**Assignment Questions**

**Part 1 (35 marks)**

This part is based on the MOVEHOME scenario as described in the Appendix.

1. *Using entity-relationship (ER) OR enhanced entity-relationship (EER) modelling, produce a conceptual design for the database to support the MOVEHOME business activities.*

(15 marks)

Answer Part 1 A: Insert your ER or EER Diagram Below

Diagram

Description automatically generated

1. *Convert the ER / EER diagram from Part 1(A) to produce a logical relational schema using ER / EER to relational mapping.*

(10 marks)

Answer Part 1 B: Provide your Logical Relational Design/Schema Below

Graphical user interface

Description automatically generated

*(C) Based on your logical design from Part 1 (B) and the information available in the scenario, produce an SQL script file using Oracle 11g/12c/higher.*

(10 marks)

Answer Part 1 C: Provide SQL DDL Script file contents (i.e., the SQL code for creating / altering your Tables / Constraints etc)

Set Echo on

Set Verify on

DROP TABLE Customer CASCADE CONSTRAINTS PURGE;

DROP TABLE Owner CASCADE CONSTRAINTS PURGE;

DROP TABLE Buyer CASCADE CONSTRAINTS PURGE;

DROP TABLE Tenant CASCADE CONSTRAINTS PURGE;

DROP TABLE Property CASCADE CONSTRAINTS PURGE;

DROP TABLE Accommodation CASCADE CONSTRAINTS PURGE;

DROP TABLE SoldProperty CASCADE CONSTRAINTS PURGE;

DROP TABLE RentProperty CASCADE CONSTRAINTS PURGE;

DROP TABLE EstateAgent CASCADE CONSTRAINTS PURGE;

DROP TABLE MANAGER CASCADE CONSTRAINTS PURGE;

DROP TABLE Branch CASCADE CONSTRAINTS PURGE;

DROP TABLE Staff CASCADE CONSTRAINTS PURGE;

DROP TABLE ViewProperty CASCADE CONSTRAINTS PURGE;

**Creating Table for Customer:**

CREATE TABLE Customer(

cust\_id CHAR(2) NOT NULL,

name VARCHAR(30),

email VARCHAR(30),

phone\_number NUMBER(11),

gender VARCHAR(8) CONSTRAINT check\_gender CHECK(gender IN ('Male', 'Female')),

date\_of\_birth DATE,

address\_line VARCHAR(30),

post\_code VARCHAR(10) NOT NULL,

city VARCHAR(20) NOT NULL,

state VARCHAR(20),

CONSTRAINT pk\_customer Primary Key(cust\_id));

CREATE TABLE Owner(

owner\_id CHAR(2) NOT NULL

CONSTRAINT unique\_owner\_id UNIQUE

CONSTRAINT fk\_owner REFERENCES Customer(cust\_id));

**Creating Table for Buyer:**

CREATE TABLE Buyer(

buyer\_id CHAR(2) NOT NULL

CONSTRAINT unique\_buyer\_id UNIQUE

CONSTRAINT fk\_buyer REFERENCES Customer(cust\_id));

**Creating Table for Tenant:**

CREATE TABLE Tenant(

tenant\_id CHAR(2) NOT NULL

CONSTRAINT unique\_tenant\_id UNIQUE

CONSTRAINT fk\_tenant REFERENCES Customer(cust\_id));

**Create Table Estate Agent:**

CREATE TABLE EstateAgent(

est\_id CHAR(2) NOT NULL,

name VARCHAR(30),

email VARCHAR(30),

phone\_number NUMBER(11),

headquarter VARCHAR(20),

address\_line VARCHAR(30),

post\_code VARCHAR(10) NOT NULL,

city VARCHAR(20) NOT NULL,

state VARCHAR(20),

CONSTRAINT pk\_estagent Primary Key(est\_id));

**Create Table Manager:**

CREATE TABLE Manager(

manager\_id CHAR(2) NOT NULL,

name VARCHAR(50),

phone\_number NUMBER(11),

email VARCHAR(20) NOT NULL,

CONSTRAINT pk\_manager Primary Key(manager\_id));

**Create Table Branch:**

CREATE TABLE Branch(

code CHAR(2) NOT NULL,

agent\_id CHAR(2) NOT NULL

CONSTRAINT branch\_fk\_agent\_id REFERENCES EstateAgent(est\_id),

manager CHAR(2) NOT NULL

CONSTRAINT manager\_fk\_manager REFERENCES MANAGER(manager\_id),

email VARCHAR(20),

phone\_number NUMBER(11),

address\_line VARCHAR(30),

post\_code VARCHAR(10) NOT NULL,

city VARCHAR(20) NOT NULL,

state VARCHAR(20),

CONSTRAINT pk\_branch Primary Key(code));

CREATE TABLE Staff(

staff\_id CHAR(2) NOT NULL,

branch\_id CHAR(2) NOT NULL

CONSTRAINT staff\_fk\_branch\_id REFERENCES Branch(code),

name VARCHAR(30),

phone\_number NUMBER(13),

address VARCHAR(20) NOT NULL,

email VARCHAR(20),

CONSTRAINT pk\_staff Primary Key(staff\_id));

**Create Table Property:**

CREATE TABLE Property(

property\_id CHAR(2) NOT NULL,

owner\_id CHAR(2) NOT NULL

CONSTRAINT property\_fk\_owner\_id REFERENCES Owner(owner\_id),

branch\_id CHAR(2) NOT NULL

CONSTRAINT property\_fk\_branch\_id REFERENCES Branch(code),

post\_code VARCHAR(10) NOT NULL,

city VARCHAR(20) NOT NULL,

state VARCHAR(20),

location VARCHAR(20),

number\_of\_rooms NUMBER(2)

CONSTRAINT min\_max\_no

CHECK (number\_of\_rooms >= 1 AND number\_of\_rooms <=10),

rental\_demand NUMBER(10)

CONSTRAINT min\_rental\_demand

CHECK (rental\_demand > 0),

asking\_price NUMBER(10)

CONSTRAINT min\_asking\_price

CHECK (asking\_price > 0),

type VARCHAR(30)

CONSTRAINT property\_type CHECK (type IN

('FLATS', 'DETACHED', 'SEMI-DETACHED', 'TERRACED', 'COTTAGE', 'BUNGALOWS')),

status VARCHAR(15) CONSTRAINT property\_status CHECK (status IN ('FOR-SALE', 'FOR-RENT', 'SOLD', 'RENTED')),

adding\_date DATE DEFAULT sysdate,

CONSTRAINT pk\_property Primary Key(property\_id));

**Create Table Accommodation:**

CREATE TABLE Accommodation(

acc\_id CHAR(2) NOT NULL,

property\_id CHAR(2) NOT NULL

CONSTRAINT property\_acc\_fk\_property\_id REFERENCES Property(property\_id),

length NUMBER(3) CONSTRAINT check\_length CHECK(length > 0),

width NUMBER(3) CONSTRAINT check\_width CHECK(width > 0),

area NUMBER(10) CONSTRAINT check\_area CHECK(area > 0),

description VARCHAR(20),

CONSTRAINT pkey\_property\_accommodation Primary Key(acc\_id));

CREATE TABLE SoldProperty(

sold\_id CHAR(2) NOT NULL,

property\_id CHAR(2) NOT NULL

CONSTRAINT sold\_property\_fk\_property\_id REFERENCES property(property\_id),

buyer\_id CHAR(2) NOT NULL

CONSTRAINT sold\_property\_fk\_buyer\_id REFERENCES buyer(buyer\_id),

selling\_price NUMBER(10) CONSTRAINT check\_selling\_price CHECK(selling\_price > 0),

stump\_duty NUMBER(10),

cm\_paid NUMBER(4),

selling\_date DATE,

CONSTRAINT pk\_sold\_property Primary Key(sold\_id));

**Create Table Rent Property:**

CREATE TABLE RentProperty(

rent\_id CHAR(2) NOT NULL,

property\_id CHAR(2) NOT NULL

CONSTRAINT rented\_property\_fk\_property\_id REFERENCES property(property\_id),

tenant\_id CHAR(2) NOT NULL

CONSTRAINT sold\_property\_fk\_tenant\_id REFERENCES Tenant(tenant\_id),

rent NUMBER(10) CONSTRAINT check\_rent CHECK(rent > 0 AND rent < 1000),

adv\_paid NUMBER(10),

date\_tenancy\_commenced DATE,

date\_tenancy\_finished DATE,

commission NUMBER(4),

CONSTRAINT date\_tenancy\_finished\_const

CHECK(date\_tenancy\_finished >= date\_tenancy\_commenced),

CONSTRAINT pkey\_rented\_property Primary Key(rent\_id));

**Create Table View Property:**

CREATE TABLE ViewProperty(

view\_id CHAR(2) NOT NULL,

customer\_id CHAR(2) NOT NULL

CONSTRAINT view\_fk\_customer\_id REFERENCES Customer(cust\_id),

property\_id CHAR(2) NOT NULL

CONSTRAINT view\_fk\_property\_id REFERENCES Property(property\_id),

date\_time DATE,

customer\_type CHAR(15)

CONSTRAINT customer\_type CHECK (customer\_type IN

('buyer', 'tenant')),

status CHAR(15)

CONSTRAINT status\_view\_property\_request CHECK (status IN ('not-viewed', 'viewed')),

comments VARCHAR(30),

CONSTRAINT pk\_view Primary Key(view\_id));

Answer Part 1 C: SQL DDL Output (e.g., SPOOL file contents or output you got when you executed your above SQL Table Creation code, this should show the SQL code as well as its output)

W21029971 >@C:\Users\w21029971\Downloads\ddl.sql

W21029971 >Set Echo on

W21029971 >Set Verify on

W21029971 >

W21029971 >DROP TABLE Customer CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE Owner CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE Buyer CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE Tenant CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE Property CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE Accommodation CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE SoldProperty CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE RentProperty CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE EstateAgent CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE MANAGER CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE Branch CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE Staff CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >DROP TABLE ViewProperty CASCADE CONSTRAINTS PURGE;

Table dropped.

