Example – Section B (INF 214 Exam)

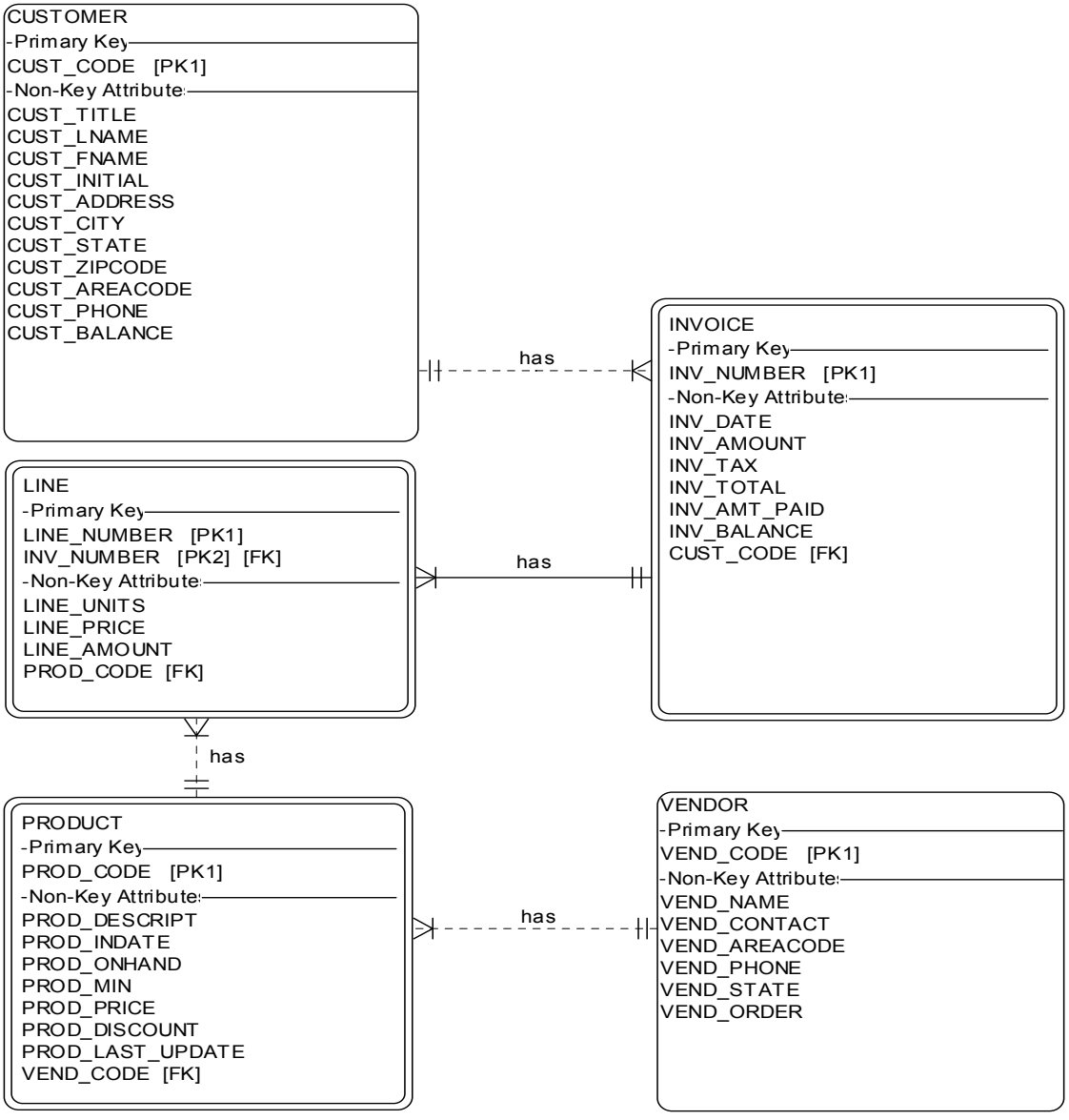
The following are two options in terms of the practical section of the exam. You may test some of the details, whilst you can review some of the other details. This is by no means a guarantee that the exam would look like this, however the approach would be similar.

Keep in mind that in the exam you will write the SQL by hand – so make sure that you understand how it works in terms of an ERD.

Feel free to prototype the following examples. You will have to import the files as I only provide the raw files for this example

JP van Deventer

EXAMPLE 1
Make use of the following information to complete the following SQL questions.



CUSTOMER

CUST_CODE	CUST_LNAME	CUST_FNAME	CUST_INITIAL	CUST_AREACODE	CUST_PHONE	CUST_BALANCE
10010	Ramas	Alfred	A	615	844-2573	R 0.00
10011	Dunne	Leona	K	713	894-1238	R 0.00
10012	Smith	Kathy	W	615	894-2285	R 345.86
10013	Olowski	Paul	F	615	894-2180	R 536.75
10014	Orlando	Myron		615	222-1672	R 0.00
10015	O'Brian	Amy	B	713	442-3381	R 0.00
10016	Brown	James	G	615	297-1228	R 221.19
10017	Williams	George		615	290-2556	R 768.93
10018	Farriss	Anne	G	713	382-7185	R 216.55
10019	Smith	Olette	K	615	297-3809	R 0.00

