

Table of Contents

Snapshots of SQL queries.....	0
1. Snapshot of query to show all users.....	1
2. Snapshot of query to show all customers.....	1
3. Snapshot of query to show all customers with the title "Mr."	2
4. Snapshot of query to show customers from 2 eastern provinces or states only (for example QC and ON)	2
5. Snapshot of query to show customers whose monthly income is in a range from \$2500.00 to \$5000.00 and order them in ascending order	3
6. Snapshot of query to count number of customers whose monthly income is less than \$3000.00.....	3
7. Snapshot of query to count total amount of monthly income for all customers	4
8. Snapshot of query to count net income (monthly income minus monthly debt) for all customers.....	4
9. Snapshot of query to count the minimum, maximum and average net income (monthly income minus monthly debt) of all customers respectively	5
10. Snapshot of query to count the total number of users and group them by the company	6
11. Snapshot of query to find the total number of customers with net income (monthly income minus monthly debt) more than \$3500.00 and group them by city	7
12. Snapshot of query to show all log records (message, timestamp, username and user_id) for the user 'admin' (that query should select from 2 tables).....	7
13. Snapshot of query to create a query using INNER JOIN for 'events' and 'users' tables. That query should display all messages, corresponding usernames, user_ids and timestamps sorted by username. Run this query against LoanApp, take a snapshot (that includes SELECT statement and results of the query) and add it to the journal.	8

Snapshots of SQL queries

1. Snapshot of query to show all users

The screenshot shows the MySQL Workbench interface with a query executed to show all users. The query is as follows:

```
1 SELECT *
2 FROM users
```

The result grid displays the following data:

USER_ID	TYPE	BRANCH_ID	BUILD	USERNAME	PASSWORD	FIRST_NAME	LAST_NAME	COMPANY	PHONE	EMAIL
23000	0	508	0	admin	ntema1004	ntema	ntema	ntema	77896578533	Apple1@abc.com
23652	1	508	8	Asample	password	Apple	Bell	ABC	77896578533	Apple1@abc.com
23709	1	508	8	Bell	password	Bell	Counter	ABC	8965560000	BellCounter@abc.com
23708	1	508	8	SethRogen	password	Seth	Rogen	ABC	778787778	sethrogen@abc.com
23710	1	508	8	Yvkephone	password	Yvke	phone	XYZ	8964562556	Yvke@xyz.com
23711	1	508	8	Young	password	Young	Zombie	XYZ	9864203215	youngzombie@xyz.com
23712	1	508	8	Zombie	password	Zom	Ble	XYZ	89546225633	zombie@xyz.com

The output pane shows the following message:

```
1 12:29:43 SELECT * FROM users LIMIT 0, 1000
7 row(s) returned
```

The duration of the query is 0.093 sec / 0.000 sec.

2. Snapshot of query to show all customers

The screenshot shows the MySQL Workbench interface with a query executed to show all customers. The query is as follows:

```
1 SELECT *
2 FROM customers
```

The result grid displays the following data:

CUSTOMER_ID	VERSION	TITLE	FIRST_NAME	LAST_NAME	STREET	CITY	PROVINCE	POSTAL_CODE	MONTHLY_INCOME	MONTHLY_DEBT
75	0	Ms.	Julia	Roberts	Thornhill	ON	M9B2H4	2500	1500	
76	0	Mr.	Seth	Rogen	Thornhill	ON	M9B2H4	2600	1200	
77	0	Mr.	Jerry	Biden	Thornhill	ON	M9B2H4	3200	2500	
78	0	Ms.	Kaley	Hunter	Thornhill	ON	M9B2H4	3800	2600	
79	0	Mrs.	Kelly	Davidson	Thornhill	ON	M9B2H4	4000	2600	
80	0	Ms.	Angela	Kinley	Montreal	QC	H3T1M6	2500	2600	
81	0	Dr.	Oscar	Martinez	Montreal	QC	H3T1M6	5000	2500	
82	0	Mr.	David	Attenborough	Boston	MA	02118	3000	1200	
83	0	Mr.	Michael	Jackson	Boston	MA	02118	4500	3600	
84	0	Mr.	Dwight	Shrute	Boston	MA	02118	6000	2000	

The output pane shows the following messages:

```
1 12:29:43 SELECT * FROM users LIMIT 0, 1000
7 row(s) returned
2 12:30:35 SELECT * FROM customers LIMIT 0, 1000
10 row(s) returned
3 12:36:10 SELECT * FROM customers LIMIT 0, 1000
10 row(s) returned
```

The duration of the query is 0.078 sec / 0.000 sec.

