INT205 – Database Management System
1/2564

LAB05: View and Subquery (V2)

Submission:

- Submit a lab file named "int205_lab05_xxxxxxxxxxx.docx/.pdf" into the LEB2 system. xxxxxxxxxxx = your student id

Due Date & Time:

- Lecturer will inform the LAB05 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

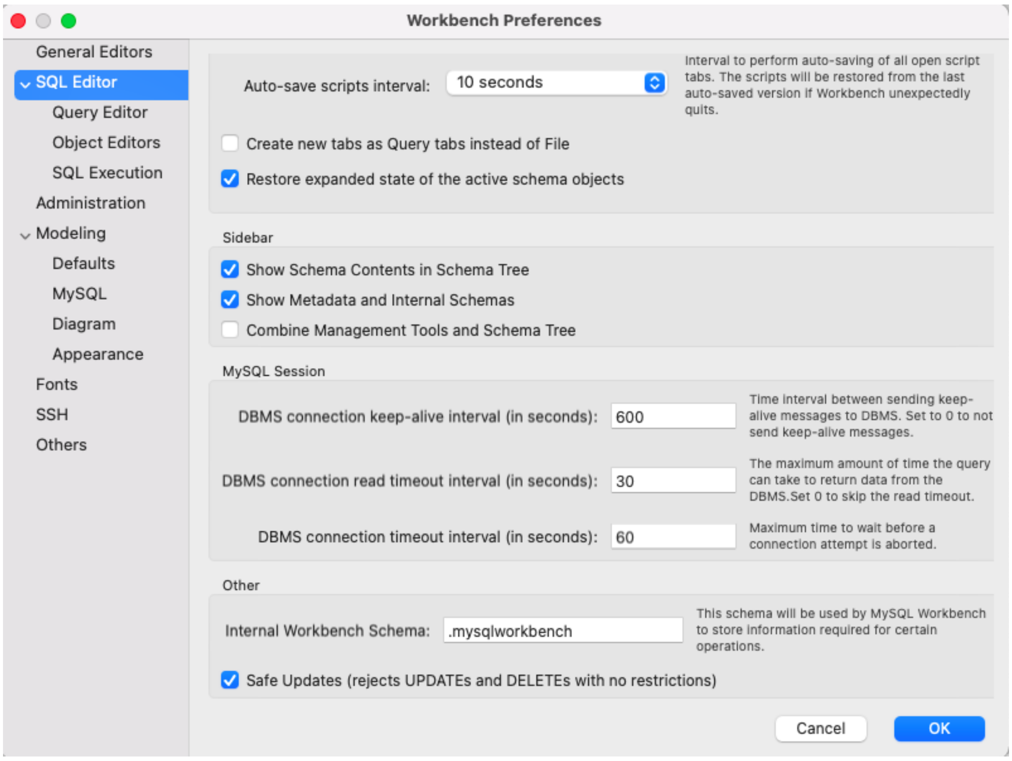
-- Syntax --

```
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



Auto-save scripts interval: 10 seconds
Interval to perform auto-saving of all open script tabs. The scripts will be restored from the last auto-saved version if Workbench unexpectedly quits.

□ Create new tabs as Query tabs instead of File
☑ Restore expanded state of the active schema objects

Sidebar
☑ Show Schema Contents in Schema Tree
☑ Show Metadata and Internal Schemas
□ Combine Management Tools and Schema Tree

MySQL Session
DBMS connection keep-alive interval (in seconds): 600
Time interval between sending keep-alive messages to DBMS. Set to 0 to not send keep-alive messages.

DBMS connection read timeout interval (in seconds): 30
The maximum amount of time the query can take to return data from the DBMS.Set 0 to skip the read timeout.

DBMS connection timeout interval (in seconds): 60
Maximum time to wait before a connection attempt is aborted.

Other
