CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

LAB-4: DISTRIBUTED DATABASE IMPLEMENTATION

Table of Contents

1: Set up a simple e-commerce database system

Create Database at local

create database locallab4;

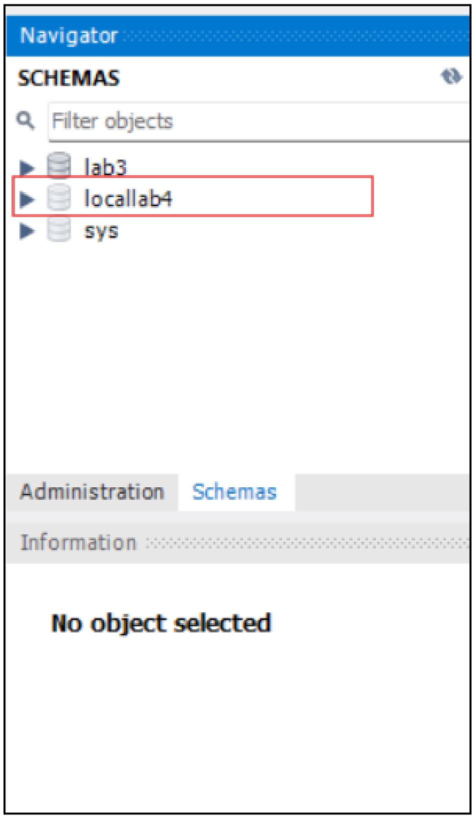
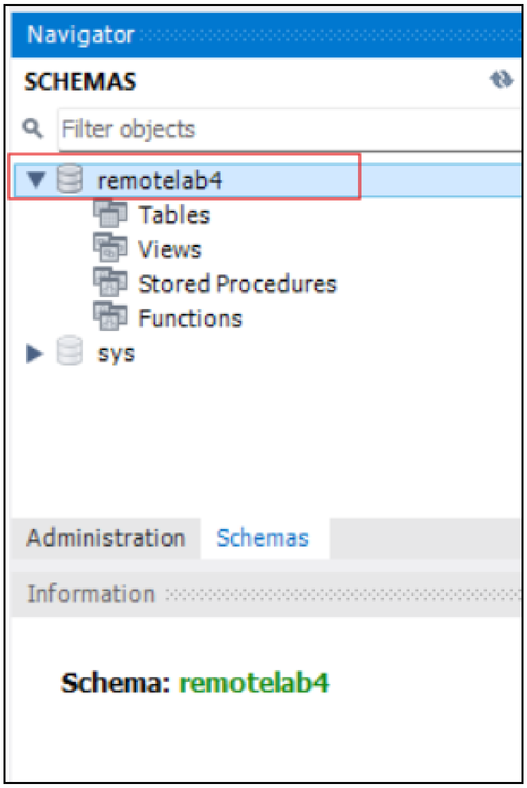


Figure 1: Database locallab4 Created

Create Database at remote

CREATE DATABASE remotelab4;



Schema: remotelab4

Figure 2: Database remotelab4 Created

Create Table User at local

```
CREATE TABLE User (
 id INT PRIMARY KEY AUTO_INCREMENT,
 name VARCHAR(255) NOT NULL,
 email VARCHAR(255) UNIQUE NOT NULL,
 phone VARCHAR(255),
 address VARCHAR(255)
);
```

