Northumbria University NEWCASTLE

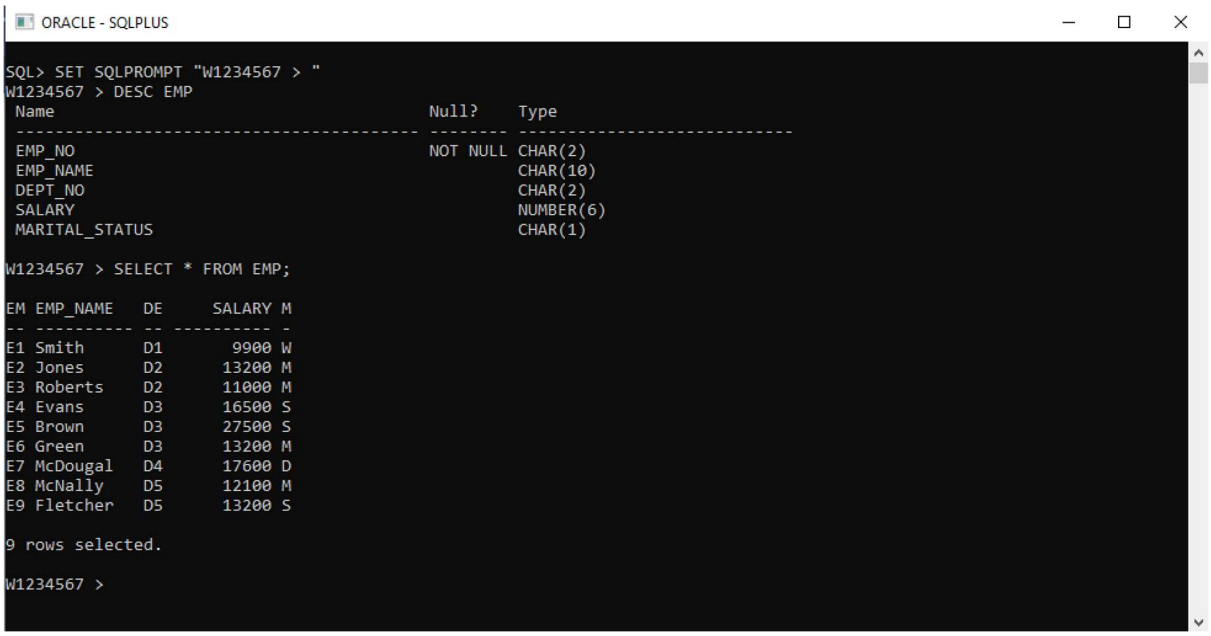
Department of Computer & Information Sciences

ASSESSMENT SUBMISSION	
Module Title:	Data Modelling and Analytics
Module Code:	PE7044
Academic Year:	2021-22
Module Tutor / Email (all queries):	Akhtar Ali [redacted]
% Weighting (to overall module):	100%
Assessment Title:	Written coursework on analysis, design and implementation of database and analytics using a range of contemporary tools and methods
Date of Handout to Students:	13th May 2022
Mechanism for Handout:	Module Blackboard Site & Live Session in Week 4
Deadline for Submission Attempt by Students:	3rd July 2022 @ 23:59 British Summer Time (BST) which is equal to GMT + 1
Mechanism for Submission:	Document upload to Module Blackboard Site
Submission Format / Word Count	Please upload your written report as a single PDF document (approximately 4000 words).
Date by which Work, Feedback and Marks will be returned:	1st August 2022
Mechanism for return of Feedback and Marks:	Mark and written feedback will be uploaded to the Module Site on Blackboard. For further queries please email module tutor.
Student ID **Oracle Username** **Data Mining Username**	[redacted]
Student Name	Scott CUMMING

SUMMATIVE ASSIGNMENT

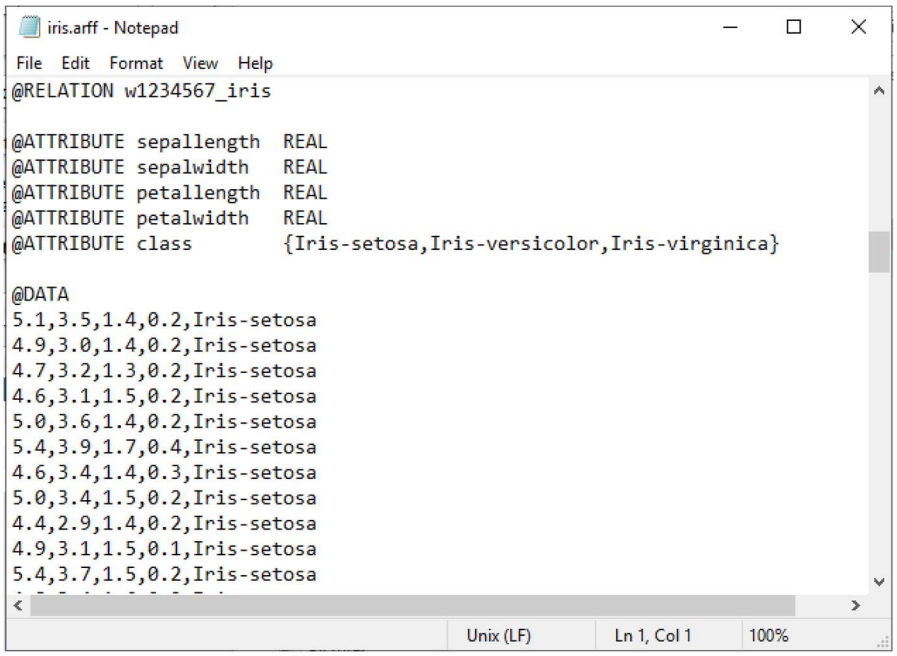
Personalising your SQL output/prompt

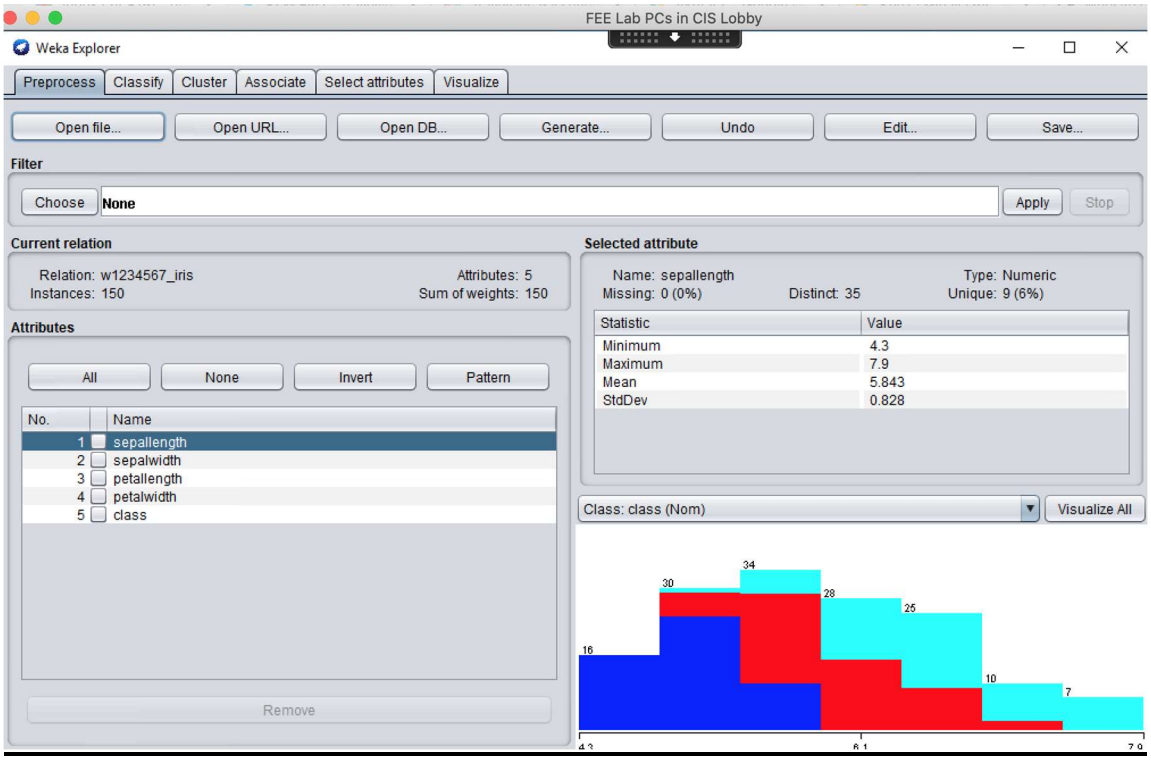
Before executing any SQL code for the assignment, you should personalise your SQL output / prompt by running SET SQLPROMPT “YourUserName > ”, i.e., double-quote followed by your UserName followed by > and then a space and double-quote as shown in the screenshot below:

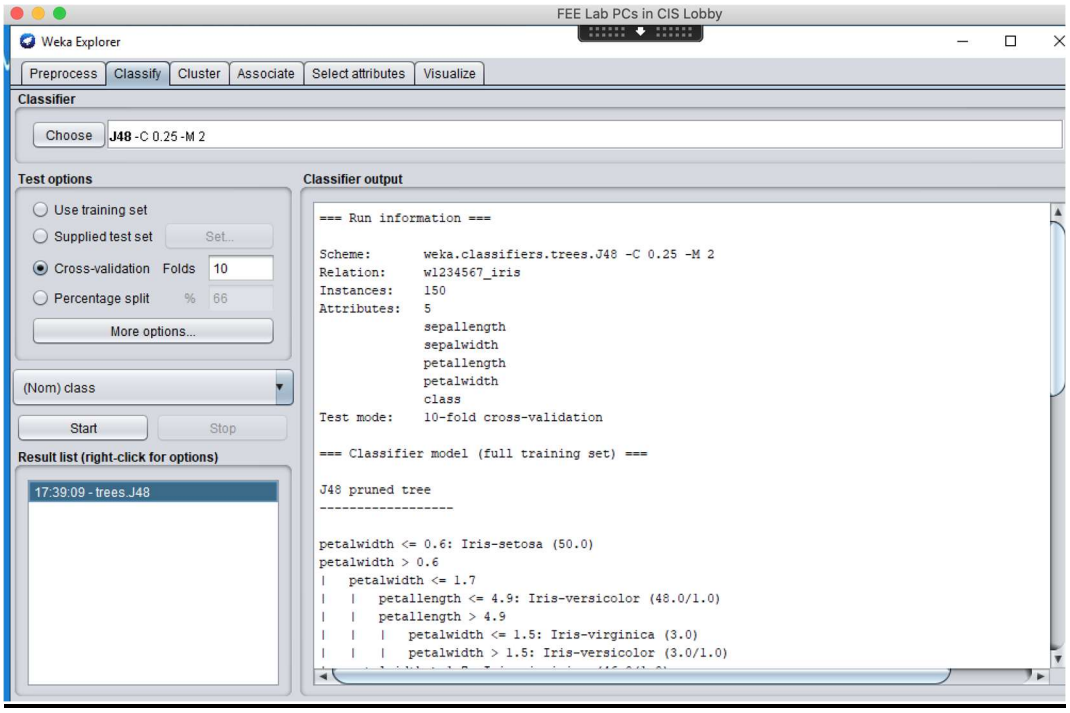


Personalising your WEKA output

Before you process any of the dataset, open the arff file in a text editor (e.g., Notepad) and prefix the name of the relation with your username followed by _. For example, @RELATION w1234567_iris as shown in the screenshots below.

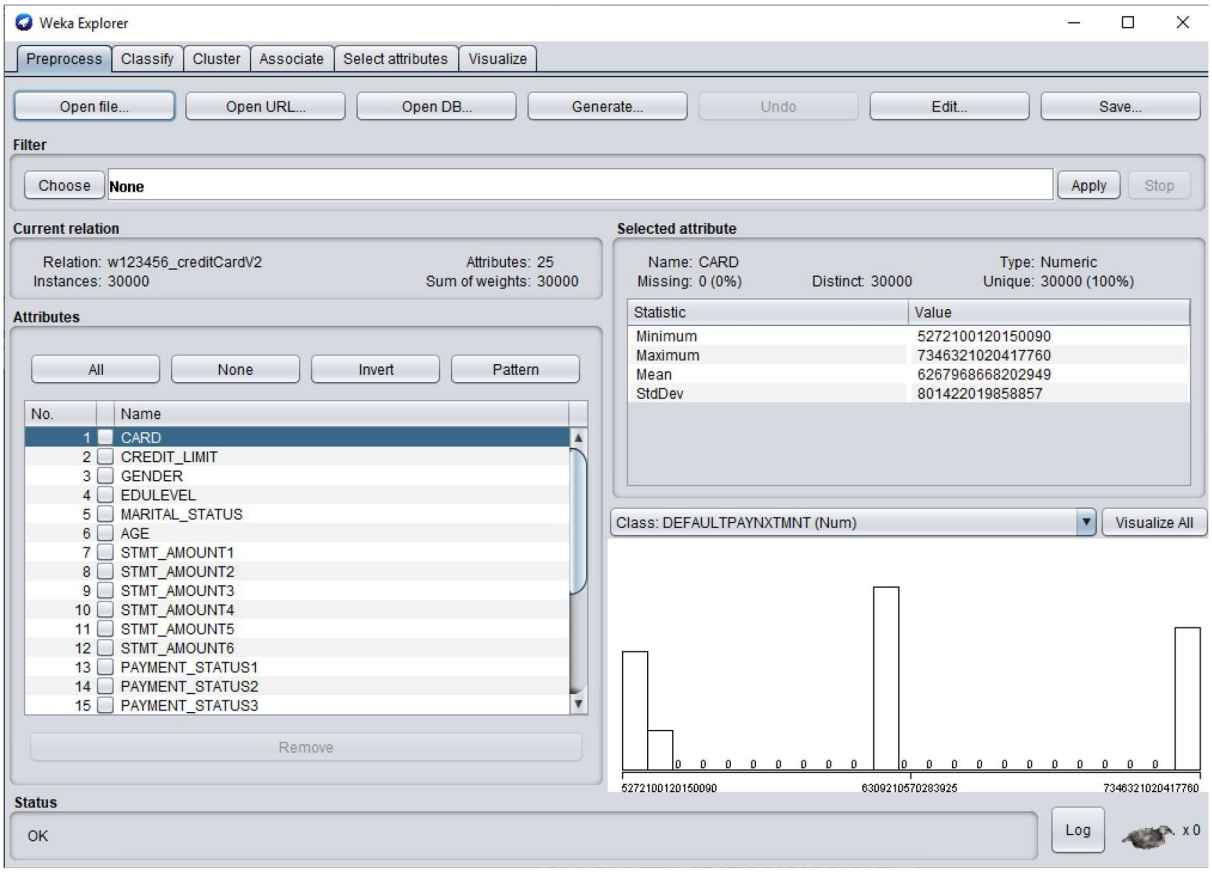



FEE Lab PCs in CIS Lobby
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: w1234567_iris
Instances: 150
Attributes: 5
Sum of weights: 150
Selected attribute
Name: sepallength
Missing: 0 (0%)
Distinct: 35
Type: Numeric
Unique: 9 (6%)
Attributes
All
None
Invert
Pattern
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Statistic
Value
Minimum 4.3
Maximum 7.9
Mean 5.843
StdDev 0.828
Class: class (Nom)
Visualize All
Remove


FEE Lab PCs in CIS Lobby
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
17:39:09 - trees.J48
Classifier output
=== Run information ===
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: w1234567_iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
class
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
| petalwidth <= 1.7
| | petallength <= 4.9: Iris-versicolor (48.0/1.0)
| | petallength > 4.9
| | | petalwidth <= 1.5: Iris-virginica (3.0)
| | | petalwidth > 1.5: Iris-versicolor (3.0/1.0)

Personalising your WEKA output using a CSV file

Before you process any CSV dataset, rename the CSV file by prefixing it with your username followed by _. For example, `w123456_creditCardV2.csv` as shown in the screenshot below when the file is opened in Weka Explorer.



Assignment Questions

Part 1 (30 marks)

(A) Using a database design approach of your choice, produce a logical design for the database to support the information system, which is needed at PropertyPortal.

(20 marks)

Your answer must consist of **ONE** of the following:

- *An entity-relationship (ER) diagram (10 marks) and its mapping into a set of relations (10 marks). The ER diagram should show all relevant entity types, relationship types, attributes, keys, and structural constraints. Note that not all keys are identified/mentioned in the scenario, so you are required to identify/devise appropriate keys for all the entity types. Your ER diagram must not show any foreign keys.*

  As part of the mapping process, for each relation, you should identify appropriate primary keys as well as foreign keys (if applicable). Furthermore, you need to make sure your relations obtained from mapping your ER diagram are in 3rd normal form.

- *A set of normalised relations (10 marks) obtained through applying the normalisation process (10 marks) instead of ER modelling. You should make clear how the normalisation process has been carried out, and the reasoning employed, in particular quoting evidence (series of steps) to support the decisions made and how various relations have been derived. Each final relation in your answer should be in 3rd normal form.*

Answer Part 1A:

The PropertyPortal conceptual data model was created using an Enhanced Entity Relationship (EER) diagram, which is shown below on the next page as Figure 1. It is an enhanced diagram as superclasses and subclasses were used (Connolly and Begg, 2015, p.435). The entity Customer was established as a superclass and both Property_Seeker and Owner were designated subclasses. This approach was taken as the classes more accurately represent the relationships with the other entities in the model. Similarly, Property was made a superclass with both Sales_Property and Rental_Property as its subclasses. As well as being more representational, a lot of nulls are prevented by using this approach, as both Sales_Property and Rental_Property have distinct attributes which would not always require a value to be entered if included in a generalised Property class.

It was noted that the following entities all contained address details: Estate_Agent, Branch, Staff_Member, Property and Customer. Including these details in each entity would have violated the Don't Repeat Yourself (DRY) principle, therefore a general Address entity was created to avoid this and relationships with the other entities were formed.

The conceptual data model was mapped into a set of relations to create a logical database design, as shown in Figure 2 on the following pages. Elmasri and Navathe (2017, pp.328-331) offer guidance on how to map superclasses and subclasses to a logical model. Using this guidance, the Sales_Property and Rental_Property subclasses were retained as separate relations from their Property superclass, adopting the latter's primary key. This was done to avoid creating many nulls in a generalised Property class, as noted above. Conversely, the Property_Seeker and Owner subclasses were incorporated into the Customer superclass as one relation, because they don't possess any distinct attributes.

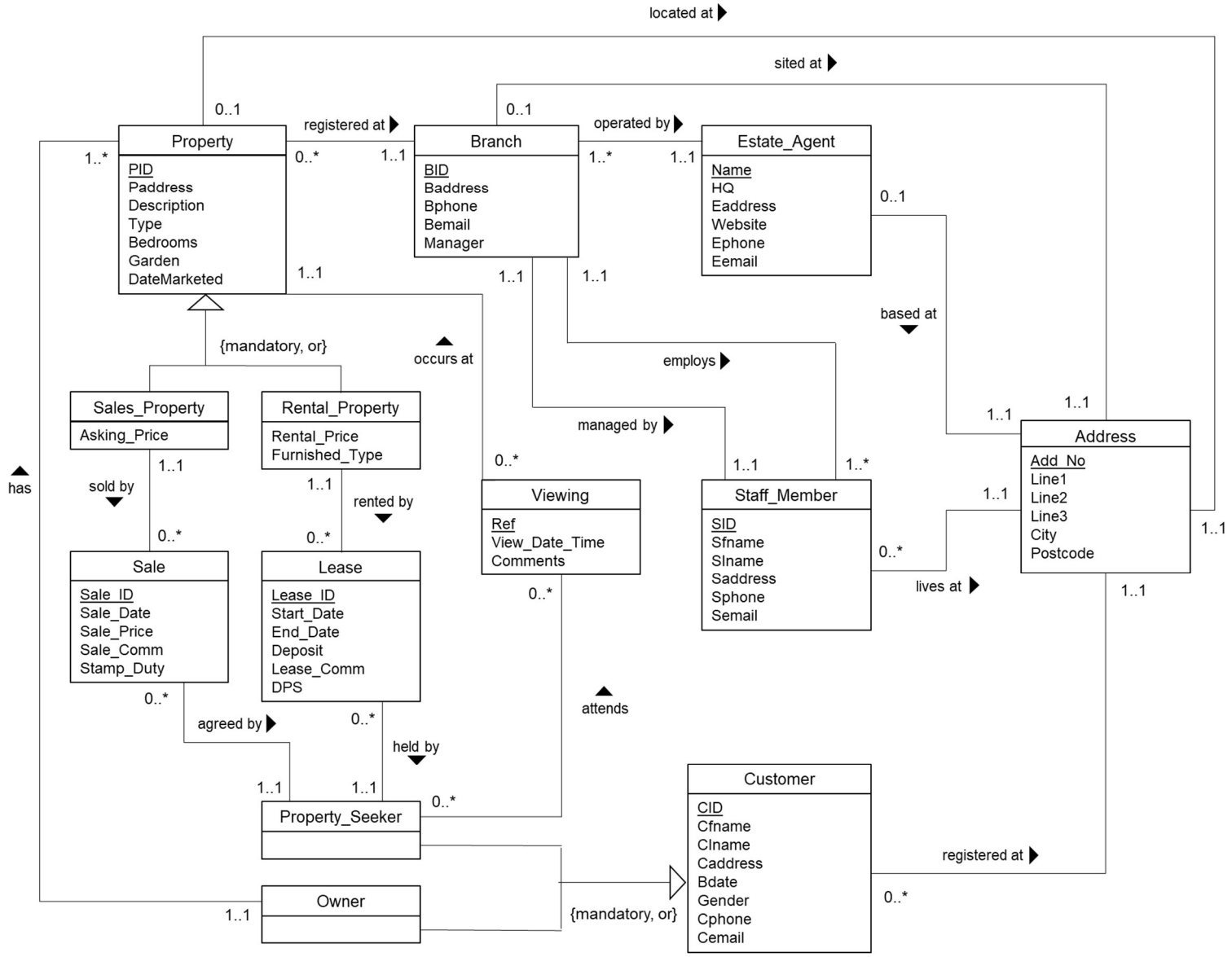


Figure 1 – PropertyPortal Enhanced Entity Relationship (EER) Diagram

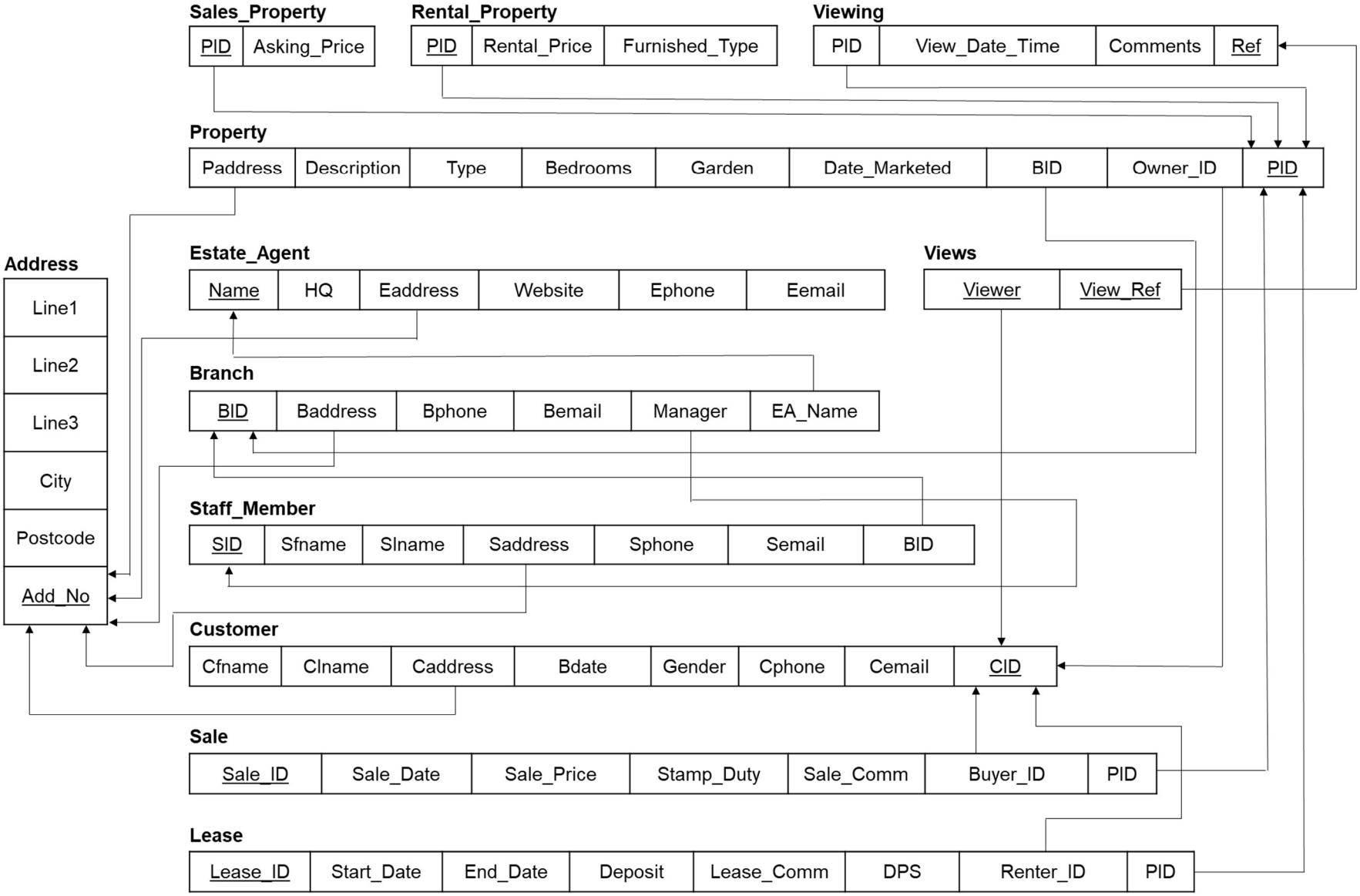


Figure 2 – PropertyPortal Mapped Logical Database Design

(B) Based on your logical design from Part 1 (A) and the information available in the scenario, produce an SQL script file using Oracle 11g/12c/19c.

