CS411, Winter 2023

Homework 2. Due by midnight Feb 7

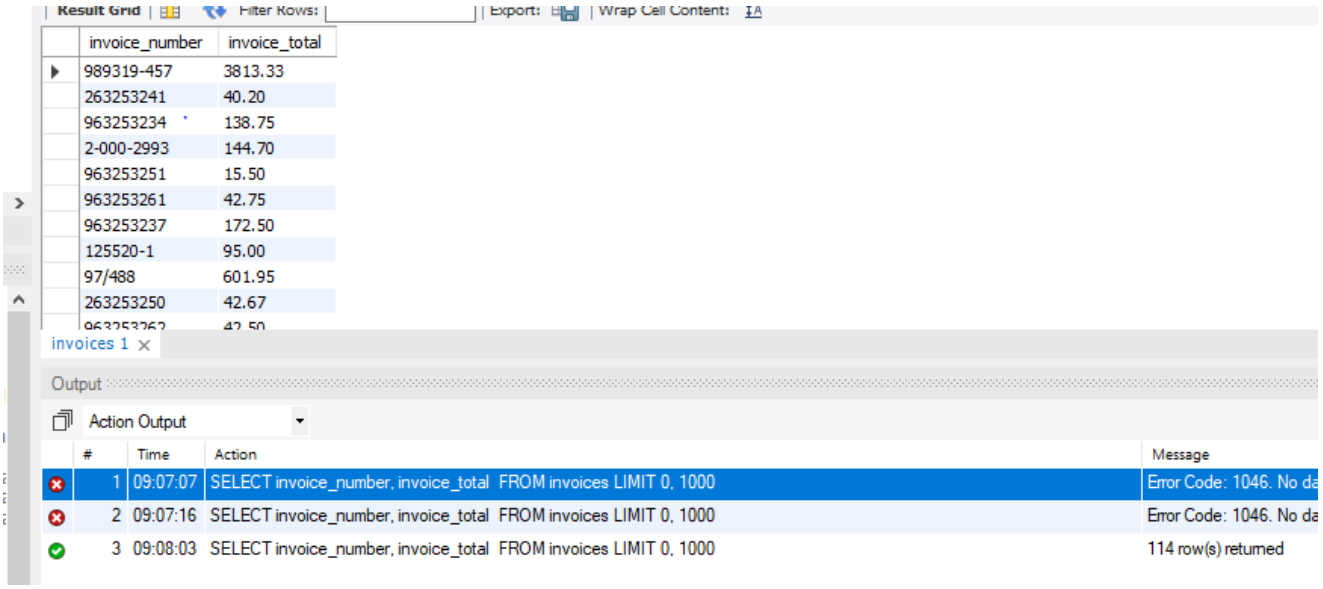
Follow the instructions and provide the result. All the problems are based on the ap database. You should show your query in text (so that I can copy/paste and run) and screenshots of the result. Make sure that you only copy the result part so that it can be easily read. You do not need to show all rows of the result. If the result has more than 10 rows, just show first 10 rows. For each result, show the total number of rows returned.

If there is more than one question in each problem you should show the result of each question. Show the problem number, problem description, and your answer for each question.

1. **(5) Write a SELECT statement that shows invoice_number and invoice_total from the invoices table.**

