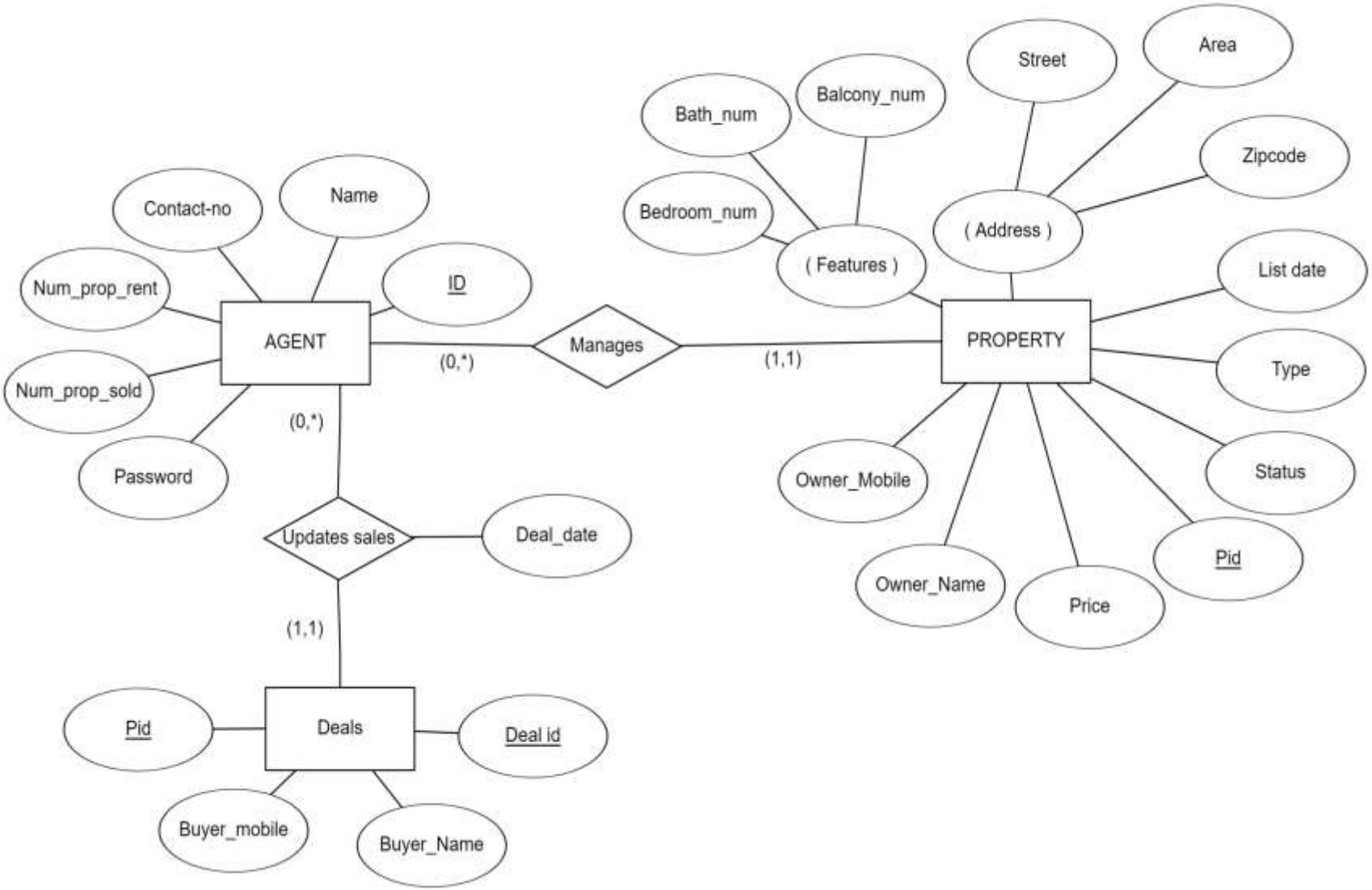
Real Estate DBMS Project
CS241

Deliverable 2

Name : Thirumurugan R
Roll no : 1801185
Section : CG 2

MODIFIED - ER DIAGRAM



SQL TABLES

PROPERTY

Columns
Pid (*Primary key)*
Agent_id (Foreign key)
Owner_name
Owner_mobile
Price
Status
Type
List_date
Area
Street
Zip_code

FEATURES

Columns
Pid (Foreign key)
Bedroom_num
Bath_num
Balcony_num

AGENT

Columns
Id (Primary key)
Name
Contact_no
Password
Num_prop_sold
Num_prop_rent

SALES

Columns
Deal_no (Primary key)
Pid (Foreign key)
Buyer_name
Buyer_mobile
Deal_date

User-Interfaces

1)Database-Administrator:

Type of interface : Command-line-interface

-> Admin has access to the Real-estate database and all tables available in it.
-> Admin can run queries for retrieving information from the database using MySQL command line.

