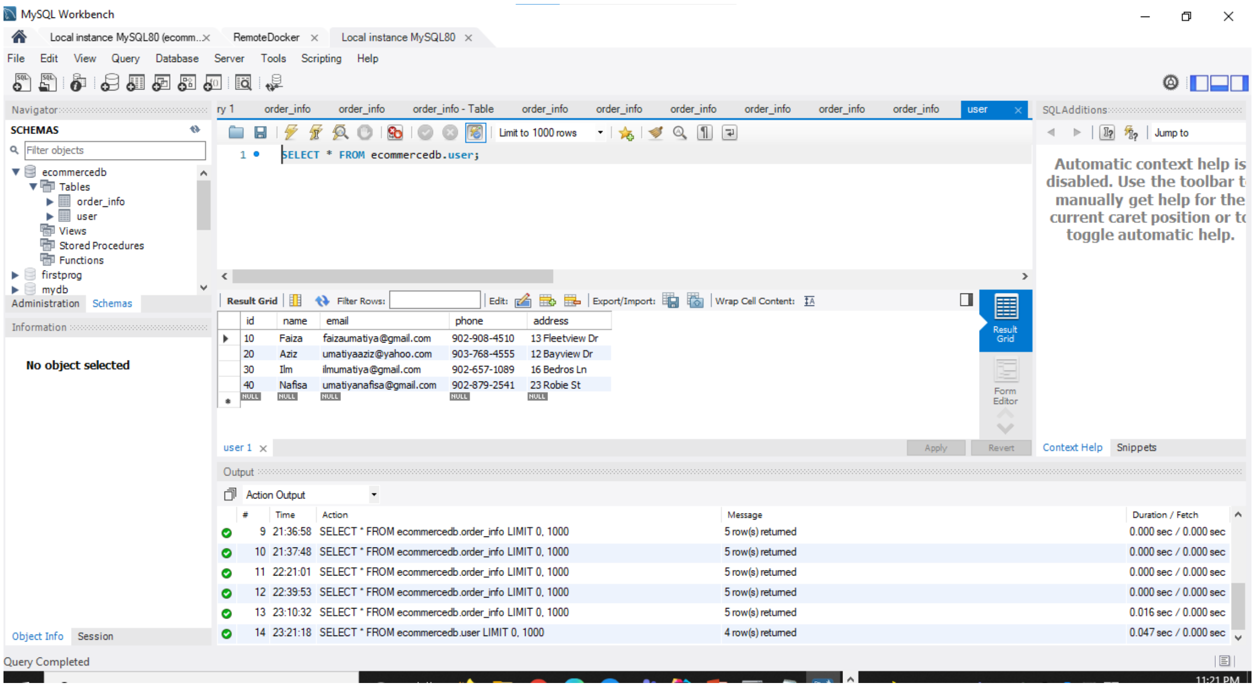
LAB TASK – REPORT:

Local instance tables:

User table:



```sql
SELECT * FROM ecommercedb.user;
```

id	name	email	phone	address
10	Faiza	faizaumatiya@gmail.com	902-908-4510	13 Fleetview Dr
20	Aziz	umatiyaaziz@yahoo.com	903-768-4555	12 Bayview Dr
30	Ilm	ilmumatiya@gmail.com	902-657-1089	16 Bedros Ln
40	Nafisa	umatiyanafisa@gmail.com	902-879-2541	23 Robie St
NULL	NULL	NULL	NULL	NULL

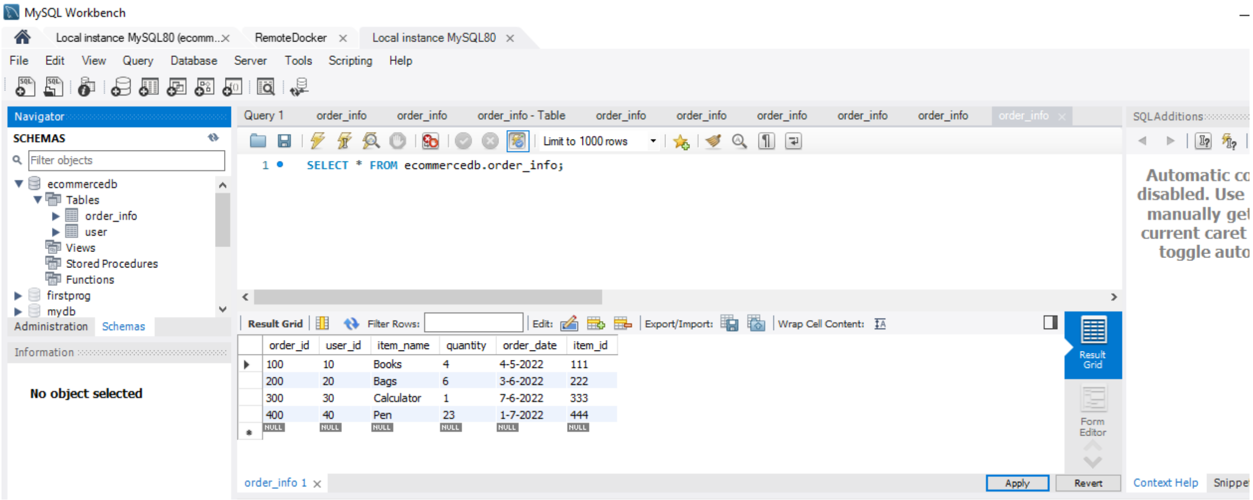
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or to toggle automatic help.

No object selected

```
 #  Time      Action                                               Message              Duration / Fetch
 9  21:36:58  SELECT * FROM ecommercedb.order_info LIMIT 0, 1000  5 row(s) returned    0.000 sec / 0.000 sec
10  21:37:48  SELECT * FROM ecommercedb.order_info LIMIT 0, 1000  5 row(s) returned    0.000 sec / 0.000 sec
11  22:21:01  SELECT * FROM ecommercedb.order_info LIMIT 0, 1000  5 row(s) returned    0.000 sec / 0.000 sec
12  22:39:53  SELECT * FROM ecommercedb.order_info LIMIT 0, 1000  5 row(s) returned    0.000 sec / 0.000 sec
13  23:10:32  SELECT * FROM ecommercedb.order_info LIMIT 0, 1000  5 row(s) returned    0.016 sec / 0.000 sec
14  23:21:18  SELECT * FROM ecommercedb.user LIMIT 0, 1000        4 row(s) returned    0.047 sec / 0.000 sec
```

