CS F212 DBMS

PROPERTY RENTAL AGENCY_ER DIAGRAM AND RELATIONAL SCHEMA

PROPERTY RENTAL AGENCY

ENTITIES MENTIONED:

- USER
- PROPERTY
- PROPERTY RENT HISTORY
- TENANT
- OWNER
- DBA
- MANAGER

In DBMS, there are several constraints that are important to consider when modeling data using Entity-Relationship (ER) or Enhanced Entity-Relationship (EER) diagrams, and when mapping those diagrams to relational schemas. These constraints with the instructions given in question include:

- Entity integrity constraints :
  - Each property has an ID which is unique.
  - A user has adhar ID which is unique.

- Referential integrity constraints :
  - Each property has an owner

- Cardinality constraints :
  - A user can have zero to many phone numbers.
  - A user can upload zero to many properties.
  - A user can be a tenant for zero to many properties.

- Participation constraints :
  - A property may not have a tenant if it is not rented.

- Attribute constraints :
  - Age attribute of a user must have a value.
  - Address attribute of a user must include door#, street, city, state, and PIN code.
  - Rent per month, start_date, end_date, yearly hike in rent(in %), agency commission, and other info must have values.

➢ General Constraint:

- A user can play both owner and tenant roles.
- A property can be a residential property (independent-house/flat) or a commercial property (shop or warehouse).
- Every property is entered into the system with the details like available from which date, available till what date, rent pm, %ge of annual hike in rent, total area(not null), plinth area(not null), number of bed-rooms(if residential), number of floors, year of construction (not null), locality, address, other facilities etc.
- All users will have login credentials.

➢ Domain constraints:

- The age attribute of a user is restricted to a range of values.
- The available from and available till dates for a property must be valid dates.

➢ Unique constraints:

- Each property has an ID which is unique.
- Each user has an adharID which is unique.

➢ Check constraints:

- The available till date for a property must be greater than or equal to the available from date.

ENTITIES WITH ATTRIBUTES

❑ USER

- ADHAR_ID (PK)
- NAME
- AGE
- DOOR_NO
- STREET
- CITY
- STATE
- PINCODE
- LOGIN_USERNAME
- LOGIN_PASSWORD

❑ PROPERTY

- PROPERTY_ID (PK)
- OWNER_ID (FK)
- AVAIL FROM DATE
- AVAIL TILL DATE
- RENT_PM
- ANNUAL RENT %HIKE
- TOTAL AREA
- PLINTH AREA
- NUM_BEDROOMS
- NUM_FLOORS
- YEARS OF CONSTRUCTION
- LOCALITY
- ADDRESS
- OTHER_FACILITIES

❑ PROP_RENT HISTORY

- PROPERTY_ID (PK,FK)
- TENANT_ID (FK)
- START DATE (PK)
- END DATE
- RENT_PM
- ANNUAL RENT %HIKE
- AGENCY COMMISSION %

❑ OWNER

- ADHAR_ID (PK,FK)
- OWNER_ID (PK)
- TOTAL PROPS UPLOADED

❑ TENANT

- ADHAR_ID (PK,FK)
- TENANT_ID (PK)
- TOTAL PROPS RENTED

❑ MANAGER

- MANAGER_ID
- M_LOGIN USERNAME
- M_LOGIN PASSWORD

❑ DBA

- LOGIN_USERNAME
- LOGIN PASSWORD

1. CREATING TABLES

USERS:

```
create table users(adharid int primary key,password varchar(30),age int,name varchar(30),door int,street
varchar(30),city varchar(30),state_ varchar(30),pincode int);
```

PROPERTY:

```
create table property( floors  int, owner_id int,property_id  int primary key,hike  int, year_of_construction
int,start_date  DATE, end_date  DATE,total_area  int,plinth_area  int,rent_per_month  int,locality  varchar(30),
rooms  int, address VARCHAR(30));
```

RENTAL:

```
Create table rental(property_id int,tenant_id int,start_date date,end_date date,hike int,rent int,commision int,
 primary key(tenant_id,property_id));
```

```
Alter table rental add constraint rent_fk foreign key(tenant_id) references users(adharid);
Alter table rental add constraint rent_fk2 foreign key(property_id) references property(property_id);
Alter table property add constraint ppr_fk foreign key(owner_id) references users(adharid);
```

2. INSERTING VALUES

```
INSERT INTO users VALUES (10001, 'pass123', 24, 'John Smith', 101, 'Main Street', 'New York', 'NY', 10001);
INSERT INTO users VALUES (10002, 'qwerty', 31, 'Emma Watson', 502, 'Park Lane', 'London', 'UK', 12345);
INSERT INTO users VALUES (10003, 'passpass', 45, 'Michael Jordan', 23, 'North Street', 'Chicago', 'IL', 60601);
INSERT INTO users VALUES (10004, 'iloveyou', 29, 'Adele Smith', 201, '5th Avenue', 'New York', 'NY', 10002);
INSERT INTO users VALUES (10005, 'hello123', 22, 'Sarah Williams', 302, 'Oxford Street', 'London', 'UK', 56789);
INSERT INTO users VALUES (10006, 'mypassword', 27, 'Peter Parker', 601, 'Queens Boulevard', 'New York', 'NY', 11221);
INSERT INTO users VALUES (10007, 'password123', 33, 'Jackie Chan', 301, 'China Town', 'Los Angeles', 'CA', 90001);
INSERT INTO users VALUES (10008, 'football', 28, 'Cristiano Ronaldo', 801, 'Madrid Avenue', 'Madrid', 'Spain', 28001);
INSERT INTO users VALUES (10009, 'admin123', 40, 'Mark Zuckerberg', 501, 'Palo Alto Road', 'Palo Alto', 'CA', 94301);
INSERT INTO users VALUES (10010, 'letmein', 36, 'George Lucas', 501, 'Lucasfilm Street', 'San Francisco', 'CA', 94129);
INSERT INTO users VALUES (10011, 'test123', 25, 'Kate Hudson', 701, 'Beverly Hills', 'Los Angeles', 'CA', 90210);
INSERT INTO users VALUES (10012, 'password', 41, 'Arnold Schwarzenegger', 1001, 'Main Street', 'Santa Monica', 'CA', 90401);
INSERT INTO users VALUES (10013, 'superman', 32, 'Henry Cavill', 901, 'Westminster', 'London', 'UK', 45678);
INSERT INTO users VALUES (10014, 'welcome123', 27, 'Miley Cyrus', 301, 'Hollywood Hills', 'Los Angeles', 'CA', 90068);
INSERT INTO users VALUES (10015, 'pass1234', 29, 'Justin Bieber', 501, 'Rodeo Drive', 'Beverly Hills', 'CA', 90212);
INSERT INTO users VALUES (10016, 'secret123', 35, 'Brad Pitt', 601, 'Sunset Boulevard', 'Los Angeles', 'CA', 90028);
INSERT INTO users VALUES (10017, 'qwerty123', 30, 'Emma Stone', 802, 'Laurel Canyon', 'Los Angeles', 'CA', 90046);
INSERT INTO users VALUES (10018, '123456', 26, 'Taylor Swift', 401, 'Nashville Road', 'Nashville', 'TN', 37203);
INSERT INTO users VALUES (10019, '123896', 29, 'beyonce', 404, 'Nashville Road', 'Nashville', 'TN', 37203);
INSERT INTO property VALUES (3, 10004,20001, 5, 2005, date'2023-05-01',date '2024-05-01', 1500, 1200, 2000, 'Midtown', 4, '101 Park Ave’);

 INSERT INTO property VALUES (1, 10006,20002, 2, 1995, date '2023-05-01',  date '2024-05-01', 800, 600, 1500, 'Downtown', 2, '20 W 34th St');

 INSERT INTO property VALUES (4, 10010, 20003, 7, 2010, date  '2023-05-01', date  '2024-05-01', 2000, 1600, 3000, 'Westside', 5, '1001 5th Ave');

 INSERT INTO property VALUES (2, 10004, 20004, 3, 1985,date  '2023-05-01', date '2024-05-01', 1200, 900, 1200, 'Eastside', 3, '10 E Lake St');

 INSERT INTO property VALUES (5, 10012, 20005, 6, 2015,date  '2023-05-01', date '2024-05-01', 2500, 2000, 3500, 'Uptown', 6, '555 10th Ave');

 INSERT INTO property VALUES (3, 10006, 20006, 4, 2000,date  '2023-05-01',date  '2024-05-01', 1500, 1100, 1800, 'Midtown', 4, '45 W 30th St');

 INSERT INTO property VALUES (2, 10004, 20007, 3, 1990,date  '2023-05-01',date  '2024-05-01', 1000, 800, 1000, 'Downtown', 2, '35 5th Ave');

 INSERT INTO property VALUES (4, 10010, 20008, 5, 2005, date  '2023-05-01',date   '2024-05-01', 1800, 1400, 2500, 'Westside', 4, '1200 6th Ave');

 INSERT INTO property VALUES (1, 10012, 20009, 2, 1998,date  '2023-05-01',date  '2024-05-01', 900, 700, 1300, 'Eastside', 2, '15 E 23rd St');

 INSERT INTO property VALUES (3, 10015, 1001, 50, 2010, date'2022-01-01',date '2024-01-01', 2000, 1500, 3000, 'Downtown', 4, '123 Main Street');

INSERT INTO property VALUES (2,10002, 1002, 30, 2005,date  '2022-02-01', date '2024-02-01', 1500, 1000, 2000, 'Suburb', 3, '456 Oak Avenue’);


INSERT INTO property VALUES (4,10017, 1003, 70, 2015,date  '2022-03-01', date '2024-03-01', 3000, 2000, 4000, 'Beachfront', 5, '789 Ocean Boulevard');
```

3. Write a Stored Procedure to enter a record for property. Name of the stored procedure is ***InsertPropertyRecord*** (with necessary arguments/parameters for the attributes of the record).

```
CREATE OR REPLACE PROCEDURE insertproperty(
    floors IN NUMBER,
    owner_id IN NUMBER,
    property_id IN NUMBER,
    hike IN NUMBER,
    year_of_construction IN NUMBER,
    start_date IN DATE,
    end_date IN DATE,
    total_area IN NUMBER,
    plinth_area IN NUMBER,
    rent_per_month IN NUMBER,
    locality IN VARCHAR,
    rooms IN NUMBER,
    address IN VARCHAR)
AS
BEGIN
    INSERT INTO property  VALUES (floors, owner_id, property_id, hike, year_of_construction, start_date, end_date, total_area,plinth_area,
rent_per_month, locality, rooms, address);
End;
/
```

4. Write a Stored Procedure to display the properties (with all details) for a given ownerID. Name of the stored procedure is ***GetPropertyRecords*** (with Owner ID as the parameter).

```
CREATE OR REPLACE PROCEDURE GetPropertyRecords(owner IN NUMBER) AS
BEGIN
  FOR prop IN (SELECT * FROM property WHERE owner_id = owner)
  LOOP
    DBMS_OUTPUT.PUT_LINE(‘Property ID: ' || prop.property_id);
    DBMS_OUTPUT.PUT_LINE(‘Locality: ' || prop.locality);
    DBMS_OUTPUT.PUT_LINE(‘Rooms: ' || prop.rooms);
    DBMS_OUTPUT.PUT_LINE(‘Address: ' || prop.address);
    DBMS_OUTPUT.PUT_LINE(‘Start Date: ' || TO_CHAR(prop.start_date, 'DD-MON-YYYY'));
    DBMS_OUTPUT.PUT_LINE(‘End Date: ' || TO_CHAR(prop.end_date, 'DD-MON-YYYY'));
    DBMS_OUTPUT.PUT_LINE(‘Rent Per Month: ' || prop.rent_per_month);
    DBMS_OUTPUT.PUT_LINE(‘Total Area: ' || prop.total_area);
    DBMS_OUTPUT.PUT_LINE(‘Plinth Area: ' || prop.plinth_area);
    DBMS_OUTPUT.PUT_LINE(‘Hike: ' || prop.hike);
    DBMS_OUTPUT.PUT_LINE(‘Year of Construction: ' || prop.year_of_construction);
    DBMS_OUTPUT.PUT_LINE(‘---');
  END LOOP;
END;
/
```