3. Snapshot of query to show all customers with the title "Mr."

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT *
2 FROM customers
3 WHERE title = "Mr."
```

The result grid displays the following data:

CUSTOMER_ID	VERSION	TITLE	FIRST_NAME	LAST_NAME	STREET	CITY	PROVINCE	POSTAL_CODE	MONTHLY_INCOME	MONTHLY_DEBT
76	0	Mr.	Seth	Rogen	Thornhill	ON		2600	1200	
77	0	Mr.	Joey	Biden	Thornhill	ON		3200	2500	
82	0	Mr.	David	Attenborough	Boston	MA		3000	1200	
83	0	Mr.	Michael	Jackson	Boston	MA		4500	1600	
84	0	Mr.	Dought	Shrute	Boston	MA		6000	2000	

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	12:29:43	SELECT * FROM users LIMIT 0, 1000	7 row(s) returned	0.093 sec / 0.000 sec
2	12:30:35	SELECT * FROM customers LIMIT 0, 1000	10 row(s) returned	0.078 sec / 0.000 sec
3	12:36:10	SELECT * FROM customers WHERE title = "Mr." LIMIT 0, 1000	5 row(s) returned	0.078 sec / 0.000 sec
4	12:37:48	SELECT * FROM customers WHERE title = "Mr." LIMIT 0, 1000	5 row(s) returned	0.078 sec / 0.000 sec

4. Snapshot of query to show customers from 2 eastern provinces or states only (for example QC and ON)

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT *
2 FROM customers
3 WHERE province = "QC" OR province = "ON"
```

The result grid displays the following data:

CUSTOMER_ID	VERSION	TITLE	FIRST_NAME	LAST_NAME	STREET	CITY	PROVINCE	POSTAL_CODE	MONTHLY_INCOME	MONTHLY_DEBT
75	0	Ms.	Julia	Roberts	Thornhill	ON		59624	2500	1500
76	0	Mr.	Seth	Rogen	Thornhill	ON		2600	1200	
77	0	Mr.	Joey	Biden	Thornhill	ON		3200	2500	
78	0	Ms.	Katey	Hunter	Thornhill	ON		3800	2600	
79	0	Ms.	Kelly	Davidson	Montreal	QC		4000	2600	
80	0	Ms.	Angela	Kinsley	Montreal	QC		25000	2600	
81	0	Dr.	Oscar	Martinez	Montreal	QC		5000	2500	

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	12:29:43	SELECT * FROM users LIMIT 0, 1000	7 row(s) returned	0.093 sec / 0.000 sec
2	12:30:35	SELECT * FROM customers LIMIT 0, 1000	10 row(s) returned	0.078 sec / 0.000 sec
3	12:36:10	SELECT * FROM customers WHERE title = "Mr." LIMIT 0, 1000	5 row(s) returned	0.078 sec / 0.000 sec
4	12:37:48	SELECT * FROM customers WHERE title = "Mr." LIMIT 0, 1000	5 row(s) returned	0.078 sec / 0.000 sec
5	12:39:13	SELECT * FROM customers WHERE province = "QC" OR province = "ON" LIMIT 0, 1000	7 row(s) returned	0.078 sec / 0.000 sec

5. Snapshot of query to show customers whose monthly income is in a range from \$2500.00 to \$5000.00 and order them in ascending order

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT *
2 FROM customers
3 WHERE monthly_income BETWEEN 2500 AND 5000
```

The result grid displays the following data:

CUSTOMER_ID	VERSION	TITLE	FIRST_NAME	LAST_NAME	STREET	CITY	PROVINCE	POSTAL_CODE	MONTHLY_INCOME	MONTHLY_DEBT
75	0	Ms.	Julia	Roberts	Thornhill	OH	59624	2500	1300	
76	0	Mr.	Seeth	Roggen	Thornhill	OH	3600	1200		
77	0	Mr.	Joey	Biden	Thornhill	OH	3300	2500		
78	0	Ms.	Kaley	Hunter	Thornhill	OH	3800	2600		
79	0	Mrs.	Kathy	Davidson	Montreal	QC	4000	2600		
81	0	Dr.	Oscar	Hartinez	Montreal	QC	5000	2300		
82	0	Mr.	David	Attenborough	Boston	MA	3000	1200		
83	0	Mr.	Michael	Jackson	Boston	MA	4500	1600		

The output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
1	12:29:43	SELECT * FROM users LIMIT 0, 1000	7 row(s) returned	0.093 sec / 0.000 sec
2	12:30:35	SELECT * FROM customers LIMIT 0, 1000	10 row(s) returned	0.078 sec / 0.000 sec
3	12:36:10	SELECT * FROM customers WHERE title = "Ms." LIMIT 0, 1000	10 row(s) returned	0.078 sec / 0.000 sec
4	12:37:45	SELECT * FROM customers WHERE title = "Mr." LIMIT 0, 1000	5 row(s) returned	0.078 sec / 0.000 sec
5	12:39:13	SELECT * FROM customers WHERE province = "QC" OR province = "ON" LIMIT 0, 1000	7 row(s) returned	0.078 sec / 0.000 sec
6	12:40:52	SELECT * FROM customers WHERE monthly_income BETWEEN 2500 AND 5000 LIMIT 0, 1000	8 row(s) returned	0.078 sec / 0.000 sec

6. Snapshot of query to count number of customers whose monthly income is less than \$3000.00

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT COUNT(CUSTOMER_ID)
2 FROM customers
3 WHERE monthly_income < 3000
```

The result grid displays the following data:

COUNT(CUSTOMER_ID)
2

The output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
3	12:36:10	SELECT * FROM customers LIMIT 0, 1000	10 row(s) returned	0.078 sec / 0.000 sec
4	12:37:48	SELECT * FROM customers WHERE title = "Ms." LIMIT 0, 1000	5 row(s) returned	0.078 sec / 0.000 sec
5	12:39:13	SELECT * FROM customers WHERE province = "QC" OR province = "ON" LIMIT 0, 1000	7 row(s) returned	0.078 sec / 0.000 sec
6	12:40:52	SELECT * FROM customers WHERE monthly_income BETWEEN 2500 AND 5000 LIMIT 0, 1000	8 row(s) returned	0.078 sec / 0.000 sec
7	12:43:37	SELECT * FROM customers WHERE monthly_income < 3000 LIMIT 0, 1000	2 row(s) returned	0.093 sec / 0.000 sec
8	12:45:55	SELECT COUNT(CUSTOMER_ID) FROM customers WHERE monthly_income < 3000 LIMIT 0, 1000	1 row(s) returned	0.094 sec / 0.000 sec

7. Snapshot of query to count total amount of monthly income for all customers

MySQL Workbench interface showing a query to count total monthly income for all customers.

Query 1:

```
1 SELECT SUM(monthly_income) AS sum
2 FROM customers
3
```

Result Grid:

sum
159000

Execution Log (Result 12):

#	Time	Action	Message	Duration / Fetch
5	12:39:13	SELECT * FROM customers WHERE province = "QC" OR province = "ON" LIMIT 0, 1000	7 row(s) returned	0.078 sec / 0.000 sec
6	12:40:52	SELECT * FROM customers WHERE monthly_income BETWEEN 2500 AND 5000 LIMIT 0, 1000	8 row(s) returned	0.079 sec / 0.000 sec
7	12:43:37	SELECT * FROM customers WHERE monthly_income < 3000 LIMIT 0, 1000	2 row(s) returned	0.093 sec / 0.000 sec
8	12:45:55	SELECT COUNT(CUSTOMER_ID) FROM customers WHERE monthly_income < 3000 LIMIT 0, 1000	1 row(s) returned	0.094 sec / 0.000 sec
9	12:49:31	SELECT COUNT(*) AS count, monthly_income FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec
10	12:50:35	SELECT SUM(monthly_income) AS sum FROM customers LIMIT 0, 1000	1 row(s) returned	0.094 sec / 0.000 sec

8. Snapshot of query to count net income (monthly income minus monthly debt) for all customers

MySQL Workbench interface showing a query to count net income (monthly income minus monthly debt) for all customers.

Query 1:

```
1 SELECT SUM(monthly_income - monthly_debt) AS sum
2 FROM customers
3
```

Result Grid:

sum
39300

Execution Log (Result 13):

#	Time	Action	Message	Duration / Fetch
6	12:40:52	SELECT * FROM customers WHERE monthly_income BETWEEN 2500 AND 5000 LIMIT 0, 1000	8 row(s) returned	0.079 sec / 0.000 sec
7	12:43:37	SELECT * FROM customers WHERE monthly_income < 3000 LIMIT 0, 1000	2 row(s) returned	0.093 sec / 0.000 sec
8	12:45:55	SELECT COUNT(CUSTOMER_ID) FROM customers WHERE monthly_income < 3000 LIMIT 0, 1000	1 row(s) returned	0.094 sec / 0.000 sec
9	12:49:31	SELECT COUNT(*) AS count, monthly_income FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec
10	12:50:35	SELECT SUM(monthly_income) AS sum FROM customers LIMIT 0, 1000	1 row(s) returned	0.094 sec / 0.000 sec
11	12:52:26	SELECT SUM(monthly_income - monthly_debt) AS sum FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec

9. Snapshot of query to count the minimum, maximum and average net income (monthly income minus monthly debt) of all customers respectively

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT MIN(monthly_income - monthly_debt) AS min
2 FROM customers
3
```

The result grid displays the minimum net income as 700.

The output pane shows the execution details for the query:

#	Time	Action	Message	Duration / Fetch
7	12:43:37	SELECT - FROM customers WHERE monthly_income < 3000 LIMIT 0, 1000	2 row(s) returned	0.093 sec / 0.000 sec
8	12:45:55	SELECT COUNT(CUSTOMER_ID) FROM customers WHERE monthly_income < 3000 LIMIT 0, 1000	1 row(s) returned	0.094 sec / 0.000 sec
9	12:49:31	SELECT COUNT(*) AS count, monthly_income FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec
10	12:50:35	SELECT SUM(monthly_income) AS sum FROM customers LIMIT 0, 1000	1 row(s) returned	0.094 sec / 0.000 sec
11	12:52:26	SELECT SUM(monthly_income - monthly_debt) AS sum FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec
12	12:53:56	SELECT MIN(monthly_income - monthly_debt) AS min FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT MAX(monthly_income - monthly_debt) AS max
2 FROM customers
3
```

The result grid displays the maximum net income as 22400.

The output pane shows the execution details for the query:

#	Time	Action	Message	Duration / Fetch
1	12:55:16	SELECT MAX(monthly_income - monthly_debt) AS max FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec

Continued..

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT AVG(monthly_income - monthly_debt) AS avg
2 FROM customers
3
```

The result grid shows a single row with the value 3930 under the column 'avg'.

The output pane shows the execution details:

#	Time	Action	Message	Duration / Fech
1	12:55:16	SELECT MAX(monthly_income - monthly_debt) AS max FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec
2	12:56:01	SELECT AVG(monthly_income - monthly_debt) AS avg FROM customers LIMIT 0, 1000	1 row(s) returned	0.093 sec / 0.000 sec

10. Snapshot of query to count the total number of users and group them by the company

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT COUNT(*) AS count, company
2 FROM users
3 GROUP BY company
```

The result grid shows three rows of data:

	count	COMPANY
1	1008	
2	3	ABC
3	3	XYZ

The output pane shows the execution details:

#	Time	Action	Message	Duration / Fech
1	12:55:16	SELECT MAX(monthly_income - monthly_debt) AS max FROM customers LIMIT 0, 1000	1 row(s) returned	0.078 sec / 0.000 sec
2	12:56:01	SELECT AVG(monthly_income - monthly_debt) AS avg FROM customers LIMIT 0, 1000	1 row(s) returned	0.093 sec / 0.000 sec
3	12:57:53	SELECT COUNT(*) AS count, company FROM users GROUP BY company LIMIT 0, 1000	3 row(s) returned	0.094 sec / 0.000 sec

11. Snapshot of query to find the total number of customers with net income (monthly income minus monthly debt) more than \$3500.00 and group them by city

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT COUNT(*) AS count, city
2 FROM customers
3 GROUP BY city
4 HAVING SUM(monthly_income-monthly_debt)>3500
```

The result grid shows the following data:

count	city
3	Boston
3	Montreal
4	Thornton

The output pane shows the execution details: 1 13:00:20 SELECT COUNT(*) AS count, city FROM customers GROUP BY city HAVING SUM(monthly_income-monthly_debt)>3500 LIMIT 0, 1000. Message: 3 row(s) returned. Duration / Fetch: 0.093 sec / 0.000 sec.

12. Snapshot of query to show all log records (message, timestamp, username and user_id) for the user 'admin' (that query should select from 2 tables)

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

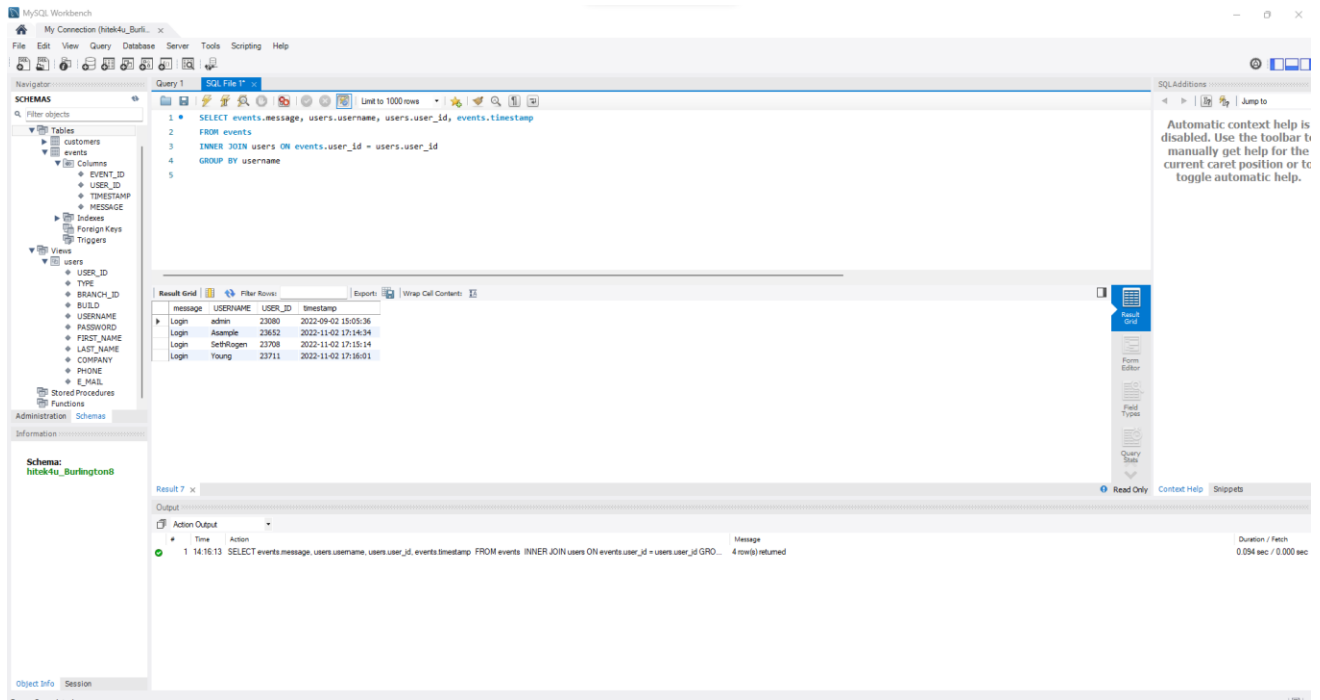
```
1 SELECT events.message, events.timestamp, users.username, events.user_id
2 FROM events, users
3 WHERE events.user_id = users.user_id
4 AND users.username = 'admin'
```

The result grid shows the following data:

message	timestamp	username	user_id
Login	2022-09-02 15:05:36	admin	23080
Request: Find Customers ()	2022-09-02 15:06:37	admin	23080
Response: []	2022-09-02 15:06:37	admin	23080
Request: FindOption(PrimeRate)	2022-09-02 15:08:04	admin	23080
Response: Option (PrimeRate,3.50)	2022-09-02 15:08:04	admin	23080
Request: FindOption(PrimeRate)	2022-09-02 15:08:10	admin	23080
Response: Option (PrimeRate,3.50)	2022-09-02 15:08:10	admin	23080
Request: Add Customer (Id=null,FirstName=H...	2022-09-02 15:08:34	admin	23080
Response: OK	2022-09-02 15:08:34	admin	23080
Request: Find Customers ()	2022-09-02 15:08:37	admin	23080
Response: [Customer (Id=L,FirstName=Hoyim...	2022-09-02 15:08:37	admin	23080
Request: FindOption(PrimeRate)	2022-09-02 15:09:16	admin	23080
Response: Option (PrimeRate,3.50)	2022-09-02 15:09:16	admin	23080
Request: Add Customer (Id=null,FirstName=Set...	2022-09-02 15:09:24	admin	23080
Response: OK	2022-09-02 15:09:24	admin	23080
Request: Find Customers ()	2022-09-02 15:09:28	admin	23080
Response: [Customer (Id=L,FirstName=Hoyim...	2022-09-02 15:10:17	admin	23080
Request: FindOption(PrimeRate)	2022-09-02 15:10:17	admin	23080
