**DBMS ASSIGNMENT- 4,5**

**(SOCIETY MANAGEMENT SYSTEM)**

* CREATE TABLE
* Owner

CREATE TABLE Owner(

Aadhar\_No NUMBER(12) NOT NULL,

Name VARCHAR2(20) NOT NULL,

Email VARCHAR2(20) NOT NULL,

RoleId NUMBER(2) NOT NULL,

CONSTRAINT owner\_pk PRIMARY KEY (Aadhar\_No),

CONSTRAINT role\_fk FOREIGN KEY (RoleId)

REFERENCES Role\_Owner(RoleId)

);

* Tenant

CREATE TABLE Tenant(

Aadhar\_No NUMBER(12) NOT NULL,

Name VARCHAR2(20) NOT NULL,

Email VARCHAR2(20) NOT NULL,

CONSTRAINT tenant\_pk PRIMARY KEY (Aadhar\_No)

);

* Employee

CREATE TABLE Employee(

Aadhar\_No NUMBER(12) NOT NULL,

Name VARCHAR2(20) NOT NULL,

Email VARCHAR2(20) NOT NULL,

RoleId NUMBER(2) NOT NULL,

CONSTRAINT tenant\_pk PRIMARY KEY (Aadhar\_No)

);

* Owner\_Mobile

CREATE TABLE Owner\_Mobile

(Owner\_MobileNo\_Id NUMBER NOT NULL,

MobileNo NUMBER(10) NOT NULL UNIQUE,

AadharNo NUMBER(12) NOT NULL,

CONSTRAINT owner\_mobile\_pk PRIMARY KEY(Owner\_MobileNo\_Id),

CONSTRAINT owner\_aadhar\_fk FOREIGN KEY(AadharNo)

REFERENCES Owner(AadharNo)

);

* Tenant\_Mobile

CREATE TABLE Tenant\_Mobile

(Tenant\_MobileNo\_Id NUMBER NOT NULL,

MobileNo NUMBER(10) NOT NULL UNIQUE,

AadharNo NUMBER(12) NOT NULL,

CONSTRAINT tenant\_mobile\_pk PRIMARY KEY(Tenant\_MobileNo\_Id),

CONSTRAINT tenant\_aadhar\_fk FOREIGN KEY(AadharNo)

REFERENCES Tenant(AadharNo)

);

* Employee\_Mobile

CREATE TABLE Employee\_Mobile

(Employee\_MobileNo\_Id NUMBER NOT NULL,

MobileNo NUMBER(10) NOT NULL UNIQUE,

AadharNo NUMBER(12) NOT NULL,

CONSTRAINT employee\_mobile\_pk PRIMARY KEY(Employee\_MobileNo\_Id),

CONSTRAINT employee\_aadhar\_fk FOREIGN KEY(AadharNo)

REFERENCES Employee(AadharNo)

);

* Flat

CREATE TABLE Flat

(FlatNo VARCHAR2(4) NOT NULL,

Owner\_AadharNo NUMBER(12),

Tenant\_AadharNo NUMBER(12),

CONSTRAINT flat\_pk PRIMARY KEY(FlatNo),

CONSTRAINT flat\_owner\_aadhar\_fk FOREIGN KEY(Owner\_AadharNo)

REFERENCES Owner(AadharNo),

CONSTRAINT flat\_tenant\_aadhar\_fk FOREIGN KEY(Tenant\_AadharNo)

REFERENCES Tenant(AadharNo)

);

* Maintainence\_Bill

CREATE TABLE Maintenance\_Bill

(PaymentNo VARCHAR2(20) NOT NULL,

Amount NUMBER(10) NOT NULL,

PayDate DATE,

Due\_Date DATE NOT NULL,

AadharNo NUMBER(12) NOT NULL,

FlatNo VARCHAR2(4) NOT NULL,

CONSTRAINT Maintenance\_Bill\_PK PRIMARY KEY(PaymentNo),

CONSTRAINT Maintenance\_Owner\_Aadhar\_FK FOREIGN KEY(AadharNo)

