Introduction to Database Systems - CSC 675/775
Spring 2022

PROJECT TITLE: ONLINE BOOK STORE APPLICATION

Team Members: Hirva Patel
Aarshil Patel
Jasmine Jahan
Umar Rama
Gabriel Gonzalez

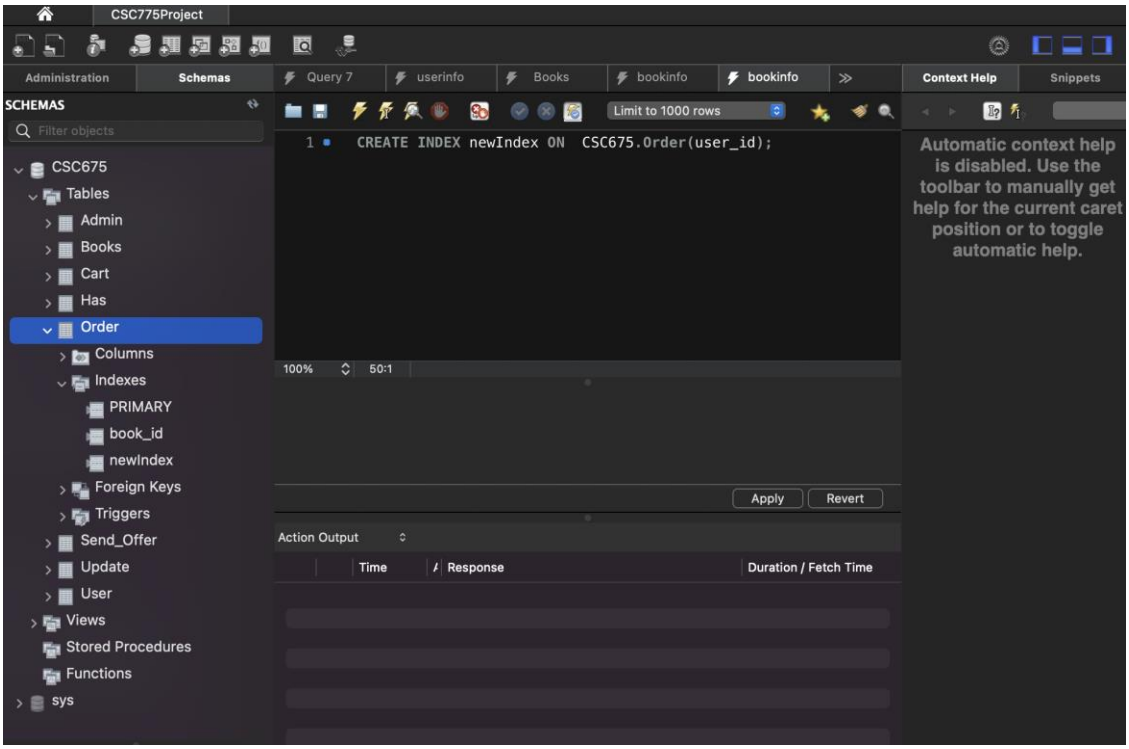
1. Create Tables, statements , constraints, indexes, views :

```
CREATE TABLE if not exists Books(
        ISBN VARCHAR(45) NOT NULL AUTO_INCREMENT,
        price FLOAT,
        title VARCHAR(32) NOT NULL,
        category VARCHAR(32),
        author VARCHAR(32),
        year YEAR(4)
        PRIMARY KEY (ISBN)
);

CREATE TABLE if not exists User (
        user_id INT,
        name VARCHAR(32) NOT NULL,
        address VARCHAR(32),
        email VARCHAR(32),
        phone VARCHAR(32)
        PRIMARY KEY (user_id)
);

CREATE TABLE Cart (
        cartId INT NOT NULL,
        total_price INT NOT NULL,
        quantity INT NOT NULL,
        PRIMARY KEY (cartId)
);

CREATE TABLE Has (
        user_id INT NOT NULL,
        cartId INT NOT NULL,
        PRIMARY KEY (userId, cartId),
        INDEX cartId_idx (cartId ASC) VISIBLE,
        FOREIGN KEY (userId) REFERENCES User (userId) ON DELETE CASCADE
        FOREIGN KEY (cartId) REFERENCES Cart (cartId) ON DELETE CASCADE
)
```

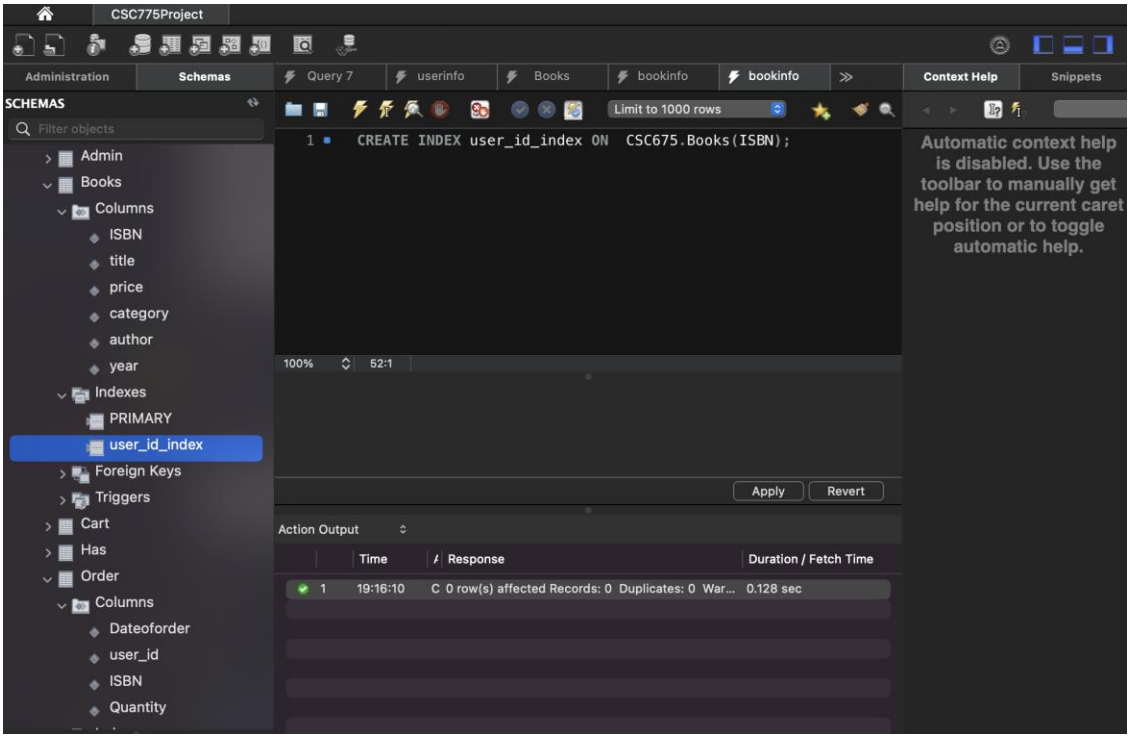
```
CREATE TABLE Admin (
        username VARCHAR(45) NOT NULL,
        password VARCHAR(45) NOT NULL,
        PRIMARY KEY (username)
);

CREATE TABLE Update (
        username VARCHAR(45) NOT NULL,
        ISBN INT NOT NULL,
        PRIMARY KEY (username, ISBN),
        FOREIGN KEY (username) REFERENCES Admin (username) ON DELETE
        CASCADE
);

CREATE TABLE Send_Offer (
        offerid INT NOT NULL,
        user_id INT NOT NULL,
        username VARCHAR(45) NOT NULL,
        PRIMARY KEY (offerid, user_id, username),
        CONSTRAINT `customername`
        FOREIGN KEY (`username`) REFERENCES `Admin` (`username`) ON
        DELETE CASCADE ON UPDATE CASCADE,
        CONSTRAINT `person_id`
        FOREIGN KEY (`user_id`) REFERENCES `User` (`user_id`) ON DELETE
        CASCADE ON UPDATE CASCADE
);

CREATE TABLE Order (
        Dateoforder date NOT NULL,
        user_id int NOT NULL,
        ISBN varchar(45) NOT NULL,
        Quantity int NOT NULL,
        PRIMARY KEY (`Dateoforder`,`user_id`,`ISBN`),
        CONSTRAINT `book_id`
        FOREIGN KEY (`ISBN`) REFERENCES `Books` (`ISBN`) ON DELETE
        CASCADE    ON UPDATE CASCADE,
        CONSTRAINT `customer_id`
        FOREIGN KEY (`user_id`) REFERENCES `User` (`user_id`) ON DELETE
        CASCADE ON UPDATE CASCADE)
```