INVOICE

INV_NUMBER	CUST_CODE	INV_DATE	INV_SUBTOTAL	INV_TAX	INV_TOTAL
1001	10014	16-Jan-04	R 24.90	R 1.99	R 26.89
1002	10011	16-Jan-04	R 9.98	R 0.80	R 10.78
1003	10012	16-Jan-04	R 153.85	R 12.31	R 166.16
1004	10011	17-Jan-04	R 34.97	R 2.80	R 37.77
1005	10018	17-Jan-04	R 70.44	R 5.64	R 76.08
1006	10014	17-Jan-04	R 397.83	R 31.83	R 429.66
1007	10015	17-Jan-04	R 34.97	R 2.80	R 37.77
1008	10011	17-Jan-04	R 399.15	R 31.93	R 431.08

LINE

INV_NUMBER	LINE_NUMBER	PROD_CODE	LINE_UNITS	LINE_PRICE	LINE_TOTAL
1001	1	13-Q2/P2	1	R 14.99	R 14.99
1001	2	23109-HB	1	R 9.95	R 9.95
1002	1	54778-2T	2	R 4.99	R 9.98
1003	1	2238/QPD	1	R 38.95	R 38.95
1003	2	1546-QQ2	1	R 39.95	R 39.95
1003	3	13-Q2/P2	5	R 14.99	R 74.95
1004	1	54778-2T	3	R 4.99	R 14.97
1004	2	23109-HB	2	R 9.95	R 19.90
1005	1	PVC23DRT	12	R 5.87	R 70.44
1006	1	SM-18277	3	R 6.99	R 20.97
1006	2	2232/QTY	1	R 109.92	R 109.92
1006	3	23109-HB	1	R 9.95	R 9.95
1006	4	89-WRE-Q	1	R 256.99	R 256.99
1007	1	13-Q2/P2	2	R 14.99	R 29.98
1007	2	54778-2T	1	R 4.99	R 4.99
1008	1	PVC23DRT	5	R 5.87	R 29.35
1008	2	WR3/TT3	3	R 119.95	R 359.85
1008	3	23109-HB	1	R 9.95	R 9.95

PRODUCT

PROD_CODE	PROD_DESCRIPT	PROD_INDATE	PROD_ONHAND	PROD_MIN	PROD_PRICE	PROD_DISCOUNT	VEND_CODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-03	8	5	R 109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-03	32	15	R 14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-03	18	12	R 17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-04	15	8	R 39.95	0.00	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-04	23	5	R 43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-03	8	5	R 109.92	0.05	24288
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-03	6	5	R 99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-04	12	5	R 38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-04	23	10	R 9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-04	8	5	R 14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-03	43	20	R 4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-04	11	5	R 256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-04	188	75	R 5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-04	172	75	R 6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-04	237	100	R 8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-04	18	5	R 119.95	0.10	25595

VENDOR						
VEND_CODE	VEND_NAME	VEND_CONTACT	VEND_AREACODE	VEND_PHONE	VEND_STATE	VEND_ORDER
21225	Bryson, Inc.	Smithson	615	223-3234	TN	Y
21226	SuperLoo, Inc.	Flushing	904	215-8995	FL	N
21231	D&E Supply	Singh	615	228-3245	TN	Y
21344	Gomez Bros.	Ortega	615	889-2546	KY	N
22567	Dome Supply	Smith	901	678-1419	GA	N
23119	Randsets Ltd.	Anderson	901	678-3998	GA	Y
24004	Brackman Bros.	Browning	615	228-1410	TN	N
24288	ORDVA, Inc.	Hakford	615	898-1234	TN	Y
25443	B&K, Inc.	Smith	904	227-0093	FL	N
25501	Damal Supplies	Smythe	615	890-3529	TN	N
25595	Rubicon Systems	Orton	904	456-0092	FL	Y

QUESTIONS

1. Write the SQL query that can be used to generate a listing of all the purchases made by a CUSTOMER. In your query, ensure that the resultant output is systematically ordered. **(6)**

Purchases Listing

CUST_CODE	INV_NUMBER	INV_DATE	PROD_DESCRIPT	LINE_UNITS	LINE_PRICE
10011	1002	16-Jan-04	Rat-tail file, 1/8-in. fine	2	R 4.99
10011	1004	17-Jan-04	Claw hammer	2	R 9.95
10011	1004	17-Jan-04	Rat-tail file, 1/8-in. fine	3	R 4.99
10011	1008	17-Jan-04	Claw hammer	1	R 9.95
10011	1008	17-Jan-04	PVC pipe, 3.5-in., 8-ft	5	R 5.87
10011	1008	17-Jan-04	Steel matting, 4'x8'x1/6", .5" mesh	3	R 119.95
10012	1003	16-Jan-04	7.25-in. pwr. saw blade	5	R 14.99
10012	1003	16-Jan-04	B&D cordless drill, 1/2-in.	1	R 38.95
10012	1003	16-Jan-04	Hrd. cloth, 1/4-in., 2x50	1	R 39.95
10014	1001	16-Jan-04	7.25-in. pwr. saw blade	1	R 14.99
10014	1001	16-Jan-04	Claw hammer	1	R 9.95
10014	1006	17-Jan-04	1.25-in. metal screw, 25	3	R 6.99
10014	1006	17-Jan-04	B&D jigsaw, 12-in. blade	1	R 109.92
10014	1006	17-Jan-04	Claw hammer	1	R 9.95
10014	1006	17-Jan-04	Hicut chain saw, 16 in.	1	R 256.99
10015	1007	17-Jan-04	7.25-in. pwr. saw blade	2	R 14.99
10015	1007	17-Jan-04	Rat-tail file, 1/8-in. fine	1	R 4.99
10018	1005	17-Jan-04	PVC pipe, 3.5-in., 8-ft	12	R 5.87