REFERENCES Owner(AadharNo),

CONSTRAINT Maintenance\_Flat\_No\_FK FOREIGN KEY(FlatNo)

REFERENCES Flat(FlatNo)

);

* Pay\_Rent

CREATE TABLE Pay\_Rent

(PaymentNo VARCHAR2(20) NOT NULL,

Amount NUMBER(10) NOT NULL,

PayDate DATE,

Due\_Date DATE NOT NULL,

Owner\_AadharNo NUMBER(12) NOT NULL,

Tenant\_AadharNo NUMBER(12) NOT NULL,

CONSTRAINT Pay\_Rent\_PK PRIMARY KEY(PaymentNo),

CONSTRAINT Pay\_Rent\_Owner\_Aadhar\_FK FOREIGN KEY(Owner\_AadharNo)

REFERENCES Owner(AadharNo),

CONSTRAINT Pay\_Rent\_Aadhar\_FK FOREIGN KEY(Tenant\_AadharNo)

REFERENCES Tenant(AadharNo)

);

* Previous\_Owner

CREATE TABLE Previous\_Owner(

Name VARCHAR2(20) NOT NULL,

AadharNo NUMBER(12) NOT NULL,

FlatNo VARCHAR2(4) NOT NULL,

CONSTRAINT Previous\_Owner\_PK PRIMARY KEY(AadharNo)

);

* Department

CREATE TABLE Department(

DepartmentId NUMBER(3) NOT NULL,

Department\_Name VARCHAR(20) NOT NULL,

Head\_AadharNo NUMBER(12) NOT NULL,

CONSTRAINT Department\_PK PRIMARY KEY(DepartmentId),

CONSTRAINT Department\_AadharNo\_FK FOREIGN KEY(Head\_AadharNo)

REFERENCES Owner(AadharNo)

);

* Department\_Programmes

CREATE TABLE Department\_Programmes(

ProgramNo NUMBER(20) NOT NULL,

ProgramDescription VARCHAR2(100) NOT NULL,

DepartmentId NUMBER(3) NOT NULL,

CONSTRAINT Department\_Programmes\_PK PRIMARY KEY(ProgramNo),

CONSTRAINT Department\_Programmes\_DepId\_FK FOREIGN KEY(DepartmentId)

REFERENCES Department(DepartmentId)

);

* Dependent

CREATE TABLE Dependent(

DependentId NUMBER(10) NOT NULL,

DependentName VARCHAR(20) NOT NULL,

DependentAadhar NUMBER(12) NOT NULL,

AadharNo NUMBER(12) NOT NULL,

CONSTRAINT Dependent\_PK PRIMARY KEY(DependentId),

CONSTRAINT Dependent\_AadharNo\_FK FOREIGN KEY(AadharNo)

REFERENCES Owner(AadharNo)

);

* Role\_Worker

CREATE TABLE Role\_Worker(

RoleId NUMBER(3) NOT NULL,

Role VARCHAR2(20) NOT NULL,

CONSTRAINT Role\_Worker\_PK PRIMARY KEY(RoleId)

);

* Role\_Owner

CREATE TABLE Role\_Owner(

RoleId NUMBER(3) NOT NULL,

Role VARCHAR2(20) NOT NULL,

CONSTRAINT Role\_Owner\_PK PRIMARY KEY(RoleId)

);

* Role\_Employee

CREATE TABLE Role\_Employee(

RoleId NUMBER(3) NOT NULL,

Role VARCHAR2(20) NOT NULL,

CONSTRAINT Role\_Employee\_PK PRIMARY KEY(RoleId)

);

* Complaint\_Status

CREATE TABLE Complaint\_Status(

StatusId NUMBER(3) NOT NULL,

Status VARCHAR2(20) NOT NULL,

CONSTRAINT Comlaint\_Status\_PK PRIMARY KEY(StatusId)

);