Internal Workbench Schema: .mysqlworkbench
This schema will be used by MySQL Workbench to store information required for certain operations.

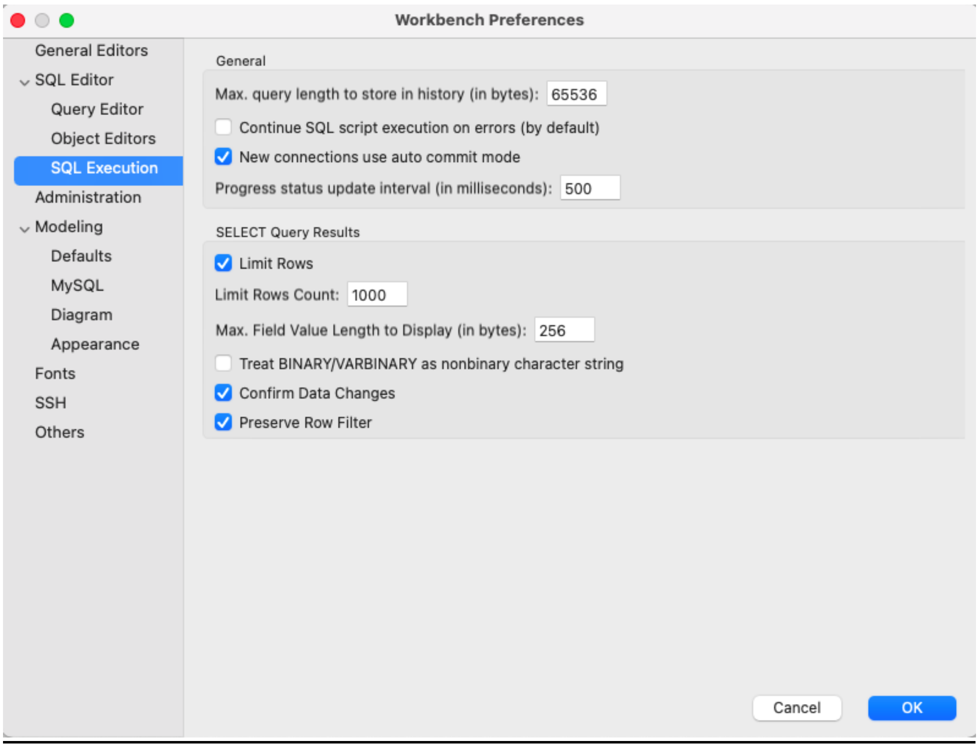
☑ Safe Updates (rejects UPDATEs and DELETEs with no restrictions)

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



General

Max. query length to store in history (in bytes): 65536

□ Continue SQL script execution on errors (by default)

☑ New connections use auto commit mode

Progress status update interval (in milliseconds): 500

SELECT Query Results

☑ Limit Rows

Limit Rows Count: 1000

Max. Field Value Length to Display (in bytes): 256

□ Treat BINARY/VARBINARY as nonbinary character string

☑ Confirm Data Changes

☑ Preserve Row Filter

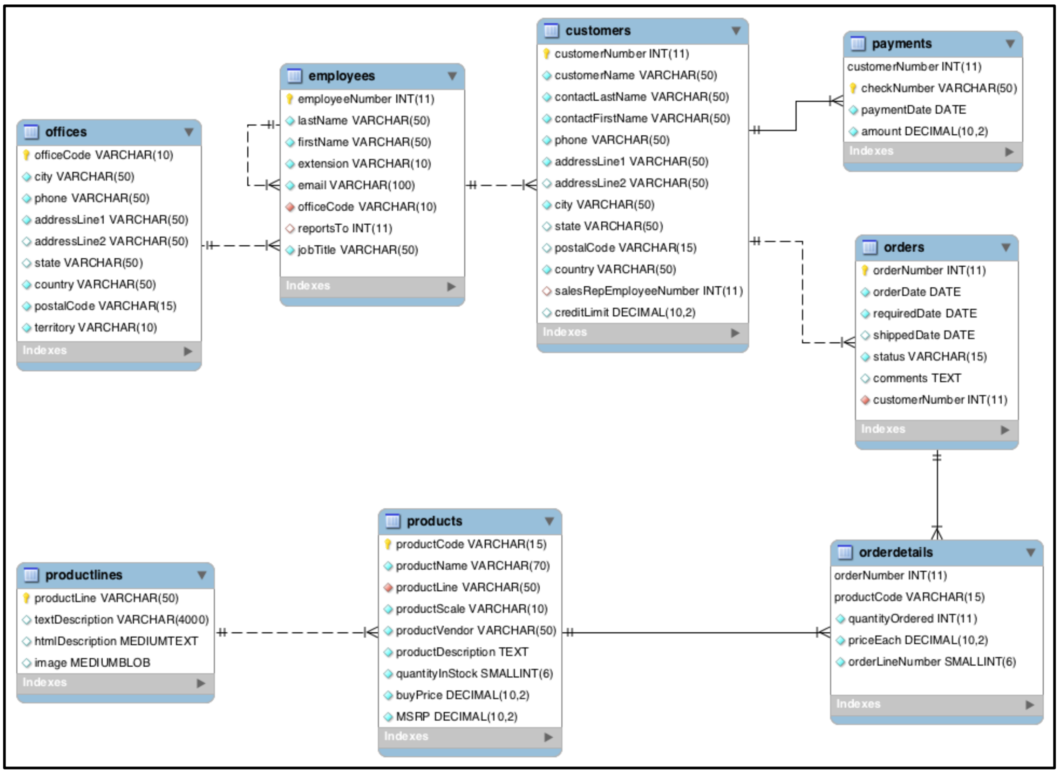
Switch to SQL Editor

- You should specify the classicmodels database before writing SQL statements using the following command:

  USE db_name;

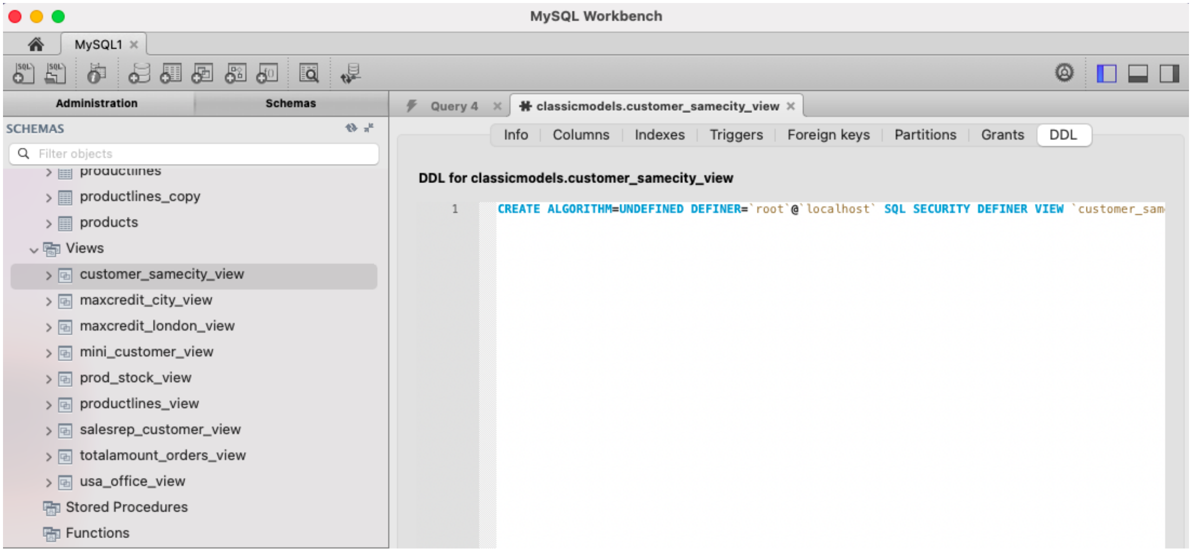
The USE statement tells MySQL to use the named database as the default (current) database for subsequent statements. This statement requires some privilege for the database or some object within it.

The 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:



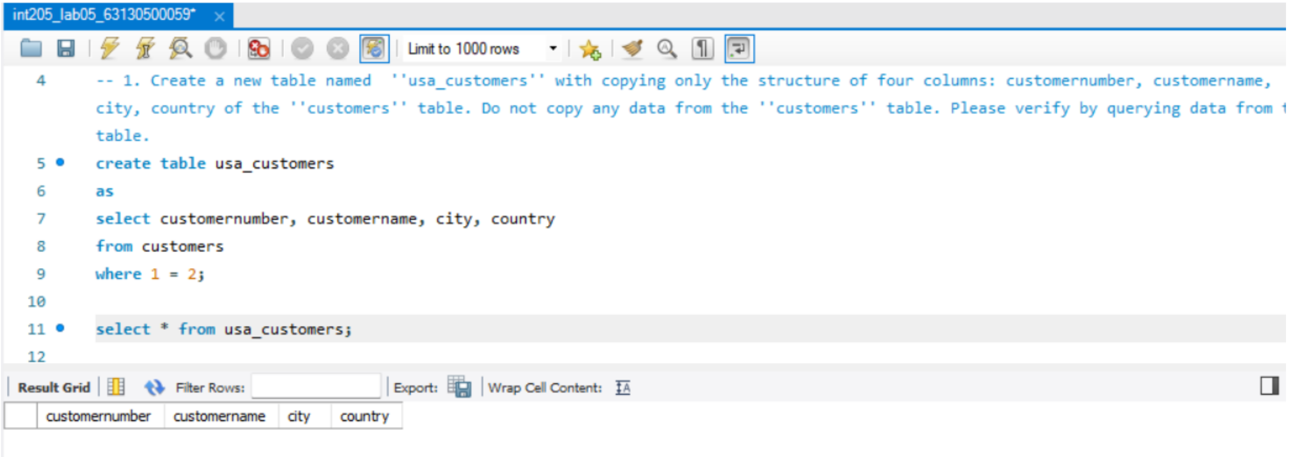
DDL for classicmodels.customer_samecity_view