ANSWER:

```
SELECT INVOICE.CUST_CODE, INVOICE.INV_NUMBER, INVOICE.INV_DATE,
PRODUCT.PROD_DESCRIPT, LINE.LINE_UNITS, LINE.LINE_PRICE
FROM CUSTOMER, INVOICE, LINE, PRODUCT
WHERE CUSTOMER.CUST_CODE = INVOICE.CUST_CODE
AND INVOICE.INV_NUMBER = LINE.INV_NUMBER
AND PRODUCT.PROD_CODE = LINE.PROD_CODE
ORDER BY INVOICE.CUST_CODE, INVOICE.INV_NUMBER, PRODUCT.PROD_DESCRIPT;
```

2. By referencing the following output result, write an SQL query that would generate the listing of CUSTOMER purchases, including the subtotals for each of the invoice LINE numbers. (Hint: Make use of the attributes LINE_UNITS * LINE_PRICE to calculate the subtotals.) **(6)**

Purchases Listing (Including LINE Subtotals)

CUST_CODE	INV_NUMBER	PROD_DESCRIPT	Units Bought	Unit Price	Subtotal
10011	1002	Rat-tail file, 1/8-in. fine	2	R 4.99	R 9.98
10011	1004	Claw hammer	2	R 9.95	R 19.90
10011	1004	Rat-tail file, 1/8-in. fine	3	R 4.99	R 14.97
10011	1008	Claw hammer	1	R 9.95	R 9.95
10011	1008	PVC pipe, 3.5-in., 8-ft	5	R 5.87	R 29.35
10011	1008	Steel matting, 4'x8'x1/6", .5" mesh	3	R 119.95	R 359.85
10012	1003	7.25-in. pwr. saw blade	5	R 14.99	R 74.95
10012	1003	B&D cordless drill, 1/2-in.	1	R 38.95	R 38.95
10012	1003	Hrd. cloth, 1/4-in., 2x50	1	R 39.95	R 39.95
10014	1001	7.25-in. pwr. saw blade	1	R 14.99	R 14.99
10014	1001	Claw hammer	1	R 9.95	R 9.95
10014	1006	1.25-in. metal screw, 25	3	R 6.99	R 20.97
10014	1006	B&D jigsaw, 12-in. blade	1	R 109.92	R 109.92
10014	1006	Claw hammer	1	R 9.95	R 9.95
10014	1006	Hicut chain saw, 16 in.	1	R 256.99	R 256.99
10015	1007	7.25-in. pwr. saw blade	2	R 14.99	R 29.98
10015	1007	Rat-tail file, 1/8-in. fine	1	R 4.99	R 4.99
10018	1005	PVC pipe, 3.5-in., 8-ft	12	R 5.87	R 70.44

ANSWER:

```
SELECT INVOICE.CUST_CODE, INVOICE.INV_NUMBER, PRODUCT.PROD_DESCRIPT,
LINE.LINE_UNITS AS [Units Bought], LINE.LINE_PRICE AS [Unit Price],
LINE.LINE_UNITS*LINE.LINE_PRICE AS Subtotal
FROM CUSTOMER, INVOICE, LINE, PRODUCT
WHERE CUSTOMER.CUST_CODE = INVOICE.CUST_CODE
AND INVOICE.INV_NUMBER = LINE.INV_NUMBER
AND PRODUCT.PROD_CODE = LINE.PROD_CODE
ORDER BY INVOICE.CUST_CODE, INVOICE.INV_NUMBER, PRODUCT.PROD_DESCRIPT;
```

3. Write a SQL query to determine the customer balance characteristics for all customers, including the total of the outstanding balances. The results of this query should appear as follows. **(5)**

All Customers' Balance Summary

Total Balance	Minimum Balance	Maximum Balance	Average Balance
R 2 089.28	R 0.00	R 768.93	R 208.93

ANSWER:

```
SELECT
Sum(CUST_BALANCE) AS [Total Balance],
Min(CUST_BALANCE) AS [Minimum Balance],
Max(CUST_BALANCE) AS [Maximum Balance],
Avg(CUST_BALANCE) AS [Average Balance]
FROM CUSTOMER;
```

4. Modify the SQL query in the previous question to produce the summary as stipulated in the following generated output. **(5)**

Updated Purchases Listing (Including LINE Subtotals)

CUST_CODE	CUST_BALANCE	Total Purchases
10011	R 0.00	R 444.00
10012	R 345.86	R 153.85
10014	R 0.00	R 422.77
10015	R 0.00	R 34.97
10018	R 216.55	R 70.44

ANSWER:

```
SELECT INVOICE.CUST_CODE, CUSTOMER.CUST_BALANCE,
Sum(LINE.LINE_UNITS*LINE.LINE_PRICE) AS [Total Purchases]
FROM CUSTOMER, INVOICE, LINE
WHERE INVOICE.INV_NUMBER = LINE.INV_NUMBER
AND CUSTOMER.CUST_CODE = INVOICE.CUST_CODE
GROUP BY INVOICE.CUST_CODE, CUSTOMER.CUST_BALANCE;
```

5. Use a SQL query to compute the average purchase amount per product made by each customer. Note that the Average Purchase Amount is equal to the Total Purchases divided by the Number of Purchases. (Hint: Make use of the attributes LINE_UNITS * LINE_PRICE to calculate the subtotals for [Total Purchases].) **(5)**

