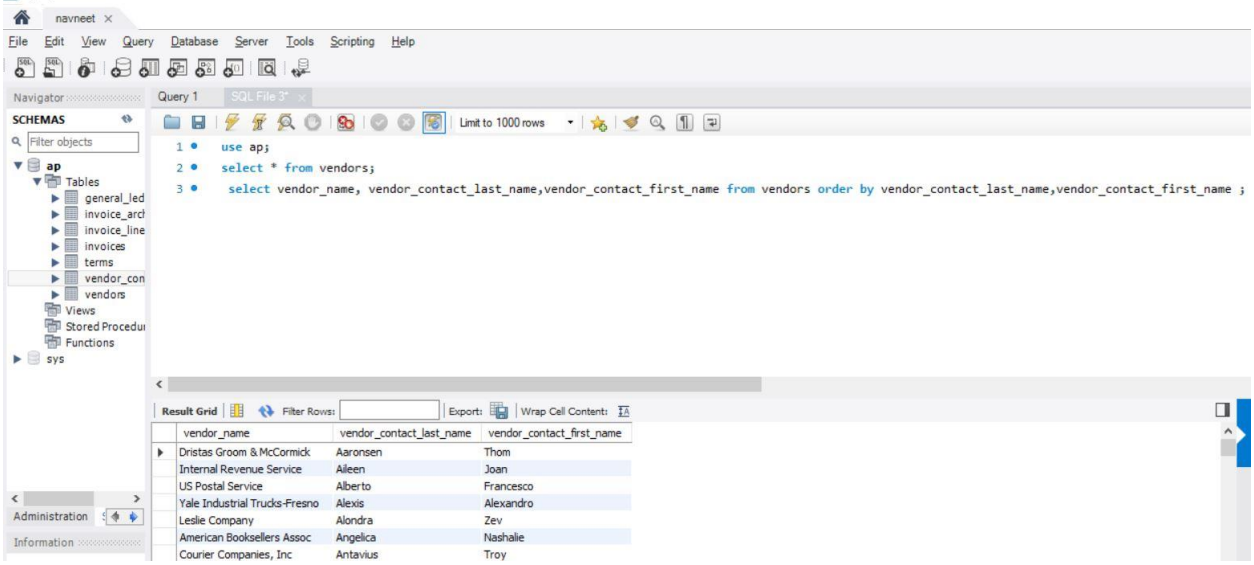
MYSQL LAB 3

6) Write a SELECT statement that returns three columns from the Vendors table: vendor_name, vendor_contact_last_name, and vendor_contact_first_name.

Then, run this statement to make sure it works correctly.

Add an ORDER BY clause to this statement that sorts the result set by last name and then first name, both in ascending sequence.



7) Write a SELECT statement that returns one column from the Vendors table named full_name that joins the vendor_contact_last_name and vendor_contact_first_name columns.

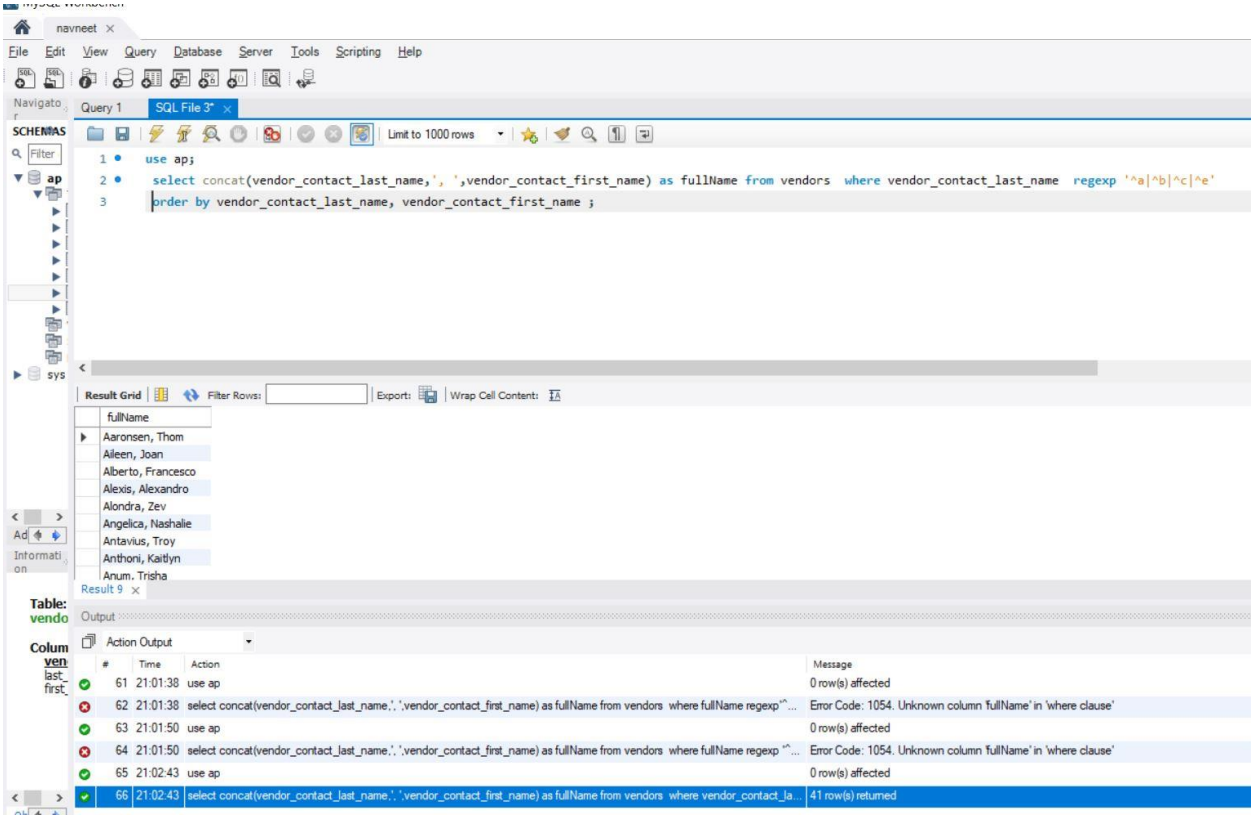
Format this column with the last name, a comma, a space, and the first name like this: Doe, John Sort the result set by last name and then first name in ascending sequence.

