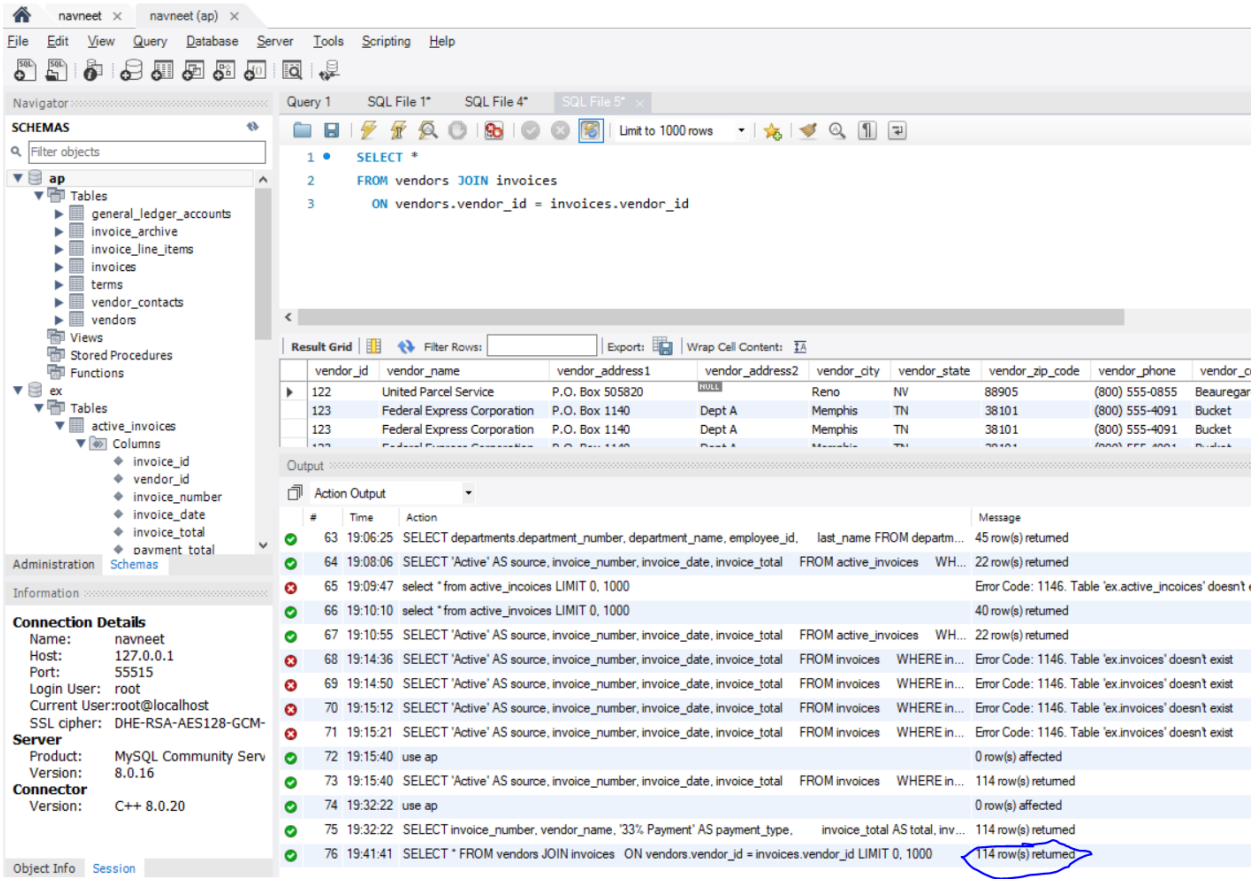
# MYSQL LAB 4 Exercise

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.*

## Answer: SELECT * FROM vendors JOIN invoices

## ON vendors.vendor_id = invoices.vendor_id




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 |

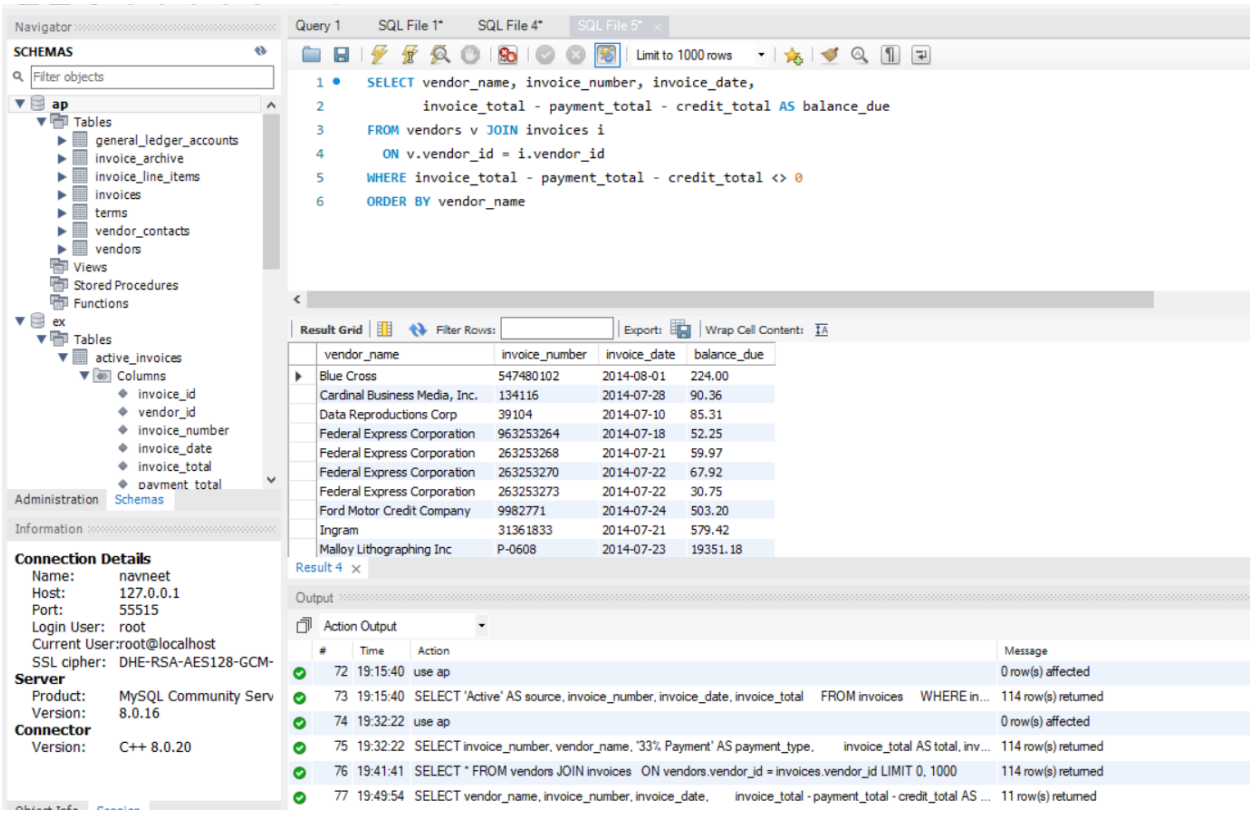
Use these aliases for the tables: v for Vendors and i for Invoices.

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.