Average Purchases Amount Per Product

CUST_CODE	CUST_BALANCE	Total Purchases	Number of Purchases	Average Purchase Amount
10011	R 0.00	R 444.00	6	R 74.00
10012	R 345.86	R 153.85	3	R 51.28
10014	R 0.00	R 422.77	6	R 70.46
10015	R 0.00	R 34.97	2	R 17.49
10018	R 216.55	R 70.44	1	R 70.44

ANSWER:

```
SELECT INVOICE.CUST_CODE, CUSTOMER.CUST_BALANCE,
Sum(LINE.LINE_UNITS*LINE.LINE_PRICE) AS [Total Purchases],
Count(*) AS [Number of Purchases],
AVG(LINE.LINE_UNITS*LINE.LINE_PRICE) AS [Average Purchase Amount]
FROM CUSTOMER, INVOICE, LINE
WHERE INVOICE.INV_NUMBER = LINE.INV_NUMBER
AND CUSTOMER.CUST_CODE = INVOICE.CUST_CODE
GROUP BY INVOICE.CUST_CODE, CUSTOMER.CUST_BALANCE;
```

6. Write an SQL query that would generate the total purchases per invoice ask may be seen in the following resultant output. The Invoice Total is the sum of the product purchases in the LINEs that corresponds to the INVOICE. **(3)**

Total Purchases Per Invoice

INV_NUMBER	Invoice Total
1001	R 24.94
1002	R 9.98
1003	R 153.85
1004	R 34.87
1005	R 70.44
1006	R 397.83
1007	R 34.97
1008	R 399.15

ANSWER:

```
SELECT LINE.INV_NUMBER,
Sum(LINE.LINE_UNITS*LINE.LINE_PRICE) AS [Invoice Total]
FROM LINE
GROUP BY LINE.INV_NUMBER;
```

SECTION B TOTAL: (30)

EXAMPLE 2
Make use of the following information to complete the following SQL questions.

Brief Description: Property Rental

OUR COMPANY arranges rentals of properties owned by both private and business owners. OUR COMPANY assigns every property owner a unique owner number for identification and records its address (consisting of a street, street number, town or city, and province), the owner's name (consisting of first, middle, and last name for a person or name of a business), and the owners email addresses and the owner phone numbers. For a business owner, OUR COMPANY records the type (description) of its business. Each property is identified by a unique property number, its address and its type. Each property may be placed in several advertisements. Each such advertisement may be displayed in many newspapers on several dates. The newspapers are identified by unique names. The term "renter" refers to a private person or a business who signed a rental agreement for a property. Each such rental agreement is identified in OUR COMPANY's database by a unique rental number. OUR COMPANY records the date of the signing of the rental agreement and the start and end date of the rental agreement. A renter can rent many properties. A renter, prior to accepting the rental agreement may view the property repeatedly and OUR COMPANY records the date of viewing. For each renter, OUR COMPANY recordsthe renter's address, name, email address and phone numbers. Each renter has a unique renter number in the OUR COMPANY database. The OUR COMPANY agency is organized into branches and every staff member is allocated to exactly one branch. Each branch has one manager who is a member of the staff. In the OUR COMPANY database, each staff member is identified by a unique staff number. For each staff member OUR COMPANY records an address, name, email address, phone numbers, sex, position, and salary. Each property is in care of one of our branches. Each renter refers to the branch that is in care of the property it rents. Each property is overseen by a unique staff member. Each branch has an address, phone number, and a unique branch number.