W21029971 >

W21029971 >

W21029971 >

W21029971 >CREATE TABLE Customer(

2 cust\_id CHAR(2) NOT NULL,

3 name VARCHAR(30),

4 email VARCHAR(30),

5 phone\_number NUMBER(11),

6 gender VARCHAR(8) CONSTRAINT check\_gender CHECK(gender IN ('Male', 'Female')),

7 date\_of\_birth DATE,

8 address\_line VARCHAR(30),

9 post\_code VARCHAR(10) NOT NULL,

10 city VARCHAR(20) NOT NULL,

11 state VARCHAR(20),

12 CONSTRAINT pk\_customer Primary Key(cust\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE Owner(

2 owner\_id CHAR(2) NOT NULL

3 CONSTRAINT unique\_owner\_id UNIQUE

4 CONSTRAINT fk\_owner REFERENCES Customer(cust\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE Buyer(

2 buyer\_id CHAR(2) NOT NULL

3 CONSTRAINT unique\_buyer\_id UNIQUE

4 CONSTRAINT fk\_buyer REFERENCES Customer(cust\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE Tenant(

2 tenant\_id CHAR(2) NOT NULL

3 CONSTRAINT unique\_tenant\_id UNIQUE

4 CONSTRAINT fk\_tenant REFERENCES Customer(cust\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE EstateAgent(

2 est\_id CHAR(2) NOT NULL,

3 name VARCHAR(30),

4 email VARCHAR(30),

5 phone\_number NUMBER(11),

6 headquarter VARCHAR(20),

7 address\_line VARCHAR(30),

8 post\_code VARCHAR(10) NOT NULL,

9 city VARCHAR(20) NOT NULL,

10 state VARCHAR(20),

11 CONSTRAINT pk\_estagent Primary Key(est\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE Manager(

2 manager\_id CHAR(2) NOT NULL,

3 name VARCHAR(50),

4 phone\_number NUMBER(11),

5 email VARCHAR(20) NOT NULL,

6 CONSTRAINT pk\_manager Primary Key(manager\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE Branch(

2 code CHAR(2) NOT NULL,

3 agent\_id CHAR(2) NOT NULL

4 CONSTRAINT branch\_fk\_agent\_id REFERENCES EstateAgent(est\_id),

5 manager CHAR(2) NOT NULL

6 CONSTRAINT manager\_fk\_manager REFERENCES MANAGER(manager\_id),

7 email VARCHAR(20),

8 phone\_number NUMBER(11),

9 address\_line VARCHAR(30),

10 post\_code VARCHAR(10) NOT NULL,

11 city VARCHAR(20) NOT NULL,

12 state VARCHAR(20),

13 CONSTRAINT pk\_branch Primary Key(code));

Table created.

W21029971 >

W21029971 >CREATE TABLE Staff(

2 staff\_id CHAR(2) NOT NULL,

3 branch\_id CHAR(2) NOT NULL

4 CONSTRAINT staff\_fk\_branch\_id REFERENCES Branch(code),

5 name VARCHAR(30),

6 phone\_number NUMBER(13),

7 address VARCHAR(20) NOT NULL,

8 email VARCHAR(20),

9 CONSTRAINT pk\_staff Primary Key(staff\_id));

Table created.

W21029971 >

W21029971 >

W21029971 >CREATE TABLE Property(

2 property\_id CHAR(2) NOT NULL,

3 owner\_id CHAR(2) NOT NULL

4 CONSTRAINT property\_fk\_owner\_id REFERENCES Owner(owner\_id),

5 branch\_id CHAR(2) NOT NULL

6 CONSTRAINT property\_fk\_branch\_id REFERENCES Branch(code),

7 post\_code VARCHAR(10) NOT NULL,

8 city VARCHAR(20) NOT NULL,

9 state VARCHAR(20),

10 location VARCHAR(20),

11 number\_of\_rooms NUMBER(2)

12 CONSTRAINT min\_max\_no

13 CHECK (number\_of\_rooms >= 1 AND number\_of\_rooms <=10),

14 rental\_demand NUMBER(10)

15 CONSTRAINT min\_rental\_demand

16 CHECK (rental\_demand > 0),

17 asking\_price NUMBER(10)

18 CONSTRAINT min\_asking\_price

19 CHECK (asking\_price > 0),

20 type VARCHAR(30)

21 CONSTRAINT property\_type CHECK (type IN

22 ('FLATS', 'DETACHED', 'SEMI-DETACHED', 'TERRACED', 'COTTAGE', 'BUNGALOWS')),

23 status VARCHAR(15) CONSTRAINT property\_status CHECK (status IN ('FOR-SALE', 'FOR-RENT', 'SOLD', 'RENTED')),

24 adding\_date DATE DEFAULT sysdate,

25 CONSTRAINT pk\_property Primary Key(property\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE Accommodation(

2 acc\_id CHAR(2) NOT NULL,

3 property\_id CHAR(2) NOT NULL

4 CONSTRAINT property\_acc\_fk\_property\_id REFERENCES Property(property\_id),

5 length NUMBER(3) CONSTRAINT check\_length CHECK(length > 0),

6 width NUMBER(3) CONSTRAINT check\_width CHECK(width > 0),

7 area NUMBER(10) CONSTRAINT check\_area CHECK(area > 0),

8 description VARCHAR(20),

9 CONSTRAINT pkey\_property\_accommodation Primary Key(acc\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE SoldProperty(

2 sold\_id CHAR(2) NOT NULL,

3 property\_id CHAR(2) NOT NULL

4 CONSTRAINT sold\_property\_fk\_property\_id REFERENCES property(property\_id),

5 buyer\_id CHAR(2) NOT NULL

6 CONSTRAINT sold\_property\_fk\_buyer\_id REFERENCES buyer(buyer\_id),

7 selling\_price NUMBER(10) CONSTRAINT check\_selling\_price CHECK(selling\_price > 0),

8 stump\_duty NUMBER(10),

9 cm\_paid NUMBER(4),

10 selling\_date DATE,

11 CONSTRAINT pk\_sold\_property Primary Key(sold\_id));

Table created.

W21029971 >

W21029971 >CREATE TABLE RentProperty(

2 rent\_id CHAR(2) NOT NULL,

3 property\_id CHAR(2) NOT NULL

4 CONSTRAINT rented\_property\_fk\_property\_id REFERENCES property(property\_id),

5 tenant\_id CHAR(2) NOT NULL

6 CONSTRAINT sold\_property\_fk\_tenant\_id REFERENCES Tenant(tenant\_id),

7 rent NUMBER(10) CONSTRAINT check\_rent CHECK(rent > 0 AND rent < 1000),

8 adv\_paid NUMBER(10),

9 date\_tenancy\_commenced DATE,

10 date\_tenancy\_finished DATE,

11 commission NUMBER(4),

12 CONSTRAINT date\_tenancy\_finished\_const

13 CHECK(date\_tenancy\_finished >= date\_tenancy\_commenced),

14 CONSTRAINT pkey\_rented\_property Primary Key(rent\_id));

Table created.

W21029971 >

W21029971 >

W21029971 >CREATE TABLE ViewProperty(

2 view\_id CHAR(2) NOT NULL,

3 customer\_id CHAR(2) NOT NULL

4 CONSTRAINT view\_fk\_customer\_id REFERENCES Customer(cust\_id),

5 property\_id CHAR(2) NOT NULL

6 CONSTRAINT view\_fk\_property\_id REFERENCES Property(property\_id),

7 date\_time DATE,

8 customer\_type CHAR(15)

9 CONSTRAINT customer\_type CHECK (customer\_type IN

10 ('buyer', 'tenant')),

11 status CHAR(15)

12 CONSTRAINT status\_view\_property\_request CHECK (status IN ('not-viewed', 'viewed')),

13 comments VARCHAR(30),

14 CONSTRAINT pk\_view Primary Key(view\_id));

Table created.

W21029971 >spool off

**Part 2 (20 marks)**

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

*(A) Populate the database with some sample data (e.g., you should generate your own dummy data and load it into the MOVIEHOME database, consider 5 to 10 rows for each table and enough data to see meaningful output for the queries below)*.

(10 marks)

Answer Part 2 A: Provide SQL code below for populating the above relational database

**Customer:**

INSERT INTO Customer

VALUES ('C1', 'Edward Smith', 'edward@gmail.com', 07878941774, 'Male', '12-FEB-1989', '60 Hersham Close', 'NE3 2TN', 'Newcastle', 'England');

INSERT INTO Customer

VALUES ('C2', 'Eden Hazard', 'eden@gmail.com', 07878941775, 'Male', '23-AUG-1999', 'Alexandra Park', 'SK3 2EY', 'Sutherland', 'England');

INSERT INTO Customer

VALUES ('C3', 'Muhammad Abdullah', 'ab@gmail.com', 07878941776, 'Male', '12-SEP-1994', '12 street Gatehead', 'NE4 2YZ', 'Newcastle', 'England');

INSERT INTO Customer

VALUES ('C4', 'Babar Azam', 'babar@gmail.com', 07878941777, 'Male', '11-OCT-1996', 'Coventry Club street','DH6 2XY', 'Durham', 'England');

INSERT INTO Customer

VALUES ('C5', 'Joe Root', 'joe@gmail.com', 07878941778, 'Male', '19-MAR-1992', 'City Center','NE4 2TN', 'Newcastle', 'England');

INSERT INTO Customer

VALUES ('C6', 'Alex Hales', 'alex@gmail.com', 07878941779, 'Male', '09-MAY-1986', 'Scotfield road','SK10 4XN', 'Sutherland', 'England');

