

# CS411, Winter 2023

## Homework 2. Due by midnight Feb 7

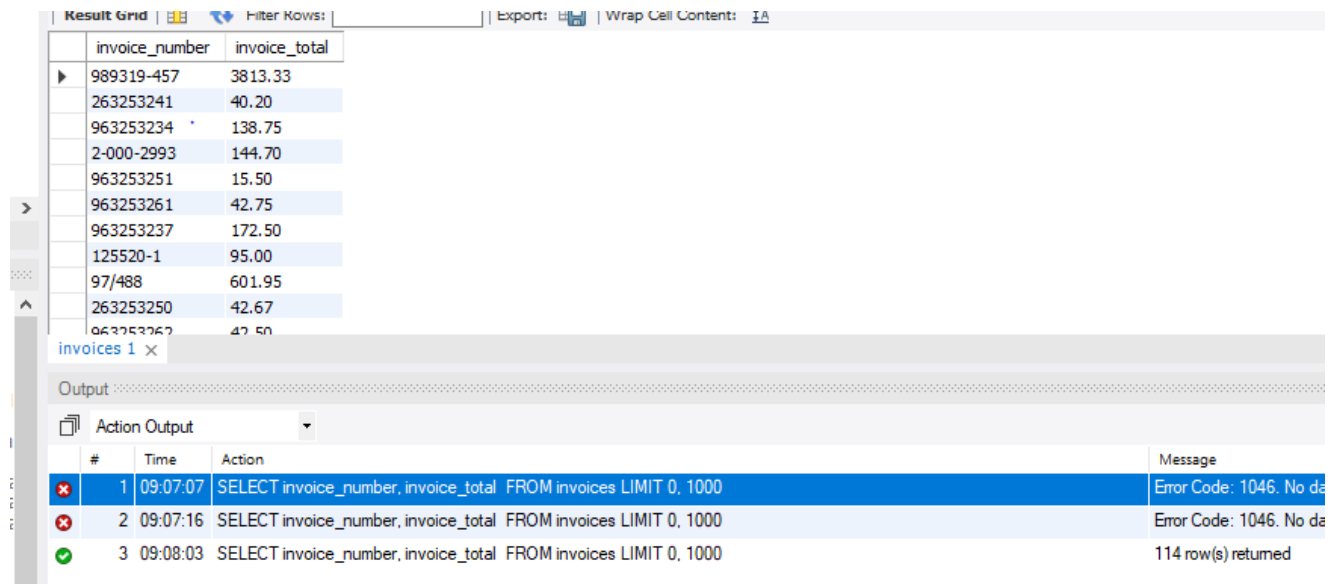
Follow the instructions and provide the result. All the problems are based on the ap database. You should show your query in text (so that I can copy/paste and run) and screenshots of the result. Make sure that you only copy the result part so that it can be easily read. You do not need to show all rows 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



The screenshot displays a database interface. At the top, a 'Result Grid' shows the first 10 rows of a query result. Below this, an 'Output' section titled 'Action Output' shows a log of three actions. The first two actions resulted in an error (Error Code: 1046. No database selected), while the third action was successful and returned 114 rows.

	invoice_number	invoice_total
▶	989319-457	3813.33
	263253241	40.20
	963253234	138.75
	2-000-2993	144.70
	963253251	15.50
>	963253261	42.75
	963253237	172.50
	125520-1	95.00
	97/488	601.95
^	263253250	42.67
	063253262	42.50

#	Time	Action	Message
✖	1 09:07:07	SELECT invoice_number, invoice_total FROM invoices LIMIT 0, 1000	Error Code: 1046. No database selected
✖	2 09:07:16	SELECT invoice_number, invoice_total FROM invoices LIMIT 0, 1000	Error Code: 1046. No database selected
✔	3 09:08:03	SELECT invoice_number, invoice_total FROM invoices LIMIT 0, 1000	114 row(s) returned