Response: Option (PrimeRate,3.50)	2022-09-02 15:10:22	admin	23080
Request: Add Customer (Id=null,FirstName=Joa...	2022-09-02 15:10:22	admin	23080
Response: OK	2022-09-02 15:10:22	admin	23080
Request: Find Customers ()	2022-09-02 15:10:25	admin	23080
Response: [Customer (Id=L,FirstName=Hoyim...	2022-09-02 15:11:04	admin	23080
Request: FindOption(PrimeRate)	2022-09-02 15:11:04	admin	23080
Response: Option (PrimeRate,3.50)	2022-09-02 15:11:09	admin	23080
Request: FindOption(PrimeRate)	2022-09-02 15:11:09	admin	23080
Response: Option (PrimeRate,3.50)	2022-09-02 15:11:15	admin	23080
Request: Add Customer (Id=null,FirstName=H...	2022-09-02 15:11:15	admin	23080
Response: OK	2022-09-02 15:11:15	admin	23080
Request: Find Customers ()	2022-09-02 15:11:18	admin	23080
Response: [Customer (Id=L,FirstName=Hoyim...	2022-09-02 15:11:18	admin	23080
Request: FindOption(PrimeRate)	2022-09-02 15:11:39	admin	23080

The output pane shows the execution details: 1 13:28:04 SELECT events.message, events.timestamp, users.username, events.user_id FROM events, users WHERE events.user_id = users.user_id AND users.us... 1000 row(s) returned. Duration / Fetch: 0.196 sec / 0.250 sec.

13. Snapshot of query to create a query using INNER JOIN for 'events' and 'users' tables. That query should display all messages, corresponding usernames, user_ids and timestamps sorted by username. Run this query against LoanApp, take a snapshot (that includes SELECT statement and results of the query) and add it to the journal.



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

1 SELECT events.message, users.username, users.user_id, events.timestamp
2 FROM events
3 INNER JOIN users ON events.user_id = users.user_id
4 GROUP BY username
5

```

The Results grid displays the following data:

message	USERNAME	USER_ID	timestamp
Login	adrian	23060	2022-09-02 16:05:36
Login	Asample	23652	2022-11-02 17:14:34
Login	SethRogen	23708	2022-11-02 17:15:14
Login	Young	23711	2022-11-02 17:16:01

The bottom panel shows the Action Output with the following details:

#	Time	Action	Message	Duration / Fetch
1	14:16:13	SELECT events.message, users.username, users.user_id, events.timestamp FROM events INNER JOIN users ON events.user_id = users.user_id GRO...	4 row(s) returned	0.094 sec / 0.000 sec