Student: Thanh Nam Tran
CMSC 461
Database Management Systems
Project 1 Milestone 3

Report of Pluto Reality Database

1. Review Milestone I feedbacks:

Fixed Email should be a multivalued attribute ✓

Fixed Date time (property viewings) should be multivalued attribute ✓

Fixed Name should be composite attribute ✓

Fixed lease should be a weak entity – should have no PK ✓ (For my implementation I need to have lease_id be a primary key.)

Added Property has a lease ✓

Added Associate shows property - Date & time ✓

Fixed Property ISA Residential commercial – Industrial – disjoint ✓

2. Updated the primary keys, foreign keys and related key constraints in this database:

Please read explanation for more detail on how I design the database, choices of primary/foreign key

person(person_id, last_name, first_name, telephone,address,city,state, zipcode)

PK: person_id

FK:

Explanation: Person entity holds information of `employees` and `clients`

employees(employee_id, roles, hired_date)

PK: employee_id

FK: employees(employee_id) -> person(person_id) ON DELETE CASCADE

Explanation: `Employees` hold role and hired_date of each employee. `employee_id` is uniquely distinct from one another, thus `employee_id` is a primary key.

employee_email(id, email)

PK:

FK: employee_email(id) -> employees(employee_id)

Explanation: multivalued attribute of email

supervisor(emp_id, spv_id)

PK: emp_id

FK: supervisor(emp_id) -> employees(employee_id)

supervisor(spv_id) -> employees(employee_id)

property_viewings(property_viewing_id, property_id, client_id, associate_id)

PK: property_viewing_id

FK:

property_viewings(associate_id) -> associates(id)

property_viewings(property_id) -> rental_properties(id)

property_viewings(client_id) -> client(client_id)

Explanation: Each time an associate with a client proceeds to view a property, a `property_viewing_id` is created. `property_viewing_id` is uniquely distinct from one to another. Thus, `property_viewing_id` is a primary key.

prop_view_date(prop_view_id, date_time)

PK:

FK: prop_view_date(prop_view_id) -> property_viewings(property_viewing_id)

Explanation: `prop_view_date` entity holds the date and time of the viewing. `prop_view_id` is not a primary key in this entity since it could be repeated with different date & time.

Rental_properties (id, type, area_sqr_footage, rent_month, rent_fee, status, owner_id, lease_id)

PK: id

FK: rental_properties(lease_id) -> leases(lease_id) ON DELETE CASCADE

Check:

CHECK (`area_sqr_footage` > 0)

CHECK (`rent_month` > 0)

CHECK (`rent_fee` > 0)

CHECK (`status` = 0 or `status` = 1)

Explanation: `Rental_properties` holds information and lease of each property. "The property number is unique across Pluto", thus `id` is a primary key.

commercial_prop(id, address, city, state, zipcode)

PK: id

FK: commercial_prop(id) -> rental_properties(id) ON DELETE CASCADE

industrial_prop(id, address, city, state, zipcode)

PK: id

FK: industrial_prop(id) -> rental_properties(id) ON DELETE CASCADE

residential_prop(id, address, city, state, zipcode, number_of_bedrooms, number_of_bathrooms)

PK: id

FK: residential_prop(id) -> rental_properties(id) ON DELETE CASCADE

CHECK (`number_of_bedrooms` >=0)

CHECK (`number_of_bathrooms` >=0)

associates(id, property_id, number_of_prop, prop_date_time)

PK:

FK:

associates(id) -> employees(employee_id) ON DELETE CASCADE

associates(property_id) -> rental_properties(id) ON DELETE CASCADE

Check:

CHECK (0<= `number_of_prop` <=12)

Explanation: `Associates` keeps track of properties. Since one associate can be assigned to manage more than one property. `id` is not a primary key.

partners(id, owner_id, client_id, lease_id)

PK:

FK:

Partners(id) -> employees (employee_id) ON DELETE CASCADE

Partners(owner_id) -> leases (lease_id) ON DELETE CASCADE

Partners(client_id) -> clients (client_id) ON DELETE CASCADE

partners(lease_id) -> property_owner(owner_id) ON DELETE CASCADE

Explanation: partner representing the owner holding the information of client and lease of the property.

Since each partner can be working with one or more owners as well as clients. Partner`id` is not a primary key.

clients(client_id, email, prop_preference, rent_pay_rate)

PK: client_id

FK: clients(client_id) -> person(person_id) ON DELETE CASCADE

CHECK (`prop_preference` = 'Residential' or `prop_preference` = 'Commercial' or `prop_preference` = 'Industrial')

CHECK (`rent_pay_rate`> 0)

Explanation: clients are people interested in renting property. Each client is uniquely distinct from one another, thus `client_id` is a primary key. I'm assuming each client only wants to rent one type of property. Hence, `client_id` is not repeated in this table. In the future, `client_id` cannot be a primary key due to the fact that one client may be interested in both "commercial" as well as "residential" property.

leases(lease_id, lease_date, monthly_rent, deposit, duration, start, finish)

PK: lease_id

FK:

CHECK (`monthly_rent` > 0)

CHECK (`deposit` > 0)

CHECK (3 months < `duration` < 36 months)

property_owner(owner_id, property_id, first_name, last_name, corporation_name, address, phone, email)

PK:

FK: property_owner(property_id) -> Rental_properties(id) ON DELETE CASCADE

Explanation: `property_owner` holds information of the owner as well as keeps record of their property(s).

I'm assuming that one owner could own one or more properties. Thus, `owner_id` is not a primary key.

3. Updated MySQL EER Data Model

EER.pdf file in the folder**.**

4. User's interface design

For clients:

- Show available renting property options.
- Show properties that fit client' preferences
- Show properties that have monthly renting fit maximum monthly rent that clients are willing to pay

For employees:

- Employee's supervisor: to access, look up, and manage other employees in the company.
- Partners: to sign up leases, look up information of property's owners as well as clients
- Associates: to access and manage information of properties.

For property's owner:

- Show information of their partners.
- See status of their properties that are being rented.

Queries and reports

1. List the names of all the unique clients.
2. Find the unique names of owners and total square footage of all the properties they own.
3. Find the properties shown by each associate in a given month.

Example provided uses MARCH as month.

4. Find the most popular properties (in terms of number of viewings in a given year).

Example provided uses 2019 as year.

5. Find the total rent due to each property owner.
6. Find the unique names of associates supervised (directly or indirectly) by a given employee.

7. Find the unique names of owners that have a residential property in every city where Pat Doe owns a commercial property.
8. Find the top-3 partners with respect to the number of properties leased in the current year. Example provided uses 2020 as year.

These last two sql statements must be executed in the workbench.
9. Write a SQL function to compute the total management fees due to Pluto in the last 3 months.
10. Create a SQL trigger to automatically set to FALSE the advertisement flag of a property when it is
leased.