## Answer:

```
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
```



3. Write a SELECT statement that returns these three columns:

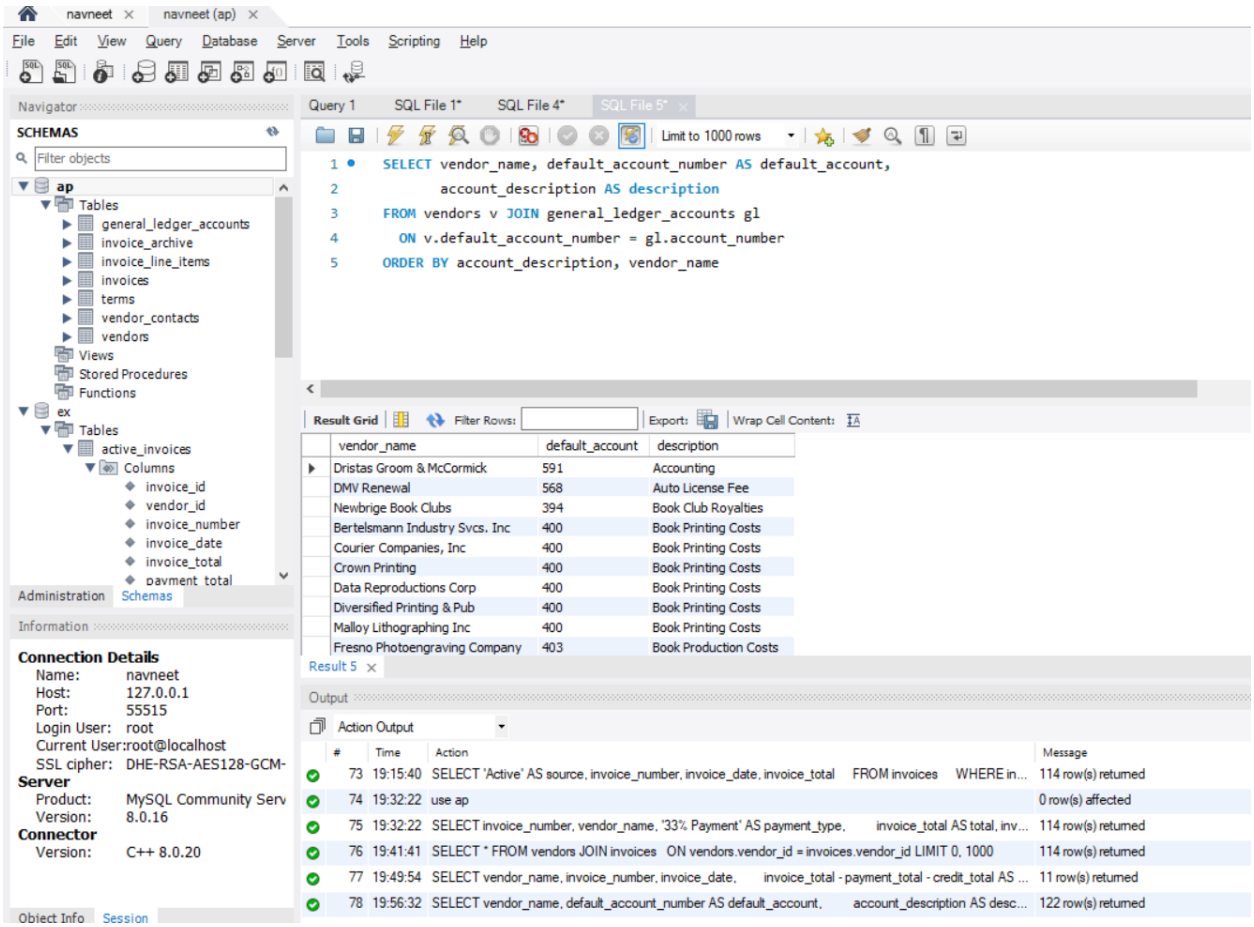
| | |
|---|---|
| vendor_name | The vendor_name column from the Vendors table |
| default_account | The default_account_number column from the Vendors table |
| description | The account_description column from the General_Ledger_Accounts table |

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

Sort the result set by account_description and then by vendor_name.

**Answer:**

```
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
```




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.

## Answer:

```
SELECT vendor_name, invoice_date, invoice_number,
    invoice_sequence AS li_sequence,
    line_item_amount AS li_amount
```

