**INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG.**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

REAL STATE MANAGEMENT SYSTEM

Prepared for Tata Realty and Infrastructure Ltd.

PROJeCt RePORt

Database Management System



**COURSE CODE : COURSE T IT L E :**

CSE-2424.

Database Management System.

Submitted To,

Emaz Uddin Ahmed.

Adjunct Faculty,  
Dept. Of CSE, IIUC.

Submitted By,

# Mohammad Arefin Hosain.

ID : C211121.

Section : 4CM. Semester : 4th. Dept. Of CSE, IIUC.

REAL STATE MANAGEMENT SYSTEM

A S S U M E C O M P A N Y : TATA REALTY AND IN S U R A N C E L T D .

**ABSTRACT**

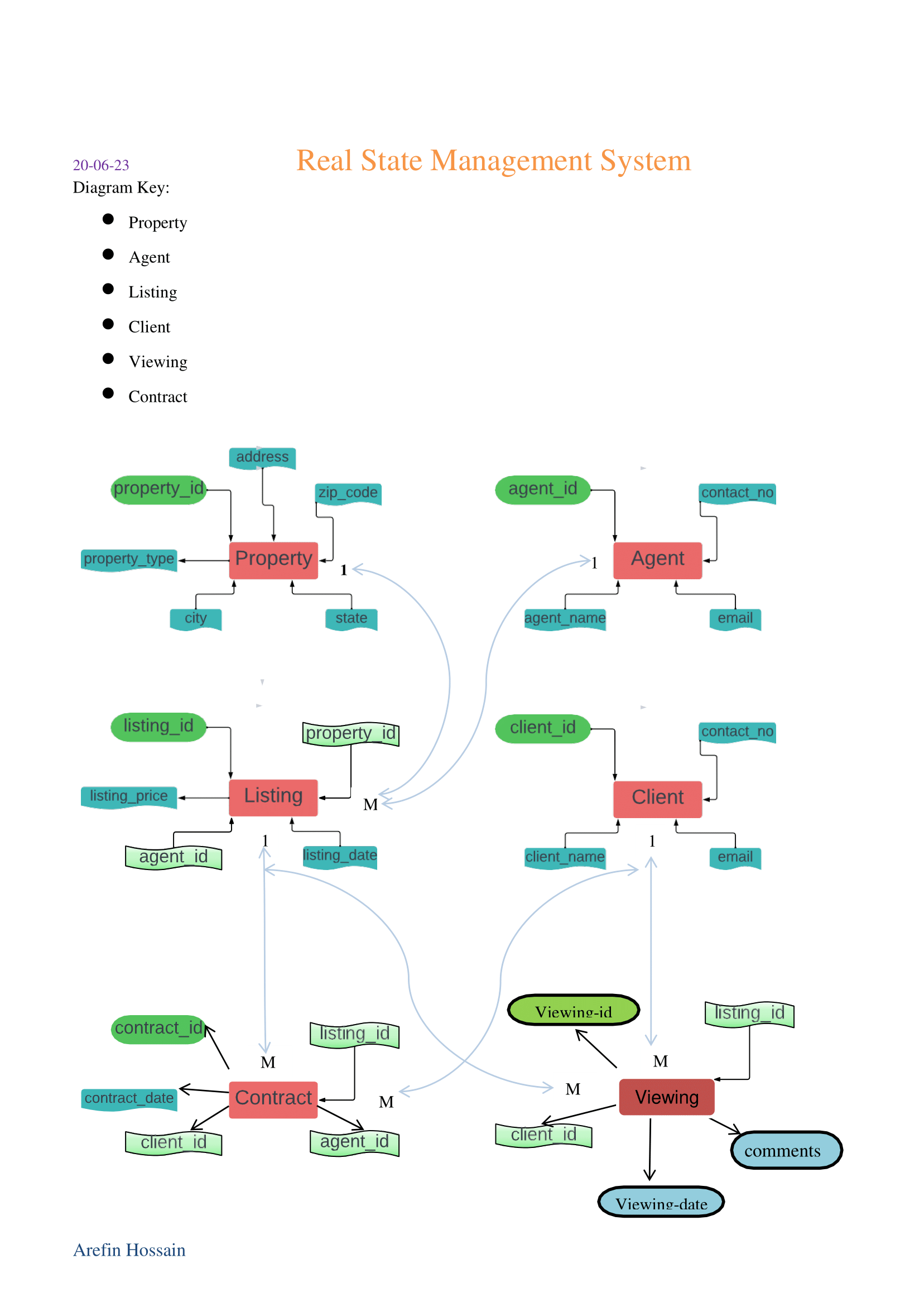
This project is about "Real Estate Management", Assume a Real Estate company named "Tata Realty and Infrastructure Ltd".This company is our client.