Index:

CREATE INDEX newIndex ON Order(user_id);

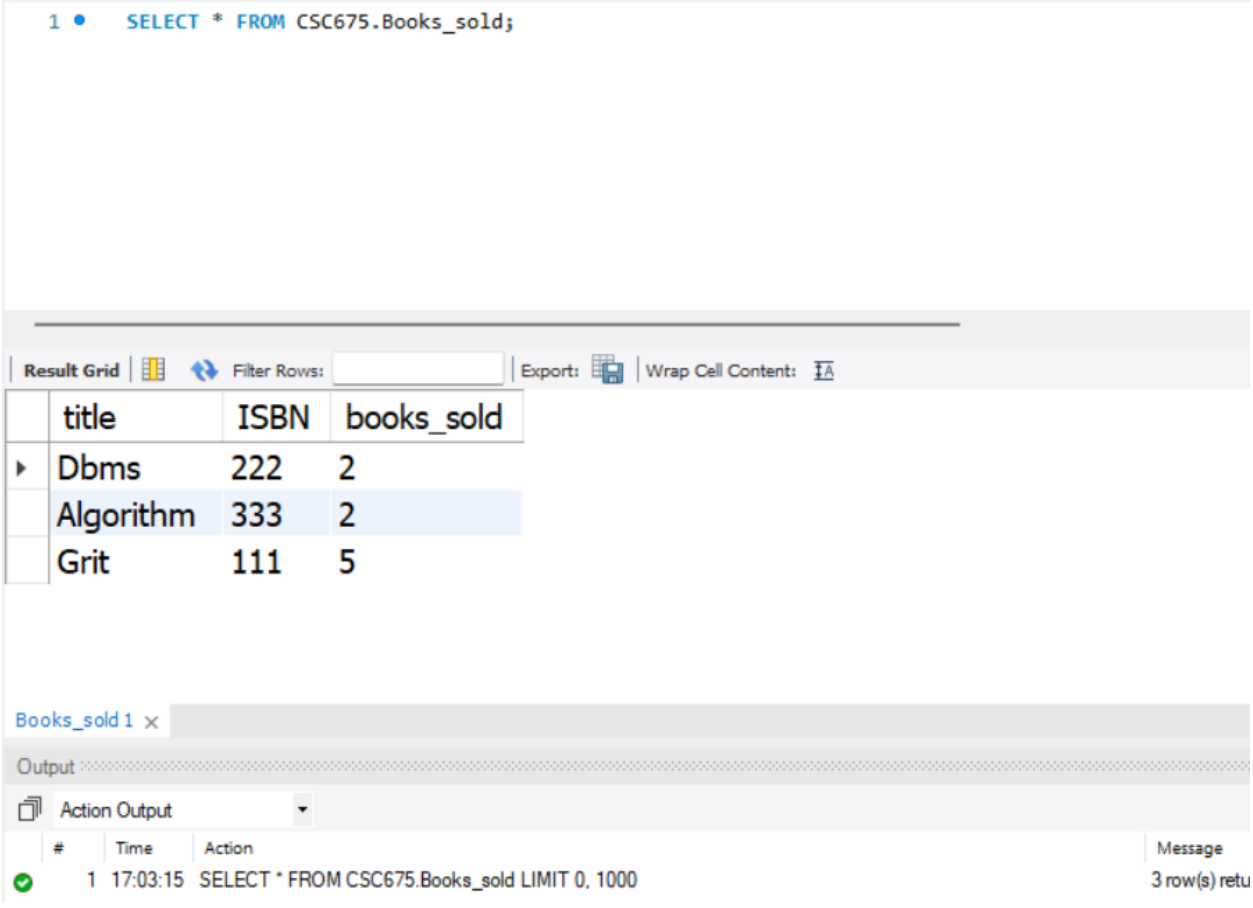


CREATE INDEX user_id_index ON CSC675.Books(ISBN);



Views:

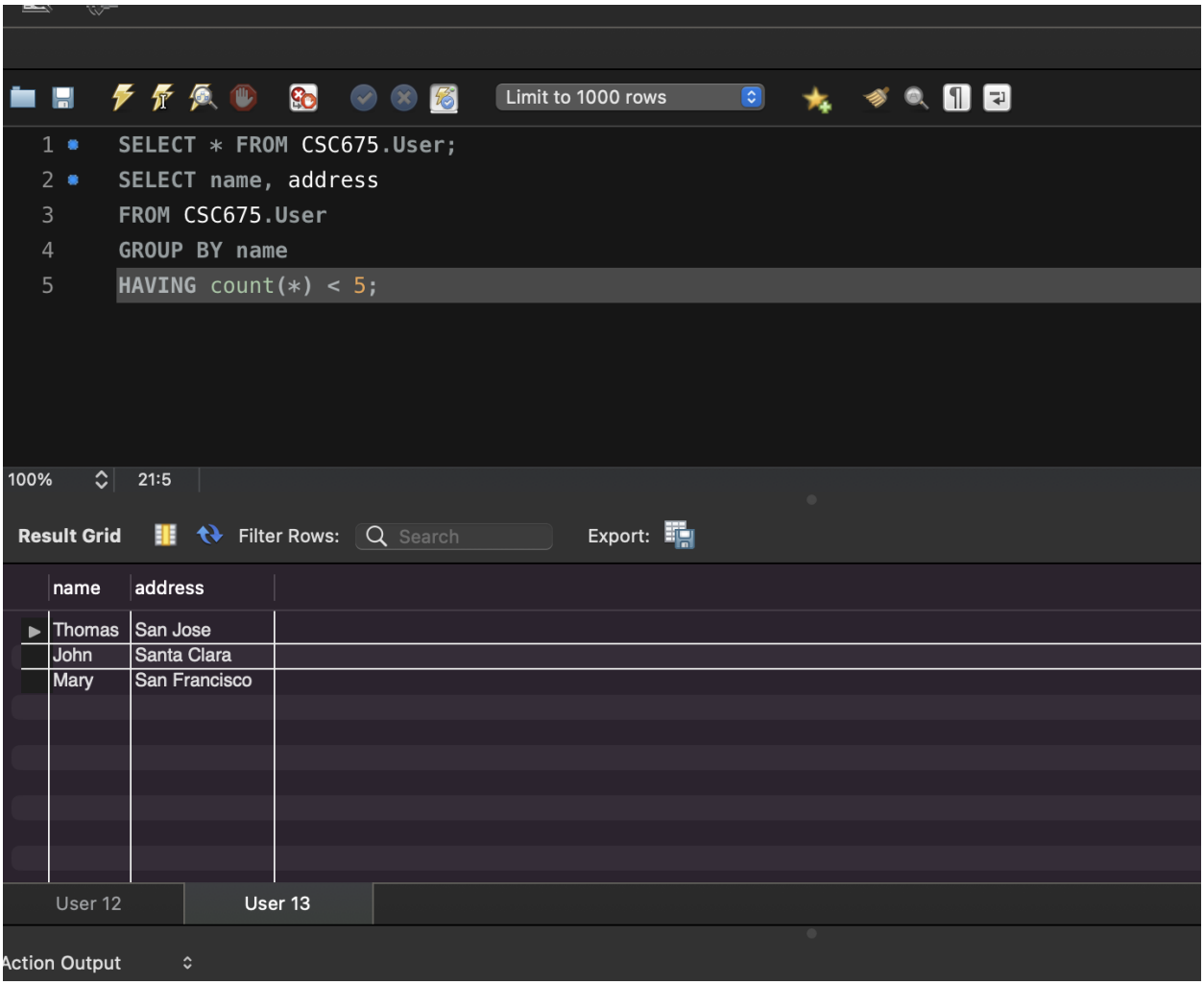
CREATE VIEW `CSC675`.`Books_sold` AS SELECT `CSC675`.`Books`.`title` AS `title`, `CSC675`.`Order`.`ISBN` AS `ISBN`, SUM(`CSC675`.`Order`.`Quantity`) AS `books_sold` FROM (`CSC675`.`Order` JOIN `CSC675`.`Books`) WHERE (`CSC675`.`Order`.`ISBN` = `CSC675`.`Books`.`ISBN`) GROUP BY `CSC675`.`Order`.`ISBN` ORDER BY `books_sold`	CREATE VIEW CSC675.userinfo AS SELECT email FROM CSC675.User, CSC675.Order WHERE CSC675.User.user_id = CSC675.Order.user_id;



2. Queries and Screenshots

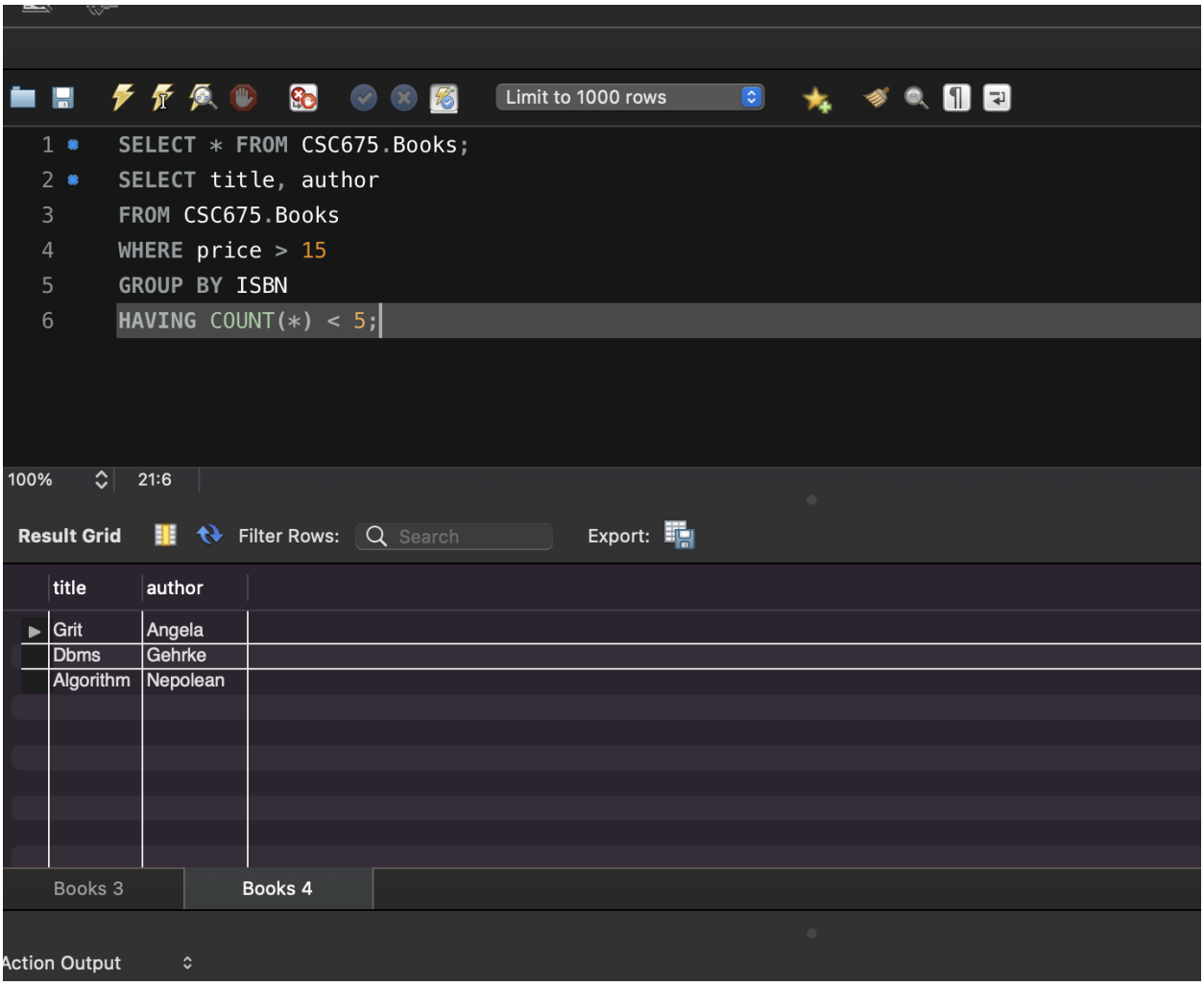
1. Display the name of all the users grouped by user id.

```
SELECT * FROM User;
SELECT name, address
FROM User
GROUP BY name
HAVING COUNT(*) < 5;
```