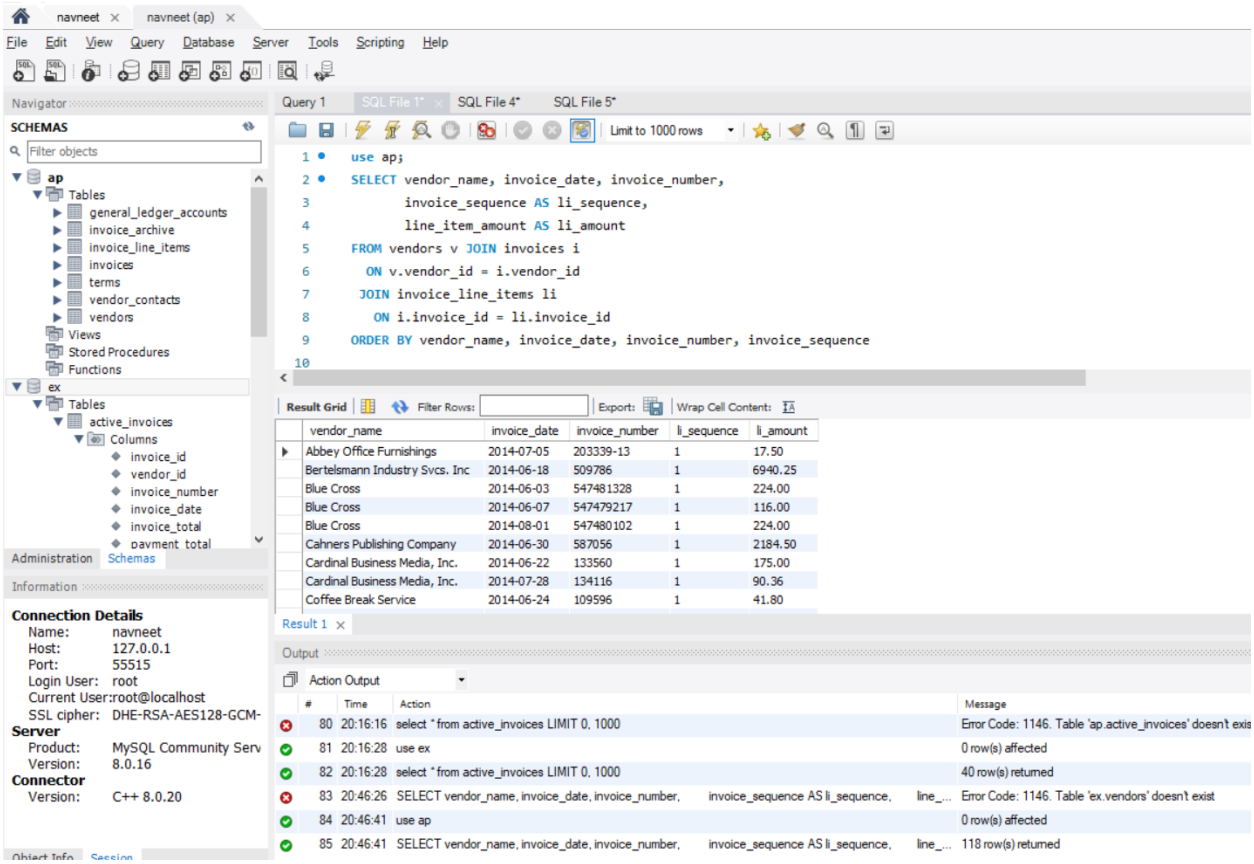
**FROM** vendors **v JOIN** invoices **i**

**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



5. Write a SELECT statement that returns three columns:

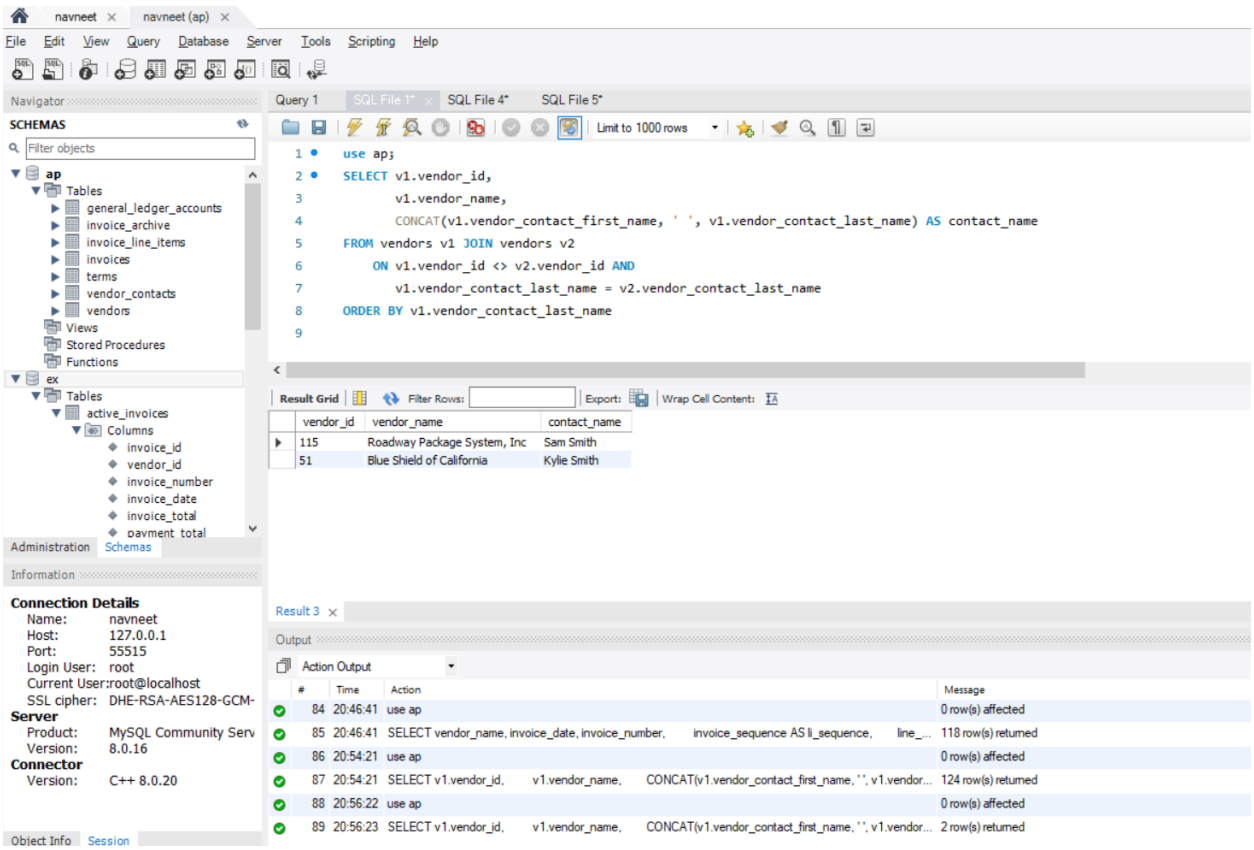
| | |
|---|---|
| vendor_id | The vendor_id column from the Vendors table |
| vendor_name | The vendor_name column from the Vendors table |
| contact_name | A concatenation of the vendor_contact_first_name and vendor_contact_last_name columns with a space between |

