Practical - 1

Creating tables.

SQL Query to create a table is given as:

```
CREATE TABLE Person1
        (
                PersonID int  primary key,
                LastName varchar(255) unique,
                FirstName varchar(255) not null,
                Address varchar(255) default 'balaganj',
                City varchar(255) default 91
        );
```

Query to get detail of table

```
SQL> desc person1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 PERSONID                                  NOT NULL NUMBER(38)
 LASTNAME                                           VARCHAR2(255)
 FIRSTNAME                                 NOT NULL VARCHAR2(255)
 ADDRESS                                            VARCHAR2(255)
 CITY                                               VARCHAR2(255)
```

Practical - 2

Insertion, Deletion, Updation and Retrieval of data.

Query to insert a row in table person2 .

insert into person2 (personid1,lastname,firstname, personid) values (113,'dixit','ram',111);

```
SQL> insert into person2 (personid1,lastname,firstname,personid) values (113,'dixt','ram',111);

1 row created.
```

Query to select all row from table person2 .

Select * from person2;

```
SQL> select * from person2;

       PERSONID1
----------------
LASTNAME
------------------------------
FIRSTNAME
------------------------------
ADDRESS
------------------------------
CITY
------------------------------
        PERSONID
----------------
             113
dixt
ram
balaganj
91
             111
```

Query to update table person1

update person1 set lastname='awast',firstname='atul',address='banpur' where ((personid=111 or address='banpur') and lastname='awasthi');

select * from person1;

Query Output

```
SQL> update person1 set lastname='awast',firstname='atul',address='banpur' where ((personid=111 or address='banpur') and lastname='awasthi');

0 rows updated.

SQL> update person1 set lastname='awast',firstname='atul',address='banpur' where ((personid=111 or address='banpur') and lastname='awasthi');

0 rows updated.

SQL> select * from person1;

        PERSONID
---------------
LASTNAME
----------------------------------------------------------------------------------------------------
FIRSTNAME
----------------------------------------------------------------------------------------------------
ADDRESS
----------------------------------------------------------------------------------------------------
CITY
----------------------------------------------------------------------------------------------------
             111
awast
atul
banpur


             112
dix
ram
balaganj


             113
dixy
ram
balaganj



SQL>
```

Query to delete all rows from table ‘aa’

delete from aa;

```
SQL> delete from aa;

10 rows deleted.

SQL> select * from aa
  2  ;

no rows selected

SQL> desc aa
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- -----------
 QWERTYUIOPASD                                                  NUMBER(38)
```

Query to delete selected rows from table ‘person1’

DELETE FROM person1 WHERE ((personid=111 or address='balaganj') and lastname='dixy');

```
SQL> DELETE FROM person1 WHERE ((personid=111 or address='balaganj') and lastname='dixy');

1 row deleted.

SQL> select * from person1;

        PERSONID
----------------
LASTNAME
----------------------------------------------------------------------------------------
FIRSTNAME
----------------------------------------------------------------------------------------
ADDRESS
----------------------------------------------------------------------------------------
CITY
----------------------------------------------------------------------------------------
             111
awast
atul
banpur


             112
dix
ram
balaganj
```

Practical - 3

Arithmetic operations, Logical operations and Pattern matching.

Sql> SELECT * FROM person2 WHERE lastname LIKE 'a%';

SQL> SELECT * FROM person2 WHERE lastname LIKE 'a%' and city like '%o';

```
SQL> SELECT * FROM person2 WHERE lastname LIKE 'a%' and city like '%o';

       PERSONID1
----------------
LASTNAME
--------------------------------------------------------------------------
FIRSTNAME
--------------------------------------------------------------------------
ADDRESS
--------------------------------------------------------------------------
CITY
--------------------------------------------------------------------------
        PERSONID
----------------
             121
awas
rahu
banpu
hardo
             111
```