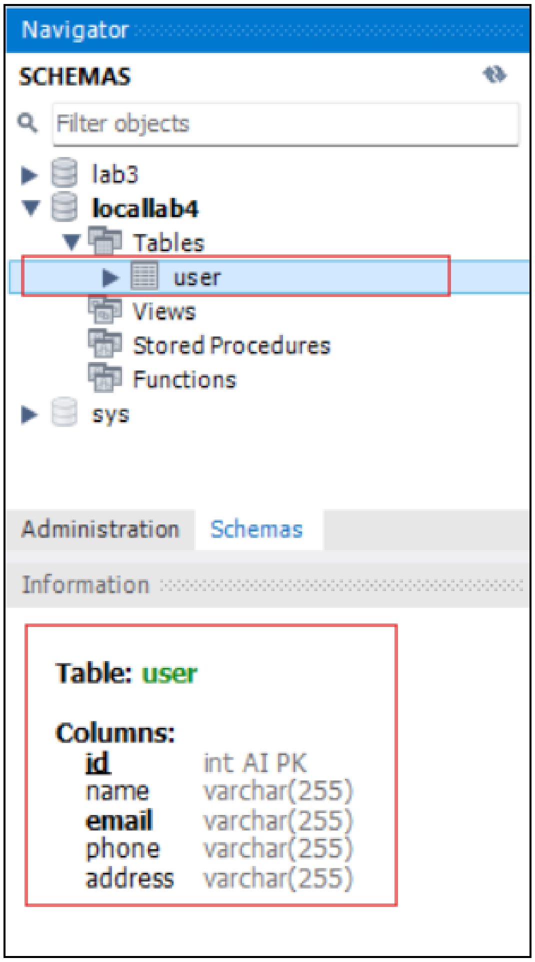


Table: user

Columns:

```
id       int AI PK
name     varchar(255)
email    varchar(255)
phone    varchar(255)
address  varchar(255)
```

Figure 3: Table user Created

Create Table order_info at local

```
CREATE TABLE Order_info (
 order_id INT PRIMARY KEY AUTO_INCREMENT,
 user_id INT,
 item_name VARCHAR(255) NOT NULL,
 quantity INT NOT NULL,
 order_date DATETIME NOT NULL
);
```

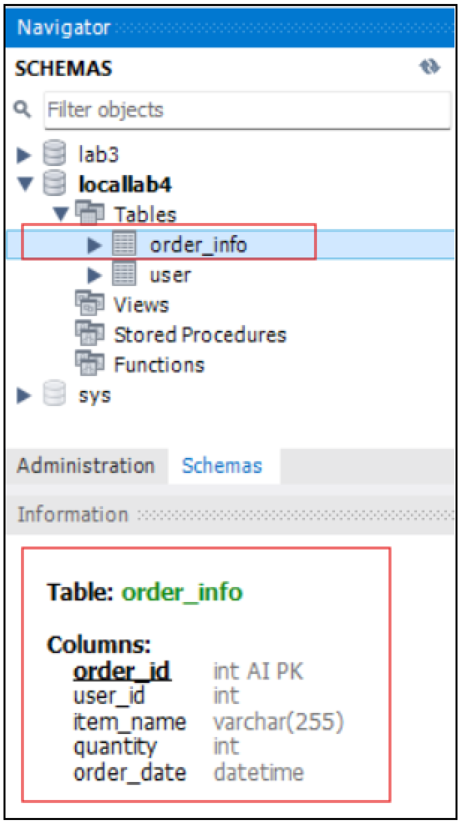


Table: order_info

Columns:

```
order_id     int AI PK
user_id      int
item_name    varchar(255)
quantity     int
order_date   datetime
```

Figure 4: Table order_info Created

Create Table Inventory at remote

```
CREATE TABLE Inventory(
item_id INT PRIMARY KEY,
item_name VARCHAR(15),
available_quantity INT
);
```

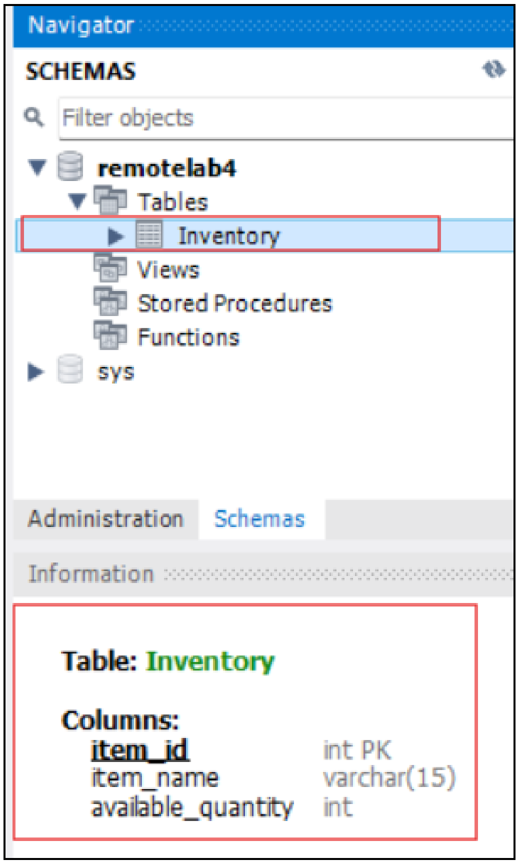


Table: Inventory

Columns:

```
item_id             int PK
item_name           varchar(15)
available_quantity  int
```