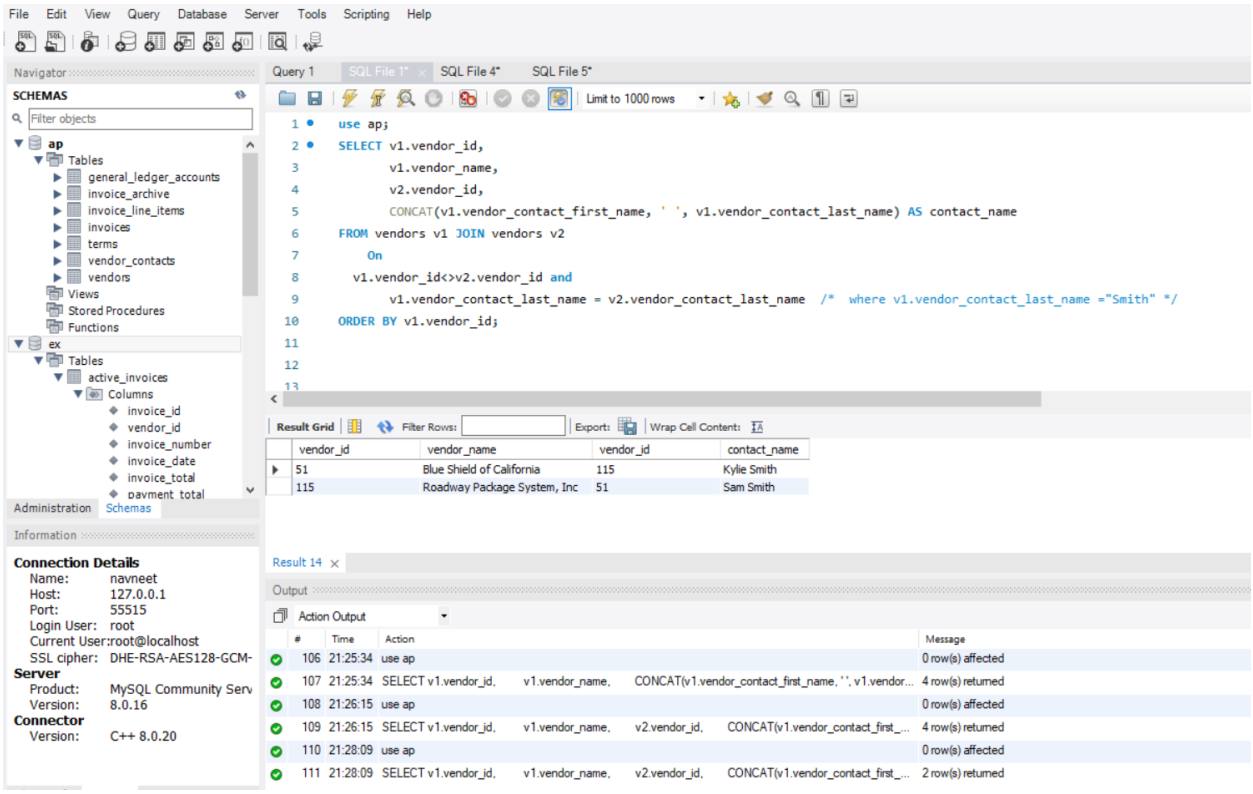
Return one row for each vendor whose contact has the same last name as another vendor's contact. This should return 2 rows. *Hint: Use a self-join to check that the vendor_id columns aren't equal but the vendor_contact_last_name columns are equal.*

Sort the result set by vendor_contact_last_name.

