Project Phase - I

Apartment Management System

Team Name : cheems **Team No :** 9
Team Members : 2020101044, 2020115008, 2020101056

Introduction

The mini-world chosen is an Apartment Complex.

With increase in urbanisation the need for large apartment complexes in big and small towns of India has only increased and apartments do need good maintenance and management and hence a good database.

In this mini-world, information about the apartment owners and also the information about various activities, amenities,and utilities of the apartment complex like water Supply , Electricity, Lifts, Parking, Security etc, Common Areas and Amenities like gym, parks, play areas, swimming pool, clubhouse, and Social Events, Accounts etc..etc. is to be maintained.

For the purpose of good maintenance of the Apartment Complex every Apartment complex legally forms an owners association which in turn elects a committee from the members to oversee the various maintenance activities . The committee may appoint professional property managers too depending on the size of the Apartment complex.

A database of various entities like income and expenditure,list of owners and their contact details, tenants, maintenance staff, equipment like office equipment, sports equipment, recreational or clubhouse equipment etc has to be maintained.

Maintenance of a property involves money and so we need to maintain information about the income, expenditure and financial status of the association.

Of course there will be complaints from people when there are problems in the apartment (like tap leakage, electricity issue, disturbance from neighbours ..etc..) and so a database of the complaints and their status has to be maintained so that the problems get resolved as soon as possible.

Purpose

The database of this mini-world aims to maintain a database of the

- Owners
- Tenants
- Visitors
- Committee members
- Staff members
- Complaints
- Accounts
- Common Amenities and Equipment

Users

The users of the database are as follows

- Elected Apartment Committee members.
- Owners Association members
- Property Managers
- Tenants
- Anyone else the association gives access to

Applications

Some important Applications on the database are

Applications to Access, Modify, Search and Sort

- Records of owners and or tenants
- Records of Past and present employees

- Records of material or equipment belonging to the Apartment complex
- Records of the accounts
- Records of daily Visitors / Delivery boys

Communication Applications

- Application to send e receipts to tenants/owners who paid maintenance charges
- Application to send notices to owners and tenants with dues
- Application to forward a complaint from owner/tenant to the relevant staff member.
- Application to send complaint status messages to the respective owner or tenant

Statistical applications

- Income and expenditure graphs

Database Requirements

ENTITIES

1. APARTMENT_COMPLEX
    a. Block name (VARCHAR(30))
    b. Block Manager id (Primary key) (VARCHAR(20))
    c. Block Manager name (VARCHAR(30))
    d. phone number (Alternate key) (Multivalued :{phoneNo1, phoneNo2,...} BIGINT)
    e. Email id (Alternate key) (BIGINT)
    f. Block Color (Multivalued : {VARCHAR(20)})

2. APARTMENT (weak entity, foreign key = block name)
    a. Apartment number (INT)
    b. Apartment size (FLOAT)
    c. Owner id (VARCHAR(20))
    d. Owner name (VARCHAR(30))
    e. Phone number (Multivalued :{phoneNo1, phoneNo2,...} BIGINT)
    f. Email id (VARCHAR(30))
    g. Monthly maintenance Charges (FLOAT)
    h. Owner since (composite attribute : {DAY, MONTH, YEAR})

3. TENANT (weak entity, foreign key = block name)
    a. Apartment number (INT)
    b. Tenant Name (VARCHAR(30))
    c. Phone number (Multivalued :{phoneNo1, phoneNo2,...} BIGINT)
    d. Email_id (VARCHAR(30))
    e. Owner_id (VARCHAR(20))
    f. Tenant since (composite attribute : {DAY, MONTH, YEAR})

4. OWNERSHIP_HISTORY (weak entity, foreign key = block name)
    a. Apartment number
    b. Owner name
    c. Owner_id
    d. Phone number
    e. Email_id
    f. Ownership period
        (composite attribute : {{DAY, MONTH, YEAR} to {DAY, MONTH, YEAR}})

5. TENANT_HISTORY (weak entity, foreign key = block name)
    a. Apartment number
    b. Tenant Name
    c. Phone number
    d. Email id
    e. Owner id
    f. Tenancy period

6. COMMITTEE_TENURE
    a. Committee period (mm/yy ...mm/yy)
    b. Committe_id (INT)

7. COMMITTEE_HISTORY (weak entity, foreign key = Committee_id)
    a. Member name
    b. Member id
    c. Email_id
    d. Phone number

8. COMPLAINTS (weak entity, foreign key = block name)
    a. Complainant (VARCHAR (30))
    b. Apartment number (INT)
    c. Complaint (TINYTEXT (255))
    d. Date of complaint (composite attribute : {{DAY, MONTH, YEAR})
    e. Status (ENUM attribute : {pending, inprogress, resolved})
    f. Sub classes:
        i. Plumbing
        ii. Electrical
        iii. Lift
        iv. Others
            1. Complaint type (TINYTEXT (30))

9. COMPLAINTS_HISTORY (weak entity, foreign key = block name)
    a. Complainant
    b. Apartment number
    c. Complaint
    d. Date of complaint
    e. Resolver name
    f. Resolver phone number
    g. Resolver email id
    h. Sub classes:
        i. Plumbing
        ii. Electrical
        iii. Lift
        iv. Others
            1. Complaint type

10. MMC (monthly maintenance charges) (weak entity, foreign key = block name)
    a. Apartment number (INT)
    b. Paid upto (composite attribute : {{DAY, MONTH, YEAR})

11. ACCOUNTS
    a. Date (composite attribute : {{DAY, MONTH, YEAR})
    b. Income (derived type) (FLOAT)
    c. Total expenditure (derived type) (FLOAT)

12. EXPENDITURE
    a. Amount spent (FLOAT)
    b. Date (composite attribute : {{DAY, MONTH, YEAR})
    c. Sub classes:
        i. Renovations and Repairs
            1. Type of renovation or repair (TINYTEXT (255))
        ii. Purchase of new equipment
            1. Name of new equipment (VARCHAR(20))
        iii. Festive or special events
            1. Name of event (TINYTEXT (255))
        iv. Salaries
            1. Monthly salaries (derived type) (FLOAT)
            2. Advances (FLOAT)
            3. Bonus (FLOAT)

13. EMPLOYEE
    a. Employee_id
    b. Employee name
    c. Email id
    d. Phone number
    e. Salary (FLOAT)
    f. Recruitment date
    g. Sub classes:
        i. Managers
        ii. Workers
            1. Type of work (TINYTEXT (255))

14. EMPLOYEE_HISTORY
    a. Employee_id
    b. Employee name
    c. Email id
    d. Phone number
    e. Salary
    f. Period of service (mm/yy --mm/yy)
    g. Remarks (TINYTEXT (255))
    h. Sub classes:
        i. Managers
        ii. Workers
            1. Type of work

15. VISITORS
    a. Name (VARCHAR(255))
    b. Phone number (BIGINT)
    c. Date (composite attribute : {{DAY, MONTH, YEAR})
    d. purpose (TINYTEXT (255))

WEAK ENTITY:

1. APARTMENT (foreign key = block name)
2. TENANT (foreign key = block name)
3. OWNERSHIP_HISTORY (foreign key = block name)
4. TENANT_HISTORY (foreign key = block name)
5. COMMITTEE_HISTORY (foreign key = committee_id)
6. COMPLAINTS (foreign key = block name)
7. COMPLAINTS_HISTORY (foreign key = block name)
8. MMC (foreign key = block name)

RELATIONSHIPS:

1. Managers **manage** Employees
    a. Min-max constraint :
        i. Managers (1 .. n) -------- **manage** ------ Employees (1 .. 1)
    b. Binary relationship
    c. Total participation
    d. Manager manages minimum 1 and maximum n employees.
    e. One Employee can be in a relationship with only one manager.

2. Committee **pays** Employees.
    a. Min-max constraint :
        i. Committee (1 … n) -------- **pays** ------- Employees (1 ...1)
    b. Binary relationship
    c. Total participation
    d. One committee pays all employees.