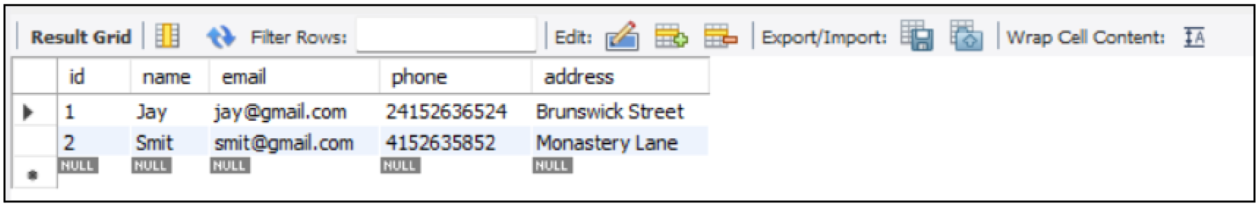
Figure 5: Table Inventory Created

2: Insert Dummy Data

Insert Dummy Data into the user table

INSERT INTO User (name, email, phone, address) VALUES ('Jay', 'jay@gmail.com', '24152636524', 'Brunswick Street');

INSERT INTO User (name, email, phone, address) VALUES ('Smit', 'smit@gmail.com', '4152635852', 'Monastery Lane');



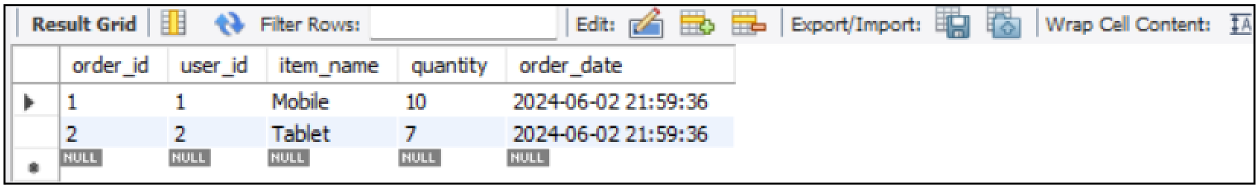
id	name	email	phone	address
1	Jay	jay@gmail.com	24152636524	Brunswick Street
2	Smit	smit@gmail.com	4152635852	Monastery Lane
NULL	NULL	NULL	NULL	NULL

Figure 6: Insert Dummy Data into user Table

Insert Dummy Data into the order_info table

INSERT INTO Order_info (user_id, item_name, quantity, order_date) VALUES (1, 'Mobile', 10, now());

INSERT INTO Order_info (user_id, item_name, quantity, order_date) VALUES (2, 'Tablet', 7, now());



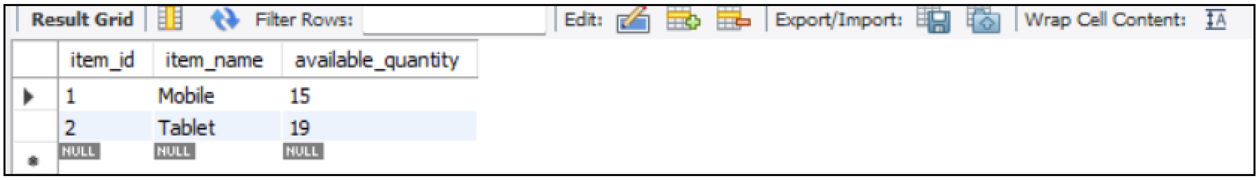
order_id	user_id	item_name	quantity	order_date
1	1	Mobile	10	2024-06-02 21:59:36
2	2	Tablet	7	2024-06-02 21:59:36
NULL	NULL	NULL	NULL	NULL

Figure 7: Insert Dummy Data into order_info Table

Insert Dummy Data into the Inventory table

```
INSERT INTO Inventory (item_id, item_name, available_quantity)
VALUES (1, 'Mobile', 15);

INSERT INTO Inventory (item_id, item_name, available_quantity)
VALUES (2, 'Tablet', 19);
```



item_id	item_name	available_quantity
1	Mobile	15
2	Tablet	19
NULL	NULL	NULL

Figure 8: Insert Dummy Data into Inventory Table

3: Java Program

Program Flow

Step 1: Connect to the local database (locallab4) and the remote database (remotelab4).