```
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `customer_sam
```

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

use classicmodels;

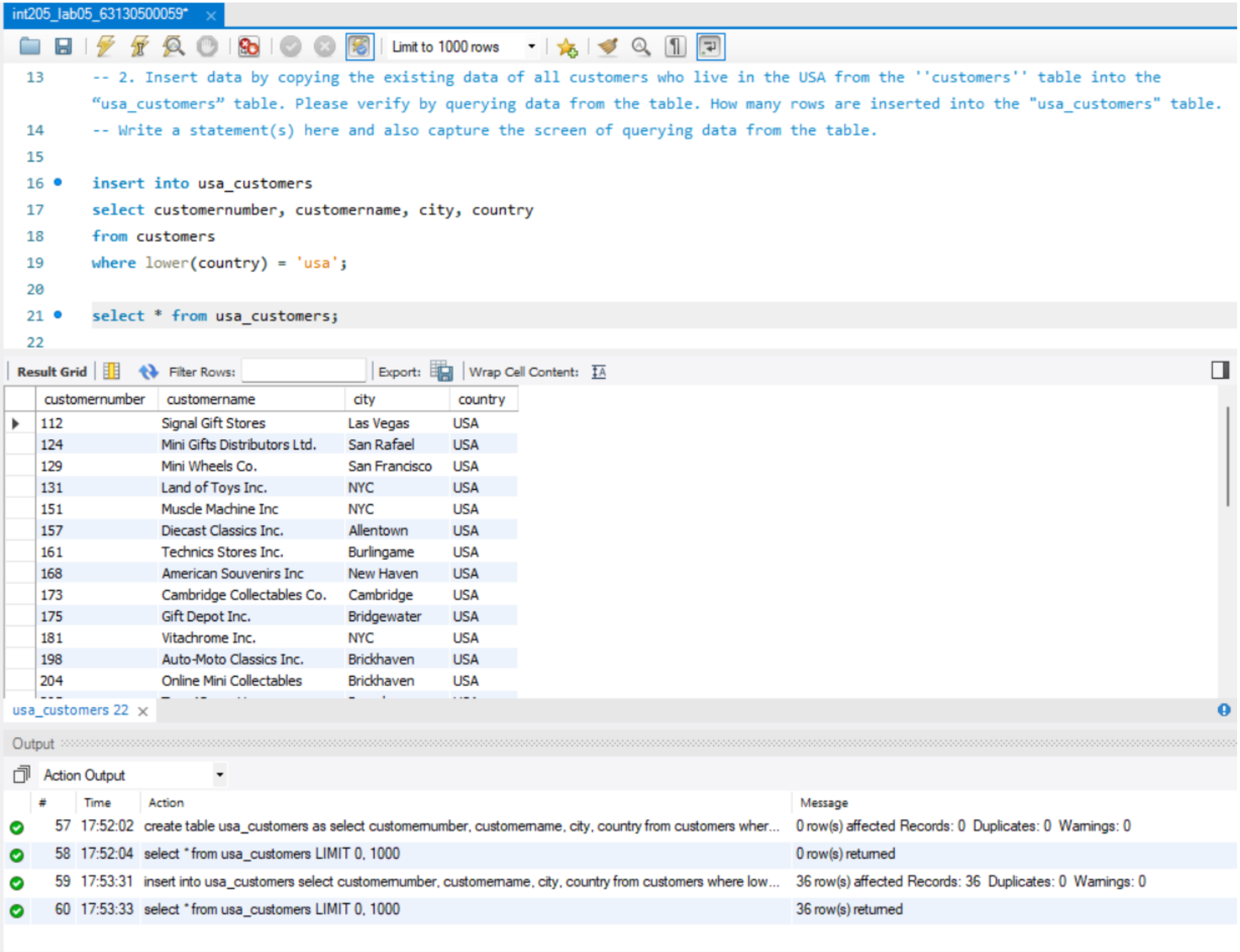
1. Create a new table named "usa_customers" with copying only the structure of four columns: customernumber, customername, city, country of the "customers" table. Do not copy any data from the "customers" table. Please verify by querying data from the table.



```sql
-- 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 t
table.
create table usa_customers
as
select customernumber, customername, city, country
from customers
where 1 = 2;

select * from usa_customers;
```

customernumber	customername	city	country

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.

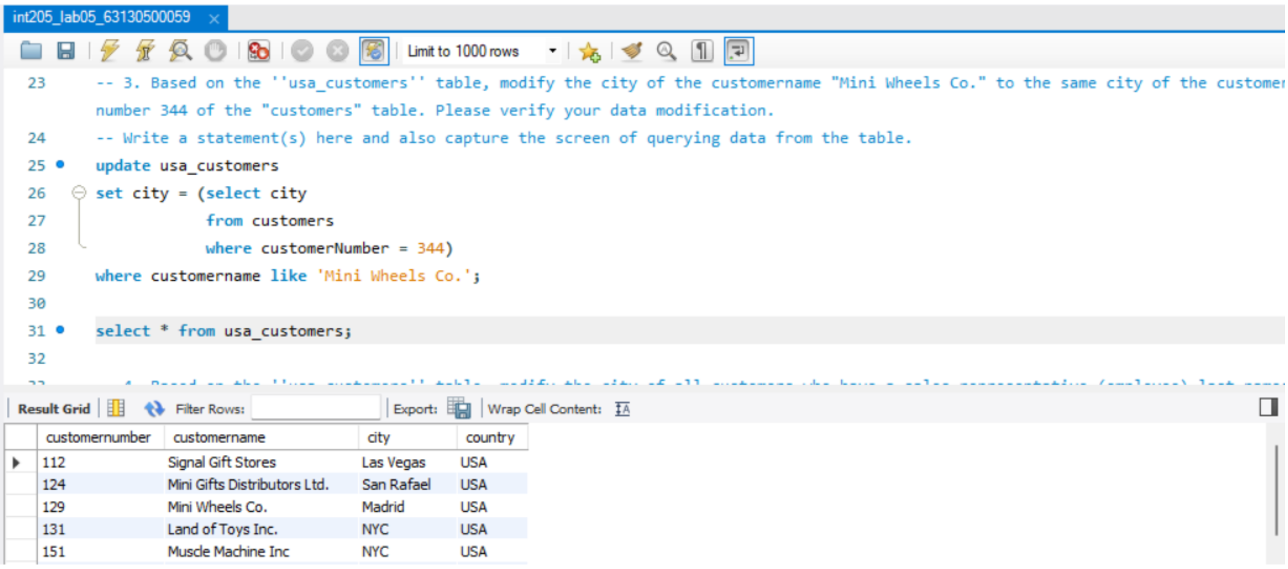


```sql
-- 2. Insert data by copying the existing data of all customers who live in the USA from the ''customers'' table into the
“usa_customers” table. Please verify by querying data from the table. How many rows are inserted into the "usa_customers" table.
-- Write a statement(s) here and also capture the screen of querying data from the table.