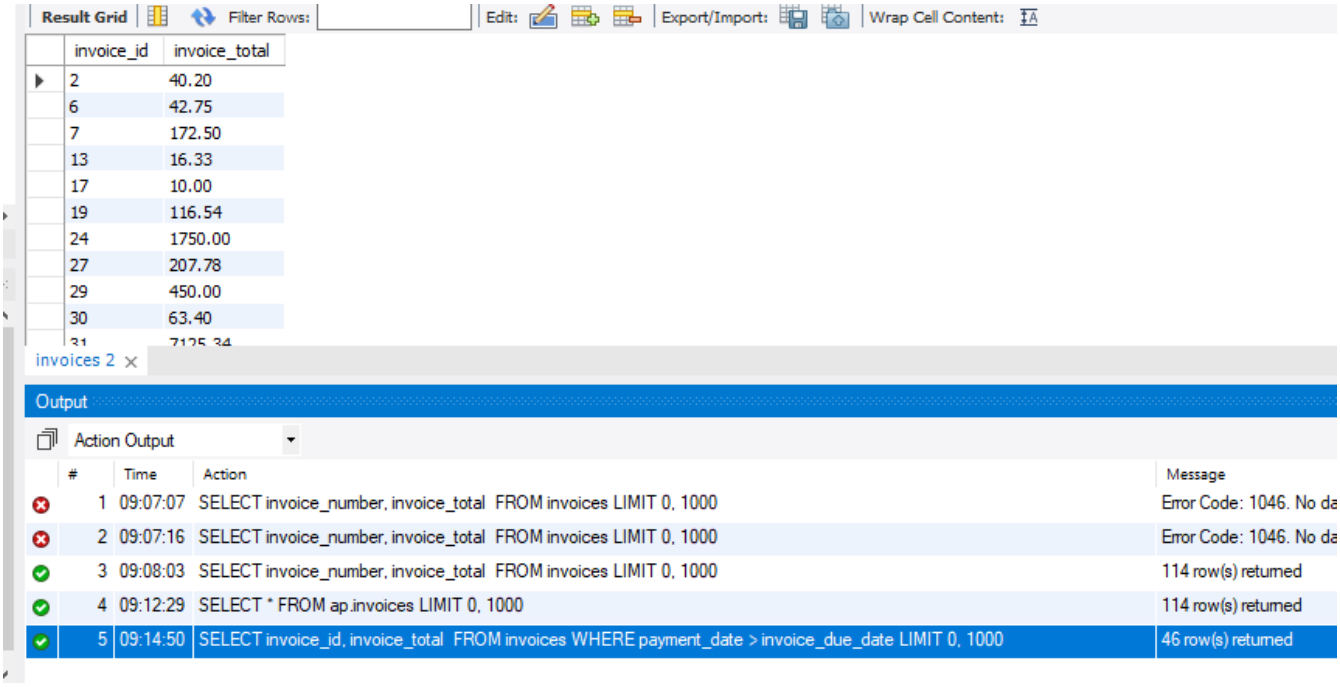
   SELECT invoice_number, invoice_total

   FROM invoices



2. **(5) Write a SELECT statement that shows invoice_id and invoice_total where the payment is made past the due date.**

```
SELECT invoice_id, invoice_total

FROM invoices

WHERE payment_date > invoice_due_date
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

invoice_id	invoice_total
2	40.20
6	42.75
7	172.50
13	16.33
17	10.00
19	116.54
24	1750.00
27	207.78
29	450.00
30	63.40
31	7125.34

invoices 2

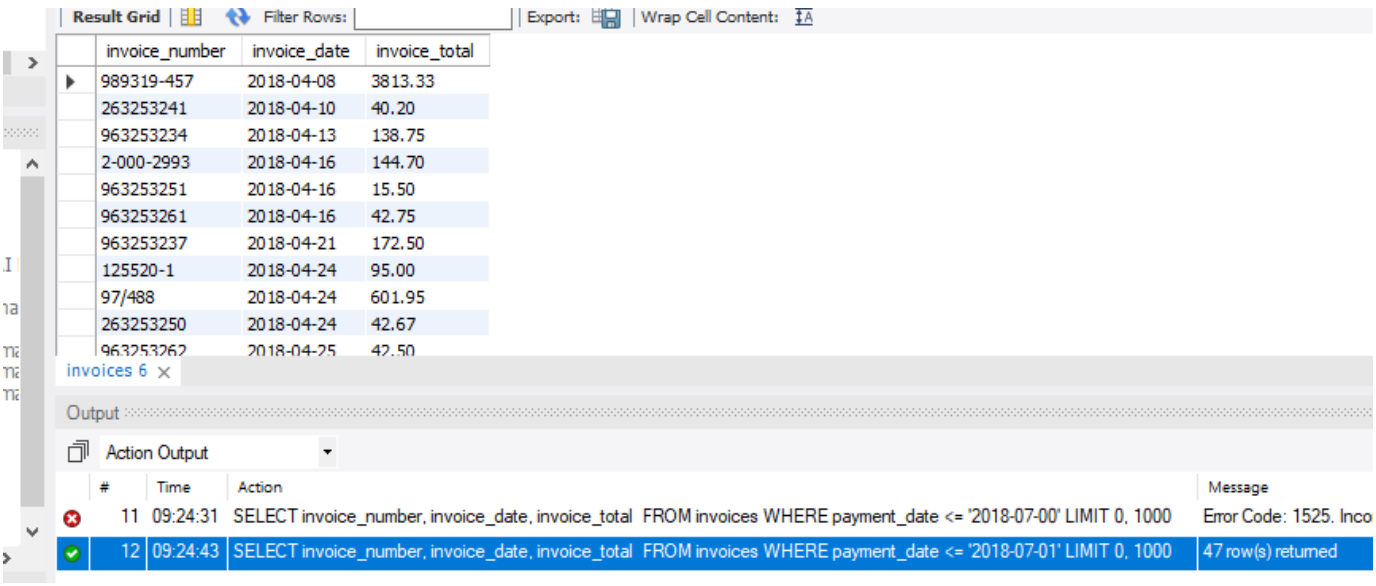
Output

Action Output

#	Time	Action	Message
1	09:07:07	SELECT invoice_number, invoice_total FROM invoices LIMIT 0, 1000	Error Code: 1046. No da
2	09:07:16	SELECT invoice_number, invoice_total FROM invoices LIMIT 0, 1000	Error Code: 1046. No da
3	09:08:03	SELECT invoice_number, invoice_total FROM invoices LIMIT 0, 1000	114 row(s) returned
4	09:12:29	SELECT * FROM ap.invoices LIMIT 0, 1000	114 row(s) returned
5	09:14:50	SELECT invoice_id, invoice_total FROM invoices WHERE payment_date > invoice_due_date LIMIT 0, 1000	46 row(s) returned

3. **(5) Write a SELECT statement that shows invoice_number, invoice_date, and invoice_total where the payment is made on or before July 2018.**

```
SELECT invoice_number, invoice_date, invoice_total

FROM invoices

WHERE payment_date <= '2018-07-01'
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

invoice_number	invoice_date	invoice_total
989319-457	2018-04-08	3813.33
263253241	2018-04-10	40.20
963253234	2018-04-13	138.75
2-000-2993	2018-04-16	144.70
963253251	2018-04-16	15.50
963253261	2018-04-16	42.75
963253237	2018-04-21	172.50
125520-1	2018-04-24	95.00
97/488	2018-04-24	601.95
263253250	2018-04-24	42.67
963253262	2018-04-25	42.50