Return only the contacts whose last name begins with the letter A, B, C, or E. This should retrieve 41 rows

Answer:

```
select concat(vendor_contact_last_name,', ',vendor_contact_first_name) as fullName from vendors
where vendor_contact_last_name  regexp '^a|^b|^c|^e'

order by vendor_contact_last_name, vendor_contact_first_name ;
```



8) Write a SELECT statement that returns these column names and data from the Invoices table:

the Invoices table:

Due Date	The invoice_due_date column
Invoice Total	The invoice_total column
10%	10% of the value of invoice_total
Plus 10%	The value of invoice_total plus 10%

Return only the rows with an **invoice total** that's greater than or equal to 500 and less than or equal to 1000. This should retrieve 12 rows.

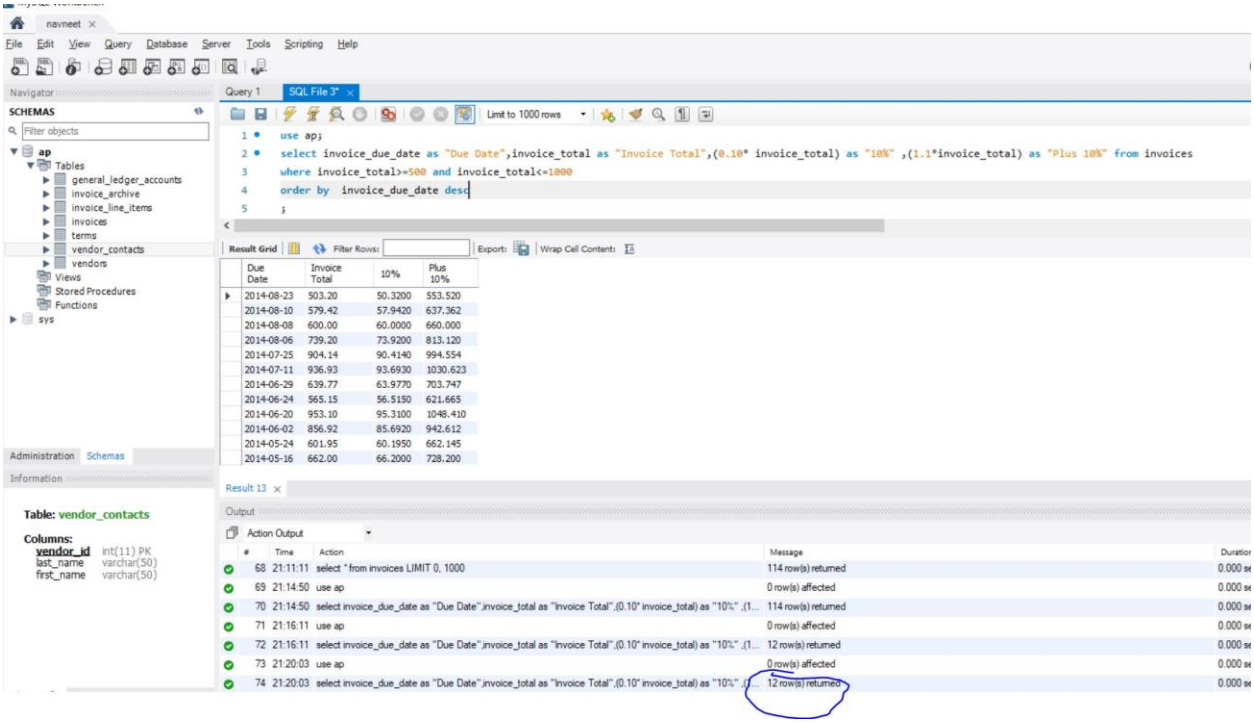
Sort the result set in descending sequence by invoice_due_date.

Answer:

select invoice_due_date as "Due Date",invoice_total as "Invoice Total",(0.10* invoice_total) as "10%" ,(1.1*invoice_total) as "Plus 10%" from invoices

where invoice_total>=500 and invoice_total<=1000

order by invoice_due_date desc



9) Write a SELECT statement that returns these columns from the Invoices table:

invoice_number	The invoice_number column
invoice_total	The invoice_total column
payment_credit_total	Sum of the payment_total and credit_total columns
balance_due	The invoice_total column minus the payment_total and credit_total columns

Return only invoices that have a balance due that's greater than $50.