Step 2: Execute a query to fetch data from the Inventory table in remotelab4. Display the retrieved data and the time taken for this operation on the terminal.

Step 3: Prompt the user to enter their user ID, the item name they want to buy, and the desired quantity.

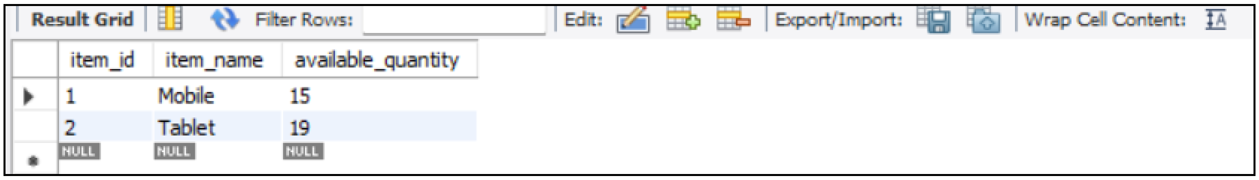
Step 4: Insert the new order information into the Order_info table in locallab4. Measure and display the time taken for this insertion.

Step 5: Update the Inventory table in remotelab4 by decreasing the available quantity of the specified item. Measure and display the time taken for this update.

Step 6: Handle any exceptions that occur during the above steps and print the error message.

Step 7: Close Connections.

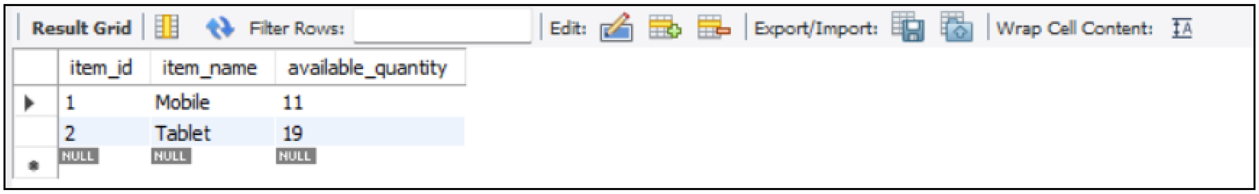
order_info table (Before)



item_id	item_name	available_quantity
1	Mobile	15
2	Tablet	19
NULL	NULL	NULL

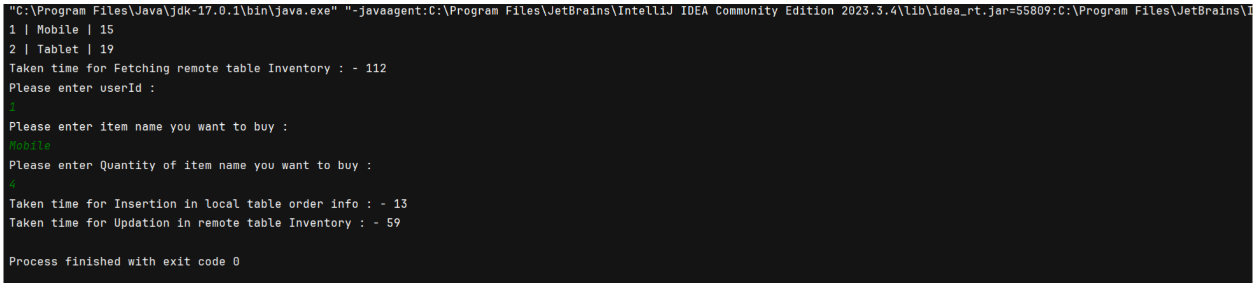
Figure 9: Table order_info before order placed

order_info table (After)