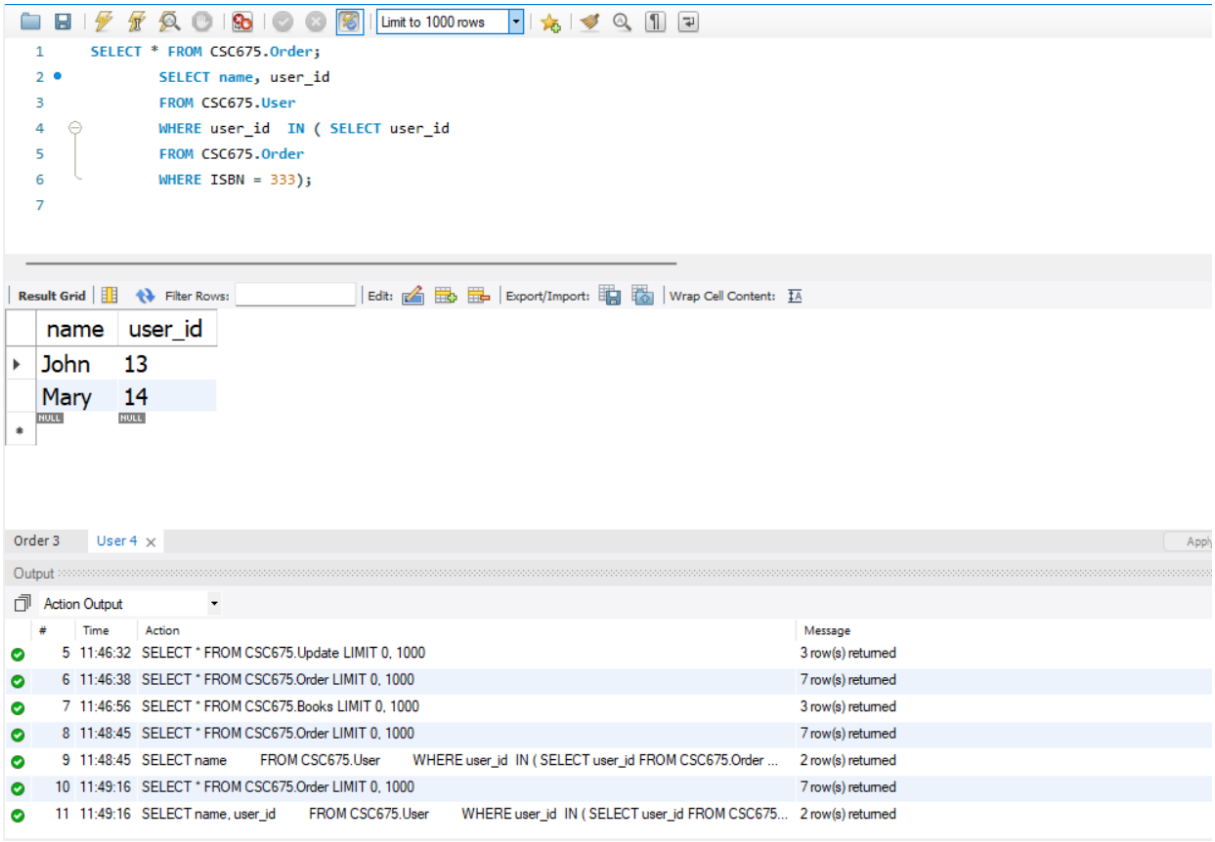
2. Display the title and author of all the books where the price of the books is greater than 15 and the count is less than 5.

```
SELECT * FROM Books;
SELECT title, author
FROM Books
WHERE price > 15
GROUP BY ISBN
HAVING COUNT(*) < 5;
```



