

## LAB04: View and Subquery

---

### Submission:

- Submit a lab file named “int205\_**lab04**\_xxxxxxxxxxx.docx/.pdf” into the LEB2 system. xxxxxxxxxxxx = your student id

### Due Date & Time:

- Lecturer will inform the **LAB04** due date and time in lab class.
- 

### The View WITH CHECK OPTION Clause

The WITH CHECK OPTION is an integrity constraint on **an updatable view** to prevent inserts to rows for which the WHERE clause in the select\_statement is not true. WITH CHECK OPTION testing is standard-compliant.

### Syntax for creating a view

```
CREATE [OR REPLACE]
    VIEW view_name [(column_list)]
    AS select_statement
[WITH [CASCADED | LOCAL] CHECK OPTION];
```

### The Syntax of CREATE VIEW statement:

Documentation: <https://dev.mysql.com/doc/refman/8.0/en/create-view.html>

The WITH CHECK OPTION clause can be given for an updatable view to prevent inserts or updates to rows except those for which the WHERE clause in the select\_statement is true.

In a WITH CHECK OPTION clause for an updatable view, the LOCAL and CASCADED keywords determine the scope of check testing when the view is defined in terms of another view. The LOCAL keyword restricts the CHECK OPTION only to the view being defined. CASCADED causes the checks for underlying views to be evaluated as well. When neither keyword is given, the default is CASCADED.

Resource: <https://www.mysqltutorial.org/mysql-view-local-cascaded-in-with-check-option>

### Example:

```
CREATE TABLE t1 (a INT);
```

```
CREATE OR REPLACE VIEW v1
AS SELECT *
   FROM t1
  WHERE a < 2;
```

```
CREATE OR REPLACE VIEW v2
AS SELECT *
  FROM v1 WHERE a > 1
  WITH LOCAL CHECK OPTION;
```

```
CREATE OR REPLACE VIEW v3
AS SELECT *
  FROM v1
  WHERE a > 0
  WITH CASCADED CHECK OPTION;
```

**Evaluate the following INSERT statements:**

-- 1. What is the result?

```
INSERT INTO v2 VALUES (1);
```

-- The "CHECK OPTION failed" error is returned because the "a > 1" WHERE condition of V2 is False.

-- 2. What is the result?

```
INSERT INTO v2 VALUES (3);
```

-- The INSERT statement is executed successfully because the "a > 1" WHERE condition of V2 is True.

-- 3. What is the result?

```
INSERT INTO v3 VALUES (1);
```

-- The INSERT statement is executed successfully because both the "a > 0" WHERE condition of V3 and the "a < 2" WHERE condition of V1 are True.

-- 4. What is the result?

```
INSERT INTO v3 VALUES (3);
```

-- The "CHECK OPTION failed" error is returned because only the "a > 0" WHERE condition of V3 is True while the "a < 2" WHERE condition of V1 is False.

**Subquery Review:**

A subquery is a SELECT statement within another statement.

- Type 1 – **Nested Subquery**: Database evaluates the whole query in two steps:
  - First, execute the subquery (inner query).
  - Second, use the result of the subquery in the parent statement (outer query).
- Type 2 - **Correlated Subquery**: Database evaluated once for each row processed by the parent statement.
  - This operation is used when a subquery refers to a column from a table in an outer query.
  - The unqualified columns in the subquery are resolved by looking in the tables named in the inner query and then in the tables named in the outer query.

Subquery Documentation: <https://dev.mysql.com/doc/refman/8.0/en/subqueries.html>

## Subquery in DML statements

- INSERT statement – adds new rows of data to a table
- UPDATE statement – modifies existing data in a table
- DELETE statement – removes rows of data from a table

-- Syntax --

```
INSERT INTO table_name|view_name [(column_list)]
SELECT column(s)
FROM table_name| view_name
[WHERE condition(s)];
```

```
UPDATE table_name|view_name
SET column = value [,column2 = value2,...]
[WHERE condition(s)];
```

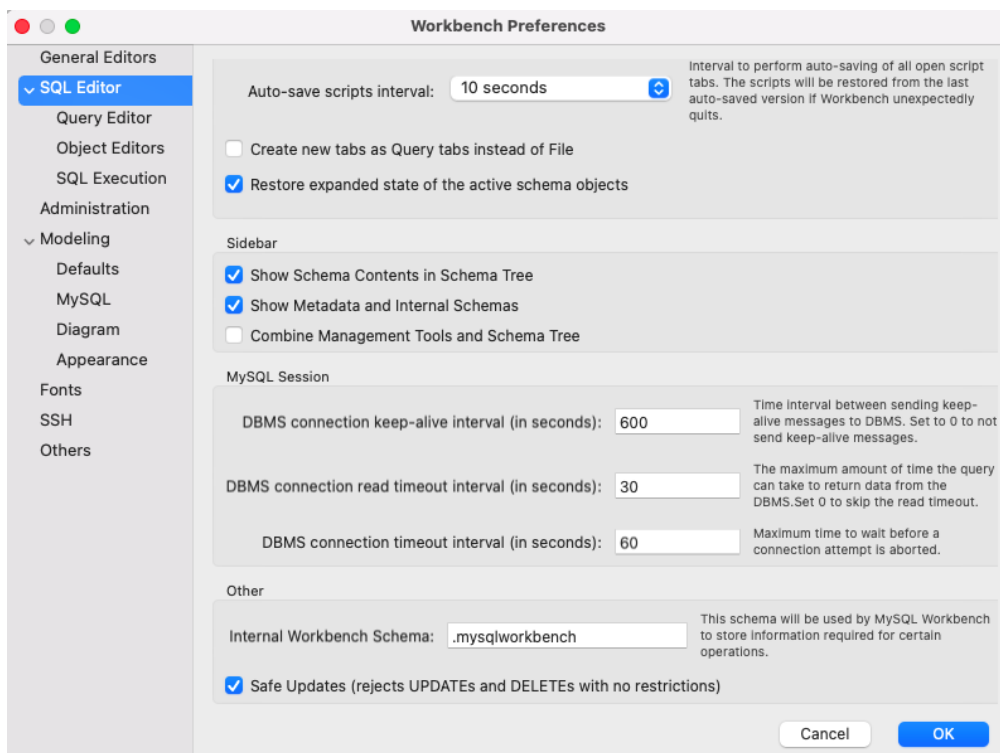
```
DELETE table_name|view_name
[WHERE condition(s)];
```

