Assignment 2:

SQL> create table branch(

branch\_name varchar(10) NOT NULL,

branch\_city varchar(10) NOT NULL,

assets number NOT NULL,

CONSTRAINT branch\_pk PRIMARY KEY(branch\_name)

);

Table created.

SQL> create table customer(

cust\_name varchar(20) NOT NULL,

cust\_street varchar(10) NOT NULL,

cust\_city varchar(10) NOT NULL,

CONSTRAINT customer\_pk PRIMARY KEY(cust\_name)

);

Table created.

SQL> create table depositor(

cust\_name varchar(20) NOT NULL,

acc\_no number NOT NULL,

CONSTRAINT depositor\_pk PRIMARY KEY(acc\_no),

CONSTRAINT fk\_depositor FOREIGN KEY(cust\_name) references customer(cust\_name)

);

Table created.

SQL> create table account(

acc\_no number NOT NULL,

branch\_name varchar(10) NOT NULL,

balance number NOT NULL,

CONSTRAINT fk1\_account FOREIGN KEY(acc\_no) references depositor(acc\_no),

CONSTRAINT fk2\_account FOREIGN KEY(branch\_name) references branch(branch\_name)

);

Table created.

SQL > create table borrower(

cust\_name varchar(20) NOT NULL,

loan\_no number NOT NULL,

CONSTRAINT borrower\_pk PRIMARY KEY(loan\_no),

CONSTRAINT fK\_borrower FOREIGN KEY(cust\_name) references customer(cust\_name)

);

Table created.

SQL> create table loan(

loan\_no number NOT NULL,

branch\_name varchar(10) NOT NULL,

amount number NOT NULL,

CONSTRAINT fK1\_loan FOREIGN KEY(loan\_no) references borrower(loan\_no),

CONSTRAINT fK2\_loan FOREIGN KEY(branch\_name) references branch(branch\_name)

);

Table created.

SQL> insert into branch values('Akurdi', 'Pune', 1000000);

1 row created.

SQL> insert into branch values('Nigdi', 'Pune', 900000);

1 row created.

SQL> select \* from branch;

BRANCH\_NAM BRANCH\_CIT ASSETS

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

Akurdi Pune 1000000

Nigdi Pune 900000

SQL> insert into customer values('Nikhil', 'Nigdi', 'Pune');

1 row created.

SQL> insert into customer values('Pranav', 'Chinchwad', 'Pune');

1 row created.

SQL> insert into customer values('Aniket', 'Bhosari', 'Pune');

1 row created.

SQL> insert into customer values('Darshan', 'Chinchwad', 'Pune');

1 row created.

SQL> select \* from customer;

CUST\_NAME CUST\_STREE CUST\_CITY

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

Nikhil Nigdi Pune

Pranav Chinchwad Pune

Aniket Bhosari Pune

Darshan Chinchwad Pune

SQL> insert into depositor values('Nikhil', 1001);

1 row created.

SQL> insert into depositor values('Pranav', 1002);

1 row created.

SQL> select \* from depositor;

CUST\_NAME ACC\_NO

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

Nikhil 1001

Pranav 1002

SQL> insert into account values(1001, 'Nigdi', 15000);

1 row created.

SQL> insert into account values(1002, 'Chinchwad', 11000);

insert into account values(1002, 'Chinchwad', 11000)

\*

ERROR at line 1:

ORA-02291: integrity constraint (DARKKNIGHT532.FK2\_ACCOUNT) violated - parent

key not found

SQL> insert into account values(1002, 'Akurdi', 11000);

1 row created.

SQL> select \* from account;

ACC\_NO BRANCH\_NAM BALANCE

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

1001 Nigdi 15000

1002 Akurdi 11000

SQL> insert into borrower values('Aniket', 2001);

1 row created.

SQL> insert into borrower values('Darshan', 2001);

insert into borrower values('Darshan', 2001)

\*

ERROR at line 1:

ORA-00001: unique constraint (DARKKNIGHT532.BORROWER\_PK) violated

SQL> insert into borrower values('Darshan', 2002);

1 row created.