3. Display names of users who ordered books having ISBN of 333.

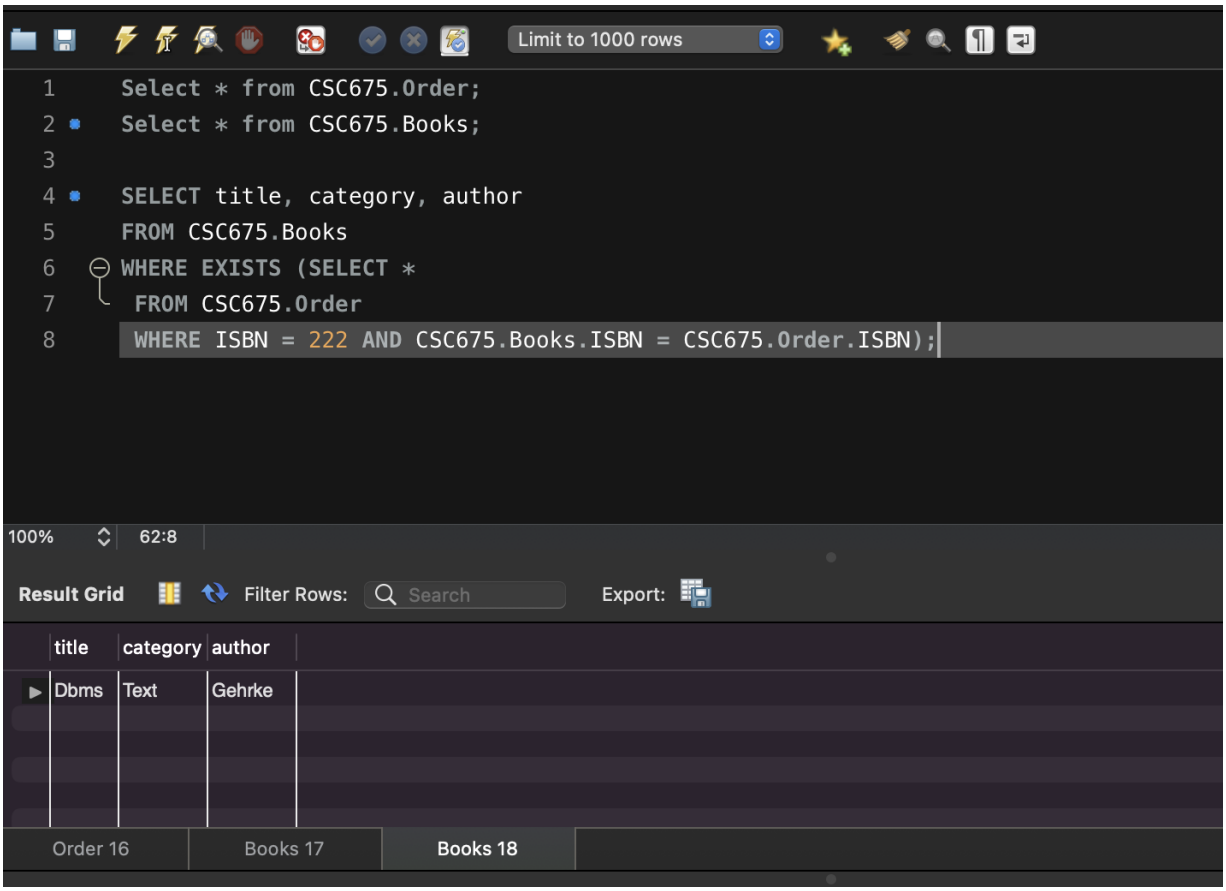
```
SELECT * FROM Order;
SELECT name, user_id
FROM User
WHERE user_id  IN ( SELECT user_id
        FROM Order
        WHERE ISBN = 333);
```



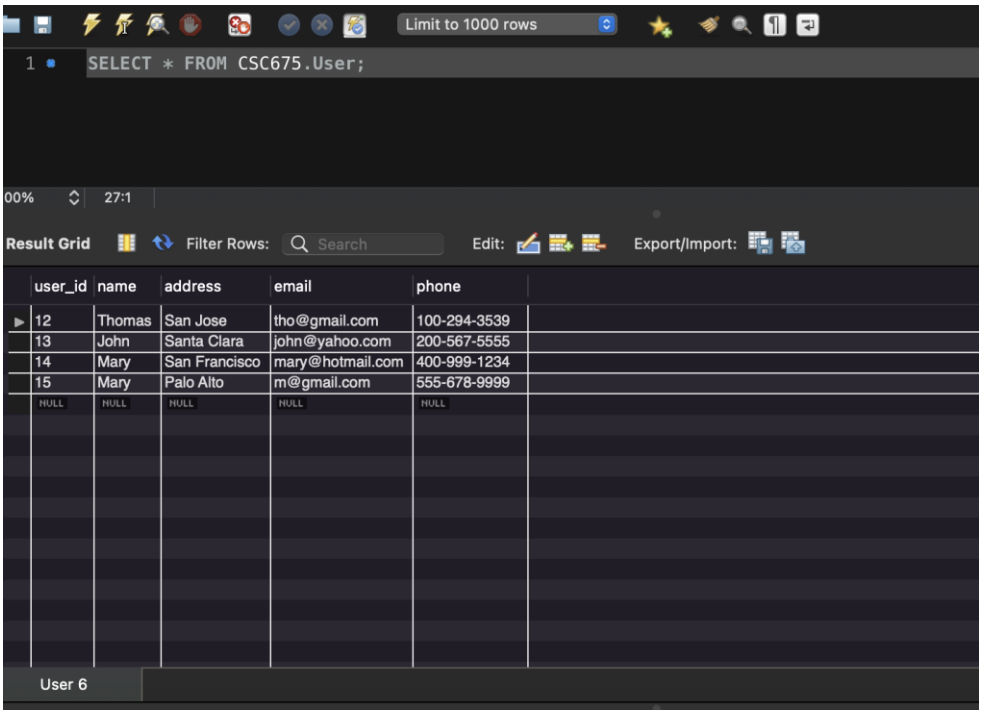
4. Display title, category and author of books that were ordered and have

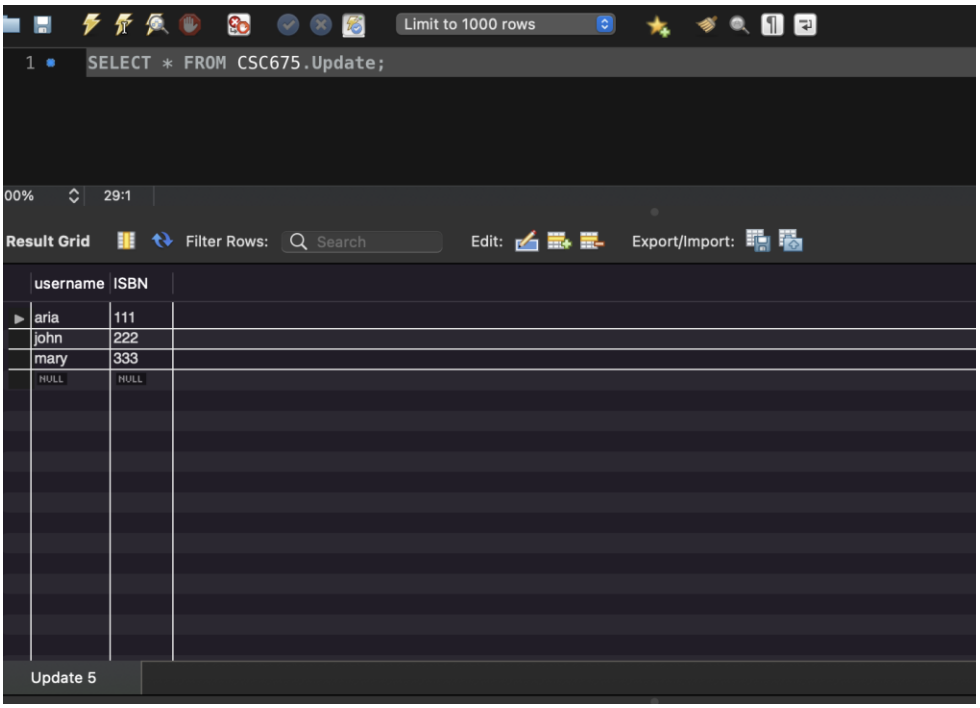
ISBN as 222.