(10 marks)

Answer Part 1B: SQL DDL Script file contents: 10 marks

```
SET ECHO ON
SET VERIFY ON

Rem   SQL script file used to create relations for the PropertyPortal
database.
Rem   Provides solution to PE7044 assessment part 1B for student S CUMMING
(21056374).

CREATE TABLE ADDRESS
(ADD_NO           CHAR(8)           CHECK (REGEXP_LIKE (ADD_NO,
'^AD\d{6}$')) CONSTRAINT PKEY_ADD PRIMARY KEY,
 LINE1            VARCHAR(30),
 LINE2            VARCHAR(30),
 LINE3            VARCHAR(30),
 CITY       VARCHAR(20),
 POSTCODE   VARCHAR(8)
);

CREATE TABLE PROPERTY
(PID        CHAR(7)           CHECK (REGEXP_LIKE (PID, '^P\d{6}$'))
CONSTRAINT PKEY_PROP PRIMARY KEY,
 PADDRESS   CHAR(8)           NOT NULL,
 DESCRIPTION      VARCHAR(500),
 TYPE       CHAR(1)           NOT NULL CONSTRAINT TYPE_RANGE CHECK (TYPE IN
('D', 'S', 'T', 'F', 'B')),
 BEDROOMS   NUMBER(2)   NOT NULL CONSTRAINT BED_NO CHECK (BEDROOMS > 0),
 GARDEN           CHAR(1)           NOT NULL CONSTRAINT IS_GARD CHECK
(GARDEN IN ('Y', 'N')),
 DATE_MARKETED    DATE        NOT NULL,
 OWNER_ID   CHAR(7)           NOT NULL,
 BID        CHAR(3)           NOT NULL
);

CREATE TABLE SALES_PROPERTY
(PID        CHAR(7)           CONSTRAINT PKEY_SALES_PROP PRIMARY KEY,
 ASKING_PRICE     NUMBER(10,2)      NOT NULL CONSTRAINT ASK_PRICE CHECK
(ASKING_PRICE > 0)
);

CREATE TABLE RENTAL_PROPERTY
(PID        CHAR(7)           CONSTRAINT PKEY_RENTAL_PROP PRIMARY KEY,
 RENTAL_PRICE     NUMBER(7,2) NOT NULL CONSTRAINT RENT_PRICE CHECK
(RENTAL_PRICE > 0),
 FURNISHED_TYPE   CHAR(2)           NOT NULL CONSTRAINT IS_FURNISHED CHECK
(FURNISHED_TYPE IN ('FF', 'PF', 'UF'))
```

```
);

CREATE TABLE VIEWING
(REF        CHAR(9)           CHECK (REGEXP_LIKE (REF, '^V\d{8}$'))
CONSTRAINT PKEY_VIEWING PRIMARY KEY,
 VIEW_DATE_TIME   TIMESTAMP   NOT NULL,
 COMMENTS   VARCHAR(500),
 PID        CHAR(7)           NOT NULL
);

CREATE TABLE ESTATE_AGENT
(NAME       VARCHAR(30) CONSTRAINT PKEY_ENAME PRIMARY KEY,
 HQ         VARCHAR(20) NOT NULL,
 EADDRESS   CHAR(8)           NOT NULL,
 WEBSITE    VARCHAR(50) NOT NULL,
 EPHONE           CHAR(11)    NOT NULL,
 EEMAIL           VARCHAR(50) NOT NULL
);

CREATE TABLE VIEWS
(VIEWER          CHAR(7),
 VIEW_REF   CHAR(9),
 CONSTRAINT     PKEY_VIEWS PRIMARY KEY (VIEWER, VIEW_REF)
);

CREATE TABLE BRANCH
(BID        CHAR(3)           CHECK (REGEXP_LIKE (BID, '^B\d{2}$'))
CONSTRAINT PKEY_BRANCH PRIMARY KEY,
 BADDRESS   CHAR(8)           NOT NULL,
 BPHONE           CHAR(11)    NOT NULL,
 BEMAIL           VARCHAR(50) NOT NULL,
 MANAGER    CHAR(5)           NOT NULL UNIQUE,
 EA_NAME    VARCHAR(30)
);

CREATE TABLE STAFF_MEMBER
(SID        CHAR(5)           CHECK (REGEXP_LIKE (SID, '^S\d{4}$'))
CONSTRAINT PKEY_STAFF PRIMARY KEY,
 SFNAME           VARCHAR(20) NOT NULL,
 SLNAME           VARCHAR(30) NOT NULL,
 SADDRESS   CHAR(8)           NOT NULL,
 SPHONE           CHAR(11)    NOT NULL,
 SEMAIL           VARCHAR(50),
 BID        CHAR(30)
);

CREATE TABLE CUSTOMER
(CID        CHAR(7)           CHECK (REGEXP_LIKE (CID, '^C\d{6}$'))
CONSTRAINT PKEY_CUSTOMER PRIMARY KEY,
 CFNAME           VARCHAR(20) NOT NULL,
 CLNAME           VARCHAR(30) NOT NULL,
 CADDRESS   CHAR(8),
 BDATE            DATE,
 GENDER           CHAR(1)           CONSTRAINT GENDER_RANGE CHECK (GENDER
IN ('M', 'F')),
 CPHONE           CHAR(11),
 CEMAIL           VARCHAR(50)
);

CREATE TABLE SALE
```

```
(SALE_ID    CHAR(7)           CHECK (REGEXP_LIKE (SALE_ID, '^SA\d{5}$'))
CONSTRAINT PKEY_SALE PRIMARY KEY,
 SALE_DATE  DATE        NOT NULL,
 SALE_PRICE NUMBER(10,2)      NOT NULL CONSTRAINT SP_RANGE CHECK
(SALE_PRICE > 0),
 STAMP_DUTY NUMBER(9,2) NOT NULL CONSTRAINT SD_RANGE CHECK (STAMP_DUTY >=
0),
 SALE_COMM  NUMBER(9,2) NOT NULL CONSTRAINT SC_RANGE CHECK (SALE_COMM >=
0),
 BUYER_ID   CHAR(7)           NOT NULL,
 PID        CHAR(7)           NOT NULL
);

CREATE TABLE LEASE
(LEASE_ID   CHAR(7)           CHECK (REGEXP_LIKE (LEASE_ID, '^LE\d{5}$'))
CONSTRAINT PKEY_LEASE PRIMARY KEY,
 START_DATE DATE        NOT NULL,
 END_DATE   DATE        NOT NULL,
 DEPOSIT    NUMBER(7,2) NOT NULL CONSTRAINT DEP_RANGE CHECK (DEPOSIT >= 0),
 LEASE_COMM NUMBER(9,2) NOT NULL CONSTRAINT LC_RANGE CHECK (LEASE_COMM >=
0),
 DPS        VARCHAR(50) NOT NULL,
 RENTER_ID  CHAR(7)           NOT NULL,
 PID        CHAR(7)           NOT NULL,
 CONSTRAINT VALID_TERM CHECK (START_DATE < END_DATE)
);

Rem   Add foreign keys below once data has been entered into each table.

/*
ALTER TABLE SALES_PROPERTY ADD CONSTRAINT FKEY_SP FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);

ALTER TABLE RENTAL_PROPERTY ADD CONSTRAINT FKEY_RP FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);

ALTER TABLE VIEWING ADD CONSTRAINT FKEY_VG FOREIGN KEY (PID) REFERENCES
PROPERTY (PID);

ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_ADD FOREIGN KEY (PADDRESS)
REFERENCES ADDRESS (ADD_NO);
ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_BRA FOREIGN KEY (BID) REFERENCES
BRANCH (BID);
ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_CUS FOREIGN KEY (OWNER_ID)
REFERENCES CUSTOMER (CID);

ALTER TABLE ESTATE_AGENT ADD CONSTRAINT FKEY_ADD2 FOREIGN KEY (EADDRESS)
REFERENCES ADDRESS (ADD_NO);

ALTER TABLE VIEWS ADD CONSTRAINT FKEY_V1 FOREIGN KEY (VIEWER) REFERENCES
CUSTOMER (CID);
ALTER TABLE VIEWS ADD CONSTRAINT FKEY_V2 FOREIGN KEY (VIEW_REF) REFERENCES
VIEWING (REF);

ALTER TABLE BRANCH ADD CONSTRAINT FKEY_ADD3 FOREIGN KEY (BADDRESS)
REFERENCES ADDRESS (ADD_NO);
ALTER TABLE BRANCH ADD CONSTRAINT FKEY_EA FOREIGN KEY (EA_NAME) REFERENCES
ESTATE_AGENT (NAME);
ALTER TABLE BRANCH ADD CONSTRAINT FKEY_MAN FOREIGN KEY (MANAGER) REFERENCES
STAFF_MEMBER (SID);
```

```
ALTER TABLE STAFF_MEMBER ADD CONSTRAINT FKEY_ADD4 FOREIGN KEY (SADDRESS)
REFERENCES ADDRESS (ADD_NO);
ALTER TABLE STAFF_MEMBER ADD CONSTRAINT FKEY_BRA2 FOREIGN KEY (BID)
REFERENCES BRANCH (BID);

ALTER TABLE CUSTOMER ADD CONSTRAINT FKEY_ADD5 FOREIGN KEY (CADDRESS)
REFERENCES ADDRESS (ADD_NO);

ALTER TABLE SALE ADD CONSTRAINT FKEY_BUYER FOREIGN KEY (BUYER_ID)
REFERENCES CUSTOMER (CID);
ALTER TABLE SALE ADD CONSTRAINT FKEY_SPROP FOREIGN KEY (PID) REFERENCES
PROPERTY (PID);

ALTER TABLE LEASE ADD CONSTRAINT FKEY_RENTER FOREIGN KEY (RENTER_ID)
REFERENCES CUSTOMER (CID);
ALTER TABLE LEASE ADD CONSTRAINT FKEY_RPROP FOREIGN KEY (PID) REFERENCES
PROPERTY (PID);
*/
```

Answer Part 1B: SQL DDL's Output (e.g., SPOOL file contents or screenshots): Minus 2 marks (if output is missing or deduct partial marks accordingly for partial or inadequate output)

```
W21056374 > @D:\pp_new_tabs.sql
W21056374 > SET VERIFY ON
W21056374 >
W21056374 > Rem     SQL script file used to create relations for the
PropertyPortal database.
W21056374 > Rem     Provides solution to PE7044 assessment part 1B for
student S CUMMING (21056374).
W21056374 >
W21056374 > CREATE TABLE ADDRESS
  2  (ADD_NO           CHAR(8)           CHECK (REGEXP_LIKE (ADD_NO,
'^AD\d{6}$')) CONSTRAINT PKEY_ADD PRIMARY KEY,
  3   LINE1      VARCHAR(30),
  4   LINE2      VARCHAR(30),
  5   LINE3      VARCHAR(30),
  6   CITY       VARCHAR(20),
  7   POSTCODE         VARCHAR(8)
  8  );

Table created.

W21056374 >
W21056374 > CREATE TABLE PROPERTY
  2  (PID        CHAR(7)           CHECK (REGEXP_LIKE (PID, '^P\d{6}$'))
CONSTRAINT PKEY_PROP PRIMARY KEY,
  3   PADDRESS         CHAR(8)           NOT NULL,
  4   DESCRIPTION    VARCHAR(500),
  5   TYPE       CHAR(1)           NOT NULL CONSTRAINT TYPE_RANGE CHECK
(TYPE IN ('D', 'S', 'T', 'F', 'B')),
  6   BEDROOMS         NUMBER(2)         NOT NULL CONSTRAINT BED_NO CHECK
(BEDROOMS > 0),
  7   GARDEN           CHAR(1)           NOT NULL CONSTRAINT IS_GARD CHECK
(GARDEN IN ('Y', 'N')),
  8   DATE_MARKETED  DATE          NOT NULL,
  9   OWNER_ID         CHAR(7)           NOT NULL,
 10   BID        CHAR(3)           NOT NULL
```

```
 11  );

Table created.

W21056374 >
W21056374 > CREATE TABLE SALES_PROPERTY
  2  (PID        CHAR(7)           CONSTRAINT PKEY_SALES_PROP PRIMARY KEY,
  3   ASKING_PRICE   NUMBER(10,2)    NOT NULL CONSTRAINT ASK_PRICE CHECK
(ASKING_PRICE > 0)
  4  );

Table created.

W21056374 >
W21056374 > CREATE TABLE RENTAL_PROPERTY
  2  (PID        CHAR(7)           CONSTRAINT PKEY_RENTAL_PROP PRIMARY KEY,
  3   RENTAL_PRICE   NUMBER(7,2)     NOT NULL CONSTRAINT RENT_PRICE CHECK
(RENTAL_PRICE > 0),
  4   FURNISHED_TYPE CHAR(2)       NOT NULL CONSTRAINT IS_FURNISHED CHECK
(FURNISHED_TYPE IN ('FF', 'PF', 'UF'))
  5  );

Table created.

W21056374 >
W21056374 > CREATE TABLE VIEWING
  2  (REF        CHAR(9)           CHECK (REGEXP_LIKE (REF, '^V\d{8}$'))
CONSTRAINT PKEY_VIEWING PRIMARY KEY,
  3   VIEW_DATE_TIME TIMESTAMP           NOT NULL,
  4   COMMENTS         VARCHAR(500),
  5   PID        CHAR(7)           NOT NULL
  6  );

Table created.

W21056374 >
W21056374 > CREATE TABLE ESTATE_AGENT
  2  (NAME       VARCHAR(30)     CONSTRAINT PKEY_ENAME PRIMARY KEY,
  3   HQ         VARCHAR(20)     NOT NULL,
  4   EADDRESS         CHAR(8)           NOT NULL,
  5   WEBSITE          VARCHAR(50)     NOT NULL,
  6   EPHONE           CHAR(11)          NOT NULL,
  7   EEMAIL           VARCHAR(50)     NOT NULL
  8  );

Table created.

W21056374 >
W21056374 > CREATE TABLE VIEWS
  2  (VIEWER           CHAR(7),
  3   VIEW_REF         CHAR(9),
  4   CONSTRAINT     PKEY_VIEWS PRIMARY KEY (VIEWER, VIEW_REF)
  5  );

Table created.

W21056374 >
W21056374 > CREATE TABLE BRANCH
  2  (BID        CHAR(3)           CHECK (REGEXP_LIKE (BID, '^B\d{2}$'))
CONSTRAINT PKEY_BRANCH PRIMARY KEY,
  3   BADDRESS         CHAR(8)           NOT NULL,
```

```
  4   BPHONE          CHAR(11)          NOT NULL,
  5   BEMAIL          VARCHAR(50)     NOT NULL,
  6   MANAGER         CHAR(5)           NOT NULL UNIQUE,
  7   EA_NAME         VARCHAR(30)
  8  );

Table created.

W21056374 >
W21056374 > CREATE TABLE STAFF_MEMBER
  2  (SID        CHAR(5)           CHECK (REGEXP_LIKE (SID, '^S\d{4}$'))
CONSTRAINT PKEY_STAFF PRIMARY KEY,
  3   SFNAME          VARCHAR(20)     NOT NULL,
  4   SLNAME          VARCHAR(30)     NOT NULL,
  5   SADDRESS        CHAR(8)           NOT NULL,
  6   SPHONE          CHAR(11)          NOT NULL,
  7   SEMAIL          VARCHAR(50),
  8   BID        CHAR(30)
  9  );

Table created.

W21056374 >
W21056374 > CREATE TABLE CUSTOMER
  2  (CID        CHAR(7)           CHECK (REGEXP_LIKE (CID, '^C\d{6}$'))
CONSTRAINT PKEY_CUSTOMER PRIMARY KEY,
  3   CFNAME          VARCHAR(20)     NOT NULL,
  4   CLNAME          VARCHAR(30)     NOT NULL,
  5   CADDRESS        CHAR(8),
  6   BDATE      DATE,
  7   GENDER          CHAR(1)           CONSTRAINT GENDER_RANGE CHECK
(GENDER IN ('M', 'F')),
  8   CPHONE          CHAR(11),
  9   CEMAIL          VARCHAR(50)
 10  );

Table created.

W21056374 >
W21056374 > CREATE TABLE SALE
  2  (SALE_ID          CHAR(7)           CHECK (REGEXP_LIKE (SALE_ID,
'^SA\d{5}$')) CONSTRAINT PKEY_SALE PRIMARY KEY,
  3   SALE_DATE      DATE          NOT NULL,
  4   SALE_PRICE     NUMBER(10,2)    NOT NULL CONSTRAINT SP_RANGE CHECK
(SALE_PRICE > 0),
  5   STAMP_DUTY     NUMBER(9,2)     NOT NULL CONSTRAINT SD_RANGE CHECK
(STAMP_DUTY >= 0),
  6   SALE_COMM      NUMBER(9,2)     NOT NULL CONSTRAINT SC_RANGE CHECK
(SALE_COMM >= 0),
  7   BUYER_ID          CHAR(7)           NOT NULL,
  8   PID        CHAR(7)           NOT NULL
  9  );

Table created.

W21056374 >
W21056374 > CREATE TABLE LEASE
  2  (LEASE_ID         CHAR(7)           CHECK (REGEXP_LIKE (LEASE_ID,
'^LE\d{5}$')) CONSTRAINT PKEY_LEASE PRIMARY KEY,
  3   START_DATE     DATE          NOT NULL,
  4   END_DATE          DATE        NOT NULL,
```

```
  5   DEPOSIT          NUMBER(7,2)     NOT NULL CONSTRAINT DEP_RANGE CHECK
(DEPOSIT >= 0),
  6   LEASE_COMM     NUMBER(9,2)     NOT NULL CONSTRAINT LC_RANGE CHECK
(LEASE_COMM >= 0),
  7   DPS        VARCHAR(50)     NOT NULL,
  8   RENTER_ID      CHAR(7)       NOT NULL,
  9   PID        CHAR(7)           NOT NULL,
 10   CONSTRAINT VALID_TERM CHECK (START_DATE < END_DATE)
 11  );

Table created.

W21056374 >
W21056374 > Rem     Add foreign keys below once data has been entered into
each table.
W21056374 >
W21056374 > /*
W21056374 > ALTER TABLE SALES_PROPERTY ADD CONSTRAINT FKEY_SP FOREIGN KEY
(PID) REFERENCES PROPERTY (PID);
W21056374 >
W21056374 > ALTER TABLE RENTAL_PROPERTY ADD CONSTRAINT FKEY_RP FOREIGN KEY
(PID) REFERENCES PROPERTY (PID);
W21056374 >
W21056374 > ALTER TABLE VIEWING ADD CONSTRAINT FKEY_VG FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);
W21056374 >
W21056374 > ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_ADD FOREIGN KEY
(PADDRESS) REFERENCES ADDRESS (ADD_NO);
W21056374 > ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_BRA FOREIGN KEY (BID)
REFERENCES BRANCH (BID);
W21056374 > ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_CUS FOREIGN KEY
(OWNER_ID) REFERENCES CUSTOMER (CID);
W21056374 >
W21056374 > ALTER TABLE ESTATE_AGENT ADD CONSTRAINT FKEY_ADD2 FOREIGN KEY
(EADDRESS) REFERENCES ADDRESS (ADD_NO);
W21056374 >
W21056374 > ALTER TABLE VIEWS ADD CONSTRAINT FKEY_V1 FOREIGN KEY (VIEWER)
REFERENCES CUSTOMER (CID);
W21056374 > ALTER TABLE VIEWS ADD CONSTRAINT FKEY_V2 FOREIGN KEY (VIEW_REF)
REFERENCES VIEWING (REF);
W21056374 >
W21056374 > ALTER TABLE BRANCH ADD CONSTRAINT FKEY_ADD3 FOREIGN KEY
(BADDRESS) REFERENCES ADDRESS (ADD_NO);
W21056374 > ALTER TABLE BRANCH ADD CONSTRAINT FKEY_EA FOREIGN KEY (EA_NAME)
REFERENCES ESTATE_AGENT (NAME);
W21056374 > ALTER TABLE BRANCH ADD CONSTRAINT FKEY_MAN FOREIGN KEY
(MANAGER) REFERENCES STAFF_MEMBER (SID);
W21056374 >
W21056374 > ALTER TABLE STAFF_MEMBER ADD CONSTRAINT FKEY_ADD4 FOREIGN KEY
(SADDRESS) REFERENCES ADDRESS (ADD_NO);
W21056374 > ALTER TABLE STAFF_MEMBER ADD CONSTRAINT FKEY_BRA2 FOREIGN KEY
(BID) REFERENCES BRANCH (BID);
W21056374 >
W21056374 > ALTER TABLE CUSTOMER ADD CONSTRAINT FKEY_ADD5 FOREIGN KEY
(CADDRESS) REFERENCES ADDRESS (ADD_NO);
W21056374 >
W21056374 > ALTER TABLE SALE ADD CONSTRAINT FKEY_BUYER FOREIGN KEY
(BUYER_ID) REFERENCES CUSTOMER (CID);
W21056374 > ALTER TABLE SALE ADD CONSTRAINT FKEY_SPROP FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);
W21056374 >
```

```
W21056374 > ALTER TABLE LEASE ADD CONSTRAINT FKEY_RENTER FOREIGN KEY
(RENTER_ID) REFERENCES CUSTOMER (CID);
W21056374 > ALTER TABLE LEASE ADD CONSTRAINT FKEY_RPROP FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);
W21056374 > */
W21056374 > spool off
```

Part 2 (20 marks)

(A) Populate the PropertyPortal database with some data (e.g., you should generate your own dummy data and load it into the PropertyPortal database, consider 5 to 8 rows for each table and enough data to see meaningful output for the queries below).

(8 marks)

Answer Part 2A: SQL code for populating your database: 8 marks

```
SET VERIFY ON
SET ECHO ON

REM Loading data into ADDRESS table.

INSERT INTO ADDRESS
  VALUES ('AD175002', 'Newcastle House', 'Albany Court', 'Newcastle
Business Park', 'Newcastle upon Tyne', 'NE4 7YB');

INSERT INTO ADDRESS
  VALUES ('AD336028', '566 Building One', 'Chiswick High Road', 'Chiswick
Park', 'London', 'W4 5BE');

INSERT INTO ADDRESS
  VALUES ('AD528406', 'Bishop Court', 'Front St', 'Whickham', 'Newcastle
upon Tyne', 'NE16 4JQ');

INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD125837', '49 Bunyan Road', 'Fenham', 'Newcastle upon Tyne',
'NE4 1SK');

INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD229016', '17 Fraser Drive', 'Fenham', 'Newcastle upon Tyne',
'NE4 2BB');

INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD853037', '12 Dunecht Road', 'Fenham', 'Newcastle upon Tyne',
'NE4 5HN');

INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD002385', 'The Willows', 'Grove Road', 'Gateshead', 'NE9 4KL');

INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD456732', '3 North Street', 'Heaton', 'Newcastle upon Tyne',
'NE6 9LT');

INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD123673', '88 Carpenters Croft', 'Heaton', 'Newcastle upon
Tyne', 'NE6 9FD');

INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD284145', 'Helix', 'St James Blvd', 'Newcastle upon Tyne', 'NE4
5BZ');
```

```
INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD010264', '5 - 9 Bon Accord Crescent', 'Aberdeen', 'AB11 6DN');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD693570', '33 Margaret Street', 'London', 'W1G 0JD');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD469321', '55 Baker Street', 'London', 'W1U 8AN');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD183994', '33 Union Street', 'Newcastle upon Tyne', 'NE1 7DN');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD809835', '30 Cloth Market', 'Newcastle upon Tyne', 'NE1 1EE');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD324873', '6 High Street', 'Newcastle upon Tyne', 'NE2 7SD');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD000184', '77 Station Road', 'Newcastle upon Tyne', 'NE6 8HH');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD117439', '45 Main Street', 'Newcastle upon Tyne', 'NE9 8DC');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD405337', '67 Church Road', 'Newcastle upon Tyne', 'NE5 3ED');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD811003', '345 Victoria Road', 'Newcastle upon Tyne', 'NE7
3EE');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD222235', '8 Church Lane', 'Newcastle upon Tyne', 'NE4 4NL');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD101018', '103 Mill Lane', 'Newcastle upon Tyne', 'NE8 1NN');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD388093', '23 York Road', 'Newcastle upon Tyne', 'NE2 4FA');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD999420', '29 Victoria Crescent', 'Newcastle upon Tyne', 'NE7
2XY');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD939346', '98 Windmill Road', 'Newcastle upon Tyne', 'NE8
3AS');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD913021', '15 Gray Road', 'Blyth', 'NE22 0FF');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD902784', '88 School Lane', 'Cramlington', 'NE23 3DD');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD090898', '33 Green Lane', 'Cramlington', 'NE23 0CM');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD059867', '8 Lancaster Road', 'Cramlington', 'NE23 2VT');
```

```
INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD111539', '212 King Street', 'Cramlington', 'NE23 1NJ');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD620439', '54 Castle Road', 'Cramlington', 'NE23 6SA');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD110293', '25 New Street', 'Gateshead', 'NE9 8HG');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD000005', '38 West Street', 'Gateshead', 'NE7 9BC');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD394765', '22 Stormont Road', 'Gateshead', 'NE8 9DN');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD950367', '66 Highfield Road', 'Sunderland', 'SR5 8DS');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD893274', '101 Albert Road', 'Sunderland', 'SR6 1TR');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD192832', '58 Mill Road', 'Sunderland', 'SR4 7JJ');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD109275', '59 Victory Lane', 'Sunderland', 'SR6 6LM');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD753947', '68 Park Street', 'Sunderland', 'SR6 1BC');

INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  VALUES ('AD539564', '2 George Street', 'Sunderland', 'SR7 9GG');


REM Loading data into CUSTOMER table.

INSERT INTO CUSTOMER
  VALUES ('C856290', 'Steffan', 'Schultz', 'AD950367', '17-APR-1954', 'M',
'07738973447', 's_schultz@hotmail.com');

INSERT INTO CUSTOMER
  VALUES ('C302849', 'Olivia', 'Martins', 'AD110293', '23-JUL-1980', 'F',
'07343982634', 'oliviam80@aol.com');

INSERT INTO CUSTOMER
  VALUES ('C552901', 'Mike', 'Wilson', 'AD000005', '03-MAY-1973', 'M',
'07704751623', 'mike_wilson@protonmail.com');

INSERT INTO CUSTOMER
  VALUES ('C000032', 'Carla', 'Barton', 'AD893274', '08-OCT-1978', 'F',
'07273859673', 'carla1978@gmail.com');

INSERT INTO CUSTOMER
  VALUES ('C174439', 'George', 'Davis', 'AD902784', '11-OCT-1990', 'M',
'07112048573', 'gdavis@bing.com');

INSERT INTO CUSTOMER
  VALUES ('C986032', 'Harry', 'Miller', 'AD192832', '05-NOV-1982', 'M',
'07795937475', 'harry_miller@aol.com');

INSERT INTO CUSTOMER
```

```
  VALUES ('C678201', 'Charlotte', 'Reid', 'AD090898', '23-OCT-1960', 'F',
'07807325212', 'charlotte1960@hotmail.com');

INSERT INTO CUSTOMER
  VALUES ('C345750', 'Ava', 'Robertson', 'AD125837', '02-AUG-1954', 'F',
'07734630932', 'arobertson@gmail.com');

INSERT INTO CUSTOMER
  VALUES ('C220847', 'Isabella', 'Lopez', 'AD939346', '15-JUL-1965', 'F',
'07763444412', 'lopez_i@bing.com');

INSERT INTO CUSTOMER
  VALUES ('C740283', 'Jack', 'Dunn', 'AD456732', '25-JAN-1978', 'M',
'07653094618', 'jackydunn@gmail.com');

INSERT INTO CUSTOMER
  VALUES ('C110834', 'Mia', 'Beech', 'AD059867', '27-FEB-1988', 'F',
'07778395733', 'mia_beech@protonmail.com');

INSERT INTO CUSTOMER
  VALUES ('C985422', 'Sophia', 'Walker', 'AD109275', '18-SEP-1998', 'F',
'07453627345', 'sophiawalker@hotmail.com');

INSERT INTO CUSTOMER
  VALUES ('C002398', 'Elijah', 'Wood', 'AD394765', '10-DEC-1950', 'M',
'07720173845', 'e_wood@hotmail.com');

INSERT INTO CUSTOMER
  VALUES ('C309496', 'Reuben', 'Diaz', 'AD999420', '06-JUN-1972', 'M',
'07699012364', 'reuben_diaz72@gmail.com');


REM Loading data into ESTATE_AGENT table.

INSERT INTO ESTATE_AGENT
  VALUES ('Your Move', 'Newcastle', 'AD175002', 'www.your-move.co.uk',
'01912676374', 'headoffice@your-move.co.uk');

INSERT INTO ESTATE_AGENT
  VALUES ('Savills', 'London', 'AD693570', 'www.savills.co.uk',
'02074998644', 'enquiries@savills.com');

INSERT INTO ESTATE_AGENT
  VALUES ('Foxtons', 'London', 'AD336028', 'www.foxtons.co.uk',
'08003698667', 'hq@foxtons.co.uk');

INSERT INTO ESTATE_AGENT
  VALUES ('Knight Frank', 'London', 'AD469321', 'www.knightfrank.co.uk',
'02039445824', 'admin@knightfrank.com');

INSERT INTO ESTATE_AGENT
  VALUES ('Aberdein Considine', 'Aberdeen', 'AD010264', 'www.acandco.com',
'01224589700', 'ask@acandco.com');


REM Loading data into BRANCH table.

INSERT INTO BRANCH
  VALUES ('B01', 'AD528406', '01914887968', 'whickham@your-move.co.uk',
'S9921', 'Your Move');
```

```
INSERT INTO BRANCH
  VALUES ('B02', 'AD284145', '01919171444', 'newcastle@savills.com',
'S5790', 'Savills');

INSERT INTO BRANCH
  VALUES ('B03', 'AD183994', '01915382636', 'newc@foxtons.co.uk', 'S5693',
'Foxtons');

INSERT INTO BRANCH
  VALUES ('B04', 'AD809835', '01916078450', 'newcastle@acandco.com',
'S0137', 'Aberdein Considine');


REM Loading data into STAFF_MEMBER table.

INSERT INTO STAFF_MEMBER
  VALUES ('S6294', 'John', 'Smith', 'AD324873', '07737824938',
'jsmith@hotmail.com', 'B01');

INSERT INTO STAFF_MEMBER
  VALUES ('S0137', 'Sarah', 'Bloggs', 'AD000184', '07364925844',
'sarahb@aol.com', 'B04');

INSERT INTO STAFF_MEMBER
  VALUES ('S8835', 'Rachel', 'Santos', 'AD117439', '07123758362',
'rachel_santos@bing.com', 'B03');

INSERT INTO STAFF_MEMBER
  VALUES ('S5693', 'Paul', 'Laporte', 'AD405337', '07788243944',
'plaporte1980@hotmail.com', 'B03');

INSERT INTO STAFF_MEMBER
  VALUES ('S0937', 'Trent', 'Davidson', 'AD811003', '07832013885',
'trent568@gmail.com', 'B04');

INSERT INTO STAFF_MEMBER
  VALUES ('S5790', 'Claire', 'Beattie', 'AD222235', '07437293172',
'beattie_girl@gmail.com', 'B02');

INSERT INTO STAFF_MEMBER
  VALUES ('S4529', 'Emma', 'Jones', 'AD101018', '07234844921',
'ejones53@aol.com', 'B02');

INSERT INTO STAFF_MEMBER
  VALUES ('S9921', 'Kareem', 'Jabbar', 'AD388093', '07732831932',
'kj21@protonmail.com', 'B01');


REM Loading data into PROPERTY table.

INSERT INTO PROPERTY
  VALUES ('P349879', 'AD229016', 'This exceptional home has been created
with great imagination and finished to a high specification.', 'D', 5, 'Y',
'01-JUN-2022', 'C552901', 'B01');

INSERT INTO PROPERTY
  VALUES ('P018234', 'AD123673', 'A spacious extended four bedroom
Victorian terraced home, with a loft conversion, south-west facing private
enclosed garden and detached garage with power.', 'S', 4, 'Y', '27-MAY-
2022', 'C174439', 'B01');
```

```
INSERT INTO PROPERTY
  VALUES ('P832748', 'AD853037', 'This spacious and well-presented three
bedroom semi-detached family home comes to market offering generous living
space throughout.', 'S', 3, 'Y', '08-SEP-2021', 'C986032', 'B04');

INSERT INTO PROPERTY
  VALUES ('P247587', 'AD913021', 'The property benefits from a lovely wrap
around garden, off-road parking, and a large attached garage.', 'B', 3,
'Y', '04-AUG-2019', 'C000032', 'B03');

INSERT INTO PROPERTY
  VALUES ('P347853', 'AD753947', 'Beautifully presented two bedroom first
floor apartment town centre location, two bedrooms master with en suite and
built in wardrobe, modern bright and airy open plan living room with
balcony, kitchen with appliances and granite worktops, main bathroom, entry
phone system, lifts to all floors, allocated underground parking.', 'F', 2,
'N', '09-NOV-2021', 'C856290', 'B04');

INSERT INTO PROPERTY
  VALUES ('P984752', 'AD539564', 'The conclusion of a short private road,
this 4 bedroom detached modern build sits back from the High Street.', 'D',
4, 'Y', '28-APR-2022', 'C000032', 'B02');

INSERT INTO PROPERTY
  VALUES ('P223743', 'AD002385', 'The property boasts laminate flooring
throughout, a separate bathroom semi open plan kitchen and reception with a
single bedroom.', 'F', 1, 'N', '05-MAY-2022', 'C856290', 'B01');

INSERT INTO PROPERTY
  VALUES ('P118364', 'AD111539', 'A beautifully stylish two bedroom first
floor apartment in this prestigious and modern development, with extremely
spacious accommodation, modern kitchen, and benefiting from one parking
space.', 'F', 2, 'N', '05-AUG-2020', 'C856290', 'B02');

INSERT INTO PROPERTY
  VALUES ('P658439', 'AD620439', 'The house is arranged over two floors. To
the ground floor there are two spacious reception rooms, a modern fitted
kitchen and downstairs shower room.', 'T', 2, 'Y', '13-FEB-2018',
'C302849', 'B01');

INSERT INTO PROPERTY
  VALUES ('P102934', 'AD110293', 'A beautifully presented home in an ideal
position for access to local schooling and the railway station. Fabulous
garden, home office and off-road parking.', 'D', 3, 'Y', '17-JUN-2017',
'C740283', 'B02');

INSERT INTO PROPERTY
  VALUES ('P298473', 'AD902784', 'A substantial detached chalet bungalow
providing spacious and adaptable accommodation. This beautifully presented
individual home of character has been sympathetically extended and
refurbished to high specification now providing a comfortable home ready to
move into with no upward chain.', 'B', 4, 'Y', '11-OCT-2021', 'C985422',
'B01');


REM Loading data into SALES_PROPERTY table.

INSERT INTO SALES_PROPERTY
  VALUES ('P349879', 450000);

INSERT INTO SALES_PROPERTY
```

```
  VALUES ('P018234', 375000);

INSERT INTO SALES_PROPERTY
  VALUES ('P832748', 295500);

INSERT INTO SALES_PROPERTY
  VALUES ('P247587', 220000);

INSERT INTO SALES_PROPERTY
  VALUES ('P984752', 350000);

INSERT INTO SALES_PROPERTY
  VALUES ('P658439', 185000);

INSERT INTO SALES_PROPERTY
  VALUES ('P102934', 275000);

INSERT INTO SALES_PROPERTY
  VALUES ('P298473', 305000);


REM Loading data into RENTAL_PROPERTY table.

INSERT INTO RENTAL_PROPERTY
  VALUES ('P347853', 700, 'UF');

INSERT INTO RENTAL_PROPERTY
  VALUES ('P223743', 450, 'UF');

INSERT INTO RENTAL_PROPERTY
  VALUES ('P118364', 675, 'FF');


REM Loading data into SALE table.

INSERT INTO SALE
  VALUES ('SA46734', '20-DEC-2021', 305000, 0, 305, 'C220847', 'P832748');

INSERT INTO SALE
  VALUES ('SA58673', '05-NOV-2019', 230000, 4600, 230, 'C740283',
'P247587');

INSERT INTO SALE
  VALUES ('SA19384', '18-JUN-2018', 208500, 4170, 209, 'C985422',
'P658439');

INSERT INTO SALE
  VALUES ('SA98358', '27-SEP-2017', 278000, 2780, 278, 'C302849',
'P102934');

INSERT INTO SALE
  VALUES ('SA23148', '25-JAN-2022', 310000, 3100, 310, 'C174439',
'P298473');


REM Loading data into LEASE table.

INSERT INTO LEASE
  VALUES ('LE38563', '28-JAN-2022', '27-JAN-2023', 700, 70, 'My Deposits',
'C002398', 'P347853');
```

```
INSERT INTO LEASE
  VALUES ('LE85773', '10-NOV-2020', '09-NOV-2022', 675, 67.50, 'Tenancy
Deposit Scheme', 'C309496', 'P118364');


REM Loading data into VIEWING table.

INSERT INTO VIEWING
  VALUES ('V10273477', TO_TIMESTAMP('02-JUN-2022 18:00:00', 'DD-MON-YYYY
HH24:MI:SS'), 'Not interested.', 'P349879');

INSERT INTO VIEWING
  VALUES ('V39827456', TO_TIMESTAMP('29-MAY-2022 14:05:00', 'DD-MON-YYYY
HH24:MI:SS'), 'Want to return with partner to look around.', 'P018234');

INSERT INTO VIEWING
  VALUES ('V44824307', TO_TIMESTAMP('20-SEP-2021 10:30:00', 'DD-MON-YYYY
HH24:MI:SS'), 'Very keen.', 'P832748');

INSERT INTO VIEWING
  VALUES ('V34264472', TO_TIMESTAMP('19-AUG-2019 12:15:00', 'DD-MON-YYYY
HH24:MI:SS'), 'Stated will submit an offer.', 'P247587');

INSERT INTO VIEWING
  VALUES ('V73826125', TO_TIMESTAMP('05-MAY-2022 15:00:00', 'DD-MON-YYYY
HH24:MI:SS'), 'Not quite what they are looking for.', 'P984752');


REM Loading data into VIEWS table.

INSERT INTO VIEWS
  VALUES ('C678201', 'V10273477');

INSERT INTO VIEWS
  VALUES ('C345750', 'V39827456');

INSERT INTO VIEWS
  VALUES ('C220847', 'V44824307');

INSERT INTO VIEWS
  VALUES ('C740283', 'V34264472');

INSERT INTO VIEWS
  VALUES ('C110834', 'V73826125');


COMMIT;

ALTER TABLE SALES_PROPERTY ADD CONSTRAINT FKEY_SP FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);

ALTER TABLE RENTAL_PROPERTY ADD CONSTRAINT FKEY_RP FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);

ALTER TABLE VIEWING ADD CONSTRAINT FKEY_VG FOREIGN KEY (PID) REFERENCES
PROPERTY (PID);

ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_ADD FOREIGN KEY (PADDRESS)
REFERENCES ADDRESS (ADD_NO);
ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_BRA FOREIGN KEY (BID) REFERENCES
BRANCH (BID);
```

```
ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_CUS FOREIGN KEY (OWNER_ID)
REFERENCES CUSTOMER (CID);

ALTER TABLE ESTATE_AGENT ADD CONSTRAINT FKEY_ADD2 FOREIGN KEY (EADDRESS)
REFERENCES ADDRESS (ADD_NO);

ALTER TABLE VIEWS ADD CONSTRAINT FKEY_V1 FOREIGN KEY (VIEWER) REFERENCES
CUSTOMER (CID);
ALTER TABLE VIEWS ADD CONSTRAINT FKEY_V2 FOREIGN KEY (VIEW_REF) REFERENCES
VIEWING (REF);

ALTER TABLE BRANCH ADD CONSTRAINT FKEY_ADD3 FOREIGN KEY (BADDRESS)
REFERENCES ADDRESS (ADD_NO);
ALTER TABLE BRANCH ADD CONSTRAINT FKEY_EA FOREIGN KEY (EA_NAME) REFERENCES
ESTATE_AGENT (NAME);
ALTER TABLE BRANCH ADD CONSTRAINT FKEY_MAN FOREIGN KEY (MANAGER) REFERENCES
STAFF_MEMBER (SID);

ALTER TABLE STAFF_MEMBER ADD CONSTRAINT FKEY_ADD4 FOREIGN KEY (SADDRESS)
REFERENCES ADDRESS (ADD_NO);
ALTER TABLE STAFF_MEMBER ADD CONSTRAINT FKEY_BRA2 FOREIGN KEY (BID)
REFERENCES BRANCH (BID);

ALTER TABLE CUSTOMER ADD CONSTRAINT FKEY_ADD5 FOREIGN KEY (CADDRESS)
REFERENCES ADDRESS (ADD_NO);

ALTER TABLE SALE ADD CONSTRAINT FKEY_BUYER FOREIGN KEY (BUYER_ID)
REFERENCES CUSTOMER (CID);
ALTER TABLE SALE ADD CONSTRAINT FKEY_SPROP FOREIGN KEY (PID) REFERENCES
PROPERTY (PID);

ALTER TABLE LEASE ADD CONSTRAINT FKEY_RENTER FOREIGN KEY (RENTER_ID)
REFERENCES CUSTOMER (CID);
ALTER TABLE LEASE ADD CONSTRAINT FKEY_RPROP FOREIGN KEY (PID) REFERENCES
PROPERTY (PID);

SET ECHO OFF
```

Answer Part 2 A: SQL code's output (e.g., SPOOL file contents or screenshots): Minus 2 marks (if output is missing or deduct partial marks accordingly for partial or inadequate output)

```
W21056374 > @D:\pp_data_inserts.sql
W21056374 > SET VERIFY ON
W21056374 > SET ECHO ON
W21056374 >
W21056374 > REM Loading data into ADDRESS table.
W21056374 >
W21056374 > INSERT INTO ADDRESS
  2    VALUES ('AD175002', 'Newcastle House', 'Albany Court', 'Newcastle
Business Park', 'Newcastle upon Tyne', 'NE4 7YB');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS
```

```
  2    VALUES ('AD336028', '566 Building One', 'Chiswick High Road',
'Chiswick Park', 'London', 'W4 5BE');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS
  2    VALUES ('AD528406', 'Bishop Court', 'Front St', 'Whickham',
'Newcastle upon Tyne', 'NE16 4JQ');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD125837', '49 Bunyan Road', 'Fenham', 'Newcastle upon
Tyne', 'NE4 1SK');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD229016', '17 Fraser Drive', 'Fenham', 'Newcastle upon
Tyne', 'NE4 2BB');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD853037', '12 Dunecht Road', 'Fenham', 'Newcastle upon
Tyne', 'NE4 5HN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD002385', 'The Willows', 'Grove Road', 'Gateshead', 'NE9
4KL');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD456732', '3 North Street', 'Heaton', 'Newcastle upon
Tyne', 'NE6 9LT');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD123673', '88 Carpenters Croft', 'Heaton', 'Newcastle upon
Tyne', 'NE6 9FD');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD284145', 'Helix', 'St James Blvd', 'Newcastle upon Tyne',
'NE4 5BZ');

1 row created.
```

```
W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD010264', '5 - 9 Bon Accord Crescent', 'Aberdeen', 'AB11
6DN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD693570', '33 Margaret Street', 'London', 'W1G 0JD');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD469321', '55 Baker Street', 'London', 'W1U 8AN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD183994', '33 Union Street', 'Newcastle upon Tyne', 'NE1
7DN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD809835', '30 Cloth Market', 'Newcastle upon Tyne', 'NE1
1EE');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD324873', '6 High Street', 'Newcastle upon Tyne', 'NE2
7SD');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD000184', '77 Station Road', 'Newcastle upon Tyne', 'NE6
8HH');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD117439', '45 Main Street', 'Newcastle upon Tyne', 'NE9
8DC');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD405337', '67 Church Road', 'Newcastle upon Tyne', 'NE5
3ED');

1 row created.
```

```
W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD811003', '345 Victoria Road', 'Newcastle upon Tyne', 'NE7
3EE');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD222235', '8 Church Lane', 'Newcastle upon Tyne', 'NE4
4NL');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD101018', '103 Mill Lane', 'Newcastle upon Tyne', 'NE8
1NN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD388093', '23 York Road', 'Newcastle upon Tyne', 'NE2
4FA');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD999420', '29 Victoria Crescent', 'Newcastle upon Tyne',
'NE7 2XY');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD939346', '98 Windmill Road', 'Newcastle upon Tyne', 'NE8
3AS');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD913021', '15 Gray Road', 'Blyth', 'NE22 0FF');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD902784', '88 School Lane', 'Cramlington', 'NE23 3DD');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD090898', '33 Green Lane', 'Cramlington', 'NE23 0CM');

1 row created.

W21056374 >
```

```
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD059867', '8 Lancaster Road', 'Cramlington', 'NE23 2VT');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD111539', '212 King Street', 'Cramlington', 'NE23 1NJ');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD620439', '54 Castle Road', 'Cramlington', 'NE23 6SA');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD110293', '25 New Street', 'Gateshead', 'NE9 8HG');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD000005', '38 West Street', 'Gateshead', 'NE7 9BC');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD394765', '22 Stormont Road', 'Gateshead', 'NE8 9DN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD950367', '66 Highfield Road', 'Sunderland', 'SR5 8DS');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD893274', '101 Albert Road', 'Sunderland', 'SR6 1TR');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD192832', '58 Mill Road', 'Sunderland', 'SR4 7JJ');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD109275', '59 Victory Lane', 'Sunderland', 'SR6 6LM');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
```

```
  2    VALUES ('AD753947', '68 Park Street', 'Sunderland', 'SR6 1BC');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS (Add_No, Line1, City, Postcode)
  2    VALUES ('AD539564', '2 George Street', 'Sunderland', 'SR7 9GG');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into CUSTOMER table.
W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C856290', 'Steffan', 'Schultz', 'AD950367', '17-APR-1954',
'M', '07738973447', 's_schultz@hotmail.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C302849', 'Olivia', 'Martins', 'AD110293', '23-JUL-1980',
'F', '07343982634', 'oliviam80@aol.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C552901', 'Mike', 'Wilson', 'AD000005', '03-MAY-1973', 'M',
'07704751623', 'mike_wilson@protonmail.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C000032', 'Carla', 'Barton', 'AD893274', '08-OCT-1978',
'F', '07273859673', 'carla1978@gmail.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C174439', 'George', 'Davis', 'AD902784', '11-OCT-1990',
'M', '07112048573', 'gdavis@bing.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C986032', 'Harry', 'Miller', 'AD192832', '05-NOV-1982',
'M', '07795937475', 'harry_miller@aol.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C678201', 'Charlotte', 'Reid', 'AD090898', '23-OCT-1960',
'F', '07807325212', 'charlotte1960@hotmail.com');

1 row created.
```

```
W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C345750', 'Ava', 'Robertson', 'AD125837', '02-AUG-1954',
'F', '07734630932', 'arobertson@gmail.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C220847', 'Isabella', 'Lopez', 'AD939346', '15-JUL-1965',
'F', '07763444412', 'lopez_i@bing.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C740283', 'Jack', 'Dunn', 'AD456732', '25-JAN-1978', 'M',
'07653094618', 'jackydunn@gmail.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C110834', 'Mia', 'Beech', 'AD059867', '27-FEB-1988', 'F',
'07778395733', 'mia_beech@protonmail.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C985422', 'Sophia', 'Walker', 'AD109275', '18-SEP-1998',
'F', '07453627345', 'sophiawalker@hotmail.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C002398', 'Elijah', 'Wood', 'AD394765', '10-DEC-1950', 'M',
'07720173845', 'e_wood@hotmail.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO CUSTOMER
  2    VALUES ('C309496', 'Reuben', 'Diaz', 'AD999420', '06-JUN-1972', 'M',
'07699012364', 'reuben_diaz72@gmail.com');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into ESTATE_AGENT table.
W21056374 >
W21056374 > INSERT INTO ESTATE_AGENT
  2    VALUES ('Your Move', 'Newcastle', 'AD175002', 'www.your-move.co.uk',
'01912676374', 'headoffice@your-move.co.uk');

1 row created.

W21056374 >
```

```
W21056374 > INSERT INTO ESTATE_AGENT
  2    VALUES ('Savills', 'London', 'AD693570', 'www.savills.co.uk',
'02074998644', 'enquiries@savills.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO ESTATE_AGENT
  2    VALUES ('Foxtons', 'London', 'AD336028', 'www.foxtons.co.uk',
'08003698667', 'hq@foxtons.co.uk');

1 row created.

W21056374 >
W21056374 > INSERT INTO ESTATE_AGENT
  2    VALUES ('Knight Frank', 'London', 'AD469321',
'www.knightfrank.co.uk', '02039445824', 'admin@knightfrank.com');

1 row created.

W21056374 >
W21056374 > INSERT INTO ESTATE_AGENT
  2    VALUES ('Aberdein Considine', 'Aberdeen', 'AD010264',
'www.acandco.com', '01224589700', 'ask@acandco.com');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into BRANCH table.
W21056374 >
W21056374 > INSERT INTO BRANCH
  2    VALUES ('B01', 'AD528406', '01914887968', 'whickham@your-
move.co.uk', 'S9921', 'Your Move');

1 row created.

W21056374 >
W21056374 > INSERT INTO BRANCH
  2    VALUES ('B02', 'AD284145', '01919171444', 'newcastle@savills.com',
'S5790', 'Savills');

1 row created.

W21056374 >
W21056374 > INSERT INTO BRANCH
  2    VALUES ('B03', 'AD183994', '01915382636', 'newc@foxtons.co.uk',
'S5693', 'Foxtons');

1 row created.

W21056374 >
W21056374 > INSERT INTO BRANCH
  2    VALUES ('B04', 'AD809835', '01916078450', 'newcastle@acandco.com',
'S0137', 'Aberdein Considine');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into STAFF_MEMBER table.
```

```
W21056374 >
W21056374 > INSERT INTO STAFF_MEMBER
  2    VALUES ('S6294', 'John', 'Smith', 'AD324873', '07737824938',
'jsmith@hotmail.com', 'B01');

1 row created.

W21056374 >
W21056374 > INSERT INTO STAFF_MEMBER
  2    VALUES ('S0137', 'Sarah', 'Bloggs', 'AD000184', '07364925844',
'sarahb@aol.com', 'B04');

1 row created.

W21056374 >
W21056374 > INSERT INTO STAFF_MEMBER
  2    VALUES ('S8835', 'Rachel', 'Santos', 'AD117439', '07123758362',
'rachel_santos@bing.com', 'B03');

1 row created.

W21056374 >
W21056374 > INSERT INTO STAFF_MEMBER
  2    VALUES ('S5693', 'Paul', 'Laporte', 'AD405337', '07788243944',
'plaporte1980@hotmail.com', 'B03');

1 row created.

W21056374 >
W21056374 > INSERT INTO STAFF_MEMBER
  2    VALUES ('S0937', 'Trent', 'Davidson', 'AD811003', '07832013885',
'trent568@gmail.com', 'B04');

1 row created.

W21056374 >
W21056374 > INSERT INTO STAFF_MEMBER
  2    VALUES ('S5790', 'Claire', 'Beattie', 'AD222235', '07437293172',
'beattie_girl@gmail.com', 'B02');

1 row created.

W21056374 >
W21056374 > INSERT INTO STAFF_MEMBER
  2    VALUES ('S4529', 'Emma', 'Jones', 'AD101018', '07234844921',
'ejones53@aol.com', 'B02');

1 row created.

W21056374 >
W21056374 > INSERT INTO STAFF_MEMBER
  2    VALUES ('S9921', 'Kareem', 'Jabbar', 'AD388093', '07732831932',
'kj21@protonmail.com', 'B01');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into PROPERTY table.
W21056374 >
W21056374 > INSERT INTO PROPERTY
```

```
  2    VALUES ('P349879', 'AD229016', 'This exceptional home has been
created with great imagination and finished to a high specification.', 'D',
5, 'Y', '01-JUN-2022', 'C552901', 'B01');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P018234', 'AD123673', 'A spacious extended four bedroom
Victorian terraced home, with a loft conversion, south-west facing private
enclosed garden and detached garage with power.', 'S', 4, 'Y', '27-MAY-
2022', 'C174439', 'B01');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P832748', 'AD853037', 'This spacious and well-presented
three bedroom semi-detached family home comes to market offering generous
living space throughout.', 'S', 3, 'Y', '08-SEP-2021', 'C986032', 'B04');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P247587', 'AD913021', 'The property benefits from a lovely
wrap around garden, off-road parking, and a large attached garage.', 'B',
3, 'Y', '04-AUG-2019', 'C000032', 'B03');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P347853', 'AD753947', 'Beautifully presented two bedroom
first floor apartment town centre location, two bedrooms master with en
suite and built in wardrobe, modern bright and airy open plan living room
with balcony, kitchen with appliances and granite worktops, main bathroom,
entry phone system, lifts to all floors, allocated underground parking.',
'F', 2, 'N', '09-NOV-2021', 'C856290', 'B04');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P984752', 'AD539564', 'The conclusion of a short private
road, this 4 bedroom detached modern build sits back from the High
Street.', 'D', 4, 'Y', '28-APR-2022', 'C000032', 'B02');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P223743', 'AD002385', 'The property boasts laminate
flooring throughout, a separate bathroom semi open plan kitchen and
reception with a single bedroom.', 'F', 1, 'N', '05-MAY-2022', 'C856290',
'B01');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
```

```
  2    VALUES ('P118364', 'AD111539', 'A beautifully stylish two bedroom
first floor apartment in this prestigious and modern development, with
extremely spacious accommodation, modern kitchen, and benefiting from one
parking space.', 'F', 2, 'N', '05-AUG-2020', 'C856290', 'B02');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P658439', 'AD620439', 'The house is arranged over two
floors. To the ground floor there are two spacious reception rooms, a
modern fitted kitchen and downstairs shower room.', 'T', 2, 'Y', '13-FEB-
2018', 'C302849', 'B01');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P102934', 'AD110293', 'A beautifully presented home in an
ideal position for access to local schooling and the railway station.
Fabulous garden, home office and off-road parking.', 'D', 3, 'Y', '17-JUN-
2017', 'C740283', 'B02');

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY
  2    VALUES ('P298473', 'AD902784', 'A substantial detached chalet
bungalow providing spacious and adaptable accommodation. This beautifully
presented individual home of character has been sympathetically extended
and refurbished to high specification now providing a comfortable home
ready to move into with no upward chain.', 'B', 4, 'Y', '11-OCT-2021',
'C985422', 'B01');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into SALES_PROPERTY table.
W21056374 >
W21056374 > INSERT INTO SALES_PROPERTY
  2    VALUES ('P349879', 450000);

1 row created.

W21056374 >
W21056374 > INSERT INTO SALES_PROPERTY
  2    VALUES ('P018234', 375000);

1 row created.

W21056374 >
W21056374 > INSERT INTO SALES_PROPERTY
  2    VALUES ('P832748', 295500);

1 row created.

W21056374 >
W21056374 > INSERT INTO SALES_PROPERTY
  2    VALUES ('P247587', 220000);
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO SALES_PROPERTY
  2    VALUES ('P984752', 350000);

1 row created.

W21056374 >
W21056374 > INSERT INTO SALES_PROPERTY
  2    VALUES ('P658439', 185000);

1 row created.

W21056374 >
W21056374 > INSERT INTO SALES_PROPERTY
  2    VALUES ('P102934', 275000);

1 row created.

W21056374 >
W21056374 > INSERT INTO SALES_PROPERTY
  2    VALUES ('P298473', 305000);

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into RENTAL_PROPERTY table.
W21056374 >
W21056374 > INSERT INTO RENTAL_PROPERTY
  2    VALUES ('P347853', 700, 'UF');

1 row created.

W21056374 >
W21056374 > INSERT INTO RENTAL_PROPERTY
  2    VALUES ('P223743', 450, 'UF');

1 row created.

W21056374 >
W21056374 > INSERT INTO RENTAL_PROPERTY
  2    VALUES ('P118364', 675, 'FF');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into SALE table.
W21056374 >
W21056374 > INSERT INTO SALE
  2    VALUES ('SA46734', '20-DEC-2021', 305000, 0, 305, 'C220847',
'P832748');

1 row created.

W21056374 >
W21056374 > INSERT INTO SALE
  2    VALUES ('SA58673', '05-NOV-2019', 230000, 4600, 230, 'C740283',
'P247587');
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO SALE
  2    VALUES ('SA19384', '18-JUN-2018', 208500, 4170, 209, 'C985422',
'P658439');

1 row created.

W21056374 >
W21056374 > INSERT INTO SALE
  2    VALUES ('SA98358', '27-SEP-2017', 278000, 2780, 278, 'C302849',
'P102934');

1 row created.

W21056374 >
W21056374 > INSERT INTO SALE
  2    VALUES ('SA23148', '25-JAN-2022', 310000, 3100, 310, 'C174439',
'P298473');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into LEASE table.
W21056374 >
W21056374 > INSERT INTO LEASE
  2    VALUES ('LE38563', '28-JAN-2022', '27-JAN-2023', 700, 70, 'My
Deposits', 'C002398', 'P347853');

1 row created.

W21056374 >
W21056374 > INSERT INTO LEASE
  2    VALUES ('LE85773', '10-NOV-2020', '09-NOV-2022', 675, 67.50,
'Tenancy Deposit Scheme', 'C309496', 'P118364');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into VIEWING table.
W21056374 >
W21056374 > INSERT INTO VIEWING
  2    VALUES ('V10273477', TO_TIMESTAMP('02-JUN-2022 18:00:00', 'DD-MON-
YYYY HH24:MI:SS'), 'Not interested.', 'P349879');

1 row created.

W21056374 >
W21056374 > INSERT INTO VIEWING
  2    VALUES ('V39827456', TO_TIMESTAMP('29-MAY-2022 14:05:00', 'DD-MON-
YYYY HH24:MI:SS'), 'Want to return with partner to look around.',
'P018234');

1 row created.

W21056374 >
W21056374 > INSERT INTO VIEWING
```

```
  2    VALUES ('V44824307', TO_TIMESTAMP('20-SEP-2021 10:30:00', 'DD-MON-
YYYY HH24:MI:SS'), 'Very keen.', 'P832748');

1 row created.

W21056374 >
W21056374 > INSERT INTO VIEWING
  2    VALUES ('V34264472', TO_TIMESTAMP('19-AUG-2019 12:15:00', 'DD-MON-
YYYY HH24:MI:SS'), 'Stated will submit an offer.', 'P247587');

1 row created.

W21056374 >
W21056374 > INSERT INTO VIEWING
  2    VALUES ('V73826125', TO_TIMESTAMP('05-MAY-2022 15:00:00', 'DD-MON-
YYYY HH24:MI:SS'), 'Not quite what they are looking for.', 'P984752');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into VIEWS table.
W21056374 >
W21056374 > INSERT INTO VIEWS
  2    VALUES ('C678201', 'V10273477');

1 row created.

W21056374 >
W21056374 > INSERT INTO VIEWS
  2    VALUES ('C345750', 'V39827456');

1 row created.

W21056374 >
W21056374 > INSERT INTO VIEWS
  2    VALUES ('C220847', 'V44824307');

1 row created.

W21056374 >
W21056374 > INSERT INTO VIEWS
  2    VALUES ('C740283', 'V34264472');

1 row created.

W21056374 >
W21056374 > INSERT INTO VIEWS
  2    VALUES ('C110834', 'V73826125');

1 row created.

W21056374 >
W21056374 >
W21056374 > COMMIT;

Commit complete.

W21056374 >
W21056374 > ALTER TABLE SALES_PROPERTY ADD CONSTRAINT FKEY_SP FOREIGN KEY
(PID) REFERENCES PROPERTY (PID);
```

```
Table altered.

W21056374 >
W21056374 > ALTER TABLE RENTAL_PROPERTY ADD CONSTRAINT FKEY_RP FOREIGN KEY
(PID) REFERENCES PROPERTY (PID);

Table altered.

W21056374 >
W21056374 > ALTER TABLE VIEWING ADD CONSTRAINT FKEY_VG FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);

Table altered.

W21056374 >
W21056374 > ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_ADD FOREIGN KEY
(PADDRESS) REFERENCES ADDRESS (ADD_NO);

Table altered.

W21056374 > ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_BRA FOREIGN KEY (BID)
REFERENCES BRANCH (BID);

Table altered.

W21056374 > ALTER TABLE PROPERTY ADD CONSTRAINT FKEY_CUS FOREIGN KEY
(OWNER_ID) REFERENCES CUSTOMER (CID);

Table altered.

W21056374 >
W21056374 > ALTER TABLE ESTATE_AGENT ADD CONSTRAINT FKEY_ADD2 FOREIGN KEY
(EADDRESS) REFERENCES ADDRESS (ADD_NO);

Table altered.

W21056374 >
W21056374 > ALTER TABLE VIEWS ADD CONSTRAINT FKEY_V1 FOREIGN KEY (VIEWER)
REFERENCES CUSTOMER (CID);

Table altered.

W21056374 > ALTER TABLE VIEWS ADD CONSTRAINT FKEY_V2 FOREIGN KEY (VIEW_REF)
REFERENCES VIEWING (REF);

Table altered.

W21056374 >
W21056374 > ALTER TABLE BRANCH ADD CONSTRAINT FKEY_ADD3 FOREIGN KEY
(BADDRESS) REFERENCES ADDRESS (ADD_NO);

Table altered.

W21056374 > ALTER TABLE BRANCH ADD CONSTRAINT FKEY_EA FOREIGN KEY (EA_NAME)
REFERENCES ESTATE_AGENT (NAME);

Table altered.

W21056374 > ALTER TABLE BRANCH ADD CONSTRAINT FKEY_MAN FOREIGN KEY
(MANAGER) REFERENCES STAFF_MEMBER (SID);
```

```
Table altered.

W21056374 >
W21056374 > ALTER TABLE STAFF_MEMBER ADD CONSTRAINT FKEY_ADD4 FOREIGN KEY
(SADDRESS) REFERENCES ADDRESS (ADD_NO);

Table altered.

W21056374 > ALTER TABLE STAFF_MEMBER ADD CONSTRAINT FKEY_BRA2 FOREIGN KEY
(BID) REFERENCES BRANCH (BID);

Table altered.

W21056374 >
W21056374 > ALTER TABLE CUSTOMER ADD CONSTRAINT FKEY_ADD5 FOREIGN KEY
(CADDRESS) REFERENCES ADDRESS (ADD_NO);

Table altered.

W21056374 >
W21056374 > ALTER TABLE SALE ADD CONSTRAINT FKEY_BUYER FOREIGN KEY
(BUYER_ID) REFERENCES CUSTOMER (CID);

Table altered.

W21056374 > ALTER TABLE SALE ADD CONSTRAINT FKEY_SPROP FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);

Table altered.

W21056374 >
W21056374 > ALTER TABLE LEASE ADD CONSTRAINT FKEY_RENTER FOREIGN KEY
(RENTER_ID) REFERENCES CUSTOMER (CID);

Table altered.

W21056374 > ALTER TABLE LEASE ADD CONSTRAINT FKEY_RPROP FOREIGN KEY (PID)
REFERENCES PROPERTY (PID);

Table altered.

W21056374 >
W21056374 > SET ECHO OFF
W21056374 > spool off
```

(B) Answer the following queries (retrievals) using SQL and/or PL/SQL and relational algebra.

(12 marks)

q1) Display details of *detached* or *semi-detached* properties for sale having at least four bedrooms in the *Fenham* or *Heaton* areas of *Newcastle* that were added to the system in the last 42 days.

Answer Part 2B (q1): Relational Algebra Expression: 2 marks

Π p.PID, p.TYPE, p.BEDROOMS, p.GARDEN, p.DATE_MARKETED, s.ASKING_PRICE, a.LINE2, a.LINE3, a.CITY (σ s.PID = p.PID and p.PADDRESS = a.ADD_NO and (p.TYPE = "D" or p.TYPE = "S") and p.BEDROOMS >= 4 and a.CITY like "%ewcastle%" and p.DATE_MARKETED > sysdate - 42 and ((a.LINE2 = "Fenham" or a.LINE2 = "fenham" or a.LINE2 = "Heaton" or a.LINE2 = "heaton") or (a.LINE3 = "Fenham" or a.LINE3 = "fenham" or a.LINE3 = "Heaton" or a.LINE3 = "heaton"))
(SALES_PROPERTY * PROPERTY * ADDRESS))

Answer Part 2B (q1): SQL and/or PL/SQL code: 4 marks

```
SET VERIFY ON
SET ECHO ON
SET SERVEROUTPUT ON

/*
Display details of detached or semi-detached properties for sale having at
least four bedrooms
in the Fenham or Heaton areas of Newcastle that were added to the system in
the last 42 days.
*/

CREATE OR REPLACE PROCEDURE sel_prop1 AS

CURSOR pp IS select p.PID, p.TYPE, p.BEDROOMS, p.DATE_MARKETED,
s.ASKING_PRICE, a.LINE1, a.LINE2, a.LINE3, a.CITY, a.POSTCODE
From PROPERTY p, SALES_PROPERTY s, ADDRESS a
Where s.PID = p.PID
  and p.PADDRESS = a.ADD_NO
  and p.TYPE in ('D', 'S')
```

```
  and p.BEDROOMS >= 4
  and a.CITY LIKE '%ewcastle%'
  and p.DATE_MARKETED > sysdate-42
  and (a.LINE2 IN ('Fenham', 'fenham', 'Heaton', 'heaton') or a.LINE3 IN
('Fenham', 'fenham', 'Heaton', 'heaton'));

v_prop pp%ROWTYPE;

BEGIN

Open pp;

LOOP
  FETCH  pp INTO v_prop;
  EXIT WHEN pp%NOTFOUND;

  DBMS_OUTPUT.PUT_LINE (chr(10));
  DBMS_OUTPUT.PUT_LINE ('Property ID is: ' || v_prop.PID);
  DBMS_OUTPUT.PUT_LINE ('Type is: ' || v_prop.TYPE);
  DBMS_OUTPUT.PUT_LINE ('Number of bedrooms: ' || v_prop.BEDROOMS);
  DBMS_OUTPUT.PUT_LINE ('Date marketed: ' || v_prop.DATE_MARKETED);
  DBMS_OUTPUT.PUT_LINE ('Asking price: ' || v_prop.ASKING_PRICE);
  DBMS_OUTPUT.PUT_LINE ('Address: ' || v_prop.LINE1 || ', ' || v_prop.LINE2
|| ', ' || v_prop.LINE3 || ', ' || v_prop.CITY || ', ' || v_prop.POSTCODE);

END LOOP;
CLOSE pp;

      EXCEPTION
            WHEN others THEN
                  DBMS_OUTPUT.PUT_LINE ('Error occurred');
               DBMS_OUTPUT.PUT_LINE ('SQL Error Code:  ' || SQLCODE);
               DBMS_OUTPUT.PUT_LINE ('SQL Error Message:  ' || SQLERRM);
               IF pp%ISOPEN THEN CLOSE pp;
               END IF;
END;
/
```

Answer Part 2B (q1): SQL code's output (e.g., SPOOL file contents or screenshots):

```
W21056374 > @D:\pp_query1.sql
W21056374 > SET SERVEROUTPUT ON
W21056374 >
W21056374 > /*
W21056374 > Display details of detached or semi-detached properties for
sale having at least four bedrooms
W21056374 > in the Fenham or Heaton areas of Newcastle that were added to
the system in the last 42 days.
W21056374 > */
W21056374 >
W21056374 > CREATE OR REPLACE PROCEDURE sel_prop1 AS
  2
  3  CURSOR pp IS select p.PID, p.TYPE, p.BEDROOMS, p.DATE_MARKETED,
s.ASKING_PRICE, a.LINE1, a.LINE2, a.LINE3, a.CITY, a.POSTCODE
  4  From PROPERTY p, SALES_PROPERTY s, ADDRESS a
  5  Where s.PID = p.PID
  6    and p.PADDRESS = a.ADD_NO
  7    and p.TYPE in ('D', 'S')
  8    and p.BEDROOMS >= 4
  9    and a.CITY LIKE '%ewcastle%'
 10    and p.DATE_MARKETED > sysdate-42
 11    and (a.LINE2 IN ('Fenham', 'fenham', 'Heaton', 'heaton') or a.LINE3
IN ('Fenham', 'fenham', 'Heaton', 'heaton'));
 12
 13  v_prop pp%ROWTYPE;
 14
 15  BEGIN
 16
 17  Open pp;
 18
 19  LOOP
 20    FETCH  pp INTO v_prop;
 21    EXIT WHEN pp%NOTFOUND;
 22
 23    DBMS_OUTPUT.PUT_LINE (chr(10));
 24    DBMS_OUTPUT.PUT_LINE ('Property ID is: ' || v_prop.PID);
 25    DBMS_OUTPUT.PUT_LINE ('Type is: ' || v_prop.TYPE);
```

```
 26    DBMS_OUTPUT.PUT_LINE ('Number of bedrooms: ' || v_prop.BEDROOMS);
 27    DBMS_OUTPUT.PUT_LINE ('Date marketed: ' || v_prop.DATE_MARKETED);
 28    DBMS_OUTPUT.PUT_LINE ('Asking price: ' || v_prop.ASKING_PRICE);
 29    DBMS_OUTPUT.PUT_LINE ('Address: ' || v_prop.LINE1 || ', ' ||
v_prop.LINE2 || ', ' || v_prop.LINE3 || ', ' || v_prop.CITY || ', ' ||
v_prop.POSTCODE);
 30
 31  END LOOP;
 32  CLOSE pp;
 33
 34        EXCEPTION
 35          WHEN others THEN
 36            DBMS_OUTPUT.PUT_LINE ('Error occurred');
 37                DBMS_OUTPUT.PUT_LINE ('SQL Error Code:  ' || SQLCODE);
 38                DBMS_OUTPUT.PUT_LINE ('SQL Error Message:  ' ||
SQLERRM);
 39                IF pp%ISOPEN THEN CLOSE pp;
 40                END IF;
 41  END;
 42  /

Procedure created.

W21056374 > execute sel_prop1;


Property ID is: P349879
Type is: D
Number of bedrooms: 5
Date marketed: 01-JUN-22
Asking price: 450000
Address: 17 Fraser Drive, Fenham, , Newcastle upon Tyne, NE4 2BB


Property ID is: P018234
Type is: S
Number of bedrooms: 4
Date marketed: 27-MAY-22
Asking price: 375000
Address: 88 Carpenters Croft, Heaton, , Newcastle upon Tyne, NE6 9FD
```

```
PL/SQL procedure successfully completed.


W21056374 > spool off
```

---

q2)Display details of properties sold in Newcastle, Sunderland, Gateshead or Cramlington between £195,000 and £375,000 in the years 2018 to 2022.

Answer Part 2B (q2): Relational Algebra Expression: 2 marks

Π p.PID, p.TYPE, p.BEDROOMS, p.GARDEN, s.SALE_PRICE, s.SALE_DATE, a.CITY (σ s.PID = p.PID and p.PADDRESS = a.ADD_NO and (s.SALE_PRICE >= 195000 and s.SALE_PRICE <= 375000) and (s.SALE_DATE >= '01-JAN-2018' and s.SALE_DATE <= '31-DEC-2022') and (a.CITY like "%ewcastle%" or a.CITY like "%underland" or a.CITY like "%ateshead" or a.CITY like "%ramlington") (SALE * PROPERTY * ADDRESS))

Answer Part 2B (q2): SQL and/or PL/SQL code: 4 marks

```
SET VERIFY ON
SET ECHO ON
SET SERVEROUTPUT ON


/*
Display details of properties sold in Newcastle, Sunderland, Gateshead or
Cramlington between
£195,000 and £375,000 in the years 2018 to 2022.
*/

CREATE OR REPLACE PROCEDURE sel_prop2 AS

CURSOR pp IS select p.PID, p.TYPE, p.BEDROOMS, p.GARDEN, s.SALE_PRICE,
s.SALE_DATE, a.CITY
From SALE s, PROPERTY p, ADDRESS a
Where s.PID = p.PID
  and p.PADDRESS = a.ADD_NO
  and s.SALE_PRICE between 195000 and 375000
  and s.SALE_DATE between '01-JAN-2018' and '31-DEC-2022'
  and (a.CITY LIKE '%ewcastle%' or a.CITY LIKE '%underland' or a.CITY LIKE
'%ateshead' or a.CITY LIKE '%ramlington');

v_prop pp%ROWTYPE;
```

```
BEGIN

Open pp;

LOOP
  FETCH  pp INTO v_prop;
  EXIT WHEN pp%NOTFOUND;

  DBMS_OUTPUT.PUT_LINE (chr(10));
  DBMS_OUTPUT.PUT_LINE ('Property ID is: ' || v_prop.PID);
  DBMS_OUTPUT.PUT_LINE ('Type is: ' || v_prop.TYPE);
  DBMS_OUTPUT.PUT_LINE ('Number of bedrooms: ' || v_prop.BEDROOMS);
  DBMS_OUTPUT.PUT_LINE ('Garden: ' || v_prop.GARDEN);
  DBMS_OUTPUT.PUT_LINE ('Sale price: ' || v_prop.SALE_PRICE);
  DBMS_OUTPUT.PUT_LINE ('Sale date: ' || v_prop.SALE_DATE);
  DBMS_OUTPUT.PUT_LINE ('City: ' || v_prop.CITY);

END LOOP;
CLOSE pp;

      EXCEPTION
            WHEN others THEN
                  DBMS_OUTPUT.PUT_LINE ('Error occurred');
               DBMS_OUTPUT.PUT_LINE ('SQL Error Code:  ' || SQLCODE);
               DBMS_OUTPUT.PUT_LINE ('SQL Error Message:  ' || SQLERRM);
               IF pp%ISOPEN THEN CLOSE pp;
               END IF;
END;
/
```

Answer Part 2B (q2): SQL code's output (e.g., SPOOL file contents or screenshots): Minus 2 marks (if output is missing or deduct partial marks accordingly for partial or inadequate output)

```
W21056374 > @D:\pp_query2.sql
W21056374 > SET VERIFY ON
W21056374 > SET ECHO ON
W21056374 > SET SERVEROUTPUT ON
W21056374 >
```

```
W21056374 >
W21056374 > /*
W21056374 > Display details of properties sold in Newcastle, Sunderland,
Gateshead or Cramlington between
W21056374 > £195,000 and £375,000 in the years 2018 to 2022.
W21056374 > */
W21056374 >
W21056374 > CREATE OR REPLACE PROCEDURE sel_prop2 AS
  2
  3  CURSOR pp IS select p.PID, p.TYPE, p.BEDROOMS, p.GARDEN, s.SALE_PRICE,
s.SALE_DATE, a.CITY
  4  From SALE s, PROPERTY p, ADDRESS a
  5  Where s.PID = p.PID
  6    and p.PADDRESS = a.ADD_NO
  7    and s.SALE_PRICE between 195000 and 375000
  8    and s.SALE_DATE between '01-JAN-2018' and '31-DEC-2022'
  9    and (a.CITY LIKE '%ewcastle%' or a.CITY LIKE '%underland' or a.CITY
LIKE '%ateshead' or a.CITY LIKE '%ramlington');
 10
 11  v_prop pp%ROWTYPE;
 12
 13  BEGIN
 14
 15  Open pp;
 16
 17  LOOP
 18    FETCH  pp INTO v_prop;
 19    EXIT WHEN pp%NOTFOUND;
 20
 21    DBMS_OUTPUT.PUT_LINE (chr(10));
 22    DBMS_OUTPUT.PUT_LINE ('Property ID is: ' || v_prop.PID);
 23    DBMS_OUTPUT.PUT_LINE ('Type is: ' || v_prop.TYPE);
 24    DBMS_OUTPUT.PUT_LINE ('Number of bedrooms: ' || v_prop.BEDROOMS);
 25    DBMS_OUTPUT.PUT_LINE ('Garden: ' || v_prop.GARDEN);
 26    DBMS_OUTPUT.PUT_LINE ('Sale price: ' || v_prop.SALE_PRICE);
 27    DBMS_OUTPUT.PUT_LINE ('Sale date: ' || v_prop.SALE_DATE);
 28    DBMS_OUTPUT.PUT_LINE ('City: ' || v_prop.CITY);
 29
 30  END LOOP;
 31  CLOSE pp;
 32
```

```
 33        EXCEPTION
 34          WHEN others THEN
 35            DBMS_OUTPUT.PUT_LINE ('Error occurred');
 36                DBMS_OUTPUT.PUT_LINE ('SQL Error Code:  ' || SQLCODE);
 37                DBMS_OUTPUT.PUT_LINE ('SQL Error Message:  ' ||
SQLERRM);
 38                IF pp%ISOPEN THEN CLOSE pp;
 39                END IF;
 40  END;
 41  /

Procedure created.

W21056374 > execute sel_prop2;


Property ID is: P832748
Type is: S
Number of bedrooms: 3
Garden: Y
Sale price: 305000
Sale date: 20-DEC-21
City: Newcastle upon Tyne


Property ID is: P658439
Type is: T
Number of bedrooms: 2
Garden: Y
Sale price: 208500
Sale date: 18-JUN-18
City: Cramlington


Property ID is: P298473
Type is: B
Number of bedrooms: 4
Garden: Y
Sale price: 310000
Sale date: 25-JAN-22
City: Cramlington
```

```
PL/SQL procedure successfully completed.

W21056374 > spool off
```

Part 3 (20 marks)

(A) Create an object-based subset of the PropertyPortal database using object-relational (O-R) features of Oracle 11g/12c/19c. Select and justify any two entity types / relations of your choice which have relationships with each other and design and implement them using nested-relational and object-relational approach covered in this module. Your answer should include object types, object tables, data loading into object tables, and answering a suitable sample query.

(12 marks)

Selection and justification of entity types / relations for implementation using nested-relational and object-relational approach code (2 marks):

Three entities were chosen to implement the nested-relational and object-relational approach in this scenario: Property, Branch and Address. A property is registered at just one branch (Property 0..* <<--registered at--> 1..1 Branch) and a branch registers many properties (Branch 1..1 <--registers--> 0..* Property). Consequently, it makes sense to insert a reference to Branch in the Property User Defined Type (UDT) and a nested table referencing Property in the Branch UDT, as follows:

Property (PID, Description, Type, Bedrooms, Garden, Date Marketed, Owner_ID, **registered_at: REF Branch**)

Branch (BID, Bphone, Bemail, Manager, EA_Name, **registers: nested table [REF Property]**)

A property and branch are both located at just one address, therefore it is appropriate to insert a reference to Address in the Property and Branch UDTs. Conversely, an address can refer to a property, branch or any other entity type which may be added later. Therefore, a bidirectional relationship between the Address UDT and Property or Branch UDTs is not desirable as this will result in many nulls in the former.

The final UDTs are as follows:

Property (PID, Description, Type, Bedrooms, Garden, Date Marketed, Owner_ID, **registered_at: REF Branch**, **located_at: REF Address**)

Branch (BID, Bphone, Bemail, Manager, EA_Name, **registers: nested table [REF Property]**, **sited_at: REF Address**)

Address (Add_No, Line1, Line2, Line3, City, Postcode)

SQL code for creating object types including nested table types (4 marks) and SQL code for creating object tables including nested tables (2 marks):

```
SET VERIFY ON
SET ECHO ON

Create Type ADDRESS_T as Object
  (ADD_NO         CHAR(8),
   LINE1          VARCHAR(30),
   LINE2          VARCHAR(30),
   LINE3          VARCHAR(30),
   CITY                 VARCHAR(20),
   POSTCODE       VARCHAR(8)
  )
/

Create Type PROPERTY_T
/

Create Type PROPERTY_NT_TYPE as Table of REF PROPERTY_T
/

Create Type BRANCH_T as Object
  (BID                  CHAR(3),
   ADD_REF        REF ADDRESS_T,
   BPHONE         CHAR(11),
   BEMAIL         VARCHAR(50),
   MANAGER        CHAR(5),
   EA_NAME        VARCHAR(30),
   PROP_REF       PROPERTY_NT_TYPE
  )
/

Create or Replace Type PROPERTY_T as Object
  (PID                  CHAR(7),
   ADD_REF        REF ADDRESS_T,
   DESCRIPTION          VARCHAR(500),
   TYPE                 CHAR(1),
   BEDROOMS       NUMBER(2),
   GARDEN         CHAR(1),
   DATE_MARKETED  DATE,
   OWNER_ID       CHAR(7),
   BRA_REF        REF BRANCH_T
```

```
  )
/

Create Table ADDRESS_TAB of ADDRESS_T
(primary key (ADD_NO))
/

Create Table BRANCH_TAB of BRANCH_T
(primary key (BID),
foreign key (ADD_REF) references ADDRESS_TAB)
Nested Table PROP_REF Store As BRANCH_PROP_NTAB
/

Create Table PROPERTY_TAB of PROPERTY_T
(primary key (PID),
 foreign key (BRA_REF) references BRANCH_TAB,
 foreign key (ADD_REF) references ADDRESS_TAB)
/

SET ECHO OFF
```

Populating the object tables:

Provide SQL code for creating / inserting objects in object tables **and populating** relevant one-to-many, many-to-one, and many-to-many relationships among the objects in your object tables (2 Marks)

```
SET ECHO ON

SET VERIFY ON

REM Loading data into ADDRESS_TAB table.

INSERT INTO ADDRESS_TAB
  VALUES ('AD175002', 'Newcastle House', 'Albany Court', 'Newcastle
Business Park', 'Newcastle upon Tyne', 'NE4 7YB');

INSERT INTO ADDRESS_TAB
  VALUES ('AD336028', '566 Building One', 'Chiswick High Road', 'Chiswick
Park', 'London', 'W4 5BE');

INSERT INTO ADDRESS_TAB
  VALUES ('AD528406', 'Bishop Court', 'Front St', 'Whickham', 'Newcastle
upon Tyne', 'NE16 4JQ');

INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
```

```
  VALUES ('AD125837', '49 Bunyan Road', 'Fenham', 'Newcastle upon Tyne',
'NE4 1SK');

INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD229016', '17 Fraser Drive', 'Fenham', 'Newcastle upon Tyne',
'NE4 2BB');

INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD853037', '12 Dunecht Road', 'Fenham', 'Newcastle upon Tyne',
'NE4 5HN');

INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD002385', 'The Willows', 'Grove Road', 'Gateshead', 'NE9 4KL');

INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD456732', '3 North Street', 'Heaton', 'Newcastle upon Tyne',
'NE6 9LT');

INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD123673', '88 Carpenters Croft', 'Heaton', 'Newcastle upon
Tyne', 'NE6 9FD');

INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  VALUES ('AD284145', 'Helix', 'St James Blvd', 'Newcastle upon Tyne', 'NE4
5BZ');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD010264', '5 - 9 Bon Accord Crescent', 'Aberdeen', 'AB11 6DN');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD693570', '33 Margaret Street', 'London', 'W1G 0JD');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD469321', '55 Baker Street', 'London', 'W1U 8AN');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD183994', '33 Union Street', 'Newcastle upon Tyne', 'NE1 7DN');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD809835', '30 Cloth Market', 'Newcastle upon Tyne', 'NE1 1EE');
```

```
INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD324873', '6 High Street', 'Newcastle upon Tyne', 'NE2 7SD');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD000184', '77 Station Road', 'Newcastle upon Tyne', 'NE6 8HH');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD117439', '45 Main Street', 'Newcastle upon Tyne', 'NE9 8DC');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD405337', '67 Church Road', 'Newcastle upon Tyne', 'NE5 3ED');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD811003', '345 Victoria Road', 'Newcastle upon Tyne', 'NE7
3EE');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD222235', '8 Church Lane', 'Newcastle upon Tyne', 'NE4 4NL');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD101018', '103 Mill Lane', 'Newcastle upon Tyne', 'NE8 1NN');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD388093', '23 York Road', 'Newcastle upon Tyne', 'NE2 4FA');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD999420', '29 Victoria Crescent', 'Newcastle upon Tyne', 'NE7
2XY');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD939346', '98 Windmill Road', 'Newcastle upon Tyne', 'NE8
3AS');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD913021', '15 Gray Road', 'Blyth', 'NE22 0FF');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD902784', '88 School Lane', 'Cramlington', 'NE23 3DD');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD090898', '33 Green Lane', 'Cramlington', 'NE23 0CM');
```

```
INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD059867', '8 Lancaster Road', 'Cramlington', 'NE23 2VT');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD111539', '212 King Street', 'Cramlington', 'NE23 1NJ');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD620439', '54 Castle Road', 'Cramlington', 'NE23 6SA');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD110293', '25 New Street', 'Gateshead', 'NE9 8HG');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD000005', '38 West Street', 'Gateshead', 'NE7 9BC');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD394765', '22 Stormont Road', 'Gateshead', 'NE8 9DN');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD950367', '66 Highfield Road', 'Sunderland', 'SR5 8DS');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD893274', '101 Albert Road', 'Sunderland', 'SR6 1TR');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD192832', '58 Mill Road', 'Sunderland', 'SR4 7JJ');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD109275', '59 Victory Lane', 'Sunderland', 'SR6 6LM');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD753947', '68 Park Street', 'Sunderland', 'SR6 1BC');

INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  VALUES ('AD539564', '2 George Street', 'Sunderland', 'SR7 9GG');


REM Loading data into BRANCH_TAB table.

INSERT INTO BRANCH_TAB
```

```
  VALUES ('B01', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD528406'), '01914887968', 'whickham@your-move.co.uk', 'S9921', 'Your
Move', PROPERTY_NT_TYPE());

INSERT INTO BRANCH_TAB
  VALUES ('B02', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD284145'), '01919171444', 'newcastle@savills.com', 'S5790', 'Savills',
PROPERTY_NT_TYPE());

INSERT INTO BRANCH_TAB
  VALUES ('B03', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD183994'), '01915382636', 'newc@foxtons.co.uk', 'S5693', 'Foxtons',
PROPERTY_NT_TYPE());

INSERT INTO BRANCH_TAB
  VALUES ('B04', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD809835'), '01916078450', 'newcastle@acandco.com', 'S0137', 'Aberdein
Considine', PROPERTY_NT_TYPE());



REM Loading data into PROPERTY_TAB table and
REM include REF to the Branch ID (BID)

INSERT INTO PROPERTY_TAB
  VALUES ('P349879', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD229016'), 'This exceptional home has been created with great imagination
and finished to a high specification.', 'D', 5, 'Y', '01-JUN-2022',
'C552901', (select REF(e) from BRANCH_TAB e where e.BID = 'B01'));

INSERT INTO PROPERTY_TAB
  VALUES ('P018234', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD123673'), 'A spacious extended four bedroom Victorian terraced home,
with a loft conversion, south-west facing private enclosed garden and
detached garage with power.', 'S', 4, 'Y', '27-MAY-2022', 'C174439',
(select REF(e) from BRANCH_TAB e where e.BID = 'B01'));

INSERT INTO PROPERTY_TAB
  VALUES ('P832748', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD853037'), 'This spacious and well-presented three bedroom semi-detached
family home comes to market offering generous living space throughout.',
```

```
'S', 3, 'Y', '08-SEP-2021', 'C986032', (select REF(e) from BRANCH_TAB e
where e.BID = 'B04'));

INSERT INTO PROPERTY_TAB
  VALUES ('P247587', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD913021'), 'The property benefits from a lovely wrap around garden, off-
road parking, and a large attached garage.', 'B', 3, 'Y', '04-AUG-2019',
'C000032', (select REF(e) from BRANCH_TAB e where e.BID = 'B03'));

INSERT INTO PROPERTY_TAB
  VALUES ('P347853', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD753947'), 'Beautifully presented two bedroom first floor apartment town
centre location, two bedrooms master with en suite and built in wardrobe,
modern bright and airy open plan living room with balcony, kitchen with
appliances and granite worktops, main bathroom, entry phone system, lifts
to all floors, allocated underground parking.', 'F', 2, 'N', '09-NOV-2021',
'C856290', (select REF(e) from BRANCH_TAB e where e.BID = 'B04'));

INSERT INTO PROPERTY_TAB
  VALUES ('P984752', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD539564'), 'The conclusion of a short private road, this 4 bedroom
detached modern build sits back from the High Street.', 'D', 4, 'Y', '28-
APR-2022', 'C000032', (select REF(e) from BRANCH_TAB e where e.BID =
'B02'));

INSERT INTO PROPERTY_TAB
  VALUES ('P223743', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD002385'), 'The property boasts laminate flooring throughout, a separate
bathroom semi open plan kitchen and reception with a single bedroom.', 'F',
1, 'N', '05-MAY-2022', 'C856290', (select REF(e) from BRANCH_TAB e where
e.BID = 'B01'));

INSERT INTO PROPERTY_TAB
  VALUES ('P118364', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD111539'), 'A beautifully stylish two bedroom first floor apartment in
this prestigious and modern development, with extremely spacious
accommodation, modern kitchen, and benefiting from one parking space.',
'F', 2, 'N', '05-AUG-2020', 'C856290', (select REF(e) from BRANCH_TAB e
where e.BID = 'B02'));

INSERT INTO PROPERTY_TAB
```

```
  VALUES ('P658439', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD620439'), 'The house is arranged over two floors. To the ground floor
there are two spacious reception rooms, a modern fitted kitchen and
downstairs shower room.', 'T', 2, 'Y', '13-FEB-2018', 'C302849', (select
REF(e) from BRANCH_TAB e where e.BID = 'B01'));

INSERT INTO PROPERTY_TAB
  VALUES ('P102934', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD110293'), 'A beautifully presented home in an ideal position for access
to local schooling and the railway station. Fabulous garden, home office
and off-road parking.', 'D', 3, 'Y', '17-JUN-2017', 'C740283', (select
REF(e) from BRANCH_TAB e where e.BID = 'B02'));

INSERT INTO PROPERTY_TAB
  VALUES ('P298473', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD902784'), 'A substantial detached chalet bungalow providing spacious and
adaptable accommodation. This beautifully presented individual home of
character has been sympathetically extended and refurbished to high
specification now providing a comfortable home ready to move into with no
upward chain.', 'B', 4, 'Y', '11-OCT-2021', 'C985422', (select REF(e) from
BRANCH_TAB e where e.BID = 'B01'));


REM Populate BRANCH_TAB.PROP_REF which
REM is a nested table of type PROPERTY_NT_TYPE

INSERT INTO TABLE (SELECT D.PROP_REF FROM BRANCH_TAB D WHERE D.BID = 'B01')
SELECT REF(E) FROM PROPERTY_TAB E WHERE E.BRA_REF.BID = 'B01';

INSERT INTO TABLE (SELECT D.PROP_REF FROM BRANCH_TAB D WHERE D.BID = 'B02')
SELECT REF(E) FROM PROPERTY_TAB E WHERE E.BRA_REF.BID = 'B02';

INSERT INTO TABLE (SELECT D.PROP_REF FROM BRANCH_TAB D WHERE D.BID = 'B03')
SELECT REF(E) FROM PROPERTY_TAB E WHERE E.BRA_REF.BID = 'B03';

INSERT INTO TABLE (SELECT D.PROP_REF FROM BRANCH_TAB D WHERE D.BID = 'B04')
SELECT REF(E) FROM PROPERTY_TAB E WHERE E.BRA_REF.BID = 'B04';


COMMIT;
```

```
SET ECHO OFF
```

Querying the object tables:

Provide SQL code for **running** a sample query to retrieve some data from the object tables (2 Marks).

```
SET ECHO ON
SET SERVEROUTPUT ON


REM How many properties has the Your Move branch in Whickham (B01)
REM marketed in Cramlington the last 5 years? What are their addresses?

select COUNT(b.column_value.pid)
from branch_tab a, table(a.prop_ref) b
where a.bid = 'B01'
and b.column_value.add_ref.city = 'Cramlington'
and b.column_value.date_marketed > sysdate - 1825;

select b.column_value.pid, b.column_value.add_ref.line1 as STREET,
b.column_value.add_ref.city as CITY, b.column_value.add_ref.postcode as
POSTCODE
from branch_tab a, table(a.prop_ref) b
where a.bid = 'B01'
and b.column_value.add_ref.city = 'Cramlington'
and b.column_value.date_marketed > sysdate - 1825;
```

Provide the above SQL code's output (e.g., SPOOL file contents or screenshots): Minus 2 marks (if output is missing or deduct partial marks accordingly for partial or inadequate output)

Output from type and table creation

```
W21056374 > @D:\pp_or_tabs.sql
W21056374 >
W21056374 > Create Type ADDRESS_T as Object
```

```
  2    (ADD_NO              CHAR(8),
  3   LINE1            VARCHAR(30),
  4   LINE2            VARCHAR(30),
  5   LINE3            VARCHAR(30),
  6   CITY             VARCHAR(20),
  7   POSTCODE         VARCHAR(8)
  8    )
  9  /

Type created.

W21056374 >
W21056374 > Create Type PROPERTY_T
  2  /

Type created.

W21056374 >
W21056374 > Create Type PROPERTY_NT_TYPE as Table of REF PROPERTY_T
  2  /

Type created.

W21056374 >
W21056374 > Create Type BRANCH_T as Object
  2    (BID            CHAR(3),
  3   ADD_REF          REF ADDRESS_T,
  4   BPHONE                CHAR(11),
  5   BEMAIL                VARCHAR(50),
  6   MANAGER          CHAR(5),
  7   EA_NAME          VARCHAR(30),
  8   PROP_REF         PROPERTY_NT_TYPE
  9    )
 10  /

Type created.

W21056374 >
W21056374 > Create or Replace Type PROPERTY_T as Object
  2    (PID            CHAR(7),
  3   ADD_REF          REF ADDRESS_T,
```

```
  4   DESCRIPTION      VARCHAR(500),
  5   TYPE             CHAR(1),
  6   BEDROOMS         NUMBER(2),
  7   GARDEN                CHAR(1),
  8   DATE_MARKETED         DATE,
  9   OWNER_ID         CHAR(7),
 10   BRA_REF          REF BRANCH_T
 11    )
 12  /

Type created.

W21056374 >
W21056374 > Create Table ADDRESS_TAB of ADDRESS_T
  2  (primary key (ADD_NO))
  3  /

Table created.

W21056374 >
W21056374 > Create Table BRANCH_TAB of BRANCH_T
  2  (primary key (BID),
  3  foreign key (ADD_REF) references ADDRESS_TAB)
  4  Nested Table PROP_REF Store As BRANCH_PROP_NTAB
  5  /

Table created.

W21056374 >
W21056374 > Create Table PROPERTY_TAB of PROPERTY_T
  2  (primary key (PID),
  3   foreign key (BRA_REF) references BRANCH_TAB,
  4   foreign key (ADD_REF) references ADDRESS_TAB)
  5  /

Table created.

W21056374 >
W21056374 > SET ECHO OFF
W21056374 > spool off
```

Output from inserts

```
W21056374 > @D:\pp_or_inserts.sql
W21056374 >
W21056374 > SET VERIFY ON
W21056374 >
W21056374 > REM Loading data into ADDRESS_TAB table.
W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB
  2    VALUES ('AD175002', 'Newcastle House', 'Albany Court', 'Newcastle
Business Park', 'Newcastle upon Tyne', 'NE4 7YB');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB
  2    VALUES ('AD336028', '566 Building One', 'Chiswick High Road',
'Chiswick Park', 'London', 'W4 5BE');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB
  2    VALUES ('AD528406', 'Bishop Court', 'Front St', 'Whickham',
'Newcastle upon Tyne', 'NE16 4JQ');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD125837', '49 Bunyan Road', 'Fenham', 'Newcastle upon
Tyne', 'NE4 1SK');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD229016', '17 Fraser Drive', 'Fenham', 'Newcastle upon
Tyne', 'NE4 2BB');
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD853037', '12 Dunecht Road', 'Fenham', 'Newcastle upon
Tyne', 'NE4 5HN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD002385', 'The Willows', 'Grove Road', 'Gateshead', 'NE9
4KL');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD456732', '3 North Street', 'Heaton', 'Newcastle upon
Tyne', 'NE6 9LT');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD123673', '88 Carpenters Croft', 'Heaton', 'Newcastle upon
Tyne', 'NE6 9FD');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, Line2, City, Postcode)
  2    VALUES ('AD284145', 'Helix', 'St James Blvd', 'Newcastle upon Tyne',
'NE4 5BZ');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD010264', '5 - 9 Bon Accord Crescent', 'Aberdeen', 'AB11
6DN');
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD693570', '33 Margaret Street', 'London', 'W1G 0JD');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD469321', '55 Baker Street', 'London', 'W1U 8AN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD183994', '33 Union Street', 'Newcastle upon Tyne', 'NE1
7DN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD809835', '30 Cloth Market', 'Newcastle upon Tyne', 'NE1
1EE');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD324873', '6 High Street', 'Newcastle upon Tyne', 'NE2
7SD');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD000184', '77 Station Road', 'Newcastle upon Tyne', 'NE6
8HH');
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD117439', '45 Main Street', 'Newcastle upon Tyne', 'NE9
8DC');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD405337', '67 Church Road', 'Newcastle upon Tyne', 'NE5
3ED');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD811003', '345 Victoria Road', 'Newcastle upon Tyne', 'NE7
3EE');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD222235', '8 Church Lane', 'Newcastle upon Tyne', 'NE4
4NL');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD101018', '103 Mill Lane', 'Newcastle upon Tyne', 'NE8
1NN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD388093', '23 York Road', 'Newcastle upon Tyne', 'NE2
4FA');
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD999420', '29 Victoria Crescent', 'Newcastle upon Tyne',
'NE7 2XY');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD939346', '98 Windmill Road', 'Newcastle upon Tyne', 'NE8
3AS');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD913021', '15 Gray Road', 'Blyth', 'NE22 0FF');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD902784', '88 School Lane', 'Cramlington', 'NE23 3DD');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD090898', '33 Green Lane', 'Cramlington', 'NE23 0CM');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD059867', '8 Lancaster Road', 'Cramlington', 'NE23 2VT');

1 row created.
```

```
W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD111539', '212 King Street', 'Cramlington', 'NE23 1NJ');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD620439', '54 Castle Road', 'Cramlington', 'NE23 6SA');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD110293', '25 New Street', 'Gateshead', 'NE9 8HG');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD000005', '38 West Street', 'Gateshead', 'NE7 9BC');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD394765', '22 Stormont Road', 'Gateshead', 'NE8 9DN');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD950367', '66 Highfield Road', 'Sunderland', 'SR5 8DS');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD893274', '101 Albert Road', 'Sunderland', 'SR6 1TR');

1 row created.
```

```
W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD192832', '58 Mill Road', 'Sunderland', 'SR4 7JJ');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD109275', '59 Victory Lane', 'Sunderland', 'SR6 6LM');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD753947', '68 Park Street', 'Sunderland', 'SR6 1BC');

1 row created.

W21056374 >
W21056374 > INSERT INTO ADDRESS_TAB (Add_No, Line1, City, Postcode)
  2    VALUES ('AD539564', '2 George Street', 'Sunderland', 'SR7 9GG');

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into BRANCH_TAB table.
W21056374 >
W21056374 > INSERT INTO BRANCH_TAB
  2    VALUES ('B01', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD528406'), '01914887968', 'whickham@your-move.co.uk', 'S9921', 'Your
Move', PROPERTY_NT_TYPE());

1 row created.

W21056374 >
W21056374 > INSERT INTO BRANCH_TAB
  2    VALUES ('B02', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD284145'), '01919171444', 'newcastle@savills.com', 'S5790', 'Savills',
PROPERTY_NT_TYPE());
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO BRANCH_TAB
  2    VALUES ('B03', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD183994'), '01915382636', 'newc@foxtons.co.uk', 'S5693', 'Foxtons',
PROPERTY_NT_TYPE());

1 row created.

W21056374 >
W21056374 > INSERT INTO BRANCH_TAB
  2    VALUES ('B04', (select REF(e) from ADDRESS_TAB e where e.ADD_NO =
'AD809835'), '01916078450', 'newcastle@acandco.com', 'S0137', 'Aberdein
Considine', PROPERTY_NT_TYPE());

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Loading data into PROPERTY_TAB table and
W21056374 > REM include REF to the Branch ID (BID)
W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P349879', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD229016'), 'This exceptional home has been created with great
imagination and finished to a high specification.', 'D', 5, 'Y', '01-JUN-
2022', 'C552901', (select REF(e) from BRANCH_TAB e where e.BID = 'B01'));

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P018234', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD123673'), 'A spacious extended four bedroom Victorian terraced home,
with a loft conversion, south-west facing private enclosed garden and
detached garage with power.', 'S', 4, 'Y', '27-MAY-2022', 'C174439',
(select REF(e) from BRANCH_TAB e where e.BID = 'B01'));

1 row created.
```

```
W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P832748', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD853037'), 'This spacious and well-presented three bedroom semi-
detached family home comes to market offering generous living space
throughout.', 'S', 3, 'Y', '08-SEP-2021', 'C986032', (select REF(e) from
BRANCH_TAB e where e.BID = 'B04'));

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P247587', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD913021'), 'The property benefits from a lovely wrap around garden,
off-road parking, and a large attached garage.', 'B', 3, 'Y', '04-AUG-
2019', 'C000032', (select REF(e) from BRANCH_TAB e where e.BID = 'B03'));

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P347853', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD753947'), 'Beautifully presented two bedroom first floor apartment
town centre location, two bedrooms master with en suite and built in
wardrobe, modern bright and airy open plan living room with balcony,
kitchen with appliances and granite worktops, main bathroom, entry phone
system, lifts to all floors, allocated underground parking.', 'F', 2, 'N',
'09-NOV-2021', 'C856290', (select REF(e) from BRANCH_TAB e where e.BID =
'B04'));

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P984752', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD539564'), 'The conclusion of a short private road, this 4 bedroom
detached modern build sits back from the High Street.', 'D', 4, 'Y', '28-
APR-2022', 'C000032', (select REF(e) from BRANCH_TAB e where e.BID =
'B02'));
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P223743', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD002385'), 'The property boasts laminate flooring throughout, a
separate bathroom semi open plan kitchen and reception with a single
bedroom.', 'F', 1, 'N', '05-MAY-2022', 'C856290', (select REF(e) from
BRANCH_TAB e where e.BID = 'B01'));

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P118364', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD111539'), 'A beautifully stylish two bedroom first floor apartment in
this prestigious and modern development, with extremely spacious
accommodation, modern kitchen, and benefiting from one parking space.',
'F', 2, 'N', '05-AUG-2020', 'C856290', (select REF(e) from BRANCH_TAB e
where e.BID = 'B02'));

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P658439', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD620439'), 'The house is arranged over two floors. To the ground floor
there are two spacious reception rooms, a modern fitted kitchen and
downstairs shower room.', 'T', 2, 'Y', '13-FEB-2018', 'C302849', (select
REF(e) from BRANCH_TAB e where e.BID = 'B01'));

1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P102934', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD110293'), 'A beautifully presented home in an ideal position for
access to local schooling and the railway station. Fabulous garden, home
office and off-road parking.', 'D', 3, 'Y', '17-JUN-2017', 'C740283',
(select REF(e) from BRANCH_TAB e where e.BID = 'B02'));
```

```
1 row created.

W21056374 >
W21056374 > INSERT INTO PROPERTY_TAB
  2    VALUES ('P298473', (select REF(e) from ADDRESS_TAB e where e.ADD_NO
= 'AD902784'), 'A substantial detached chalet bungalow providing spacious
and adaptable accommodation. This beautifully presented individual home of
character has been sympathetically extended and refurbished to high
specification now providing a comfortable home ready to move into with no
upward chain.', 'B', 4, 'Y', '11-OCT-2021', 'C985422', (select REF(e) from
BRANCH_TAB e where e.BID = 'B01'));

1 row created.

W21056374 >
W21056374 >
W21056374 > REM Populate BRANCH_TAB.PROP_REF which
W21056374 > REM is a nested table of type PROPERTY_NT_TYPE
W21056374 >
W21056374 > INSERT INTO TABLE (SELECT D.PROP_REF FROM BRANCH_TAB D WHERE
D.BID = 'B01')
  2  SELECT REF(E) FROM PROPERTY_TAB E WHERE E.BRA_REF.BID = 'B01';

5 rows created.

W21056374 >
W21056374 > INSERT INTO TABLE (SELECT D.PROP_REF FROM BRANCH_TAB D WHERE
D.BID = 'B02')
  2  SELECT REF(E) FROM PROPERTY_TAB E WHERE E.BRA_REF.BID = 'B02';

3 rows created.

W21056374 >
W21056374 > INSERT INTO TABLE (SELECT D.PROP_REF FROM BRANCH_TAB D WHERE
D.BID = 'B03')
  2  SELECT REF(E) FROM PROPERTY_TAB E WHERE E.BRA_REF.BID = 'B03';

1 row created.

W21056374 >
```

```
W21056374 > INSERT INTO TABLE (SELECT D.PROP_REF FROM BRANCH_TAB D WHERE
D.BID = 'B04')
  2  SELECT REF(E) FROM PROPERTY_TAB E WHERE E.BRA_REF.BID = 'B04';

2 rows created.

W21056374 >
W21056374 >
W21056374 > COMMIT;

Commit complete.

W21056374 >
W21056374 > SET ECHO OFF
W21056374 > spool off
```

Output from queries

```
W21056374 > @D:\pp_or_queries.sql
W21056374 > SET SERVEROUTPUT ON
W21056374 >
W21056374 >
W21056374 > REM How many properties has the Your Move branch in Whickham
(B01)
W21056374 > REM marketed in Cramlington the last 5 years? What are their
addresses?
W21056374 >
W21056374 > select COUNT(b.column_value.pid)
  2  from branch_tab a, table(a.prop_ref) b
  3  where a.bid = 'B01'
  4  and b.column_value.add_ref.city = 'Cramlington'
  5  and b.column_value.date_marketed > sysdate - 1825;

COUNT(B.COLUMN_VALUE.PID)
-------------------------
                        2

W21056374 >
```

```
W21056374 > select b.column_value.pid, b.column_value.add_ref.line1 as
STREET, b.column_value.add_ref.city as CITY,
b.column_value.add_ref.postcode as POSTCODE
  2  from branch_tab a, table(a.prop_ref) b
  3  where a.bid = 'B01'
  4  and b.column_value.add_ref.city = 'Cramlington'
  5  and b.column_value.date_marketed > sysdate - 1825;

COLUMN_ STREET                         CITY                POSTCODE
------- ------------------------------ ------------------- --------
P658439 54 Castle Road                 Cramlington         NE23 6SA
P298473 88 School Lane                 Cramlington         NE23 3DD

W21056374 > spool off
```

(B) Create a NOSQL subset of the PropertyPortal database using MongoDB. Select and justify any two entity types / relations of your choice which have relationships with each other and design and implement them using NOSQL database approach covered in this module. Your answer should include creating and populating collection(s) and answering a suitable sample query.

(8 marks)

Answer Part 3B

Selection and justification of entity types / relations for implementation using NOSQL database approach (2 mark):

The Branch, Property and Address entities have been selected to form the subset that will be created. They have been chosen as they provide an opportunity to employ both the embedded and normalised data models in MongoDB.

Alotaibi and Pardede (2019) propose some simple rules for transforming a relational database schema for use in NoSQL. If two entities are frequently accessed together, they state that the two should be transformed into collections with one embedded in

the other if there is a one-to-one or one-to-many relationship between them. Otherwise, they recommend inserting the ID of one collection as a reference into the other.

In the PropertyPortal conceptual data model, the Branch and Property entities have a one-to-many relationship and will be accessed together on a regular basis, as individuals interesting in viewing a property will need to contact the relevant branch which manages it. Consequently, it makes sense to embed the Property collection in the Branch collection in MongoDB.

These entities also have one-to-one relationship with the Address entity, which could be embedded in both Branch and Property. However, the reference approach using user-generated IDs is preferred as Branch is unlikely to be accessed concurrently with Address on a regular basis, as viewings are often scheduled via email or on the phone rather than in person. The same assumption regarding access does not apply to Property and Address, as both are intrinsically linked and are likely to be retrieved simultaneously in many instances. Despite this, the reference approach is being utilised to ensure consistency in the database. Moreover, if further entities are added to the database and they are related to Address (such as Staff_Member or Customer), it is assumed that frequent concurrent access will not be required.