insert into usa_customers
select customernumber, customername, city, country
from customers
where lower(country) = 'usa';

select * from usa_customers;
```

customernumber	customername	city	country
112	Signal Gift Stores	Las Vegas	USA
124	Mini Gifts Distributors Ltd.	San Rafael	USA
129	Mini Wheels Co.	San Francisco	USA
131	Land of Toys Inc.	NYC	USA
151	Muscle Machine Inc	NYC	USA
157	Diecast Classics Inc.	Allentown	USA
161	Technics Stores Inc.	Burlingame	USA
168	American Souvenirs Inc	New Haven	USA
173	Cambridge Collectables Co.	Cambridge	USA
175	Gift Depot Inc.	Bridgewater	USA
181	Vitachrome Inc.	NYC	USA
198	Auto-Moto Classics Inc.	Brickhaven	USA
204	Online Mini Collectables	Brickhaven	USA

#	Time	Action	Message
57	17:52:02	create table usa_customers as select customernumber, customername, city, country from customers wher...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
58	17:52:04	select * from usa_customers LIMIT 0, 1000	0 row(s) returned
59	17:53:31	insert into usa_customers select customernumber, customername, city, country from customers where low...	36 row(s) affected Records: 36 Duplicates: 0 Warnings: 0
60	17:53:33	select * from usa_customers LIMIT 0, 1000	36 row(s) returned

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.

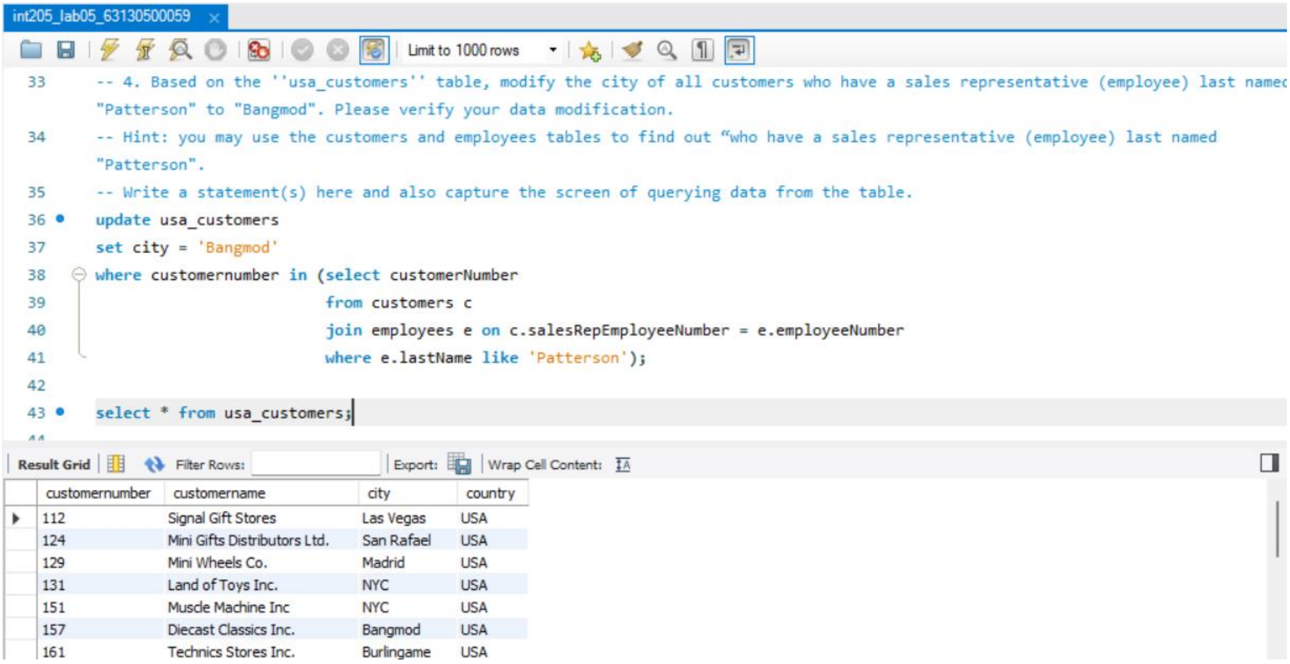


```
-- 3. Based on the ''usa_customers'' table, modify the city of the customername "Mini Wheels Co." to the same city of the custome
number 344 of the "customers" table. Please verify your data modification.
-- Write a statement(s) here and also capture the screen of querying data from the table.
update usa_customers
set city = (select city
            from customers
            where customerNumber = 344)
where customername like 'Mini Wheels Co.';

select * from usa_customers;

-- 4. Based on the ''usa_customers'' table, modify the city of all customers who have a sales representative (employee) last name
```

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

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.

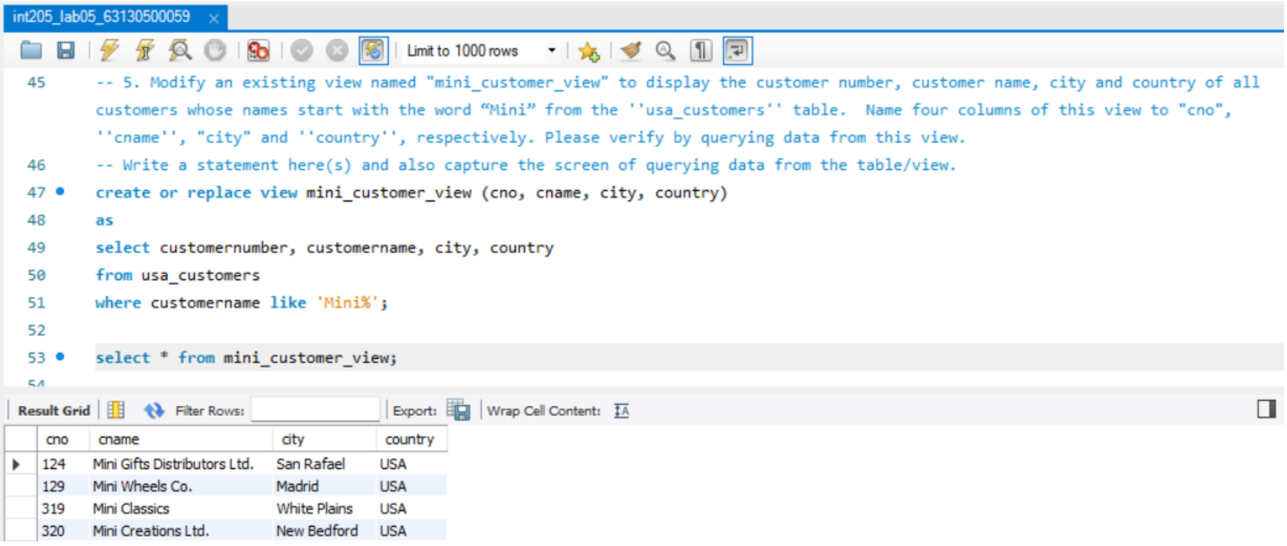
Hint: you may use the customers and employees tables to find out "who have a sales representative (employee) last named "Patterson".



