Critical Thinking Assignment 4: Modifying Tables

Alexander Ricciardi

Colorado State University Global

ITS410: Database Management

Dr. Murthy Rallapalli

May 11, 2025

Critical Thinking Assignment 4: Modifying Tables

This documentation is part of the Critical Thinking 4 Assignment from ITS410: Database Management at Colorado State University Global. The documentation provides screenshots showcasing modifying tables using MySQL and the My Guitar Shop database.

The Assignment Direction:

Modifying Tables
Using the My Guitar Shop database you installed in Module 1, develop the following queries.
To test whether a table has been modified correctly as you do these exercises, you can write and run an appropriate `SELECT` statement.

1. Write an `INSERT` statement that adds this row to the Categories table:
   ```
   category_name:              Brass
   ```
   Code the `INSERT` statement so MySQL automatically generates the `category_id` column. Execute the query and take a screenshot of the query and the results.

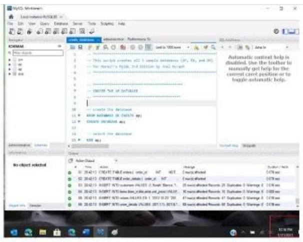
2. Write an `UPDATE` statement that modifies the row you just added to the Categories table. This statement should change the `category_name` column to "`Woodwinds`," and it should use the `category_id` column to identify the row. Execute the query and take a screenshot of the query and the results.

3. Write a `DELETE` statement that deletes the row you added to the `categories` table in exercise 1. This statement should use the `category_id` column to identify the row. Execute the query and take a screenshot of the query and the results.

4. Write an `INSERT` statement that adds this row to the `products` table:
   ```
   product_id:            The next automatically generated ID
   category_id:           4
   product_code:          dgx_640
   product_name:          Yamaha DGX 640 88-Key Digital Piano
   description:           Long description to come
   list_price:            799.99
   discount_percent:      0
   date_added:            Today's date/time
   ```

   Use a column list for this statement. Execute the query and take a screenshot of the query and the results

All the screenshots should show current date. Example of screenshot.