# Answer:

```
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
```






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.

## Answer:

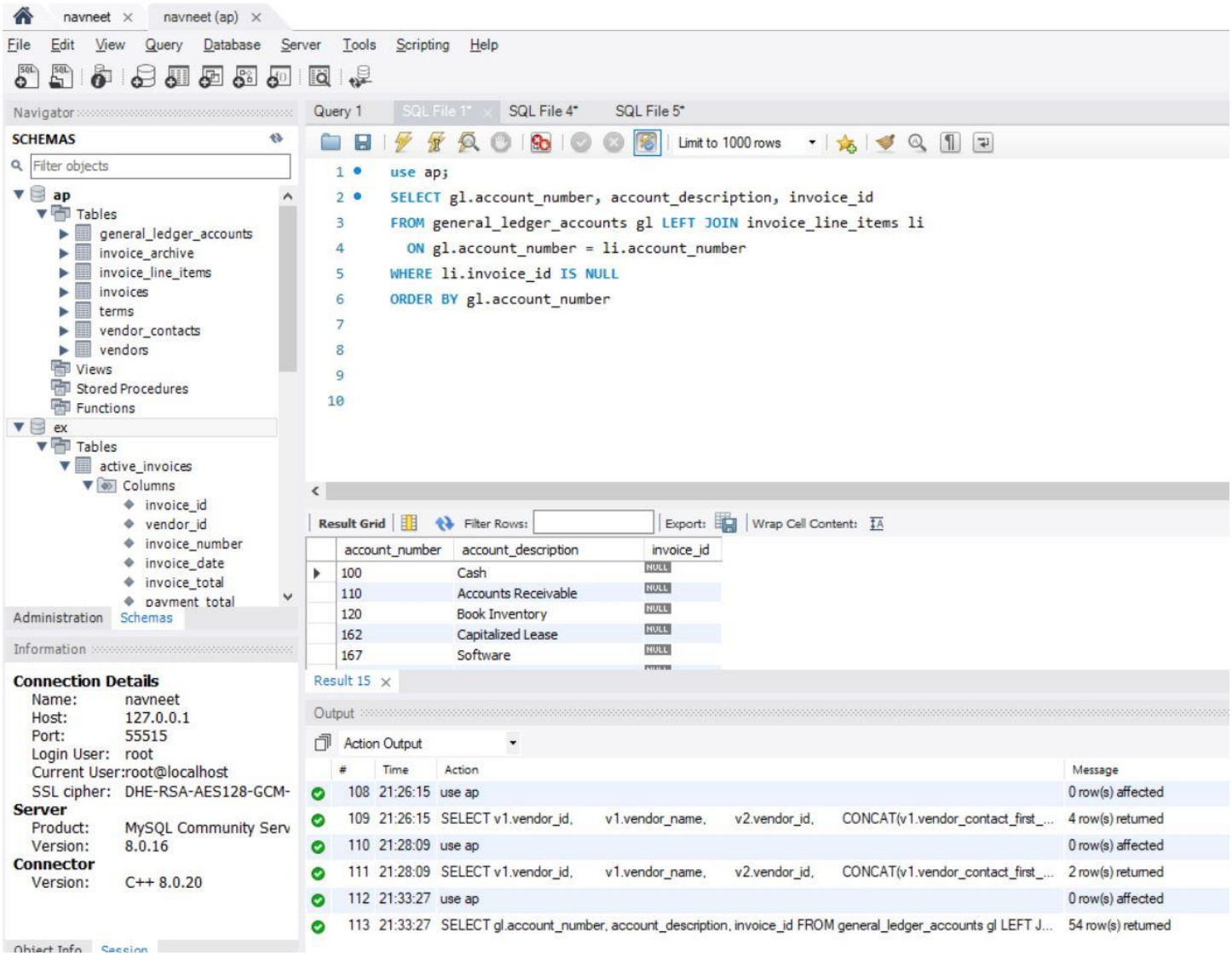
**SELECT** gl.account_number, account_description, invoice_id

**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




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.

**Answer:** **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