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				

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	47.50

invoices 6 x

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**

The screenshot shows a database application interface. At the top, there is a toolbar with icons for editing, exporting, and fetching rows. Below the toolbar is a 'Result Grid' showing a table with three columns: 'invoice\_id', 'vendor\_id', and 'invoice\_total'. The table contains five rows of data, with the last row having null values. Below the result grid is an 'Output' section with a tab labeled 'Action Output'. The output log shows a successful execution of the SQL query at 09:31:36, returning 5 rows.

invoice_id	vendor_id	invoice_total
102	110	20551.18
112	110	10976.06
98	83	579.42
105	106	503.20
113	37	224.00
NULL	NULL	NULL

invoices 8 x

Output

Action Output

#	Time	Action	Message
13	09:31:36	SELECT invoice_id, vendor_id, invoice_total FROM invoices WHERE payment_date IS NULL ORDER BY invoice_total LI...	5 row(s) returned

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**

Result Grid			
Filter Rows:   Edit:   Export/Import:   Wrap Cell Content:			
	invoice_id	invoice_total	payment_total
▶	78	1962.13	1762.13
	106	23517.58	21221.63
✱	NULL	NULL	NULL

invoices 12 x

Output			
Action Output			
#	Time	Action	Message
✓ 17	09:36:43	SELECT invoice_id, invoice_total , payment_total FROM invoices WHERE payment_total < invoice_total LIMIT 0, 1000	13 row(s) returned
✓ 18	09:37:03	SELECT invoice_id, invoice_total , payment_total FROM invoices WHERE payment_total > 0 AND payment_total < invoice_t...	2 row(s) returned

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
```

Result Grid			
	invoice_number	balance_due	invoice_total
	P-0608	19351.18	20551.18
	0-2436	10976.06	10976.06
	31361833	579.42	579.42
	9982771	503.20	503.20
	547480102	224.00	224.00

Result 15 x

Output			
Action Output			
#	Time	Action	Message
22	09:46:35	SELECT invoice_number, invoice_total - payment_total - credit_total AS balance_due, invoice_total FROM invoices WHERE...	Error Code: 1054. Unknown
23	09:48:16	SELECT invoice_number, invoice_total - payment_total - credit_total AS balance_due, invoice_total FROM invoices WHERE...	5 row(s) returned

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'.

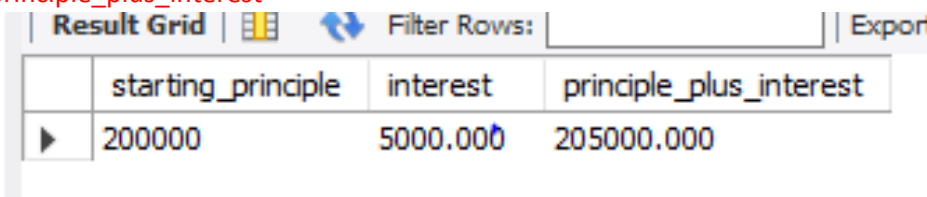
```
SELECT DATE_FORMAT(current_date, "%d-%M-%Y") AS 'Current Date'
```

Result Grid	
	Current Date
	07-February-2023

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_principal, 0.025 * 200000 AS interest, 200000 + (0.025 * 200000) AS  
principal_plus_interest
```