SQL> select lastname, personid1, personid from person2 order by personid asc, personid1 asc;

```
SQL> select lastname, personid1, personid from person2 order by personid asc, personid1 asc;

LASTNAME
------------------------------------------------------------------------------------------------
       PERSONID1        PERSONID
---------------- ----------------
dixt
            113             111

awas
            121             111

awast
            122             112
```

SQL> UPDATE person2 SET personid1 = personid1 + 1;

SQL> UPDATE person2 SET personid1 = personid1 - 1;

SQL> UPDATE person2 SET personid1 = personid1 * 1;

SQL> UPDATE person2 SET personid1 = personid1 / 1;

```
SQL> UPDATE person2  SET personid1 = personid1 + 1;

3 rows updated.

SQL> UPDATE person2  SET personid1 = personid1 - 1;

3 rows updated.

SQL> UPDATE person2  SET personid1 = personid1 * 1;

3 rows updated.

SQL> UPDATE person2  SET personid1 = personid1 / 1;

3 rows updated.
```

Practical - 4

Concept of Grouping (Group by clause, Having Clause).

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

SQL> SELECT count(*), address from employee GROUP BY address;

```
SQL> SELECT count(*), address from employee GROUP BY address;

        COUNT(*) ADDRESS
---------------- ----------------------------------------------
               1 ab balaganj
               1 bb balaganj
               1 cb balaganj
               1 db balaganj
               1 eb balaganj
               1 fb balaganj
               1 gb balaganj
               9 hb balaganj

8 rows selected.
```

SQL> SELECT COUNT(empid), name FROM employee GROUP BY name ORDER BY COUNT(empID) DESC;

```
SQL> SELECT COUNT(empid), name FROM employee GROUP BY name ORDER BY COUNT(empID) DESC;

    COUNT(EMPID) NAME
---------------- ------------------------------------------------
               4 jb
               2 kb
               1 cb
               1 db
               1 eb
               1 lb
               1 gb
               1 hb
               1 ib
               1 bb
               1 ab
               1 fb

12 rows selected.
```

SQL> SELECT count(empid), name from employee GROUP BY name;

The SQL HAVING Clause

```
SQL> select * from employee;

           EMPID NAME                                               ADDRESS                                                   SAL              AGE
---------------- -------------------------------------------------- -------------------------------------------------- ---------------- ----------------
               1 ab                                                 ab balaganj                                                1000               18
               2 bb                                                 bb balaganj                                                2000               19
               3 cb                                                 cb balaganj                                                3000               20
               4 db                                                 db balaganj                                                4000               21
               5 eb                                                 eb balaganj                                                5000               22
               6 fb                                                 fb balaganj                                                6000               23
               7 gb                                                 gb balaganj                                                7000               24
               8 hb                                                 hb balaganj                                                8000               25
               9 ib                                                 hb balaganj                                                8000               25
              10 jb                                                 hb balaganj                                                8000               25
              10 jb                                                 hb balaganj                                                8000               25
              11 jb                                                 hb balaganj                                                8000               25
              12 jb                                                 hb balaganj                                                8000               25
              12 kb                                                 hb balaganj                                                8000               25
              13 kb                                                 hb balaganj                                                8000               26
              14 lb                                                 hb balaganj                                                8000               27

16 rows selected.

SQL> SELECT count(empid), name from employee GROUP BY name;

    COUNT(EMPID) NAME
---------------- --------------------------------------------------
               1 ab
               1 bb
               1 cb
               1 db
               1 eb
               1 fb
               1 gb
               1 hb
               1 ib
               4 jb
               2 kb
               1 lb

12 rows selected.
```

The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

SQL> SELECT COUNT(empID), name FROM employee GROUP BY name HAVING COUNT(empID) >=4;

SQL> SELECT COUNT(empID), name FROM employee GROUP BY name HAVING COUNT(empID) >=1;