## SAFE-UPDATES option

MySQL session has the safe-updates option set (SET SQL\_SAFE\_UPDATES = 1). This means that you can't update or delete records without specifying a key (ex. primary key) in the WHERE clause. If you want to disable the safe-updates option, you can set SET SQL\_SAFE\_UPDATES = 0.

MySQL Workbench: Checking the safe-updates option

Menu => Tools/MySQLWorkbench => Preferences => SQL Editor => Safe-updates



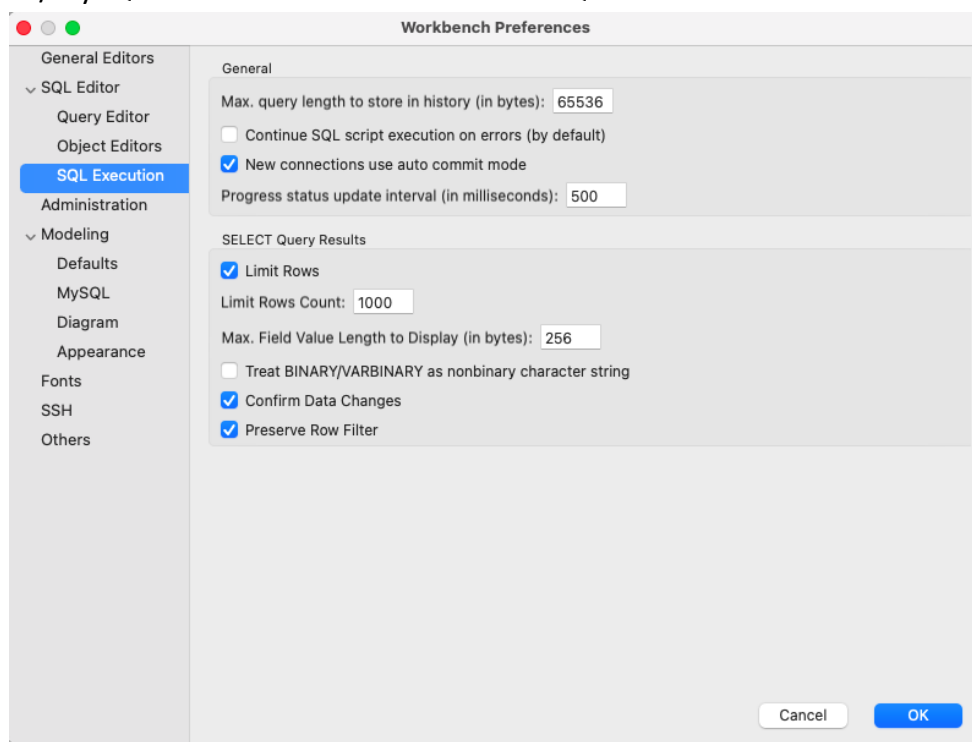
## AUTOCOMMIT Mode

By default, MySQL starts the session for each new connection [with autocommit enabled](#), so MySQL does a commit after each SQL statement if that statement did not return an error.

If **autocommit** mode is disabled within a session with `SET autocommit = 0`, the session always has a transaction open. A `COMMIT` or `ROLLBACK` statement ends the current transaction and a new one starts. If a session that has autocommit disabled ends without explicitly committing the final transaction, MySQL rolls back that transaction.

## MySQL Workbench: Checking the autocommit mode

Menu => Tools/MySQLWorkbench => Preferences => SQL Execution



## Switch to SQL Editor

- You should specify the classicmodels database before writing SQL statements using the following command:  
`USE db_name;`

The `USE` statement tells MySQL to use the named database as the default (current) database for subsequent statements. This statement requires some privilege for the database or some object within it.

The example of the result screen: Please insert your student id in the first line of the query section