* Workers

CREATE TABLE Workers(

AadharNo NUMBER(12) NOT NULL,

Shift NUMBER(2) NOT NULL,

Area VARCHAR2(10) NOT NULL,

DepartmentId NUMBER(3) NOT NULL,

RoleId NUMBER(3) NOT NULL,

CONSTRAINT Worker\_PK PRIMARY KEY(AadharNo),

CONSTRAINT Worker\_DepartmentId\_FK FOREIGN KEY(DepartmentId)

REFERENCES Department(DepartmentId),

CONSTRAINT Worker\_RoleId\_FK FOREIGN KEY(RoleId)

REFERENCES Role\_Worker(RoleId),

CONSTRAINT Worker\_AadharNo\_FK FOREIGN KEY(AadharNo)

REFERENCES Employee(AadharNo)

);

* Complaints

CREATE TABLE Complaints(

ComplaintNo NUMBER(10) NOT NULL PRIMARY KEY,

Complaint\_Date DATE NOT NULL,

Complaint\_Description VARCHAR2(100) NOT NULL,

AadharNo NUMBER(12) NOT NULL,

StatusId NUMBER(3) NOT NULL,

CONSTRAINT Complaints\_AadharNo\_FK FOREIGN KEY(AadharNo)

REFERENCES Employee(AadharNo),

CONSTRAINT Complaints\_Status\_FK FOREIGN KEY(StatusId)

REFERENCES Complaint\_Status(StatusId)

);

* Staff

CREATE TABLE Staff(

AadharNo NUMBER(12) NOT NULL PRIMARY KEY,

Designation VARCHAR2(20) NOT NULL,

CONSTRAINT Staff\_FK FOREIGN KEY(AadharNo)

REFERENCES Employee(AadharNo)

);

* Parking

CREATE TABLE Parking (

ParkingNo VARCHAR2(10) NOT NULL,

VehicleRegNo VARCHAR2(20),

FlatNo VARCHAR2(2) NOT NULL,

CONSTRAINT Parking\_PK PRIMARY KEY(ParkingNo),

CONSTRAINT Parking\_FlatNo\_FK FOREIGN KEY (FlatNo)

REFERENCES Flat(FlatNo)

);

* ALTER TABLE
* Employee-Rename Column

ALTER TABLE Employee RENAME COLUMN Aadhar\_No to AadharNo;

* Employee-Add Constraint

ALTER TABLE Employee

ADD

(CONSTRAINT Employee\_Role\_FK FOREIGN KEY(RoleId)

REFERENCES Role\_Employee(RoleId));

* Maintainance\_Bill-Add Column

ALTER TABLE Maintainance\_Bill ADD Paid NUMBER(1) NOT NULL;

* VIEWS
* Department\_Head

CREATE VIEW Department\_Head AS

SELECT \* FROM Owner

INNER JOIN Department

ON Owner.AadharNo = Department.Head\_AadharNo;

* Department\_Programmes\_Details

CREATE VIEW Department\_Programmes\_Details AS

SELECT Department.Department\_Name,Department\_Programmes.ProgramNo,Department\_Programmes.ProgramDescription

FROM Department

INNER JOIN Department\_Programmes

ON Department.DepartmentId=Department\_Programmes.DepartmentId;

* Maintainance\_Not\_Paid

CREATE VIEW Maintainance\_Not\_Paid AS

SELECT \* FROM Maintainance\_Bill

WHERE (SYSDATE>Due\_Date) AND Paid=0;

* Rent\_Not\_Paid

CREATE VIEW Not\_Paid\_Rent AS

SELECT Pay\_Rent.PaymentNo,Tenant.AadharNo,Tenant.Name,Pay\_Rent.Owner\_AadharNo,Pay\_Rent.Due\_Date,Pay\_Rent.Amount FROM Tenant

INNER JOIN Pay\_Rent

ON Tenant.AadharNo=Pay\_Rent.Tenant\_AadharNo AND Pay\_Rent.Paid=0;

