1. (5) Write a SELECT statement that returns these four columns where the balance due is greater than 200.

vendor\_name The vendor\_name column from the Vendors table

invoice\_number The invoice\_number column from the Invoices table

invoice\_date The invoice\_date column from the Invoices table

balance\_due The invoice\_total column minus the payment\_total and credit\_total columns from the Invoices table

Sort the result by balance\_due.

Returned 5 rows

SELECT ven.vendor\_name

,inv.invoice\_number

,inv.invoice\_date

,(inv.invoice\_total - inv.payment\_total - inv.credit\_total) AS balance\_due

FROM ap.vendors AS ven

JOIN ap.invoices AS inv

ON ven.vendor\_id = inv.vendor\_id

WHERE (inv.invoice\_total - inv.payment\_total - inv.credit\_total) > 200

ORDER BY (inv.invoice\_total - inv.payment\_total - inv.credit\_total)

Graphical user interface, application, table

Description automatically generated

1. (5) Write a SELECT statement that returns three columns:

vendor\_id The vendor\_id column from the Vendors table

vendor\_name The vendor\_name column from the Vendors table

contact\_name A concatenation of the vendor\_contact\_first\_name and vendor\_contact\_last\_name columns with a space between

Return one row for each vendor whose contact has the same first name as another vendor’s contact. This should return 6 rows. Hint: Use a self-join to check that the vendor\_id columns aren’t equal but the vendor\_contact\_first\_name columns are equal. Sort the result set by vendor\_contact\_first\_name.

Returned 6 rows

SELECT ven1.vendor\_id, ven1.vendor\_name, concat(concat(ven1.vendor\_contact\_first\_name, ' '), ven1.vendor\_contact\_last\_name) AS contact\_name

FROM ap.vendors as ven1

join ap.vendors as ven2

ON ven1.vendor\_contact\_first\_name = ven2.vendor\_contact\_first\_name

AND ven1.vendor\_id != ven2.vendor\_id

ORDER BY ven1.vendor\_contact\_first\_name

Graphical user interface

Description automatically generated

1. (5) Show the vendor name, line\_item\_description, and account number for each line items. Find vendors which use more than one account number from the result.

Returned 13 rows

SELECT ven.vendor\_name, invItem.line\_item\_description, invItem.account\_number

FROM ap.invoice\_line\_items AS invItem

INNER JOIN ap.invoices AS inv

ON inv.invoice\_id = invItem.invoice\_id

INNER JOIN ap.Vendors AS ven

ON inv.vendor\_id = ven.vendor\_id

WHERE ven.vendor\_id IN (

SELECT ven.vendor\_id

FROM ap.invoice\_line\_items AS invItem

INNER JOIN ap.invoices AS inv

ON inv.invoice\_id = invItem.invoice\_id

INNER JOIN ap.Vendors AS ven

ON inv.vendor\_id = ven.vendor\_id

GROUP BY ven.vendor\_id

HAVING COUNT(DISTINCT invItem.account\_number) > 1) Table

Description automatically generated

1. (5) For each vendor, show vendor name, its invoice number, invoice id, line item amount, and line item description. Sort the result by the vendor name.

Returned 118 rows

SELECT ven.vendor\_name

,inv.invoice\_number

,inv.invoice\_id

,invItem.line\_item\_amount

,invItem.line\_item\_description

FROM ap.invoice\_line\_items AS invItem

INNER JOIN ap.invoices AS inv

ON inv.invoice\_id = invItem.invoice\_id

INNER JOIN ap.Vendors AS ven

ON inv.vendor\_id = ven.vendor\_id

ORDER BY ven.vendor\_name

**Table

Description automatically generated**

1. (10) Show two columns from the Vendors table: vendor\_name and vendor\_phone. If the vendor has a phone number, the vendor\_phone value should be its phone number. Otherwise, the vendor\_phone value should be “No Phone.” Sort the result set by vendor\_name.  
   Hint: You can use UNION to combine vendors with phone number and vendors without phone number.

Returned 122 rows

SELECT vendor\_name, vendor\_phone

FROM ap.Vendors

WHERE vendor\_phone IS NOT null

UNION

SELECT vendor\_name, "No Phone”

FROM ap.Vendors

WHERE vendor\_phone IS null

ORDER BY vendor\_name

Table

Description automatically generated with medium confidence

1. (10) Consider the general\_ledger\_accounts and invoice\_line\_items table.  
   Diagram

   Description automatically generated  
   Write a SELECT statement that returns these two columns, showing the accounts that do not appear on the invoice\_line\_items table.