The screenshot shows the MySQL Workbench interface with the classicmodels database selected. The query editor (Query 1) contains the following SQL code:

```

1 -- 6419999999
2 • use classicmodels;
3 -- 1. List customer name of all customers whose name starts with the letter 'S'.
4 • select customername
5   from customers
6  where customername like 'S%';

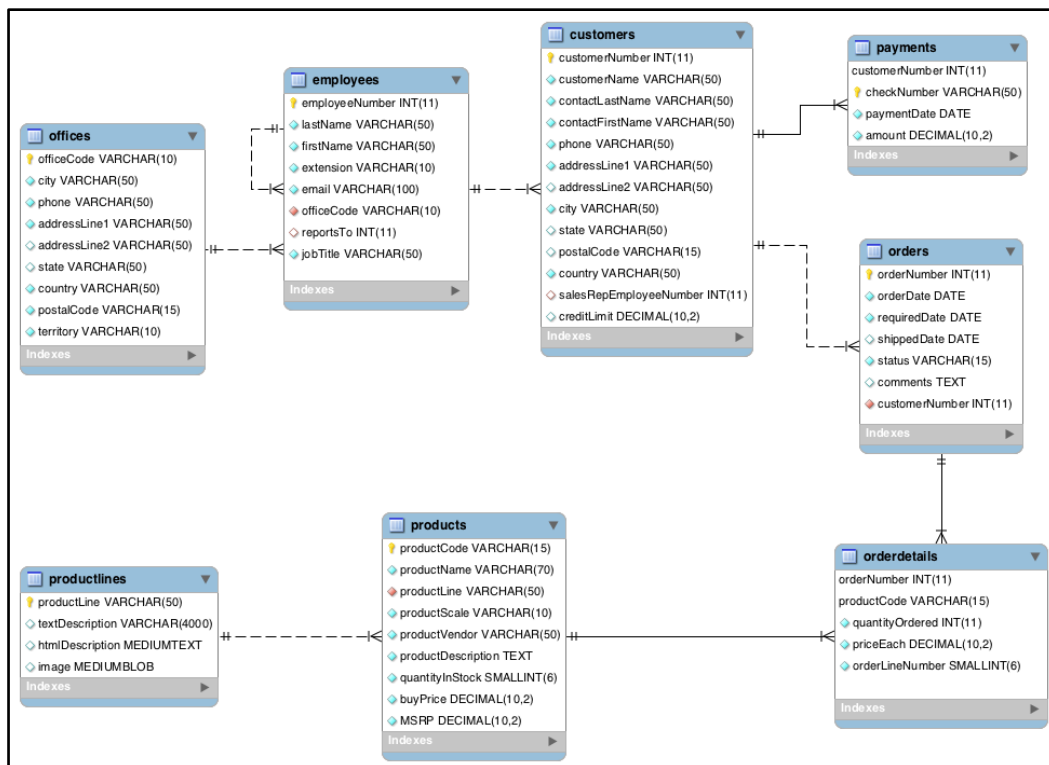
```

The Results Grid shows the following customer names:

| customername                   |
|--------------------------------|
| Signal Gift Stores             |
| Saveley & Henriot, Co.         |
| Souvenirs And Things Co.       |
| Schuyler Imports               |
| Stylish Desk Decors, Co.       |
| Suominen Souvenirs             |
| SAR Distributors, Co           |
| Salzburg Collectables          |
| Stuttgart Collectable Exchange |
| Scandinavian Gift Ideas        |
| Super Scale Inc.               |

The Action Output section shows the execution of the query, returning 12 row(s).

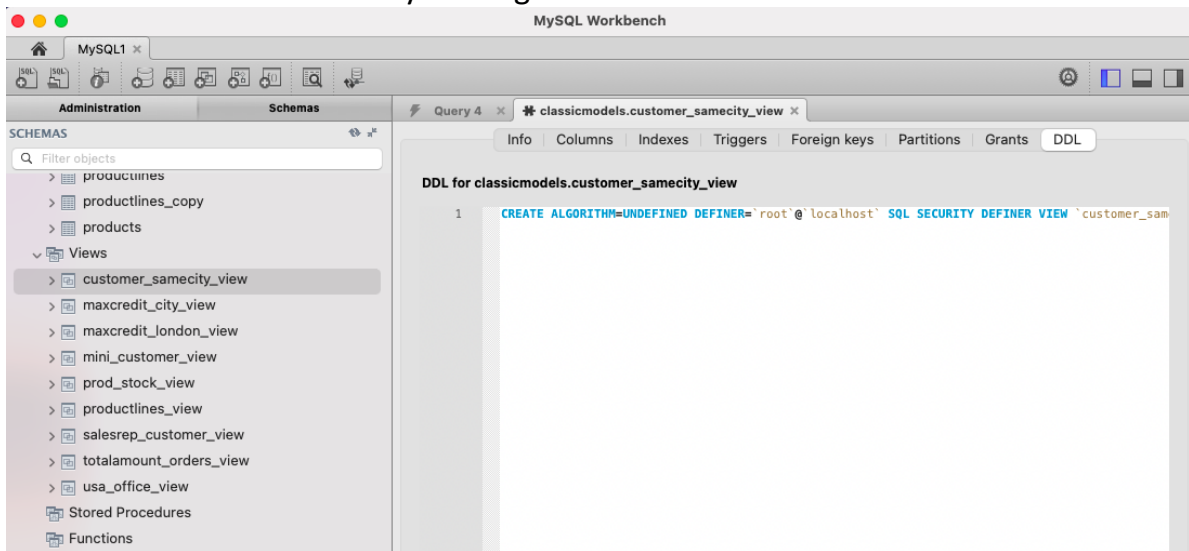
The ER diagram for the classicmodels.



Note: The MSRP is “Manufacturer's suggested retail price” (ราคาขายปลีกแนะนำของผู้ผลิต).

## MySQL Workbench:

- You can see details of a view by clicking i button below:

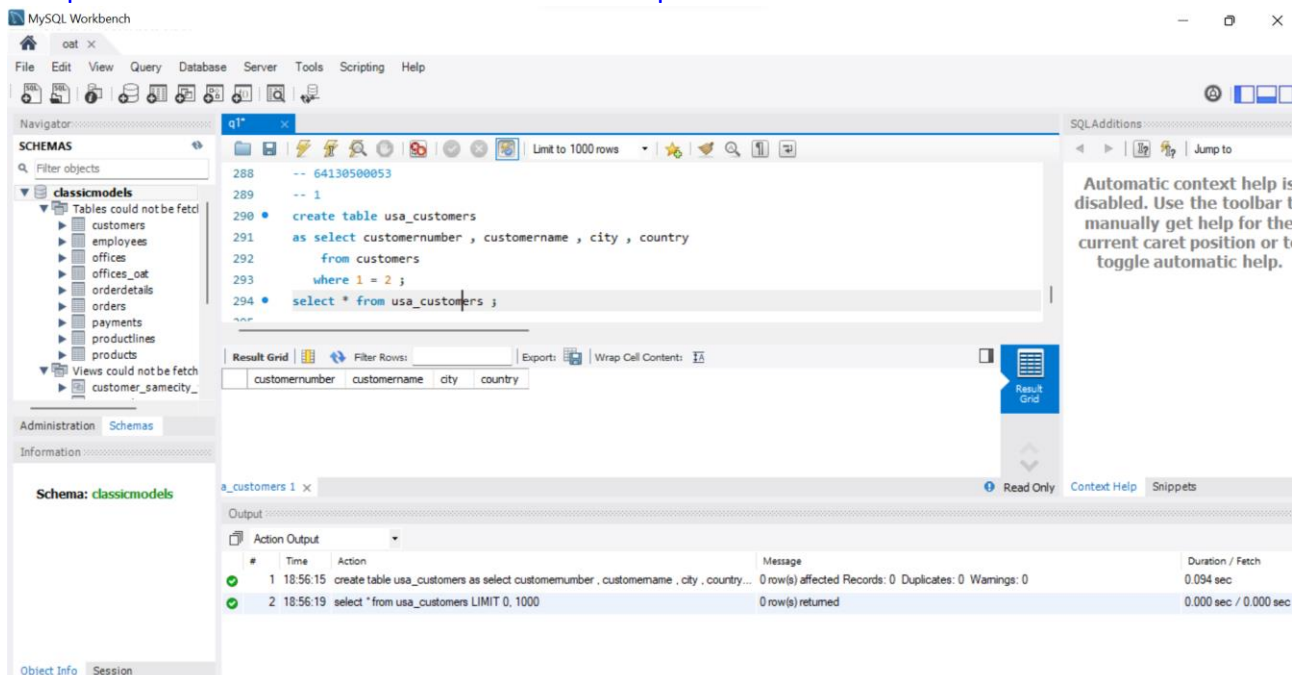


## Task 1: Using the "classicmodels" database and write SQL statements to answer the following questions.

use classicmodels;

1. Create a new table named "usa\_customers" with copying only the structure of four columns: customernumber, customername, city, country of the "customers" table. Do not copy any data from the "customers" table. Please verify by querying data from the table.

-- Capture the SQL statement + Result Screen and place here



2. Insert data by copying the existing data of all customers who live in the USA from the "customers" table into the "usa\_customers" table. Please verify by querying data from the table. How many rows are inserted into the "usa\_customers" table.



MySQL Workbench interface showing a query execution. The query is as follows:

```

296 -- 64130500053
297 -- 2
298 insert into usa_customers select customernumber,customername,city,country
299 from customers
300 where country = 'USA' ;
301 select count(*) from usa_customers ;
302 commit ;
303

```

The result grid shows the following output:

| count(*) |
|----------|
| 36       |

The output pane shows the following messages:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 26 | 19:23:24 | insert into usa_customers select customernumber,customername,city,country from c... | 36 row(s) affected Records: 36 Duplicates: 0 Warnings: 0 | 0.000 sec             |
| 27 | 19:23:44 | select count(*) from usa_customers LIMIT 0, 1000                                    | 1 row(s) returned  | 0.016 sec / 0.000 sec |

3. Based on the "usa\_customers" table, modify the city of the customername "Mini Wheels Co." to the same city of the customer number 344 of the "customers" table. Please verify your data modification.

MySQL Workbench interface showing a query execution. The query is as follows:

```

305 -- 64130500053
306 -- 3
307 SET SQL_SAFE_UPDATES = 0;
308 update usa_customers u
309 set u.city = (select c.city from customers c
310 where c.customernumber = 344)
311 where u.customername = 'Mini Wheels Co.' ;
312

```

The result grid shows the following output:

| customernumber | customername                 | city       | country |
|----------------|------------------------------|------------|---------|
| 112            | Signal Gift Stores           | Las Vegas  | USA     |
| 124            | Mini Gifts Distributors Ltd. | San Rafael | USA     |
| 129            | Mini Wheels Co.              | Madrid     | USA     |
| 131            | Land of Toys Inc.            | NYC        | USA     |
| 151            | Muscle Machine Inc.          | NYC        | USA     |

The output pane shows the following messages:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 8  | 19:35:58 | SET SQL_SAFE_UPDATES = 0  | 0 row(s) affected  | 0.000 sec             |
| 9  | 19:36:03 | update usa_customers u set u.city = (select c.city from customers c where c custom... | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 | 0.000 sec             |
| 10 | 19:36:47 | select * from usa_customers LIMIT 0, 1000   | 36 row(s) returned                                       | 0.000 sec / 0.000 sec |

4. Based on the "usa\_customers" table, modify the city of all customers who have a sales representative (employee) last named "Patterson" to "Bangmod". Please verify your data modification.

Hint: you may use the customers and employees tables to find out "who have a sales representative (employee) last named "Patterson".

MySQL Workbench interface showing a query execution. The query is:

```

313
314 -- 64130500053
315 -- 4
316 • update usa_customers u
317   set u.city = 'Bangmod'
318   where u.customernumber in (select customernumber from customers c join employees e
319     on c.salesRepEmployeeNumber = e.employeeNumber where e.lastName='Patterson');

```

The result grid shows the following data:

| customernumber | customername               | city       | country |
|----------------|----------------------------|------------|---------|
| 151            | Muscle Machine Inc         | NYC        | USA     |
| 157            | Diecast Classics Inc.      | Bangmod    | USA     |
| 161            | Technics Stores Inc.       | Burlingame | USA     |
| 168            | American Souvenirs Inc     | New Haven  | USA     |
| 173            | Cambridge Collectables Co. | Cambridge  | USA     |

The output pane shows the following actions:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 10 | 19:36:47 | select * from usa_customers LIMIT 0, 1000   | 36 row(s) returned                                       | 0.000 sec / 0.000 sec |
| 11 | 19:41:18 | update usa_customers u set u.city = 'Bangmod' where u.customernumber in (select ... | 6 row(s) affected Rows matched: 6 Changed: 6 Warnings: 0 | 0.015 sec             |
| 12 | 19:41:31 | select * from usa_customers LIMIT 0, 1000   | 36 row(s) returned                                       | 0.000 sec / 0.000 sec |

5. Modify an existing view named "mini\_customer\_view" to display the customer number, customer name, city and country of all customers whose names start with the word "Mini" from the "usa\_customers" table. Name four columns of this view to "cno", "cname", "city" and "country", respectively. Please verify by querying data from this view.

