**LAB 2**

**Please take screenshots after running each query and put all these screenshots on one Word file. Submit this file on Blackboard**

**Create another database**

* Open the script file named my\_guitar\_shop.sql. Execute the entire script.

**Part I - Chapter 4 Exercises**

**Exercise 1**

Write a SELECT statement that returns all columns from the Vendors table inner-joined with all columns from the Invoices table. **This should return 114 rows**. Hint: You can use an asterisk (\*) to select the columns from both tables.

SELECT \*

FROM vendors JOIN invoices

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

A screenshot of a computer

Description automatically generated

**Exercise 2**

Write a SELECT statement that returns these four columns:

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

Use these aliases for the tables: v for Vendors and i for Invoices. Return one row for each invoice with a non-zero balance. **This should return 11 rows**.

Sort the result set by vendor\_name in ascending order.

SELECT vendor\_name, invoice\_number, invoice\_date,

invoice\_total - payment\_total - credit\_total AS balance\_due

FROM vendors v JOIN invoices i

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

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

ORDER BY vendor\_name

A screenshot of a computer

Description automatically generated

**Exercise 3**

Write a SELECT statement that returns these three columns:

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

default\_account The default\_account\_number column from the Vendors table

description The account\_description column from the

General\_Ledger\_Accounts table

Return one row for each vendor. **This should return 122 rows.**

Sort the result set by account\_description and then by vendor\_name.

SELECT vendor\_name, default\_account\_number AS default\_account,

account\_description AS description

FROM vendors v JOIN general\_ledger\_accounts gl

ON v.default\_account\_number = gl.account\_number

ORDER BY account\_description, vendor\_name

A screenshot of a computer

Description automatically generated

**Exercise 4**

Write a SELECT statement that returns these five columns:

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

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

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

li\_sequence The invoice\_sequence column from the Invoice\_Line\_Items

table

Li\_amount The line\_item\_amount column from the Invoice\_Line\_Items

table

Use aliases for the tables. **This should return 118 rows.**

Sort the final result set by vendor\_name, invoice\_date, invoice\_number, and invoice\_sequence.

SELECT vendor\_name, invoice\_date, invoice\_number,

invoice\_sequence AS li\_sequence,

line\_item\_amount AS li\_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

ORDER BY vendor\_name, invoice\_date, invoice\_number, invoice\_sequence

A screenshot of a computer

Description automatically generated

**Exercise 5**

Write a SELECT statement that returns these 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 last name as another vendor’s contact. **This should return 2 rows.**

**Hint:** Use a self-join to check that the vendor\_id columns aren’t equal but the vendor\_contact\_last\_name columns are equal.

Sort the result set by vendor\_contact\_last\_name.

SELECT v1.vendor\_id,

v1.vendor\_name,

CONCAT(v1.vendor\_contact\_first\_name, ' ', v1.vendor\_contact\_last\_name) AS contact\_name

FROM vendors v1 JOIN vendors v2

ON v1.vendor\_id <> v2.vendor\_id AND

v1.vendor\_contact\_last\_name = v2.vendor\_contact\_last\_name

ORDER BY v1.vendor\_contact\_last\_name

A screenshot of a computer

Description automatically generated

**Exercise 6**

Write a SELECT statement that returns these three columns:

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

table

account\_description The account\_description column from the

General\_Ledger\_Accounts table

Invoice\_id The invoice\_id column from the Invoice\_Line\_Items table

Return one row for each account number that has never been used. **This should return 54 rows. Hint:** Use an outer join and only return rows where the invoice\_id column contains a null value. Remove the invoice\_id column from the SELECT clause.

Sort the final result set by the account\_number column.

SELECT gl.account\_number, account\_description, invoice\_id

FROM general\_ledger\_accounts gl LEFT JOIN invoice\_line\_items li

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

WHERE li.invoice\_id IS NULL

ORDER BY gl.account\_number

A screenshot of a computer

Description automatically generated

**Exercise 7**

Use the UNION operator to generate a result set consisting of two columns from the Vendors table: vendor\_name and vendor\_state. If the vendor is in California, the vendor\_state value should be “CA”; otherwise, the vendor\_state value should be “Outside CA.” Sort the final result set by vendor\_name.