INSERT INTO Customer

VALUES ('C7', 'Steve Smith', 'steve@gmail.com', 07878941784, 'Male', '23-NOV-1999', 'Riverlake Bypass', 'DH6 2AB', 'Durham', 'England');

INSERT INTO Customer

VALUES ('C8', 'Jason', 'jason@gmail.com', 07878977784, 'Male', '11-DEC-1998', 'Fawdon', 'NH6 2AB', 'Durham', 'England');

INSERT INTO Customer

VALUES ('C9', 'Karen', 'karen@gmail.com', 07878998784, 'Female', '23-JAN-1999', 'Skull Rpad', 'SK 2AB', 'Sutherland', 'England');

INSERT INTO Customer

VALUES ('C0', 'Daria', 'steveki@gmail.com', 07878941454, 'Female', '23-NOV-1999', 'Riverlake Bypass', 'DH6 2AC', 'Durham', 'England');

**Owner, Buyer and Tenant:**

INSERT INTO OWNER

VALUES ('C1');

INSERT INTO OWNER

VALUES ('C2');

INSERT INTO OWNER

VALUES ('C3');

INSERT INTO Buyer

VALUES ('C4');

INSERT INTO Buyer

VALUES ('C5');

INSERT INTO Buyer

VALUES ('C6');

INSERT INTO Tenant

VALUES ('C7');

INSERT INTO Tenant

VALUES ('C8');

INSERT INTO Tenant

VALUES ('C9');

INSERT INTO Tenant

VALUES ('C0');

**Estate Agent:**

INSERT INTO EstateAgent

VALUES ('E1', 'Manson Jack', 'edward@gmail.com', 07878941774, 'Manson Property Consultants', '49 Hersham Close', 'NE3 2TN', 'Newcastle', 'England');

INSERT INTO EstateAgent

VALUES ('E2', 'Jan Forster', 'jan@gmail.com', 07876641774, 'Jan Forster Estates', '21 Ring Gateshead','NE32 1HE', 'Newcastle', 'England');

INSERT INTO EstateAgent

VALUES ('E3', 'JW Woods', 'jw@gmail.com', 07879941774, 'JW Estate Durham', 'Jackson Park hill road','DH1 3HL', 'Durham', 'England');

INSERT INTO EstateAgent

VALUES ('E4', 'Mike Robinson', 'mike@gmail.com', 07879941774, 'Robinsons Estate', 'Hall lane houghton', 'DH5 4HA', 'Durham', 'England');

INSERT INTO EstateAgent

VALUES ('E5', 'Aseon King', 'asi@gmail.com', 07872241774, 'Kings Estate', 'blackturn sillway', 'SK23 0HP', 'Sutherland', 'England');

**Manager:**

INSERT INTO Manager

VALUES ('M1', 'Adam Callum', 07878955156, 'ad@gmail.com');

INSERT INTO Manager

VALUES ('M2', 'Charles', 07878954136, 'charles@gmail.com');

INSERT INTO Manager

VALUES ('M3', 'Joe', 07878355116, 'joe@gmail.com');

INSERT INTO Manager

VALUES ('M4', 'Ben', 07878945116, 'ben@gmail.com');

INSERT INTO Manager

VALUES ('M5', 'Daria', 07878919156, 'daria@gmail.com');

INSERT INTO Manager

VALUES ('M6', 'Lion', 07878975116, 'li@gmail.com');

INSERT INTO Manager

VALUES ('M7', 'Jacob', 07878154156, 'jacob@gmail.com');

INSERT INTO Manager

VALUES ('M8', 'Iris', 07878911656, 'iris@gmail.com');

**Branch:**

INSERT INTO Branch

VALUES ('B1', 'E1', 'M1', 'edwardb1@gmail.com', 07878941714, '60 Hersham Close', 'NE3 2TN', 'Newcastle', 'England');

INSERT INTO Branch

VALUES ('B2', 'E1', 'M2', 'edwardb1@gmail.com', 07878941725, '62 Gateshead', 'NE32 2EY', 'Newcastle', 'England');

INSERT INTO Branch

VALUES ('B3', 'E2', 'M3', 'jan@gmail.com', 07878941736, , 'City Center Mall', 'NE4 2YZ', 'Newcastle', 'England');

INSERT INTO Branch

VALUES ('B4', 'E3', 'M4', 'jw@gmail.com', 07878941747, 'White House Crescent', 'DH6 2XY', 'Durham', 'England');

INSERT INTO Branch

VALUES ('B5', 'E4', 'M5', 'rebestate@gmail.com', 07878941758, 'South Hetton', 'DH4 2TN', 'Durham', 'England');

INSERT INTO Branch

VALUES ('B6', 'E4', 'M6', 'robestate@gmail.com', 07878941769, 'The Avenue', 'DH1 4XN', 'Durham', 'England');

INSERT INTO Branch

VALUES ('B7', 'E5', 'M7', 'aseon@gmail.com', 07878941789, 'Chapel Frith', 'SK23 0GP', 'Sutherland', 'England');

INSERT INTO Branch

VALUES ('B8', 'E5', 'M8', 'aseon@gmail.com', 07878941799, 'Prestbury', 'SK10 4XN', 'Sutherland', 'England');

**Staff:**

INSERT INTO Staff

VALUES ('S1', 'B1', 'Haris', '07878951226', 'Hilton Park NE32 2WE', 'haris@gmail.com');

INSERT INTO Staff

VALUES ('S2', 'B2', 'Ben', '07878891226', 'City Center NE4 2XY', 'ben@gmail.com');

INSERT INTO Staff

VALUES ('S3', 'B3', 'Karen', '07875551226', 'Airport Road DH3 2WE', 'karen@gmail.com');

INSERT INTO Staff

VALUES ('S4', 'B4', 'Paul', '07878456226', 'Coventry club SK4 3NE', 'paul@gmail.com');

INSERT INTO Staff

VALUES ('S5', 'B5', 'Logan', '07870981226', 'Riverside NE32 2XY', 'logan@gmail.com');

INSERT INTO Staff

VALUES ('S6', 'B6', 'Hyman', '07878987226', 'Crosside road DH1 4ER', 'hyman@gmail.com');

INSERT INTO Staff

VALUES ('S7', 'B7', 'Ali', '07877871226', 'Houghton SK11 2WE', 'ali@gmail.com');

INSERT INTO Staff

VALUES ('S8', 'B8', 'Kumar', '07878909826', '49 Hersham NE3 2WE', 'kumar@gmail.com');

**Property:**

INSERT INTO PROPERTY

VALUES ('P1', 'C1', 'B1', 'NE32 2t3', 'Newcastle', 'England', 'Jesmond', 3, '200', '150000', 'DETACHED', 'FOR-SALE', '21-NOV-2021');

INSERT INTO PROPERTY

VALUES ('P2', 'C2', 'B2', 'NE3 2TN', 'Newcastle', 'England', 'Jesmond', 3, '590', '250000', 'DETACHED', 'FOR-SALE', '11-NOV-2021');

INSERT INTO PROPERTY

VALUES ('P3', 'C3', 'B4', 'DH3 2AV', 'Durham', 'England', 'City Center', 2, '400', '150000', 'SEMI-DETACHED', 'SOLD', '19-NOV-2019');

INSERT INTO PROPERTY

VALUES ('P4', 'C1', 'B6', 'DH6 2AB', 'Durham', 'England', 'City Center', 4, '900', '250000', 'SEMI-DETACHED', 'SOLD', '10-SEP-2020');

INSERT INTO PROPERTY

VALUES ('P5', 'C2', 'B7', 'SK3 2AB', 'Sutherland', 'England', 'Houghton', 3, '600', '350000', 'SEMI-DETACHED', 'SOLD', '12-NOV-2021');

INSERT INTO PROPERTY

VALUES ('P6', 'C3', 'B4', 'DH3 2t3', 'Durham', 'England', 'City Center', 2, '900', '250000', 'BUNGALOWS', 'RENTED', '19-AUG-2021');

INSERT INTO PROPERTY

VALUES ('P7', 'C1', 'B4', 'DH3 2t3', 'Durham', 'England', 'City Center', 2, '400', '150000', 'SEMI-DETACHED', 'RENTED', '19-NOV-2021');

INSERT INTO PROPERTY VALUES ('P8', 'C2', 'B1', 'NE4 2t3', 'Newcastle', 'England', 'Fawdon', 2, '400', '200000', 'FLATS', 'FOR-SALE', '19-NOV-2021');

INSERT INTO PROPERTY

VALUES ('P9', 'C3', 'B3', 'NE3 2TN', 'Newcastle', 'England', 'Jesmond', 2, '400', '175000', 'DETACHED', 'FOR-SALE', '15-NOV-2021');

INSERT INTO PROPERTY

VALUES ('P0', 'C1', 'B8', 'SK3 2YZ', 'Sutherland', 'England', 'City Center', 2, '600', '150000', 'SEMI-DETACHED', 'FOR-SALE', '09-MAY-2021');

**Property Accommodation:**

INSERT INTO Accomodation

VALUES ('1', 'P1', 20, 10, 200,'Comfortable rooms');

INSERT INTO Accomodation

VALUES ('2', 'P1', 20, 20, 400,'Backyard');

INSERT INTO Accomodation

VALUES ('3', 'P2', 20, 10, 200,'Comfortable rooms');

INSERT INTO Accomodation

VALUES ('4', 'P3', 20, 10, 200,'Comfortable rooms');

INSERT INTO Accomodation

VALUES ('5', 'P4', 20, 10, 200,'Comfortable rooms');

**Sold Property:**

INSERT INTO SoldProperty

VALUES ('S3', 'P3', 'C6', 150000, 40, 1000, '19-NOV-2019');

INSERT INTO SoldProperty

VALUES ('S4', 'P4', 'C4', 186000, 40, 1000, '10-NOV-2020');

