	B
69	65	B-
64	55	C
54	45	D
44	0	E

VIVA Questions:

1What is normalization?

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data.It divides larger tables to smaller tables and links them using relationships.

2.What is Functional Dependency?

A **functional dependency** is a **constraint** between two sets of attributes in a relation from a database. In other words, functional dependency is a constraint that describes the relationship between attributes in a relation.

3. What is Fully Functional dependency?

The term full functional dependency (FFD) is used to indicate the minimum set of attributes in of a functional dependency (FD). In other words, the set of attributes X will be fufunctionally dependent on the set of attributes Y if the following conditions are satisfied:

- X is functionally dependent on Y and
- X is not functionally dependent on any subset of Y.

4. List different types Normal Forms?

- 1NF
- 2NF
- 3NF
- 4NF
- BCNF
- MULTIVALUED FUNCTIONAL DEPENDENCY

5.What is transitive property?

Same as transitive rule in algebra, if a → b holds and b → c holds, then a → c also holds. a → b is called as a functionally that determines b.