--===============================================================================  
--=========================Exercise 7 SQL Element Checklist ===========================

--=========================Prof Wang’s Example Question & Query. ========================

--0. Retrieve Customer ID, Address of all customers who have a credit limit >= 100 and <= 1200, and have a valid address. Credit limit upper lower bounds must be indicated by special keyword operator and not comparison operators. Valid address means no missing or absence of value.

SELECT CUST\_ID, ADDRESS

FROM CUSTOMER

WHERE CREDIT\_LIMIT BETWEEN 100 AND 1200

AND ADDRESS IS NOT NULL;

SELECT

FROM

WHERE

BETWEEN

AND

NOT

IS NULL/IS NOT NULL

--========================== Checklist of 25 Items ==========================

---SQL Query Clauses:

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY

---Aggregate Functions:

SUM

AVG

COUNT

MAX

MIN

DISTINCT

---Simple & Compound Conditions:

AND

OR

NOT

---Other Operators/Keywords/Characters:

Comparison Operators < > = !

BETWEEN

IS NULL/IS NOT NULL

TOP N

LIKE %

ASC/DESC

IN (,,,)

---Computed Column:

Computed Column

AS Alias

---Subquery/Nesting Query/Nested Query:

Nested Query

My Examples Given

1)For each Rep, query their ID, total customers, the total/maximum/minimum balance/credit, and total available credit (credit - balance) for customers that have either not created an invoice, don’t have an email sent, or have an available credit less than or equal to zero.

SELECT REP\_ID, COUNT(\*) AS TOTAL\_CUS,

SUM(BALANCE) AS TOTAL\_BAL, SUM(CREDIT\_LIMIT) AS TOTAL\_CRED,

MAX(BALANCE) AS MAX\_BAL, MIN(BALANCE) AS MIN\_BAL,

MAX(CREDIT\_LIMIT) AS MAX\_CRED, MIN(CREDIT\_LIMIT) AS MIN\_CRED,

SUM(CREDIT\_LIMIT - BALANCE) AS TOTAL\_AVAIL\_CRED

FROM CUSTOMER

WHERE CUST\_ID NOT IN (SELECT CUST\_ID FROM INVOICES)

OR (CREDIT\_LIMIT – BALANCE) <= 0

OR EMAIL IS NULL

GROUP BY REP\_ID

SELECT

FROM

WHERE

GROUP BY

SUM

COUNT

MAX

MIN

OR

NOT

Comparison Operators < > = !

BETWEEN

IS NULL/IS NOT NULL

IN (,,,)

Computed Column

AS Alias

Nested Query

2)For each unique item, query their ID, description, category, location, in-stock amount, individual price, and total price (in-stock multiplied by per price). Due to an understocking issue in location B, filter only the items that don’t appear there and have a total value in-between $500 and $1,000 that should be relocated into location B. Sort these items by their on-hand amount, from largest to smallest then by their individual price also from largest to smallest

SELECT

FROM

WHERE

ORDER BY

DISTINCT

AND

Comparison Operators < > = !

ASC/DESC

Computed Column

AS Alias

3) For each invoice number, query the total and average quote price (quantity \* quoted price), but only for invoice numbers that appear more than once, then sort them by the invoice number in ascending order.

SELECT INVOICE\_NUM, SUM(QUANTITY \* QUOTED\_PRICE) AS TOTAL\_QUOTE,

AVG(QUANTITY \* QUOTED\_PRICE) AS AVG\_QUOTE

FROM INVOICE\_LINE

GROUP BY INVOICE\_NUM

HAVING COUNT(\*) > 1

ORDER BY INVOICE\_NUM ASC;

SELECT

FROM

GROUP BY

HAVING

ORDER BY

SUM

AVG

COUNT

Comparison Operators < > = !

ASC/DESC

Computed Column

AS Alias