MySQL Workbench interface showing the creation of a view. The query is:

```

327 -- 64130500053
328 -- 5
329 • create or replace view mini_customer_view(cno,cname,city,country)
330   as select customernumber,customername,city,country
331   from usa_customers
332   where customername like 'Mini%';
333 • select*from mini_customer_view ;
334

```

The result grid shows the following data:

| cno | cname                        | city         | country |
|-----|------------------------------|--------------|---------|
| 124 | Mini Gifts Distributors Ltd. | San Rafael   | USA     |
| 129 | Mini Wheels Co.              | Madrid       | USA     |
| 319 | Mini Classics                | White Plains | USA     |
| 320 | Mini Creations Ltd.          | New Bedford  | USA     |
| 124 | Mini Gifts Distributors Ltd. | San Rafael   | USA     |

The output pane shows the following actions:

| #  | Time     | Action   | Message            | Duration / Fetch      |
|----|----------|--|--------------------|-----------------------|
| 12 | 20:07:16 | select*from mini_customer_view LIMIT 0, 1000   | 0 row(s) returned  | 0.000 sec / 0.000 sec |
| 13 | 20:07:33 | create or replace view mini_customer_view(cno,cname,city,country) as select custo... | 0 row(s) affected  | 0.000 sec             |
| 14 | 20:07:37 | select*from mini_customer_view LIMIT 0, 1000   | 32 row(s) returned | 0.000 sec / 0.000 sec |

6. Create a view named "mini\_ltd\_customer\_view" to display the customer number, customer name, city and country of all customers whose names end with the word "Ltd." from the "mini\_customer\_view" view. Please ensure that the rows that are being changed through this view are conformable to the definition of the "mini\_ltd\_customer\_view" view. Name four columns of this view to "custno", "custname", "custcity" and "custcountry", respectively. Please verify by querying data from this view.



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

| custno | custname                     | custcity    | custcountry |
|--------|------------------------------|-------------|-------------|
| 124    | Mini Gifts Distributors Ltd. | San Rafael  | USA         |
| 320    | Mini Creations Ltd.          | New Bedford | USA         |
| 124    | Mini Gifts Distributors Ltd. | San Rafael  | USA         |
| 320    | Mini Creations Ltd.          | New Bedford | USA         |
| 124    | Mini Gifts Distributors Ltd. | San Rafael  | USA         |

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 14 | 20:07:37 | select from mini_customer_view LIMIT 0, 1000  | 32 row(s) returned | 0.000 sec / 0.000 sec |
| 15 | 20:08:38 | create or replace view mini_ltd_customer_view (custno, custname, custcity, custcountry) | 0 row(s) affected  | 0.000 sec             |
| 16 | 20:08:42 | select*from mini_ltd_customer_view LIMIT 0, 1000  | 16 row(s) returned | 0.000 sec / 0.000 sec |

7. Insert new data {customer number "9000", customer name "SUNISA Ltd.", city "Texas" and country "USA"} through the "mini\_ltd\_customer\_view" view. Please verify by querying data from both this view and the base table. Can the data be inserted through this view? If not, please explain.

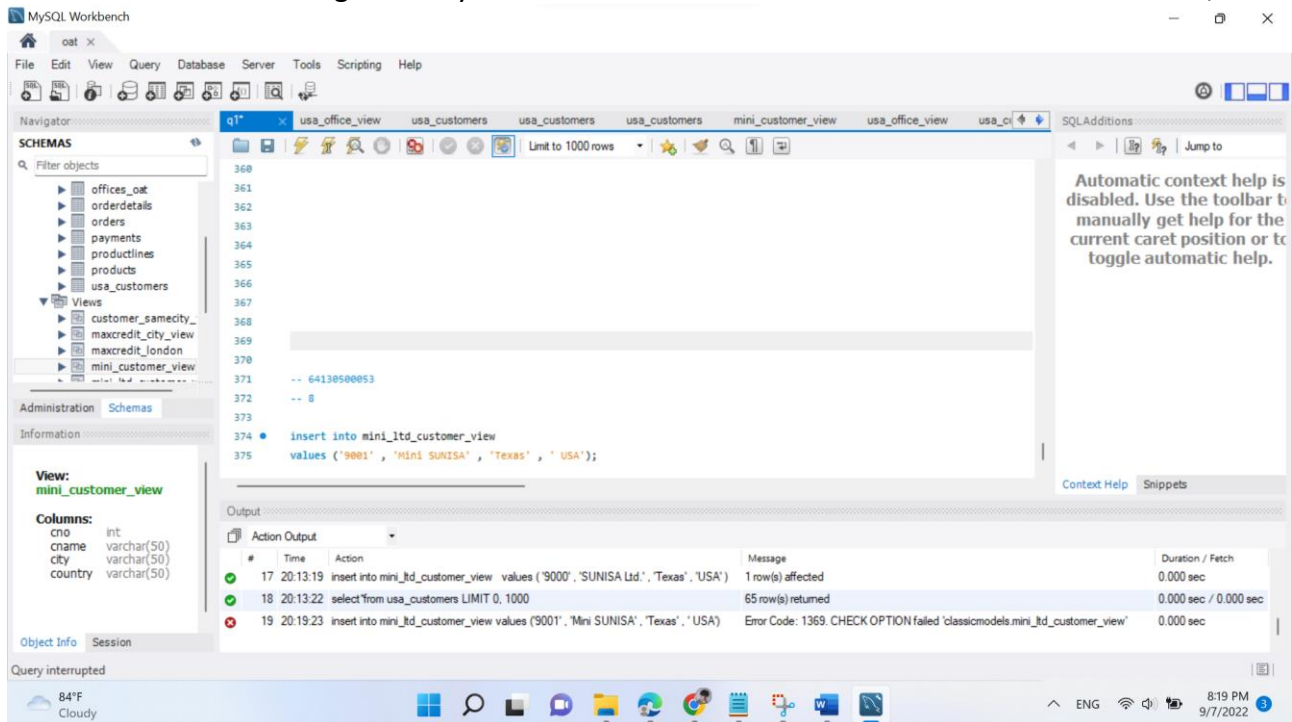
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