Sort the result set by balance due in descending sequence.

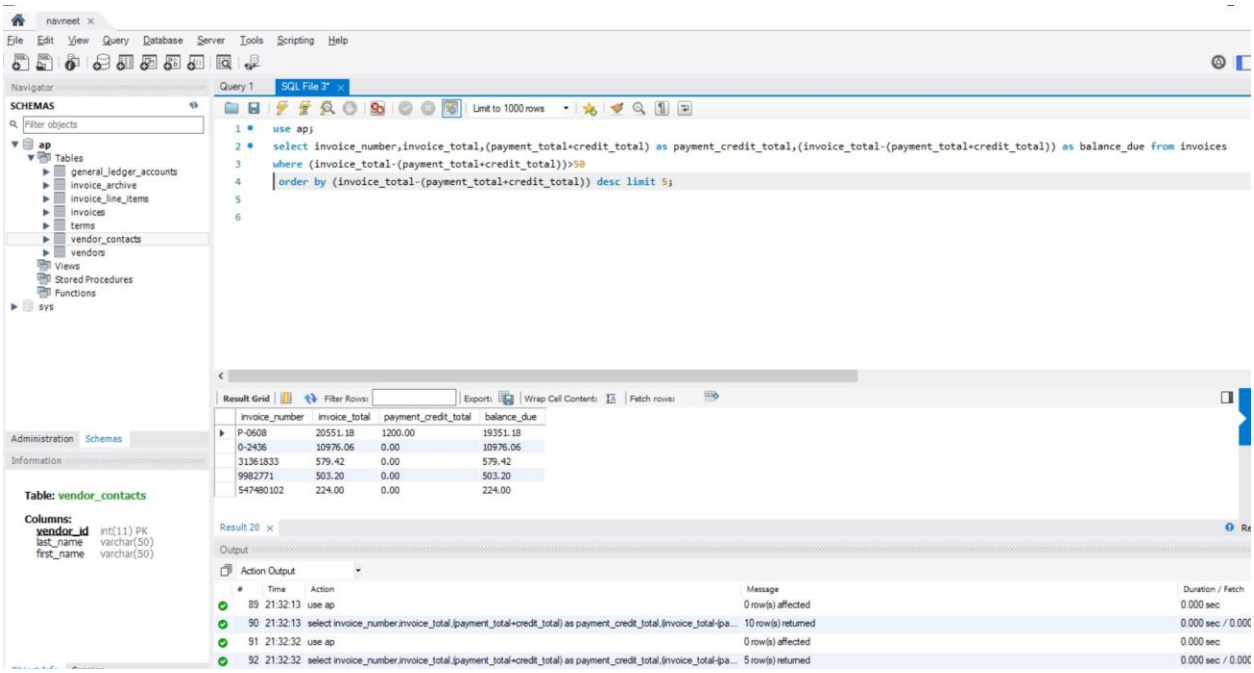
Use the LIMIT clause so the result set contains only the rows with the 5 largest balances.

Answer:

select invoice_number,invoice_total,(payment_total+credit_total) as payment_credit_total,(invoice_total-(payment_total+credit_total)) as balance_due from invoices

where (invoice_total-(payment_total+credit_total))>50

order by (invoice_total-(payment_total+credit_total)) desc limit 5;



