**LAB 3**

**Chapter 6 exercises**

**Exercise 1**:

Write a SELECT statement that returns one row for each vendor in the Invoices table that contains these columns:

The vendor\_id column from the Invoices table

The sum of the invoice\_total columns in the Invoices table for that vendor

**use ap;**

**SELECT vendor\_id, SUM(invoice\_total) AS invoice\_total\_sum**

**FROM invoices**

**GROUP BY vendor\_id**

**A screenshot of a computer

Description automatically generated**

**Exercise 2**:

Write a SELECT statement that returns one row for each vendor that contains these columns:

The vendor\_name column from the Vendors table

The sum of the payment\_total columns in the Invoices table for that vendor

Sort the result set in descending sequence by the payment total sum for each vendor.

SELECT vendor\_name, SUM(payment\_total) AS payment\_total\_sum

FROM vendors v JOIN invoices i

ON v.vendor\_id = i.vendor\_id

GROUP BY vendor\_name

ORDER BY payment\_total\_sum DESC

A screenshot of a computer

Description automatically generated

**Exercise 3**:

Write a SELECT statement that returns one row for each vendor that contains three columns:

The vendor\_name column from the Vendors table

The count of the invoices in the Invoices table for each vendor

The sum of the invoice\_total columns in the Invoices table for each vendor

Sort the result set so the vendor with the most invoices appears first.

SELECT vendor\_name, COUNT(\*) AS invoice\_count,

SUM(invoice\_total) AS invoice\_total\_sum

FROM vendors v JOIN invoices i

ON v.vendor\_id = i.vendor\_id

GROUP BY vendor\_name

ORDER BY invoice\_count DESC

**A screenshot of a computer

Description automatically generated**

**Exercise 4**:

Write a SELECT statement that returns one row for each general ledger account number that contains three columns:

The account\_description column from the General\_Ledger\_Accounts table

The count of the items in the Invoice\_Line\_Items table that have the same account\_number

The sum of the line\_item\_amount columns in the Invoice\_Line\_Items table that have the same account\_number

Return only those rows where the count of line items is greater than 1.

Group the result set by the account\_description column.

Sort the result set in descending sequence by the sum of the line item amounts.

SELECT account\_description, COUNT(\*) AS line\_item\_count,

SUM(line\_item\_amount) AS line\_item\_amount\_sum

FROM general\_ledger\_accounts gl

JOIN invoice\_line\_items li

ON gl.account\_number = li.account\_number

GROUP BY account\_description

HAVING line\_item\_count > 1

ORDER BY line\_item\_amount\_sum DESC

A screenshot of a computer

Description automatically generated

**Exercise 5**:

Modify the solution to exercise 4 so it returns only invoices dated in the second quarter of 2018 (April 1, 2018 to June 30, 2018). This should still return 10 rows but with some different line item counts for each vendor.

Hint: Join to the Invoices table to code a search condition based on invoice\_date.

SELECT account\_description, COUNT(\*) AS line\_item\_count,

SUM(line\_item\_amount) AS line\_item\_amount\_sum

FROM general\_ledger\_accounts gl

JOIN invoice\_line\_items li

ON gl.account\_number = li.account\_number

JOIN invoices i

ON li.invoice\_id = i.invoice\_id

WHERE invoice\_date BETWEEN '2018-04-01' AND '2018-06-30'

GROUP BY account\_description

HAVING line\_item\_count > 1

ORDER BY line\_item\_amount\_sum DESC

A screenshot of a computer

Description automatically generated

**Exercise 6**:

Write a SELECT statement that answers this question: What is the total amount invoiced for each general ledger account number? Return these columns:

The account\_number column from the Invoice\_Line\_Items table

The sum of the line\_item\_amount columns from the Invoice\_Line\_Items table

Use the WITH ROLLUP operator to include a row that gives the grand total.

SELECT account\_number, SUM(line\_item\_amount) AS line\_item\_sum

FROM invoice\_line\_items

GROUP BY account\_number WITH ROLLUP

A screenshot of a computer

Description automatically generated

**Chapter 7 exercises**

**Exercise 1**:

Write a SELECT statement that returns the same result set as this SELECT

statement, but don’t use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

SELECT vendor\_name

FROM vendors

WHERE vendor\_id IN

(SELECT DISTINCT vendor\_id FROM invoices)

ORDER BY vendor\_name

A screenshot of a computer

Description automatically generated

**Exercise 2**:

Write a SELECT statement that answers this question: Which invoices have a payment total that’s greater than the average payment total for all invoices with a payment total greater than 0? Return the invoice\_number and invoice\_total columns for each invoice. Sort the results by the invoice\_total column in descending order.

SELECT invoice\_number, invoice\_total

FROM invoices

WHERE payment\_total >

(SELECT AVG(payment\_total)

FROM invoices

WHERE payment\_total > 0)

ORDER BY invoice\_total DESC

**A screenshot of a computer

Description automatically generated**

**Exercise 3:**

Write a SELECT statement that returns two columns from the General\_Ledger\_Accounts table: account\_number and account\_description.

Return one row for each account number that has never been assigned to any line item in the Invoice\_Line\_Items table. To do that, use a subquery introduced with the NOT EXISTS operator.

Sort the results by the account\_number column.

SELECT account\_number, account\_description

FROM general\_ledger\_accounts gl

WHERE NOT EXISTS

(SELECT \*

FROM invoice\_line\_items

WHERE account\_number = gl.account\_number)

ORDER BY account\_number

A screenshot of a computer

Description automatically generated

**Exercise 4:**

Write a SELECT statement that returns four columns: 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.

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

FROM vendors v JOIN invoices i

ON v.vendor\_id = i.vendor\_id

JOIN invoice\_line\_items li

ON i.invoice\_id = li.invoice\_id

WHERE i.invoice\_id IN

(SELECT DISTINCT invoice\_id

FROM invoice\_line\_items

WHERE invoice\_sequence > 1)

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

A screenshot of a computer

Description automatically generated