SQL> SELECT COUNT(empID), name FROM employee GROUP BY name HAVING COUNT(empID) >=2;

```
SQL> SELECT COUNT(empID), name FROM employee GROUP BY name HAVING COUNT(empID) >=4;

    COUNT(EMPID) NAME
---------------- --------------------------------------------------
               4 jb

SQL> SELECT COUNT(empID), name FROM employee GROUP BY name HAVING COUNT(empID) >=1;

    COUNT(EMPID) NAME
---------------- --------------------------------------------------
               1 ab
               1 bb
               1 cb
               1 db
               1 eb
               1 fb
               1 gb
               1 hb
               1 ib
               4 jb
               2 kb
               1 lb

12 rows selected.

SQL> SELECT COUNT(empID), name FROM employee GROUP BY name HAVING COUNT(empID) >=2;

    COUNT(EMPID) NAME
---------------- --------------------------------------------------
               4 jb
               2 kb
```

Practical - 5

Use Aggregate function in query.

SQL>SELECT SUM(personid1) FROM person2;

SQL>SELECT count(personid1) FROM person2;

SQL>SELECT avg(personid1) FROM person2;

SQL> SELECT min(personid1) FROM person2;

SQL> SELECT max(personid1) FROM person2;

```
SQL> select * from person2;

       PERSONID1
----------------
LASTNAME
----------------------------------------
FIRSTNAME
----------------------------------------
ADDRESS
----------------------------------------
CITY
----------------------------------------
        PERSONID
----------------
             113
dixt
ram
balaganj
91
             111

             121
awas
rahu
banpu
hardo
             111

             122
awast
rahul
banpur
hardoi
             112


SQL> SELECT SUM(personid1) FROM person2;

  SUM(PERSONID1)
----------------
             356

SQL> SELECT avg(personid1) FROM person2;

  AVG(PERSONID1)
----------------
118.666666666667

SQL> SELECT count(personid1) FROM person2;

COUNT(PERSONID1)
----------------
               3
```

```
SQL> SELECT SUM(personid1) FROM person2;

  SUM(PERSONID1)
----------------
             356

SQL> SELECT avg(personid1) FROM person2;

  AVG(PERSONID1)
----------------
118.666666666667

SQL> SELECT count(personid1) FROM person2;

COUNT(PERSONID1)
----------------
               3

SQL> SELECT min(personid1) FROM person2;

  MIN(PERSONID1)
----------------
             113

SQL> SELECT max(personid1) FROM person2;

  MAX(PERSONID1)
----------------
             122
```

Practical - 6

Write commands for Union and Intersection.

Union and intersection

SQL> select name from employee where age >= 20

2 union

3 select name from employee where age >= 22;

SQL> select name from employee where age >= 22

2 intersect

3 select name from employee where age >= 20;

```
SQL> select name from employee where age >= 20
  2  union
  3  select name from employee where age >= 22;

NAME
----------------------------------------------
cb
db
eb
fb
gb
hb
ib
jb
kb
lb

10 rows selected.

SQL> select name from employee where age >= 22
  2  intersect
  3  select name from employee where age >= 20;

NAME
----------------------------------------------
eb
fb
gb
hb
ib
jb
kb
lb

8 rows selected.
```

Practical - 7

Concept of Sub-query.

SELECT personid, firstname, address,city FROM person1 WHERE personid= (SELECT MAX(personid) FROM person1);

```
SQL> SELECT personid, firstname, address,city FROM person1 WHERE personid= (SELECT MAX( personid)  FROM person1 );

  PERSONID
----------
FIRSTNAME
--------------------------------------------------------------------------------
ADDRESS
--------------------------------------------------------------------------------
CITY
--------------------------------------------------------------------------------
       112
ram
balaganj
```

Practical - 8

Concept of Data constraints (Unique Key, Primary Key, Foreign Key).