5. Write a stored procedure to Print the tenant details for a given property. Name of the stored procedure is ***GetTenantDetails*** (with property id as the argument).

```
CREATE OR REPLACE PROCEDURE GetTenantDetails(propertyw IN NUMBER) AS
BEGIN
  FOR tenant IN (SELECT u.* FROM users u JOIN rental t ON u.adharid = t.tenant_id WHERE t.property_id =
propertyw)
  LOOP
    DBMS_OUTPUT.PUT_LINE('Tenant ID: ' || tenant.adharid);
    DBMS_OUTPUT.PUT_LINE('Name: ' || tenant.name);
    DBMS_OUTPUT.PUT_LINE('Email: ' || tenant.age);
    DBMS_OUTPUT.PUT_LINE('Phone: ' || tenant.door);
    DBMS_OUTPUT.PUT_LINE('Address: ' || tenant.pincode);
    DBMS_OUTPUT.PUT_LINE('---');
  END LOOP;
END;
/
```

6. Write a Stored Procedure to enter a record for a new User. Name of the stored procedure is ***CreateNewUser*** (with necessary arguments/parameters for the attributes like- user name, address, login credentials etc. of the record).

```
CREATE OR REPLACE PROCEDURE CreateNewUser(
    p_adharid IN users.adharid%TYPE,
    p_password IN users.password%TYPE,
    p_age IN users.age%TYPE,
    p_name IN users.name%TYPE,
    p_door IN users.door%TYPE,
    p_street IN users.street%TYPE,
    p_city IN users.city%TYPE,
    p_state IN users.state_%TYPE,
    p_pincode IN users.pincode%TYPE
) AS
BEGIN
    INSERT INTO users(adharid, password, age, name, door, street, city, state_, pincode)
    VALUES (p_adharid, p_password, p_age, p_name, p_door, p_street, p_city, p_state, p_pincode);
    COMMIT;
END;
/
```

7. Write a stored procedure that prints details of properties available in a city/locality. Name of the stored procedure is ***SearchProperyForRent*** (with city/locality as the argument).

```
CREATE OR REPLACE PROCEDURE SearchPropertyForRent(city IN VARCHAR2)
  IS
  cursor property_cursor is
  SELECT *
  FROM property
  WHERE locality = city;
  property_rec property_cursor%ROWTYPE;
  BEGIN
  OPEN property_cursor;
  LOOP
  FETCH property_cursor INTO property_rec;
  EXIT WHEN property_cursor%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(property_rec.property_id || ' ' || property_rec.address);
  END LOOP;
  CLOSE property_cursor;
END;
/
```

8. Write a stored procedure to print the rent history for a given property. Name of the stored procedure is ***GetRentHistory*** (with property id as the argument).

```
CREATE OR REPLACE PROCEDURE GetRentHistory(propertyw IN NUMBER) AS
BEGIN
   FOR property IN (
      SELECT start_date, end_date
      FROM property
      WHERE property_id = propertyw)
      LOOP
      DBMS_OUTPUT.PUT_LINE('Start date: ' || property.start_date || ', End date: ' ||   property.end_date);
   END LOOP;
END;
/
```

```
Exec CreateNewUser(100198, 'pass123', 24, 'John Smith', 101, 'Main Street', 'New York', 'NY', 10001);

 Exec insertproperty (3, 10004, 2005, 5, 2005, date '2023-05-01', date '2024-05-01', 1500, 1200, 2000,
'Midtown', 4, 'ParkAve’);

Exec  GetTenantDetails(20006);

Exec GetRentHistory(20008);

Exec GetPropertyRecords(10004);

Exec SearchPropertyForRent('Eastside');
```

< THANK YOU >