invoices 6

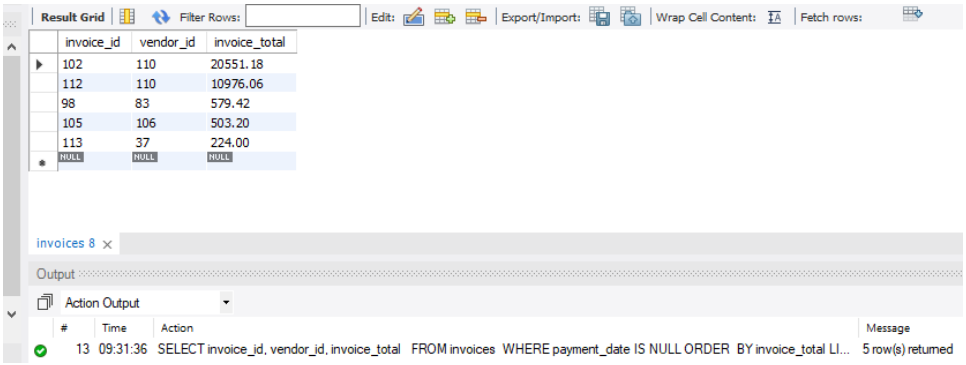
Output

Action Output

#	Time	Action	Message
11	09:24:31	SELECT invoice_number, invoice_date, invoice_total FROM invoices WHERE payment_date <= '2018-07-00' LIMIT 0, 1000	Error Code: 1525. Inco
12	09:24:43	SELECT invoice_number, invoice_date, invoice_total FROM invoices WHERE payment_date <= '2018-07-01' LIMIT 0, 1000	47 row(s) returned

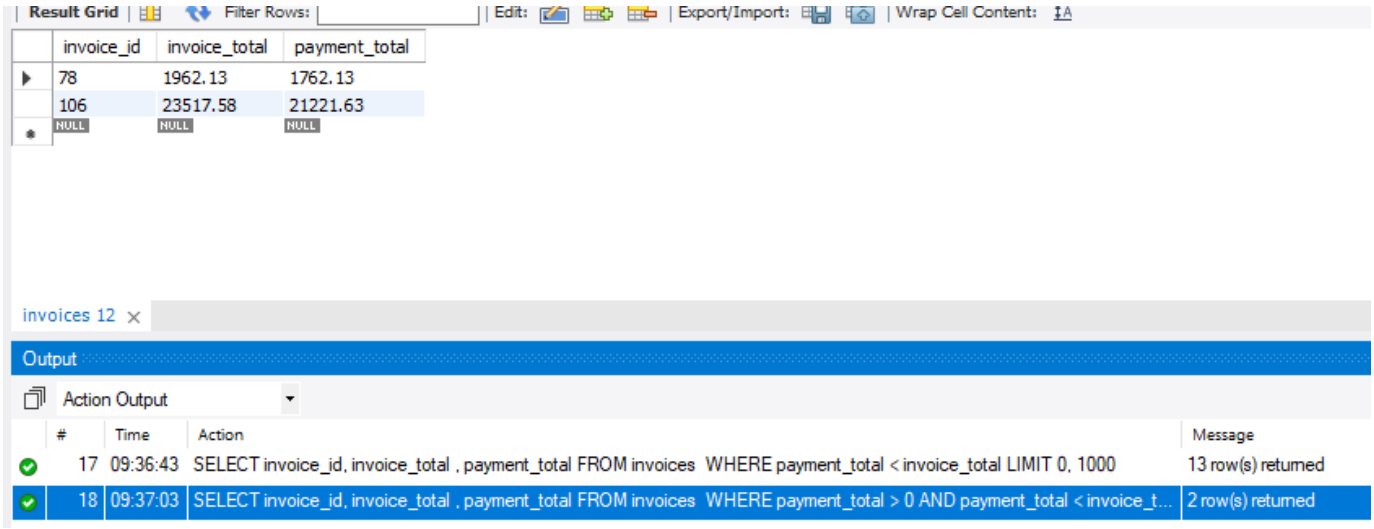
4. **(5) Write a SELECT statement that shows invoice_id, vendor_id, and invoice_total where the payment is not made. Use the LIMIT clause so the result set contains only the rows with the 5 largest invoice_total.**

```
SELECT invoice_id, vendor_id, invoice_total

FROM invoices

WHERE payment_date IS NULL

ORDER  BY invoice_total DESC

LIMIT 5
```



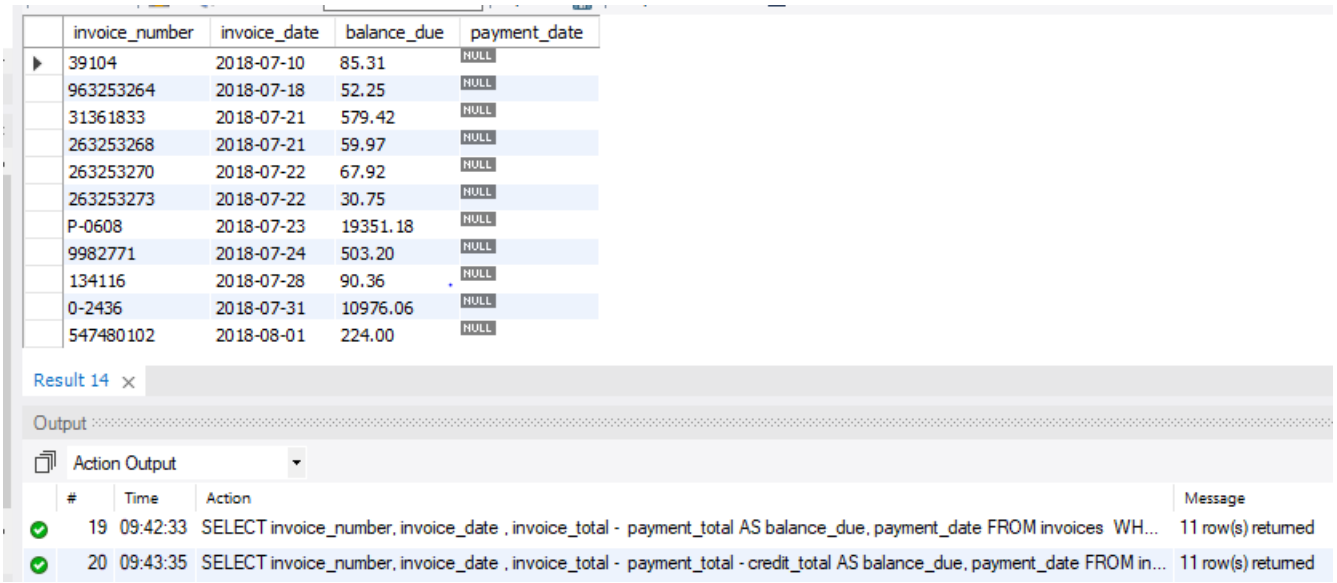
5. **(5) Write a SELECT statement that shows invoice_id, invoice_total, and payment_total where the payment is made partial, that is, the payment is made but not made in full.**

```
SELECT invoice_id, invoice_total , payment_total

FROM invoices

WHERE payment_total > 0 AND payment_total < invoice_total
```