This project is to designed for streamline and optimize the operations of real estate management companies. REMS offers a user-friendly interface and a wide range of features to facilitate efficient property management, enhance customer satisfaction, and drive business growth.

The REMS platform provides robust functionalities such as property listing management, agent and client management, contract management, and viewing scheduling. With its intuitive design, users can easily navigate through the system, ensuring a seamless experience in handling various aspects of real estate operations.

**INTRODUCTION**

Boost your speed-to-market with automation solutions designed for specialty Real Estate.It's a complete solution for Optimize property listings, agent-client interactions, contracts, and viewing schedules. Boost operational efficiency, enhance customer satisfaction, and drive business growth. Gain valuable insights through reporting and analytics. Existing manual system of real estate management system takes long time to manage client-agents records.



**E R D I A G R A M**

**SCEMA**

1. Property - (Property\_Id, Property\_Type, Address, City, State, Zip, , No\_of\_bedrooms, No\_of\_bathrooms)
2. Agent - (Agent\_Id, Agent\_Name, Contact\_Number, Email)
3. Client - (Client\_Id, Client\_Name, Contact\_Number, Email)
4. Listing - (Listing\_Id, Property\_Id, Agent\_Id, Listing\_Price, Listing\_Date)
5. Viewing - (Viewing\_Id, Listing\_Id, Client\_Id, Viewing\_Date, Comments)
6. Constract - (Constract\_Id, Listing\_Id, Client\_Id, Agent\_Id, Constract\_Date, Constract\_Terms)

**DDL STATEMENTS & TABLES WITH DATA**

**PROPERTY SQL**

**create table Property (  
 property\_id number primary key,  
 property\_type varchar2(30) not null,  
 address varchar2(25) not null,  
 city varchar2(20),**