| customernumber | customername                 | city         | country |
|----------------|------------------------------|--------------|---------|
| 124            | Mini Gifts Distributors Ltd. | San Rafael   | USA     |
| 129            | Mini Wheels Co.              | Madrid       | USA     |
| 319            | Mini Classics                | White Plains | USA     |
| 320            | Mini Creations Ltd.          | New Bedford  | USA     |
| 9000           | SUNISA Ltd.                  | Texas        | USA     |

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 16 | 20:08:42 | select*from mini_ltd_customer_view LIMIT 0, 1000                                  | 16 row(s) returned | 0.000 sec / 0.000 sec |
| 17 | 20:13:19 | insert into mini_ltd_customer_view values ('9000', 'SUNISA Ltd.', 'Texas', 'USA') | 1 row(s) affected  | 0.000 sec             |
| 18 | 20:13:22 | select*from usa_customers LIMIT 0, 1000   | 65 row(s) returned | 0.000 sec / 0.000 sec |

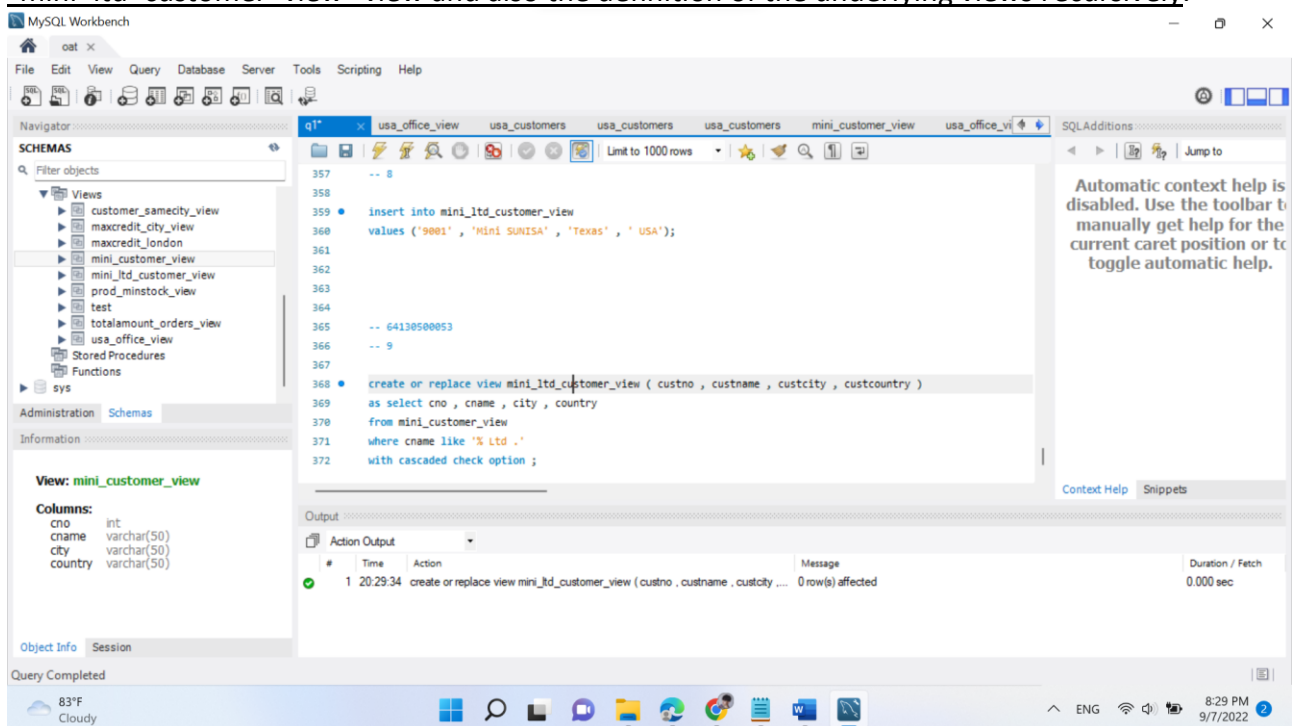
สามารถเพิ่มได้แต่จะไม่สามารถดูได้ใน view ของ mini\_ltd\_customers\_view เนื่องจากอ้างอิงมาจาก mini\_customers\_view ที่มีเงื่อนไขของ Like 'Mini%' แต่ยังสามารถเพิ่มลงไปได้เนื่องจาก Like '%Ltd.' และมี with local ที่เช็คการ insert เฉพาะ mini\_ltd นี้

8. Insert new data {customer number "9001", customer name "Mini SUNISA", city = "Texas" and country "USA"} through the "mini\_ltd\_customer\_view" view. Please verify by querying data from both this view and the base table. Can the data be inserted through this view? If not, please explain.