CREATE TABLE Person2 (

PersonID1 int ,

LastName varchar(255) unique,

FirstName varchar(255) not null,

Address varchar(255) default ‘balaganj’,

City varchar(255) default 91,

PersonID int,

primary key(PersonID1),

foreign key(PersonID) references person1(PersonID)

);

```
SQL> CREATE TABLE Person2 (
  2                        PersonID1 int  ,
  3                        LastName varchar(255) unique,
  4                       FirstName varchar(255) not null,
  5                        Address varchar(255) default ‘balaganj’,
  6                      City varchar(255) default 91,
  7  PersonID int,
  8  primary key(PersonID1),
  9   foreign key(PersonID) references person1(PersonID)
 10            );

Table created.

SQL> desc person2;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PERSONID1                                 NOT NULL NUMBER(38)
 LASTNAME                                           VARCHAR2(255)
 FIRSTNAME                                 NOT NULL VARCHAR2(255)
 ADDRESS                                            VARCHAR2(255)
 CITY                                               VARCHAR2(255)
 PERSONID                                           NUMBER(38)
```

Practical - 9

Creating Views and Indexes.

SQL> create view vperson as select personid, firstname, address from person1;

SQL> select * from vperson;

```
SQL> create index indxp on person1(firstname);

Index created.
```

```
SQL> create view vperson as
  2  select personid, firstname, address
  3  from person1;

View created.

SQL> select * from vperson;

  PERSONID
----------
FIRSTNAME
----------------------------------------
ADDRESS
----------------------------------------
       111
atul
banpur

       112
ram
balaganj

  PERSONID
----------
FIRSTNAME
----------------------------------------
ADDRESS
----------------------------------------


SQL> create view vper as
  2  select personid, firstname, address
  3  from person1;

View created.

SQL> drop view vper;

View dropped.
```

Practical -10

Library Management System

Library Books(Accession No, Title, Author, Department, Purchase Date , Price)

Issued Books(Accession No ,Borrower)

SQL> create table library_books(acc_no number(4)primary key,title varchar(10),author varchar(15),department varchar(10),p_date date,price number(5,2));

SQL> create table issued_books(acc_no ,borrower varchar(50),foreign key(acc_no)references library_books(acc_no));

```
SQL> desc library_books;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                    NOT NULL NUMBER(4)
 TITLE                                              VARCHAR2(10)
 AUTHOR                                             VARCHAR2(10)
 DEPARTMENT                                         VARCHAR2(8)
 P_DATE                                             DATE
 PRICE                                              NUMBER(5,2)

SQL> desc issued_books;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                             NUMBER(4)
 BORROWER                                           VARCHAR2(15)
```

- **Identify Primary key and Foreign keys, Create the tables and insert at least 5 records in each table.**
  - **Primary keys for library_books are Acc_no**
  - **Foreign key for issued_books is Acc_no**

SQL> insert into library_books(101,'DBMS','A W,'C.S','01-JUL-2022',750);

SQL> insert into library_books values(102,'SE','K Pressman','C.S','02-JUL-2022,800);

SQL> insert into library_books values(103,'MM,'Mcgrew ','C.S','03-JUL-2022',900);

SQL> insert into library_books values(104,'DM Manohar','C.S','04-JUL2022',900);

SQL> insert into library_books values(105,'BE ','D.N Divedi','C.S','05-JUL-2022',850);

```
QL> SELECT * FROM LIBRARY_BOOKS;

  ACC_NO TITLE     AUTHOR     DEPARTME P_DATE       PRICE
--------- ---------- ---------- -------- --------- ----------
      101 DBMS       A.W        C.S      01-JUL-22       750
      102 S E        K.PRESSMAN C.S      02-JUL-22       800
      103 M M        Mcgrew     C.S      03-JUL-22       900
      104 D D        R MANOHAR  C.S      04-JUL-22       900
      105 B E        D N        C.S      06-JUL-22       850

QL>
```

SQL> insert into issued_books values(101,'Pradeep');

SQL> insert into issued_books values(102,'Amit');

SQL> insert into issued_books values(103,'Sujeet');

SQL> insert into issued_books values(104,'Anujt');

SQL> insert into issued_books values(105,'Aditi');

```
SQL> select * from issued_books;

   ACC_NO BORROWER
---------- ---------------
       101 Sujeet
       102 Amit
       103 Pradeep
       104 Anup

SQL>
```

❖ Delete the record of book Title "DBMS" .