MongoDB code for creating collections for the above entity types (1 mark):

```
// Create a collection for the Branch entity

db.createCollection("Branch")


// Create a collection for the Address entity

db.createCollection("Address")
```

Populating the above MongoDB collections: (3 Marks)

```
// Inserting data into the Branch collection
// which includes the embedded Property collection

db.Branch.insert(
   {_id: "B01",
    add_no: "AD528406",
    tel_no: "01914887968",
    email: "whickham@your-move.co.uk",
    manager_id: "S9921",
    estate_agent: "Your Move",
    properties:
    [{property_id: "P349879", add_no: "AD229016", type: "D", bedrooms: 5,
garden: "Y", date_marketed: ISODate("2022-06-01"), owner_id: "C552901",
description: "This exceptional home has been created with great imagination
and finished to a high specification."},
     {property_id: "P018234", add_no: "AD123673", type: "S", bedrooms: 4,
garden: "Y", date_marketed: ISODate("2022-05-27"), owner_id: "C174439",
description: "A spacious extended four bedroom Victorian terraced home,
with a loft conversion, south-west facing private enclosed garden and
detached garage with power."},
     {property_id: "P223743", add_no: "AD002385", type: "F", bedrooms: 1,
garden: "N", date_marketed: ISODate("2022-05-05"), owner_id: "C856290",
description: "The property boasts laminate flooring throughout, a separate
bathroom semi open plan kitchen and reception with a single bedroom."},
     {property_id: "P658439", add_no: "AD620439", type: "T", bedrooms: 2,
garden: "Y", date_marketed: ISODate("2018-02-13"), owner_id: "C302849",
description: "The house is arranged over two floors. To the ground floor
there are two spacious reception rooms, a modern fitted kitchen and
downstairs shower room."},
     {property_id: "P298473", add_no: "AD902784", type: "B", bedrooms: 4,
garden: "Y", date_marketed: ISODate("2021-10-11"), owner_id: "C985422",
description: "A substantial detached chalet bungalow providing spacious and
adaptable accommodation. This beautifully presented individual home of
character has been sympathetically extended and refurbished to high
specification now providing a comfortable home ready to move into with no
upward chain."}]
});

db.Branch.insert(
   {_id: "B02",
```

```
    add_no: "AD284145",
    tel_no: "01919171444",
    email: "newcastle@savills.com",
    manager_id: "S5790",
    estate_agent: "Savills",
    properties:
    [{property_id: "P984752", add_no: "AD539564", type: "D", bedrooms: 4,
garden: "Y", date_marketed: ISODate("2022-04-28"), owner_id: "C000032",
description: "The conclusion of a short private road, this 4 bedroom
detached modern build sits back from the High Street."},
     {property_id: "P118364", add_no: "AD111539", type: "F", bedrooms: 2,
garden: "N", date_marketed: ISODate("2020-08-05"), owner_id: "C856290",
description: "A beautifully stylish two bedroom first floor apartment in
this prestigious and modern development, with extremely spacious
accommodation, modern kitchen, and benefiting from one parking space."},
     {property_id: "P102934", add_no: "AD110293", type: "D", bedrooms: 3,
garden: "Y", date_marketed: ISODate("2017-06-17"), owner_id: "C740283",
description: "A beautifully presented home in an ideal position for access
to local schooling and the railway station. Fabulous garden, home office
and off-road parking."}]
});

db.Branch.insert(
   {_id: "B03",
    add_no: "AD183994",
    tel_no: "01915382636",
    email: "newc@foxtons.co.uk",
    manager_id: "S5693",
    estate_agent: "Foxtons",
    properties:
    [{property_id: "P247587", add_no: "AD913021", type: "B", bedrooms: 3,
garden: "Y", date_marketed: ISODate("2019-08-04"), owner_id: "C000032",
description: "The property benefits from a lovely wrap around garden, off-
road parking, and a large attached garage."}]
});

db.Branch.insert(
   {_id: "B04",
    add_no: "AD809835",
    tel_no: "01916078450",
    email: "newcastle@acandco.com",
```

```
    manager_id: "S0137",
    estate_agent: "Aberdein Considine",
    properties:
    [{property_id: "P832748", add_no: "AD853037", type: "S", bedrooms: 3,
garden: "Y", date_marketed: ISODate("2022-09-08"), owner_id: "C986032",
description: "This spacious and well-presented three bedroom semi-detached
family home comes to market offering generous living space throughout."},
     {property_id: "P347853", add_no: "AD753947", type: "F", bedrooms: 2,
garden: "N", date_marketed: ISODate("2021-11-09"), owner_id: "C856290",
description: "Beautifully presented two bedroom first floor apartment town
centre location, two bedrooms master with en suite and built in wardrobe,
modern bright and airy open plan living room with balcony, kitchen with
appliances and granite worktops, main bathroom, entry phone system, lifts
to all floors, allocated underground parking."}]
});


// Check that all data has been inserted into
// the Branch collection

db.Branch.find();

// Inserting data into the Address collection

db.Address.insertMany([
{_id: "AD175002", line1: "Newcastle House", line2: "Albany Court", line3:
"Newcastle Business Park", city: "Newcastle upon Tyne", postcode: "NE4
7YB"},
{_id: "AD528406", line1: "Bishop Court", line2: "Front St", line3:
"Whickham", city: "Newcastle upon Tyne", postcode: "NE16 4JQ"},
{_id: "AD284145", line1: "Helix", line2: "St James Blvd", city: "Newcastle
upon Tyne", postcode: "NE4 5BZ"},
{_id: "AD183994", line1: "33 Union Street", city: "Newcastle upon Tyne",
postcode: "NE1 7DN"},
{_id: "AD809835", line1: "30 Cloth Market", city: "Newcastle upon Tyne",
postcode: "NE1 1EE"},
{_id: "AD853037", line1: "12 Dunecht Road", line2: "Fenham", city:
"Newcastle upon Tyne", postcode: "NE4 5HN"},
{_id: "AD753947", line1: "68 Park Street", city: "Sunderland", postcode:
"SR6 1BC"},
```

```
{_id: "AD913021", line1: "15 Gray Road", city: "Blyth", postcode: "NE22
0FF"},
{_id: "AD539564", line1: "2 George Street", city: "Sunderland", postcode:
"SR7 9GG"},
{_id: "AD111539", line1: "212 King Street", city: "Cramlington", postcode:
"NE23 1NJ"},
{_id: "AD110293", line1: "25 New Street", city: "Gateshead", postcode: "NE9
8HG"},
{_id: "AD229016", line1: "17 Fraser Drive", line2: "Fenham", city:
"Newcastle upon Tyne", postcode: "NE4 2BB"},
{_id: "AD123673", line1: "88 Carpenters Croft", line2: "Heaton", city:
"Newcastle upon Tyne", postcode: "NE6 9FD"},
{_id: "AD002385", line1: "The Willows", line2: "Grove Road", city:
"Gateshead", postcode: "NE9 4KL"},
{_id: "AD620439", line1: "54 Castle Road", city: "Cramlington", postcode:
"NE23 6SA"},
{_id: "AD902784", line1: "88 School Lane", city: "Cramlington", postcode:
"NE23 3DD"},
]);


// Check that all data has been inserted into
// the Address collection

db.Address.find();
```

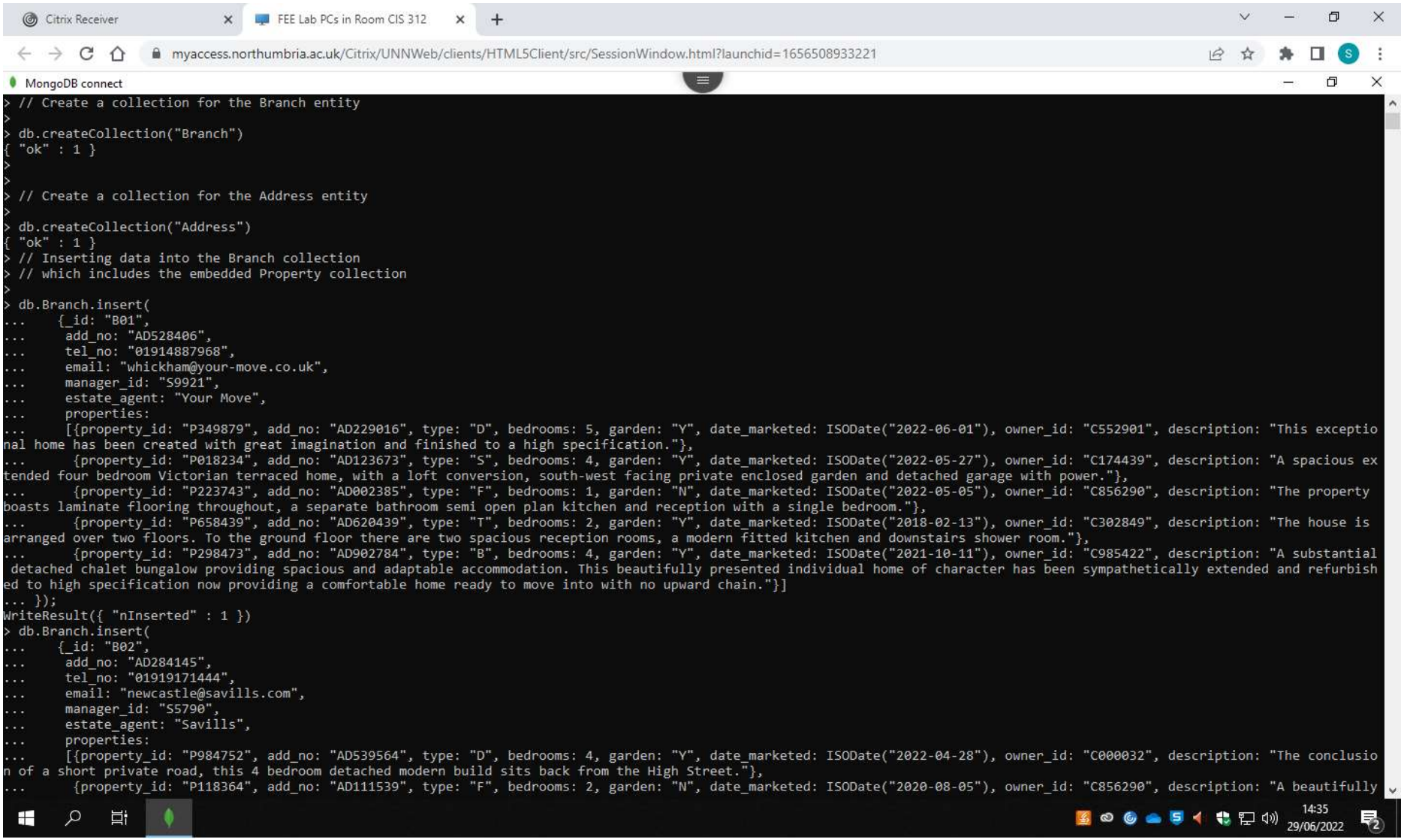
Querying the above MongoDB collections: (2 Marks)

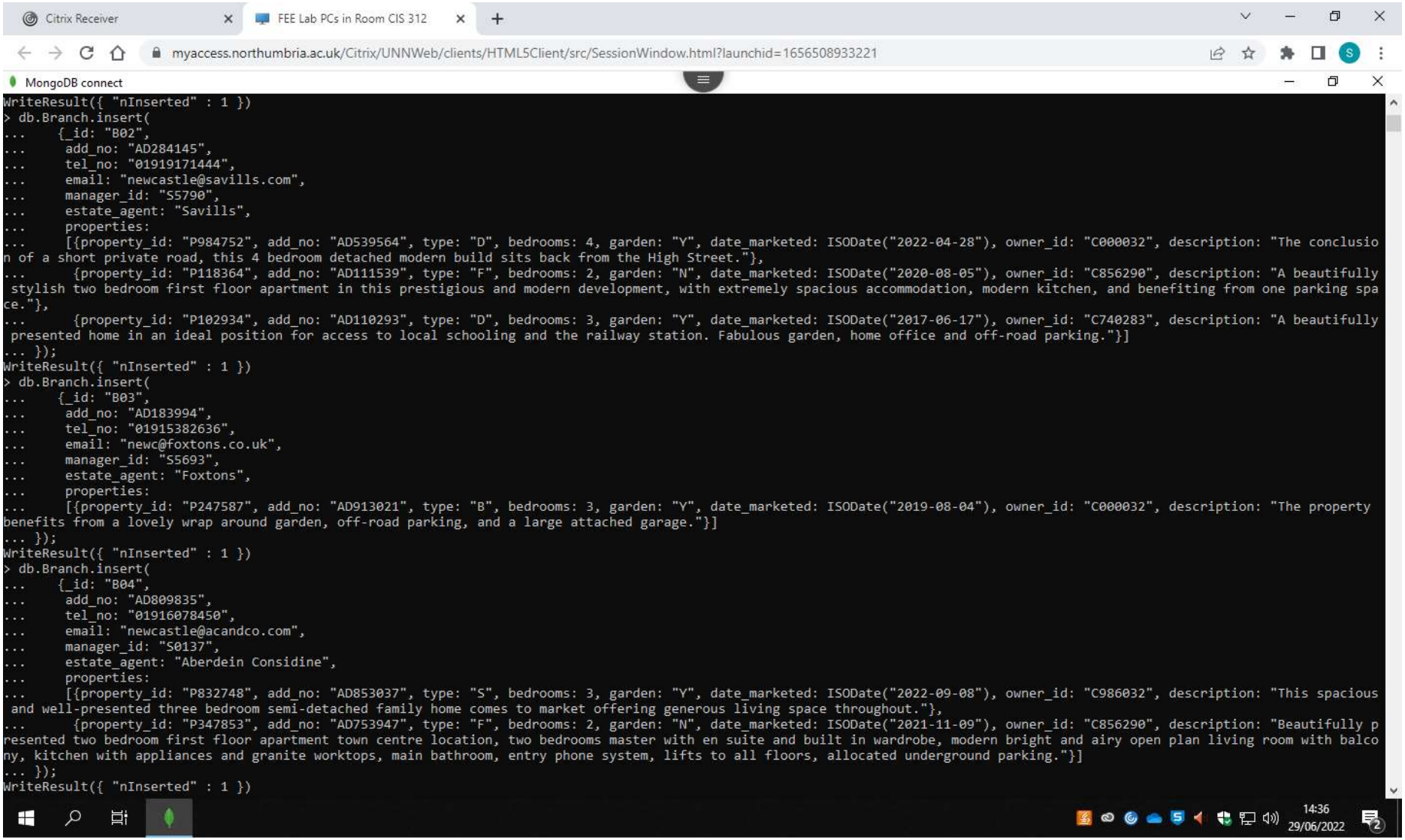
```
// Query to find the address of all properties managed
//by the Savills branch in Newcastle (B02)


db.Branch.aggregate ([
 {$match: {_id: "B02"}},
 {$lookup:
     {
       from: "Address",
       localField: "properties.add_no",
       foreignField: "_id",
```

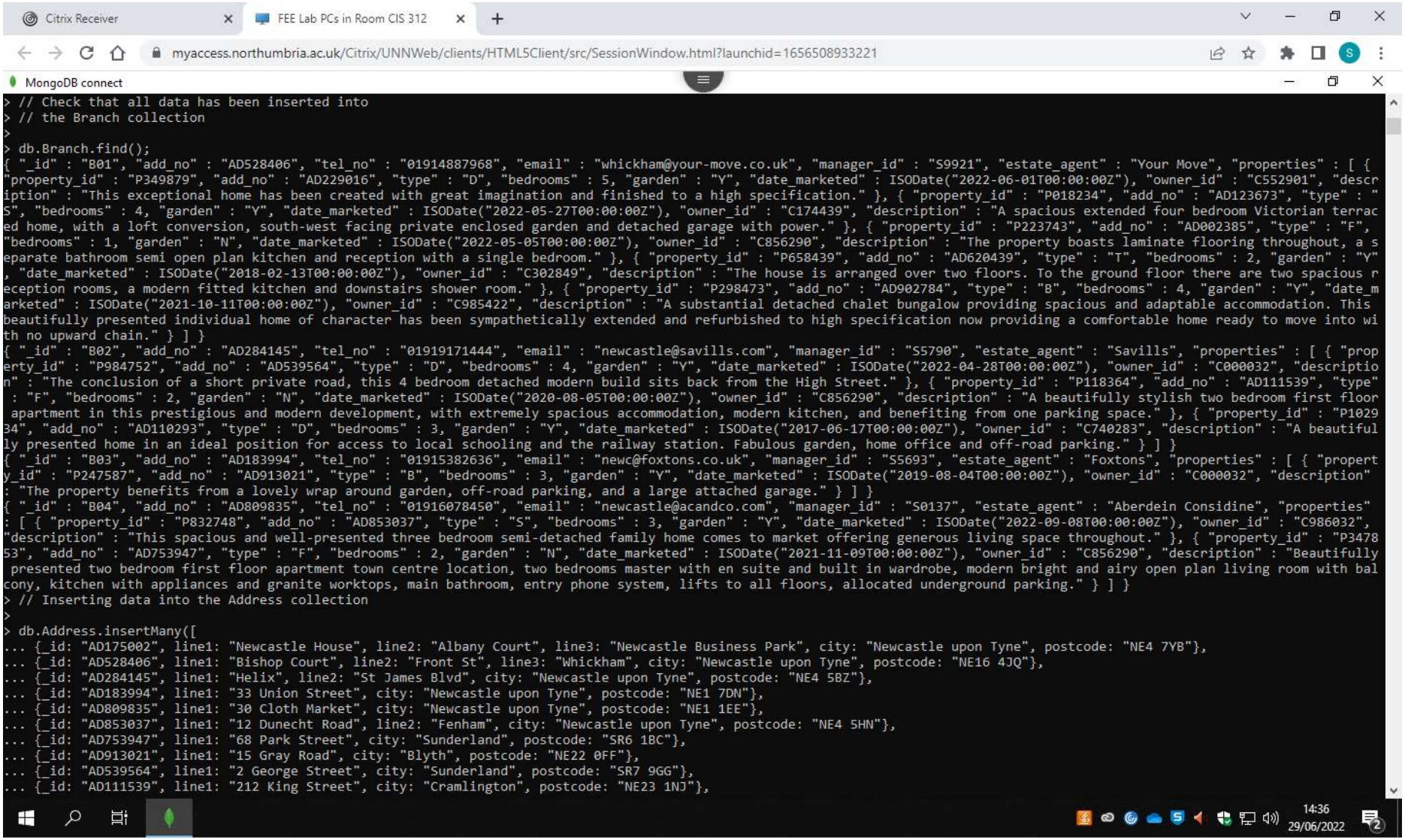
```
        as: "address"
      }
  }
]);
```

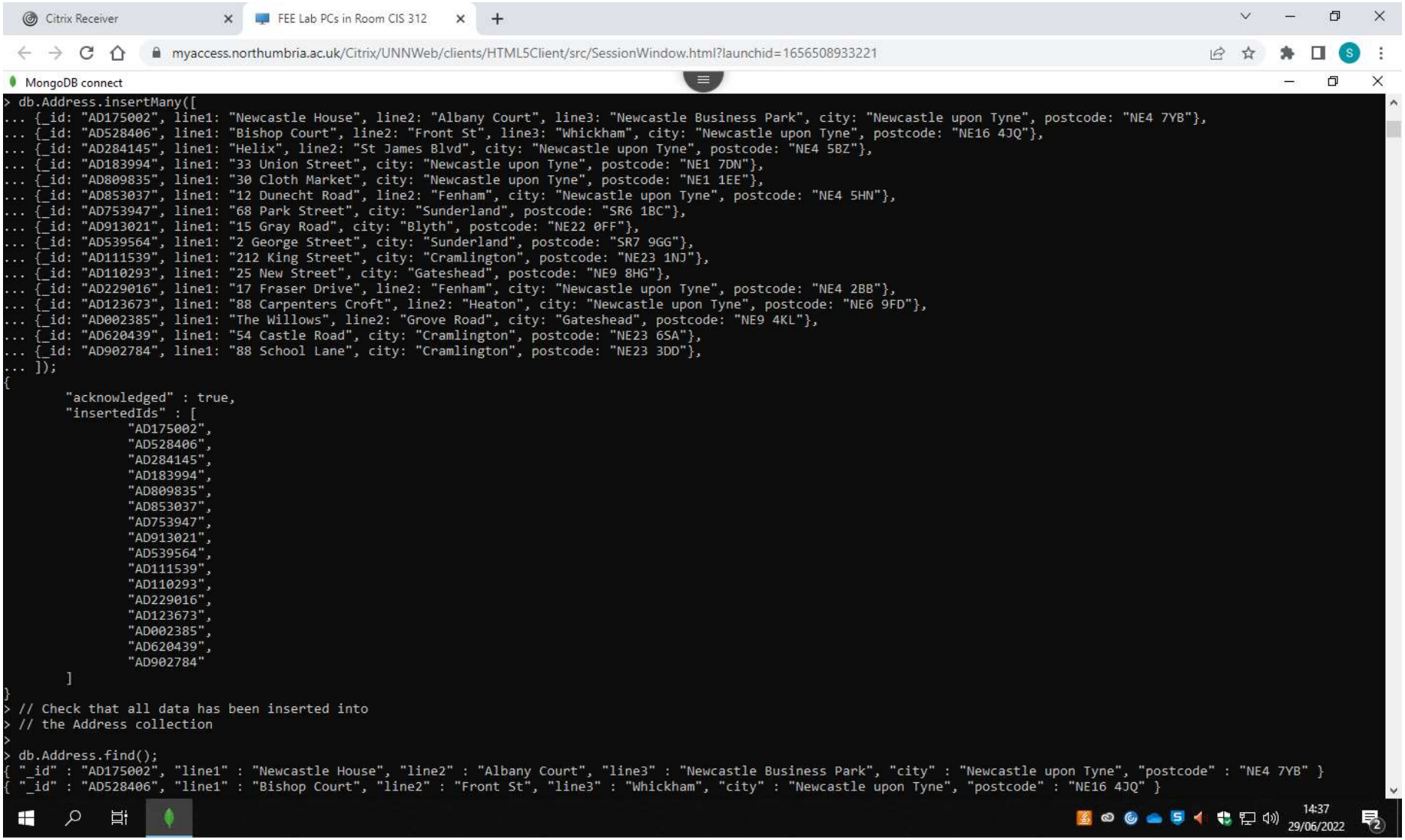
Provide the above MongoDB code's output (e.g., copy & paste of code and output from the MongoDB window or screenshots): Minus 2 marks (if output is missing or deduct partial marks accordingly for partial or inadequate output)

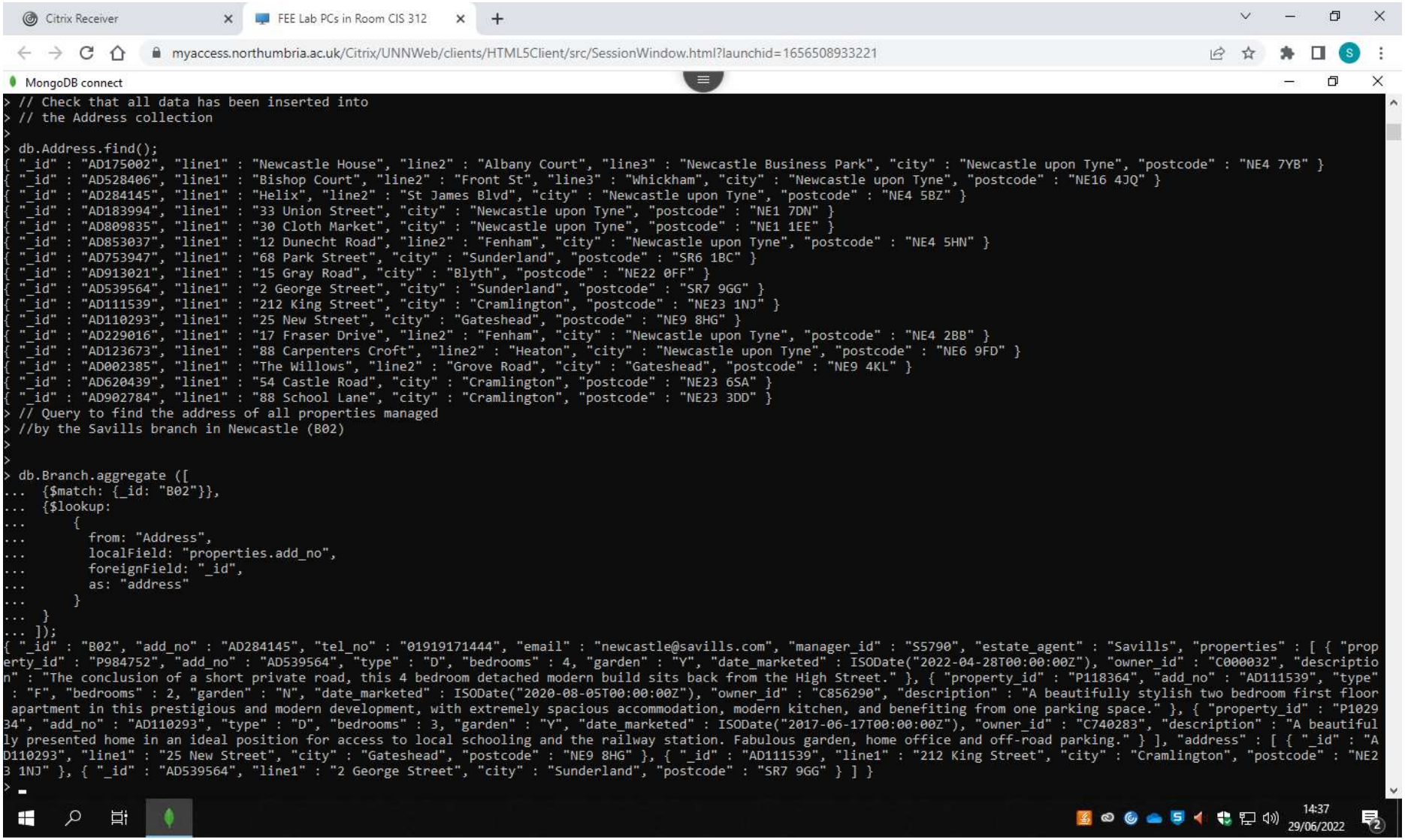
The output is provided in screenshots starting on the next page, which is set to landscape orientation.


Citrix Receiver
FEE Lab PCs in Room CIS 312
myaccess.northumbria.ac.uk/Citrix/UNNWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1656508933221
MongoDB connect
> // Create a collection for the Branch entity
>
> db.createCollection("Branch")
{ "ok" : 1 }
>
>
> // Create a collection for the Address entity
>
> db.createCollection("Address")
{ "ok" : 1 }
> // Inserting data into the Branch collection
> // which includes the embedded Property collection
>
> db.Branch.insert(
... {_id: "B01",
... add_no: "AD528406",
... tel_no: "01914887968",
... email: "whickham@your-move.co.uk",
... manager_id: "S9921",
... estate_agent: "Your Move",
... properties:
... [{property_id: "P349879", add_no: "AD229016", type: "D", bedrooms: 5, garden: "Y", date_marketed: ISODate("2022-06-01"), owner_id: "C552901", description: "This exceptio
nal home has been created with great imagination and finished to a high specification."},
... {property_id: "P018234", add_no: "AD123673", type: "S", bedrooms: 4, garden: "Y", date_marketed: ISODate("2022-05-27"), owner_id: "C174439", description: "A spacious ex
tended four bedroom Victorian terraced home, with a loft conversion, south-west facing private enclosed garden and detached garage with power."},
... {property_id: "P223743", add_no: "AD002385", type: "F", bedrooms: 1, garden: "N", date_marketed: ISODate("2022-05-05"), owner_id: "C856290", description: "The property
boasts laminate flooring throughout, a separate bathroom semi open plan kitchen and reception with a single bedroom."},
... {property_id: "P658439", add_no: "AD620439", type: "T", bedrooms: 2, garden: "Y", date_marketed: ISODate("2018-02-13"), owner_id: "C302849", description: "The house is
arranged over two floors. To the ground floor there are two spacious reception rooms, a modern fitted kitchen and downstairs shower room."},
... {property_id: "P298473", add_no: "AD902784", type: "B", bedrooms: 4, garden: "Y", date_marketed: ISODate("2021-10-11"), owner_id: "C985422", description: "A substantial
detached chalet bungalow providing spacious and adaptable accommodation. This beautifully presented individual home of character has been sympathetically extended and refurbish
ed to high specification now providing a comfortable home ready to move into with no upward chain."}]
... });
WriteResult({ "nInserted" : 1 })
> db.Branch.insert(
... {_id: "B02",
... add_no: "AD284145",
... tel_no: "01919171444",
... email: "newcastle@savills.com",
... manager_id: "S5790",
... estate_agent: "Savills",
... properties:
... [{property_id: "P984752", add_no: "AD539564", type: "D", bedrooms: 4, garden: "Y", date_marketed: ISODate("2022-04-28"), owner_id: "C000032", description: "The conclusio
n of a short private road, this 4 bedroom detached modern build sits back from the High Street."},
... {property_id: "P118364", add_no: "AD111539", type: "F", bedrooms: 2, garden: "N", date_marketed: ISODate("2020-08-05"), owner_id: "C856290", description: "A beautifully
14:35
29/06/2022


Citrix Receiver
FEE Lab PCs in Room CIS 312
myaccess.northumbria.ac.uk/Citrix/UNNWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1656508933221
MongoDB connect
WriteResult({ "nInserted" : 1 })
> db.Branch.insert(
... {_id: "B02",
... add_no: "AD284145",
... tel_no: "01919171444",
... email: "newcastle@savills.com",
... manager_id: "S5790",
... estate_agent: "Savills",
... properties:
... [{property_id: "P984752", add_no: "AD539564", type: "D", bedrooms: 4, garden: "Y", date_marketed: ISODate("2022-04-28"), owner_id: "C000032", description: "The conclusion of a short private road, this 4 bedroom detached modern build sits back from the High Street."},
... {property_id: "P118364", add_no: "AD111539", type: "F", bedrooms: 2, garden: "N", date_marketed: ISODate("2020-08-05"), owner_id: "C856290", description: "A beautifully stylish two bedroom first floor apartment in this prestigious and modern development, with extremely spacious accommodation, modern kitchen, and benefiting from one parking space."},
... {property_id: "P102934", add_no: "AD110293", type: "D", bedrooms: 3, garden: "Y", date_marketed: ISODate("2017-06-17"), owner_id: "C740283", description: "A beautifully presented home in an ideal position for access to local schooling and the railway station. Fabulous garden, home office and off-road parking."}]
... });
WriteResult({ "nInserted" : 1 })
> db.Branch.insert(
... {_id: "B03",
... add_no: "AD183994",
... tel_no: "01915382636",
... email: "newc@foxtons.co.uk",
... manager_id: "S5693",
... estate_agent: "Foxtons",
... properties:
... [{property_id: "P247587", add_no: "AD913021", type: "B", bedrooms: 3, garden: "Y", date_marketed: ISODate("2019-08-04"), owner_id: "C000032", description: "The property benefits from a lovely wrap around garden, off-road parking, and a large attached garage."}]
... });
WriteResult({ "nInserted" : 1 })
> db.Branch.insert(
... {_id: "B04",
... add_no: "AD809835",
... tel_no: "01916078450",
... email: "newcastle@acandco.com",
... manager_id: "S0137",
... estate_agent: "Aberdein Considine",
... properties:
... [{property_id: "P832748", add_no: "AD853037", type: "S", bedrooms: 3, garden: "Y", date_marketed: ISODate("2022-09-08"), owner_id: "C986032", description: "This spacious and well-presented three bedroom semi-detached family home comes to market offering generous living space throughout."},
... {property_id: "P347853", add_no: "AD753947", type: "F", bedrooms: 2, garden: "N", date_marketed: ISODate("2021-11-09"), owner_id: "C856290", description: "Beautifully presented two bedroom first floor apartment town centre location, two bedrooms master with en suite and built in wardrobe, modern bright and airy open plan living room with balcony, kitchen with appliances and granite worktops, main bathroom, entry phone system, lifts to all floors, allocated underground parking."}]
... });
WriteResult({ "nInserted" : 1 })
14:36
29/06/2022


Citrix Receiver
FEE Lab PCs in Room CIS 312
myaccess.northumbria.ac.uk/Citrix/UNNWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1656508933221
MongoDB connect
> // Check that all data has been inserted into
> // the Branch collection
>
> db.Branch.find();
{ "_id" : "B01", "add_no" : "AD528406", "tel_no" : "01914887968", "email" : "whickham@your-move.co.uk", "manager_id" : "S9921", "estate_agent" : "Your Move", "properties" : [{ "property_id" : "P349879", "add_no" : "AD229016", "type" : "D", "bedrooms" : 5, "garden" : "Y", "date_marketed" : ISODate("2022-06-01T00:00:00Z"), "owner_id" : "C552901", "description" : "This exceptional home has been created with great imagination and finished to a high specification." }, { "property_id" : "P018234", "add_no" : "AD123673", "type" : "S", "bedrooms" : 4, "garden" : "Y", "date_marketed" : ISODate("2022-05-27T00:00:00Z"), "owner_id" : "C174439", "description" : "A spacious extended four bedroom Victorian terraced home, with a loft conversion, south-west facing private enclosed garden and detached garage with power." }, { "property_id" : "P223743", "add_no" : "AD002385", "type" : "F", "bedrooms" : 1, "garden" : "N", "date_marketed" : ISODate("2022-05-05T00:00:00Z"), "owner_id" : "C856290", "description" : "The property boasts laminate flooring throughout, a separate bathroom semi open plan kitchen and reception with a single bedroom." }, { "property_id" : "P658439", "add_no" : "AD620439", "type" : "T", "bedrooms" : 2, "garden" : "Y", "date_marketed" : ISODate("2018-02-13T00:00:00Z"), "owner_id" : "C302849", "description" : "The house is arranged over two floors. To the ground floor there are two spacious reception rooms, a modern fitted kitchen and downstairs shower room." }, { "property_id" : "P298473", "add_no" : "AD902784", "type" : "B", "bedrooms" : 4, "garden" : "Y", "date_marketed" : ISODate("2021-10-11T00:00:00Z"), "owner_id" : "C985422", "description" : "A substantial detached chalet bungalow providing spacious and adaptable accommodation. This beautifully presented individual home of character has been sympathetically extended and refurbished to high specification now providing a comfortable home ready to move into with no upward chain." }] }
{ "_id" : "B02", "add_no" : "AD284145", "tel_no" : "01919171444", "email" : "newcastle@savills.com", "manager_id" : "S5790", "estate_agent" : "Savills", "properties" : [{ "property_id" : "P984752", "add_no" : "AD539564", "type" : "D", "bedrooms" : 4, "garden" : "Y", "date_marketed" : ISODate("2022-04-28T00:00:00Z"), "owner_id" : "C000032", "description" : "The conclusion of a short private road, this 4 bedroom detached modern build sits back from the High Street." }, { "property_id" : "P118364", "add_no" : "AD111539", "type" : "F", "bedrooms" : 2, "garden" : "N", "date_marketed" : ISODate("2020-08-05T00:00:00Z"), "owner_id" : "C856290", "description" : "A beautifully stylish two bedroom first floor apartment in this prestigious and modern development, with extremely spacious accommodation, modern kitchen, and benefiting from one parking space." }, { "property_id" : "P102934", "add_no" : "AD110293", "type" : "D", "bedrooms" : 3, "garden" : "Y", "date_marketed" : ISODate("2017-06-17T00:00:00Z"), "owner_id" : "C740283", "description" : "A beautifully presented home in an ideal position for access to local schooling and the railway station. Fabulous garden, home office and off-road parking." }] }
{ "_id" : "B03", "add_no" : "AD183994", "tel_no" : "01915382636", "email" : "newc@foxtons.co.uk", "manager_id" : "S5693", "estate_agent" : "Foxtons", "properties" : [{ "property_id" : "P247587", "add_no" : "AD913021", "type" : "B", "bedrooms" : 3, "garden" : "Y", "date_marketed" : ISODate("2019-08-04T00:00:00Z"), "owner_id" : "C000032", "description" : "The property benefits from a lovely wrap around garden, off-road parking, and a large attached garage." }] }
{ "_id" : "B04", "add_no" : "AD809835", "tel_no" : "01916078450", "email" : "newcastle@acandco.com", "manager_id" : "S0137", "estate_agent" : "Aberdein Considine", "properties" : [{ "property_id" : "P832748", "add_no" : "AD853037", "type" : "S", "bedrooms" : 3, "garden" : "Y", "date_marketed" : ISODate("2022-09-08T00:00:00Z"), "owner_id" : "C986032", "description" : "This spacious and well-presented three bedroom semi-detached family home comes to market offering generous living space throughout." }, { "property_id" : "P347853", "add_no" : "AD753947", "type" : "F", "bedrooms" : 2, "garden" : "N", "date_marketed" : ISODate("2021-11-09T00:00:00Z"), "owner_id" : "C856290", "description" : "Beautifully presented two bedroom first floor apartment town centre location, two bedrooms master with en suite and built in wardrobe, modern bright and airy open plan living room with balcony, kitchen with appliances and granite worktops, main bathroom, entry phone system, lifts to all floors, allocated underground parking." }] }
> // Inserting data into the Address collection
>
> db.Address.insertMany([
... {_id: "AD175002", line1: "Newcastle House", line2: "Albany Court", line3: "Newcastle Business Park", city: "Newcastle upon Tyne", postcode: "NE4 7YB"},
... {_id: "AD528406", line1: "Bishop Court", line2: "Front St", line3: "Whickham", city: "Newcastle upon Tyne", postcode: "NE16 4JQ"},
... {_id: "AD284145", line1: "Helix", line2: "St James Blvd", city: "Newcastle upon Tyne", postcode: "NE4 5BZ"},
... {_id: "AD183994", line1: "33 Union Street", city: "Newcastle upon Tyne", postcode: "NE1 7DN"},
... {_id: "AD809835", line1: "30 Cloth Market", city: "Newcastle upon Tyne", postcode: "NE1 1EE"},
... {_id: "AD853037", line1: "12 Dunecht Road", line2: "Fenham", city: "Newcastle upon Tyne", postcode: "NE4 5HN"},
... {_id: "AD753947", line1: "68 Park Street", city: "Sunderland", postcode: "SR6 1BC"},
... {_id: "AD913021", line1: "15 Gray Road", city: "Blyth", postcode: "NE22 0FF"},
... {_id: "AD539564", line1: "2 George Street", city: "Sunderland", postcode: "SR7 9GG"},
... {_id: "AD111539", line1: "212 King Street", city: "Cramlington", postcode: "NE23 1NJ"},
14:36
29/06/2022


Citrix Receiver
FEE Lab PCs in Room CIS 312
myaccess.northumbria.ac.uk/Citrix/UNNWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1656508933221
MongoDB connect
> db.Address.insertMany([
... {_id: "AD175002", line1: "Newcastle House", line2: "Albany Court", line3: "Newcastle Business Park", city: "Newcastle upon Tyne", postcode: "NE4 7YB"},
... {_id: "AD528406", line1: "Bishop Court", line2: "Front St", line3: "Whickham", city: "Newcastle upon Tyne", postcode: "NE16 4JQ"},
... {_id: "AD284145", line1: "Helix", line2: "St James Blvd", city: "Newcastle upon Tyne", postcode: "NE4 5BZ"},
... {_id: "AD183994", line1: "33 Union Street", city: "Newcastle upon Tyne", postcode: "NE1 7DN"},
... {_id: "AD809835", line1: "30 Cloth Market", city: "Newcastle upon Tyne", postcode: "NE1 1EE"},
... {_id: "AD853037", line1: "12 Dunecht Road", line2: "Fenham", city: "Newcastle upon Tyne", postcode: "NE4 5HN"},
... {_id: "AD753947", line1: "68 Park Street", city: "Sunderland", postcode: "SR6 1BC"},
... {_id: "AD913021", line1: "15 Gray Road", city: "Blyth", postcode: "NE22 0FF"},
... {_id: "AD539564", line1: "2 George Street", city: "Sunderland", postcode: "SR7 9GG"},
... {_id: "AD111539", line1: "212 King Street", city: "Cramlington", postcode: "NE23 1NJ"},
... {_id: "AD110293", line1: "25 New Street", city: "Gateshead", postcode: "NE9 8HG"},
... {_id: "AD229016", line1: "17 Fraser Drive", line2: "Fenham", city: "Newcastle upon Tyne", postcode: "NE4 2BB"},
... {_id: "AD123673", line1: "88 Carpenters Croft", line2: "Heaton", city: "Newcastle upon Tyne", postcode: "NE6 9FD"},
... {_id: "AD002385", line1: "The Willows", line2: "Grove Road", city: "Gateshead", postcode: "NE9 4KL"},
... {_id: "AD620439", line1: "54 Castle Road", city: "Cramlington", postcode: "NE23 6SA"},
... {_id: "AD902784", line1: "88 School Lane", city: "Cramlington", postcode: "NE23 3DD"},
...]);
{
"acknowledged" : true,
"insertedIds" : [
"AD175002",
"AD528406",
"AD284145",
"AD183994",
"AD809835",
"AD853037",
"AD753947",
"AD913021",
"AD539564",
"AD111539",
"AD110293",
"AD229016",
"AD123673",
"AD002385",
"AD620439",
"AD902784"
]
}
> // Check that all data has been inserted into
> // the Address collection
>
> db.Address.find();
{ "_id" : "AD175002", "line1" : "Newcastle House", "line2" : "Albany Court", "line3" : "Newcastle Business Park", "city" : "Newcastle upon Tyne", "postcode" : "NE4 7YB" }
{ "_id" : "AD528406", "line1" : "Bishop Court", "line2" : "Front St", "line3" : "Whickham", "city" : "Newcastle upon Tyne", "postcode" : "NE16 4JQ" }
14:37
29/06/2022


Citrix Receiver
FEE Lab PCs in Room CIS 312
myaccess.northumbria.ac.uk/Citrix/UNNWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1656508933221
MongoDB connect
> // Check that all data has been inserted into
> // the Address collection
>
> db.Address.find();
{ "_id" : "AD175002", "line1" : "Newcastle House", "line2" : "Albany Court", "line3" : "Newcastle Business Park", "city" : "Newcastle upon Tyne", "postcode" : "NE4 7YB" }
{ "_id" : "AD528406", "line1" : "Bishop Court", "line2" : "Front St", "line3" : "Whickham", "city" : "Newcastle upon Tyne", "postcode" : "NE16 4JQ" }
{ "_id" : "AD284145", "line1" : "Helix", "line2" : "St James Blvd", "city" : "Newcastle upon Tyne", "postcode" : "NE4 5BZ" }
{ "_id" : "AD183994", "line1" : "33 Union Street", "city" : "Newcastle upon Tyne", "postcode" : "NE1 7DN" }
{ "_id" : "AD809835", "line1" : "30 Cloth Market", "city" : "Newcastle upon Tyne", "postcode" : "NE1 1EE" }
{ "_id" : "AD853037", "line1" : "12 Dunecht Road", "line2" : "Fenham", "city" : "Newcastle upon Tyne", "postcode" : "NE4 5HN" }
{ "_id" : "AD753947", "line1" : "68 Park Street", "city" : "Sunderland", "postcode" : "SR6 1BC" }
{ "_id" : "AD913021", "line1" : "15 Gray Road", "city" : "Blyth", "postcode" : "NE22 0FF" }
{ "_id" : "AD539564", "line1" : "2 George Street", "city" : "Sunderland", "postcode" : "SR7 9GG" }
{ "_id" : "AD111539", "line1" : "212 King Street", "city" : "Cramlington", "postcode" : "NE23 1NJ" }
{ "_id" : "AD110293", "line1" : "25 New Street", "city" : "Gateshead", "postcode" : "NE9 8HG" }
{ "_id" : "AD229016", "line1" : "17 Fraser Drive", "line2" : "Fenham", "city" : "Newcastle upon Tyne", "postcode" : "NE4 2BB" }
{ "_id" : "AD123673", "line1" : "88 Carpenters Croft", "line2" : "Heaton", "city" : "Newcastle upon Tyne", "postcode" : "NE6 9FD" }
{ "_id" : "AD002385", "line1" : "The Willows", "line2" : "Grove Road", "city" : "Gateshead", "postcode" : "NE9 4KL" }
{ "_id" : "AD620439", "line1" : "54 Castle Road", "city" : "Cramlington", "postcode" : "NE23 6SA" }
{ "_id" : "AD902784", "line1" : "88 School Lane", "city" : "Cramlington", "postcode" : "NE23 3DD" }
> // Query to find the address of all properties managed
> //by the Savills branch in Newcastle (B02)
>
>
> db.Branch.aggregate ([
... {$match: {_id: "B02"}},
... {$lookup:
...   {
...   from: "Address",
...   localField: "properties.add_no",
...   foreignField: "_id",
...   as: "address"
...   }
... }
...]);
{ "_id" : "B02", "add_no" : "AD284145", "tel_no" : "01919171444", "email" : "newcastle@savills.com", "manager_id" : "S5790", "estate_agent" : "Savills", "properties" : [{ "property_id" : "P984752", "add_no" : "AD539564", "type" : "D", "bedrooms" : 4, "garden" : "Y", "date_marketed" : ISODate("2022-04-28T00:00:00Z"), "owner_id" : "C000032", "description" : "The conclusion of a short private road, this 4 bedroom detached modern build sits back from the High Street." }, { "property_id" : "P118364", "add_no" : "AD111539", "type" : "F", "bedrooms" : 2, "garden" : "N", "date_marketed" : ISODate("2020-08-05T00:00:00Z"), "owner_id" : "C856290", "description" : "A beautifully stylish two bedroom first floor apartment in this prestigious and modern development, with extremely spacious accommodation, modern kitchen, and benefiting from one parking space." }, { "property_id" : "P102934", "add_no" : "AD110293", "type" : "D", "bedrooms" : 3, "garden" : "Y", "date_marketed" : ISODate("2017-06-17T00:00:00Z"), "owner_id" : "C740283", "description" : "A beautifully presented home in an ideal position for access to local schooling and the railway station. Fabulous garden, home office and off-road parking." }], "address" : [{ "_id" : "AD110293", "line1" : "25 New Street", "city" : "Gateshead", "postcode" : "NE9 8HG" }, { "_id" : "AD111539", "line1" : "212 King Street", "city" : "Cramlington", "postcode" : "NE23 1NJ" }, { "_id" : "AD539564", "line1" : "2 George Street", "city" : "Sunderland", "postcode" : "SR7 9GG" }] }
>
14:37
29/06/2022

Part 4 (20 Marks)

This part is based on the Universal Finance Credit Card (UFCrC) company's customers defaulting scenario as described in the Appendix 2. The main purpose of this part is correctly predicting if credit card customers will default on their due payments. You are required to perform the following tasks:

1. Explore the dataset and justify whether UFCrC's problem belongs to predictive or descriptive data mining models. Choose which data mining task (e.g., classification, association rules, clustering, regression, etc) will be used to produce data mining models for the UFCrC's scenario.

(2 marks)

Answer

Descriptive models are concerned with identifying patterns and relationships in a dataset and generally involve unsupervised learning, where "the machine has not been told what it is to learn" (Defence Science and Technology Laboratory, 2019, p.23). Conversely, predictive models are trained to forecast a specific variable or value using supervised learning. They also require labelled data to make their predictions (Defence Science and Technology Laboratory, 2019). In this instance, the dataset contains labels and UFCrC have been explicit about what they want the model to learn, which can be predicted using the DEFAULTPAYNXTMNT binary attribute. Therefore, the problem that UFCrC are trying to solve is best served by a predictive data mining model.

Regression, prediction or classification can be used as the foundation to build a predictive model. Regression uses and predicts numerical values, so it is not appropriate in this scenario as the aim is to establish whether a customer belongs to a defined category. Likewise, prediction forecasts values whereas classification assigns objects or items to classes or categories "based on combinations of parameters" (Elmasri and Navathe, 2017, p.1102). In addition, classification algorithms are often used to determine credit risks (Çığşar and Ünal, 2019) and they are accustomed to handling binary classifiers (Oracle, 2010), such as whether a

customer will default on a credit card payment or not. As such, classification will be used to produce the data mining model in this scenario.

---

2. Prepare and setup your data/views/tables for accessing the shared `CreditCardsV2` dataset, which also includes splitting the dataset for building, testing and applying the data mining models.

(3 marks)

Answer

Including more data in the training or build set can potentially improve the accuracy of a model and prevent issues like overfitting (IBM, 2021). Consequently, different sample sizes were utilised to find the most accurate model, using the three 'view collections' in the table below.

View Collection No	Build Data Cases	Test Data Cases	Apply Data Cases
1	10,000	10,000	10,000
2	15,000	9,000	6,000
3	21,000	6,000	3,000

The code below was used to setup the views. Please note, the spool output is shown.

```
W21056374 > @D:\UFCrC_Views.sql
W21056374 >
W21056374 > REM Split the CreditCardsV2 dataset into 9 views and three view
collections
W21056374 > REM for use with different build, test and apply
subsets/samples. The table
W21056374 > REM below outlines how the 30,000 records in the base dataset
are divided amongst
W21056374 > REM the collections.
W21056374 >
```

```
W21056374 > REM     Collection No   Build Data      Test Data         Apply
Data
W21056374 > REM    1          10,000            10,000            10,000
W21056374 > REM    2          15,000             9,000             6,000
W21056374 > REM    3          21,000             6,000             3,000
W21056374 >
W21056374 >
W21056374 > REM Create first view collection
W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_build_data_col_1 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK <= 10000;

View created.

W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_test_data_col_1 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK > 10000 AND RNK <= 20000;

View created.

W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_apply_data_col_1 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK > 20000 AND RNK <= 30000;

View created.

W21056374 >
W21056374 >
W21056374 > REM Create second view collection
```

```
W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_build_data_col_2 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK <= 15000;

View created.

W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_test_data_col_2 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK > 15000 AND RNK <= 24000;

View created.

W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_apply_data_col_2 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK > 24000 AND RNK <= 30000;

View created.

W21056374 >
W21056374 >
W21056374 > REM Create third view collection
W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_build_data_col_3 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK <= 21000;
```

```
View created.

W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_test_data_col_3 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK > 21000 AND RNK <= 27000;

View created.

W21056374 >
W21056374 > CREATE OR REPLACE VIEW ufcrc_apply_data_col_3 as
  2  SELECT *
  3  FROM
  4  (SELECT c.*, row_number() over (order by c.card) as RNK
  5   FROM creditcardsv2 c)
  6  WHERE RNK > 27000 AND RNK <= 30000;

View created.

W21056374 >
W21056374 >
W21056374 > spool off
```

---

3. Develop at least TWO models using suitable algorithms for performing your chosen data mining task for the UFCrC's dataset.

(8 marks)

Answer

Six models were developed in total: three using the random forest algorithm and three employing naive bayes. In each algorithm group, the three view collections were used to check how the models would perform with different sample sizes.

When the models were being developed, the number of trees used by the random forest algorithm were increased to ascertain what impact this would have on performance. It was quickly established that the accuracy gains were minimal, yet the time taken to build the model increased significantly. As a result, the idea of adjusting the number of trees to improve accuracy was abandoned and no code from this experiment was retained.