SELECT vendor\_name, vendor\_state

FROM vendors

WHERE vendor\_state = 'CA'

UNION

SELECT vendor\_name, 'Outside CA'

FROM vendors

WHERE vendor\_state <> 'CA'

ORDER BY vendor\_name

A screenshot of a computer

Description automatically generated

**Part II - Chapter 5 Exercises**

*To test whether a table has been modified correctly as you do these exercises, you can write and run an appropriate SELECT statement. Or, when you’re using MySQL Workbench, you can right-click on a table name in the Navigator window and select the Select Rows - Limit 1000 command to display the data for the table in a Result tab. To refresh the data in this tab after modifying the table data, click the Refresh button in the toolbar at the top of the tab.*

**Exercise 1**

Write an INSERT statement that adds this row to the Terms table:

terms\_id: 6

terms\_description: Net due 120 days

terms\_due\_days: 120

Use MySQL Workbench to review the column definitions for the Terms table, and include a column list with the required columns in the INSERT statement.

INSERT INTO terms

(terms\_id, terms\_description, terms\_due\_days)

VALUES

(6, 'Net due 120 days', 120)

A screenshot of a computer

Description automatically generated

**Exercise 2**

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 125 days”, and it should change the terms\_due\_days column to 125.

UPDATE terms

SET terms\_description = 'Net due 125 days',

terms\_due\_days = 125

WHERE terms\_id = 6

A screenshot of a computer

Description automatically generated

**Exercise 3**

Write a DELETE statement that deletes the row you added to the Terms table in Exercise 1

DELETE FROM terms

WHERE terms\_id = 6

A screenshot of a computer

Description automatically generated

**Exercise 4**

Write an INSERT statement that adds this row to the Invoices table:

invoice\_id: The next automatically generated ID

vendor\_id: 32

invoice\_number: AX-014-027

invoice\_date: 8/1/2018

invoice\_total: $434.58

payment\_total: $0.00

credit\_total: $0.00

terms\_id: 2

Invoice\_due\_date: 8/31/2018

Payment\_date: null

Write this statement without using a column list.

INSERT INTO invoices

VALUES (DEFAULT, 32, 'AX-014-027', '2018-08-01', 434.58, 0, 0,

2, '2018-08-31', NULL)

A screenshot of a computer

Description automatically generated

**Exercise 5**

Write an INSERT statement adds these rows to the Invoice\_Line\_Items table: invoice\_sequence: 1 2

account\_number: 160 527

line\_item\_amount: $180.23 $254.35

line\_item\_description: Hard drive Exchange Server update

Set the invoice\_id column of these two rows to the invoice ID that was gener-ated by MySQL for the invoice you added in exercise 4.

INSERT INTO invoice\_line\_items VALUES

(115, 1, 160, 180.23, 'Hard drive'),

(115, 2, 527, 254.35, 'Exchange Server update')

A screenshot of a computer

Description automatically generated

**Exercise 6**

Write an UPDATE statement that modifies the invoice you added in exercise 4. This statement should change the credit\_total column so it’s 10% of the invoice\_total column, and it should change the payment\_total column so the sum of the payment\_total and credit\_total columns are equal to the invoice\_total column.

UPDATE invoices

SET credit\_total = invoice\_total \* .1,

payment\_total = invoice\_total - credit\_total

WHERE invoice\_id = 115

A screenshot of a computer

Description automatically generated

**Exercise 7**

Write an UPDATE statement that modifies the Vendors table. Change the default\_account\_number column to 403 for the vendor with an ID of 44.

UPDATE vendors

SET default\_account\_number = 403

WHERE vendor\_id = 44

A screenshot of a computer

Description automatically generated

**Exercise 8**

Write an UPDATE statement that modifies the Invoices table. Change the terms\_id column to 2 for each invoice that’s for a vendor with a default\_terms\_id of 2.

UPDATE invoices

SET terms\_id = 2

WHERE vendor\_id IN

(SELECT vendor\_id

FROM vendors

WHERE default\_terms\_id = 2)

A screenshot of a computer

Description automatically generated

**Exercise 9**

Write a DELETE statement that deletes the row that you added to the Invoices table in exercise 4. When you execute this statement, it will produce an error since the invoice has related rows in the Invoice\_Line\_Items table. To fix that, precede the DELETE statement with another DELETE statement that deletes the line items for this invoice. (Remember that to code two or more statements in a script, you must end each statement with a semicolon.)

DELETE FROM invoice\_line\_items

WHERE invoice\_id = 115;

DELETE FROM invoices

WHERE invoice\_id = 115;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated