

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

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT *
2 FROM vendors JOIN invoices
3 ON vendors.vendor_id = invoices.vendor_id
```

The 'Result Grid' shows the first few rows of the query results:

vendor_id	vendor_name	vendor_address1	vendor_address2	vendor_city	vendor_state	vendor_zip_code	vendor_phone	vendor_o
122	United Parcel Service	P.O. Box 505820		Reno	NV	88905	(800) 555-0855	Beauregar
123	Federal Express Corporation	P.O. Box 1140	Dept A	Memphis	TN	38101	(800) 555-4091	Bucket
123	Federal Express Corporation	P.O. Box 1140	Dept A	Memphis	TN	38101	(800) 555-4091	Bucket

The 'Output' tab shows the execution log, with the final message indicating that 114 rows were returned:

```
76 19:41:41 SELECT * FROM vendors JOIN invoices ON vendors.vendor_id = invoices.vendor_id LIMIT 0, 1000 114 row(s) returned
```

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
```

The screenshot displays a database management interface with a Navigator pane on the left, a SQL editor in the center, and a Results pane at the bottom.

Navigator: Shows a tree view of the database schema. The 'ap' schema is expanded, showing tables like 'general_ledger_accounts', 'invoice_archive', 'invoice_line_items', 'invoices', 'terms', 'vendor_contacts', and 'vendors'. The 'ex' schema is also expanded, showing 'active_invoices' and its columns: 'invoice_id', 'vendor_id', 'invoice_number', 'invoice_date', 'invoice_total', and 'payment_total'.

SQL Editor: Contains the following query:

```
1 SELECT vendor_name, invoice_number, invoice_date,  
2       invoice_total - payment_total - credit_total AS balance_due  
3 FROM vendors v JOIN invoices i  
4   ON v.vendor_id = i.vendor_id  
5 WHERE invoice_total - payment_total - credit_total <> 0  
6 ORDER BY vendor_name
```

Results: The query results are displayed in a table with the following columns: 'vendor_name', 'invoice_number', 'invoice_date', and 'balance_due'.

vendor_name	invoice_number	invoice_date	balance_due
Blue Cross	547480102	2014-08-01	224.00
Cardinal Business Media, Inc.	134116	2014-07-28	90.36
Data Reproductions Corp	39104	2014-07-10	85.31
Federal Express Corporation	963253264	2014-07-18	52.25
Federal Express Corporation	263253268	2014-07-21	59.97
Federal Express Corporation	263253270	2014-07-22	67.92
Federal Express Corporation	263253273	2014-07-22	30.75
Ford Motor Credit Company	9982771	2014-07-24	503.20
Ingram	31361833	2014-07-21	579.42
Malloy Lithographing Inc	P-0608	2014-07-23	19351.18

Action Output: The bottom pane shows a log of database actions. The last action is a SELECT query that returned 11 rows.

#	Time	Action	Message
72	19:15:40	use ap	0 row(s) affected
73	19:15:40	SELECT 'Active' AS source, invoice_number, invoice_date, invoice_total FROM invoices WHERE in...	114 row(s) returned
74	19:32:22	use ap	0 row(s) affected
75	19:32:22	SELECT invoice_number, vendor_name, '33% Payment' AS payment_type, invoice_total AS total, inv...	114 row(s) returned
76	19:41:41	SELECT * FROM vendors JOIN invoices ON vendors.vendor_id = invoices.vendor_id LIMIT 0, 1000	114 row(s) returned
77	19:49:54	SELECT vendor_name, invoice_number, invoice_date, invoice_total - payment_total - credit_total AS ...	11 row(s) returned

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
```

The screenshot shows the Navicat for MySQL interface. On the left, the 'SCHEMAS' panel displays the database structure, including tables like 'general_ledger_accounts', 'invoice_archive', 'invoice_line_items', 'invoices', 'terms', 'vendor_contacts', and 'vendors'. The 'active_invoices' table is expanded, showing columns: 'invoice_id', 'vendor_id', 'invoice_number', 'invoice_date', 'invoice_total', and 'payment_total'.

The main query window shows the following SQL query:

```

1 SELECT vendor_name, default_account_number AS default_account,
2    account_description AS description
3 FROM vendors v JOIN general_ledger_accounts gl
4    ON v.default_account_number = gl.account_number
5 ORDER BY account_description, vendor_name

```

The 'Result Grid' displays the following data:

vendor_name	default_account	description
Dristas Groom & McCormick	591	Accounting
DMV Renewal	568	Auto License Fee
Newbrige Book Clubs	394	Book Club Royalties
Bertelsmann Industry Svcs. Inc	400	Book Printing Costs
Courier Companies, Inc	400	Book Printing Costs
Crown Printing	400	Book Printing Costs
Data Reproductions Corp	400	Book Printing Costs
Diversified Printing & Pub	400	Book Printing Costs
Malloy Lithographing Inc	400	Book Printing Costs
Fresno Photoengraving Company	403	Book Production Costs

The 'Output' panel shows the execution log with the following actions:

#	Time	Action	Message
73	19:15:40	SELECT 'Active' AS source, invoice_number, invoice_date, invoice_total FROM invoices WHERE in...	114 row(s) returned
74	19:32:22	use ap	0 row(s) affected
75	19:32:22	SELECT invoice_number, vendor_name, '33% Payment' AS payment_type, invoice_total AS total, inv...	114 row(s) returned
76	19:41:41	SELECT * FROM vendors JOIN invoices ON vendors.vendor_id = invoices.vendor_id LIMIT 0, 1000	114 row(s) returned
77	19:49:54	SELECT vendor_name, invoice_number, invoice_date, invoice_total - payment_total - credit_total AS ...	11 row(s) returned
78	19:56:32	SELECT vendor_name, default_account_number AS default_account, account_description AS desc...	122 row(s) returned

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

The screenshot shows the Navicat for MySQL interface. The left sidebar displays the database schema for 'navneet' and 'ex'. The main window shows a SQL query in 'Query 1' that joins 'vendors', 'invoices', and 'invoice_line_items' tables. The 'Result Grid' displays the query results with columns: vendor_name, invoice_date, invoice_number, li_sequence, and li_amount. The 'Output' pane shows the execution log with messages for each step of the query execution.

Query 1

```
1 use ap;
2 SELECT vendor_name, invoice_date, invoice_number,
3        invoice_sequence AS li_sequence,
4        line_item_amount AS li_amount
5 FROM vendors v JOIN invoices i
6      ON v.vendor_id = i.vendor_id
7      JOIN invoice_line_items li
8      ON i.invoice_id = li.invoice_id
9 ORDER BY vendor_name, invoice_date, invoice_number, invoice_sequence
10
```

Result Grid

vendor_name	invoice_date	invoice_number	li_sequence	li_amount
Abbey Office Furnishings	2014-07-05	203339-13	1	17.50
Bertelsmann Industry Svcs. Inc	2014-06-18	509786	1	6940.25
Blue Cross	2014-06-03	547481328	1	224.00
Blue Cross	2014-06-07	547479217	1	116.00
Blue Cross	2014-08-01	547480102	1	224.00
Cahners Publishing Company	2014-06-30	587056	1	2184.50
Cardinal Business Media, Inc.	2014-06-22	133560	1	175.00
Cardinal Business Media, Inc.	2014-07-28	134116	1	90.36
Coffee Break Service	2014-06-24	109596	1	41.80

Output

#	Time	Action	Message
80	20:16:16	select * from active_invoices LIMIT 0, 1000	Error Code: 1146. Table 'ap.active_invoices' doesn't exist
81	20:16:28	use ex	0 row(s) affected
82	20:16:28	select * from active_invoices LIMIT 0, 1000	40 row(s) returned
83	20:46:26	SELECT vendor_name, invoice_date, invoice_number, invoice_sequence AS li_sequence, line_...	Error Code: 1146. Table 'ex.vendors' doesn't exist
84	20:46:41	use ap	0 row(s) affected
85	20:46:41	SELECT vendor_name, invoice_date, invoice_number, invoice_sequence AS li_sequence, line_...	118 row(s) returned

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
```

The screenshot shows the Navicat for MySQL interface. The left sidebar displays the database schema, including tables like 'general_ledger_accounts', 'invoice_archive', 'invoice_line_items', 'invoices', 'terms', 'vendor_contacts', and 'vendors'. The main window shows a SQL query (Query 1) that performs a self-join on the 'vendors' table to find vendors with the same last name. The query is as follows:

```
1 use ap;
2 SELECT v1.vendor_id,
3        v1.vendor_name,
4        CONCAT(v1.vendor_contact_first_name, ' ', v1.vendor_contact_last_name) AS contact_name
5 FROM vendors v1 JOIN vendors v2
6 ON v1.vendor_id <> v2.vendor_id AND
7    v1.vendor_contact_last_name = v2.vendor_contact_last_name
8 ORDER BY v1.vendor_contact_last_name
```

The 'Result Grid' shows the output of the query, displaying two rows of vendor information:

vendor_id	vendor_name	contact_name
115	Roadway Package System, Inc	Sam Smith
51	Blue Shield of California	Kyle Smith

Below the result grid, the 'Action Output' pane shows the execution log, including the time taken for each step and the number of rows affected or returned.

#	Time	Action	Message
84	20:46:41	use ap	0 row(s) affected
85	20:46:41	SELECT vendor_name, invoice_date, invoice_number, invoice_sequence AS li_sequence, line...	118 row(s) returned
86	20:54:21	use ap	0 row(s) affected
87	20:54:21	SELECT v1.vendor_id, v1.vendor_name, CONCAT(v1.vendor_contact_first_name, ' ', v1.vendor...	124 row(s) returned
88	20:56:22	use ap	0 row(s) affected
89	20:56:23	SELECT v1.vendor_id, v1.vendor_name, CONCAT(v1.vendor_contact_first_name, ' ', v1.vendor...	2 row(s) returned

The screenshot shows a MySQL IDE interface. On the left is the 'SCHEMAS' tree with 'ap' and 'ex' databases. The main window displays a SQL query:

```

1 use ap;
2 SELECT v1.vendor_id,
3        v1.vendor_name,
4        v2.vendor_id,
5        CONCAT(v1.vendor_contact_first_name, ' ', v1.vendor_contact_last_name) AS contact_name
6 FROM vendors v1 JOIN vendors v2
7 ON
8    v1.vendor_id <> v2.vendor_id and
9    v1.vendor_contact_last_name = v2.vendor_contact_last_name /* where v1.vendor_contact_last_name = "Smith" */
10 ORDER BY v1.vendor_id;
11
12
13

```

Below the query is the 'Result Grid' showing two rows of data:

vendor_id	vendor_name	vendor_id	contact_name
51	Blue Shield of California	115	Kylie Smith
115	Roadway Package System, Inc	51	Sam Smith

At the bottom, the 'Action Output' pane shows the execution of the query:

#	Time	Action	Message
106	21:25:34	use ap	0 row(s) affected
107	21:25:34	SELECT v1.vendor_id, v1.vendor_name, CONCAT(v1.vendor_contact_first_name, ' ', v1.vendor...	4 row(s) returned
108	21:26:15	use ap	0 row(s) affected
109	21:26:15	SELECT v1.vendor_id, v1.vendor_name, v2.vendor_id, CONCAT(v1.vendor_contact_first_...	4 row(s) returned
110	21:28:09	use ap	0 row(s) affected
111	21:28:09	SELECT v1.vendor_id, v1.vendor_name, v2.vendor_id, CONCAT(v1.vendor_contact_first_...	2 row(s) returned

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

```


The screenshot shows the MySQL Workbench interface with a query window open. The query is as follows:

```

1 • use ap;
2 • SELECT gl.account_number, account_description, invoice_id
3 FROM general_ledger_accounts gl LEFT JOIN invoice_line_items li
4 ON gl.account_number = li.account_number
5 WHERE li.invoice_id IS NULL
6 ORDER BY gl.account_number
7
8
9
10

```

The result grid shows the following data:

account_number	account_description	invoice_id
100	Cash	NULL
110	Accounts Receivable	NULL
120	Book Inventory	NULL
162	Capitalized Lease	NULL
167	Software	NULL

The bottom panel shows the Action Output log:

#	Time	Action	Message
108	21:26:15	use ap	0 row(s) affected
109	21:26:15	SELECT v1.vendor_id, v1.vendor_name, v2.vendor_id, CONCAT(v1.vendor_contact_first_...	4 row(s) returned
110	21:28:09	use ap	0 row(s) affected
111	21:28:09	SELECT v1.vendor_id, v1.vendor_name, v2.vendor_id, CONCAT(v1.vendor_contact_first_...	2 row(s) returned
112	21:33:27	use ap	0 row(s) affected
113	21:33:27	SELECT gl.account_number, account_description, invoice_id FROM general_ledger_accounts gl LEFT J...	54 row(s) returned

- 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

navneet x navneet (ap) x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

ap

Tables

general_ledger_accounts

invoice_archive

invoice_line_items

invoices

terms

vendor_contacts

vendors

Views

Stored Procedures

Functions

ex

Tables

active_invoices

Columns

invoice_id

vendor_id

invoice_number

invoice_date

invoice_total

payment_total

Administration Schemas

Information

Connection Details

Name: navneet

Host: 127.0.0.1

Port: 55515

Login User: root

Current User: root@localhost

SSL cipher: DHE-RSA-AES128-GCM-

Server

Product: MySQL Community Serv

Version: 8.0.16

Connector

Version: C++ 8.0.20

Query 1 SQL File 1* SQL File 4* SQL File 5*

Limit to 1000 rows

1 use ap;

2 SELECT vendor_name, vendor_state

3 FROM vendors

4 WHERE vendor_state = 'CA'

5 UNION

6 SELECT vendor_name, 'Outside CA'

7 FROM vendors

8 WHERE vendor_state <> 'CA'

9 ORDER BY vendor_name

10

11

12

13

Result Grid

Filter Rows:

Export: Wrap Cell Content: 15

vendor_name vendor_state

Abbey Office Furnishings CA

American Booksellers Assoc Outside CA

American Express CA

ASC Signs CA

Ascom Hasler Mailing Systems Outside CA

Result 18 x

Output

Action Output

Time Action Message

114 23:18:00 use ap 0 row(s) affected

115 23:18:00 SELECT vendor_name, vendor_state FROM vendors WHERE vendor_state = 'CA' UNION SELECT v... 122 row(s) returned

116 23:18:21 use ap 0 row(s) affected

117 23:18:21 SELECT vendor_name, vendor_state FROM vendors WHERE vendor_state = 'CA' UNION SELECT v... 122 row(s) returned

118 23:20:08 use ap 0 row(s) affected

119 23:20:08 SELECT vendor_name, vendor_state FROM vendors WHERE vendor_state = 'CA' UNION SELECT v... 122 row(s) returned