The following code was used to develop the six models. Please note, the spool output is shown.

```
W21056374 > @D:\UFCrC_Models.sql
W21056374 >
W21056374 > REM Create settings table for the Random Forest (RF) model and
populate it.
W21056374 >
W21056374 > CREATE TABLE ufcrc_rf_model_settings (
  2    setting_name  VARCHAR2(50),
  3    setting_value VARCHAR2(50));

Table created.

W21056374 >
W21056374 > BEGIN
  2    INSERT INTO ufcrc_rf_model_settings (setting_name, setting_value)
  3    VALUES (dbms_data_mining.algo_name,
dbms_data_mining.algo_random_forest);
  4    INSERT INTO ufcrc_rf_model_settings (setting_name, setting_value)
  5    VALUES (dbms_data_mining.prep_auto,dbms_data_mining.prep_auto_on);
  6    COMMIT;
  7  END;
  8  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 >
W21056374 > REM Now create the first RF model using the first view
collection.
W21056374 >
```

```
W21056374 > BEGIN
  2    DBMS_DATA_MINING.CREATE_MODEL(
  3    model_name      => 'ufcrc_rf_model_1',
  4    mining_function     => dbms_data_mining.classification,
  5    data_table_name     => 'ufcrc_build_data_col_1',
  6    case_id_column_name => 'card',
  7    target_column_name  => 'defaultpaynxtmnt',
  8    settings_table_name => 'ufcrc_rf_model_settings');
  9  END;
 10  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 >
W21056374 > REM Test the first RF model.
W21056374 >
W21056374 > SELECT defaultpaynxtmnt AS actual_target_value,
  2       PREDICTION(ufcrc_rf_model_1 USING *) AS predicted_target_value,
  3       COUNT(*) AS total_value
  4  FROM ufcrc_test_data_col_1
  5  GROUP BY defaultpaynxtmnt, PREDICTION(ufcrc_rf_model_1 USING *)
  6  ORDER BY 1,2;

ACTUAL_TARGET_VALUE PREDICTED_TARGET_VALUE TOTAL_VALUE
------------------- ---------------------- -----------
                  0                      0        7267
                  0                      1         482
                  1                      0        1469
                  1                      1         782

W21056374 >
W21056374 >
W21056374 > REM Display the accuracy of the first RF model.
W21056374 >
W21056374 > COLUMN ACCURACY FORMAT 99.99
W21056374 > SELECT ROUND(SUM(correct)/COUNT(*),4)*100 AS accuracy
  2    FROM (SELECT DECODE(defaultpaynxtmnt,
  3            PREDICTION(ufcrc_rf_model_1 USING *), 1, 0) AS correct
  4          FROM ufcrc_test_data_col_1);
```

```
ACCURACY
--------
   80.49

W21056374 >
W21056374 >
W21056374 > REM Now create the second RF model using the second view
collection.
W21056374 >
W21056374 > BEGIN
  2    DBMS_DATA_MINING.CREATE_MODEL(
  3    model_name      => 'ufcrc_rf_model_2',
  4    mining_function     => dbms_data_mining.classification,
  5    data_table_name     => 'ufcrc_build_data_col_2',
  6    case_id_column_name => 'card',
  7    target_column_name  => 'defaultpaynxtmnt',
  8    settings_table_name => 'ufcrc_rf_model_settings');
  9  END;
 10  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 >
W21056374 > REM Test the second RF model.
W21056374 >
W21056374 > SELECT defaultpaynxtmnt AS actual_target_value,
  2       PREDICTION(ufcrc_rf_model_2 USING *) AS predicted_target_value,
  3       COUNT(*) AS total_value
  4  FROM ufcrc_test_data_col_2
  5  GROUP BY defaultpaynxtmnt, PREDICTION(ufcrc_rf_model_2 USING *)
  6  ORDER BY 1,2;

ACTUAL_TARGET_VALUE PREDICTED_TARGET_VALUE TOTAL_VALUE
------------------- ---------------------- -----------
                  0                      0        6751
                  0                      1         325
                  1                      0        1285
                  1                      1         639

W21056374 >
```

```
W21056374 >
W21056374 > REM Display the accuracy of the second RF model.
W21056374 >
W21056374 > COLUMN ACCURACY FORMAT 99.99
W21056374 > SELECT ROUND(SUM(correct)/COUNT(*),4)*100 AS accuracy
  2    FROM (SELECT DECODE(defaultpaynxtmnt,
  3            PREDICTION(ufcrc_rf_model_2 USING *), 1, 0) AS correct
  4          FROM ufcrc_test_data_col_2);

ACCURACY
--------
   82.11

W21056374 >
W21056374 >
W21056374 > REM Now create the third RF model using the third view
collection.
W21056374 >
W21056374 > BEGIN
  2    DBMS_DATA_MINING.CREATE_MODEL(
  3    model_name      => 'ufcrc_rf_model_3',
  4    mining_function     => dbms_data_mining.classification,
  5    data_table_name     => 'ufcrc_build_data_col_3',
  6    case_id_column_name => 'card',
  7    target_column_name  => 'defaultpaynxtmnt',
  8    settings_table_name => 'ufcrc_rf_model_settings');
  9  END;
 10  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 >
W21056374 > REM Test the third RF model.
W21056374 >
W21056374 > SELECT defaultpaynxtmnt AS actual_target_value,
  2       PREDICTION(ufcrc_rf_model_3 USING *) AS predicted_target_value,
  3       COUNT(*) AS total_value
  4  FROM ufcrc_test_data_col_3
  5  GROUP BY defaultpaynxtmnt, PREDICTION(ufcrc_rf_model_3 USING *)
  6  ORDER BY 1,2;
```

```
ACTUAL_TARGET_VALUE PREDICTED_TARGET_VALUE TOTAL_VALUE
------------------- ---------------------- -----------
                  0                      0        4662
                  0                      1         171
                  1                      0         765
                  1                      1         402

W21056374 >
W21056374 >
W21056374 > REM Display the accuracy of the third RF model.
W21056374 >
W21056374 > COLUMN ACCURACY FORMAT 99.99
W21056374 > SELECT ROUND(SUM(correct)/COUNT(*),4)*100 AS accuracy
  2    FROM (SELECT DECODE(defaultpaynxtmnt,
  3            PREDICTION(ufcrc_rf_model_3 USING *), 1, 0) AS correct
  4          FROM ufcrc_test_data_col_3);

ACCURACY
--------
   84.40

W21056374 >
W21056374 >
W21056374 >
W21056374 > REM Create settings table for the Naive Bayes (NB) model and
populate it.
W21056374 > REM Note the algorithm doesn't need to be specified as it is
the default.
W21056374 >
W21056374 > CREATE TABLE ufcrc_nb_model_settings (
  2    setting_name  VARCHAR2(50),
  3    setting_value VARCHAR2(50));

Table created.

W21056374 >
W21056374 > BEGIN
  2    INSERT INTO ufcrc_nb_model_settings (setting_name, setting_value)
  3    VALUES (dbms_data_mining.prep_auto,dbms_data_mining.prep_auto_on);
  4    COMMIT;
```

```
  5  END;
  6  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 >
W21056374 > REM Now create the first NB model using the first view
collection.
W21056374 >
W21056374 > BEGIN
  2    DBMS_DATA_MINING.CREATE_MODEL(
  3    model_name      => 'ufcrc_nb_model_1',
  4    mining_function     => dbms_data_mining.classification,
  5    data_table_name     => 'ufcrc_build_data_col_1',
  6    case_id_column_name => 'card',
  7    target_column_name  => 'defaultpaynxtmnt',
  8    settings_table_name => 'ufcrc_nb_model_settings');
  9  END;
 10  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 >
W21056374 > REM Test the first NB model.
W21056374 >
W21056374 > SELECT defaultpaynxtmnt AS actual_target_value,
  2       PREDICTION(ufcrc_nb_model_1 USING *) AS predicted_target_value,
  3       COUNT(*) AS total_value
  4  FROM ufcrc_test_data_col_1
  5  GROUP BY defaultpaynxtmnt, PREDICTION(ufcrc_nb_model_1 USING *)
  6  ORDER BY 1,2;

ACTUAL_TARGET_VALUE PREDICTED_TARGET_VALUE TOTAL_VALUE
------------------- ---------------------- -----------
                  0                      0        6813
                  0                      1         936
                  1                      0        1232
                  1                      1        1019
```

```
W21056374 >
W21056374 >
W21056374 > REM Display the accuracy of the first NB model.
W21056374 >
W21056374 > COLUMN ACCURACY FORMAT 99.99
W21056374 > SELECT ROUND(SUM(correct)/COUNT(*),4)*100 AS accuracy
  2    FROM (SELECT DECODE(defaultpaynxtmnt,
  3            PREDICTION(ufcrc_nb_model_1 USING *), 1, 0) AS correct
  4          FROM ufcrc_test_data_col_1);

ACCURACY
--------
   78.32

W21056374 >
W21056374 >
W21056374 > REM Now create the second NB model using the second view
collection.
W21056374 >
W21056374 > BEGIN
  2    DBMS_DATA_MINING.CREATE_MODEL(
  3    model_name      => 'ufcrc_nb_model_2',
  4    mining_function     => dbms_data_mining.classification,
  5    data_table_name     => 'ufcrc_build_data_col_2',
  6    case_id_column_name => 'card',
  7    target_column_name  => 'defaultpaynxtmnt',
  8    settings_table_name => 'ufcrc_nb_model_settings');
  9  END;
 10  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 >
W21056374 > REM Test the second NB model.
W21056374 >
W21056374 > SELECT defaultpaynxtmnt AS actual_target_value,
  2       PREDICTION(ufcrc_nb_model_2 USING *) AS predicted_target_value,
  3       COUNT(*) AS total_value
  4  FROM ufcrc_test_data_col_2
  5  GROUP BY defaultpaynxtmnt, PREDICTION(ufcrc_nb_model_2 USING *)
```

```
  6  ORDER BY 1,2;

ACTUAL_TARGET_VALUE PREDICTED_TARGET_VALUE TOTAL_VALUE
------------------- ---------------------- -----------
                  0                      0        6233
                  0                      1         843
                  1                      0        1029
                  1                      1         895

W21056374 >
W21056374 >
W21056374 > REM Display the accuracy of the second NB model.
W21056374 >
W21056374 > COLUMN ACCURACY FORMAT 99.99
W21056374 > SELECT ROUND(SUM(correct)/COUNT(*),4)*100 AS accuracy
  2    FROM (SELECT DECODE(defaultpaynxtmnt,
  3            PREDICTION(ufcrc_nb_model_2 USING *), 1, 0) AS correct
  4          FROM ufcrc_test_data_col_2);

ACCURACY
--------
   79.20

W21056374 >
W21056374 >
W21056374 > REM Now create the third NB model using the third view
collection.
W21056374 >
W21056374 > BEGIN
  2    DBMS_DATA_MINING.CREATE_MODEL(
  3    model_name      => 'ufcrc_nb_model_3',
  4    mining_function     => dbms_data_mining.classification,
  5    data_table_name     => 'ufcrc_build_data_col_3',
  6    case_id_column_name => 'card',
  7    target_column_name  => 'defaultpaynxtmnt',
  8    settings_table_name => 'ufcrc_nb_model_settings');
  9  END;
 10  /

PL/SQL procedure successfully completed.
```

```
W21056374 >
W21056374 >
W21056374 > REM Test the third NB model.
W21056374 >
W21056374 > SELECT defaultpaynxtmnt AS actual_target_value,
  2       PREDICTION(ufcrc_nb_model_3 USING *) AS predicted_target_value,
  3       COUNT(*) AS total_value
  4  FROM ufcrc_test_data_col_3
  5  GROUP BY defaultpaynxtmnt, PREDICTION(ufcrc_nb_model_3 USING *)
  6  ORDER BY 1,2;

ACTUAL_TARGET_VALUE PREDICTED_TARGET_VALUE TOTAL_VALUE
------------------- ---------------------- -----------
                  0                      0        4350
                  0                      1         483
                  1                      0         625
                  1                      1         542

W21056374 >
W21056374 >
W21056374 > REM Display the accuracy of the first NB model.
W21056374 >
W21056374 > COLUMN ACCURACY FORMAT 99.99
W21056374 > SELECT ROUND(SUM(correct)/COUNT(*),4)*100 AS accuracy
  2    FROM (SELECT DECODE(defaultpaynxtmnt,
  3            PREDICTION(ufcrc_nb_model_3 USING *), 1, 0) AS correct
  4          FROM ufcrc_test_data_col_3);

ACCURACY
--------
   81.53

W21056374 >
W21056374 >
W21056374 >
W21056374 >
W21056374 >
W21056374 >
W21056374 >
W21056374 >
W21056374 >
```

```
W21056374 >
W21056374 >
W21056374 >
W21056374 >
W21056374 >
W21056374 >
W21056374 > spool off
```

---

4. Evaluate capabilities of the models you have developed.

(3 marks)

Answer

Using the output of the code above, the performance of each of the six models is summarised in the table below:

Model	Number of Build Cases	Accuracy (%)	False Positives (%)	False Negatives (%)
ufcrc_rf_model_1	10,000	80.49	4.82	14.69
ufcrc_rf_model_2	15,000	82.11	3.61	14.27
ufcrc_rf_model_3	21,000	84.40	2.85	12.75
ufcrc_nb_model_1	10,000	78.32	9.36	12.32
ufcrc_nb_model_2	15,000	79.20	9.37	11.43
ufcrc_nb_model_3	21,000	81.53	8.05	10.42

The table shows that both the random forest and naive bayes models improved in accuracy as the sample size of the build data was increased. It also shows the random forest model performed slightly better in each instance.

Lift and gain charts can also be used to evaluate the performance of a data mining model and compare one against another (Larose and Larose, 2015). A lift chart provides a visual representation of how the model compares to a random guess and

uses a lift score or factor to quantify the difference (Microsoft, 2022). At each decile, a gain chart shows the ratio of cumulative observations that are correct up to that decile against the total number of correct observations in the dataset, when ranked in descending order of estimated probability.

In this scenario, lift and gain charts were not produced as the Oracle Data Mining application used did not provide the requisite functionality. Therefore, the following code was used to calculate the lift and gain scores for the third random forest and naive bayes models, as they performed the best when compared to models one and two. Please note, the spool output is shown.

```
W21056374 > @D:\UFCrC_Models_Lift.sql
W21056374 >
W21056374 > REM Calculate lift for third RF model.
W21056374 >
W21056374 > CREATE TABLE rf_eval AS
  2    SELECT card, t.prediction, t.probability
  3    FROM ufcrc_test_data_col_3, TABLE(PREDICTION_SET(ufcrc_rf_model_3
USING *)) t;

Table created.

W21056374 >
W21056374 > BEGIN
  2         DBMS_DATA_MINING.COMPUTE_LIFT (
  3            apply_result_table_name          => 'rf_eval',
  4            target_table_name              => 'ufcrc_test_data_col_3',
  5            case_id_column_name            => 'card',
  6            target_column_name             => 'defaultpaynxtmnt',
  7            lift_table_name                         => 'rf_lift_tab',
  8            positive_target_value             =>  to_char(1),
  9            score_column_name              => 'PREDICTION',
 10            score_criterion_column_name        => 'PROBABILITY',
 11            num_quantiles                    =>  10,
 12            cost_matrix_table_name            =>  null,
 13            apply_result_schema_name          =>  null,
 14            target_schema_name             =>      null,
 15            cost_matrix_schema_name          =>  null,
 16            score_criterion_type               =>  'PROBABILITY');
```

```
 17  END;
 18  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 > SELECT quantile_number AS DECILE, gain_cumulative AS GAIN,
lift_cumulative AS LIFT, quantile_total_count AS CASES
  2  FROM rf_lift_tab;

    DECILE       GAIN       LIFT      CASES
---------- ---------- ---------- ----------
         1 .355612682 3.55612682        600
         2  .52613539 2.63067695        600
         3 .630676949  2.1022565        600
         4 .712082262 1.78020566        600
         5 .789203085 1.57840617        600
         6 .849185947 1.41530991        600
         7 .896315338 1.28045048        600
         8 .946015424 1.18251928        600
         9 .984575835 1.09397315        600
        10          1          1        600

10 rows selected.

W21056374 >
W21056374 >
W21056374 > REM Calculate lift for third NB model.
W21056374 >
W21056374 > CREATE TABLE nb_eval AS
  2    SELECT card, t.prediction, t.probability
  3    FROM ufcrc_test_data_col_3, TABLE(PREDICTION_SET(ufcrc_nb_model_3
USING *)) t;

Table created.

W21056374 >
W21056374 > BEGIN
  2         DBMS_DATA_MINING.COMPUTE_LIFT (
  3            apply_result_table_name          => 'nb_eval',
  4            target_table_name              => 'ufcrc_test_data_col_3',
```

```
  5            case_id_column_name           => 'card',
  6            target_column_name            => 'defaultpaynxtmnt',
  7            lift_table_name                     => 'nb_lift_tab',
  8            positive_target_value          =>  to_char(1),
  9            score_column_name             => 'PREDICTION',
 10            score_criterion_column_name       => 'PROBABILITY',
 11            num_quantiles                 =>  10,
 12            cost_matrix_table_name         =>  null,
 13            apply_result_schema_name       =>  null,
 14            target_schema_name            =>     null,
 15            cost_matrix_schema_name       =>  null,
 16            score_criterion_type             =>  'PROBABILITY');
 17  END;
 18  /

PL/SQL procedure successfully completed.

W21056374 >
W21056374 >
W21056374 > SELECT quantile_number AS DECILE, gain_cumulative AS GAIN,
lift_cumulative AS LIFT, quantile_total_count AS CASES
  2  FROM nb_lift_tab;

    DECILE       GAIN       LIFT      CASES
---------- ---------- ---------- ----------
         1 .318766067 3.18766067        600
         2 .502999143 2.51499572        600
         3 .621251071  2.0708369        600
         4 .718747019 1.79686755        600
         5 .798628963 1.59725793        600
         6 .847472151 1.41245358        600
         7  .89545844 1.27922634        600
         8  .93916024  1.1739503        600
         9 .973436161 1.08159573        600
        10          1          1        600

10 rows selected.

W21056374 > spool off
```

These metrics also show the random forest model performing slightly better than the naive bayes model, by achieving more correct observations as a proportion of the total in the first three deciles. The difference between the remaining deciles is negligible.

---

5. Present and discuss your findings and make recommendations for UFCrC company.

(4 marks)

Answer

UFCrC requested that a highly accurate data mining model be developed to help them predict if a customer will default in the next month. Those customers identified as being most likely to default will then be offered suitable terms in an attempt to prevent them from doing so.

Six models were developed using two different algorithms and varying sample sizes derived from the CreditCardsV2 dataset. The models performed better and were more accurate as the sample size used to build them was increased. For example, the accuracy of the model using an algorithm called random forest rose from 80.49% to 84.4%, and the other model using the naive bayes algorithm increased in accuracy from 78.32% to 81.53%. These figures represent the percentage of predictions that each model got right when asked to process a number of test cases drawn from the CreditCardsV2 dataset. It is important to note that these test cases differed from the cases used to build the model.

A metric known as the lift score and another called the gain were used to evaluate the performance of the models further. Again, the random forest model performed slightly better and was able to predict 63.1% of correct cases amongst the top 30% of customers ranked as most likely to default, compared to a value of 62.1% for the naive bayes model. Equally, the random forest model performed 3.56 times better than a random guess when selecting the top 10% of customers ranked as most likely

to default. In comparison, the naive bayes model performed 3.19 times better, although it is important to note the scores narrowed considerably after the first decile.

Consequently, the model that performed best in testing accurately predicted 84.4% of defaulters using the random forest algorithm and a build data sample size of 21,000 cases. It should be noted that false negatives accounted for 12.75% of predictions when this algorithm was used, meaning 765 out of the 6,000 test cases were predicted not to default when the opposite was true. The value for false positives, where a customer was predicted to default but didn't, was lower at 2.85% of predictions, which equates to 171 test cases. These figures highlight that the model is not 100% reliable and a small but significant number of defaulters will not be identified.

The performance of the model improved as the sample size of the data used to build it increased, which suggests that greater accuracy might be achieved with more cases. As such, it is recommended that a dataset larger than 30,000 cases be provided if a more accurate model is required. Likewise, providing a statement and payment history of 12 months for each customer, rather than 6 months might result in greater accuracy as more data points are used to build the model.

The model that was developed did achieve a very good level of accuracy, predicting 84.4% of cases in the test dataset. The number of customers who were correctly predicted to default was 402, which equates to 6.7% of the dataset. Moving forward, if this percentage of customers are offered terms and just half are prevented from defaulting, UFCrC could retain their accounts and save a significant amount of money, particularly if the company has a large customer base.

Part 5 (10 marks)

Consider the PropertyPortal scenario in the Appendix. Produce a report for the Managing Director PropertyPortal elaborating on professional, legal, ethical and

security issues that need to be considered and make recommendations that you think are appropriate for the PropertyPortal database.

(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 5: 10 Marks [7 for the quality of your report addressing the above points, 2 for the quality of referencing and citation and adhering to the Harvard style, 1 for presentation of the report]

Introduction

This report sets out the professional, legal, ethical and security issues that PropertyPortal should consider throughout the lifetime of the database. It outlines threats which may harm the viability of the business and the importance of complying with legislation, as well as the consequences if things go wrong. Importantly, it offers recommendations on how to manage the risks associated with these issues.

It should be noted that the issues discussed in the report are deeply intertwined and cannot be considered in isolation. For example, an inability to secure a database (a security issue) could result in a breach of data protection legislation (a legal issue). It is also important to note that many of the issues considered will apply whether the database is hosted on the company's servers or in the cloud.

Key Issues

The PropertyPortal database contains personal information about customers and estate agent staff across England. Consequently, it is subject to the provisions of the Data Protection Act (DPA) 2018 and UK General Data Protection Regulation (GDPR). Both pieces of legislation apply to organisations that process personal data and there can be severe financial penalties for noncompliance. In some cases, fines totalling tens of millions of pounds can be issued if the principles in the DPA 2018 and GDPR are breached (Digit News, 2022).

Consequently, PropertyPortal should ensure it processes personal data in accordance with the principles in these pieces of legislation, which includes ensuring that consent is sought from an individual before their data is shared with a third party. This point is particularly relevant in this case, as the PropertyPortal database contains information that is commercially valuable to utilities providers, homeware retailers, insurance companies and a host of other businesses, including data analytics firms. As such, individuals need to be told clearly how their data will be processed and who will have access to it when consent is sought.

Another important legal issue concerns liability, particularly if personal data is held in the cloud by a service provider. It may be difficult to ascertain who is accountable if a breach occurs and responsibilities aren't set out explicitly in service level agreements or contracts. It has also been noted that hosting data in the cloud can make it difficult to comply with legislation (Dulaney and Easttom, 2018, p.15).

The last two paragraphs highlight that third party access to data is commonplace. It is therefore important to consider whether those granted access to this information are able to protect it. Similarly, it is worth considering whether they can be trusted to use the data as intended. This was illustrated by the companies Cambridge Analytica (Cadwalladr and Graham-Harrison, 2018) and Rapleaf (Sharda, Delen and Turban, 2017, p.477), who were shown to have an ulterior motive when processing Facebook user data.

Even when the processing of personal data is legal, there is still a need to consider whether it is ethical. This isn't easy, as ethics often aren't clearly defined and can vary across societies (Connolly and Begg, 2015, p.643). Moreover, ethical issues may not be immediately evident at the data processing stage. For example, an artificial intelligence tool developed by Amazon to screen job applications was later found to be biased against women (Dastin, 2018). Therefore, PropertyPortal should carefully consider how its data is used and continually review the secondary and tertiary effects of it being processed.

There are also several security issues that need to be considered. The PropertyPortal database will be accessed by independent estate agents, presumably through a web application. As a result, it will form part of a larger network of interconnected devices which will only be as secure as its weakest link (Elmasri and Navathe, 2017, p.1153). This means the security of the database cannot be viewed in isolation, particularly since an SQL injection attack using a web-based application is one of the biggest threats to a database system (Elmasri and Navathe, 2017, p.1173).

Cyber criminals can also employ denial of service attacks and ransomware to prevent an organisation and its users from accessing data (Dulaney and Easttom, 2018). Although, the threat isn't always posed by an external party as data breaches and attacks can be perpetrated from inside an organisation, either maliciously or unwittingly. Likewise, access to information can be lost if the datacentre hosting it is taken offline, such as by a natural disaster. Information can also be vulnerable when it is taken out of the database and either moved across networks or held on portable media.

Lastly, an organisation can suffer serious reputational damage if it misuses or fails to protect its data, which in turn can have significant financial consequences. The cyber attack suffered by the UK company Talk Talk in 2015 is a case in point. It was targeted by a group of teenage cyber criminals, who used a SQL injection attack to steal personal information belonging to thousands of its customers (Information Commissioner's Office, no date). The value of the company's shares plummeted following the attack (Guibourg, 2015) and its profits more than halved (Monaghan, 2016). Furthermore, it was later given a then record fine of £400,000 by the Information Commissioner's Office, which concluded that Talk Talk had the resources to prevent the breach but failed to do so (Information Commissioner's Office, no date).

<u>Recommendations</u>

To mitigate the risk of the issues above materialising, it is recommended that PropertyPortal consider implementing the following measures:

- **Employ a qualified Database Administrator (DBA)** – This individual can secure the database and ensure that the principle of least privileges is followed (Dulaney and Easttom, 2018, p376) by using database security mechanisms (Elmasri and Navathe, 2017, p1153). This should help counter both insider and external threats.
- **Employ a qualified web developer** – As web applications can be used to launch attacks against databases, a web developer who can secure these applications is essential.
- **Introduce a comprehensive information security policy and code of ethics** – These documents can help guide staff and ensure their actions comply with expected legal and ethical behaviour (Connolly and Begg, 2015, p656).
- **Ensure third parties are subjected to quality assurance** – If third parties are granted access to the PropertyPortal data, expectations of how they should use and handle that data should be detailed in an information sharing agreement or contract. Provisions for auditing their performance against these standards should be included. Alternatively, PropertyPortal could consider only working with third parties that have been accredited against a recognised information security standard, such as ISO/IEC 27001.
- **Use a backup** – If a breach or loss cannot be prevented, PropertyPortal needs to be able to recover its data in a timeframe that won't have a detrimental impact on the business.

Summary

PropertyPortal must ensure it complies with the UK's data protection legislation and actively take steps to protect the information held in its database. It should also give careful consideration to whether its data is being used in an ethical manner. Failing to act could result in the manifestation of the legal, security and ethical issues detailed in this report. In turn, this could have serious reputational and financial consequences for the business and harm its prospects.

References & Bibliography (Parts 1, 3, 4 and 5)

Alotaibi, O. and Pardede, E. (2019) 'Transformation of Schema from Relational Database (RDB) to NoSQL Databases', *Data*, 4(4), article number 148. Available at: https://doi.org/10.3390/data4040148.

Cadwalladr and Graham-Harrison (2018) 'Revealed: 50 million Facebook Profiles Harvested for Cambridge Analytica in Major Data Breach', *The Guardian*, 17 March. Available at: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election (Accessed: 1st Jul 2022).

Çığşar, B. and Ünal, D. (2019) 'Comparison of Data Mining Classification Algorithms Determining the Default Risk', *Scientific Programming*, vol. 2019, article number 8706505. Available at: https://doi.org/10.1155/2019/8706505.

Connolly, T. and Begg, C. (2015) *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th edn, Global edn. Boston: Pearson.

Dastin, G. (2018) 'Amazon Scraps Secret AI Recruiting Tool that Showed Bias Against Women', *Reuters*, 11 October. Available at: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G (Accessed: 1st Jul 2022).

Defence Science and Technology Laboratory (2019) *The Dstl Biscuit Book: Artificial Intelligence, Data Science and (mostly) Machine Learning*. Available at: https://www.gov.uk/government/publications/the-dstl-biscuit-book (Accessed: 1st Jul 2022).

Digit News (2022) *6 of the Biggest ICO Fines Ever Issued*. Available at: https://www.digit.fyi/data-protection-2020-the-biggest-fines-ever-issued-by-the-ico/ (Accessed: 1st Jul 2022).

Dulaney, E. and Easttom, C. (2018) *CompTIA Security+ Study Guide*. 7th edn. Indianapolis, Indiana: John Wiley & Sons.

Elmasri, R. and Navathe, S.B. (2017) *Fundamentals of Database Systems*. 7th edn. Boston: Pearson.

Guibourg, C. (2015) 'TalkTalk Share Price Plunges Twice as Deep as Sony, Carphone Warehouse, Barclays and EBay after Cyber Attacks', *City A.M.*, 13 November. Available at: https://www.cityam.com/talktalk-share-price-plunges-twice-as-deep-as-sony-carphone-warehouse-barclays-and-ebay-after-cyber-attacks/ (Accessed: 1st Jul 2022).

IBM (2021) *Overfitting*. Available at: https://www.ibm.com/cloud/learn/overfitting (Accessed: 1st Jul 2022).

Information Commissioner's Office (no date) *TalkTalk Cyber Attack – How the ICO's Investigation Unfolded*. Available at: https://ico.org.uk/about-the-ico/media-centre/talktalk-cyber-attack-how-the-ico-investigation-unfolded/ (Accessed: 1st Jul 2022).

Larose, D.T. and Larose, C.D. (2015) *Data Mining and Predictive Analytics*. 2nd edn. Hoboken, New Jersey: John Wiley & Sons.

Microsoft (2022) *Lift Chart (Analysis Services - Data Mining)*. Available at: https://docs.microsoft.com/en-us/analysis-services/data-mining/lift-chart-analysis-services-data-mining?view=asallproducts-allversions (Accessed: 1st Jul 2022).

Monaghan, A. (2016) 'TalkTalk Profits Halve after Cyber-attack', *The Guardian*, 12 May. Available at: https://www.theguardian.com/business/2016/may/12/talktalk-profits-halve-hack-cyber-attack#:~:text=TalkTalk%20profits%20more%20than%20halved,attack%20in%20October%20last%20year (Accessed: 1st Jul 2022).

Oracle (2010) *5 Classification*. Available at: https://docs.oracle.com/cd/E18283_01/datamine.112/e16808/classify.htm (Accessed: 1st Jul 2022).

Sharda, R., Delen, D. and Turban, E. (2017) *Business Intelligence, Analytics, and Data Science: A Managerial Perspective*. 4th edn. Harlow, England: Pearson.