```
-- 4. Based on the ''usa_customers'' table, modify the city of all customers who have a sales representative (employee) last namec
"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".
-- Write a statement(s) here and also capture the screen of querying data from the table.
update usa_customers
set city = 'Bangmod'
where customernumber in (select customerNumber
                         from customers c
                         join employees e on c.salesRepEmployeeNumber = e.employeeNumber
                         where e.lastName like 'Patterson');

select * from usa_customers;
```

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
157	Diecast Classics Inc.	Bangmod	USA
161	Technics Stores Inc.	Burlingame	USA

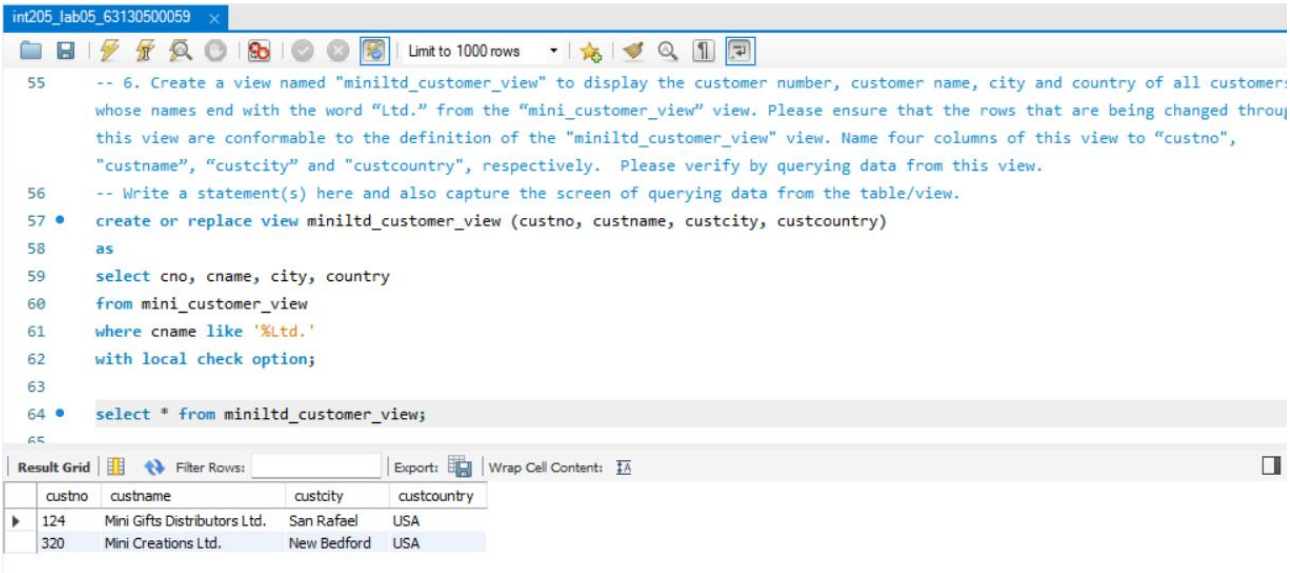
5. Modify an existing view named "mini_customer_view" to display the customer number, customer name, city and country of all customers whose names start with the word “Mini” from the ''usa_customers'' table. Name four columns of this view to "cno", ''cname'', "city" and ''country'', respectively. Please verify by querying data from this view.



```sql
-- 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.
-- Write a statement here(s) and also capture the screen of querying data from the table/view.
create or replace view mini_customer_view (cno, cname, city, country)
as
select customernumber, customername, city, country
from usa_customers
where customername like 'Mini%';

select * from mini_customer_view;
```

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

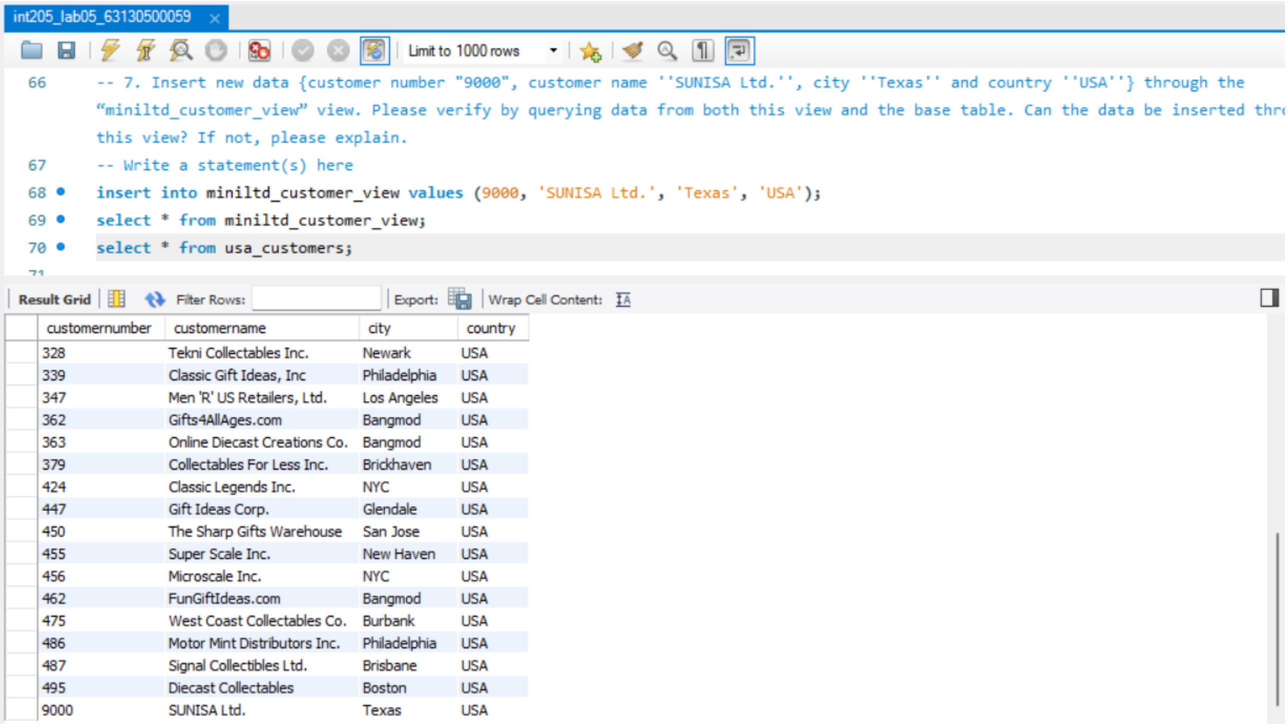
6. Create a view named "miniltd_customer_view" to display the customer number, customer name, city and country of all customers whose names end with the word “Ltd.” from the “mini_customer_view” view. Please ensure that the rows that are being changed through this view are conformable to the definition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



