**Data Assessment**

**Objective**

The objective of this individual assessment is to build a database based on the learning outcome of SQL

**Overview - Scenario**

‘**MyHomes**’ **Real Estate Company**

MyHomes is a real estate company operating globally, their main area of business lies in buying and selling properties.

The properties that the company deals in are of five types: houses, flats/apartments, bungalows, land, and commercial property. These properties may be subject to health and safety regulations, which govern how these properties must be built and maintained.

Each property that MyHomes deals in is the main responsibility of one of the company’s area sales representatives. Each representative, however, may be responsible for a number of different properties. A number of the area sales representatives act as supervisors. Supervisors are directly responsible for managing teams of fellow area sales representatives. An area sales representative may also be responsible for dealing with a number of MyHomes’s customers. Each of the company’s customers will only deal with MyHomes through one particular sales representative.

Each sales representative also covers one or more geographical sales areas, for eliciting new business from prospective customers. A number of different sales representatives may cover the same geographical area, but there will always be a single sales representative who acts as the chief salesperson for that area, and who is directly responsible to management for sales performance in that area, in terms of hitting sales targets.

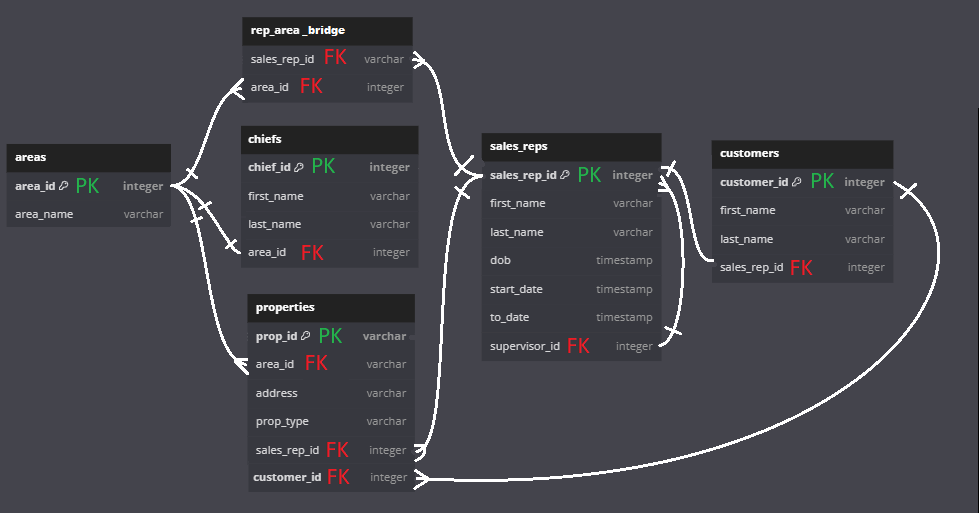
You are asked to design and implement information systems (database) for MyHomes company.

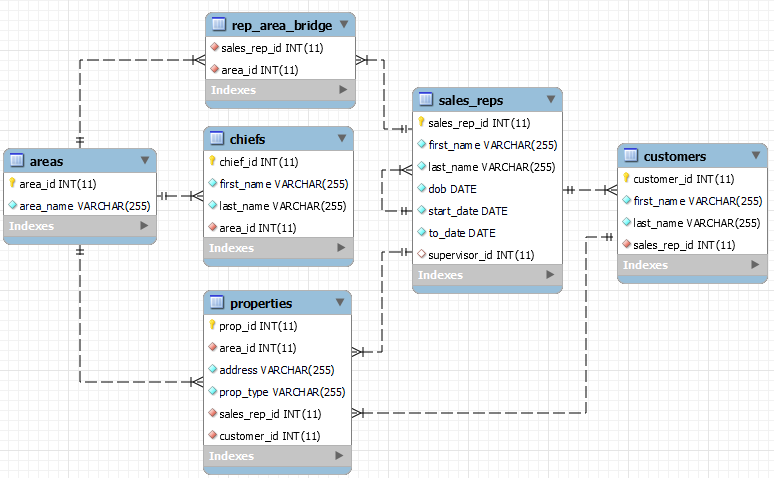
**Tasks of the Assessment**

**Building a New database with appropriate tables**

**Part 1 (25 points)**

*(A) Using a database design (relational model)/Normalization approach of your choice, produce a Conceptual, Logical and Physical design/models for the database to support the information system, which is needed at the* ***MyHomes.***





(10 points)

*(B) Based on your Physical design from Part 1 (A) and the information available in the scenario, produce an SQL script that create appropriate tables*