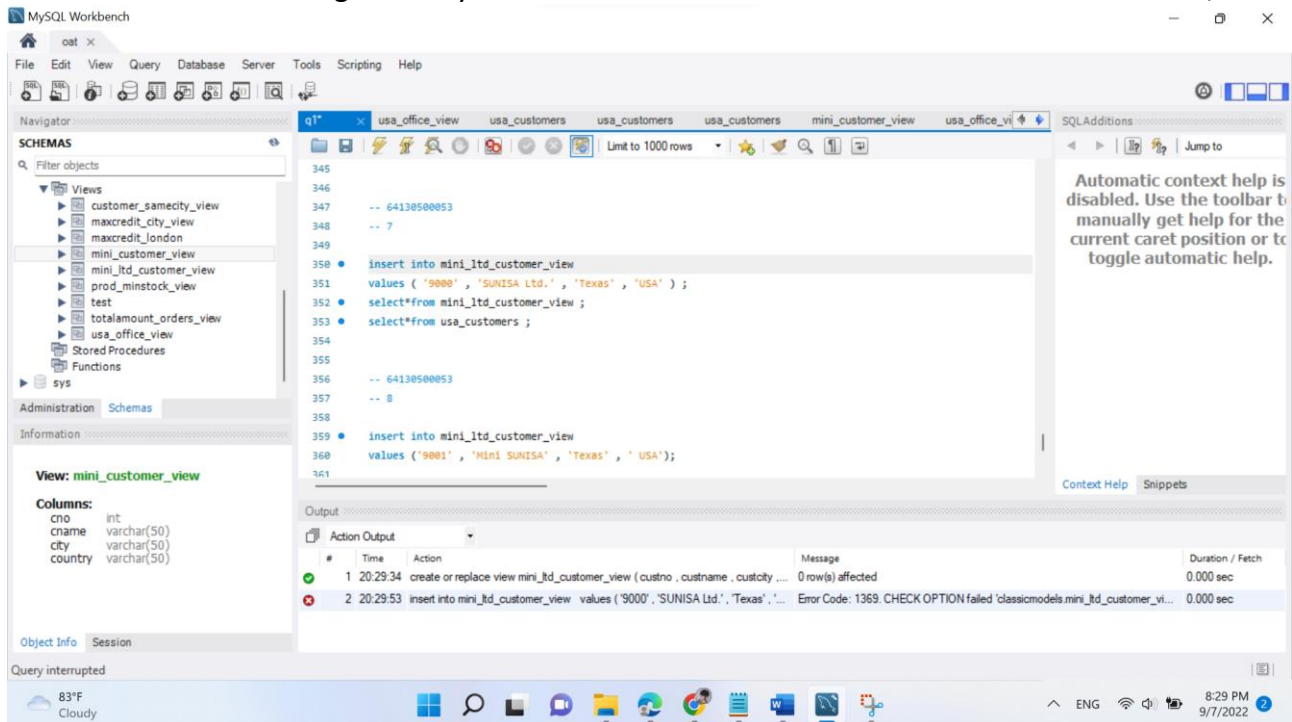
ไม่ผ่านเงื่อนไขของ with local check option ที่ถูกกำหนดไว้ว่าลงท้ายด้วย Ltd

9. Modify an existing view named the "mini\_ltd\_customer\_view" created in [Question 6](#) to ensure that the rows that are being changed through this view are conformable to the definition of the "mini\_ltd\_customer\_view" view and also the definition of the underlying views recursively.



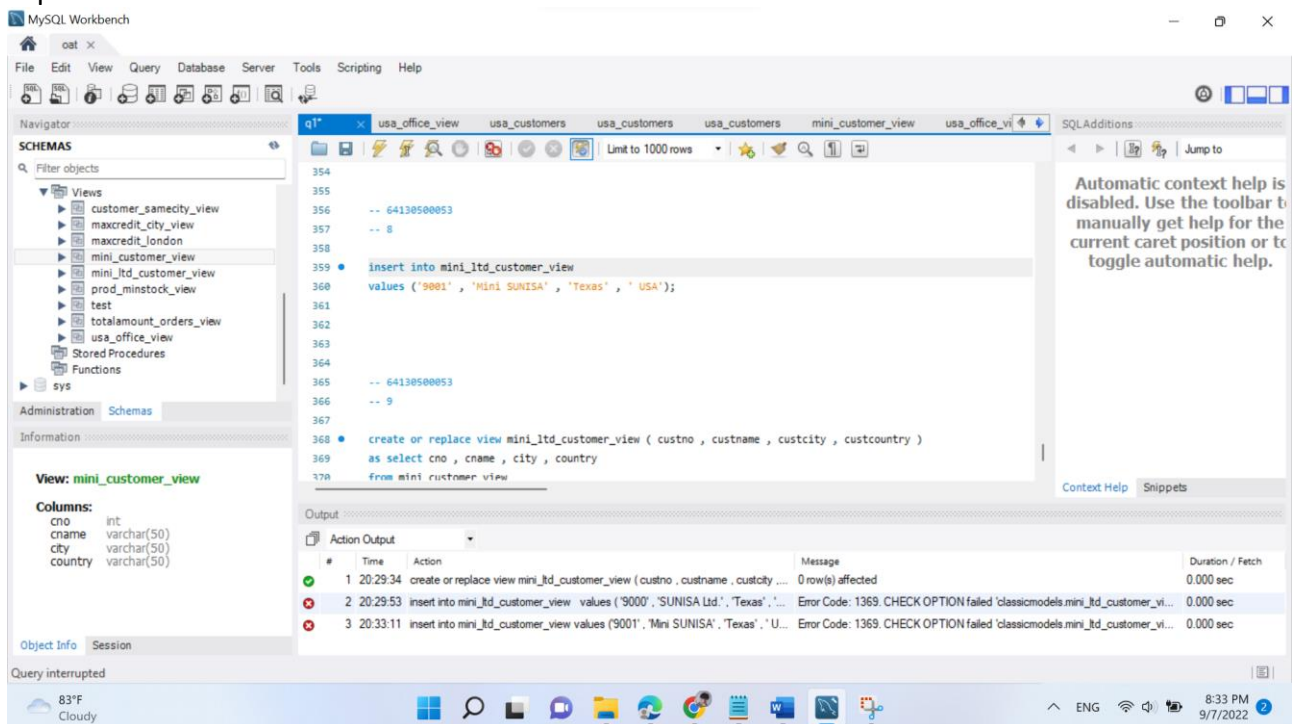
10. Try to insert the same data of [Question 7-8](#) again through the modified view in [Question 9](#).