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

No object selected

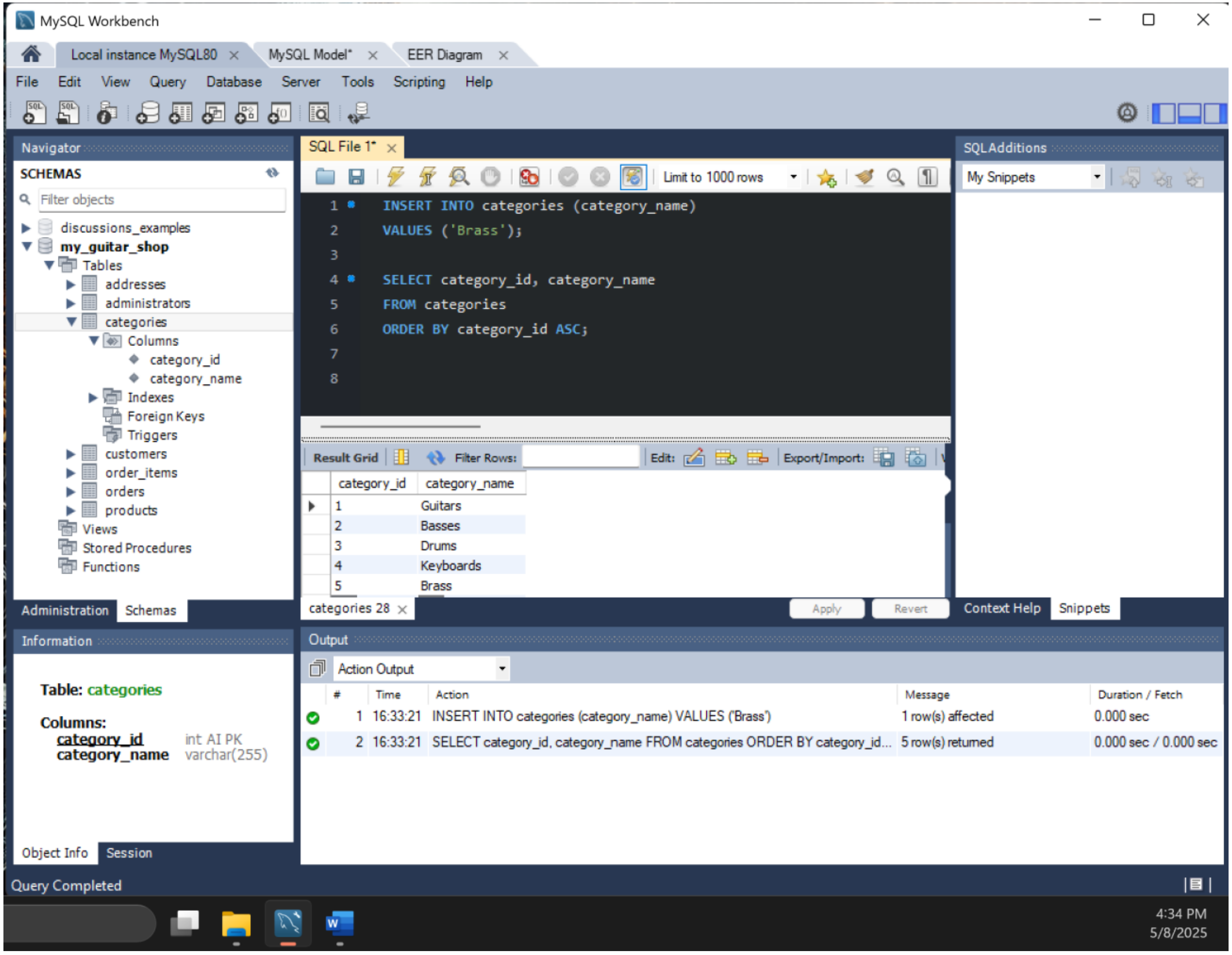
Submit your labeled results screenshots in a Word file.

Screenshots

Step 1: Write an `INSERT` statement that adds this row to the `categories` table:
`category_name:` `Brass`
Note that in Figure 1 below, MySQL Workbench provides the following information about the `category_id` column: `int AI PK`
`Int` stands for integer, `AI` for Auto Increment (meaning that when a new row is inserted, the new value in the column `category_id` is an auto increment by +1 of the previous `category_id` value), and `PK` stands for Primary Key.

Figure 1

Checking



```
INSERT INTO categories (category_name)
VALUES ('Brass');

SELECT category_id, category_name
FROM categories
ORDER BY category_id ASC;
```

category_id	category_name
1	Guitars
2	Basses
3	Drums
4	Keyboards
5	Brass

Table: categories

Columns:
category_id int AI PK
category_name varchar(255)

```
#  Time      Action                                                                   Message             Duration / Fetch
1  16:33:21  INSERT INTO categories (category_name) VALUES ('Brass')                  1 row(s) affected   0.000 sec
2  16:33:21  SELECT category_id, category_name FROM categories ORDER BY category_id...  5 row(s) returned   0.000 sec / 0.000 sec
```