(15 points)

| DROP DATABASE IF EXISTS my\_homes;  CREATE database my\_homes;  USE my\_homes;  DROP TABLE IF EXISTS properties, areas, sales\_reps, chiefs, customers, rep\_area\_bridge, prop\_cust\_bridge;  CREATE TABLE properties (  prop\_id integer PRIMARY KEY,  area\_id integer NOT NULL,  address varchar(255) NOT NULL,  prop\_type varchar(255) NOT NULL,  sales\_rep\_id integer NOT NULL,  customer\_id integer NOT NULL  );  CREATE TABLE areas (  area\_id integer PRIMARY KEY,  area\_name varchar(255) NOT NULL  );  CREATE TABLE sales\_reps (  sales\_rep\_id integer PRIMARY KEY,  first\_name varchar(255) NOT NULL,  last\_name varchar(255) NOT NULL,  dob date NOT NULL,  start\_date date NOT NULL,  to\_date date NOT NULL,  supervisor\_id integer  );  CREATE TABLE chiefs (  chief\_id integer PRIMARY KEY,  first\_name varchar(255) NOT NULL,  last\_name varchar(255) NOT NULL,  area\_id integer NOT NULL  );  CREATE TABLE customers (  customer\_id integer PRIMARY KEY,  first\_name varchar(255) NOT NULL,  last\_name varchar(255) NOT NULL,  sales\_rep\_id integer NOT NULL  );  CREATE TABLE rep\_area\_bridge (  sales\_rep\_id integer NOT NULL,  area\_id integer NOT NULL,  FOREIGN KEY (sales\_rep\_id) REFERENCES sales\_reps(sales\_rep\_id)  ON DELETE CASCADE ON UPDATE CASCADE,  FOREIGN KEY (area\_id) REFERENCES areas(area\_id)  ON DELETE CASCADE ON UPDATE CASCADE  );    ALTER TABLE properties  ADD CONSTRAINT fk\_proparea FOREIGN KEY (area\_id) REFERENCES areas(area\_id)  ON DELETE CASCADE ON UPDATE CASCADE,  ADD CONSTRAINT fk\_proprep FOREIGN KEY (sales\_rep\_id) REFERENCES sales\_reps(sales\_rep\_id)  ON DELETE CASCADE ON UPDATE CASCADE,  ADD CONSTRAINT fk\_propcust FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)  ON DELETE CASCADE ON UPDATE CASCADE;    ALTER TABLE sales\_reps  ADD CONSTRAINT fk\_superv FOREIGN KEY (supervisor\_id) REFERENCES sales\_reps(sales\_rep\_id)  ON DELETE CASCADE ON UPDATE CASCADE;    ALTER TABLE customers  ADD CONSTRAINT fk\_custarea FOREIGN KEY (sales\_rep\_id) REFERENCES sales\_reps(sales\_rep\_id)  ON DELETE CASCADE ON UPDATE CASCADE;    ALTER TABLE chiefs  ADD CONSTRAINT fk\_chiefarea FOREIGN KEY (area\_id) REFERENCES areas(area\_id)  ON DELETE CASCADE ON UPDATE CASCADE;    Describe areas;    Describe rep\_area\_bridge; |
| --- |

**Part 2 (25 points)**

This part is based on your answer / solution to Part 1, i.e., design and implementation of the database for the ‘MyHomes’ scenario. drop