Work with nulls and test expressions

10) Write a SELECT statement that returns these columns from the Invoices table

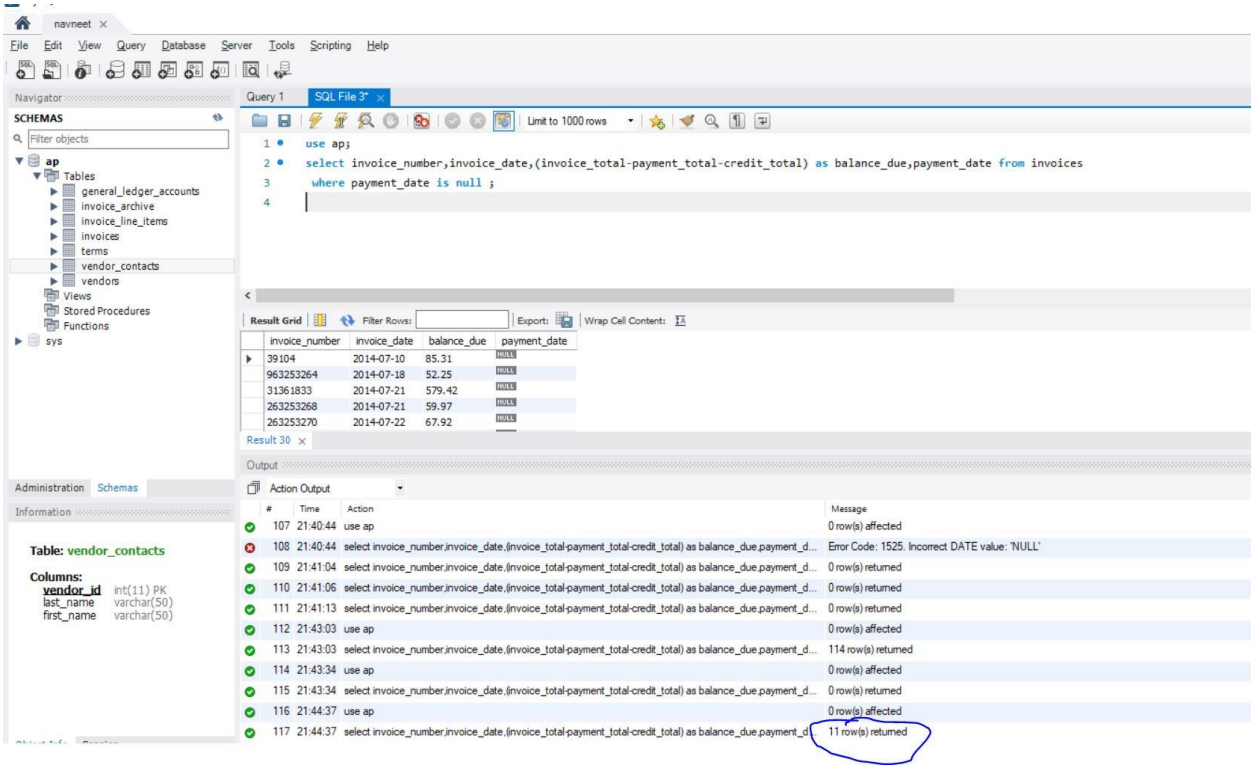
invoice_number	The invoice_number column
invoice_date	The invoice_date column
balance_due	The invoice_total column minus the payment_total and credit_total columns
payment_date	The payment_date column

Return only the rows where the payment_date column contains a null value. This should retrieve 11 rows.

Answer:

select invoice_number,invoice_date,(invoice_total-payment_total-credit_total) as balance_due,payment_date from invoices

where payment_date is null ;



11) Write a SELECT statement without a FROM clause that uses the CURRENT_DATE function to return the current date in its default format.

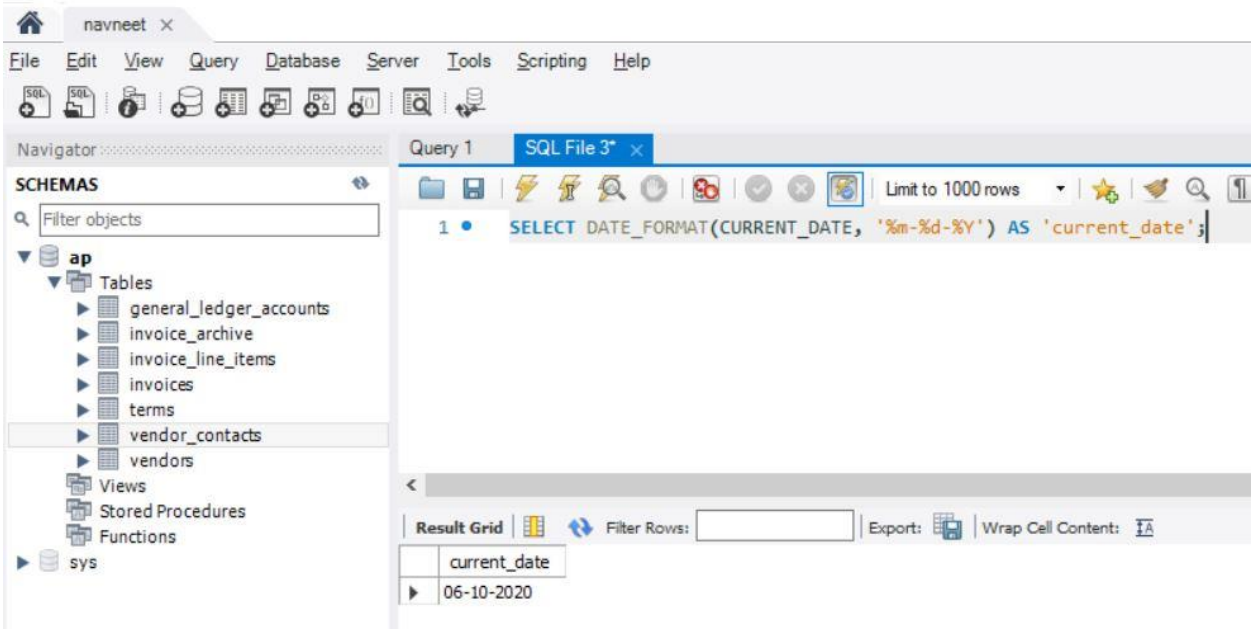
Use the DATE_FORMAT function to format the current date in this format: mm-dd-yyyy .

This displays the month, day, and four-digit year of the current date.

Give this column an alias of current_date.