Database Creation SQL: Property Rental
Databasisskeppings-SQL: Eiendom Verhuring

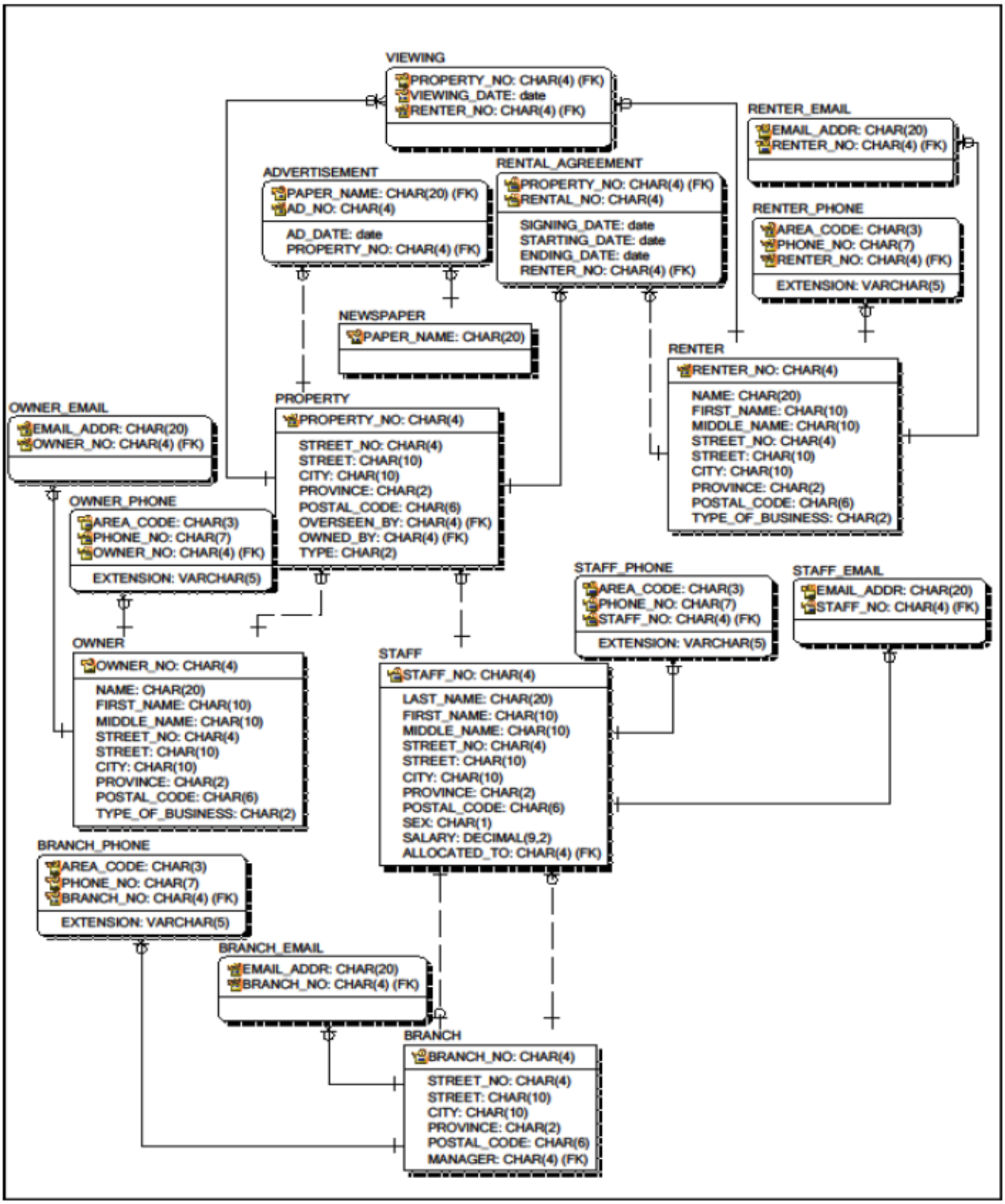
```
CREATETABLE BRANCH(
BRANCH_NO CHAR(4)NOTNULL,
STREET_NO CHAR(4)NOTNULL,
STREET CHAR(10)NOTNULL,
CITY CHAR(10)NOTNULL,
PROVINCE CHAR(2)NOTNULL,
POSTAL_CODE CHAR(6)NOTNULL,
MANAGER CHAR(4)NOTNULL,
PRIMARYKEY (BRANCH_NO),
CHECK (PROVINCE IN('AL','BC','MA','NB','NF','NT','NS','NU',
'ON','PE','QB','SA','YU')),
CHECK (('A'<=SUBSTRING(POSTAL_CODE,1,1)AND
SUBSTRING(POSTAL_CODE,1,1)<='Z')AND
('0'<=SUBSTRING(POSTAL_CODE,2,1)ANDSUBSTRING(POSTAL_CODE,2,1)<='9')
AND
('A'<=SUBSTRING(POSTAL_CODE,3,1)ANDSUBSTRING(POSTAL_CODE,3,1)<='Z')
AND
('0'<=SUBSTRING(POSTAL_CODE,4,1)ANDSUBSTRING(POSTAL_CODE,4,1)<='9')
AND
('A'<=SUBSTRING(POSTAL_CODE,5,1)ANDSUBSTRING(POSTAL_CODE,5,1)<='Z')
AND
('0'<=SUBSTRING(POSTAL_CODE,6,1)AND
SUBSTRING(POSTAL_CODE,6,1)<='9')),
UNIQUE(MANAGER)
)
CREATETABLE STAFF(
STAFF_NO CHAR(4)NOTNULL,
LAST_NAME CHAR(20)NOTNULL,
FIRST_NAME CHAR(10)NOTNULL,
MIDDLE_NAME CHAR(10),
STREET_NO CHAR(4)NOTNULL,
STREET CHAR(10)NOTNULL,
CITY CHAR(10)NOTNULL,
PROVINCE CHAR(2)NOTNULL,
POSTAL_CODE CHAR(6)NOTNULL,
SEX CHAR(1)NOTNULL,
SALARY DECIMAL(9,2)NOTNULL,
ALLOCATED_TO CHAR(4)NOTNULL,
PRIMARYKEY (STAFF_NO),
FOREIGNKEY (ALLOCATED_TO)REFERENCES BRANCH,
CHECK (PROVINCE IN('AL','BC','MA','NB','NF','NT','NS',
'NU','ON','PE','QB','SA','YU')),
CHECK (SEX IN('F','M','N')),
```

```
CHECK (SALARY > 0),
CHECK (('A'<=SUBSTRING(POSTAL_CODE,1,1)
ANDSUBSTRING(POSTAL_CODE,1,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,2,1)
ANDSUBSTRING(POSTAL_CODE,2,1)<='9')
AND('A'<=SUBSTRING(POSTAL_CODE,3,1)
ANDSUBSTRING(POSTAL_CODE,3,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,4,1)
ANDSUBSTRING(POSTAL_CODE,4,1)<='9')
AND('A'<=SUBSTRING(POSTAL_CODE,5,1)
ANDSUBSTRING(POSTAL_CODE,5,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,6,1)
ANDSUBSTRING(POSTAL_CODE,6,1)<='9'))
)
CREATETABLEOWNER (
OWNER_NO CHAR(4)NOTNULL,
NAME CHAR(20)NOTNULL,
FIRST_NAME CHAR(10),
MIDDLE_NAME CHAR(10),
STREET_NO CHAR(4)NOTNULL,
STREET CHAR(10)NOTNULL,
CITY CHAR(10)NOTNULL,
PROVINCE CHAR(2)NOTNULL,
POSTAL_CODE CHAR(6)NOTNULL,
TYPE_OF_BUSINESS CHAR(2),
PRIMARYKEY (OWNER_NO),
CHECK (PROVINCE IN('AL','BC','MA','NB','NF','NT','NS','NU',
'ON','PE','QB','SA','YU')),
CHECK (('A'<=SUBSTRING(POSTAL_CODE,1,1)ANDSUBSTRING(POSTAL_CODE,1,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,2,1)ANDSUBSTRING(POSTAL_CODE,2,1)<='9')
AND('A'<=SUBSTRING(POSTAL_CODE,3,1)ANDSUBSTRING(POSTAL_CODE,3,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,4,1)ANDSUBSTRING(POSTAL_CODE,4,1)<='9')
AND('A'<=SUBSTRING(POSTAL_CODE,5,1)ANDSUBSTRING(POSTAL_CODE,5,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,6,1)ANDSUBSTRING(POSTAL_CODE,6,1)<='9')),
CHECK(TYPE_OF_BUSINESS ISNULLOR(FIRST_NAME ISNULLAND MIDDLE_NAME
ISNULL))
)
CREATETABLE RENTER(
RENTER_NO CHAR(4)NOTNULL,
NAME CHAR(20)NOTNULL,
FIRST_NAME CHAR(10),
MIDDLE_NAME CHAR(10),
STREET_NO CHAR(4)NOTNULL,
STREET CHAR(10)NOTNULL,
CITY CHAR(10)NOTNULL,
PROVINCE CHAR(2)NOTNULL,
POSTAL_CODE CHAR(6)NOTNULL,
TYPE_OF_BUSINESS CHAR(2),
PRIMARYKEY (RENTER_NO),
CHECK (PROVINCE IN('AL','BC','MA','NB','NF','NT','NS','NU','ON',
'PE','QB','SA','YU')),
CHECK (('A'<=SUBSTRING(POSTAL_CODE,1,1)ANDSUBSTRING(POSTAL_CODE,1,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,2,1)ANDSUBSTRING(POSTAL_CODE,2,1)<='9')