INSERT INTO SoldProperty

VALUES ('S5', 'P5', 'C5', 125000, 20, 800, '12-NOV-2021');

**Rent Property:**

INSERT INTO RentProperty

VALUES ('R1', 'P6', 'C7', 400, 800, '27-JAN-2019', '27-FEB-2023', 20);

INSERT INTO RentProperty

VALUES ('R2', 'P7', 'C8', 600, 800, '25-FEB-2020', '25-FEB-2023', 20);

INSERT INTO RentProperty

VALUES ('R5', 'P0', 'C7', 400, 800, '07-MAR-2019', '07-MAR-2023', 20);

**View Property:**

INSERT INTO ViewProperty

VALUES ('V1', 'C7', 'P6', '27-JAN-2021', 'tenant', 'viewed', 'Nice house');

INSERT INTO ViewProperty

VALUES ('V2', 'C8', 'P7', '27-JAN-2022', 'tenant', 'not-viewed', 'Nice Terrace');

INSERT INTO ViewProperty

VALUES ('V3', 'C4', 'P1', '22-NOV-2021', 'buyer', 'viewed', 'Nice house');

INSERT INTO ViewProperty

VALUES ('V4', 'C5', 'P2', '02-JAN-2022', 'buyer', 'not-viewed', 'Nice house');

INSERT INTO ViewProperty

VALUES ('V5', 'C6', 'P5', '11-AUG-2021', 'buyer', 'viewed', 'Nice house');

Answer Part 2 A: Provide below output from running the above SQL code for populating your relational database (e.g., contents from Spool file or screenshots, etc)

W21029971 >@C:\Users\w21029971\Downloads\insertion.sql

W21029971 >INSERT INTO Customer

2 VALUES ('C1', 'Edward Smith', 'edward@gmail.com', 07878941774, 'Male', '12-FEB-1989', '60 Hersham Close', 'NE3 2TN', 'Newcastle', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C2', 'Eden Hazard', 'eden@gmail.com', 07878941775, 'Male', '23-AUG-1999', 'Alexandra Park', 'SK3 2EY', 'Sutherland', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C3', 'Muhammad Abdullah', 'ab@gmail.com', 07878941776, 'Male', '12-SEP-1994', '12 street Gatehead', 'NE4 2YZ', 'Newcastle', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C4', 'Babar Azam', 'babar@gmail.com', 07878941777, 'Male', '11-OCT-1996', 'Coventry Club street','DH6 2XY', 'Durham', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C5', 'Joe Root', 'joe@gmail.com', 07878941778, 'Male', '19-MAR-1992', 'City Center','NE4 2TN', 'Newcastle', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C6', 'Alex Hales', 'alex@gmail.com', 07878941779, 'Male', '09-MAY-1986', 'Scotfield road','SK10 4XN', 'Sutherland', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C7', 'Steve Smith', 'steve@gmail.com', 07878941784, 'Male', '23-NOV-1999', 'Riverlake Bypass', 'DH6 2AB', 'Durham', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C8', 'Jason', 'jason@gmail.com', 07878977784, 'Male', '11-DEC-1998', 'Fawdon', 'NH6 2AB', 'Durham', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C9', 'Karen', 'karen@gmail.com', 07878998784, 'Female', '23-JAN-1999', 'Skull Rpad', 'SK 2AB', 'Sutherland', 'England');

1 row created.

W21029971 >INSERT INTO Customer

2 VALUES ('C0', 'Daria', 'steveki@gmail.com', 07878941454, 'Female', '23-NOV-1999', 'Riverlake Bypass', 'DH6 2AC', 'Durham', 'England');

1 row created.

W21029971 >

W21029971 >

W21029971 >INSERT INTO OWNER

2 VALUES ('C1');

1 row created.

W21029971 >INSERT INTO OWNER

2 VALUES ('C2');

1 row created.

W21029971 >INSERT INTO OWNER

2 VALUES ('C3');

1 row created.

W21029971 >

W21029971 >INSERT INTO Buyer

2 VALUES ('C4');

1 row created.

W21029971 >INSERT INTO Buyer

2 VALUES ('C5');

1 row created.

W21029971 >INSERT INTO Buyer

2 VALUES ('C6');

1 row created.

W21029971 >

W21029971 >INSERT INTO Tenant

2 VALUES ('C7');

1 row created.

W21029971 >INSERT INTO Tenant

2 VALUES ('C8');

1 row created.

W21029971 >INSERT INTO Tenant

2 VALUES ('C9');

1 row created.

W21029971 >INSERT INTO Tenant

2 VALUES ('C0');

1 row created.

W21029971 >

W21029971 >

W21029971 >INSERT INTO EstateAgent

2 VALUES ('E1', 'Manson Jack', 'edward@gmail.com', 07878941774, 'Manson Property', '49 Hersham Close', 'NE3 2TN', 'Newcastle', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO EstateAgent

2 VALUES ('E2', 'Jan Forster', 'jan@gmail.com', 07876641774, 'Jan Estates', '21 Ring Gateshead','NE32 1HE', 'Newcastle', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO EstateAgent

2 VALUES ('E3', 'JW Woods', 'jw@gmail.com', 07879941774, 'JW Estate Durham', 'Jackson Park hill road','DH1 3HL', 'Durham', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO EstateAgent

2 VALUES ('E4', 'Mike Robinson', 'mike@gmail.com', 07879941774, 'Robinsons Estate', 'Hall lane houghton', 'DH5 4HA', 'Durham', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO EstateAgent

2 VALUES ('E5', 'Aseon King', 'asi@gmail.com', 07872241774, 'Kings Estate', 'blackturn sillway', 'SK23 0HP', 'Sutherland', 'England');

1 row created.

W21029971 >

W21029971 >

W21029971 >INSERT INTO Manager

2 VALUES ('M1', 'Adam Callum', 07878955156, 'ad@gmail.com');

1 row created.

W21029971 >INSERT INTO Manager

2 VALUES ('M2', 'Charles', 07878954136, 'charles@gmail.com');

1 row created.

W21029971 >INSERT INTO Manager

2 VALUES ('M3', 'Joe', 07878355116, 'joe@gmail.com');

1 row created.

W21029971 >INSERT INTO Manager

2 VALUES ('M4', 'Ben', 07878945116, 'ben@gmail.com');

1 row created.

W21029971 >INSERT INTO Manager

2 VALUES ('M5', 'Daria', 07878919156, 'daria@gmail.com');

1 row created.

W21029971 >INSERT INTO Manager

2 VALUES ('M6', 'Lion', 07878975116, 'li@gmail.com');

1 row created.

W21029971 >INSERT INTO Manager

2 VALUES ('M7', 'Jacob', 07878154156, 'jacob@gmail.com');

1 row created.

W21029971 >INSERT INTO Manager

2 VALUES ('M8', 'Iris', 07878911656, 'iris@gmail.com');

1 row created.

W21029971 >

W21029971 >

W21029971 >INSERT INTO Branch

2 VALUES ('B1', 'E1', 'M1', 'edwardb1@gmail.com', 07878941714, '60 Hersham Close', 'NE3 2TN', 'Newcastle', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO Branch

2 VALUES ('B2', 'E1', 'M2', 'edwardb1@gmail.com', 07878941725, '62 Gateshead', 'NE32 2EY', 'Newcastle', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO Branch

2 VALUES ('B3', 'E2', 'M3', 'jan@gmail.com', 07878941736, 'City Center Mall', 'NE4 2YZ', 'Newcastle', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO Branch

2 VALUES ('B4', 'E3', 'M4', 'jw@gmail.com', 07878941747, 'White House Crescent', 'DH6 2XY', 'Durham', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO Branch

2 VALUES ('B5', 'E4', 'M5', 'rebestate@gmail.com', 07878941758, 'South Hetton', 'DH4 2TN', 'Durham', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO Branch

2 VALUES ('B6', 'E4', 'M6', 'robestate@gmail.com', 07878941769, 'The Avenue', 'DH1 4XN', 'Durham', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO Branch

2 VALUES ('B7', 'E5', 'M7', 'aseon@gmail.com', 07878941789 , 'Chapel Frith', 'SK23 0GP', 'Sutherland', 'England');

1 row created.

W21029971 >

W21029971 >INSERT INTO Branch

2 VALUES ('B8', 'E5', 'M8', 'aseon@gmail.com', 07878941799, 'Prestbury', 'SK10 4XN', 'Sutherland', 'England');

1 row created.

W21029971 >

W21029971 >

W21029971 >INSERT INTO Staff

2 VALUES ('S1', 'B1', 'Haris', '07878951226', 'Hilton Park NE32 2WE', 'haris@gmail.com');

1 row created.

W21029971 >

W21029971 >INSERT INTO Staff

2 VALUES ('S2', 'B2', 'Ben', '07878891226', 'City Center NE4 2XY', 'ben@gmail.com');

1 row created.

W21029971 >

W21029971 >INSERT INTO Staff

2 VALUES ('S3', 'B3', 'Karen', '07875551226', 'Airport Road DH3 2WE', 'karen@gmail.com');

1 row created.

W21029971 >

W21029971 >INSERT INTO Staff

2 VALUES ('S4', 'B4', 'Paul', '07878456226', 'Coventry SK4 3NE', 'paul@gmail.com');

1 row created.

W21029971 >

W21029971 >INSERT INTO Staff

2 VALUES ('S5', 'B5', 'Logan', '07870981226', 'Riverside NE32 2XY', 'logan@gmail.com');

1 row created.

W21029971 >

W21029971 >INSERT INTO Staff

2 VALUES ('S6', 'B6', 'Hyman', '07878987226', 'Cross road DH1 4ER', 'hyman@gmail.com');

1 row created.

W21029971 >

W21029971 >INSERT INTO Staff

2 VALUES ('S7', 'B7', 'Ali', '07877871226', 'Houghton SK11 2WE', 'ali@gmail.com');

1 row created.