*(A) Populate the database tables with some sample data*

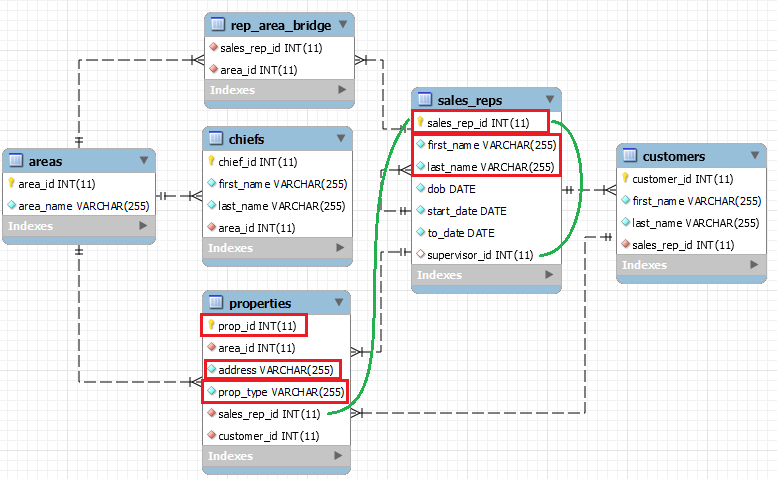
| INSERT INTO sales\_reps (sales\_rep\_id, first\_name, last\_name, dob, start\_date, to\_date, supervisor\_id) VALUES  (101, 'John', 'Doe', '1985-05-15', '2010-01-01', '2023-07-31', 103),  (102, 'Jane', 'Smith', '1990-09-23', '2012-03-15', '2023-07-31', 108),  (103, 'Michael', 'Jordan', '1988-12-10', '2015-06-01', '2023-07-31', NULL),  (104, 'Emily', 'Williams', '1992-07-20', '2017-02-10', '2023-07-31', 103),  (105, 'Robert', 'Clark', '1987-11-30', '2020-03-01', '2023-07-31', 108),  (106, 'Laura', 'Miller', '1994-04-18', '2022-01-15', '2023-07-31', 108),  (107, 'William', 'Anderson', '1991-08-25', '2018-09-20', '2023-07-31', 108),  (108, 'Susan', 'Lee', '1989-06-12', '2019-04-05', '2023-07-31', NULL),  (109, 'David', 'Brown', '1993-02-09', '2021-07-10', '2023-07-31', 103),  (110, 'Sarah', 'Martin', '1986-10-07', '2016-12-20', '2023-07-31', 103);    – Because of the self-reference, I need to break this down in two steps:  – Step 1: Insert all sales representatives without supervisors (supervisor\_id = NULL)  INSERT INTO sales\_reps (sales\_rep\_id, first\_name, last\_name, dob, start\_date, to\_date, supervisor\_id) VALUES  (101, 'John', 'Doe', '1985-05-15', '2010-01-01', '2023-07-31', NULL),  (102, 'Jane', 'Smith', '1990-09-23', '2012-03-15', '2023-07-31', NULL),  (103, 'Michael', 'Jordan', '1988-12-10', '2015-06-01', '2023-07-31', NULL),  (104, 'Emily', 'Williams', '1992-07-20', '2017-02-10', '2023-07-31', NULL),  (105, 'Robert', 'Clark', '1987-11-30', '2020-03-01', '2023-07-31', NULL),  (106, 'Laura', 'Miller', '1994-04-18', '2022-01-15', '2023-07-31', NULL),  (107, 'William', 'Anderson', '1991-08-25', '2018-09-20', '2023-07-31', NULL),  (108, 'Susan', 'Lee', '1989-06-12', '2019-04-05', '2023-07-31', NULL),  (109, 'David', 'Brown', '1993-02-09', '2021-07-10', '2023-07-31', NULL),  (110, 'Sarah', 'Martin', '1986-10-07', '2016-12-20', '2023-07-31', NULL);  -- Step 2: Update supervisors for the corresponding sales representatives  UPDATE sales\_reps  SET supervisor\_id = 103 -- Set supervisor for sales\_rep\_id 101, 104, 109, and 110  WHERE sales\_rep\_id IN (101, 104, 109, 110);  UPDATE sales\_reps  SET supervisor\_id = 108 -- Set supervisor for sales\_rep\_id 102, 105, and 106  WHERE sales\_rep\_id IN (102, 105, 106);    – I forgot 107!  UPDATE sales\_reps  SET supervisor\_id = 108  WHERE sales\_rep\_id IN (107);    -- OK!  INSERT INTO chiefs (chief\_id, first\_name, last\_name, area\_id) VALUES  (101, 'John', 'Doe', 1),  (103, 'Michael', 'Jordan', 2),  (108, 'Susan', 'Lee', 3),  (110, 'Sarah', 'Martin', 4);  INSERT INTO areas (area\_id, area\_name) VALUES  (1, 'north'),  (2, 'south'),  (3, 'west'),  (4, 'east');  INSERT INTO customers (customer\_id, first\_name, last\_name, sales\_rep\_id) VALUES  (201, 'John', 'Johnson', 101),  (202, 'Emma', 'Davis', 102),  (203, 'Michael', 'Brown', 103),  (204, 'Olivia', 'Miller', 104),  (205, 'Noah', 'Smith', 105),  (206, 'Sophia', 'Wilson', 106),  (207, 'James', 'Martinez', 107),  (208, 'Ava', 'Jones', 108),  (209, 'Oliver', 'Taylor', 109),  (210, 'Isabella', 'Anderson', 110);  INSERT INTO rep\_area\_bridge (sales\_rep\_id, area\_id) VALUES  (101, 1),  (102, 1),  (102, 2),  (103, 2),  (103, 3),  (104, 3),  (105, 3),  (106, 4),  (107, 4),  (108, 4),  (109, 1),  (110, 1);  INSERT INTO properties (prop\_id, area\_id, address, prop\_type, sales\_rep\_id, customer\_id) VALUES  (401, 1, '123 Main St', 'house', 101, 201),  (402, 1, '456 Elm St', 'flat', 102, 202),  (403, 2, '789 Oak St', 'house', 103, 203),  (404, 3, '101 Maple St', 'bungalow', 104, 204),  (405, 4, '222 Pine St', 'land', 105, 205),  (406, 1, '777 Walnut St', 'commercial property', 106, 206),  (407, 3, '333 Cedar St', 'house', 107, 207),  (408, 4, '444 Birch St', 'flat', 108, 208),  (409, 2, '888 Spruce St', 'bungalow', 109, 209),  (410, 1, '666 Birch St', 'land', 110, 210),  (411, 3, '999 Maple St', 'commercial property', 101, 201),  (412, 1, '111 Elm St', 'house', 102, 202),  (413, 2, '333 Oak St', 'flat', 103, 203),  (414, 3, '555 Pine St', 'bungalow', 104, 204),  (415, 4, '777 Cedar St', 'land', 105, 205),  (416, 1, '222 Walnut St', 'commercial property', 106, 206),  (417, 2, '888 Elm St', 'house', 107, 207),  (418, 3, '333 Oak St', 'flat', 108, 208),  (419, 4, '999 Pine St', 'bungalow', 109, 209),  (420, 1, '555 Walnut St', 'land', 110, 210),  (421, 2, '444 Elm St', 'commercial property', 101, 201),  (422, 3, '777 Oak St', 'house', 102, 202),  (423, 4, '333 Pine St', 'flat', 103, 203),  (424, 1, '777 Oak St', 'house', 104, 204),  (425, 2, '888 Pine St', 'flat', 105, 205),  (426, 3, '444 Cedar St', 'bungalow', 106, 206),  (427, 4, '333 Elm St', 'land', 107, 207),  (428, 1, '555 Walnut St', 'commercial property', 108, 208),  (429, 2, '777 Elm St', 'house', 109, 209),  (430, 3, '888 Oak St', 'flat', 110, 210); |
| --- |

(15 points)

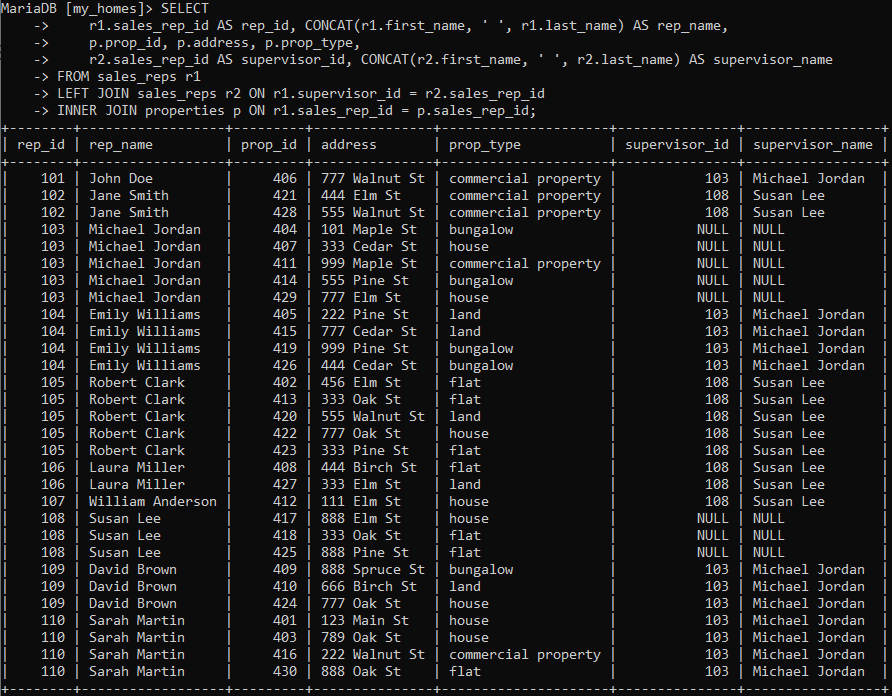
*(B) Answer the following queries (retrievals) using SQL*

*1) Display names of representatives, details of the properties they represent, and names of their supervisors.*

I am marking the needed columns in red and the connections in green:

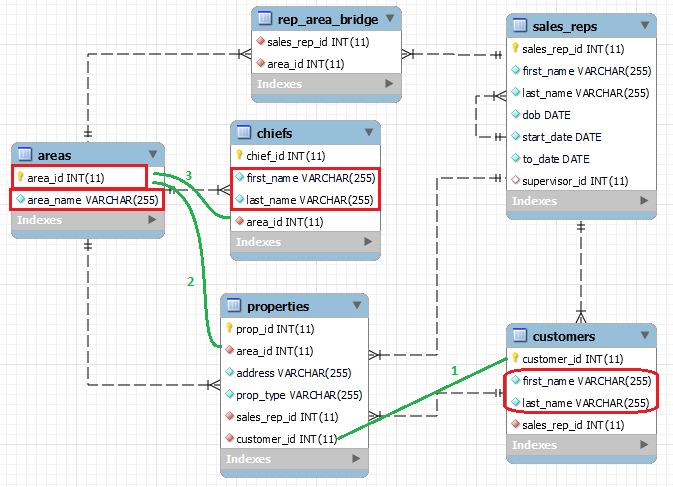


| SELECT  r1.sales\_rep\_id AS rep\_id, CONCAT(r1.first\_name, ' ', r1.last\_name) AS rep\_name,  p.prop\_id, p.address, p.prop\_type,  r2.sales\_rep\_id AS supervisor\_id, CONCAT(r2.first\_name, ' ', r2.last\_name) AS supervisor\_name  FROM sales\_reps r1  LEFT JOIN sales\_reps r2 ON r1.supervisor\_id = r2.sales\_rep\_id  INNER JOIN properties p ON r1.sales\_rep\_id = p.sales\_rep\_id; |
| --- |

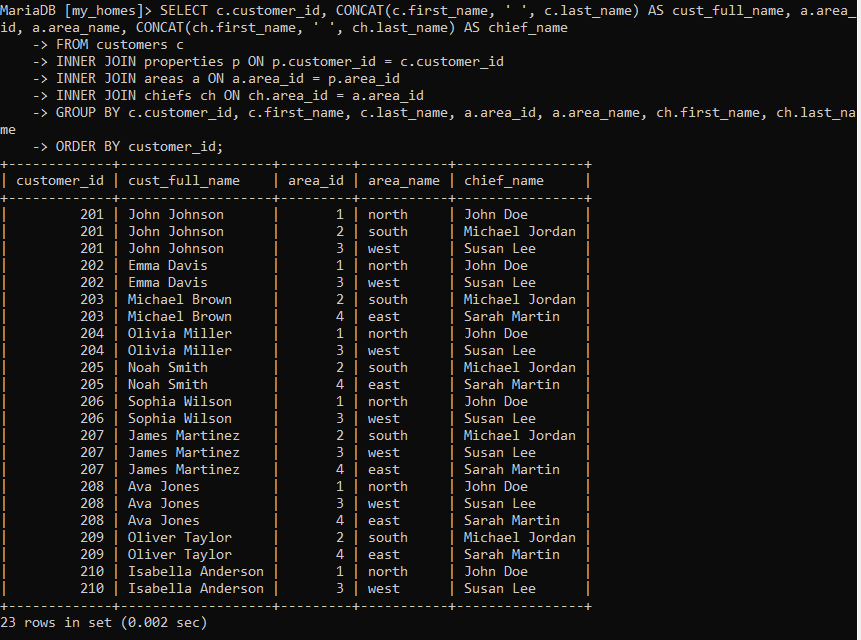


*2) Display details of customers together with details of their areas and names of the managers of their areas.*

I am marking the needed columns in red and the connections in green:



| SELECT c.customer\_id, CONCAT(c.first\_name, ' ', c.last\_name) AS cust\_full\_name, a.area\_id, a.area\_name, CONCAT(ch.first\_name, ' ', ch.last\_name) AS chief\_name  FROM customers c  INNER JOIN properties p ON p.customer\_id = c.customer\_id  INNER JOIN areas a ON a.area\_id = p.area\_id  INNER JOIN chiefs ch ON ch.area\_id = a.area\_id  GROUP BY c.customer\_id, c.first\_name, c.last\_name, a.area\_id, a.area\_name, ch.first\_name, ch.last\_name  ORDER BY customer\_id; |
| --- |



(10 points)