6. **(10) Write a SELECT statement that returns invoice_number, invoice_date, balance_due, and payment_date from the Invoices table. The balance_due is the invoice_total column minus the payment_total and credit_total columns.**
   Return only the rows where the payment_date column contains a null value.

```
SELECT invoice_number, invoice_date , invoice_total -  payment_total - credit_total AS balance_due,
payment_date

FROM invoices

WHERE payment_date IS NULL
```



invoice_number	invoice_date	balance_due	payment_date
39104	2018-07-10	85.31	NULL
963253264	2018-07-18	52.25	NULL
31361833	2018-07-21	579.42	NULL
263253268	2018-07-21	59.97	NULL
263253270	2018-07-22	67.92	NULL
263253273	2018-07-22	30.75	NULL
P-0608	2018-07-23	19351.18	NULL
9982771	2018-07-24	503.20	NULL
134116	2018-07-28	90.36	NULL
0-2436	2018-07-31	10976.06	NULL
547480102	2018-08-01	224.00	NULL

Result 14

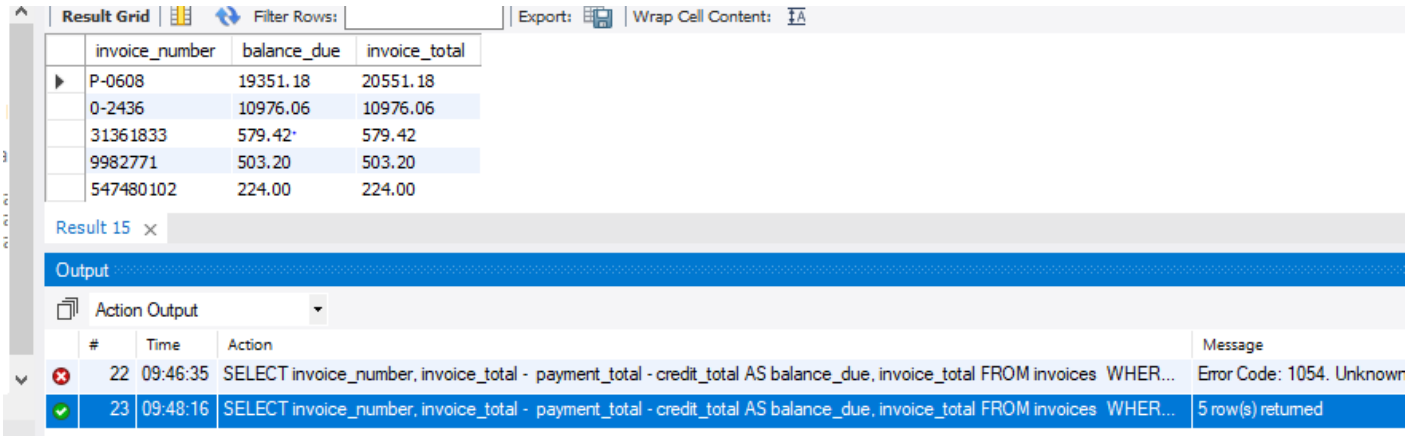
Output

Action Output

#	Time	Action	Message
19	09:42:33	SELECT invoice_number, invoice_date , invoice_total - payment_total AS balance_due, payment_date FROM invoices WH...	11 row(s) returned
20	09:43:35	SELECT invoice_number, invoice_date , invoice_total - payment_total - credit_total AS balance_due, payment_date FROM in...	11 row(s) returned

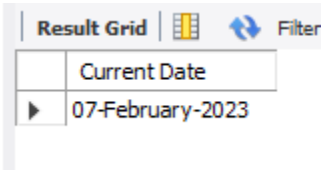
7. **(5) Write a SELECT statement to show the invoice number, invoice_total, and balance due where the balance due is greater than 100. Sort the result by the balance due in descending order.**

```
SELECT invoice_number, invoice_total - payment_total - credit_total AS balance_due, invoice_total
FROM invoices
WHERE invoice_total - payment_total - credit_total > 100
ORDER BY invoice_total - payment_total - credit_total DESC
```



8. **(5) Write a SELECT statement without a FROM clause that uses the CURRENT_DATE function to return the current date in `DD-Mon-YYYY` format. Use the DATE_FORMAT function. This displays the day, month, and four-digit year of the current date. Give this column an alias of 'Current Date'.**

```
SELECT DATE_FORMAT(current_date, "%d-%M-%Y") AS 'Current Date'
```



9. **(10) Write a SELECT statement without a FROM clause that creates a row with these columns: starting_principal Starting principal of $200,000**
   interest 2.5% 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.