W21029971 >

W21029971 >INSERT INTO Staff

2 VALUES ('S8', 'B8', 'Kumar', '07878909826', '49 Hersham NE3 2WE', 'kumar@gmail.com');

1 row created.

W21029971 >

W21029971 >

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P1', 'C1', 'B1', 'NE32 2t3', 'Newcastle', 'England', 'Jesmond', 3, '200', '150000', 'DETACHED', 'FOR-SALE', '21-NOV-2021');

1 row created.

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P2', 'C2', 'B2', 'NE3 2TN', 'Newcastle', 'England', 'Jesmond', 3, '590', '250000', 'DETACHED', 'FOR-SALE', '11-NOV-2021');

1 row created.

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P3', 'C3', 'B4', 'DH3 2AV', 'Durham', 'England', 'City Center', 2, '400', '150000', 'SEMI-DETACHED', 'SOLD', '19-NOV-2019');

1 row created.

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P4', 'C1', 'B6', 'DH6 2AB', 'Durham', 'England', 'City Center', 4, '900', '250000', 'SEMI-DETACHED', 'SOLD', '10-SEP-2020');

1 row created.

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P5', 'C2', 'B7', 'SK3 2AB', 'Sutherland', 'England', 'Houghton', 3, '600', '250000', 'SEMI-DETACHED', 'SOLD', '12-NOV-2021');

1 row created.

W21029971 >

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P6', 'C3', 'B4', 'DH3 2t3', 'Durham', 'England', 'City Center', 2, '900', '250000', 'BUNGALOWS', 'RENTED', '19-AUG-2021');

1 row created.

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P7', 'C1', 'B4', 'DH3 2t3', 'Durham', 'England', 'City Center', 2, '400', '150000', 'SEMI-DETACHED', 'RENTED', '19-NOV-2021');

1 row created.

W21029971 >INSERT INTO PROPERTY VALUES ('P8', 'C2', 'B1', 'NE4 2t3', 'Newcastle', 'England', 'Fawdon', 2, '400', '200000', 'FLATS', 'FOR-SALE', '19-NOV-2021');

1 row created.

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P9', 'C3', 'B3', 'NE3 2TN', 'Newcastle', 'England', 'Jesmond', 2, '400', '175000', 'DETACHED', 'FOR-SALE', '15-NOV-2021');

1 row created.

W21029971 >INSERT INTO PROPERTY

2 VALUES ('P0', 'C1', 'B8', 'SK3 2YZ', 'Sutherland', 'England', 'City Center', 2, '600', '150000', 'SEMI-DETACHED', 'FOR-SALE', '09-MAY-2021');

1 row created.

W21029971 >

W21029971 >INSERT INTO Accommodation

2 VALUES ('1', 'P1', 20, 10, 200,'Comfortable rooms');

1 row created.

W21029971 >INSERT INTO Accommodation

2 VALUES ('2', 'P1', 20, 20, 400,'Backyard');

1 row created.

W21029971 >INSERT INTO Accommodation

2 VALUES ('3', 'P2', 20, 10, 200,'Comfortable rooms');

1 row created.

W21029971 >INSERT INTO Accommodation

2 VALUES ('4', 'P3', 20, 10, 200,'Comfortable rooms');

1 row created.

W21029971 >INSERT INTO Accommodation

2 VALUES ('5', 'P4', 20, 10, 200,'Comfortable rooms');

1 row created.

W21029971 >

W21029971 >INSERT INTO SoldProperty

2 VALUES ('S3', 'P3', 'C6', 250000, 40, 1000, '19-NOV-2019');

1 row created.

W21029971 >INSERT INTO SoldProperty

2 VALUES ('S4', 'P4', 'C4', 186000, 40, 1000, '10-NOV-2020');

1 row created.

W21029971 >INSERT INTO SoldProperty

2 VALUES ('S5', 'P5', 'C5', 225000, 20, 800, '12-NOV-2020');

1 row created.

W21029971 >

W21029971 >

W21029971 >INSERT INTO RentProperty

2 VALUES ('R1', 'P6', 'C7', 400, 800, '27-JAN-2019', '27-FEB-2023', 20);

1 row created.

W21029971 >INSERT INTO RentProperty

2 VALUES ('R2', 'P7', 'C8', 600, 800, '25-FEB-2020', '25-FEB-2023', 20);

1 row created.

W21029971 >INSERT INTO RentProperty

2 VALUES ('R5', 'P0', 'C7', 400, 800, '07-MAR-2019', '07-MAR-2023', 20);

1 row created.

W21029971 >

W21029971 >

W21029971 >INSERT INTO ViewProperty

2 VALUES ('V1', 'C7', 'P6', '27-JAN-2021', 'tenant', 'viewed', 'Nice house');

1 row created.

W21029971 >INSERT INTO ViewProperty

2 VALUES ('V2', 'C8', 'P7', '27-JAN-2022', 'tenant', 'not-viewed', 'Nice Terrace');

1 row created.

W21029971 >INSERT INTO ViewProperty

2 VALUES ('V3', 'C4', 'P1', '22-NOV-2021', 'buyer', 'viewed', 'Nice house');

1 row created.

W21029971 >INSERT INTO ViewProperty

2 VALUES ('V4', 'C5', 'P2', '02-JAN-2022', 'buyer', 'not-viewed', 'Nice house');

1 row created.

W21029971 >INSERT INTO ViewProperty

2 VALUES ('V5', 'C6', 'P5', '11-AUG-2021', 'buyer', 'viewed', 'Nice house');

1 row created.

W21029971 >

W21029971 >

W21029971 >

W21029971 >spool off

*(B) Answer the following queries (retrievals) using Relational Algebra and SQL.*

(10 marks)

1. Display details of *semi-detached* properties for sale having at least three bedrooms in the *Jesmond* area of Newcastle upon Tyne that were added to the system in the last 14 days.

Provide Relational Algebra expression below:

**p** property\_id, owner\_id, city, location, number\_of\_rooms, asking\_price, type, adding\_date, status**( s**type <= 'SEMI-DETACHED' And number\_of\_rooms >= 3 and city = 'Newcastle' and area = 'Jesmond' and status = 'FOR\_SALE' and adding\_date = sysdate - 14**(Property) )**

Provide SQL query code and output below:

SELECT property\_id,owner\_id,city,location,number\_of\_rooms,asking\_price,type,adding\_date,status from Property where type = 'SEMI-DETACHED' and number\_of\_rooms >= 3 and city = 'Newcastle'

and location = 'Jesmond' and status = 'FOR-SALE' and adding\_date >= sysdate - 14;

**Output:**

Graphical user interface, text

Description automatically generated

1. Display details of properties sold in Newcastle, Sunderland, Gateshead or Durham for £157,000 to £279,000 in the years 2019 or 2020.

Provide Relational Algebra expression below:

R(property\_id, owner\_id, city, location, number\_of\_rooms, selling\_price, stump\_duty, commission\_paid, selling\_date) ß property\_id, owner\_id, city, location, number\_of\_rooms, selling\_price, stump\_duty, commission\_paid, selling\_date ( scity = 'Newcastle' OR 'Sutherland' OR 'Durham' OR 'Gateshead' AND selling\_price >= 157000 AND selling\_price <= 279000 AND EXTRACT(YEAR FROM TO\_DATE(sold\_property.selling\_date, 'DD-MON-RR')) = 2019 OR 2020) ( Property ⋈ property\_id = property\_id SoldProperty))

Provide SQL query code and output below:

SELECT p.property\_id,p.owner\_id,p.city,p.location,p.number\_of\_rooms, sp.selling\_price,sp.stump\_duty,sp.cm\_paid, sp.selling\_date

from Property p INNER JOIN SoldProperty sp ON p.property\_id = sp.property\_id

where p.city IN ('Newcastle', 'Sutherland', 'Durham', 'Gateshead')

AND sp.selling\_price >= 157000 AND sp.selling\_price <= 279000

AND EXTRACT(YEAR FROM TO\_DATE(sp.selling\_date, 'DD-MON-RR')) IN (2019,2020);

**Output:**

Graphical user interface, text

Description automatically generated

**Part 3 (35 marks)**

This part is based on your answer / solution to Part 1 (A), i.e., conceptual design of the database for the MOVEHOME scenario.

*(A) Choose and justify what aspects of MOVEHOME conceptual design would be better off if implemented using object-relational database; then provide logical design and implementation of the subset of the MOVEHOME using ER/EER to object-relational mapping and object-relational features of Oracle Database System (Kannan); populate the object-tables with sample data and demonstrate your choice of design and implementation by running two complex queries on your object-tables.*

(20 marks)

Answer Part 3 A

1. Provide below your choice and justification of what aspects (subset) of the MOVEHOME conceptual design from Part 1.A you would like to implement using object relational databases (2 marks)
2. The subsets I want to use is Branch and Staff combination. Branch has also a reference to manager. As it does not correspond properly to the real-world entities and object relational can help in object sharing.
3. And the second one will be of Property with its child soldProperty and rentProperty by implementing the concepts of inheritance. The concept of object sharing and type will also suit here, as it will in faster processing and querying.
4. Provide below the logical design for your chosen subset using ER/EER to object-relational mapping (2 marks)

Branch and Staff:

Branch (code, manager, email, address, phone\_number, headquater)  
Staff (staff\_id, name, email, phone\_number, address)

Branch has many staff members working for a single branch. Therefore, Branch will have a nested table of REF to Staff and Staff will have a REF to Branch:

Branch (code, **manager: Ref Manager**, email, address, phone\_number, headquater, **staff: nested table [REF Staff]**)

Staff (staff\_id, name, email, phone\_number, address, **works for: REF Branch**)

Manager(manager\_id, name, email, phone\_number)