To do that, you must enclose the alias in quotes since that name is already used by the CURRENT_DATE

Answer: SELECT DATE_FORMAT(CURRENT_DATE, '%m-%d-%Y') AS 'current_date';



12) Write a SELECT statement without a FROM clause that creates a row with these columns:

starting_principal	Starting principal of $50,000
interest	6.5% of the principal
principal_plus_interest	The principal plus the interest

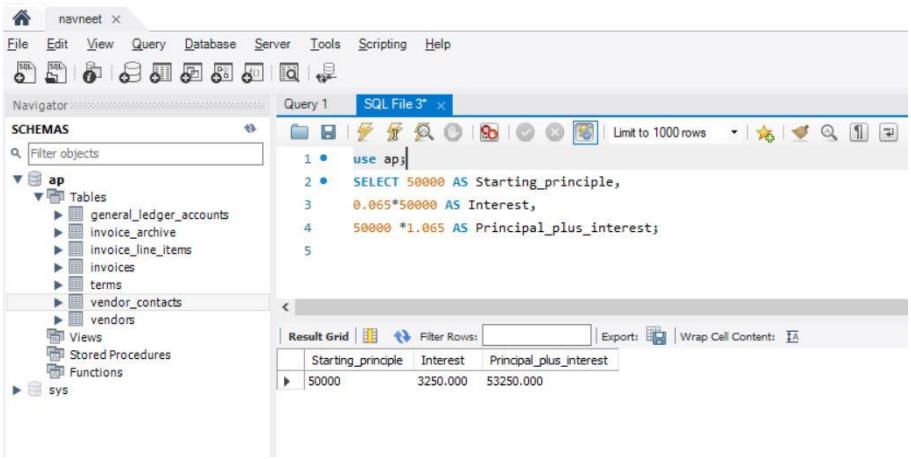
To calculate the third column, add the expressions you used for the first two columns.

Answer:

SELECT 50000 AS Starting_principle,

0.065*50000 AS Interest,

1.065*50000 AS Principal_plus_interest;