item_id	item_name	available_quantity
1	Mobile	11
2	Tablet	19
NULL	NULL	NULL

Figure 10: Table order_info after order placed

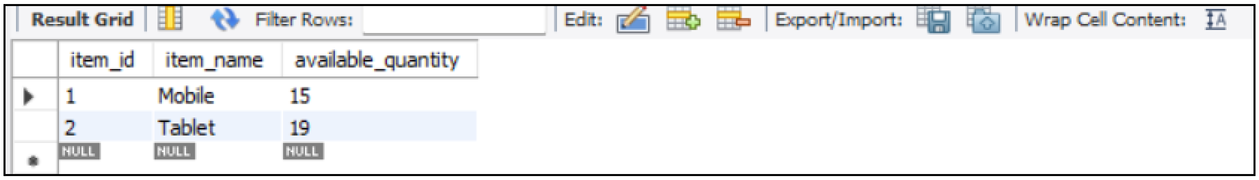


```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.4\lib\idea_rt.jar=55809:C:\Program Files\JetBrains\I
1 | Mobile | 15
2 | Tablet | 19
Taken time for Fetching remote table Inventory : - 112
Please enter userId :
1
Please enter item name you want to buy :
Mobile
Please enter Quantity of item name you want to buy :
4
Taken time for Insertion in local table order info : - 13
Taken time for Updation in remote table Inventory : - 59

Process finished with exit code 0
```

Figure 11: Java Code Terminal Output

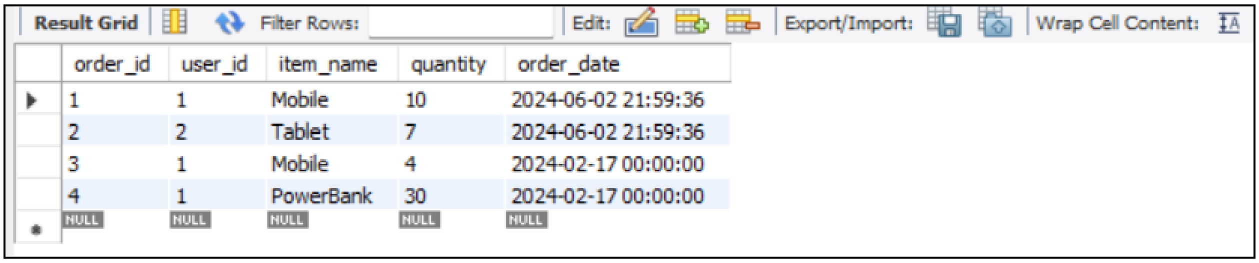
Inventory Table (Before)



item_id	item_name	available_quantity
1	Mobile	15
2	Tablet	19
NULL	NULL	NULL

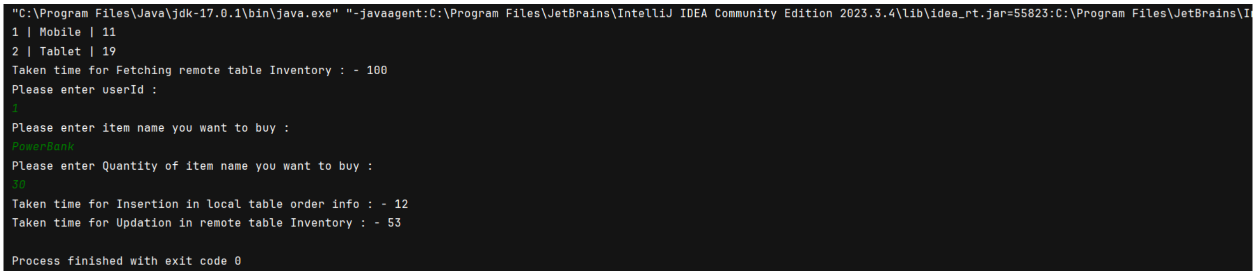
Figure 12: Table inventory before order placed

Inventory Table (After)



order_id	user_id	item_name	quantity	order_date
1	1	Mobile	10	2024-06-02 21:59:36
2	2	Tablet	7	2024-06-02 21:59:36
3	1	Mobile	4	2024-02-17 00:00:00
4	1	PowerBank	30	2024-02-17 00:00:00
NULL	NULL	NULL	NULL	NULL

Figure 13: Table inventory after order placed



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.4\lib\idea_rt.jar=55823:C:\Program Files\JetBrains\I
1 | Mobile | 11
2 | Tablet | 19
Taken time for Fetching remote table Inventory : - 100
Please enter userId :
1
Please enter item name you want to buy :
PowerBank
Please enter Quantity of item name you want to buy :
30
Taken time for Insertion in local table order info : - 12
Taken time for Updation in remote table Inventory : - 53

Process finished with exit code 0
```

Figure 14: Java Code Terminal Output

Why there is the difference in execution of time: -

- Local Database: When performing operations on a local database, the data transfer occurs within the same machine or local network, while that operation remote involves data transfer over the internet, which introduces additional latency due to the physical distance and network hops between your system and the remote server.