Branch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| code | manager | email | address | phone\_number | headquater | Staff |
| B1 | ManagerRef1\* | mk@gmail.com | Kingston Park | Newcastle | MK Estates | Nested Tab[StaffRef1, StaffRef2] |

Staff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Staff\_id | name | email | phone\_number | address | Works for |
| 1 | Abdullah | ab@gmail.com | 07878971442 | Fenham street | BranchRef1\* |
| 2 | John | john@gmail.com | 07896543212 | Gateshead | BranchRef2\* |

Manager

|  |  |  |  |
| --- | --- | --- | --- |
| manager\_id | name | email | phone\_number |
| 1 | Karen | karen@gmail.com | 07878654321 |

Property:

Property(property\_id, branch\_id, customer\_id, code, city, state, post\_code, asking\_price, rental\_price, added\_at)

SoldProperty(sold\_id, selling\_price, cm\_paid, selling\_date) inherits Property

RentProperty(rent\_id, rent, adv\_paid, commission, date\_tendancy\_started, date\_tenden y\_finished) inherits Property

1. Provide below the SQL code and output for implementing your above logical object-relational design (8 marks)

Branch and Staff:

Drop Type Branch\_t force;

Drop Type Staff\_nt\_type force;

Drop Type Manager\_t force;

Drop table Branch\_Tab;

Drop table Staff\_tab;

CREATE Type Branch\_t

/

CREATE Type Staff\_t AS Object

(staff\_id CHAR(2),

name VARCHAR(10),

email VARCHAR(20),

phone\_number NUMBER(11),

address VARCHAR(20),

Works\_for REF Branch\_t)

/

CREATE TYPE Staff\_nt\_type as Table of REF Staff\_t

/

CREATE Type Manager\_t as Object

(id CHAR(2),

name VARCHAR(20),

email VARCHAR(20),

phone\_number NUMBER(11)

)

/

CREATE OR REPLACE Type Branch\_t as Object

(code CHAR(2),

manager\_ref REF Manager\_t,

email VARCHAR(20),

phone\_number NUMBER(11),

headquater VARCHAR(20),

Staff Staff\_nt\_type)

/

Create Table Branch\_Tab of Branch\_t

(primary key (code))

Nested Table Staff Store As Branch\_Staff\_NTab;

Create Table Manager\_Tab of Manager\_t

(primary key (id))

/

Create Table Staff\_Tab of Staff\_t

(primary key (staff\_id))

/

Alter table Staff\_tab Add constraint Branch\_REF\_FK Foreign Key (works\_for) References Branch\_Tab

/

Spool File for Branch and Staff:

W21029971 >CREATE Type Branch\_t

2 /

Type created.

W21029971 >CREATE Type Staff\_t AS Object

2 (staff\_id CHAR(2),

3 name VARCHAR(10),

4 email VARCHAR(20),

5 phone\_number NUMBER(11),

6 address VARCHAR(20),

7 Works\_for REF Branch\_t)

8 /

Type created.

W21029971 >CREATE TYPE Staff\_nt\_type as Table of REF Staff\_t

2 /

Type created.

W21029971 >CREATE Type Manager\_t as Object

2 (id CHAR(2),

3 name VARCHAR(20),

4 email VARCHAR(20),

5 phone\_number NUMBER(11)

6 )

7 /

Type created.

W21029971 >CREATE OR REPLACE Type Branch\_t as Object

2 (code CHAR(2),

3 manager\_ref REF Manager\_t,

4 email VARCHAR(20),

5 phone\_number NUMBER(11),

6 headquater VARCHAR(20),

7 Staff Staff\_nt\_type)

8 /

Type created.

W21029971 >Create Table Branch\_Tab of Branch\_t

2 (primary key (code))

3 Nested Table Staff Store As Branch\_Staff\_NTab;

Table created.

W21029971 >Create Table Manager\_Tab of Manager\_t

2 (primary key (id))

3 /

Table created.

W21029971 >Create Table Staff\_Tab of Staff\_t

2 (primary key (staff\_id))

3 /

Table created.

W21029971 >Alter table Staff\_tab Add constraint Branch\_REF\_FK Foreign Key (works\_for) References Branch\_Tab

2 /

Table altered.

W21029971 >spool off

1. Provide below the SQL code and output for populating your above object-relational subset of the MOVEHOME database (4 marks)

* Populating data for Branch, Staff and Manager:

Manager:

Insert into Manager\_Tab values ('1', 'Karen', 'karen@gmail.com', 07878951344);

Insert into Manager\_Tab values ('2', 'Paul', 'paul@gmail.com', 07678951344);

Insert into Manager\_Tab values ('3', 'Lawson', 'lawson@gmail.com', 07678871344);

Insert into Manager\_Tab values ('4', 'Jack', 'jack@gmail.com', 07678809344);

**Staff:**

Insert into Staff\_Tab values ('1', 'Abdullah', 'ab@gmail.com', 07878965432, 'Kingston', NULL);

Insert into Staff\_Tab values ('2', 'Roy', 'roy@gmail.com', 07878923432, 'Fenham', NULL);

Insert into Staff\_Tab values ('3', 'Mike Tyson', 'mike@gmail.com', 07878345432, 'Gateshead', NULL);

Insert into Staff\_Tab values ('4', 'Talib', 'talib@gmail.com', 07878234122, 'Durham City', NULL);

Insert into Staff\_Tab values ('5', 'Ben', 'ben@gmail.com', 07878123432, 'Fountain', NULL);

**Branch:**

Insert into Branch\_Tab values ('1', (select REF(m) from Manager\_Tab m where m.id = '1'), 'abestates@gmail.com', 07878965432, 'Kingston Park', Staff\_nt\_type());

Insert into Branch\_Tab values ('2', (select REF(m) from Manager\_Tab m where m.id = '2'), 'durham@gmail.com', 07878965412, 'Fenham street', Staff\_nt\_type());

Insert into Branch\_Tab values ('3', (select REF(m) from Manager\_Tab m where m.id = '3'), 'johnest@gmail.com', 07878945432, 'Gateshead', Staff\_nt\_type());

Insert into Branch\_Tab values ('4', (select REF(m) from Manager\_Tab m where m.id = '14'), 'mkestates@gmail.com', 07878967732, 'South hilton', Staff\_nt\_type());

**Updating Staff:**

Update Staff\_Tab

Set

Works\_for = (select ref(b) from Branch\_Tab b where b.code = '1') where staff\_id = '1';

Update Staff\_Tab

Set

Works\_for = (select ref(b) from Branch\_Tab b where b.code = '1') where staff\_id = '2';

Update Staff\_Tab

Set

Works\_for = (select ref(b) from Branch\_Tab b where b.code = '2') where staff\_id = '3';

Update Staff\_Tab

Set

Works\_for = (select ref(b) from Branch\_Tab b where b.code = '3') where staff\_id = '4';

Update Staff\_Tab

Set

Works\_for = (select ref(b) from Branch\_Tab b where b.code = '4') where staff\_id = '5';

**Branch:**

INSERT INTO TABLE (Select b.Staff from Branch\_Tab b where b.code = '1')

SELECT REF(e) from Staff\_Tab e where e.works\_for.code = '1';

INSERT INTO TABLE (Select b.Staff from Branch\_Tab b where b.code = '2')

SELECT REF(e) from Staff\_Tab e where e.works\_for.code = '2';

INSERT INTO TABLE (Select b.Staff from Branch\_Tab b where b.code = '3')

SELECT REF(e) from Staff\_Tab e where e.works\_for.code = '3';

INSERT INTO TABLE (Select b.Staff from Branch\_Tab b where b.code = '4')

SELECT REF(e) from Staff\_Tab e where e.works\_for.code = '4';

* Spool file for Branch and Staff:

W21029971 >Insert into Manager\_Tab values ('1', 'Karen', 'karen@gmail.com', 07878951344);

1 row created.

W21029971 >Insert into Manager\_Tab values ('2', 'Paul', 'paul@gmail.com', 07678951344);

1 row created.

W21029971 >Insert into Manager\_Tab values ('3', 'Lawson', 'lawson@gmail.com', 07678871344);

1 row created.

W21029971 >Insert into Manager\_Tab values ('4', 'Jack', 'jack@gmail.com', 07678809344);

1 row created.

W21029971 >Insert into Staff\_Tab values ('1', 'Abdullah', 'ab@gmail.com', 07878965432, 'Kingston', NULL);

1 row created.

W21029971 >Insert into Staff\_Tab values ('2', 'Roy', 'roy@gmail.com', 07878923432, 'Fenham', NULL);

1 row created.

W21029971 >Insert into Staff\_Tab values ('3', 'Mike Tyson', 'mike@gmail.com', 07878345432, 'Gateshead', NULL);

1 row created.

W21029971 >Insert into Staff\_Tab values ('4', 'Talib', 'talib@gmail.com', 07878234122, 'Durham City', NULL);

1 row created.

W21029971 >Insert into Staff\_Tab values ('5', 'Ben', 'ben@gmail.com', 07878123432, 'Fountain', NULL);

1 row created.

W21029971 >Insert into Branch\_Tab values ('1', (select REF(m) from Manager\_Tab m where m.id = '1'), 'abestates@gmail.com', 07878965432, 'Kingston Park', Staff\_nt\_type());

1 row created.

W21029971 >Insert into Branch\_Tab values ('2', (select REF(m) from Manager\_Tab m where m.id = '2'), 'durham@gmail.com', 07878965412, 'Fenham street', Staff\_nt\_type());

1 row created.

W21029971 >Insert into Branch\_Tab values ('3', (select REF(m) from Manager\_Tab m where m.id = '3'), 'johnest@gmail.com', 07878945432, 'Gateshead', Staff\_nt\_type());

1 row created.

W21029971 >Insert into Branch\_Tab values ('4', (select REF(m) from Manager\_Tab m where m.id = '14'), 'mkestates@gmail.com', 07878967732, 'South hilton', Staff\_nt\_type());

1 row created.