3. Owner **owns** the apartment
    a. Min-max constraint :
        i. Owner (1…. n) --------- **owns** ------- apartment (1 ..1)
    b. Binary relationship
    c. A apartment must be owned by a owner (total relationship)
    d. One owner can own many apartments.

4. Tenant **lives** in the apartment
    a. Min-max constraint :
        i. Tenant (1 .. 1) --------- **lives** ---------- apartment (1 ...1)
    b. Binary relationship
    c. Tenant lives in only one apartment.

5. Apartment complex **contains** apartment Blocks
    a. Min-max constraint :
        i. Apartment complex (1 .. n) **contain** apartments blocks (1….1)
    b. Binary relationship.
    c. Total participation
    d. An apartment is related to only one block
    e. An Apartment complex can have many apartment blocks

N>=3 RELATIONSHIPS

1. Tenant **lives** in the apartment **owned** by the owner .
    a. Min-max constraints :
    Tenant (1...1) ----- **lives** ---- apartment (1...1) ----- **owned** ---- owner (1...1)
    b. Ternary relationship
    c. Every tenant has one owner. (total participation)
    d. Every tenant lives in one apartment. (total participation)
2. Committee **spends** income **as** an expenditure to **pay** Employees.
    a. Min-max constraints :

Committee (1...1) ------ **spends** ----- income (1..1) ------- **as** -------
-------- expenditure (1...1) ------- **pay** ------ Employees (1...1)

Functional Requirements

Retrievals

1. Selection:

    a. Select all owners and their email ids as a tuple from the apartments table.

2. Projection:

    a. Select all apartment numbers whose owners did not pay the monthly maintenance charges for this month yet.
    b. Select all apartment numbers which are owned by tenants.

3. Aggregate:

    a. Find the total expenditure of the current month.
    b. Find the income of the current month.
    c. Average expenditure of the current year.

4. Search:

    a. Get all apartment owners' names who live in their apartments.
    b. Search for names of owners which matches text given as input while searching in an application.

5. Analysis:

    a. Show the plot of expenditure
        i. Bar graph with x-axis month and y-axis expenditure.
        ii. Pie chart showing the income distributed.
    b. Show the plot of complaints as a graph .
        i. Bar graph with month and number of complaints in that month.
        ii. Pie chart showing details of different complaints.

Modifications

1. insert:

    a. Insert a new owner into the apartment table and move the previous owner to the owner history table.
    b. Add a new tenant to the tenants table.
    c. Insert a new employee, his/her work type, salary,etc.
    d. Add this month's accounting information.
    e. Take a new complaint along with necessary details.
    f. Insert a new visitor into the visitors table.

2. Delete:

    a. Remove a complaint from the complaints table.
    b. Remove an employs from employs table.
    c. Remove a tenant from the tenants table.

3. Update:

    a. Update "paid upto" in MMC table.
    b. Update the monthly maintenance charges in the apartment table.
    c. Update complaints' status.

BONUS:

1. one entity type with candidate key, super key, alternate key.
    a. **APARTMENT_COMPLEX**
    This entity type has the following keys

Set of super keys (all possible keys):
{ Block Manager id } ,{ phone number } , { Email id },
{ Block Manager id , phone number },
{ phone number, Email id },
{ Block Manager id ,Email id },
{ Block Manager id , phone number, Email id }

candidate key (minimal super keys):
Block Manager id , phone number, Email id

alternate key (other than primary and belong to candidate key):
phone number, Email id

2. For the relationships with degree n where n > 2 , it can be modeled differently as n binary relationships.
    a. The example below is from one of the n>=3 relationships used in the above database requirements mini-word

    n=3 relationship :
    Tenant **lives** in the apartment **owned** by the owner.

    There are three binary relations
    - tenant **lives** in apartment,
    - Apartment is **owned** by owner
    - Tenant **has** a owner

    Of Course it is not always possible to model n>=3 relationships using n binary relationships.

3. Relationship type with the same participating entity type in distinct roles
    a. **manage:**
    A Relationship between employee (manager) and employee (worker).