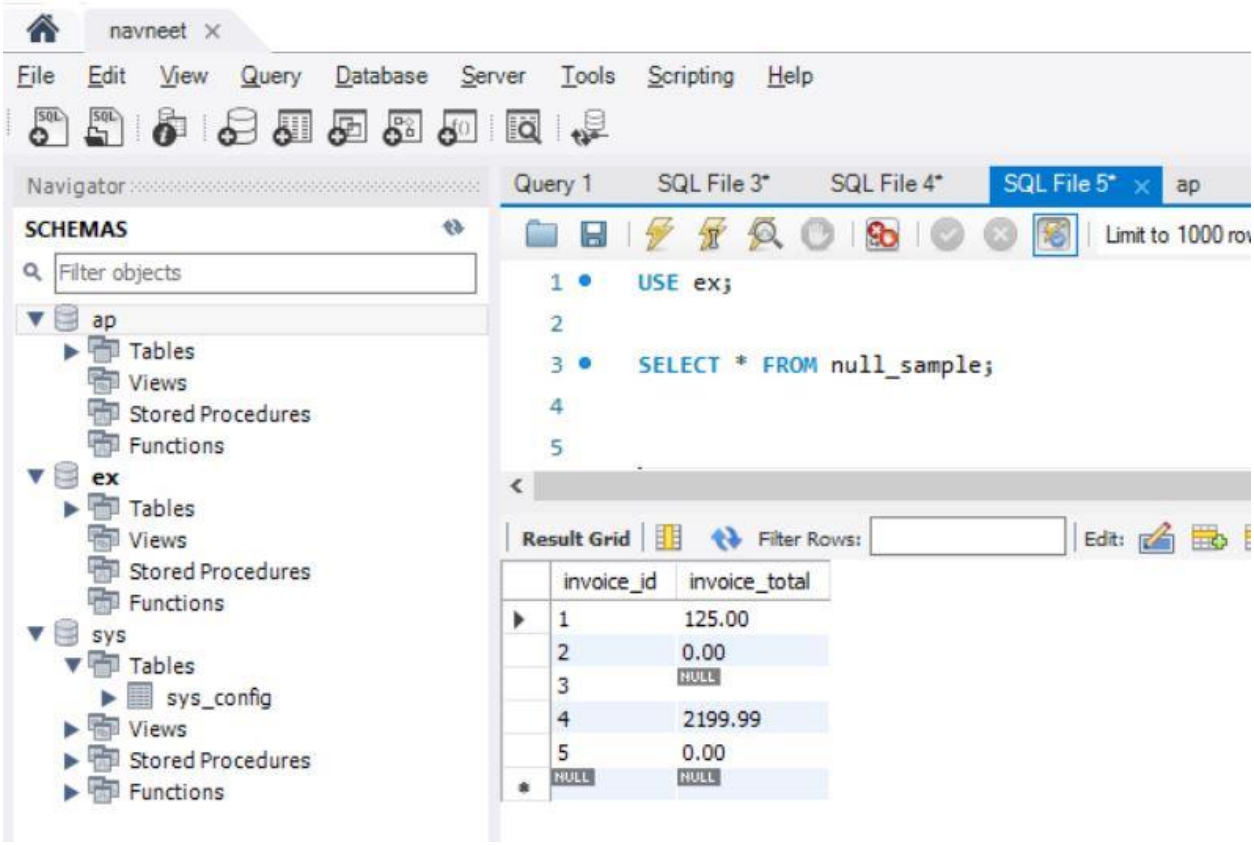


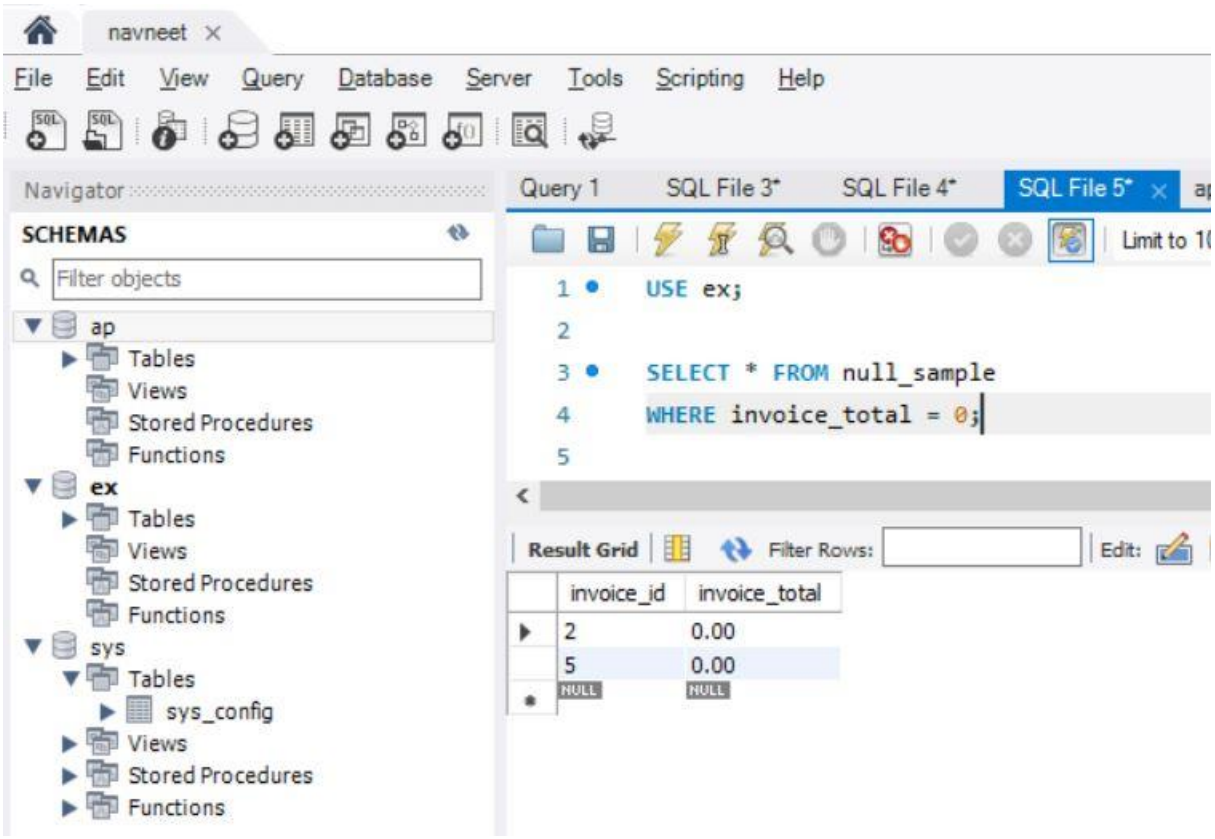
ANSWER FOR EXERCISE

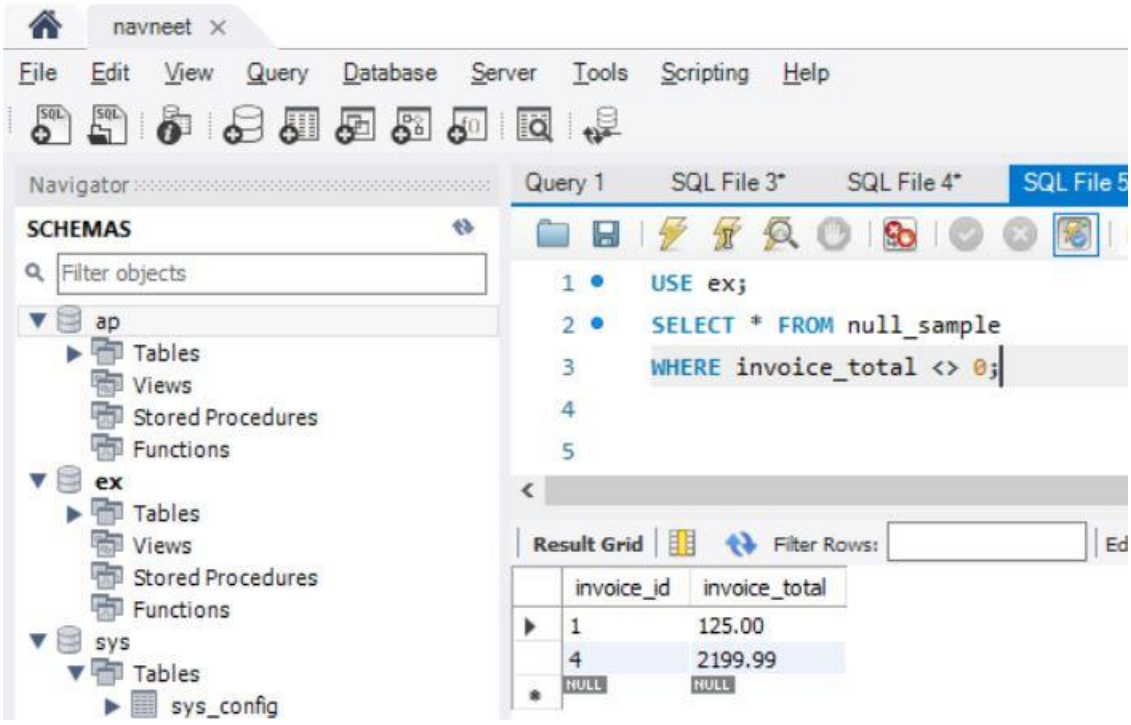
```
6) SELECT vendor_name, vendor_contact_last_name, vendor_contact_first_name
FROM vendors
ORDER BY vendor_contact_last_name, vendor_contact_first_name
7) SELECT CONCAT(vendor_contact_last_name, ', ', vendor_contact_first_name) AS full_name
FROM vendors
WHERE vendor_contact_last_name < 'D' OR vendor_contact_last_name LIKE 'E%'
ORDER BY vendor_contact_last_name, vendor_contact_first_name
8) SELECT invoice_due_date AS "Due Date",
    invoice_total AS "Invoice Total",
    invoice_total / 10 AS "10%",
    invoice_total * 1.1 AS "Plus 10%"
FROM invoices
WHERE invoice_total >= 500 AND invoice_total <= 1000
ORDER BY invoice_due_date DESC
9) SELECT invoice_number,
    invoice_total,
    payment_total + credit_total AS payment_credit_total,
    invoice_total - payment_total - credit_total AS balance_due
FROM invoices
WHERE invoice_total - payment_total - credit_total > 50
ORDER BY balance_due DESC
LIMIT 5;
10) SELECT invoice_number,
    invoice_date,
    invoice_total - payment_total - credit_total AS balance_due,
    payment_date
FROM invoices
WHERE payment_date IS NULL
11) SELECT DATE_FORMAT(CURRENT_DATE, '%m-%d-%Y') AS "current_date"
12) SELECT 50000 AS starting_principle,
    50000 * .065 AS interest,
    (50000) + (50000 * .065) AS principle_plus_interest
```

