Helen Le

These assignments had us utilize two databases to generate unique business questions and the associated SQL queries, using at least a total of 10 different elements and then 15.

Minimum 10 Elements

Business Question 1:

Retrieve the Customer ID, First Name, Last Name, and Email Address of all Customers who have placed at least 2 orders and have a credit limit greater than 750.

Elements Used:

SELECT

FROM

WHERE

IN

GROUP BY

HAVING

COUNT

AND

NESTED QUERY

COMPARISON OPERATORS

```
SELECT CUST_ID, FIRST_NAME, LAST_NAME, EMAIL
    FROM CUSTOMER
        WHERE CUST_ID IN (
            SELECT CUST_ID
            FROM INVOICES
            GROUP BY CUST_ID
```

```
        HAVING COUNT(INVOICE_NUM) >= 2
        )
    AND CREDIT_LIMIT > 750;
```

Business Question 2:

KimTay is seeking to open up inventory for new shipments in the category DOG and is thus having a 30% off sale on DOG items. Retrieve the Item ID, Description, On Hand Stock, Category, Location, and New Price of these Items and sort them in ascending order of the New Price.

Elements Used:

COMPUTED COLUMN

AS ALIAS

ORDER BY

```
SELECT ITEM_ID, DESCRIPTION, ON_HAND, CATEGORY, LOCATION, (PRICE * 0.70) AS
NEW_PRICE
    FROM ITEM
        WHERE CATEGORY = 'DOG'
            ORDER BY NEW_PRICE ASC;
```

Minimum 15 Elements

Business Question 1:

List the invoice number, invoice date, sales representative ID, sale representative full name, customer ID, customer full name, and the customer's total spent which is the sum of quantity times quoted price (to account for invoices with multiple invoice lines):

- for customers whose credit limit is between 500 and 750,
- who purchased DOG or CAT items,
- where on hand stock was greater than 10
- and total spent was greater than 40.

Group by invoice number.

Order by total spent descending.

Elements Used:

SELECT

COMPUTED COLUMN

AGGREGATE FUNCTION

AS ALIAS

INNER JOIN

FROM

WHERE

IN (,,,)

COMPARISON OPERATOR

BETWEEN

AND

GROUP BY

HAVING

ORDER BY

DESC

```
SELECT IL.INVOICE_NUM, I.INVOICE_DATE, SR.REP_ID, SR.FIRST_NAME, SR.LAST_NAME,
C.CUST_ID, C.FIRST_NAME, C.LAST_NAME, SUM(IL.QUANTITY * IL.QUOTED_PRICE) AS
TOTAL_SPENT
    FROM SALES_REP SR INNER JOIN CUSTOMER C ON SR.REP_ID = C.REP_ID
        INNER JOIN INVOICES I ON C.CUST_ID = I.CUST_ID
        INNER JOIN INVOICE_LINE IL ON I.INVOICE_NUM = IL.INVOICE_NUM
        INNER JOIN ITEM IT ON IL.ITEM_ID = IT.ITEM_ID
            WHERE IT.CATEGORY IN ('DOG', 'CAT')
                AND IT.ON_HAND > 10
                AND C.CREDIT_LIMIT BETWEEN 500 AND 750
                    GROUP BY IL.INVOICE_NUM, I.INVOICE_DATE, SR.REP_ID,
SR.FIRST_NAME, SR.LAST_NAME, C.CUST_ID, C.FIRST_NAME, C.LAST_NAME
                        HAVING SUM(IL.QUANTITY * IL.QUOTED_PRICE) > 40
                            ORDER BY SUM(IL.QUANTITY * IL.QUOTED_PRICE) DESC;
```

Business Question 2:

Identify the sales representative (ID and full name) with the highest number of invoices, whose average total spent per invoice (calculated as the sum of quantity times quoted price) is below the average price of all items, excluding sales representatives who have processed more than two invoices by using EXCEPT.

Elements Used:

SELECT

DISTINCT

TOP 1

COMPUTED COLUMN

AGGREGATED FUNCTION

AS ALIAS

LEFT JOIN

INNER JOIN

GROUP BY

HAVING

COMPARISON OPERATOR (>)

NESTED SUBQUERY

EXCEPT

```
SELECT DISTINCT TOP 1 SR.REP_ID, SR.FIRST_NAME, SR.LAST_NAME,
COUNT(I.INVOICE_NUM) AS TOTAL_INVOICES, AVG(IL.QUANTITY * IL.QUOTED_PRICE) AS
AVG_TOTAL_SPENT
    FROM SALES_REP SR LEFT JOIN CUSTOMER C ON SR.REP_ID = C.REP_ID
        LEFT JOIN INVOICES I ON C.CUST_ID = I.CUST_ID
        INNER JOIN INVOICE_LINE IL ON I.INVOICE_NUM = IL.INVOICE_NUM
            GROUP BY SR.REP_ID, SR.FIRST_NAME, SR.LAST_NAME
                HAVING AVG(IL.QUANTITY * IL.QUOTED_PRICE) < (SELECT AVG(I.PRICE)
                                                             FROM ITEM I)
EXCEPT
SELECT SR.REP_ID, SR.FIRST_NAME, SR.LAST_NAME, COUNT(I.INVOICE_NUM) AS
TOTAL_INVOICES, AVG(IL.QUANTITY * IL.QUOTED_PRICE) AS AVG_TOTAL_SPENT
    FROM SALES_REP SR LEFT JOIN CUSTOMER C ON SR.REP_ID = C.REP_ID
        LEFT JOIN INVOICES I ON C.CUST_ID = I.CUST_ID
        INNER JOIN INVOICE_LINE IL ON I.INVOICE_NUM = IL.INVOICE_NUM
```

```
GROUP BY SR.REP_ID, SR.FIRST_NAME, SR.LAST_NAME
    HAVING COUNT(I.INVOICE_NUM) > 2;
```