SQL> delete from library_books where Title="Business Econnomics";

❖ Change the Department of the book Titled "Discrete Mathematics" to "CSE".

SQL> update library_books set department="CSE" where Title ="DM ";

```
SQL> update library_books set department ='CSE' where acc_no=101;

1 row updated.

SQL> select *from library_books;

   ACC_NO TITLE     AUTHOR     DEPARTME P_DATE        PRICE
---------- --------- ---------- -------- --------- ----------
       101 DBMS      A.W        CSE      01-JUL-22        750
       102 S E       K.PRESSMAN C.S      02-JUL-22        800
       103 M M       Mcgrew     C.S      03-JUL-22        900
       104 D D       R MANOHAR  C.S      04-JUL-22        900
       105 B E       D N        C.S      06-JUL-22        850

SQL>
```

❖ List all books that belong to "CSE" department

SQL> select *from library_books where department ="CSE';

```
SQL> select *from library_books WHERE AUTHOR='A.W';

   ACC_NO TITLE      AUTHOR     DEPARTME P_DATE        PRICE
---------- ---------- ---------- -------- --------- ----------
       101 DBMS       A.W        CSE      01-JUL-22        750

SQL>
```

- **List all books that belong to “CSE” department and are written by author “A.W”.**

```
SQL> select *from library_books WHERE AUTHOR='A.W';

   ACC_NO TITLE      AUTHOR     DEPARTME P_DATE        PRICE
---------- ---------- ---------- -------- --------- ----------
       101 DBMS       A.W        CSE      01-JUL-22        750

SQL>
```

- **List all books which have a price less than 900 or purchased between “12/12/2022”.**

SQL> select * from library_books where (price<900 or p_date>="01-JUL-2022" and p_date <="05-JUL-2022");

```
SQL> select * from library_books where price<900 and P_date<='05-jul-2022';

    ACC_NO TITLE      AUTHOR     DEPARTME P_DATE         PRICE
---------- ---------- ---------- -------- --------- ----------
       101 DBMS       A.W        CSE      01-JUL-22        750
       102 S E        K.PRESSMAN C.S      02-JUL-22        800
       105 B E        D N        C.S      05-JUL-22        850

SQL>
```

Practical -11

Student Management System

Student (College_roll_no,Std_name,Dob,Address,Marks(Round off to whole number) in Percentage at 10+2 phone_no)

Paper_Details(Paper_code,Name_of_paper)

Academic_Details(College_roll_no,Paper_code,Attendance,Marks n home examination)

Identify primary key and foreign keys.create the table and insert at least 5 records in each table

SQL> create table student(Std_roll_no number(4) primary key,std_name varchar(15),dob date,address varchar(15),marks number(8,2));

```
SQL> desc student;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STD_ROLL_NO                               NOT NULL NUMBER(4)
 STD_NAME                                           VARCHAR2(15)
 DOB                                                DATE
 ADDRESS                                            VARCHAR2(15)
 MARKS                                              NUMBER(8,2)
 PHONE_NO                                           NUMBER(10)

SQL>
```

SQL> create table paper_details(paper_code number(8) primary key ,name_of_paper varchar(25));

```
SQL> desc paper_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PAPER_CODE                                NOT NULL NUMBER(8)
 NAME_OF_PAPER                                      VARCHAR2(25)

SQL>
```