**state varchar2(12),**

**zip number(6) not null,**

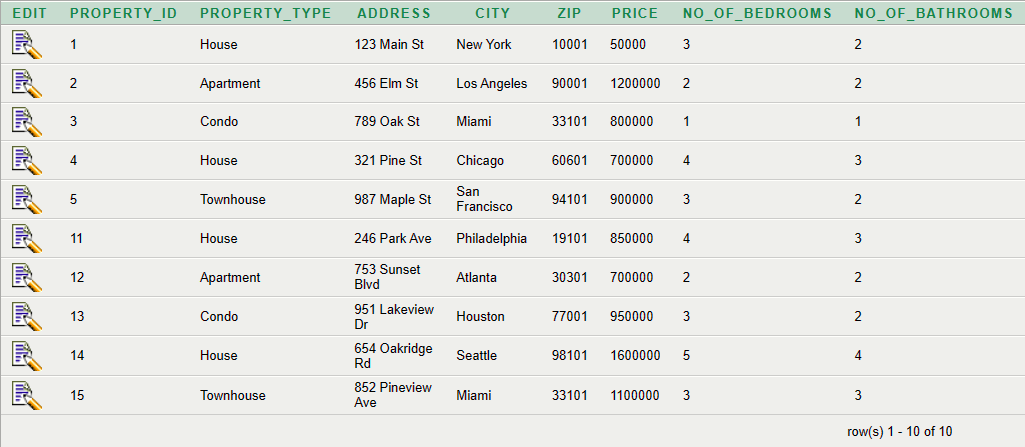
**price number(21) not null,**

**no\_of\_bedrooms number(10),**

**no\_of\_bathrooms number(10)**

**);**

**PROPERTY TABLE**



**AGENT SQL**

**create table Agent(**

**agent\_id number primary key,**

**agent\_name varchar2(30) not null,**

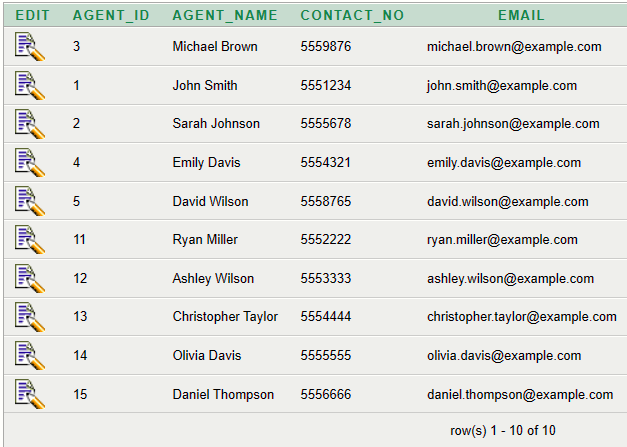
**contact\_no number(38) not null,**

**email varchar2(35) not null**

**);**

**AGENT TABLE**

**AGENT SQL**



**CLIENT SQL**

**create table Client(  
 client\_id number primary key,**

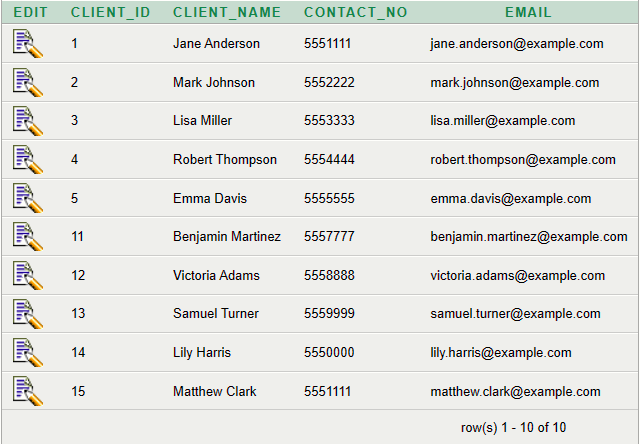
**client\_name varchar2(30) not null,**

**contact\_no number(38) not null,**

**email varchar2(35) not null**

**);**

**CLIENT TABLE**

****

**LISTING SQL**

create table Listing(

listing\_id number primary key, property\_id, agent\_id,

foreign key (property\_id) references property(property\_id),

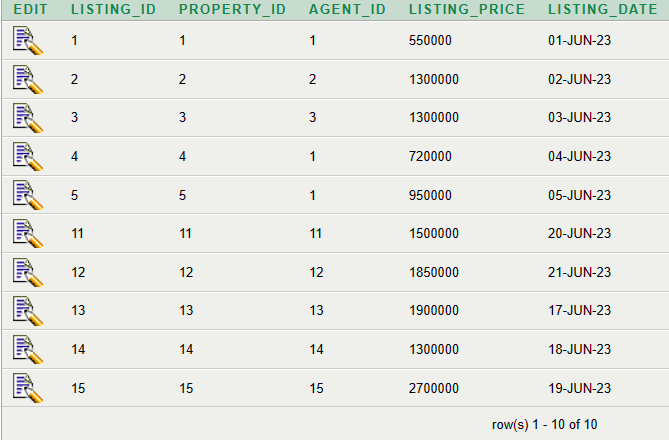
foreign key (agent\_id) references agent(agent\_id),

listing\_price number(38) not null,

listing\_date date

);