AND('A'<=SUBSTRING(POSTAL_CODE,3,1)ANDSUBSTRING(POSTAL_CODE,3,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,4,1)ANDSUBSTRING(POSTAL_CODE,4,1)<='9')
AND('A'<=SUBSTRING(POSTAL_CODE,5,1)ANDSUBSTRING(POSTAL_CODE,5,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,6,1)ANDSUBSTRING(POSTAL_CODE,6,1)<='9')),
CHECK(TYPE_OF_BUSINESS ISNULLOR(FIRST_NAME ISNULLAND MIDDLE_NAME
ISNULL))
)
CREATETABLE PROPERTY(
PROPERTY_NO CHAR(4)NOTNULL,
STREET_NO CHAR(4)NOTNULL,
STREET CHAR(10)NOTNULL,
CITY CHAR(10)NOTNULL,
PROVINCE CHAR(2)NOTNULL,
POSTAL_CODE CHAR(6)NOTNULL,
OVERSEEN_BY CHAR(4)NOTNULL,
OWNED_BY CHAR(4)NOTNULL,
TYPECHAR(2)NOTNULL,
PRIMARYKEY (PROPERTY_NO),
FOREIGNKEY (OVERSEEN_BY)REFERENCES STAFF,
FOREIGNKEY (OWNED_BY)REFERENCESOWNER,
CHECK (PROVINCE IN('AL','BC','MA','NB','NF','NT','NS','NU','ON',
'PE','QB','SA','YU')),
CHECK (('A'<=SUBSTRING(POSTAL_CODE,1,1)ANDSUBSTRING(POSTAL_CODE,1,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,2,1)ANDSUBSTRING(POSTAL_CODE,2,1)<='9')
AND('A'<=SUBSTRING(POSTAL_CODE,3,1)ANDSUBSTRING(POSTAL_CODE,3,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,4,1)ANDSUBSTRING(POSTAL_CODE,4,1)<='9')
AND('A'<=SUBSTRING(POSTAL_CODE,5,1)ANDSUBSTRING(POSTAL_CODE,5,1)<='Z')
AND('0'<=SUBSTRING(POSTAL_CODE,6,1)ANDSUBSTRING(POSTAL_CODE,6,1)<='9'))
```

```
)
CREATETABLE RENTAL_AGREEMENT(
PROPERTY_NO CHAR(4)NOTNULL,
RENTAL_NO CHAR(4)NOTNULL,
SIGNING_DATE DATENOTNULL,
STARTING_DATE DATENOTNULL,
ENDING_DATE DATENOTNULL,
RENTER_NO CHAR(4)NOTNULL,
PRIMARYKEY (PROPERTY_NO,RENTAL_NO),
FOREIGNKEY (PROPERTY_NO)REFERENCES PROPERTY,
FOREIGNKEY (RENTER_NO)REFERENCES RENTER,
CHECK (SIGNING_DATE <= STARTING_DATE),
CHECK (STARTING_DATE <= ENDING_DATE)
)
CREATETABLE RENTER_EMAIL(
EMAIL_ADDR CHAR(20)NOTNULL,
RENTER_NO CHAR(4)NOTNULL,
PRIMARYKEY (EMAIL_ADDR,RENTER_NO),
FOREIGNKEY (RENTER_NO)REFERENCES RENTER
)
CREATETABLE STAFF_EMAIL(
EMAIL_ADDR CHAR(20)NOTNULL,
STAFF_NO CHAR(4)NOTNULL,
PRIMARYKEY (EMAIL_ADDR,STAFF_NO),
FOREIGNKEY (STAFF_NO)REFERENCES STAFF
)
CREATETABLE OWNER_EMAIL(
EMAIL_ADDR CHAR(20)NOTNULL,
OWNER_NO CHAR(4)NOTNULL,
PRIMARYKEY (EMAIL_ADDR,OWNER_NO),
FOREIGNKEY (OWNER_NO)REFERENCESOWNER
)
CREATETABLE BRANCH_EMAIL(
EMAIL_ADDR CHAR(20)NOTNULL,
BRANCH_NO CHAR(4)NOTNULL,
PRIMARYKEY (EMAIL_ADDR,BRANCH_NO),
FOREIGNKEY (BRANCH_NO)REFERENCES BRANCH
)
CREATETABLE RENTER_PHONE(
AREA_CODE CHAR(3)NOTNULL,
PHONE_NO CHAR(7)NOTNULL,
EXTENSION VARCHAR(5),
RENTER_NO CHAR(4)NOTNULL,
PRIMARYKEY (AREA_CODE,PHONE_NO,RENTER_NO),
FOREIGNKEY (RENTER_NO)REFERENCES RENTER,
CHECK(('0'<=SUBSTRING(AREA_CODE,1,1)ANDSUBSTRING(AREA_CODE,1,1)<='9')
AND('0'<=SUBSTRING(AREA_CODE,2,1)ANDSUBSTRING(AREA_CODE,2,1)<='9')
AND('0'<=SUBSTRING(AREA_CODE,3,1)ANDSUBSTRING(AREA_CODE,3,1)<='9')),
CHECK(('0'<=SUBSTRING(PHONE_NO,1,1)ANDSUBSTRING(PHONE_NO,1,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,2,1)ANDSUBSTRING(PHONE_NO,2,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,3,1)ANDSUBSTRING(PHONE_NO,3,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,4,1)ANDSUBSTRING(PHONE_NO,4,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,5,1)ANDSUBSTRING(PHONE_NO,5,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,6,1)ANDSUBSTRING(PHONE_NO,6,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,7,1)ANDSUBSTRING(PHONE_NO,7,1)<='9'))
)
CREATETABLE STAFF_PHONE(
AREA_CODE CHAR(3)NOTNULL,
PHONE_NO CHAR(7)NOTNULL,
EXTENSION VARCHAR(5),
STAFF_NO CHAR(4)NOTNULL,
PRIMARYKEY (AREA_CODE,PHONE_NO,STAFF_NO),
FOREIGNKEY (STAFF_NO)REFERENCES STAFF,
CHECK(('0'<=SUBSTRING(AREA_CODE,1,1)ANDSUBSTRING(AREA_CODE,1,1)<='9')
AND('0'<=SUBSTRING(AREA_CODE,2,1)ANDSUBSTRING(AREA_CODE,2,1)<='9')
AND('0'<=SUBSTRING(AREA_CODE,3,1)ANDSUBSTRING(AREA_CODE,3,1)<='9')),
CHECK(('0'<=SUBSTRING(PHONE_NO,1,1)ANDSUBSTRING(PHONE_NO,1,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,2,1)ANDSUBSTRING(PHONE_NO,2,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,3,1)ANDSUBSTRING(PHONE_NO,3,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,4,1)ANDSUBSTRING(PHONE_NO,4,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,5,1)ANDSUBSTRING(PHONE_NO,5,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,6,1)ANDSUBSTRING(PHONE_NO,6,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,7,1)ANDSUBSTRING(PHONE_NO,7,1)<='9'))
)
CREATETABLE OWNER_PHONE(
AREA_CODE CHAR(3)NOTNULL,
PHONE_NO CHAR(7)NOTNULL,
EXTENSION VARCHAR(5),
OWNER_NO CHAR(4)NOTNULL,
PRIMARYKEY (AREA_CODE,PHONE_NO,OWNER_NO),
FOREIGNKEY (OWNER_NO)REFERENCESOWNER,
```

```
CHECK(('0'<=SUBSTRING(AREA_CODE,1,1)ANDSUBSTRING(AREA_CODE,1,1)<='9')
AND('0'<=SUBSTRING(AREA_CODE,2,1)ANDSUBSTRING(AREA_CODE,2,1)<='9')
AND('0'<=SUBSTRING(AREA_CODE,3,1)ANDSUBSTRING(AREA_CODE,3,1)<='9')),
CHECK(('0'<=SUBSTRING(PHONE_NO,1,1)ANDSUBSTRING(PHONE_NO,1,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,2,1)ANDSUBSTRING(PHONE_NO,2,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,3,1)ANDSUBSTRING(PHONE_NO,3,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,4,1)ANDSUBSTRING(PHONE_NO,4,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,5,1)ANDSUBSTRING(PHONE_NO,5,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,6,1)ANDSUBSTRING(PHONE_NO,6,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,7,1)ANDSUBSTRING(PHONE_NO,7,1)<='9'))
)
CREATETABLE BRANCH_PHONE(
AREA_CODE CHAR(3)NOTNULL,
PHONE_NO CHAR(7)NOTNULL,
EXTENSION VARCHAR(5),
BRANCH_NO CHAR(4)NOTNULL,
PRIMARYKEY (AREA_CODE,PHONE_NO,BRANCH_NO),
FOREIGNKEY (BRANCH_NO)REFERENCES BRANCH,
CHECK(('0'<=SUBSTRING(AREA_CODE,1,1)ANDSUBSTRING(AREA_CODE,1,1)<='9')
AND('0'<=SUBSTRING(AREA_CODE,2,1)ANDSUBSTRING(AREA_CODE,2,1)<='9')
AND('0'<=SUBSTRING(AREA_CODE,3,1)ANDSUBSTRING(AREA_CODE,3,1)<='9')),
CHECK(('0'<=SUBSTRING(PHONE_NO,1,1)ANDSUBSTRING(PHONE_NO,1,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,2,1)ANDSUBSTRING(PHONE_NO,2,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,3,1)ANDSUBSTRING(PHONE_NO,3,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,4,1)ANDSUBSTRING(PHONE_NO,4,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,5,1)ANDSUBSTRING(PHONE_NO,5,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,6,1)ANDSUBSTRING(PHONE_NO,6,1)<='9')
AND('0'<=SUBSTRING(PHONE_NO,7,1)ANDSUBSTRING(PHONE_NO,7,1)<='9'))
)
CREATETABLE VIEWING(
PROPERTY_NO CHAR(4)NOTNULL,
RENTER_NO CHAR(4)NOTNULL,
VIEWING_DATE DATENOTNULL,
PRIMARYKEY (PROPERTY_NO,VIEWING_DATE,RENTER_NO),
FOREIGNKEY (PROPERTY_NO)REFERENCES PROPERTY,
FOREIGNKEY (RENTER_NO)REFERENCES RENTER
)
CREATETABLE NEWSPAPER(
PAPER_NAME CHAR(20)NOTNULL,
PRIMARYKEY (PAPER_NAME)
)
CREATETABLE ADVERTISEMENT(
PAPER_NAME CHAR(20)NOTNULL,
AD_NO CHAR(4)NOTNULL,
AD_DATE DATENOTNULL,
PROPERTY_NO CHAR(4)NOTNULL,
PRIMARYKEY (PAPER_NAME,AD_NO),
FOREIGNKEY (PAPER_NAME)REFERENCES NEWSPAPER,
FOREIGNKEY (PROPERTY_NO)REFERENCES PROPERTY
)
ALTERTABLE BRANCH
ADDCONSTRAINT MANAGER_CNST FOREIGNKEY (MANAGER)REFERENCES
STAFF(STAFF_NO)
```

Design ERD: Property Rental
Ontwerp EVD: Eiendom Verhuring



1. Write a script that will provide the staff number of each of the staff members whose salary is greater than 5000. Sort the result by the staff number **(3)**

ANSWER:

```
SELECT STAFF_NO FROM STAFF
WHERE SALARY > 5000
ORDER BY STAFF_NO
```

2. Write a script that will provide all the dates of all the advertisements posted in THE GLOBE AND MAIL in 2005. Avoid duplication results. **(4)**

ANSWER:

```
SELECT DISTINCT AD_DATE FROM ADVERTISEMENT
WHERE PAPER_NAME = 'THE GLOBE AND MAIL' AND AD_DATE>='2005-01-01'
AND AD_DATE<='2005-12-31'
ORDER BY AD_DATE
```

3. Write a script that will provide the email addresses and the renter number for all the private renters. Sort the result by the renter number. **(5)**

ANSWER:

```
SELECT EMAIL_ADDR, RENTER.RENTER_NO
FROM RENTER_EMAIL, RENTER
WHERE RENTER_EMAIL.RENTER_NO = RENTER.RENTER_NO
AND TYPE_OF_BUSINESS IS NULL
ORDER BY RENTER.RENTER_NO
```

4. Write a script that will provide the staff numbers and the names of all the staff that live on the same street, city, and province as their manager. The names should be listed in an alphabetic order (by last, then by first, then by middle names). **(10)**

ANSWER:

```
SELECT STAFF.STAFF_NO, FIRST_NAME, MIDDLE_NAME, LAST_NAME
FROM STAFF,(SELECT STAFF_NO, BRANCH_NO, STAFF.STREET, STAFF.CITY,
STAFF.PROVINCE FROM STAFF,BRANCH
WHERE STAFF.STAFF_NO = BRANCH.MANAGER) AS T
WHERE STAFF.ALLOCATED_TO = T.BRANCH_NO AND
STAFF.STAFF_NO != T.STAFF_NO AND
STAFF.STREET = T.STREET AND
STAFF.CITY = T.CITY AND
STAFF.PROVINCE = T.PROVINCE
ORDER BY LAST_NAME, FIRST_NAME, MIDDLE_NAME
```

5. Write a script that will find the owners and renters who have 2 or more phone numbers. Call the owner/renter number as CUSTOMER_NO, set the value of TYPE_OF_CUSTOMER to 'owner' if the customer is an owner, and to 'renter' if he/she is a renter. Only list the CUSTOMER_NO and TYPE_OF_CUSTOMER. **(10)**

ANSWER:

```
(SELECT OWNER_NO AS CUSTOMER_NO, 'OWNER' AS TYPE_OF_CUSTOMER
FROM OWNER
WHERE OWNER_NO IN (SELECT OWNER_NO FROM OWNER_PHONE
GROUP BY OWNER_NO HAVING COUNT(*) >= 2))
UNION
(SELECT RENTER_NO AS CUSTOMER_NO, 'RENTER' AS TYPE_OF_CUSTOMER
FROM RENTER
WHERE RENTER_NO IN (SELECT RENTER_NO FROM RENTER_PHONE
GROUP BY RENTER_NO HAVING COUNT(*) >= 2))
```

SECTION B TOTAL / *AFDELING B TOTAAL* : (40)