Query Completed

Order_info table:



```sql
SELECT * FROM ecommercedb.order_info;
```

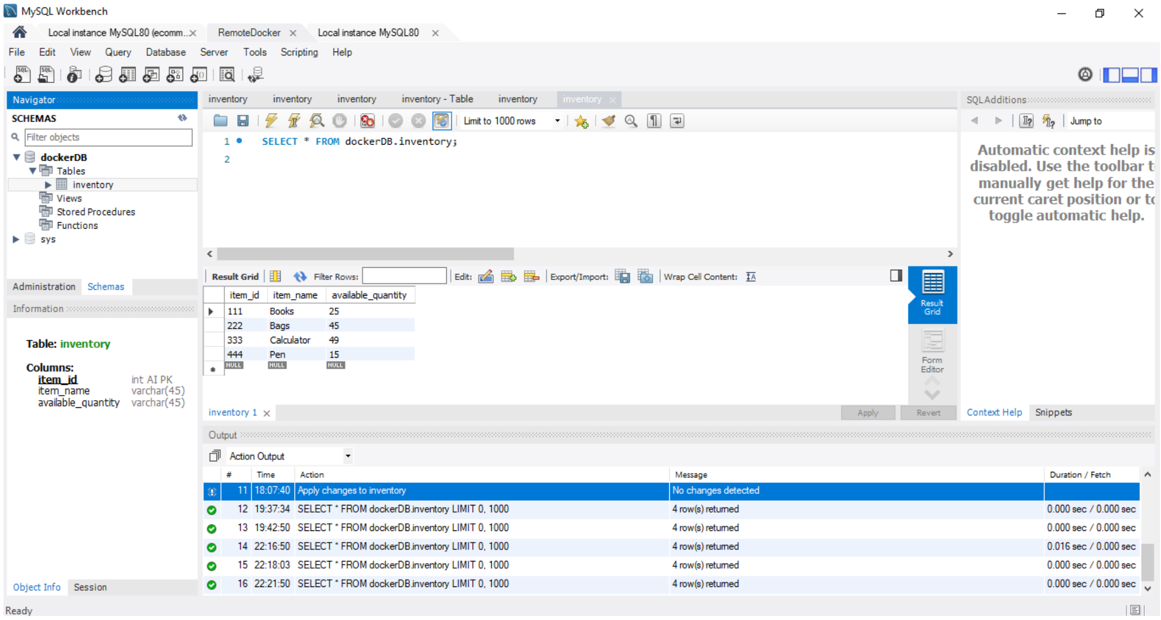
order_id	user_id	item_name	quantity	order_date	item_id
100	10	Books	4	4-5-2022	111
200	20	Bags	6	3-6-2022	222
300	30	Calculator	1	7-6-2022	333
400	40	Pen	23	1-7-2022	444
NULL	NULL	NULL	NULL	NULL	NULL

Automatic co disabled. Use manually get current caret toggle auto

No object selected

Remote Database table:

Inventory table:



```sql
SELECT * FROM dockerDB.inventory;
```

item_id	item_name	available_quantity
111	Books	25
222	Bags	45
333	Calculator	49
444	Pen	15
NULL	NULL	NULL

Table: inventory

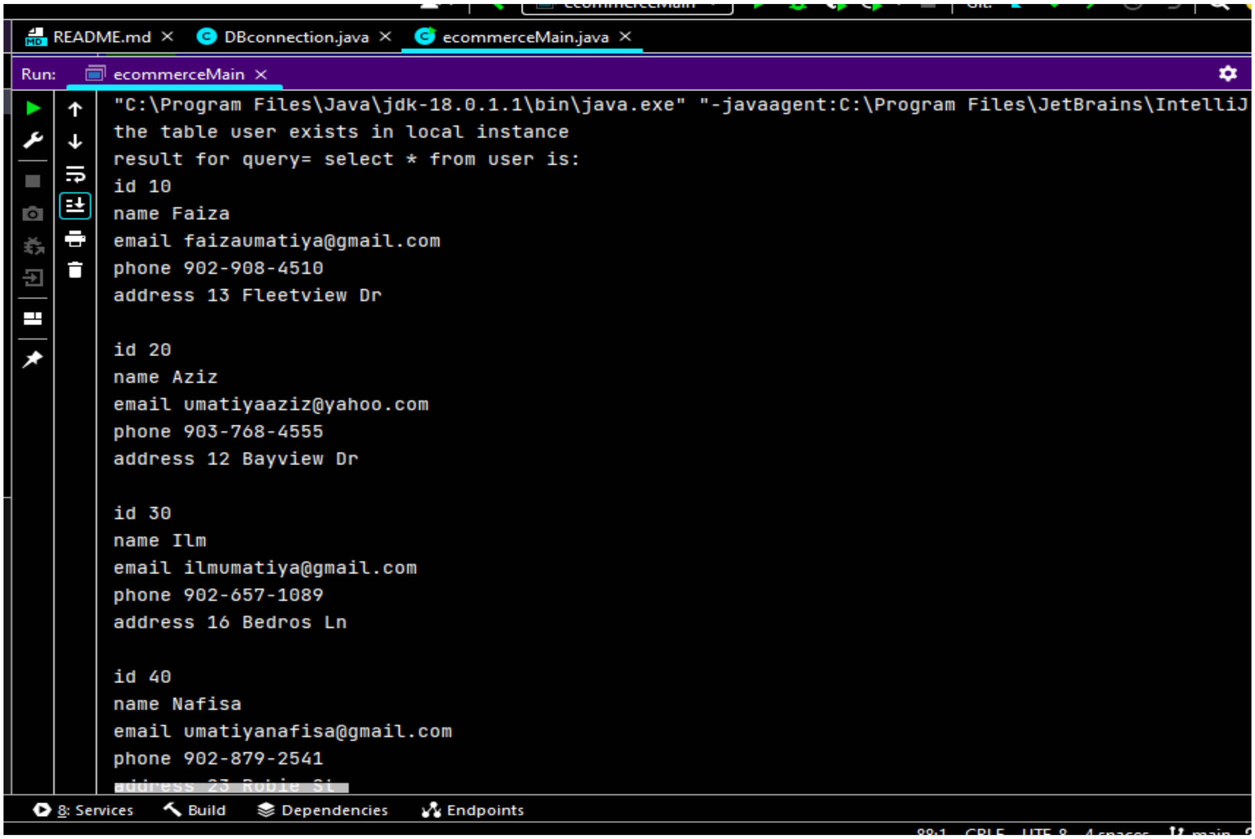
Columns:

item_id	int AI PK
item_name	varchar(45)
available_quantity	varchar(45)