W21029971 >Update Staff\_Tab

2 Set

3 Works\_for = (select ref(b) from Branch\_Tab b where b.code = '1') where staff\_id = '1';

1 row updated.

W21029971 >Update Staff\_Tab

2 Set

3 Works\_for = (select ref(b) from Branch\_Tab b where b.code = '1') where staff\_id = '2';

1 row updated.

W21029971 >Update Staff\_Tab

2 Set

3 Works\_for = (select ref(b) from Branch\_Tab b where b.code = '2') where staff\_id = '3';

1 row updated.

W21029971 >Update Staff\_Tab

2 Set

3 Works\_for = (select ref(b) from Branch\_Tab b where b.code = '3') where staff\_id = '4';

1 row updated.

W21029971 >Update Staff\_Tab

2 Set

3 Works\_for = (select ref(b) from Branch\_Tab b where b.code = '4') where staff\_id = '5';

1 row updated.

W21029971 >INSERT INTO TABLE (Select b.Staff from Branch\_Tab b where b.code = '1')

2 SELECT REF(e) from Staff\_Tab e where e.works\_for.code = '1';

2 rows created.

W21029971 >

W21029971 >INSERT INTO TABLE (Select b.Staff from Branch\_Tab b where b.code = '2')

2 SELECT REF(e) from Staff\_Tab e where e.works\_for.code = '2';

1 row created.

W21029971 >

W21029971 >INSERT INTO TABLE (Select b.Staff from Branch\_Tab b where b.code = '3')

2 SELECT REF(e) from Staff\_Tab e where e.works\_for.code = '3';

1 row created.

W21029971 >

W21029971 >INSERT INTO TABLE (Select b.Staff from Branch\_Tab b where b.code = '4')

2 SELECT REF(e) from Staff\_Tab e where e.works\_for.code = '4';

1 row created.

W21029971 >spool off;

1. Provide below the SQL code and output for running two complex queries on the object-relational subset of the above MOVEHOME database (4 marks)

* Displaying Staff details of Branch with ID = 1

select s.name, s.email, s.phone\_number, s.Works\_for.code from Staff\_Tab s where s.Works\_for.code = '1';

Output:

Text

Description automatically generated

* Getting manager detail of branches in Kingston Park and Gateshead:

Select b.manager\_ref.name "Name", b.manager\_ref.email "Email", b.manager\_ref.phone\_number "Phone Number" from Branch\_Tab b where b.headquater IN('Kingston Park', 'Gateshead');

Output:

Text

Description automatically generated

*(B) Analyse the conceptual database design from Part 1 (A) and the MOVEHOME scenario in the Appendix and propose what aspects of the MOVEHOME database would benefit from incorporating NoSQL Database concepts. Illustrate your answer with code from a representative code from NoSQL Database implementation.*

(15 marks)

Answer Part 3 B

1. Provide below your choice and justification of what aspects (subset) of the MOVEHOME databases would benefit from incorporating NoSQL Database concepts (3 marks)

Aspects that can benefit from NoSQL Database are Property, Estate, Branch and View. NOSQL is based on key-value pair unlike relational database and its best choice for unstructured data.

Firstly, Estate Agent, as it can have multiple branches as well as staff members. So, this data can be handled in NOSQL Database.  
  
As property has multiple types for sell and rent and both these entities has its own attributes and can be keep on changing in the future and that can be stored with dynamic schema. Moreover, there data can be stored in a single table in NOSQL database which can even result in faster processing when there are more records. Moreover, as property has multiple accommodation for size and description this part can be also handled in NOSQL Database. Adding to it, as in the future more information related to property can be added for example, adding pictures and videos related to marketed properties, so for relational database will require change of scheme, so here NOSQL database can be of great advantage.  
  
In view there can be multiple comments, feedback, and rate in the future. Moreover, there can be multiple comments before and after viewing so data can be unstructured.

1. Provide below code and output for implementing your proposed NoSQL Database subset of the MOVEHOME database, populate it with some data, and example queries & outputs (12 Marks)

1. **Creating and Inserting data into EstateAgent Collection:**

db.createCollection("EstateAgent")

db.EstateAgent.insertMany( [ { \_id: "1", name: "Mason Estate", email: "mason@gmail.com", phone\_number: 07878951552, headquater: "Masons Headquater", address\_line: "Gateshead", city: "Newcastle", state: "England"},

{\_id: "2", name: "Ben Estate", email: "ben@gmail.com", phone\_number: 07878951252, headquater: "Ben Headquater", address\_line: "Jesmond", city: "Newcastle", state: "England" },

{ \_id: "3", name: "MA Estate", email: "ma@gmail.com", phone\_number: 07878951232, headquater: "MA Headquater", address\_line: "Gateshead", city: "Durham", state: "England"}]);

**Output:**

Text

Description automatically generated

1. **Creating and inserting data in to Branch Collection:**

db.createCollection("Branch")

db.Branch.insertMany( [ { \_id: "1", agent\_id: "E1", manager: "Mason", phone\_number: 07878951552, headquater: "Masons Headquater", address\_line: "Gateshead", city: "Newcastle", state: "England"},

{ \_id: "2", agent\_id: "1", manager: "Ben", phone\_number: 07878916752, headquater: "Ben Headquater", address\_line: "Gateshead", city: "Newcastle", state: "England"},

{ \_id: "3", agent\_id: "2", manager: "Karen", phone\_number: 07878934552, headquater: "Karen Headquater", address\_line: "Durham City", city: "Durham", state: "England"}]);

**Output:**

Graphical user interface, text

Description automatically generated

1. **Creating and inserting data into Staff Collection:**

db.createCollection("Staff")

db.Staff.insertMany( [ { \_id: "1", name: "Emel", branch\_id: "1", phone\_number: 07878951552, address\_line: "Gateshead", city: "Newcastle", state: "England"},

{\_id: "2", name: "Joe", branch\_id: "2", phone\_number: 07878951552, address\_line: "Gateshead", city: "Newcastle", state: "England"},

{\_id: "3", name: "Imran", branch\_id: "2", phone\_number: 07878951552, address\_line: "Durham City", city: "Durham", state: "England"}]);

**Output:**

Text

Description automatically generated

1. **Showing Staff branch headquarters through JOIN operation:**

db.createView (

"Staff\_Br",

"Staff",

[

{$lookup: {from: "Branch",

localField: "\_id",

foreignField: "\_id",

as: "brs"}}

]

);

db.Staff\_Br.find({}, {name: 1, city: 1, state: 1, "brs.headquater": 1});

**Output:**

Text

Description automatically generated with medium confidence

1. **Embedding the three tables for Estate Agent, Branch and Staff through NOSQL:**

db.createCollection("EstateAgents");

db.EstateAgents.insert(

{ est\_id: "E2", name: "Ben Estate", email: "ben@gmail.com", phone\_number: 07878951252, headquater: "Ben Headquater", address\_line: "Jesmond", city: "Newcastle", state: "England",

branch: // embedded branches

[ {code: "B1", agent\_id: "E1", manager: "Mason", phone\_number: 07878951552, headquater: "Masons Headquater", address\_line: "Gateshead", city: "Newcastle", state: "England", staff: [ { staff\_id: "1", name: "Emel", branch\_id: "B1", phone\_number: 07878951552, address\_line: "Gateshead", city: "Newcastle", state: "England"},

{staff\_id: "2", name: "Joe", branch\_id: "B1", phone\_number: 07878951552, address\_line: "Gateshead", city: "Newcastle", state: "England"}]},

{code: "B2", agent\_id: "E2", manager: "Ben", phone\_number: 07878916752, headquater: "Ben Headquater", address\_line: "Gateshead", city: "Newcastle", state: "England", staff: [ { staff\_id: "3", name: "Eman", branch\_id: "B2", phone\_number: 07878951552, address\_line: "Gateshead", city: "Newcastle", state: "England"},

{staff\_id: "4", name: "Abdul", branch\_id: "B2", phone\_number: 07878951552, address\_line: "Gateshead", city: "Newcastle", state: "England"}]}]

});

**Output:**

Graphical user interface, text

Description automatically generated

1. **Query for displaying details of EstateAgent with branch\_id = 1:**

db.EstateAgents.find(

{"branch": {$elemMatch: {"code": "B1"}}},

{name: 1,

phone\_number: 1,

email: 1,

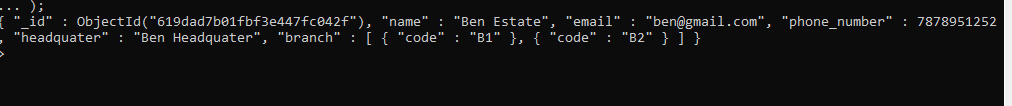
headquater: 1,

"branch.code": 1,

}

);

**Output:**



1. **Creating and Inserting data in ViewProperty Collection:**

db.createCollection("ViewProperty");

db.ViewProperty.insert(

{ \_id: "1",

customer\_id: "C1",

property\_id: "P2",

date: ISODate("2020-04-20"),

customer\_type: "buyer",

status: "viewed",

comments:

[ {id: "1", text: 'Nice house with balcony'},

{id: "2", text: 'Needs to be refurnished'}]

});

db.ViewProperty.insert(

{ \_id: "2",

customer\_id: "C2",

property\_id: "P2",

date: ISODate("2021-12-10"),

customer\_type: "tenant",

status: "not-viewed",

comments:

[ {id: "1", text: 'Still waiting for the date'}]

});

db.ViewProperty.insert(

{ \_id: "3",

customer\_id: "C3",

property\_id: "P2",

date: ISODate("2019-06-20"),

customer\_type: "buyer",

status: "viewed",

comments:

[ {id: "1", text: 'Nice house with balcony'} ]

});