SQL> create table academic_details(std_roll_no number(4),paper_code number(8),attendance varchar(8),marks number(10),foreign key (std_roll_no) references student(std_roll_no),foreign key(paper_code) references paper_details(paper_code));

```
SQL> desc academic_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STD_ROLL_NO                                        NUMBER(4)
 PAPER_CODE                                         NUMBER(8)
 ATTENDANCE                                         VARCHAR2(8)
 MARKS                                              NUMBER(10)

SQL>
```

Primary key : *For Student std_roll_no,for paper details paper_code.*

Foreign key: *std_roll_no ,paper_code for academic_details.*

Sql>insert into students values ('11','sujeet','03-feb-2002','lucknow','369','79058014')

Sql>insert into students values ('12','Amit','05-may-2003','Piparsand','340','89245364')

Sql>insert into students values ('13','Pradeep','15-aug-2004','Malihabad','524','76549809')

Sql>insert into students values ('14','Anuj','12-apr-2001','Malihabad','349','98759876')

```
SQL> select *from students;

STD_ROLL_NO STD_NAME        DOB       ADDRESS              MARKS   PHONE_NO
----------- --------------- --------- --------------- ---------- ----------
         11 sujeet          03-FEB-02 lucknow                369   79058014
         12 Amit            05-MAY-03 Piparsand              340   89245364
         13 Pradeep         15-AUG-04 Malihabad              524   76549809
         14 Anuj            12-APR-01 Malihabad              349   98759876
```

Sql>insert into paper_details values ('101','Paper1')

Sql>nsert into paper_details values ('102','Paper2')

Sql>insert into paper_details values ('103','Paper3')

Sql>insert into paper_details values ('104','Paper4')

Sql>insert into paper_details values ('105','Paper5')

```
SQL> select * from paper_details;

PAPER_CODE NAME_OF_PAPER
---------- --------------------------
       101 Paper1
       102 Paper2
       103 Paper3
       104 Paper4
       105 Paper5

SQL> _
```

insert into academic_details values('11','101','p','369')

insert into academic_details values('12','102','p','340')

insert into academic_details values('13','103','p','524')

insert into academic_details values('14','104','p','341')

```
SQL> select * from academic_details;

STD_ROLL_NO PAPER_CODE ATTENDAN      MARKS
----------- ---------- -------- ----------
         11        101 p               369
         12        102 p               340
         13        103 p               524
         14        104 p               341
```

Design a query that will return the records (from the second table along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper 2.

Sql> select paper_code,name_of_paper,std_name,from paper_detials,students,academic_details

Where paper_details.paper_code=academic_details.paper_code and academic_details.std_roll_no=students.std_roll_no and academic_details.attendance>75 and academic_detials.marks>75 and paper_details.name_of_paper='Paper2';

```
STD_NAME
--------------
Sujeet

SQL>
```

List all students who live in " Lucknow " and have marks greater than 60 in paper 1.

Sql> select std_name from paper_details,students,academic_details where paper_details.paper_code=academic_details.paper_code and students.std_roll_no=academic_details.std_roll_no and academic_details.marks>60 and paper_details.name_of_paper='Paper1';

Select sum(marks)from academic_details.

Sql>Select sum(marks)from academic_details.

```
SQL> select sum(marks) from academic_details;

SUM(MARKS)
----------
      1150

SQL>
```

Practical -12

Customer Management System

Customer(Cust_id,Email,Name,Phone,Referrer_id).

Bicycle(Bicycle_id,Date_purchased,Color,Cust_id, Model_no).

Bicycle_model(Model_no,Manufacture,Style).

Service(Start_date,Bicycle_id,End_date).

A) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.