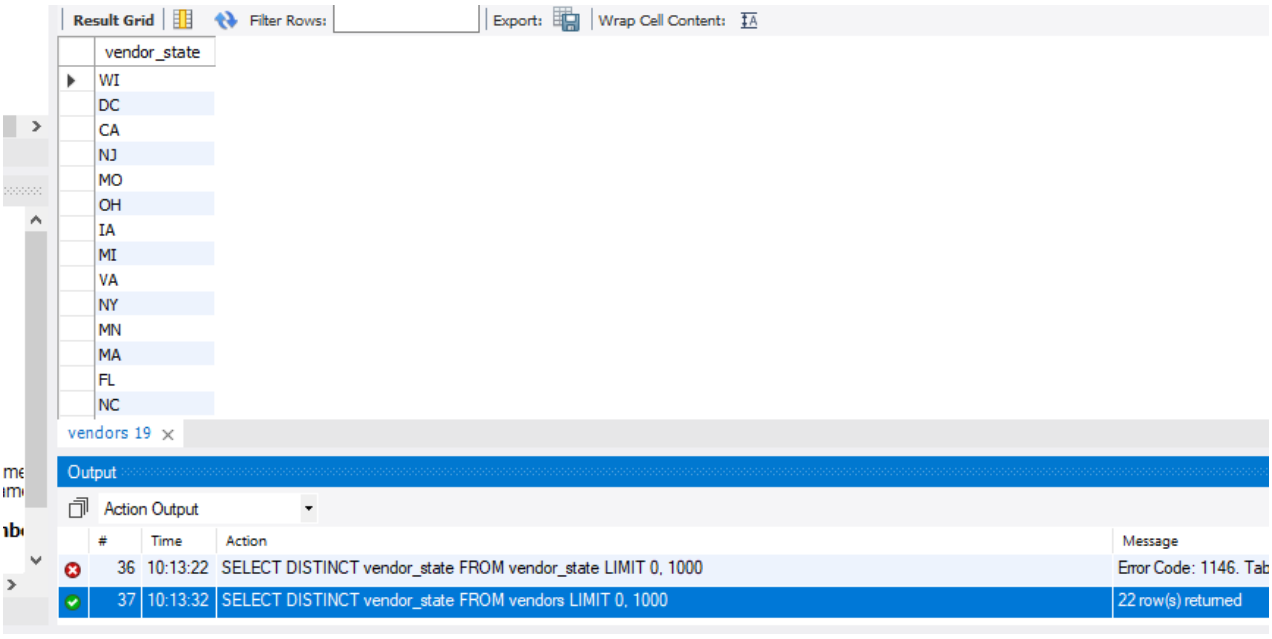
```
SELECT 200000 AS starting_principle, 0.025 * 200000 AS interest, 200000 + (0.025 * 200000) AS
principle_plus_interest
```

Result Grid | Filter Rows: | Expor

starting_principle	interest	principle_plus_interest
200000	5000.000	205000.000

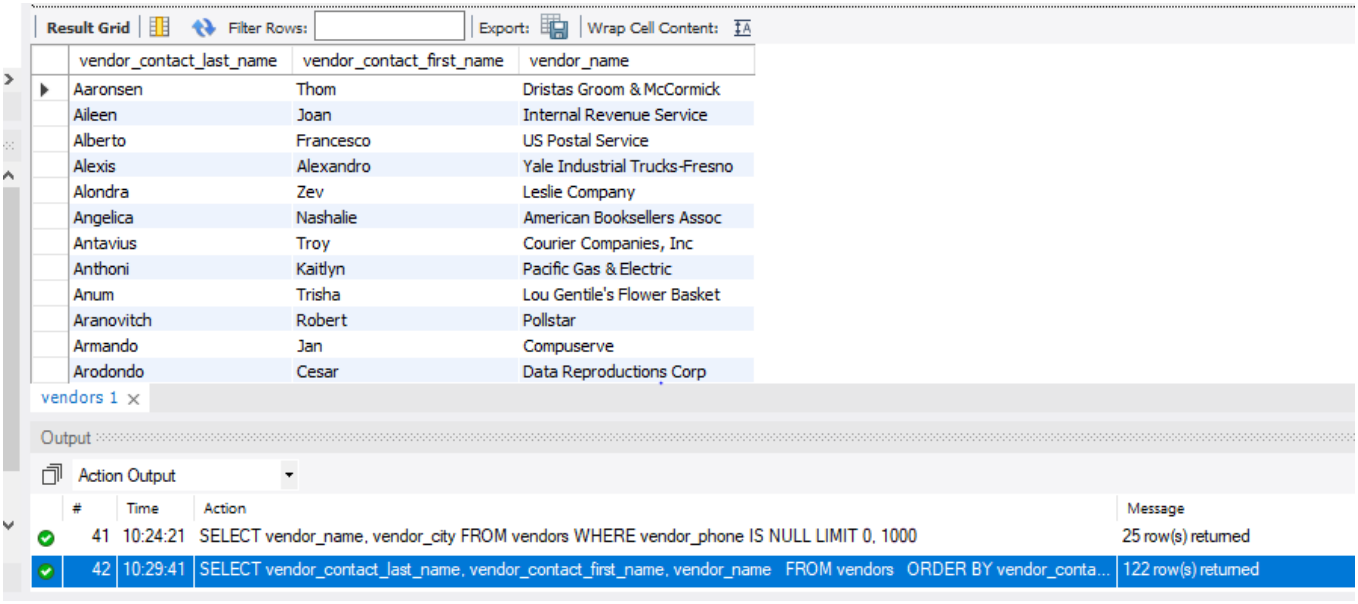
10. **(5) Write a SELECT statement to show all the states where vendors are located from the vendors table. Avoid duplicate state names.**

```
SELECT DISTINCT vendor_state

FROM vendors
```