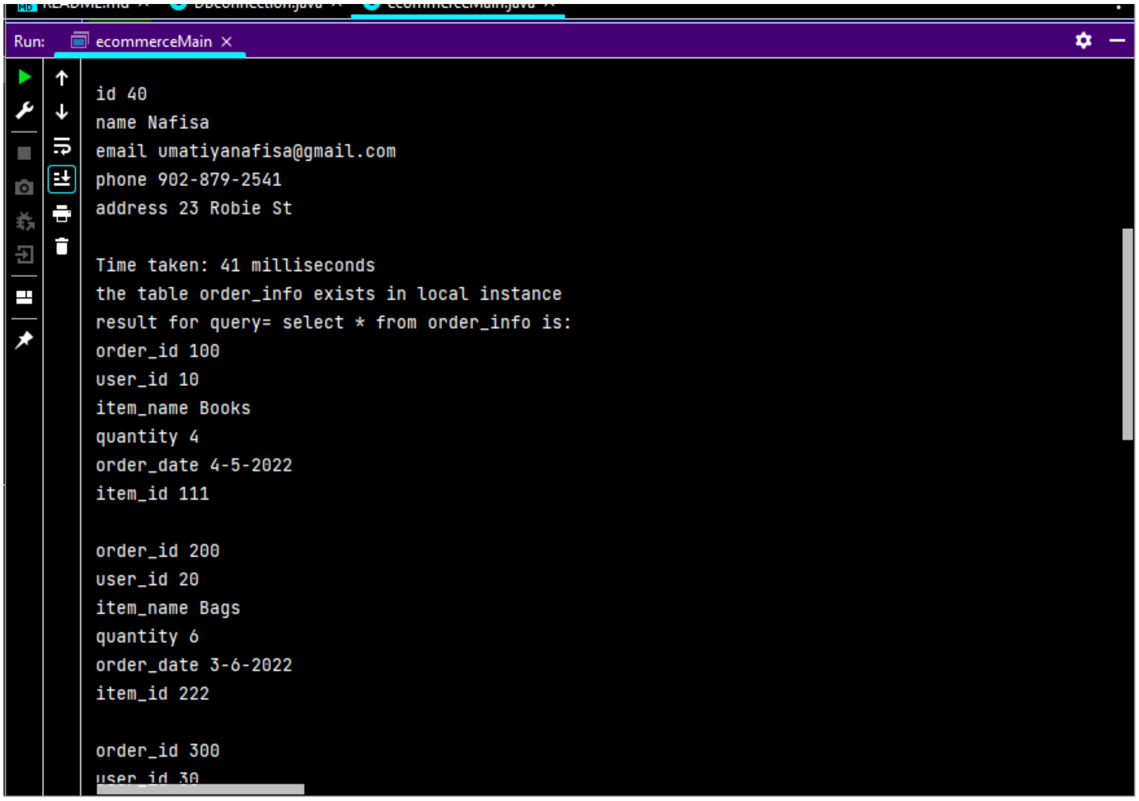
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

#	Time	Action	Message	Duration / Fetch
11	18:07:40	Apply changes to inventory	No changes detected	
12	19:37:34	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
13	19:42:50	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
14	22:16:50	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.016 sec / 0.000 sec
15	22:18:03	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
16	22:21:50	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

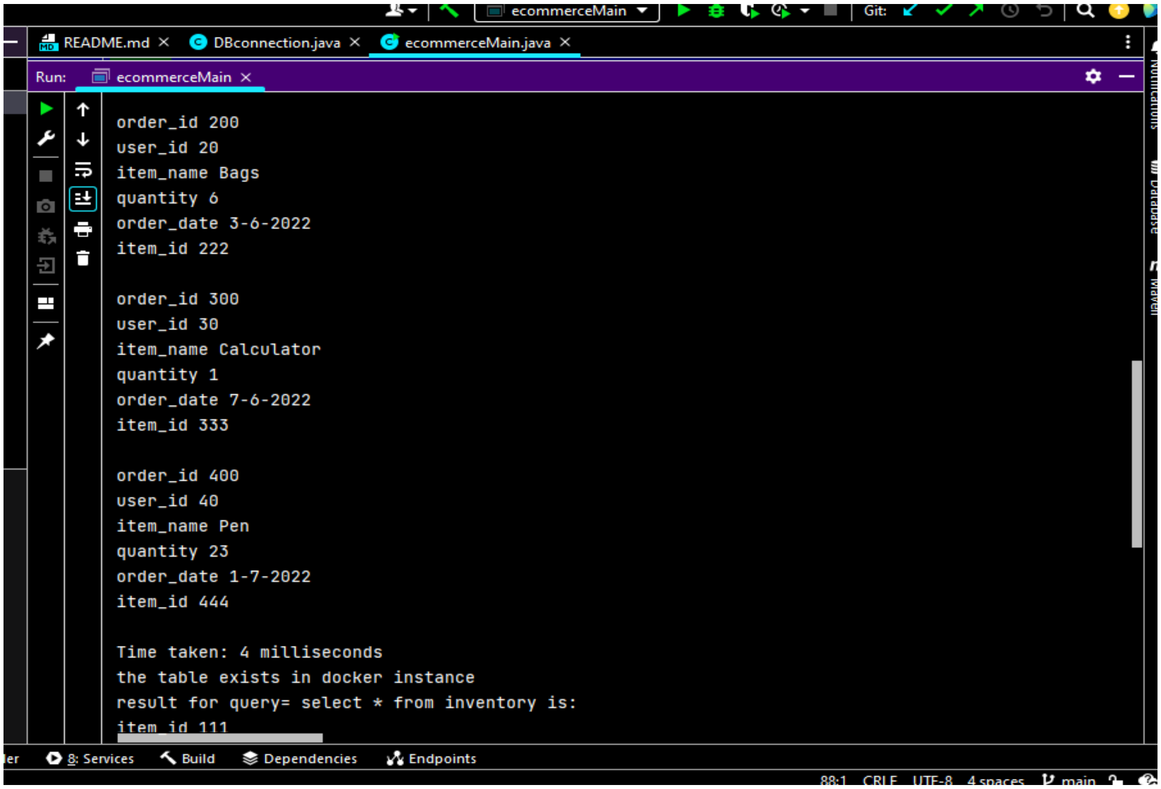
Q.1) Fetches item details from the remote database.