```sql
-- 6. Create a view named "miniltd_customer_view" to display the customer number, customer name, city and country of all customer
whose names end with the word “Ltd.” from the “mini_customer_view” view. Please ensure that the rows that are being changed throu
this view are conformable to the definition of the "miniltd_customer_view" view. Name four columns of this view to “custno",
"custname”, “custcity” and "custcountry", respectively.  Please verify by querying data from this view.
-- Write a statement(s) here and also capture the screen of querying data from the table/view.
create or replace view miniltd_customer_view (custno, custname, custcity, custcountry)
as
select cno, cname, city, country
from mini_customer_view
where cname like '%Ltd.'
with local check option;

select * from miniltd_customer_view;
```

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

7. Insert new data {customer number "9000", customer name ''SUNISA Ltd.'', city ''Texas'' and country ''USA''} through the “miniltd_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.

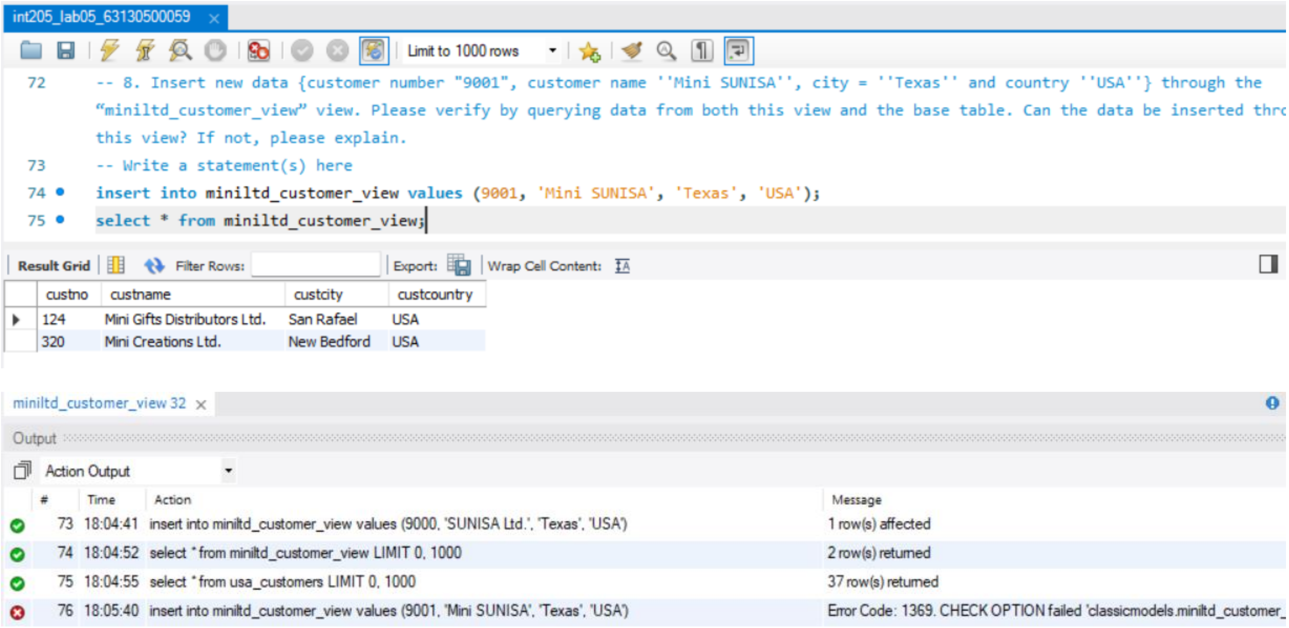


```sql
-- 7. Insert new data {customer number "9000", customer name ''SUNISA Ltd.'', city ''Texas'' and country ''USA''} through the
“miniltd_customer_view” view. Please verify by querying data from both this view and the base table. Can the data be inserted thr
this view? If not, please explain.
-- Write a statement(s) here
insert into miniltd_customer_view values (9000, 'SUNISA Ltd.', 'Texas', 'USA');
select * from miniltd_customer_view;
select * from usa_customers;
```

customernumber	customername	city	country
328	Tekni Collectables Inc.	Newark	USA
339	Classic Gift Ideas, Inc	Philadelphia	USA
347	Men 'R' US Retailers, Ltd.	Los Angeles	USA
362	Gifts4AllAges.com	Bangmod	USA
363	Online Diecast Creations Co.	Bangmod	USA
379	Collectables For Less Inc.	Brickhaven	USA
424	Classic Legends Inc.	NYC	USA
447	Gift Ideas Corp.	Glendale	USA
450	The Sharp Gifts Warehouse	San Jose	USA
455	Super Scale Inc.	New Haven	USA
456	Microscale Inc.	NYC	USA
462	FunGiftIdeas.com	Bangmod	USA
475	West Coast Collectables Co.	Burbank	USA
486	Motor Mint Distributors Inc.	Philadelphia	USA
487	Signal Collectibles Ltd.	Brisbane	USA
495	Diecast Collectables	Boston	USA
9000	SUNISA Ltd.	Texas	USA

สามารถ insert ได้ เนื่องจากตรงเงื่อนไขของ miniltd_customer_view with local check option;

8. Insert new data {customer number "9001", customer name ''Mini SUNISA'', city = ''Texas'' and country ''USA''} through the “miniltd_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.



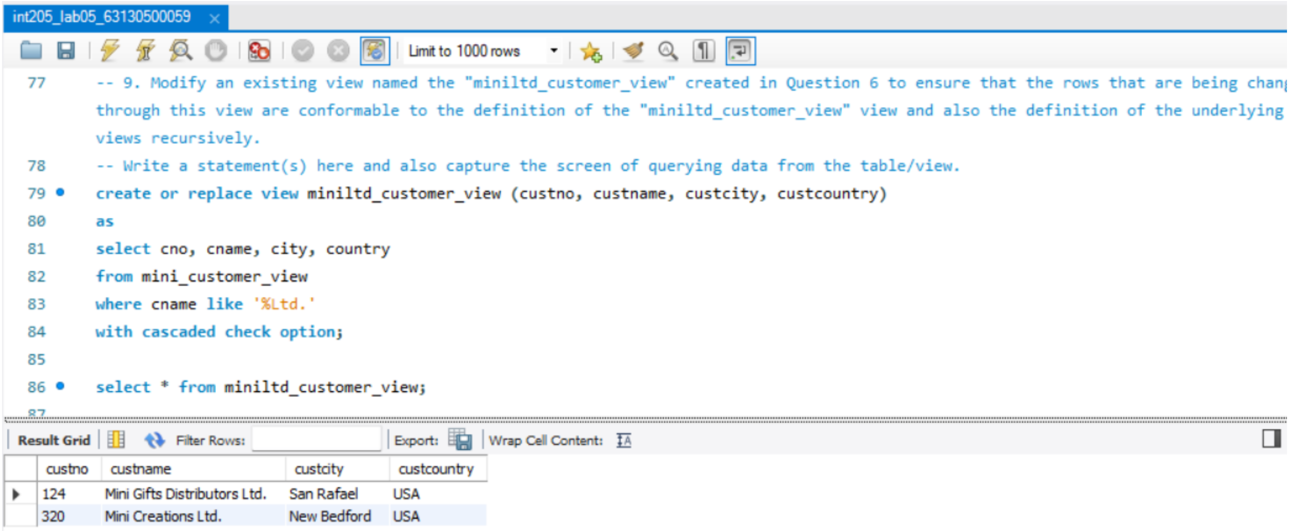
```sql
-- 8. Insert new data {customer number "9001", customer name ''Mini SUNISA'', city = ''Texas'' and country ''USA''} through the
“miniltd_customer_view” view. Please verify by querying data from both this view and the base table. Can the data be inserted thro
this view? If not, please explain.
-- Write a statement(s) here
insert into miniltd_customer_view values (9001, 'Mini SUNISA', 'Texas', 'USA');
select * from miniltd_customer_view;
```

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

	#	Time	Action	Message
✓	73	18:04:41	insert into miniltd_customer_view values (9000, 'SUNISA Ltd.', 'Texas', 'USA')	1 row(s) affected
✓	74	18:04:52	select * from miniltd_customer_view LIMIT 0, 1000	2 row(s) returned
✓	75	18:04:55	select * from usa_customers LIMIT 0, 1000	37 row(s) returned
✕	76	18:05:40	insert into miniltd_customer_view values (9001, 'Mini SUNISA', 'Texas', 'USA')	Error Code: 1369. CHECK OPTION failed 'classicmodels.miniltd_customer_

ไม่สามารถ insert ได้ เนื่องจากไม่ตรงเงื่อนไขของ miniltd_customer_view with local check option ที่ว่า customer name ต้องลงท้ายด้วย Ltd.

9. Modify an existing view named the "miniltd_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 "miniltd_customer_view" view and also the definition of the underlying views recursively.

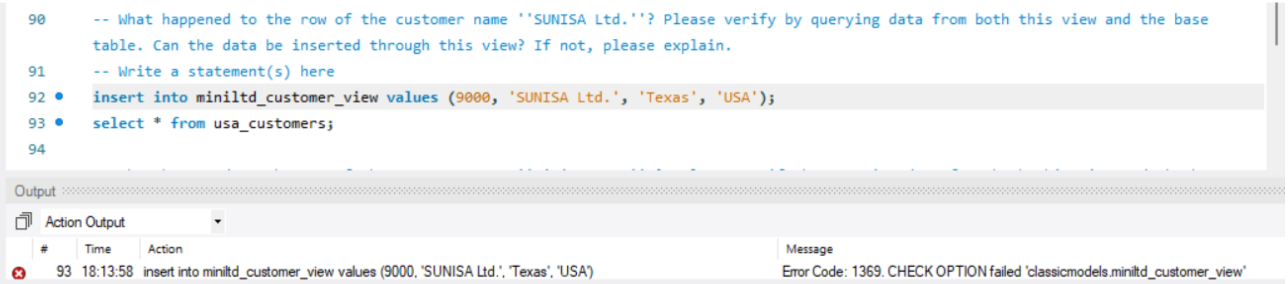