11. **(5) Write a SELECT statement that returns vendor_name, vendor_contact_last_name, and vendor_contact_first_name from the vendors table, with the result ordered by last name and then first name.**

```
SELECT vendor_contact_last_name, vendor_contact_first_name, vendor_name
FROM vendors
ORDER BY vendor_contact_last_name, vendor_contact_first_name ASC
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_contact_last_name	vendor_contact_first_name	vendor_name
Aaronsen	Thom	Dristas Groom & McCormick
Aileen	Joan	Internal Revenue Service
Alberto	Francesco	US Postal Service
Alexis	Alexandro	Yale Industrial Trucks-Fresno
Alondra	Zev	Leslie Company
Angelica	Nashalie	American Booksellers Assoc
Antavius	Troy	Courier Companies, Inc
Anthoni	Kaitlyn	Pacific Gas & Electric
Anum	Trisha	Lou Gentile's Flower Basket
Aranovitch	Robert	Pollstar
Armando	Jan	Compuserve
Arodondo	Cesar	Data Reproductions Corp

vendors 1

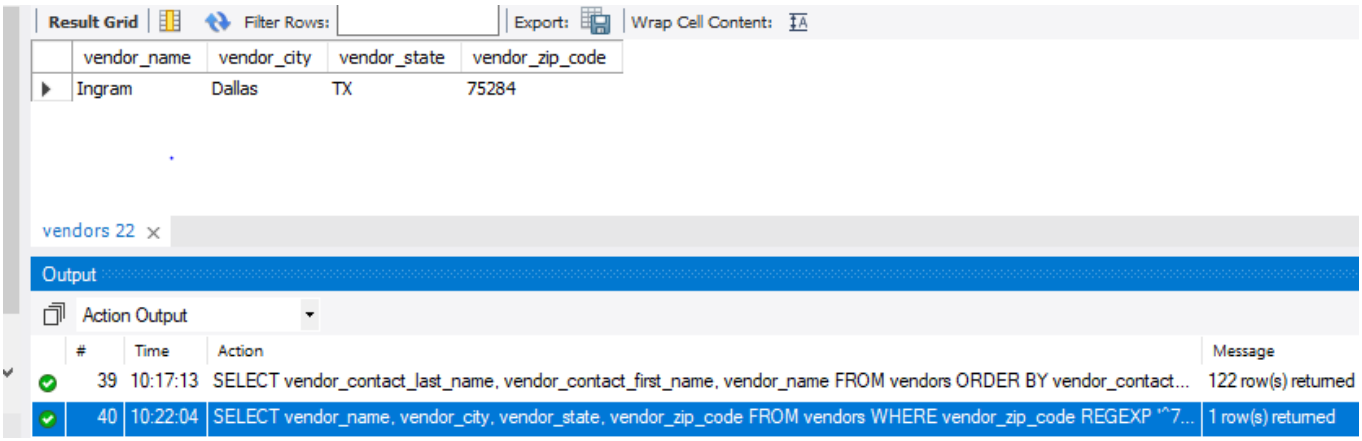
Output

Action Output

#	Time	Action	Message
41	10:24:21	SELECT vendor_name, vendor_city FROM vendors WHERE vendor_phone IS NULL LIMIT 0, 1000	25 row(s) returned
42	10:29:41	SELECT vendor_contact_last_name, vendor_contact_first_name, vendor_name FROM vendors ORDER BY vendor_conta...	122 row(s) returned

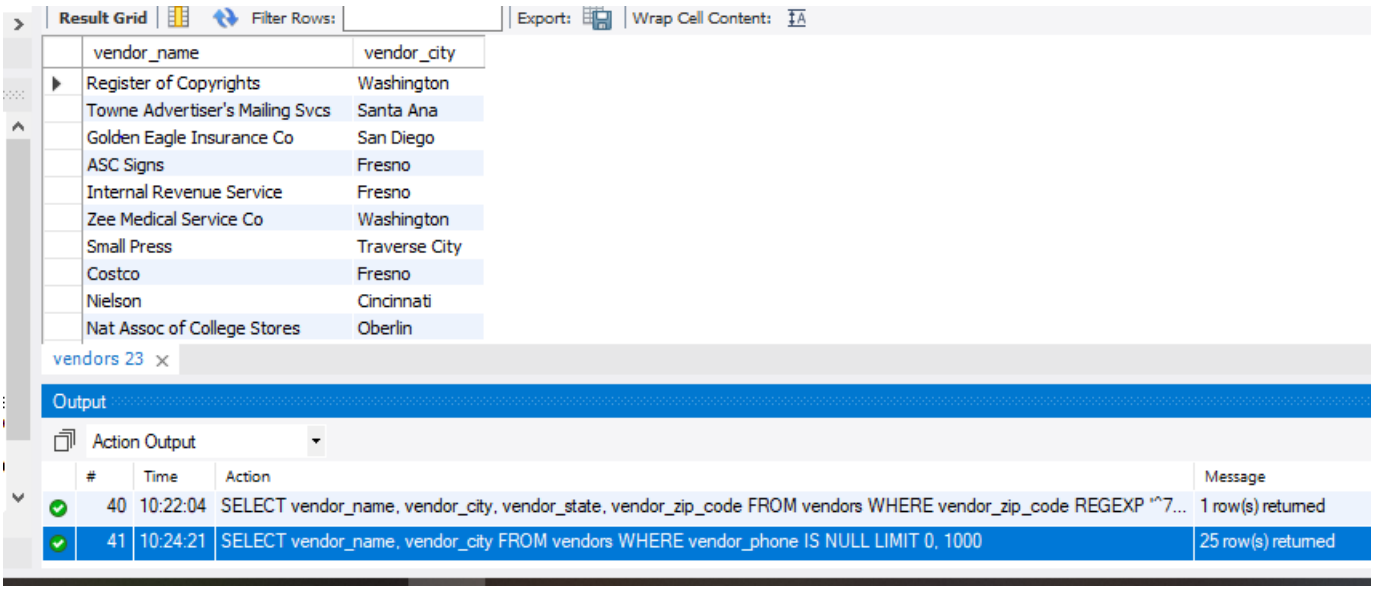
12. **(5) Write a SELECT statement to show the vendor name, vendor city, vendor state, and vendor zipcode where the vendor zip code starts with 9.**