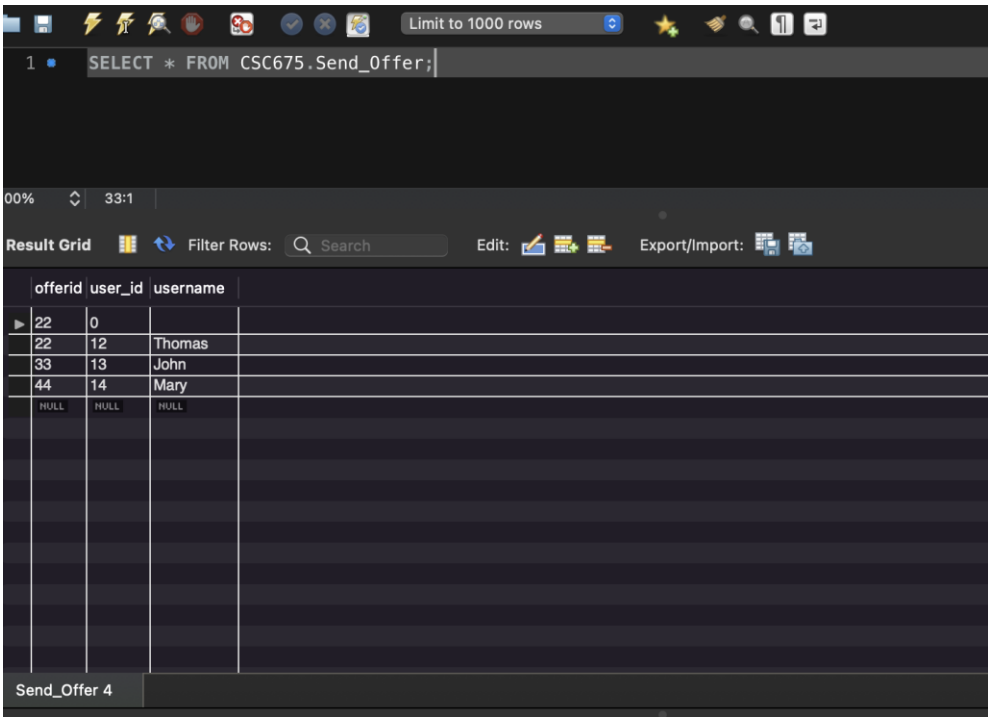
```
SELECT title, category, author
FROM Books
WHERE EXISTS ( SELECT *
FROM Order
WHERE ISBN = 222 AND Books.ISBN = Order.ISBN);
```

