**CS411, Winter 2022, Homework 4 – ch6, 7. Due by midnight March 15th**

Provide your answers in a Word file. Provide queries in text and the results in images. If any result has more than 10 rows, show the total number of rows retuned.

1. (10) Write a SELECT statement that returns these vendor\_id, vendor\_name, and the average of invoice\_totals for each vendor. Sort the result by vendor\_id.

Returned 34 rows

SELECT V.vendor\_id, V.vendor\_name, AVG(I.invoice\_total) AS 'Average Invoice'

FROM ap.vendors AS V

JOIN ap.invoices AS I

ON I.vendor\_id = V.vendor\_id

GROUP BY V.Vendor\_ID

Graphical user interface, table

Description automatically generated with medium confidence

1. (10) Write a SELECT statement that returns vendor\_name, the number of invoices, and the sum of the invoice\_total for each vendor where the number of invoices is greater than two.

Sort the result set by the number of invoices.

Returned 7 rows

SELECT V.vendor\_name, COUNT(I.invoice\_id) AS Invoice\_Count, SUM(I.invoice\_total) AS SUM\_INVOICE\_TOTAL

FROM ap.vendors AS V

JOIN ap.invoices AS I

ON I.vendor\_id = V.vendor\_id

GROUP BY V.Vendor\_ID

HAVING COUNT(I.invoice\_id) > 2

ORDER BY Invoice\_Count;

Table

Description automatically generated

1. (10) Write a SELECT statement that returns the account\_description , the number of items in the Invoice\_Line\_Items table for the account, and the sum of the line\_item\_amount for the account where number of line items is greater than two. Sort the result by the number of line items.

Returned 10 rows

SELECT GLA.account\_Description, COUNT(\*) AS Num\_Line\_Items, SUM(ILI.line\_item\_amount) AS SUM\_LINE\_ITEM\_AMOUNT

FROM ap.general\_ledger\_accounts AS GLA

JOIN ap.invoice\_line\_items AS ILI

ON ILI.account\_number = GLA.account\_number

GROUP BY ILI.account\_number

HAVING COUNT(\*) > 2

ORDER BY Num\_Line\_Items;

Table

Description automatically generated

1. (10) in the previous problem, show only those with the invoice date in June and July. Use the ROLLUP operator to include a row that shows the grand total.

Returned 7 rows

SELECT A.account\_Description AS account\_Description, SUM(A.Num\_Line\_Items) AS Num\_Line\_Items, SUM(A.SUM\_LINE\_ITEM\_AMOUNT) AS SUM\_LINE\_ITEM\_AMOUNT

FROM

(SELECT GLA.account\_Description AS account\_Description, COUNT(\*) AS Num\_Line\_Items, SUM(ILI.line\_item\_amount) AS SUM\_LINE\_ITEM\_AMOUNT

FROM ap.general\_ledger\_accounts AS GLA

JOIN ap.invoice\_line\_items AS ILI ON ILI.account\_number = GLA.account\_number

JOIN ap.invoices AS I ON I.invoice\_id = ILI.invoice\_id

WHERE MONTH(I.invoice\_date) IN (6, 7)

GROUP BY ILI.account\_number

HAVING COUNT(\*) > 2) AS A

GROUP BY A.account\_Description WITH ROLLUP

Table

Description automatically generated with medium confidence

1. (10) The following statement returns vendor names that have at least one invoice.   
   SELECT DISTINCT vendor\_name

FROM vendors JOIN invoices

ON vendors.vendor\_id = invoices.vendor\_id

ORDER BY vendor\_name

Provide a query with a subquery that returns the same result. Show the query and the result.