2) Interface for Agent:

Type of interface : Flask based application (Agent portal)

-> Agent does not have access to the Agent table , he can’t view the details of other Agents.

Screen 1: Agent Login page

-> User would have 2 options
Option 1: Register (New user) ----- Screen 2
Option 2: Login
-> Login option
-> Agent can login using his id and password.
-> Once logged in, user is redirected to agent home page.

Screen 2: Registration page

-> User is allowed to choose his id and password and must fill his details
-> After getting information in a form , database is connected. Now we don't want to have two users with the same id, so we first want to see if that id already exists. If it does, then we tell them that id already exists, and let them try again.
-> Similar process is followed for user contact number .(two users can’t have same contact number)
-> Once a unique id and contact number is matched , we insert into the database.(Agents table)
-> Backend : Agent details are added to agent table. The attributes num_prop_sold and num_prop_rent are initialised to 0.

-> A message is flashed to the user that registration is done.
-> Re-directed to Agent LOGIN PAGE.

Screen 3: Agent home page

-> Information regarding the agent would be displayed on screen.
-> Agent has 4 options, each option directs to particular screen.
Properties Assigned ------ Screen 4
Sales Update------Screen 5
Add property------Screen 6
My deals------Screen 7

Screen 4: Properties assigned page

-> Agent can view the details of properties assigned to him and their details.

Screen 5: Sales Update page

-> Agent is required to fill pid, buyer_name ,deal_date and buyer_mobile
-> After getting information in a form , database is connected. Now we don't want to have two deals with the same pid, so we first want to see if that pid deal has been made . If it does, then we tell them that property deal was over already.

-> Once matched(Submission successful),we insert into database. (Sales table)
-> A flash message would be displayed *"Sales updated successfully"*.
-> Backend: Num_prop_sold or Num_prop_rent details of the agent would be incremented based on type of property. The status of the property would be updated to CLOSED.
-> The agent has an option to add another deal or go back to home page with the help of navigation bar.

Screen 6: Add property page

-> Agent must fill in the form with property info.
-> Once matched, we insert into database(Property table) .
-> A flash message is displayed with Pid info.
-> Backend : Status of the property would be initialised with OPEN,and the Agent_id would be updated to the id of the agent logged in.
-> The agent has an option to either add another property or go back to home page with navigation bar.

Screen 7: My deals page

-> Agent can view 2 tables:
Table 1: Pending deals
Table 2: Successful deals

-> Pending deals: The details of the properties associated with agent, which is OPEN would be displayed.
-> Successful deals : The details of the properties associated with agent ,which is CLOSED would be displayed.
-> Agent can get back to home page using navigation bar.

2) Interface for Real estate office:

Type of interface : Flask based application (Estate portal)

Screen 1: Real-estate admin page(home)

-> Details of agents with their name and contact number is displayed.
-> Admin has 4 options, each option directs to a particular screen.
Sales report------Screen 2
Area wise report ------Screen 3
Add property------Screen 4
Search property------Screen 5

Screen 2: Sales report page

-> The admin is required to input Agent_id for getting the report of an agent
-> If Agent_id does not match, then admin is asked to try again.
-> If successful, successful deals made by the agent would be displayed.
-> Admin has an option to either enter another agent id or return to home page using navigation bar.

Screen 3: Area wise report page

-> Admin has 2 options for each field :
Field 1(Area): 1) all areas 2) input specific area,
Field 2(Type) 1) Rent 2)Sale
-> all areas: details of successful deals grouped by areas would be displayed based on the given type.
-> Specific areas: The admin is required to input area name.
-> If area does not match, then admin is asked to try again as the required one is not found.
-> If successful, successful deals made by the agents in the specific area would be displayed based on given type.
-> Admin has an option to give another input or return to home page using navigation bar.

Screen 4: Add property page

-> Admin must fill in the form with property info and must assign an Agent_id associated with property.
-> Once matched, we insert into database(Property table) .
-> A flash message is displayed with Pid info.
-> Backend : Status of the property would be initialised with OPEN.
-> The admin has an option to either add another property or go back to home page with navigation bar

Screen 5: Search property page

-> Admin can input the features of the type of property he needs to search.(Similar to filters in websites)
-> Details of the property satisfying given filters would be displayed.
-> Admin has option to either search another property or go back to home page with navigation bar