SQL> create table Bicycle(Bicycle_id int primary key,Date_of_purchased date,Color varchar(20),Cust_id int,Model_no int,foreign key (Cust_id) references Customer(cust_id),foreign key(Model_no)references Bicycle_model(Model_no));

SQL> create table Bicycle_model(Model_no int primary key, Manufacturer varchar(50),Style varchar(35));

create table Customer(Cust_id int primary key ,Email varchar(30) unique ,Name varchar(25),Phone int,Referrerid int);

```
SQL Plus
Table created.

SQL> desc Bicycle;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 BICYCLE_ID                                NOT NULL NUMBER(38)
 DATE_OF_PURCHASED                                  DATE
 COLOR                                              VARCHAR2(20)
 CUST_ID                                            NUMBER(38)
 MODEL_NO                                           NUMBER(38)

SQL> desc Bicycle_model;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 MODEL_NO                                  NOT NULL NUMBER(38)
 MANUFACTURER                                       VARCHAR2(50)
 STYLE                                              VARCHAR2(35)

SQL> desc Customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 CUST_ID                                   NOT NULL NUMBER(38)
 EMAIL                                              VARCHAR2(30)
 NAME                                               VARCHAR2(25)
 PHONE                                              NUMBER(38)
 REFERRERID                                         NUMBER(38)

SQL>
```

- **Primary key** :- Bicycle_id for Bicycle and Cust_id for Customer and Model_no for cycle_model.
- **Foreign Key :-** Cust_id ,Model_no for Bicycle.

SQL> insert into Customer values (212,'sujeet7@gmail.com','Sujeet',790580,20);

SQL>insert into Customer values (213,'tut7@gmail.com','Ajeet',980580,10);

SQL> insert into Customer values (214,'rut7@gmail.com','Amit',890580,15);

SQL> insert into Customer values (215,'st7@gmail.com','Smit',832580,85);

SQL> insert into Customer values (205,'stt7@gmail.com','Satu',832380,715);

```
SQL> select *from Customer;

   CUST_ID EMAIL                          NAME                          PHONE
---------- ------------------------------ ------------------------- ----------
REFERRERID
----------
       212 sujeet7@gmail.com              Sujeet                       790580
        20

       213 tut7@gmail.com                 Ajeet                        980580
        10

       214 rut7@gmail.com                 Amit                         890580
        15


   CUST_ID EMAIL                          NAME                          PHONE
---------- ------------------------------ ------------------------- ----------
REFERRERID
----------
       215 st7@gmail.com                  Smit                         832580
        85

       205 stt7@gmail.com                 Satu                         832380
       715
```

SQL> insert into Bicycle_model values (77,'Honda','Ranger');

SQL> insert into Bicycle_model values (87,'Honda','Ranger');

SQL> insert into Bicycle_model values (87,'Honda','Ranger');

SQL> insert into Bicycle_model values (197,'Anvo',' Sports');

SQL> insert into Bicycle_model values (117,'Hero',' Sports');

```
SQL> select *from Bicycle_model;

  MODEL_NO MANUFACTURER
---------- ----------------------------------------------------
STYLE
-----------------------------------
        77 Honda
Ranger

        87 Honda
Ranger

        97 Anvo
 Sports


  MODEL_NO MANUFACTURER
---------- ----------------------------------------------------
STYLE
-----------------------------------
       197 Anvo
 Sports

       117 Hero
 Sports
```

Create the new table Service Start date and End date of the Service .

```
SQL> create table Service(Start_date date,Bicycle_id int primary key,End_date date);
SQL> insert into Service values('12/jan/2000',100,'12/jan/2001');
SQL> insert into Service values('13/jan/2000',101,'13/jan/2001');
SQL> insert into Service values('14/jan/2000',102,'14/jan/2001');
SQL> insert into Service values('15/jan/2000',103,'15/jan/2001');
SQL> insert into Service values('16/jan/2000',104,'16/jan/2001');
```

```
SQL Plus
1 row created.

SQL> select *from Service;

START_DAT BICYCLE_ID END_DATE
--------- ---------- ---------
12-JAN-00        100 12-JAN-01
13-JAN-00        101 13-JAN-01
14-JAN-00        102 14-JAN-01
15-JAN-00        103 15-JAN-01
16-JAN-00        104 16-JAN-01

SQL>
```