**Output:**

Text

Description automatically generated

1. **Displaying results where property is viewed, and Customer is Buyer:**

db.ViewProperty.find( {"customer\_type" : "buyer", "status" : "viewed"});

**Output:**

A screen shot of a computer

Description automatically generated with low confidence

1. **Creating and Inserting data in Property Collection:**

db.createCollection("Property");

db.Property.insert(

{ \_id: "1",

owner\_id: "C1",

branch\_id: "B1",

city: "Newcastle",

state: "England",

postal\_code: "NE3 2TN",

location: "Fenham",

number\_of\_rooms: 4,

rental\_demand: null,

asking\_price: 110000,

type: "semi-detached",

status: "sold",

adding\_date: ISODate("2020-02-25"),

SoldProperty: {\_id: "1", buyer\_id: "C2", selling\_price: 120000, stump\_duty: 20, cm\_paid: 2000, selling\_date: ISODate("2021-04-12")}

});

db.Property.insert(

{ \_id: "2",

owner\_id: "C3",

branch\_id: "B2",

city: "Durham",

state: "England",

postal\_code: "DH3 2XY",

location: "Durham City",

number\_of\_rooms: 3,

rental\_demand: null,

asking\_price: 220000,

type: "Detached",

status: "sold",

adding\_date: ISODate("2018-09-01"),

SoldProperty: {\_id: "S2", buyer\_id: "C4", selling\_price: 120000, stump\_duty: 20, cm\_paid: 2000, selling\_date: ISODate("2021-04-12")}

});

db.Property.insert(

{\_id: "3",

owner\_id: "C5",

branch\_id: "B2",

city: "Newcastle",

state: "England",

postal\_code: "NH3 2XY",

location: "Gateshead",

number\_of\_rooms: 3,

rental\_demand: 500,

asking\_price: null,

type: "detached",

status: "FOR-RENT",

adding\_date: ISODate("2019-10-01"),

Accomodation: [{

\_id: "1", length: 20, width: 20, area: 400, description: "Good"

}, {

\_id: "2", length: 20, width: 30, area: 460, description: "Need Improvement"

}],

RentProperty: {\_id: 'R1', rent: 300, adv\_paid: 120, commission: 30, date\_tenancy\_commenced: ISODate("2019-10-28"), date\_tenancy\_finished: ISODate("2021-10-28")}

});

db.Property.insert(

{\_id: "4",

l\_id: "c2",

branch\_id: "3",

postal\_code: "NE3 2TN",

city: "Newcastle",

state: "England",

area: "Central",

no\_of\_rooms: 3,

rental\_demand: 600,

asking\_price: null,

type: "detached",

status: "FOR-RENT",

added\_at: ISODate("2019-10-26"),

Accomodation: [{

\_id: "1", length: 20, width: 20, area: 400, description: "Good"

}, {

\_id: "2", length: 20, width: 30, area: 460, description: "Need Improvement"

}],

RentProperty: {rent\_id: 'R2', property\_id: 'P2', rent: 400, adv\_paid: 220, commission: 30, date\_tenancy\_commenced: ISODate("2019-10-28"), date\_tenancy\_finished: ISODate("2021-10-28")}

});

db.Property.insert(

{ \_id: "5",

owner\_id: "C8",

branch\_id: "B5",

city: "Newcastle",

state: "England",

postal\_code: "NH3 2XY",

location: "Gateshead",

number\_of\_rooms: 3,

rental\_demand: 500,

asking\_price: null,

type: "Detached",

status: "FOR-SALE",

adding\_date: ISODate("2019-10-01"),

Accomodation: [{

\_id: "1", length: 20, width: 20, area: 400, description: "Good"

}, {

\_id: "2", length: 20, width: 30, area: 460, description: "Need Improvement"

}]

});

**Output:**

Text

Description automatically generated

1. **Displaying detached properties available to rent in Newcastle:**

db.Property.find({

status: 'FOR-RENT', city: 'Newcastle', type = 'Detached'

} );

**Output:**

A picture containing text

Description automatically generated

1. **Properties sold in Fenham and Gateshead between 2018 and 2021:**

db.Property.find({

status: 'sold', location: {$in: ['Fenham', 'Gateshead']}, city: 'Newcastle','SoldProperty.selling\_date': { $gt: new Date('2018-01-01'), $lt: new Date('2021-12-29') }

} );

**Output:**

Text

Description automatically generated

**Part 4 (10 marks)**

*Consider the MOVEHOME scenario in the Appendix. Produce a report for the managing director of the MOVEHOME group* *elaborating on professional, legal, ethical and security issues that need to be considered and make recommendations that you think are appropriate for MOVEHOME.*

(10 marks)

The report should be concise and comprehensive and in the region of 800-900 words. You should use Harvard style of citation and referencing by following the guidelines in Pears and Shields (2008).

**Answer Part 4: 10 Marks** [8 for the quality of report covering all the above issues, 1 for the quality of referencing and citation and adhering to the Harvard style, 1 for presentation]

**Introduction and Background:**

Ethics concerns with a set of moral values and behavior performed by individuals or a group of people (TechTerms.com, 2021). There can be multiple definitions of ethics and one that is legal but illegal in some other age, sex, or religion. As Computer Science field is growing rapidly, with it a lot of security breaches and other concerns are rising, so due to this computer ethics should be a part of concern while developing new technologies. Now a days, there are a lot of security issues like hacking, spoofing, as well as there is also breach of intellectual property for example, privacy policies and online etiquette (TechTerms.com, 2021).  
In the past IT professionals working in the organization has been highly affected by the actions put by their managers and executives, but now this is not the case as the rules have changed. As the amount of data and its processing day by day and even some organizations have tera and Peta bytes of data and with this leverage the IT have been asked to make use of this data and put it in use as per business point of view and with this they are asked to do some unethical things by their supervisors. Data is fuel now a days, so there is a lot of personal data of individuals in bank loan system, online ecommerce sites, social media sites and many others, so it’s important to secure data and reduce risk of any data exposure (Connolly & Begg, 2015).

**Report:**

This report aims into legal, ethical and security in databases and is going to provide a detail report to manager of *MOVEHOME*.

As a business, *MOVEHOME* interacts with different property owners, that want to sell and rent their properties and provide a marketing platform. There is a lot of rentals and sold history with different transactions for each, so this information is important and sensitive. Some rules that should be addressed are:

* **Interaction between MOVEHOME staff, customers, and owners:**
* Treating customers with respect over email and phone and fulfilling their needs.
* Treating owners well.
* Handling information requests and data properly, plus don’t disclose it to owners as well (Connolly & Begg, 2015).
* **Security of Customers and other business data:**
* Securing customers data like their properties, transaction history, as well rented and sold properties data. This can be done by limiting the inappropriate access of dataset (Connolly & Begg, 2015).
* Ensure proper measures are taken to prevent any data breaching and exposure. The data should be documented, its creation and structured should be properly defined to be safe of any kind of data manipulation.
* There should be proper measures for any kind of data requests made like from other employees for accessing sensitive data for their personal gains or data mining. Other requests include like resetting password, making sure of the rules for sharing data with owners and other law enforcements like for customers transaction history and other properties history.   
  Restrictions should be applied on user access to data and privileges should be monitored periodically (Connolly & Begg, 2015).
* Rules for using organization resources like hardware, software, and Internet. The points like company’s hardware should be only within company’s premises and will need managers approval to use its outside.   
  Licensed software should not be distributed or used improperly for own personal purposes. Any additional software should only be installed with approval. Lastly, Internet resources should not be used for business related purposes outside of company.
* There should be proper security system like firewalls and other intrusion detection system.
* Sensitive and important data should be encrypted with proper algorithms and these keys should be changed on regular basis.
* **Database security plan:**

With customer and other data security, its also important to maintain privacy infringement and provide security against other malicious activities like Identity theft (Essays, 2017). The data should be protected against various kind of malicious attacks like Trojan, malware, adware, spyware, and other virus types. But one of the biggest concerns here is protection against SQL injection (Essays, 2017). It does not have direct harm on end user, but it can affect a website by getting access to the database and by retrieving all the valuable information. There are some rules to prevent all these concerns.

* Strong username and passwords should be created. The users should be properly authenticated as well.
* If there are any built-in usernames and passwords, it should be replaced or changed as soon as possible.
* Review database configuration regularly and delete any unwanted or unused components. Avoid making complex systems and only install necessary components that are required (Essays, 2017).
* Keep software and hardware updated and latest. Moreover, only install security patches as required.
* Data is usually breached from servers, so it’s important to understand all the security patches while using third-party servers.
* Take measures to determine how much data each authorized user can access through granular access control. For example, one user should not be able to access other users’ personal data (Essays, 2017).
* Keep regular backup of data on disk or third-party servers. Ensure, it is properly encrypted so, it is not accessible for viewing (Essays, 2017).
* Maintain encryption of incoming and outgoing data for proper security.
* Everything should be documented and there should be a report log file for history and other changes.
* **Ramifications for violating security:**
* All kind of violations will be presented to board representatives and other upper management. And, in case if there is any serious violation it will be reported to the authorities (Connolly & Begg, 2015).
* If there is any malicious or any kind of intended violation may result in dismal or some serious action. The seriousness will be determined by the board.

Finally, this policy for *MOVEHOME* should be reviewed annually or on regular basis to be protected from any kind of security violations. Also, to make sure the policy is not outdated because of the latest technology changes.

# References

Connolly, T. & Begg, C., 2015. *A Practical Approach to Design, Implementation.* 6th ed. Boston: Pearson.

Essays, U., 2017. *Database Management: Law, Ethics and Security.* [Online]   
Available at: https://www.ukessays.com/essays/computer-science/database-management-law-ethics-9552.php

TechTerms.com, 2021. *Computer Ethics.* [Online]   
Available at: https://techterms.com/definition/computer\_ethics