* SEQUENCES
* Role\_Owner\_seq (similarly for other primary keys)

CREATE SEQUENCE Role\_Owner\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

* INSERT
* Role\_Owner(similarly for Role\_Workers,Role\_Employees)

INSERT INTO Role\_Owner

(RoleId,Role)

VALUES

(Role\_Owner\_seq.NEXTVAL,'Resident');

INSERT INTO Role\_Owner

(RoleId,Role)

VALUES

(Role\_Owner\_seq.NEXTVAL,'Chairman');

INSERT INTO Role\_Owner

(RoleId,Role)

VALUES

(Role\_Owner\_seq.NEXTVAL,'Secretary');

INSERT INTO Role\_Owner

(RoleId,Role)

VALUES

(Role\_Owner\_seq.NEXTVAL,'Department Head');

* Owner(similarly for Tenants,Workers,Employees)

INSERT INTO Owner

(AadharNo,Name,Email,RoleId)

VALUES

(621536158720,'Mahesh Yadav','m.yadav@gmail.com',3);

Repeat above type queries to add more rows.

* Flat

INSERT ALL

INTO FLAT(FLATNO,OWNER\_AADHARNO,TENANT\_AADHARNO) VALUES('G101',121837129812,182732984736)

INTO FLAT(FLATNO,OWNER\_AADHARNO,TENANT\_AADHARNO) VALUES('G102',NULL,NULL)

SELECT \* FROM DUAL;

Similarly insert other rows.

* Maintainance\_Bill(Similarly for Pay\_Rent)

INSERT INTO MAINTAINANCE\_BILL

(PAYMENTNO,AMOUNT,PAYDATE,AADHARNO,FLATNO,PAID,DUE\_DATE)

VALUES

(MB\_SEQ.NEXTVAL,'5000',TO\_DATE('1-DEC-2016'),'121837129812','G101','1',TO\_DATE('5-DEC-2016'));

Similarly insert other rows.

* Department

INSERT INTO DEPARTMENT

(DEPARTMENTID,DEPARTMENT\_NAME,HEAD\_AADHARNO)

VALUES

(DEPID\_SEQ.NEXTVAL,'Security',121837129812);

Simlarly insert other rows.

* OBSERVATION
* We can insert rows into table one by one or all together.
* To use sequences on primary key, we need to insert records one by one.
* The above types of insert statements can be used to insert records in tables.
* INDICES

Indices can be created using the following syntax-

CREATE *index\_name*

ON *table\_name(column\_name)*;

Index are useful on columns which are frequently used for filtering and searching as it reduces the query processing time.

Here, we do not require secondary index on tables as the primary index which are on primary key are sufficient.

* SOME DML QUERIES
* SELECT \*

FROM OWNER

WHERE ROLEID=1;

Returns all Owner who are Residents.

* SELECT \*

FROM MAINTAINANCE\_BILL

ORDER BY PAID;

Returns all Maintainance bill, all not paid then paid.

* SELECT \*

FROM PAY\_RENT

HAVING MIN(AMOUNT) < 15000

ORDER BY PAYMENTNO;

Returns all Rent Records which have minimum of amount less than 15000 and orders it by paymentno.

* SELECT SUM(AMOUNT) AS TOTAL\_MAINTAINANCE

FROM MAINTAINANCE\_BILL;

Returns all total amount earned from collecting maintainance bill.

* SELECT MOBILENO

FROM TENANT\_MOBILE

WHERE AADHARNO=625158721929 AND ROWNUM <= 2;

Returns not more than two mobile no of the particular tenant.

* SELECT TENANT.AADHARNO,TENANT.NAME,TENANT.EMAIL,PAY\_RENT.AMOUNT,PAY\_RENT.DUE\_DATE

FROM TENANT

INNER JOIN PAY\_RENT

ON TENANT.AADHAR\_NO=PAY\_RENT.TENANT\_AADHARNO;

Returns all the Pay\_Rent records having the Tenant Details joined with it.