What happened to the row of the customer name "SUNISA Ltd."? Please verify by querying data from both this view and the base table. Can the data be inserted through this view? If not, please explain.



ไม่สามารถเพิ่มได้เนื่องจากไม่ผ่านเงื่อนไขของ mini\_customer\_view ที่ให้ขึ้นต้นด้วย Mini

What happened to the row of the customer name "Mini SUNISA" ? Please verify by querying data from both this view and the base table. Can the data be inserted through this view? If not, please explain.



ไม่ผ่านเนื่องจากไม่ผ่านเงื่อนไขของ mini\_ltd\_customer\_view ที่กำหนดไว้ให้ลงท้ายด้วย Ltd.

คิด check option a casahed

11. Please insert one new row through the “mini\_ltd\_customer\_view” view. You should create the data by yourself that can be inserted through this view. Please verify by querying data from both the views and the base table.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'Views' expanded, showing a list of views including 'customer\_samecity\_view', 'maxcredit\_city\_view', 'maxcredit\_london', 'mini\_customer\_view', 'mini\_ltd\_customer\_view', 'prod\_minstock\_view', 'test', 'totalamount\_orders\_view', 'usa\_office\_view', 'Stored Procedures', and 'Functions'. The 'Information' tab is selected, showing details for the 'View: mini\_customer\_view' with columns: 'cno' (int), 'cname' (varchar(50)), 'city' (varchar(50)), and 'country' (varchar(50)).

The main query window (q1\*) contains the following SQL code:

```

373
374
375 -- 64130500053
376 -- 11
377
378 insert into mini_ltd_customer_view
379 values ('9002', 'Mini PONGPAIROCH Ltd.', 'Bangkok', 'THAILAND');
380 select*from mini_ltd_customer_view ;

```

The 'Result Grid' shows the following data:

| custno | custname                     | custcity    | custcountry |
|--------|------------------------------|-------------|-------------|
| 124    | Mini Gifts Distributors Ltd. | San Rafael  | USA         |
| 320    | Mini Creations Ltd.          | New Bedford | USA         |
| 124    | Mini Gifts Distributors Ltd. | San Rafael  | USA         |
| 320    | Mini Creations Ltd.          | New Bedford | USA         |
| 9002   | Mini PONGPAIROCH Ltd.        | Bangkok     | THAILAND    |

The 'Output' tab shows the following actions:

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 11 | 20:37:45 | create or replace view mini_ltd_customer_view ( custno , custname , custci... | 0 row(s) affected  | 0.015 sec             |
| 12 | 20:37:48 | insert into mini_ltd_customer_view values ('9002', 'Mini PONGPAIROCH Lt...    | 1 row(s) affected  | 0.000 sec             |
| 13 | 20:37:52 | select*from mini_ltd_customer_view LIMIT 0, 1000                              | 17 row(s) returned | 0.000 sec / 0.000 sec |

12. Remove two existing views that were created in Lab03. You can select two views by yourself.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'Views' expanded, showing a list of views including 'usa\_customers', 'maxcredit\_city\_view', 'maxcredit\_london', 'mini\_customer\_view', 'mini\_ltd\_customer\_view', 'prod\_minstock\_view', 'test', 'totalamount\_orders\_view', 'usa\_office\_view', 'Stored Procedures', and 'Functions'. The 'Information' tab is selected, showing details for the 'View: mini\_customer\_view' with columns: 'cno' (int), 'cname' (varchar(50)), 'city' (varchar(50)), and 'country' (varchar(50)).

The main query window (q1\*) contains the following SQL code:

```

371 where cname like '%Ltd.'
372 with cascaded check option ;
373
374
375 -- 64130500053
376 -- 11
377
378 insert into mini_ltd_customer_view
379 values ('9002', 'Mini PONGPAIROCH Ltd.', 'Bangkok', 'THAILAND');
380 select*from mini_ltd_customer_view ;
381
382
383 -- 64130500053
384 -- 12
385 drop view customer_samecity_view ;
386

```

The 'Output' tab shows the following actions:

| #  | Time     | Action   | Message   | Duration / Fetch      |
|----|----------|--|---|-----------------------|
| 13 | 20:37:52 | select*from mini_ltd_customer_view LIMIT 0, 1000     | 17 row(s) returned  | 0.000 sec / 0.000 sec |
| 14 | 20:38:42 | drop view emp_official_view , customer_samecity_view | Error Code: 1051. Unknown table 'classicmodels.emp_official_view' | 0.000 sec             |
| 15 | 20:40:00 | drop view customer_samecity_view                     | 0 row(s) affected   | 0.016 sec             |