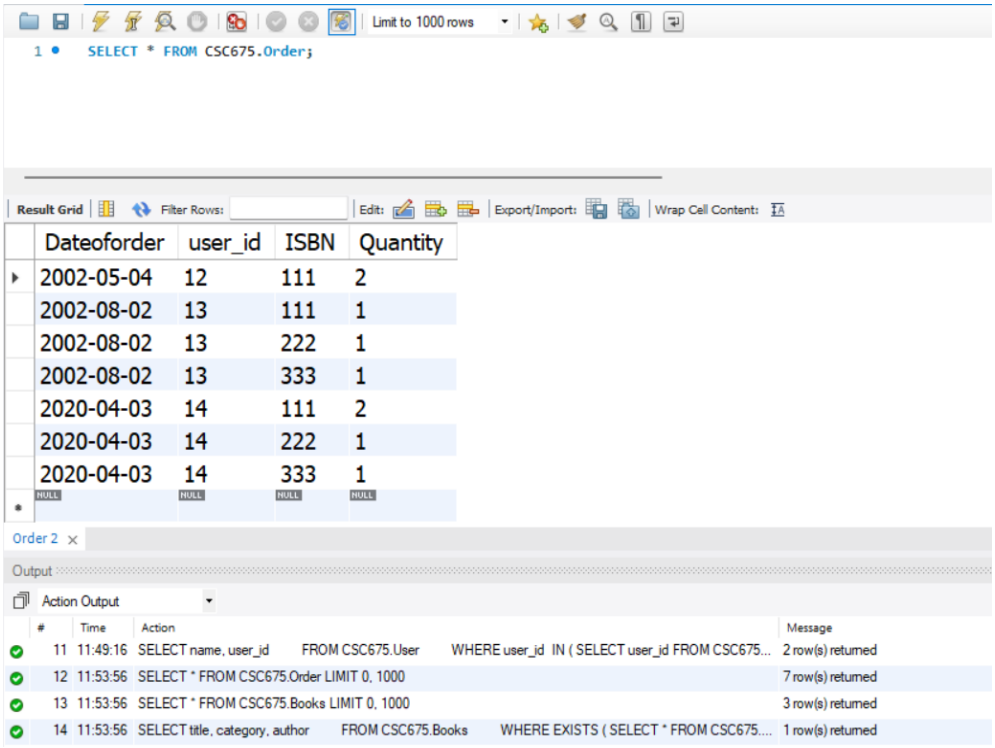


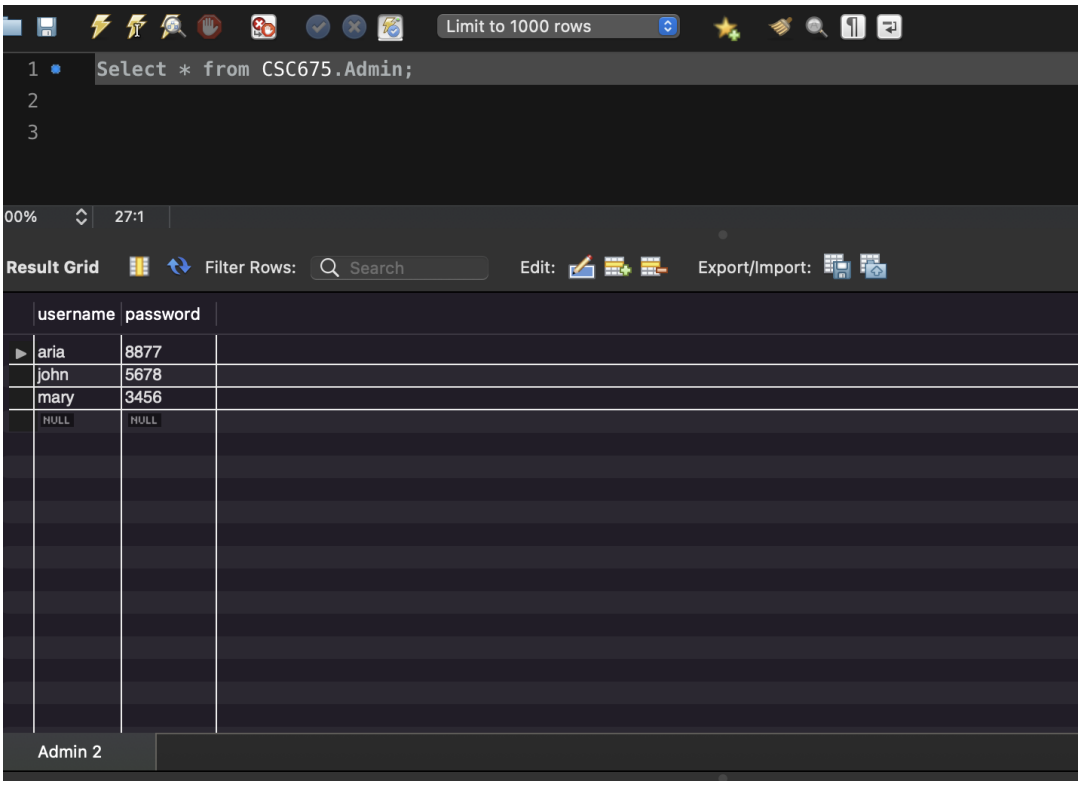
3. Tables Screenshots: All the select queries

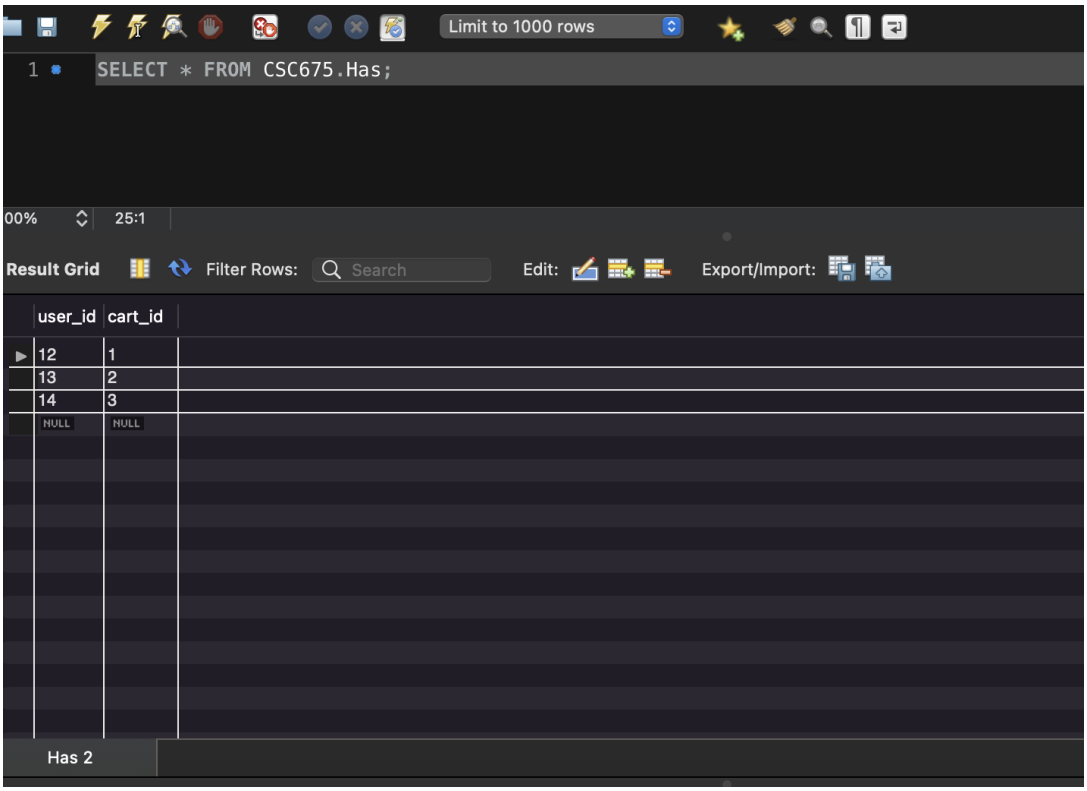

Limit to 1000 rows
1 SELECT * FROM CSC675.User;
00% 27:1
Result Grid Filter Rows: Search Edit: Export/Import:
user_id name address email phone
12 Thomas San Jose tho@gmail.com 100-294-3539
13 John Santa Clara john@yahoo.com 200-567-5555
14 Mary San Francisco mary@hotmail.com 400-999-1234
15 Mary Palo Alto m@gmail.com 555-678-9999
NULL NULL NULL NULL NULL
User 6