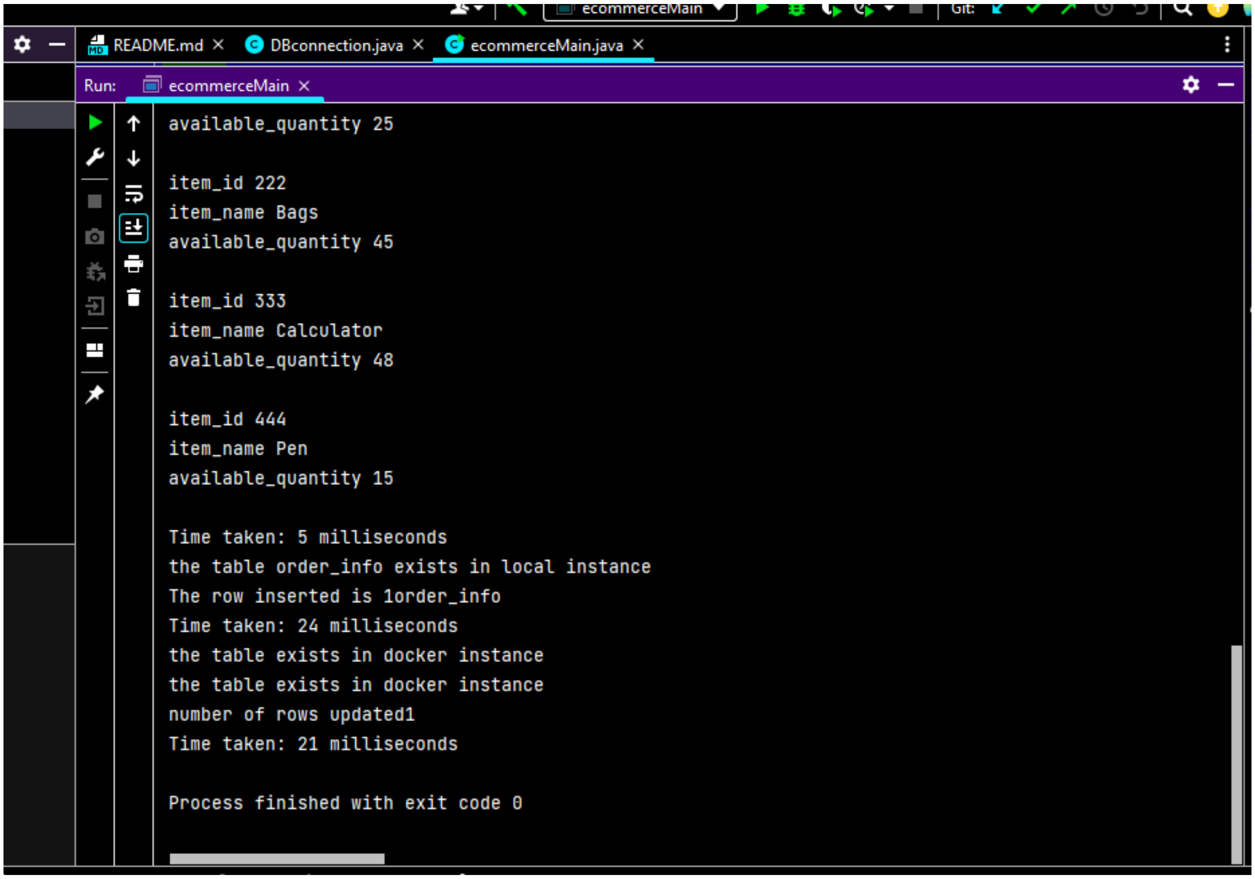
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ
the table user exists in local instance
result for query= select * from user is:
id 10
name Faiza
email faizaumatiya@gmail.com
phone 902-908-4510
address 13 Fleetview Dr

id 20
name Aziz
email umatiyaaziz@yahoo.com
phone 903-768-4555
address 12 Bayview Dr

id 30
name Ilm
email ilmumatiya@gmail.com
phone 902-657-1089
address 16 Bedros Ln

id 40
name Nafisa
email umatiyanafisa@gmail.com
phone 902-879-2541
address 23 Robie St
```



```
id 40
name Nafisa
email umatiyanafisa@gmail.com
phone 902-879-2541
address 23 Robie St

Time taken: 41 milliseconds
the table order_info exists in local instance
result for query= select * from order_info is:
order_id 100
user_id 10
item_name Books
quantity 4
order_date 4-5-2022
item_id 111

order_id 200
user_id 20
item_name Bags
quantity 6
order_date 3-6-2022
item_id 222

order_id 300
user_id 30
```



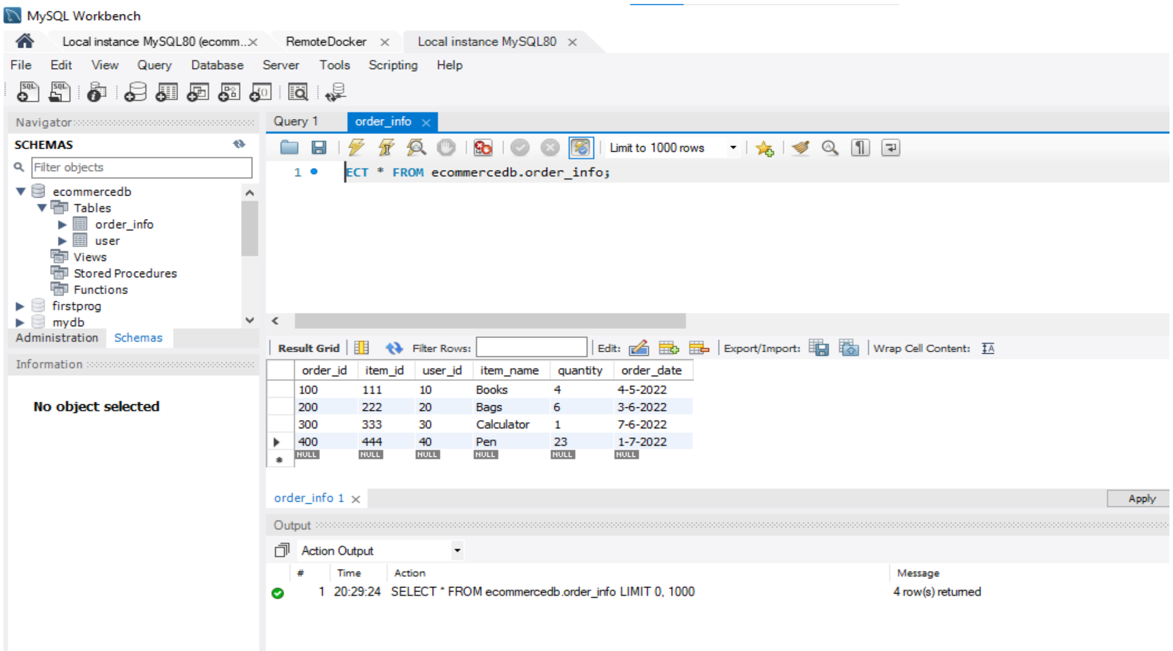
```
order_id 200
user_id 20
item_name Bags
quantity 6
order_date 3-6-2022
item_id 222

order_id 300
user_id 30
item_name Calculator
quantity 1
order_date 7-6-2022
item_id 333

order_id 400
user_id 40
item_name Pen
quantity 23
order_date 1-7-2022
item_id 444