```
-- 9. Modify an existing view named the "miniltd_customer_view" created in Question 6 to ensure that the rows that are being chan
through this view are conformable to the definition of the "miniltd_customer_view" view and also the definition of the underlying
views recursively.
-- Write a statement(s) here and also capture the screen of querying data from the table/view.
create or replace view miniltd_customer_view (custno, custname, custcity, custcountry)
as
select cno, cname, city, country
from mini_customer_view
where cname like '%Ltd.'
with cascaded check option;

select * from miniltd_customer_view;
```

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

10. Try to insert the same data of Question 7-8 again.

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.

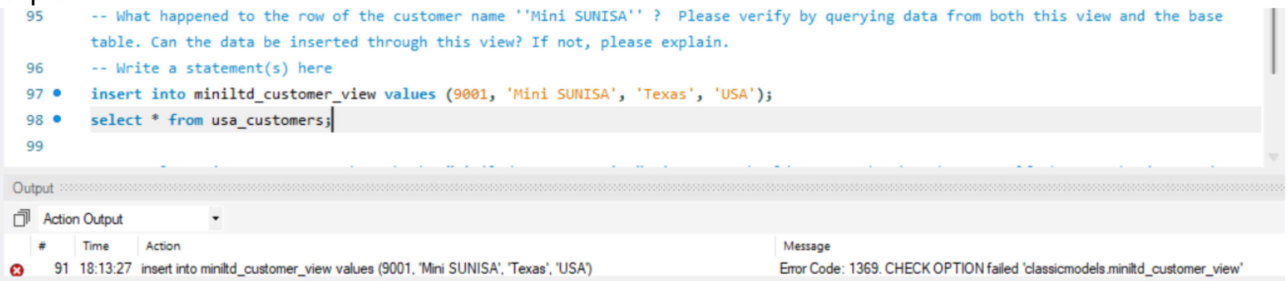


```
-- What happened to the row of the customer name ''SUNISA Ltd.''? Please verify by querying data from both this view and the base
table. Can the data be inserted through this view? If not, please explain.
-- Write a statement(s) here
insert into miniltd_customer_view values (9000, 'SUNISA Ltd.', 'Texas', 'USA');
select * from usa_customers;
```