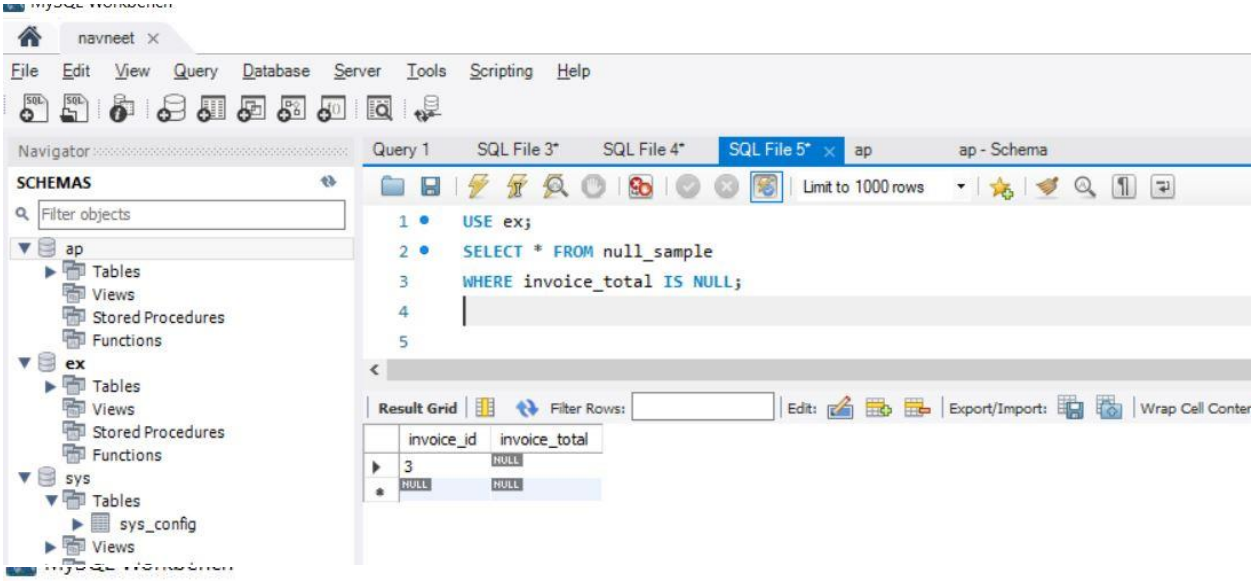
USE EX DATABASE

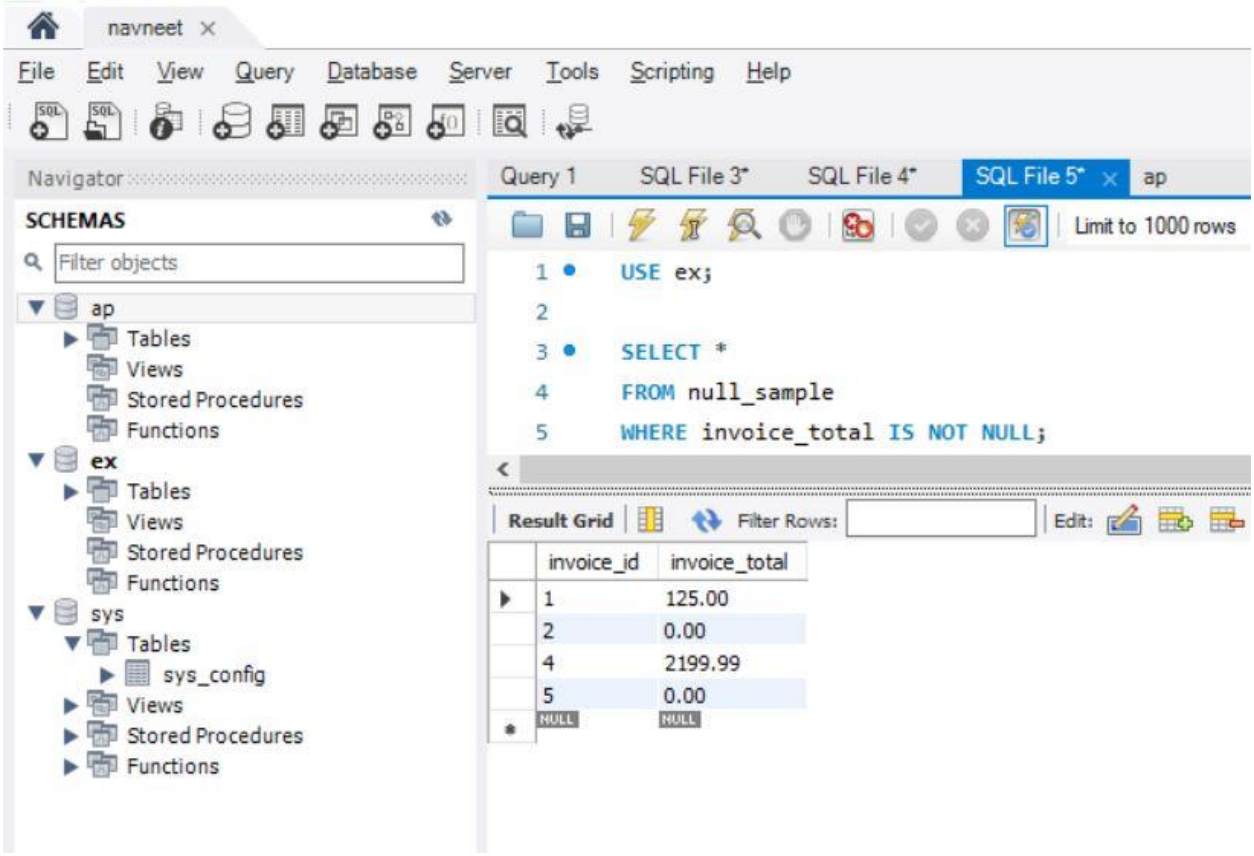
1)IS NULL




navneet
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ap
Tables
Views
Stored Procedures
Functions
ex
Tables
Views
Stored Procedures
Functions
sys
Tables
sys_config
Views
Stored Procedures
Functions
Query 1
SQL File 3*
SQL File 4*
SQL File 5*
Limit to 1
1 USE ex;
2
3 SELECT * FROM null_sample
4 WHERE invoice_total = 0;
5
Result Grid
Filter Rows:
Edit:
invoice_id
invoice_total
2
0.00
5
0.00
NULL
NULL


navneet
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ap
Tables
Views
Stored Procedures
Functions
ex
Tables
Views
Stored Procedures
Functions
sys
Tables
sys_config
Query 1
SQL File 3*
SQL File 4*
SQL File 5
1 USE ex;
2 SELECT * FROM null_sample
3 WHERE invoice_total <> 0;
4
5
Result Grid
Filter Rows:
invoice_id
invoice_total
1
125.00
4
2199.99
NULL
NULL


navneet
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ap
Tables
Views
Stored Procedures
Functions
ex
Tables
Views
Stored Procedures
Functions
sys
Tables
sys_config
Views
Query 1
SQL File 3*
SQL File 4*
SQL File 5*
ap
ap - Schema
Limit to 1000 rows
1 USE ex;
2 SELECT * FROM null_sample
3 WHERE invoice_total IS NULL;
4
5
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Conter
invoice_id
invoice_total
3
NULL
NULL
NULL


navneet
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ap
Tables
Views
Stored Procedures
Functions
ex
Tables
Views
Stored Procedures
Functions
sys
Tables
sys_config
Views
Stored Procedures
Functions
Query 1
SQL File 3*
SQL File 4*
SQL File 5*
ap
Limit to 1000 rows
1 USE ex;
2
3 SELECT *
4 FROM null_sample
5 WHERE invoice_total IS NOT NULL;
Result Grid
Filter Rows:
Edit:
invoice_id
invoice_total
1
125.00
2
0.00
4
2199.99
5
0.00
NULL
NULL