Returned 34 rows

SELECT V.vendor\_name

FROM vendors AS V

WHERE V.vendor\_name IN (

SELECT vendor\_name

FROM vendors JOIN invoices

ON vendors.vendor\_id = invoices.vendor\_id

GROUP BY vendor\_name

)

GROUP BY V.Vendor\_Name

ORDER BY V.vendor\_name;

Graphical user interface, text, application, chat or text message

Description automatically generated

1. (10) Write a query that shows the vendor name and the balance due for each vendor where the balance due is greater than the average balance due among all vendors. The average balance due must exclude fully paid balances. Note that you are not allowed to use more than one query to get the answer. You can use a subquery.

Returned 1 row

SELECT V.vendor\_name, SUM(invoice\_total - payment\_total - credit\_total) AS balance\_due

FROM ap.Vendors AS V

JOIN ap.invoices AS I ON V.vendor\_id = I.vendor\_ID

WHERE (invoice\_total - payment\_total - credit\_total ) > (

SELECT AVG(A.Balance)

FROM (

SELECT vendor\_name, AVG(invoice\_total - payment\_total - credit\_total) AS Balance

FROM ap.Vendors AS V

JOIN ap.invoices AS I ON V.vendor\_id = I.vendor\_ID

WHERE invoice\_total - payment\_total - credit\_total > 0

GROUP BY V.vendor\_id

HAVING AVG(invoice\_total - payment\_total - credit\_total) > 0) AS A)

GROUP BY V.vendor\_name

Shape

Description automatically generated

1. (10) Write a query that shows the account\_number and account\_description where the account number that has been assigned to more than one line item in the Invoice\_Line\_Items table. Use a subquery. Sort the result by the account number.

Returned 10 rows

SELECT GLA.account\_Number, GLA.account\_description

FROM ap.invoice\_Line\_Items AS ITI

JOIN ap.general\_ledger\_accounts AS GLA ON ITI.account\_Number = GLA.account\_number

WHERE GLA.account\_number IN (

SELECT account\_number

FROM ap.invoice\_Line\_Items AS ITI

GROUP BY Account\_Number

having COUNT(\*) > 1)

GROUP BY GLA.account\_Number

ORDER BY GLA.account\_number;

Table

Description automatically generated

1. (10) Write a query that returns vendor\_name, invoice\_id, invoice\_sequence, and line\_item\_amount. Return a row for each line item of each invoice that has more than one line item in the Invoice\_Line\_Items table. Hint: Use a subquery that tests for invoice\_sequence > 1. Sort the results by the vendor\_name, invoice\_id, and invoice\_sequence columns.

Returned 6 rows

SELECT vendor\_name, vendor\_id, invoice\_sequence, line\_item\_amount

FROM ap.vendors AS V

JOIN ap.Invoices AS I USING (Vendor\_ID)

JOIN ap.invoice\_line\_items as ILI USING (invoice\_id)

WHERE I.invoice\_id IN (

SELECT invoice\_id -- , COUNT(\*)

FROM ap.invoice\_line\_items

WHERE invoice\_sequence > 1)

ORDER BY vendor\_name, invoice\_id, invoice\_sequence

Table

Description automatically generated

1. (10) Write a query that returns vendor\_id, vendor\_name and the largest unpaid invoice for each vendor. Sort the reulsts by the vendor\_id.

Returned 7 rows

SELECT V.vendor\_id, V.vendor\_name, MAX(Invoice\_total - payment\_total - credit\_total) AS 'Largest unpaid invoice'

FROM ap.vendors as V

JOIN ap.invoices AS I ON V.vendor\_id = I.vendor\_id

WHERE Invoice\_total - payment\_total - credit\_total > 0

Group by V.vendor\_id;

ORDER BY V.vendor\_id

Graphical user interface, text, application

Description automatically generated

1. (10) Write a query that shows vendor names where the vendor is the only vendor located in the city and state. Sort the results by state.

SELECT vendor\_name

FROM ap.vendors

GROUP BY vendor\_state, vendor\_city

having COUNT(\*) = 1

ORDER BY vendor\_state

Graphical user interface, text, application, chat or text message

Description automatically generated