```
#   Time      Action                                                                      Message
93  18:13:58  insert into miniltd_customer_view values (9000, 'SUNISA Ltd.', 'Texas', 'USA')   Error Code: 1369. CHECK OPTION failed 'classicmodels.miniltd_customer_view'
```

ไม่สามารถ insert ได้ เพราะ check option failed เนื่องจากเงื่อนไขของ mini_customer_view ที่ชื่อต้องขึ้นต้นด้วย Mini เป็น false และ เงื่อนไขของ miniltd_customer_view ที่ชื่อต้องลงท้ายด้วย Ltd. เป็น ture จึงได้ผลออกมาเป็น false

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.



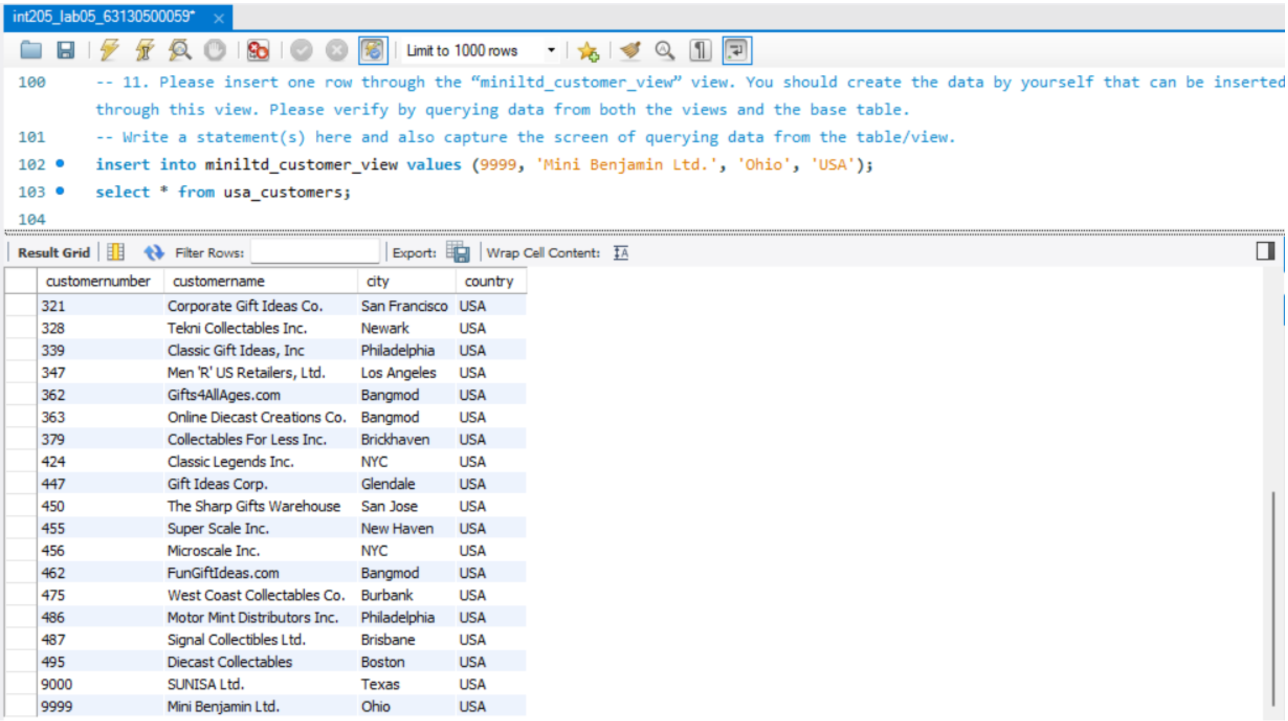
```
-- What happened to the row of the customer name ''Mini SUNISA'' ?  Please verify by querying data from both this view and the base
table. Can the data be inserted through this view? If not, please explain.
-- Write a statement(s) here
insert into miniltd_customer_view values (9001, 'Mini SUNISA', 'Texas', 'USA');
select * from usa_customers;
```

```
#   Time      Action                                                                      Message
91  18:13:27  insert into miniltd_customer_view values (9001, 'Mini SUNISA', 'Texas', 'USA')   Error Code: 1369. CHECK OPTION failed 'classicmodels.miniltd_customer_view'
```

ไม่สามารถ insert ได้ เพราะ check option failed เนื่องจากเงื่อนไขของ mini_customer_view ที่ชื่อต้องขึ้นต้นด้วย Mini เป็น ture และ เงื่อนไขของ miniltd_customer_view ที่ชื่อต้องลงท้ายด้วย Ltd. เป็น false จึงได้ผลออกมาเป็น false

INT205 – Database Management System
1/2564

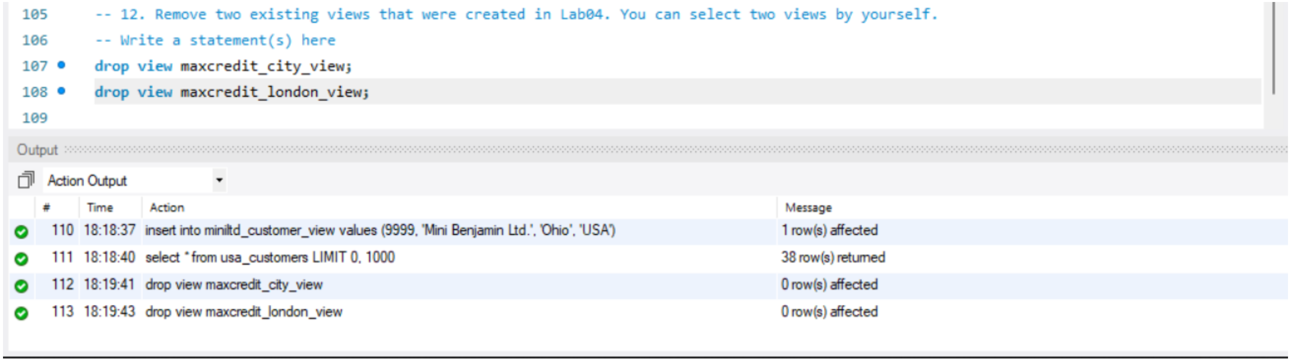
11. Please insert one row through the “miniltd_customer_view” view. You should create the data by yourself that can be inserted through this view. Please verify by querying data from both the views and the base table.



```sql
-- 11. Please insert one row through the “miniltd_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.
-- Write a statement(s) here and also capture the screen of querying data from the table/view.
insert into miniltd_customer_view values (9999, 'Mini Benjamin Ltd.', 'Ohio', 'USA');
select * from usa_customers;
```

customernumber	customername	city	country
321	Corporate Gift Ideas Co.	San Francisco	USA
328	Tekni Collectables Inc.	Newark	USA
339	Classic Gift Ideas, Inc	Philadelphia	USA
347	Men 'R' US Retailers, Ltd.	Los Angeles	USA
362	Gifts4AllAges.com	Bangmod	USA
363	Online Diecast Creations Co.	Bangmod	USA
379	Collectables For Less Inc.	Brickhaven	USA
424	Classic Legends Inc.	NYC	USA
447	Gift Ideas Corp.	Glendale	USA
450	The Sharp Gifts Warehouse	San Jose	USA
455	Super Scale Inc.	New Haven	USA
456	Microscale Inc.	NYC	USA
462	FunGiftIdeas.com	Bangmod	USA
475	West Coast Collectables Co.	Burbank	USA
486	Motor Mint Distributors Inc.	Philadelphia	USA
487	Signal Collectibles Ltd.	Brisbane	USA
495	Diecast Collectables	Boston	USA
9000	SUNISA Ltd.	Texas	USA
9999	Mini Benjamin Ltd.	Ohio	USA

12. Remove two existing views that were created in Lab04. You can select two views by yourself.



```sql
-- 12. Remove two existing views that were created in Lab04. You can select two views by yourself.
-- Write a statement(s) here
drop view maxcredit_city_view;
drop view maxcredit_london_view;
```

```
#    Time      Action                                                                  Message
110  18:18:37  insert into miniltd_customer_view values (9999, 'Mini Benjamin Ltd.', 'Ohio', 'USA')   1 row(s) affected
111  18:18:40  select * from usa_customers LIMIT 0, 1000                               38 row(s) returned
112  18:19:41  drop view maxcredit_city_view                                           0 row(s) affected
113  18:19:43  drop view maxcredit_london_view                                         0 row(s) affected
```