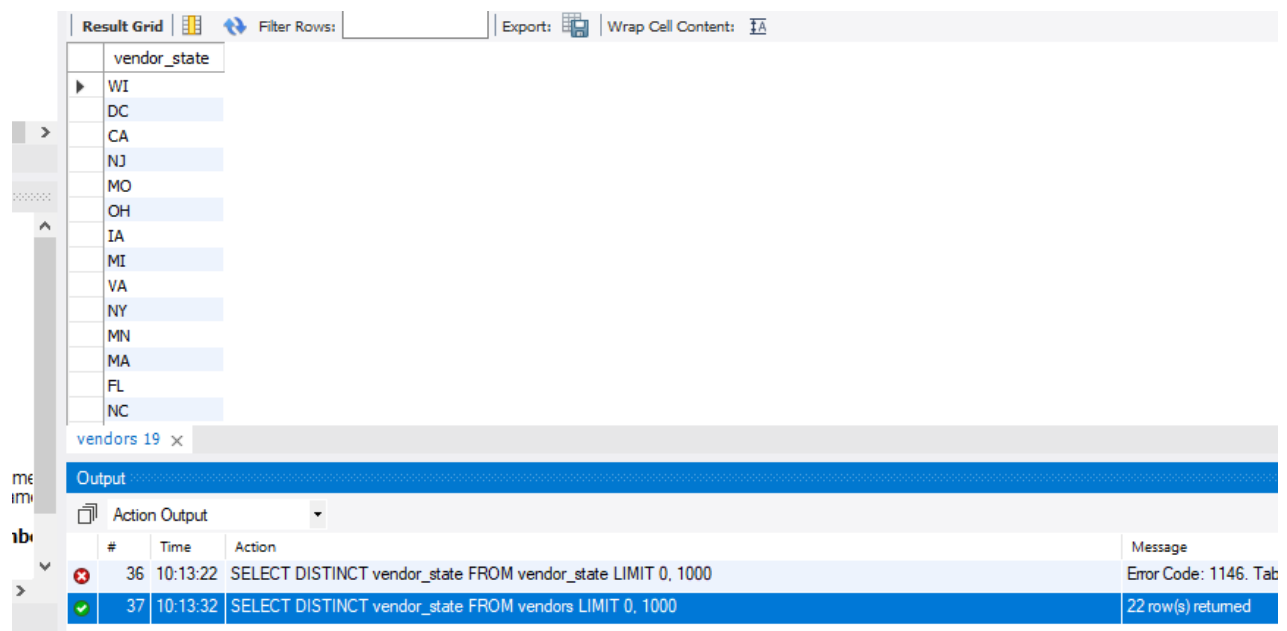
The screenshot shows a 'Result Grid' with three columns: 'starting\_principal', 'interest', and 'principal\_plus\_interest'. The first row contains the values 200000, 5000.000, and 205000.000 respectively. The interface includes a 'Filter Rows' field and an 'Export' button.

	starting_principal	interest	principal_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
```



The screenshot displays a SQL IDE interface. At the top, a 'Result Grid' shows the output of a query, listing distinct vendor states: WI, DC, CA, NJ, MO, OH, IA, MI, VA, NY, MN, MA, FL, and NC. Below this, the 'Output' pane shows the execution log. It contains two entries: a failed query (Action 36) and a successful query (Action 37). The successful query is 'SELECT DISTINCT vendor\_state FROM vendors LIMIT 0, 1000', which returned 22 rows.

	vendor_state
▶	WI
	DC
	CA
	NJ
	MO
	OH
	IA
	MI
	VA
	NY
	MN
	MA
	FL
	NC

#	Time	Action	Message
36	10:13:22	SELECT DISTINCT vendor_state FROM vendor_state LIMIT 0, 1000	Error Code: 1146. Tab
37	10:13:32	SELECT DISTINCT vendor_state FROM vendors LIMIT 0, 1000	22 row(s) returned



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	Amerian 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 x			
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'

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

	vendor_name	vendor_city	vendor_state	vendor_zip_code
▶	Ingram	Dallas	TX	75284

vendors 22 x

Output

Action Output

#	Time	Action	Message
✓ 39	10:17:13	SELECT vendor_contact_last_name, vendor_contact_first_name, vendor_name FROM vendors ORDER BY vendor_contact...	122 row(s) returned
✓ 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

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
```

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

full_name
Thom Aaronsen
Joan Aileen
Francesco Alberto
Alexandro Alexis
Zev Alondra
Nashalie Angelica
Troy Antavius
Kaitlyn Anthoni
Trisha Anum
Robert Aranovitch

Result 25 x

Output

Action Output

#	Time	Action	Message
✓ 43	10:33:11	SELECT CONCAT(vendor_contact_first_name, vendor_contact_last_name) AS full_name FROM vendors WHERE vendor_c...	52 row(s) returned
✓ 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

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 x				
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
	25	120.00	Freight

invoice\_line\_items 27 x

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 LIM...	67 row(s) returned