**LISTING TABLE**



**VIEWING SQL**

create table Viewing(

viewing\_id number primary key, listing\_id, client\_id,

foreign key (listing\_id) references listing(listing\_id),

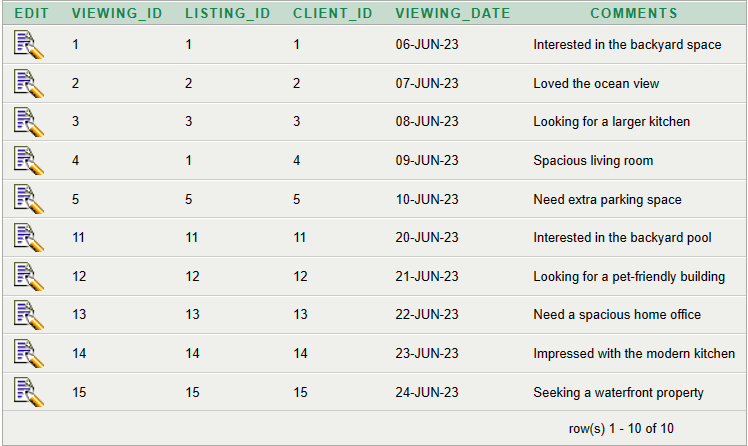
foreign key (client\_id) references client(client\_id),

viewing\_date date not null,

comments varchar2(38)

);

**VIEWING TABLE**



**CONTRACT SQL**

**Create table Contract(**

**Contract\_id number primary key, listing\_id, client\_id, agent\_id,**

**foreign key (listing\_id) references listing(listing\_id),**

**foreign key (client\_id) references client(client\_id),**

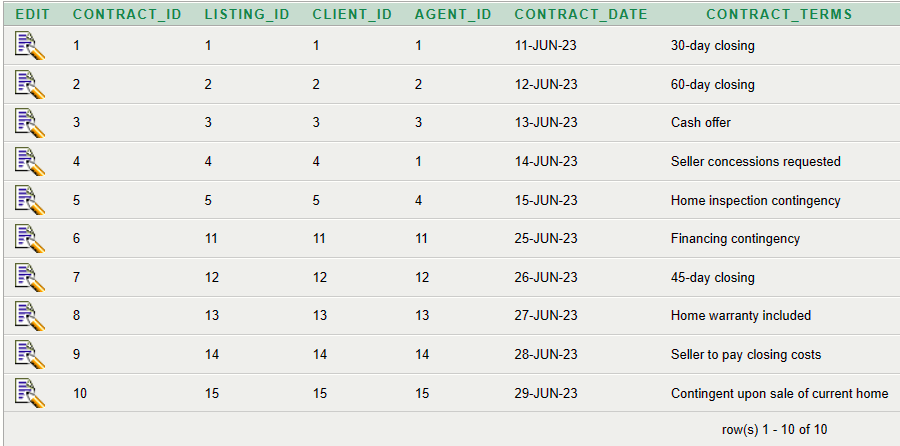
**foreign key (agent\_id) references agent(agent\_id),**

**contract\_date date not null,**

**contract\_terms varchar2(38)**

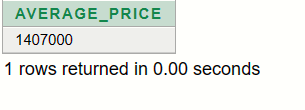
**);**

**CONTRACT TABLE**



**A) SEARCHING DATA IN POSSIBLE WAYS (AT LEAST 13 WAYS) TABLE**

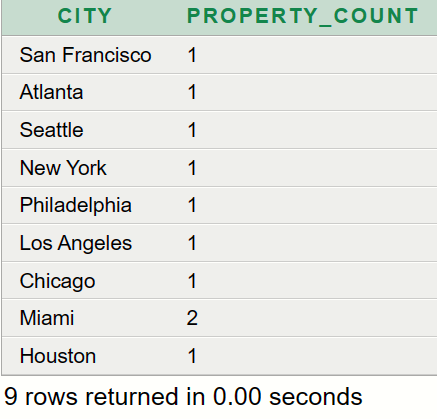
Question 01 : Calculate the average listing price:



**Answer:**

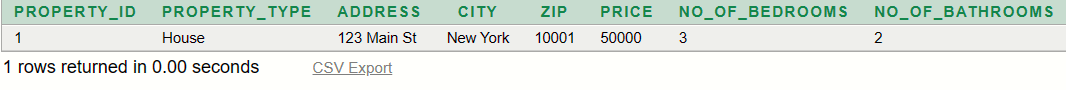
**SELECT AVG(listing\_price) AS average\_price  
FROM Listing;**

Question 02 : Retrieve the number of properties in each city:



**Answer:**

SELECT city, COUNT(\*)AS property\_count  
FROM Property GROUP BY city;

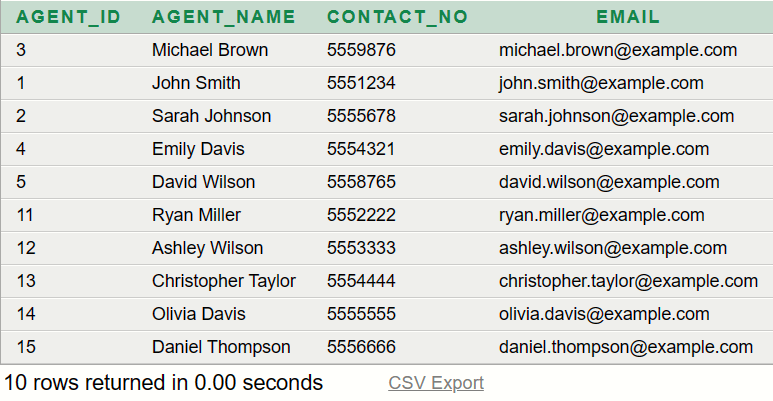


**Question 03 : Retrieve all properties in the city of New York:**

**Answer:**

**SELECT \* FROM Property**

**WHERE city = 'New York';**



Question 04 : Retrieve all agents:

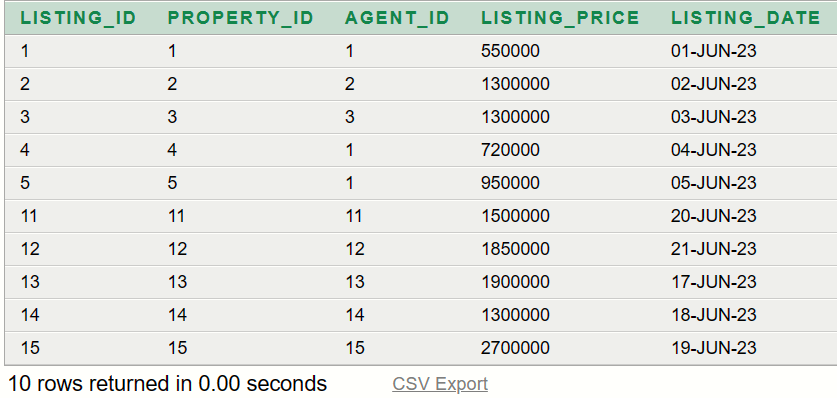
**Answer:**

**SELECT \* FROM Agent;**

Question 05 : Retrieve all listings:

**Answer:**

**SELECT \* FROM Listing;**



**Question 06 : Retrieve the listings with listing\_price less than  
 $600,000:**

**Answer:**

**SELECT \* FROM**

**Listing WHERE listing\_price < 600000;**

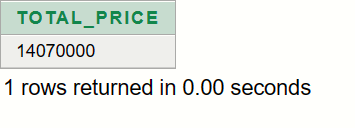
Question 07 : Retrieve the client with client\_id = 1:



**Answer:**

**SELECT \* FROM Client WHERE client\_id = 1;**

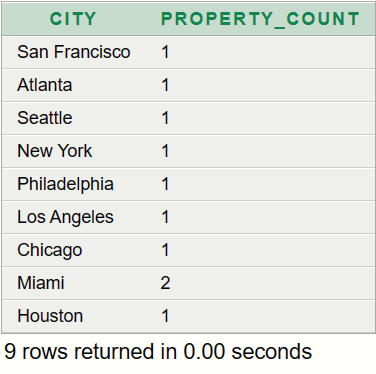
Question 08 : Calculate the sum of listing prices:



**Answer:**

**SELECT SUM(listing\_price) AS total\_price FROM Listing;**

Question 09 : Retrieve the number of properties in each city:

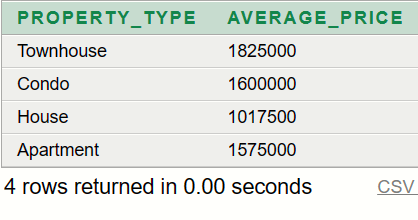


Answer:

SELECT city, COUNT(\*)

AS property\_count FROM Property

GROUP BY city;



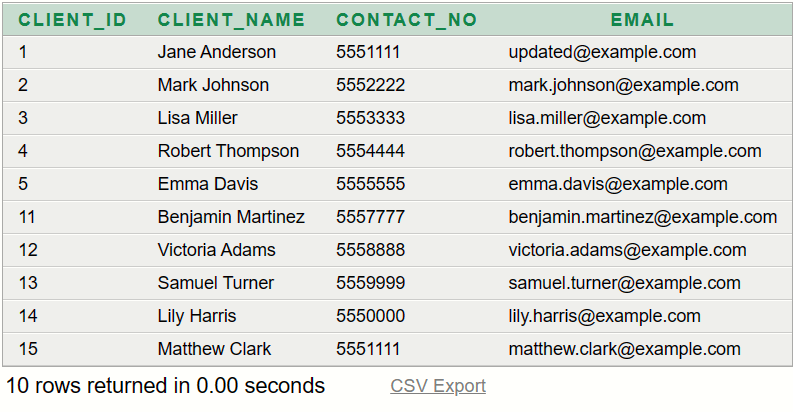
**Question 10 : Calculate the average listing price for each property type:**

**Answer:**

**SELECT property\_type, AVG(listing\_price) AS FROM Listing**

**JOIN Property ON Listing.property\_id = Property.property\_id**

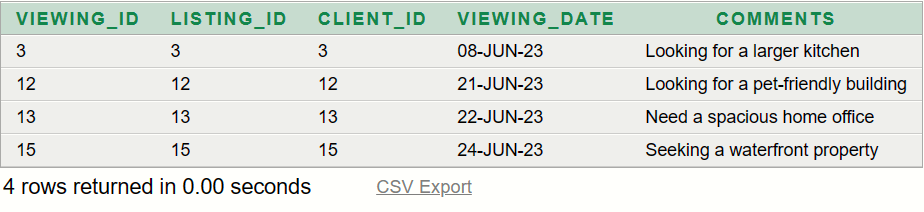
**GROUP BY property\_type;**



**Question 11 : Sorting Client Data:**

**Answer:**

**SELECT \* FROM Client ORDER BY client\_id asc, client\_name desc;**



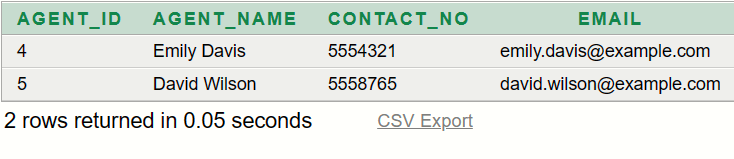
**Question 12 : Like Operator used into filtering:**

**Answer:**

**SELECT \* FROM viewing**

**where comments like'%f%'**

**Question 13 : Retrive a seller who sold not property:**



**Answer:**

**SELECT \* FROM Agent**

**WHERE agent\_id NOT IN (SELECT DISTINCT agent\_id FROM Listing);**

**Question 01 : Find the clients who have made contracts for properties listed by the top 5 agents with the highest number of list:**

**Answer:**

**SELECT client\_id, client\_name**

**FROM Client**

**WHERE client\_id IN (**

**SELECT client\_id**

**FROM Contract**

**WHERE listing\_id IN (**

**SELECT listing\_id**

**FROM Listing**

**WHERE agent\_id IN (**

**SELECT agent\_id**

**FROM (**

**SELECT agent\_id**

**FROM Agent**

**ORDER BY (SELECT COUNT(\*) FROM Listing WHERE Agent.agent\_id = Listing.agent\_id) DESC**

**)**

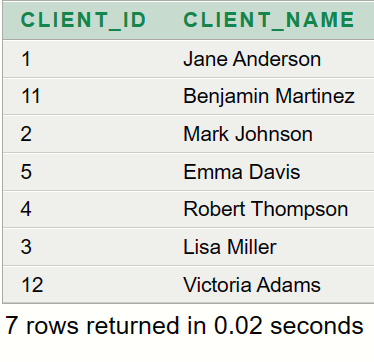
**WHERE ROWNUM <= 5**

**)**

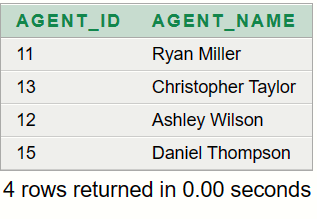
**)**

**);**

**B) ALL TYPES OF SUB-QUERIES**



Question 02 : Find the agents who have listed properties with a listing price higher than the average listing price:



**Answer:**

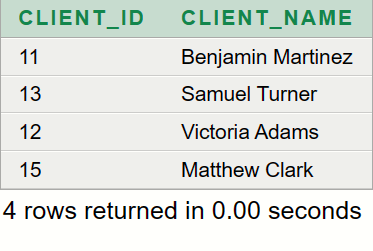
**SE SELECT agent\_id, agent\_name FROM Agent WHERE agent\_id IN (SELECT agent\_id**

**FROM Listing**

**WHERE listing\_price > (**

**SELECT AVG(listing\_price) FROM Listing  
 )  
 );**

Question 03 : Retrieve the clients who have made contracts for properties with a listing price higher than the average listing price average listing price



**Answer:**

**SELECT client\_id, client\_name**

**FROM Client**

**WHERE client\_id IN (**

**SELECT client\_id**

**FROM Contract**

**WHERE listing\_id IN (**

**SELECT listing\_id**

**FROM Listing**

**WHERE listing\_price > (**

**SELECT AVG(listing\_price)**

**FROM Listing**

**)**

**)**

**);**

**PL/SQL**

Question 01 : Retrieve client information along with agent details:

**Answer:**

DECLARE

client\_id Client.client\_id%TYPE;

client\_name Client.client\_name%TYPE;

agent\_id Agent.agent\_id%TYPE;

agent\_name Agent.agent\_name%TYPE;

BEGIN

-- Cursor to fetch client information

FOR client\_rec IN (SELECT c.client\_id, c.client\_name, a.agent\_id, a.agent\_name

FROM Client c

JOIN Contract con ON con.client\_id = c.client\_id

JOIN Listing l ON l.listing\_id = con.listing\_id

JOIN Agent a ON a.agent\_id = l.agent\_id)

LOOP

client\_id := client\_rec.client\_id;

client\_name := client\_rec.client\_name;

agent\_id := client\_rec.agent\_id;

agent\_name := client\_rec.agent\_name;

-- Display the result

DBMS\_OUTPUT.PUT\_LINE('Client ID: ' || client\_id || ', Client Name: ' || client\_name);

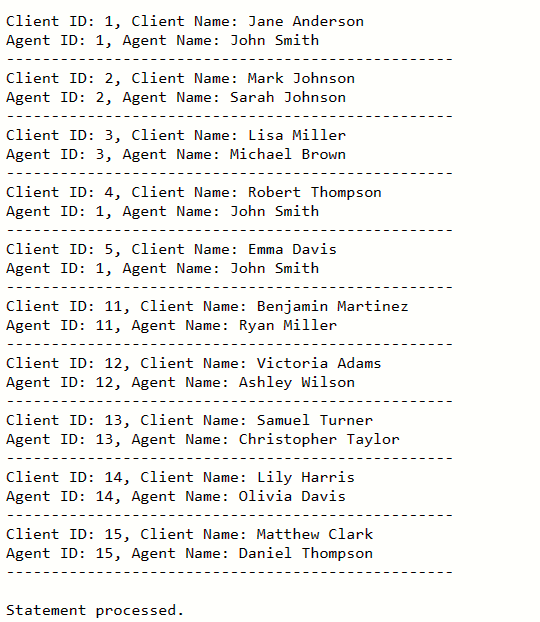
DBMS\_OUTPUT.PUT\_LINE('Agent ID: ' || agent\_id || ', Agent Name: ' || agent\_name);

DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------------- ');

END LOOP;

END;

/



**THE**



**Thank you!!**

|^|ASSALAMUALAIKUM|^|