```
SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code

FROM vendors

WHERE vendor_zip_code REGEXP '^7'
```



13. **(5) Write a SELECT statement to show the vendor name and vendor city where there is no phone number for the vendor.**

```
SELECT vendor_name, vendor_city

FROM vendors

WHERE vendor_phone IS NULL
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_name	vendor_city
Register of Copyrights	Washington
Towne Advertiser's Mailing Svcs	Santa Ana
Golden Eagle Insurance Co	San Diego
ASC Signs	Fresno
Internal Revenue Service	Fresno
Zee Medical Service Co	Washington
Small Press	Traverse City
Costco	Fresno
Nielson	Cincinnati
Nat Assoc of College Stores	Oberlin

vendors 23

Output

Action Output

#	Time	Action	Message
40	10:22:04	SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code FROM vendors WHERE vendor_zip_code REGEXP '^7...	1 row(s) returned
41	10:24:21	SELECT vendor_name, vendor_city FROM vendors WHERE vendor_phone IS NULL LIMIT 0, 1000	25 row(s) returned

14. **(10) Write a SELECT statement that returns one column from the Vendors table named full_name that joins the vendor_contact_first_name and vendor_contact_last_name columns. Format this column with the first name, a space, and the last name like this: "Jane Doe"**

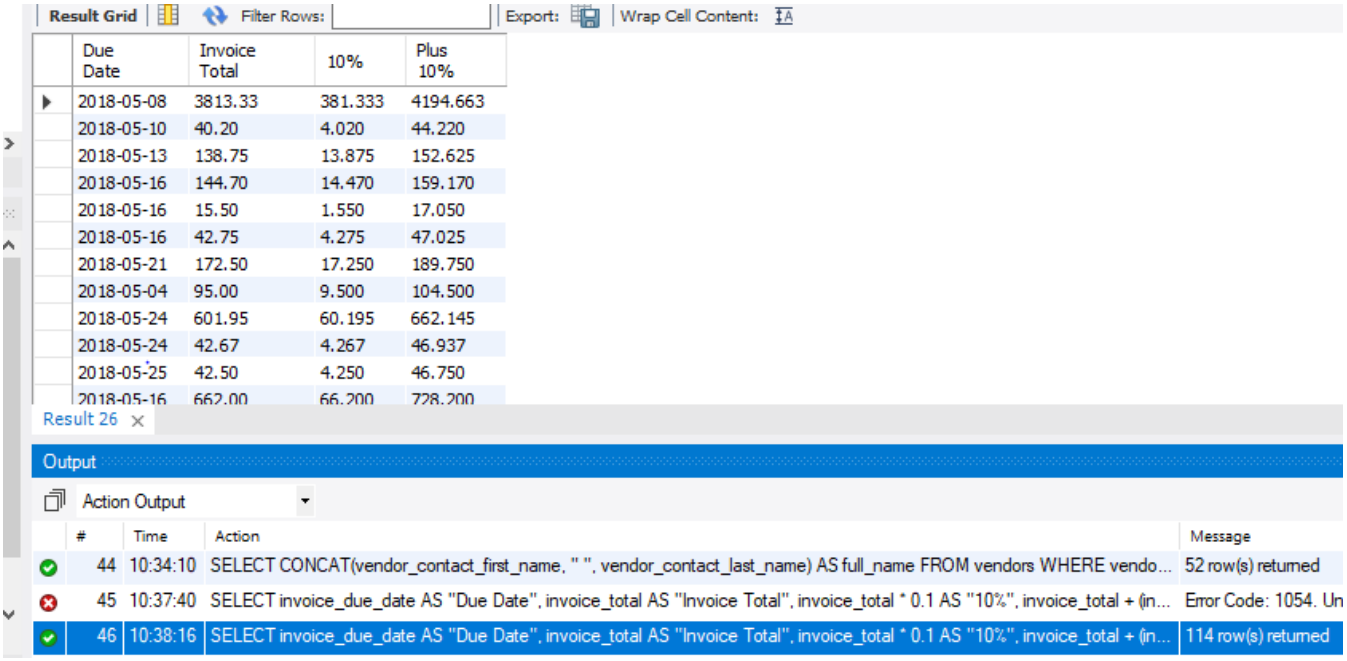
    Sort the result set by last name and then first name. Return only the contacts whose last name starts with A through H.

```
SELECT CONCAT(vendor_contact_first_name, " ", vendor_contact_last_name) AS full_name

FROM vendors

WHERE vendor_contact_last_name BETWEEN 'A' AND 'H'

ORDER BY vendor_contact_last_name, vendor_contact_first_name ASC
```