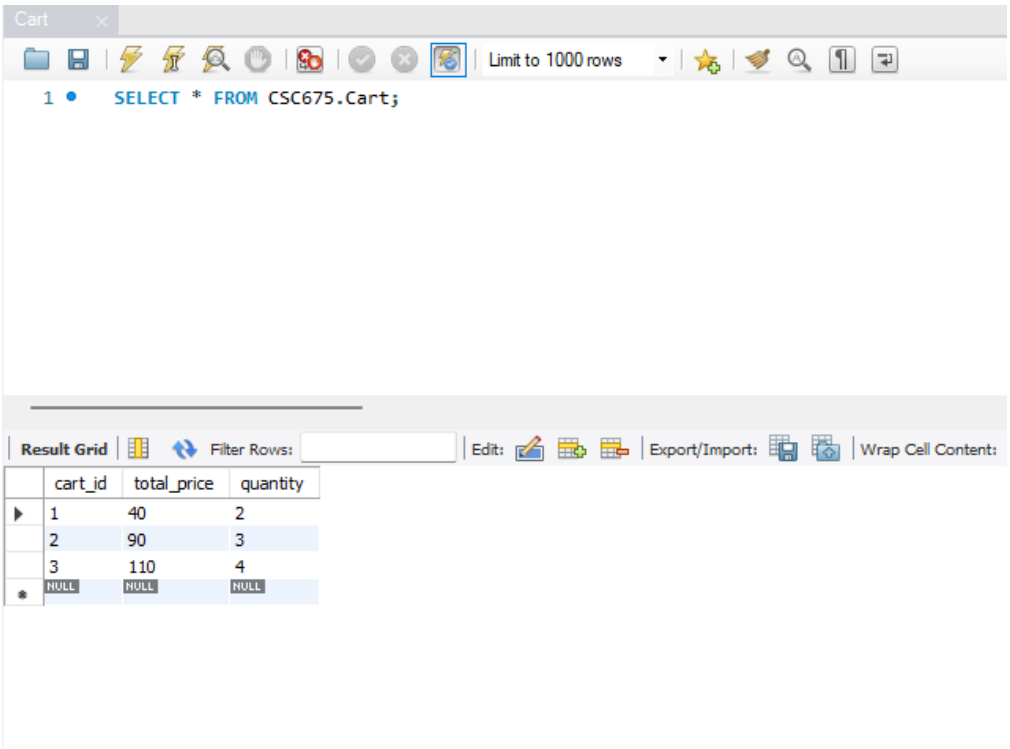

Limit to 1000 rows
1 SELECT * FROM CSC675.Update;
00% 29:1
Result Grid Filter Rows: Search Edit: Export/Import:
username ISBN
aria 111
john 222
mary 333
NULL NULL
Update 5

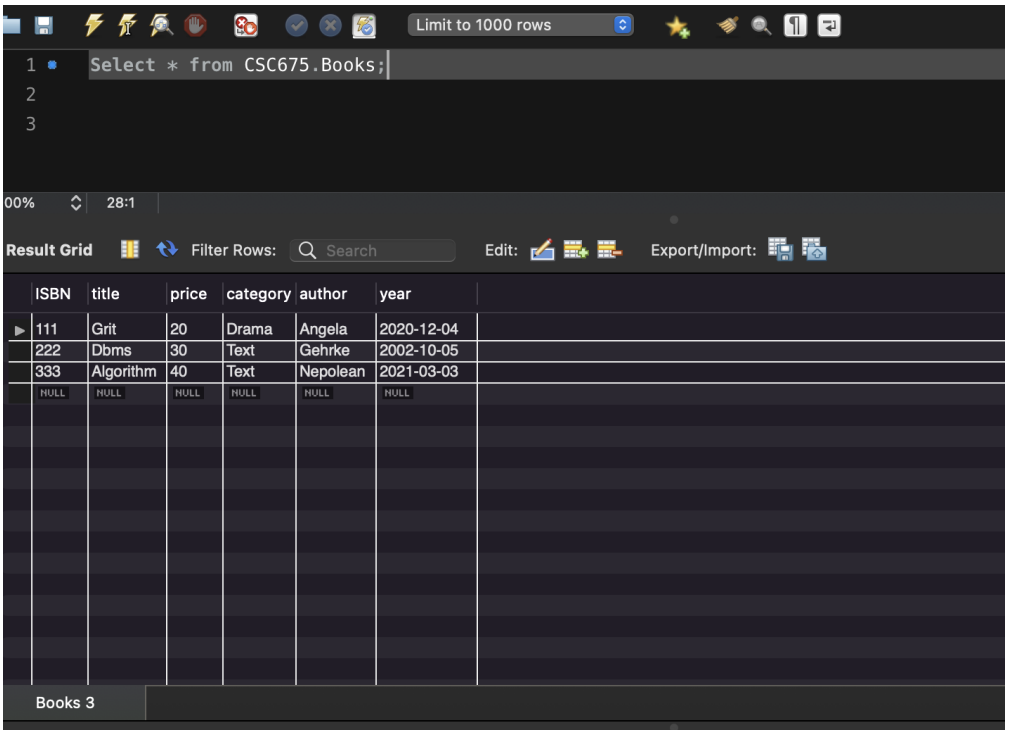

Limit to 1000 rows
1 SELECT * FROM CSC675.Send_Offer;
00% 33:1
Result Grid Filter Rows: Search Edit: Export/Import:
offerid user_id username
22 0
22 12 Thomas
33 13 John
44 14 Mary
NULL NULL NULL
Send_Offer 4


Limit to 1000 rows
1 SELECT * FROM CSC675.Order;
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
Dateoforder user_id ISBN Quantity
2002-05-04 12 111 2
2002-08-02 13 111 1
2002-08-02 13 222 1
2002-08-02 13 333 1
2020-04-03 14 111 2
2020-04-03 14 222 1
2020-04-03 14 333 1
Order 2
Output
Action Output
Time Action Message
11 11:49:16 SELECT name, user_id FROM CSC675.User WHERE user_id IN (SELECT user_id FROM CSC675... 2 row(s) returned
12 11:53:56 SELECT * FROM CSC675.Order LIMIT 0, 1000 7 row(s) returned
13 11:53:56 SELECT * FROM CSC675.Books LIMIT 0, 1000 3 row(s) returned
14 11:53:56 SELECT title, category, author FROM CSC675.Books WHERE EXISTS (SELECT * FROM CSC675.... 1 row(s) returned


Limit to 1000 rows
1 Select * from CSC675.Admin;
2
3
00% 27:1
Result Grid Filter Rows: Search Edit: Export/Import:
username password
aria 8877
john 5678
mary 3456
NULL NULL
Admin 2


Limit to 1000 rows
1 SELECT * FROM CSC675.Has;
00% 25:1
Result Grid Filter Rows: Search Edit: Export/Import:
user_id cart_id
12 1
13 2
14 3
NULL NULL
Has 2


Cart
Limit to 1000 rows
1 SELECT * FROM CSC675.Cart;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
cart_id total_price quantity
1 40 2
2 90 3
3 110 4
NULL NULL NULL


Limit to 1000 rows
1 Select * from CSC675.Books;
2
3
00% 28:1
Result Grid
Filter Rows:
Search
Edit:
Export/Import:
ISBN title price category author year
111 Grit 20 Drama Angela 2020-12-04
222 Dbms 30 Text Gehrke 2002-10-05
333 Algorithm 40 Text Nepolean 2021-03-03
NULL NULL NULL NULL NULL NULL
Books 3