ACCOUNT TABLE

```
create table ACCOUNT (acc_no varchar2(5) primary key,
name varchar2(30) not null, city varchar2(20) not null,
balance number(10,2),loan_taken varchar2(3),
check(acc_no like'A%'),
check(loan_taken='NO'or loan_taken='YES'));
```

INSERT

```
insert into account values('A001','Patel Jigar','Meshana',50000,'YES');
insert into account values('A002','Patel Ramesh','Meshana',50000,'YES');
insert into account values('A003','Dave Hardik','Ahmedabad',75000,'NO');
insert into account values('A004','Soni Hetal','Ahmedabad',100000,'NO');
insert into account values('A005','Sony Atul','Vadodara',50000,'YES');
```

LOAN TABLE

```
create table LOAN(loan_no varchar2(5) primary key check(loan_no like'L%'),
acc_no varchar2(5) references account(acc_no),loan_amt number(10,2)
not null, interest_rate number(5,2) not null, loan_date date, remaining_loan
number(10,2), check(remaining_loan<loan_amt));
```

INSERT

```
insert into loan values('L001','A001',100000,7, '1-1-04', 75000);
insert into loan values('L002','A002',300000,9, '5-18-04', 150000);
insert into loan values('L003','A003',500000,11,'6-15-04',300000);
```

INSTALLMENT TABLE

```
create table INSTALLMENT(loan_no varchar2(5) references
loan(loan_no),inst_no varchar2(5), idate date not null ,amount number(10,2)
not null, check(inst_no like'I%'), primary key(loan_no, inst_no));
```

INSERT

```
1. insert into installment values('L001','I001','2-2-04',15000);
```

2. insert into installment values('L002','I002','6-18-04',20000);

3. insert into installment values('L003','I003','7-15-04',20000);

TRANSACTION TABLE

create table TRANSACTION(acc_no varchar2(5) references account(acc_no),

trans_date date not null, amt number(10,2) not null,

type_of_tr char(1),mode_of_pay varchar2(10),primary key(acc_no),

check(type_of_tr='D' or type_of_tr='W'),

check(mode_of_pay='cash' or mode_of_pay='cheque'));

INSERT

insert into transaction values('A001','5-3-04',10000,'D','cash');

insert into transaction values('A002','7-5-04',5000,'W','cheque');

insert into transaction values('A003','8-12-04',25000,'D','cheque');

insert into transaction values('A004','5-15-04',30000,'D','cheque');

insert into transaction values('A005','10-22-04',15000,'W','cash');

Using Operator NOT, BETWEEN, NOT BETWEEN,IN, NOT IN

1. ***Retrieve specified information for the account holder who are not in 'Ahmedabad'.***

:- select * from account where not city ='Ahmedabad';

select acc_no,name,city,balance,loan_taken from account where not city in'Ahmedabad';

2. Retrieve specified information for the account holder who are not in 'Ahmedabad'

or 'Vadodara'.

:-select * from account where not city='Ahmedabad or 'Vadodara';

3. Retrieve those records of Account holder whose balance between is 50000 and 100000.

:- select * from account where balance BETWEEN 50000 and 100000;

4. Retrieve those records of Account holder whose balance not between is

50000 and 100000.

:- select * from account where balance NOT BETWEEN 50000 and 100000;

5. Display only those records whose amount is 5000, 25000, 30000.

:- select * from INSTALLMENT where amount IN(5000,25000,30000);

6. Display only those records whose amount not in 5000, 25000, 30000.

:- select * from INSTALLMENT where amount NOT IN(5000,25000,30000);

7. Display System date.

:-select sysdate from dual;

8. Find the date,15 days after today's date.

:-select sysdate+15 from dual;

9. Perform following operation using DUAL table.

5*5,34+34,1000/300,length of 'uvpce',display only month of systemdate

:- select 5*5 "multi"from dual;

:- select 5*5 as multi from dual;

:- select 34+34 as add from dual;

:- select 1000/300 as div from dual;

:- select length('uvpce') "name" from dual;

10. Find the date,20 days before today's date.

:- select sysdate-20 "date" from dual;

Function Based Queries.

1. Find the total transaction amount of account holder from transaction table.

select sum(amt) "Total Amount" from

transaction;

2. Find minimum amount of transaction.

select min(amt)"minimum"from transaction;

3. Find maximum amount of transaction.

select max(amt)"maximum"from transaction;

4. Count the total account holders.

select count(acc_no)"total account holder"from transaction;

5. Count only those records whose made of payment is 'cash'.

select count(acc_no)"total account holder"from transaction where mode_of_pay='cash';

6. Count only those records whose transaction made in the month of 'MAY'.

select count(acc_no)"transaction"from transaction where date='

7. Find the average value of transaction.

select avg(amt)"average"from transaction;

8. Display the result of 4 rest to 4.

```
select power(4,4) from dual;
```

9. Find the square root of 25.

```
select sqrt(25)"root"from dual;
```

10. Write the query for the following Function.

LOWER,INITCAP,UPPER,SUBSTR,LENGTH,LTRIM,RTRIM,LPAD,RPAD.

```
select LOWER ('ABHI') from dual;

select INITCAP('abhi') from dual;

select upper('abhiraj')from dual;

select SUBSTR('abhiraj',2,5) from dual;

select LENGTH('abhiraj') from dual;

select LTRIM('akash','a')fromdual;

select RTRIM('akash','a')fromdual;

select LPAD('akash',10,'$')from dual;

select RPAD('akash','10','&')from dual;
```

Create a table:STUDENT

```
create table STUDENT (rollno varchar2(6), name varchar2(20),

branch varchar2(6), address varchar2(20));
```

1.Add PRIMARY KEY (roll no) and provide constraint name PRIM_rollno.

```
alter table STUDENT ADD constraint PRIM_rollno

        PRIMARY KEY(rollno);
```

2. Add NOT NULL constraint to name, branch for student table.

```
alter table student MODIFY(name constraint notnul1

NOT NULL, branch constraint notnul2 NOT NULL);
```

3. Add check constraint and check name is in capital letter.

```
Alter table student add constraint AB check (name = upper(name));
```

4. Drop the primary key.

```
alter table student drop constrant prim_ rollno;
```

5. Drop the constraint.

```
alter table student drop constraaint  notnul1;
```

Create a Table REGISTER.

create table REGISTER (rollno varchar2(6),name varchar2(20));

1.Provide foreign key references rollno of student table.

alter table register add constraint fk foreign key(rollno)

references student;

2. Add check constraint to check name's first letter is always capital.

alter table REGISTER ADD constraint ck1 check

(substr(name,1,1) =upper

(substr(name,1,1)));

3. Add NOT NULL constraint to name of register table.

alter table register modify(name constraint Av NOT NULL);

4. Drop foreign key of REGISTER table.

alter table register drop constraint fk;

5. Drop NOT NULL constraint.

alter table register drop constraint Av ;