SQL> select \* from borrower;

CUST\_NAME LOAN\_NO

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

Aniket 2001

Darshan 2002

SQL> insert into loan values(2001, 'Nigdi', 15000);

1 row created.

SQL> insert into loan values(2002, 'Akurdi', 10000);

1 row created.

SQL> select \* from loan;

LOAN\_NO BRANCH\_NAM AMOUNT

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

2001 Nigdi 15000

2002 Akurdi 10000

Q1. Find the names of all branches in loan relation.

SQL> select branch\_name from loan;

BRANCH\_NAM

---------

Nigdi

Akurdi

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

Q2. Find all loan numbers for loans made at Nigdi Branch with loan amount > 12000.

SQL> select loan\_no from loan where branch\_name = 'Nigdi' and amount > 12000;

LOAN\_NO

----------

2001

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

Q3. Find all customers who have a loan from bank. Find their names, loan\_no

and loan amount.

SQL> select b.cust\_name, b.loan\_no, l.amount from borrower b, loan l

where b.loan\_no = l.loan\_no;

CUST\_NAME LOAN\_NO AMOUNT

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

Aniket 2001 15000

Darshan 2002 10000

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

Q4. List all customers in alphabetical order who have loan from Akurdi branch.

SQL> select cust\_name from borrower b, loan l

where branch\_name = 'Akurdi' and b.loan\_no =l.loan\_no order by cust\_name;

CUST\_NAME

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

Darshan

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

Q5. Find all customers who have an account or loan or both at bank

SQL> (select cust\_name from depositor) UNION (select cust\_name from borrower);

CUST\_NAME

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

Aniket

Darshan

Nikhil

Pranav

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

Q6. Find all customers who have both account and loan at bank.

SQL> (select cust\_name from depositor) INTERSECT (select cust\_name from borrower);

no rows selected

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

Q7. Find all customer who have account but no loan at the bank

SQL> select cust\_name from depositor

where cust\_name NOT IN (select cust\_name from borrower);

CUST\_NAME

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

Nikhil

Pranav

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

Q8. Find average account balance at Akurdi branch

SQL> select avg(balance) from account where branch\_name = 'Akurdi';

AVG(BALANCE)

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

11000

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

Q9. Find the average account balance at each branch

SQL> select branch\_name, avg(balance) from account group by branch\_name;

BRANCH\_NAM AVG(BALANCE)

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

Nigdi 15000

Akurdi 11000

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

Q10. Find no. of depositors at each branch.

BRANCH\_NAM COUNT(ACC\_NO)

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

Nigdi 1

Akurdi 1

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

Q11. Find the branches where average account balance > 12000

SQL> select branch\_name from account a

where (select avg(balance) from account where branch\_name = a.branch\_name ) > 12000;

BRANCH\_NAM

----------

Nigdi

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

Q12. Find number of tuples in customer relation.

SQL> select count(\*) from customer;

COUNT(\*)

----------

4

Q13. Calculate total loan amount given by bank.

SQL> select sum(amount) as TOTAL\_LOAN from loan;

TOTAL\_LOAN

---------

25000

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

Q14. Delete all loans with loan amount between 1300 and 1500.

SQL> delete from loan where amount >= 1300 and amount <= 1500;

0 rows deleted.

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

Q15. Delete all tuples at every branch located in Nigdi.

SQL> delete from account where branch\_name = 'Nigdi';

1 row deleted.

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

Q.16. Create synonym for customer table as cust.

SQL> create or replace public synonym cust for customer;

Synonym created.

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

Q.17. Create sequence roll\_seq and use in student table for roll\_no column.

SQL> create sequence roll\_seq

minvalue 1

maxvalue 100

start with 1

increment by 1

cache 10;

Sequence created.

SQL> INSERT INTO student(roll\_no, student\_name) VALUES(roll\_seq.NEXTVAL, 'Nikhil');

1 row created.

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