15. (10) Write a SELECT statement that returns these column names and data from 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%**

```
SELECT invoice_due_date AS "Due Date", invoice_total AS "Invoice Total", invoice_total * 0.1 AS "10%",
invoice_total + (invoice_total * 0.1)  AS "Plus 10%"

FROM invoices
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Due Date	Invoice Total	10%	Plus 10%
2018-05-08	3813.33	381.333	4194.663
2018-05-10	40.20	4.020	44.220
2018-05-13	138.75	13.875	152.625
2018-05-16	144.70	14.470	159.170
2018-05-16	15.50	1.550	17.050
2018-05-16	42.75	4.275	47.025
2018-05-21	172.50	17.250	189.750
2018-05-04	95.00	9.500	104.500
2018-05-24	601.95	60.195	662.145
2018-05-24	42.67	4.267	46.937
2018-05-25	42.50	4.250	46.750
2018-05-16	662.00	66.200	728.200

Result 26

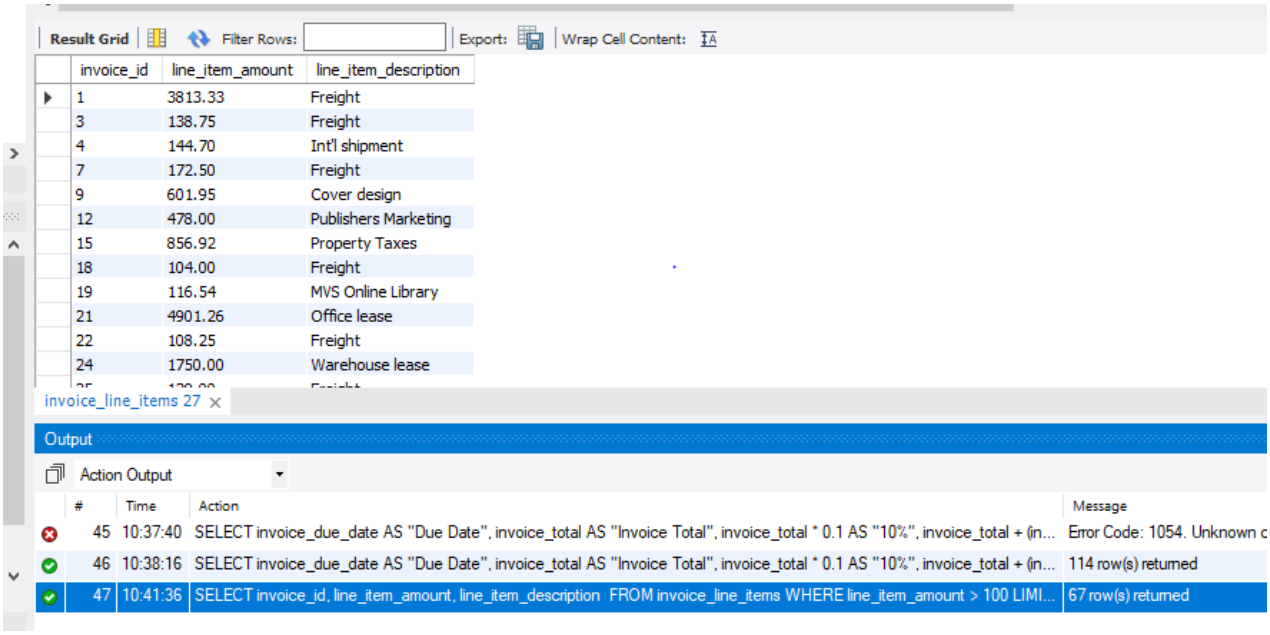
Output

Action Output

#	Time	Action	Message
44	10:34:10	SELECT CONCAT(vendor_contact_first_name, " ", vendor_contact_last_name) AS full_name FROM vendors WHERE vendo...	52 row(s) returned
45	10:37:40	SELECT invoice_due_date AS "Due Date", invoice_total AS "Invoice Total", invoice_total * 0.1 AS "10%", invoice_total + (in...	Error Code: 1054. Un
46	10:38:16	SELECT invoice_due_date AS "Due Date", invoice_total AS "Invoice Total", invoice_total * 0.1 AS "10%", invoice_total + (in...	114 row(s) returned

16. **(5) Write a SELECT statement that shows the invoice_id, line_item_amount, and line_item_description from the invoice_line_items table where the line_item_amount is greater than 100.**

```
SELECT invoice_id, line_item_amount, line_item_description

FROM invoice_line_items

WHERE line_item_amount > 100
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

invoice_id	line_item_amount	line_item_description
1	3813.33	Freight
3	138.75	Freight
4	144.70	Int'l shipment
7	172.50	Freight
9	601.95	Cover design
12	478.00	Publishers Marketing
15	856.92	Property Taxes
18	104.00	Freight
19	116.54	MVS Online Library
21	4901.26	Office lease
22	108.25	Freight
24	1750.00	Warehouse lease

invoice_line_items 27

Output

Action Output

#	Time	Action	Message
45	10:37:40	SELECT invoice_due_date AS "Due Date", invoice_total AS "Invoice Total", invoice_total * 0.1 AS "10%", invoice_total + (in...	Error Code: 1054. Unknown c
46	10:38:16	SELECT invoice_due_date AS "Due Date", invoice_total AS "Invoice Total", invoice_total * 0.1 AS "10%", invoice_total + (in...	114 row(s) returned
47	10:41:36	SELECT invoice_id, line_item_amount, line_item_description FROM invoice_line_items WHERE line_item_amount > 100 LIMI...	67 row(s) returned