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

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

Note that the account numbers returned have never been used. That is, they do not appear in invoice\_line\_items.

Returned 54 rows

SELECT account\_number, account\_description

FROM ap.general\_ledger\_accounts

WHERE account\_number NOT IN (

SELECT account\_number

FROM ap.invoice\_line\_items)

Table

Description automatically generated

1. (5) Create a table called new\_terms from the terms table. Then, write an INSERT statement that adds this row to the new\_terms table:

terms\_id: 6terms\_description: Net due 120 days

terms\_due\_days: 120

Then, show the result of the following query: SELECT \* FROM new\_terms;

Graphical user interface, application

Description automatically generated

1. (5) Write an UPDATE statement that modifies the row you just added to the Terms table. This statement should change the terms\_description column to “Net due 150 days”, and it should change the terms\_due\_days column to 150.

Then, show the result of the following query: SELECT \* FROM new\_terms;

SET SQL\_SAFE\_UPDATES = 0;

UPDATE ap.new\_terms

SET terms\_description = 'Net due 150 days', terms\_due\_days = 150;

SET SQL\_SAFE\_UPDATES = 1;

Graphical user interface, text, application

Description automatically generated

1. (5) Write a DELETE statement that deletes the row you added to the new Terms table (that is with the terms\_id 6).Then, show the result of the following query: SELECT \* FROM new\_terms;

DELETE FROM ap.new\_terms

WHERE terms\_id = 6

Graphical user interface, application

Description automatically generated

1. (5) Write an UPDATE statement that modifies the Terms table. Change the terms\_description column to “Net due 100 days” and the terms\_due\_days column to 100 for the term with an ID of 5. Then, show the result of the following query: SELECT \* FROM new\_terms;

SET SQL\_SAFE\_UPDATES = 0;

UPDATE ap.new\_terms

SET terms\_description = 'Net due 150 days', terms\_due\_days = 150

WHERE terms\_id = 5;

SET SQL\_SAFE\_UPDATES = 1;

Graphical user interface

Description automatically generated with medium confidence

1. (40) Consider the following ER diagram.

Diagram, table

Description automatically generated with medium confidence

1. (2) What is the relationship between general\_ledger\_accounts and invoice\_line\_items? (Show whether it is 1:1, 1:Many, Many:1 or 1:1.)

1:Many

1. (9) Show the account\_description that has the greatest number of invoices. Show the query that finds the answer.

SELECT account\_description, count(\*)

FROM ap.general\_ledger\_accounts AS gen

JOIN ap.invoice\_line\_items AS inv

ON gen.account\_number = inv.account\_number

GROUP BY account\_description

ORDER BY COUNT(\*) DESC

LIMIT 1

Graphical user interface, text

Description automatically generated

1. (9) What is the total invoice for the account from the above question? Show the query that finds the answer.

SELECT account\_description, SUM(invoice\_total)

FROM ap.general\_ledger\_accounts AS gen

JOIN ap.invoice\_line\_items AS inv

ON gen.account\_number = inv.account\_number

JOIN ap.invoices AS invoice

ON invoice.invoice\_id = inv.invoice\_id

GROUP BY account\_description

ORDER BY COUNT(\*) DESC

LIMIT 1

Graphical user interface

Description automatically generated with low confidence

1. (10) Find the vendor which has the biggest amount of total invoices. Show the query that shows the vendor name and the total invoice amount.

SELECT vendor\_name, SUM(invoice\_total)

FROM ap.vendors AS ven

JOIN ap.invoices AS inv

ON ven.vendor\_id = inv.vendor\_id

GROUP BY vendor\_name

ORDER BY SUM(invoice\_total) DESC

LIMIT 1

Table

Description automatically generated with medium confidence

1. (10) Find the vendor which has the single most expensive order for an item. Show the query that shows the vendor name, the item description, and the amount.

SELECT vendor\_name, invLine.Line\_item\_description, Max(invoice\_total)

FROM ap.vendors AS ven

JOIN ap.invoices AS inv

ON ven.vendor\_id = inv.vendor\_id

JOIN ap.invoice\_line\_items AS invLine

ON inv.invoice\_id = invLine.invoice\_id

GROUP BY vendor\_name

ORDER BY Max(invoice\_total) DESC

LIMIT 1

A picture containing text

Description automatically generated