Time taken: 4 milliseconds
the table exists in docker instance
result for query= select * from inventory is:
item_id 111
```



```
available_quantity 25

item_id 222
item_name Bags
available_quantity 45

item_id 333
item_name Calculator
available_quantity 48

item_id 444
item_name Pen
available_quantity 15

Time taken: 5 milliseconds
the table order_info exists in local instance
The row inserted is 1order_info
Time taken: 24 milliseconds
the table exists in docker instance
the table exists in docker instance
number of rows updated1
Time taken: 21 milliseconds

Process finished with exit code 0
```

Q.2) Creates an order in local database

Before Update of the order_info table:



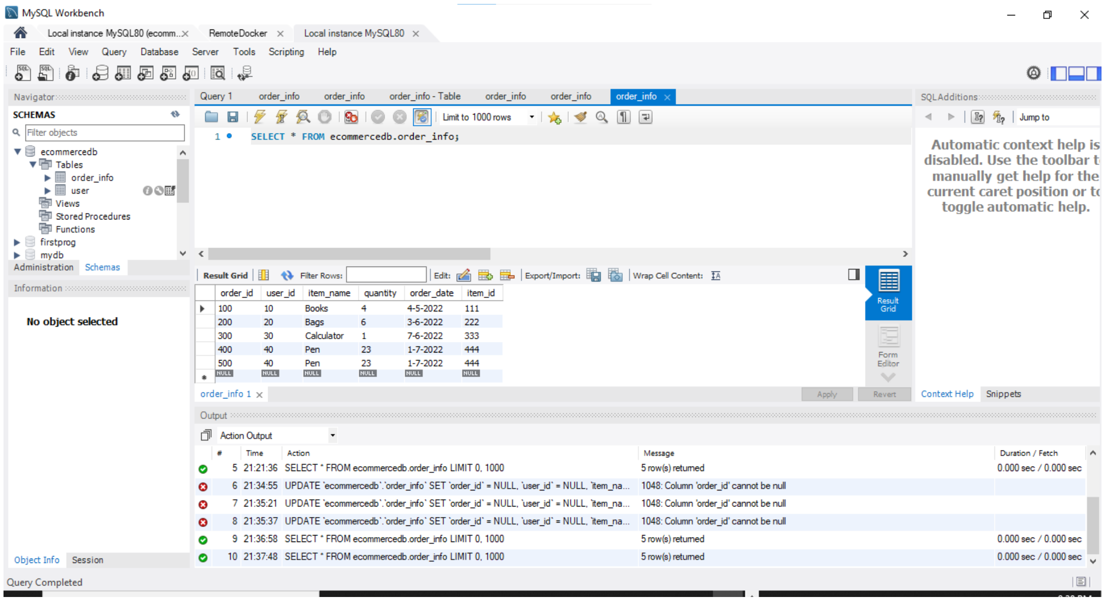
```
1  ECT * FROM ecommercedb.order_info;
```

order_id	item_id	user_id	item_name	quantity	order_date
100	111	10	Books	4	4-5-2022
200	222	20	Bags	6	3-6-2022
300	333	30	Calculator	1	7-6-2022
400	444	40	Pen	23	1-7-2022
NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message
1	20:29:24	SELECT * FROM ecommercedb.order_info LIMIT 0, 1000	4 row(s) returned

After update of order_info table:

1 row added.



```
SELECT * FROM ecommercedb.order_info;
```

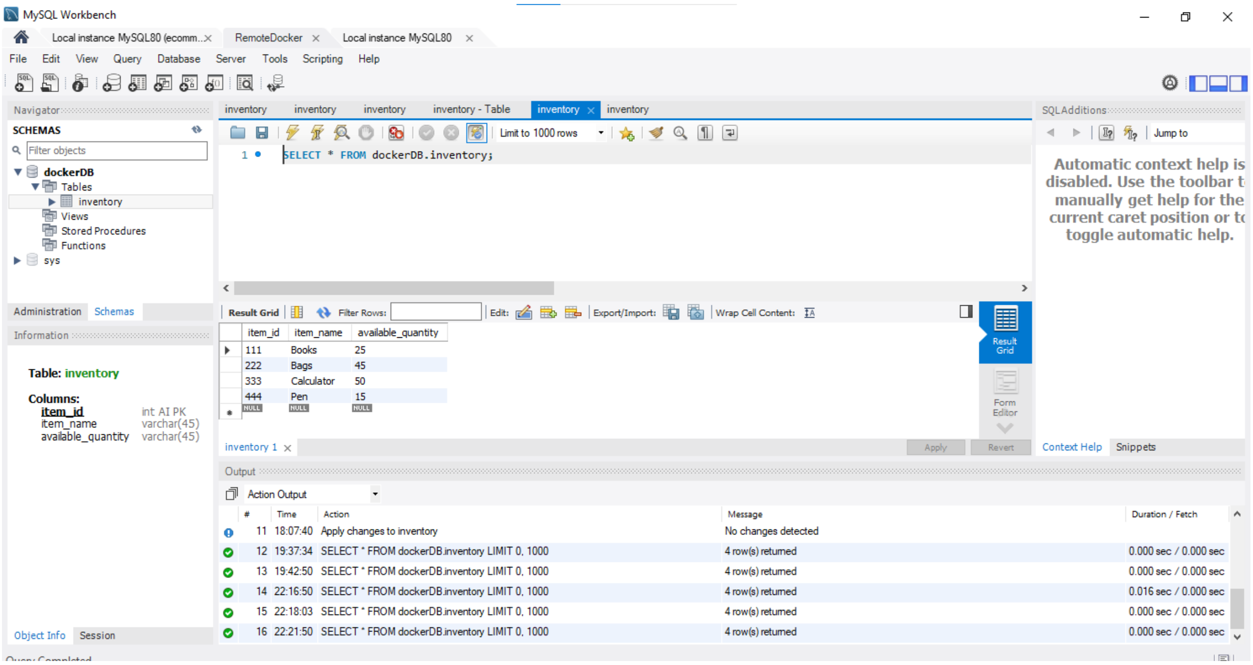
order_id	user_id	item_name	quantity	order_date	item_id
100	10	Books	4	4-5-2022	111
200	20	Bags	6	3-6-2022	222
300	30	Calculator	1	7-6-2022	333
400	40	Pen	23	1-7-2022	444
500	40	Pen	23	1-7-2022	444
NULL	NULL	NULL	NULL	NULL	NULL

```
#   Time      Action                                                                     Message                               Duration / Fetch
5   21:21:36  SELECT * FROM ecommercedb.order_info LIMIT 0, 1000                         5 row(s) returned                     0.000 sec / 0.000 sec
6   21:34:55  UPDATE `ecommercedb`.`order_info` SET `order_id` = NULL, `user_id` = NULL, `item_na...   1048: Column 'order_id' cannot be null
7   21:35:21  UPDATE `ecommercedb`.`order_info` SET `order_id` = NULL, `user_id` = NULL, `item_na...   1048: Column 'order_id' cannot be null
8   21:35:37  UPDATE `ecommercedb`.`order_info` SET `order_id` = NULL, `user_id` = NULL, `item_na...   1048: Column 'order_id' cannot be null
9   21:36:58  SELECT * FROM ecommercedb.order_info LIMIT 0, 1000                         5 row(s) returned                     0.000 sec / 0.000 sec
10  21:37:48  SELECT * FROM ecommercedb.order_info LIMIT 0, 1000                         5 row(s) returned                     0.000 sec / 0.000 sec
```

Query Completed

Q.3) Writes the updated quantity back to the remote database upon order creation

Before quantity update:



```
SELECT * FROM dockerDB.inventory;
```

Table: inventory

Columns:
item_id int AI PK
item_name varchar(45)
available_quantity varchar(45)

item_id	item_name	available_quantity
111	Books	25
222	Bags	45
333	Calculator	50
444	Pen	15
NULL	NULL	NULL

```
#   Time      Action                                            Message              Duration / Fetch
11  18:07:40  Apply changes to inventory                        No changes detected
12  19:37:34  SELECT * FROM dockerDB.inventory LIMIT 0, 1000    4 row(s) returned    0.000 sec / 0.000 sec
13  19:42:50  SELECT * FROM dockerDB.inventory LIMIT 0, 1000    4 row(s) returned    0.000 sec / 0.000 sec
14  22:16:50  SELECT * FROM dockerDB.inventory LIMIT 0, 1000    4 row(s) returned    0.016 sec / 0.000 sec
15  22:18:03  SELECT * FROM dockerDB.inventory LIMIT 0, 1000    4 row(s) returned    0.000 sec / 0.000 sec
16  22:21:50  SELECT * FROM dockerDB.inventory LIMIT 0, 1000    4 row(s) returned    0.000 sec / 0.000 sec
```

After quantity Update:

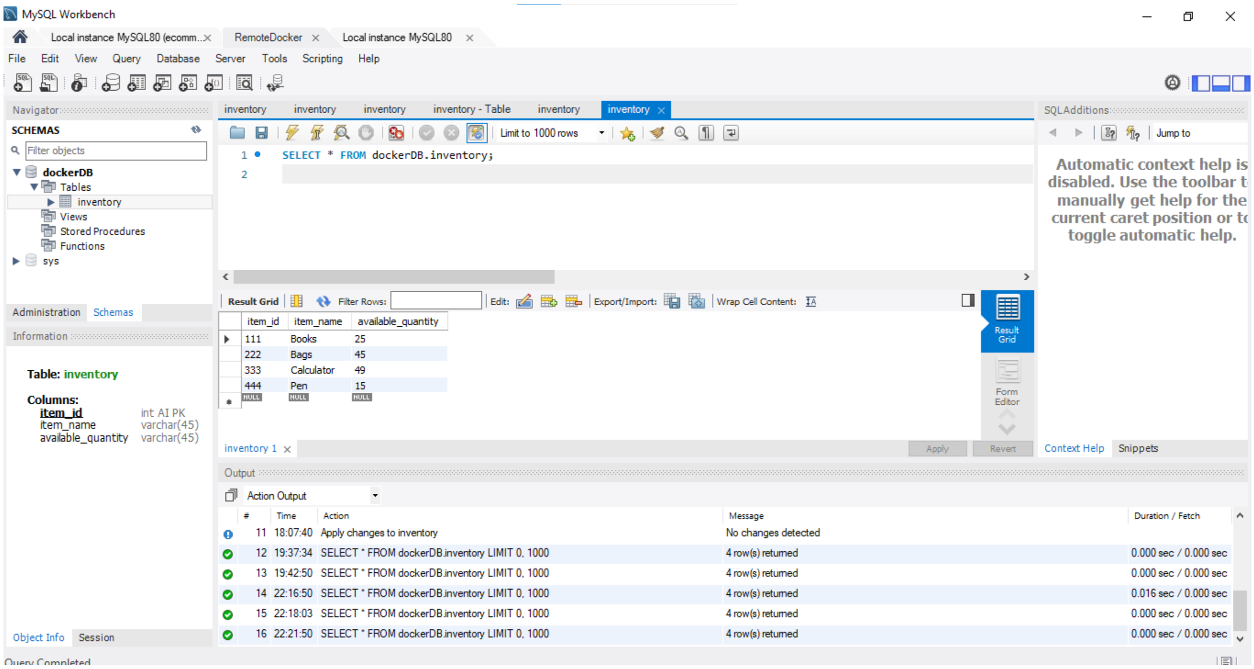


Table: inventory

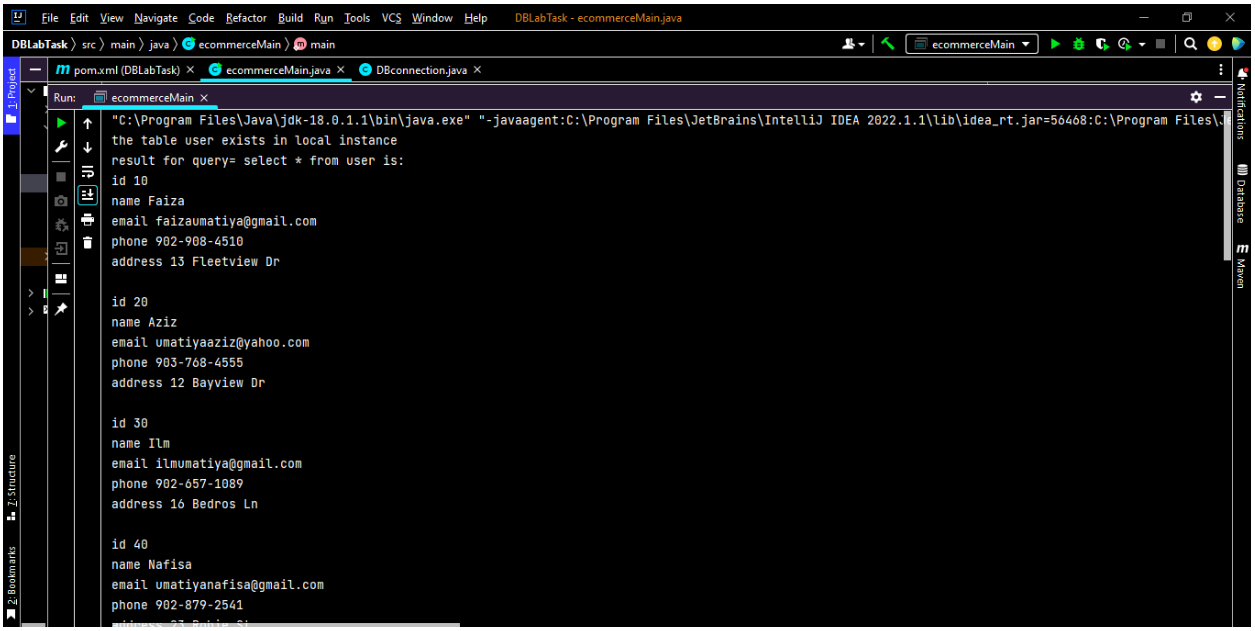
Columns:
item_id int AI PK
item_name varchar(45)
available_quantity varchar(45)

```
1  SELECT * FROM dockerDB.inventory;
2
```

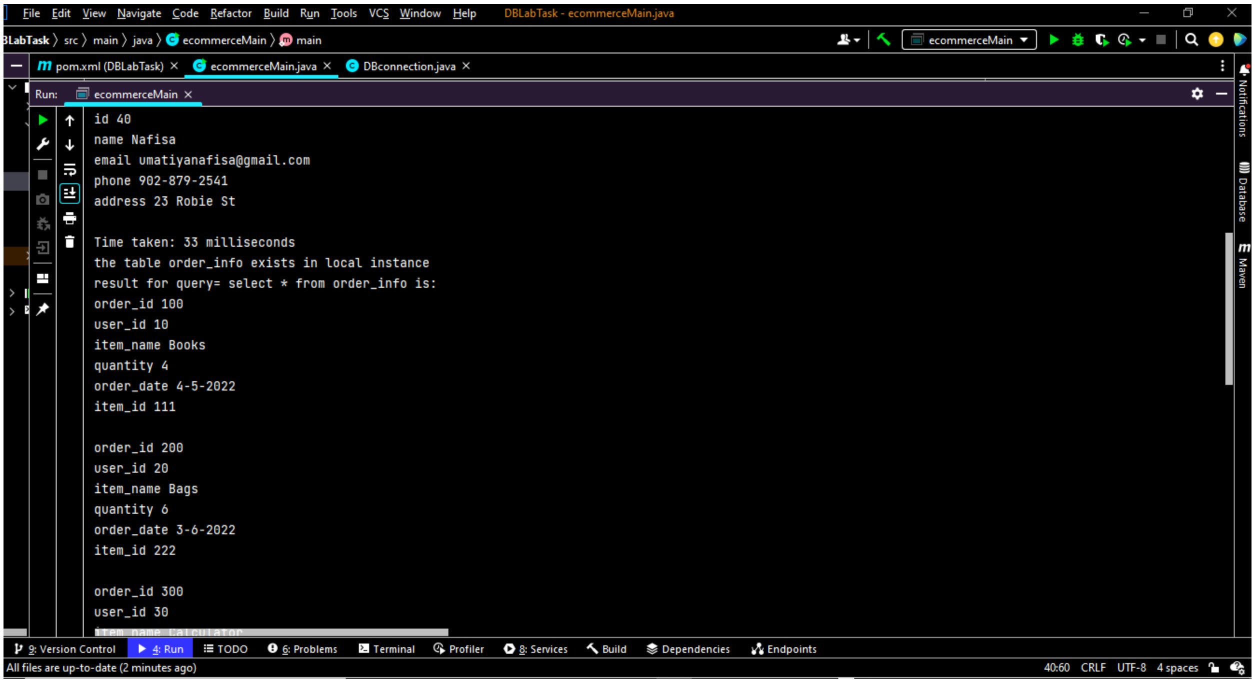
item_id	item_name	available_quantity
111	Books	25
222	Bags	45
333	Calculator	49
444	Pen	15
NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
11	18:07:40	Apply changes to inventory	No changes detected	
12	19:37:34	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
13	19:42:50	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
14	22:16:50	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.016 sec / 0.000 sec
15	22:18:03	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
16	22:21:50	SELECT * FROM dockerDB.inventory LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

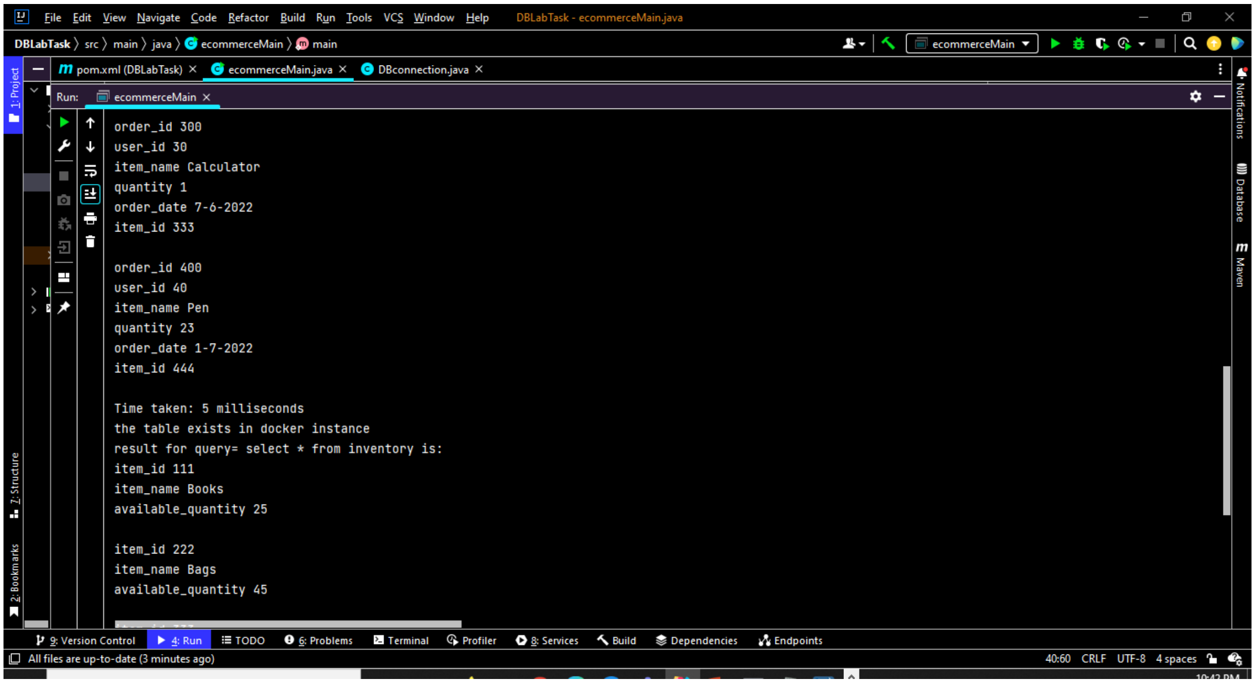
Q.4) Printing the timestamp after every Query:


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
DBLabTask - ecommerceMain.java
DBLabTask › src › main › java › ecommerceMain › main
ecommerceMain
pom.xml (DBLabTask)
ecommerceMain.java
DBconnection.java
Run:
ecommerceMain
1: Project
7: Structure
2: Bookmarks
Notifications
Database
Maven

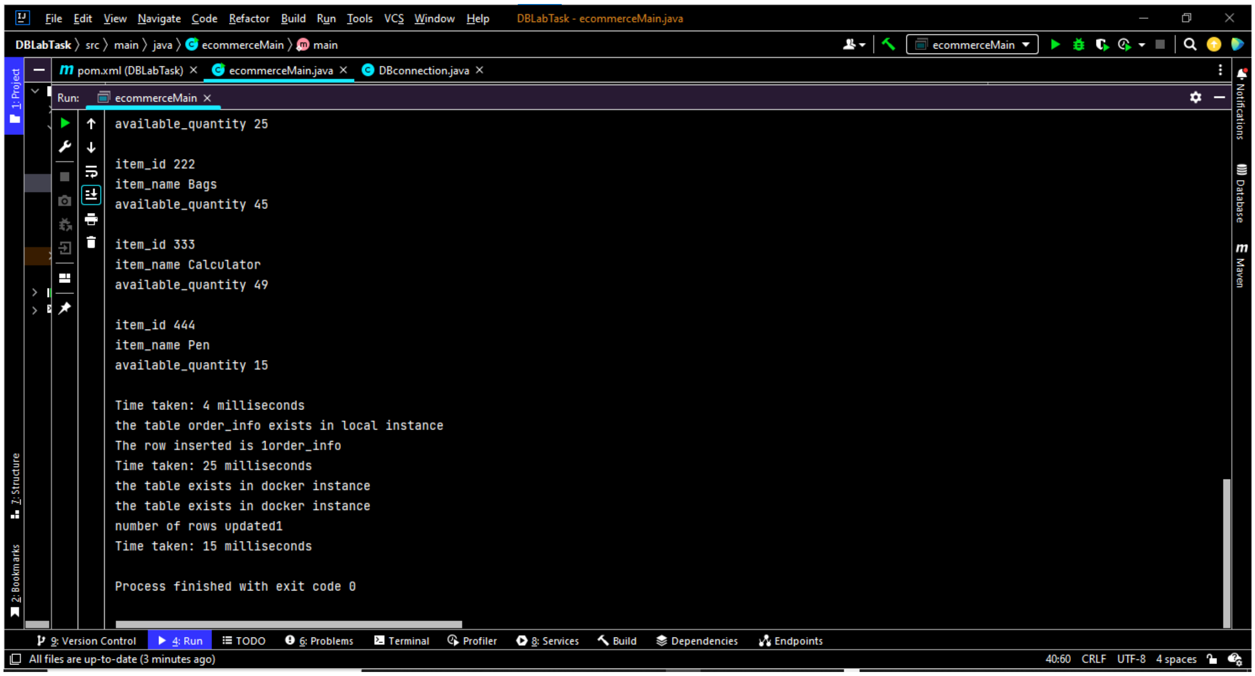
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.1.1\lib\idea_rt.jar=56468:C:\Program Files\J
the table user exists in local instance
result for query= select * from user is:
id 10
name Faiza
email faizaumatiya@gmail.com
phone 902-908-4510
address 13 Fleetview Dr

id 20
name Aziz
email umatiyaaziz@yahoo.com
phone 903-768-4555
address 12 Bayview Dr

id 30
name Ilm
email ilmumatiya@gmail.com
phone 902-657-1089
address 16 Bedros Ln

id 40
name Nafisa
email umatiyanafisa@gmail.com
phone 902-879-2541
address 23 Rubix St
```


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
DBLabTask - ecommerceMain.java
BLabTask 〉 src 〉 main 〉 java 〉 ecommerceMain 〉 main
ecommerceMain
pom.xml (DBLabTask)
ecommerceMain.java
DBconnection.java
Run:
ecommerceMain
Notifications
Database
Maven
9: Version Control
4: Run
TODO
6: Problems
Terminal
Profiler
8: Services
Build
Dependencies
Endpoints
All files are up-to-date (2 minutes ago)
40:60 CRLF UTF-8 4 spaces

```
id 40
name Nafisa
email umatiyanafisa@gmail.com
phone 902-879-2541
address 23 Robie St

Time taken: 33 milliseconds
the table order_info exists in local instance
result for query= select * from order_info is:
order_id 100
user_id 10
item_name Books
quantity 4
order_date 4-5-2022
item_id 111

order_id 200
user_id 20
item_name Bags
quantity 6
order_date 3-6-2022
item_id 222

order_id 300
user_id 30
item_name Calculator
```



```
order_id 300
user_id 30
item_name Calculator
quantity 1
order_date 7-6-2022
item_id 333

order_id 400
user_id 40
item_name Pen
quantity 23
order_date 1-7-2022
item_id 444

Time taken: 5 milliseconds
the table exists in docker instance
result for query= select * from inventory is:
item_id 111
item_name Books
available_quantity 25

item_id 222
item_name Bags
available_quantity 45

item_id 333
```



```
available_quantity 25

item_id 222
item_name Bags
available_quantity 45

item_id 333
item_name Calculator
available_quantity 49

item_id 444
item_name Pen
available_quantity 15

Time taken: 4 milliseconds
the table order_info exists in local instance
The row inserted is 1order_info
Time taken: 25 milliseconds
the table exists in docker instance
the table exists in docker instance
number of rows updated1
Time taken: 15 milliseconds

Process finished with exit code 0
```

Git Link:

https://git.cs.dal.ca/umatiya/csci-5408-lab.git

Reference:

1. https://stackoverflow.com/questions/4927856/how-can-i-calculate-a-time-difference-in-java
2. https://mkyong.com/jdbc/jdbc-preparestatement-example-update-a-record/
3. https://dal.brightspace.com/d2l/le/content/221749/viewContent/3040910/View