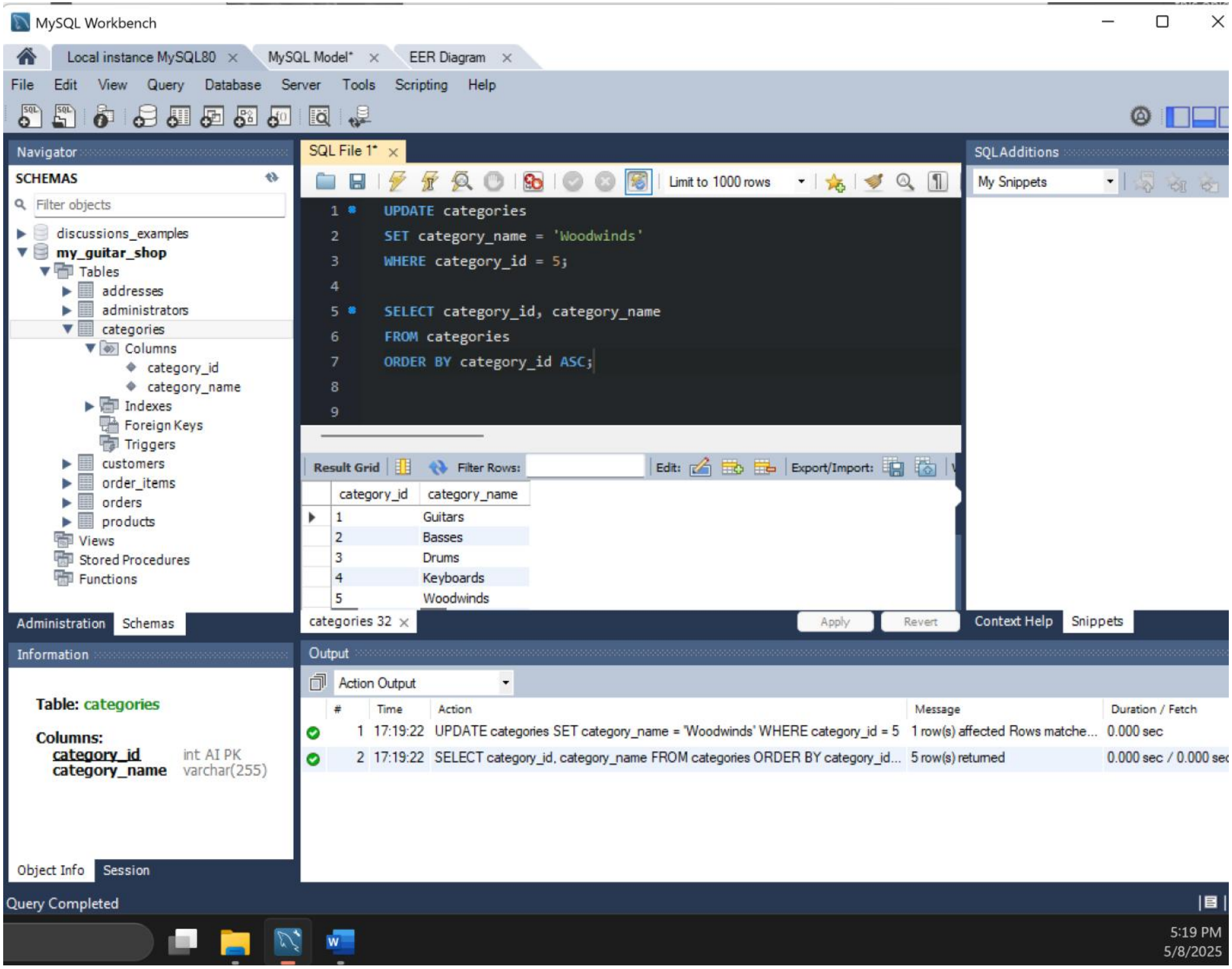
Note: The figure illustrates the MySQL Workbench result after performing steps 1.

Please see the next page.

Step 2: Writing an `UPDATE` statement that modifies the row that was just added to the Categories table. As shown in Figure 1, that row has a `category_id = 5`

Figure 2

Assignment Steps 2



```
UPDATE categories
SET category_name = 'Woodwinds'
WHERE category_id = 5;

SELECT category_id, category_name
FROM categories
ORDER BY category_id ASC;
```

category_id	category_name
1	Guitars
2	Basses
3	Drums
4	Keyboards
5	Woodwinds

Information

Table: categories

Columns:

category_id	int AI PK
category_name	varchar(255)

#	Time	Action	Message	Duration / Fetch
1	17:19:22	UPDATE categories SET category_name = 'Woodwinds' WHERE category_id = 5	1 row(s) affected Rows matche...	0.000 sec
2	17:19:22	SELECT category_id, category_name FROM categories ORDER BY category_id...	5 row(s) returned	0.000 sec / 0.000 sec

Query Completed

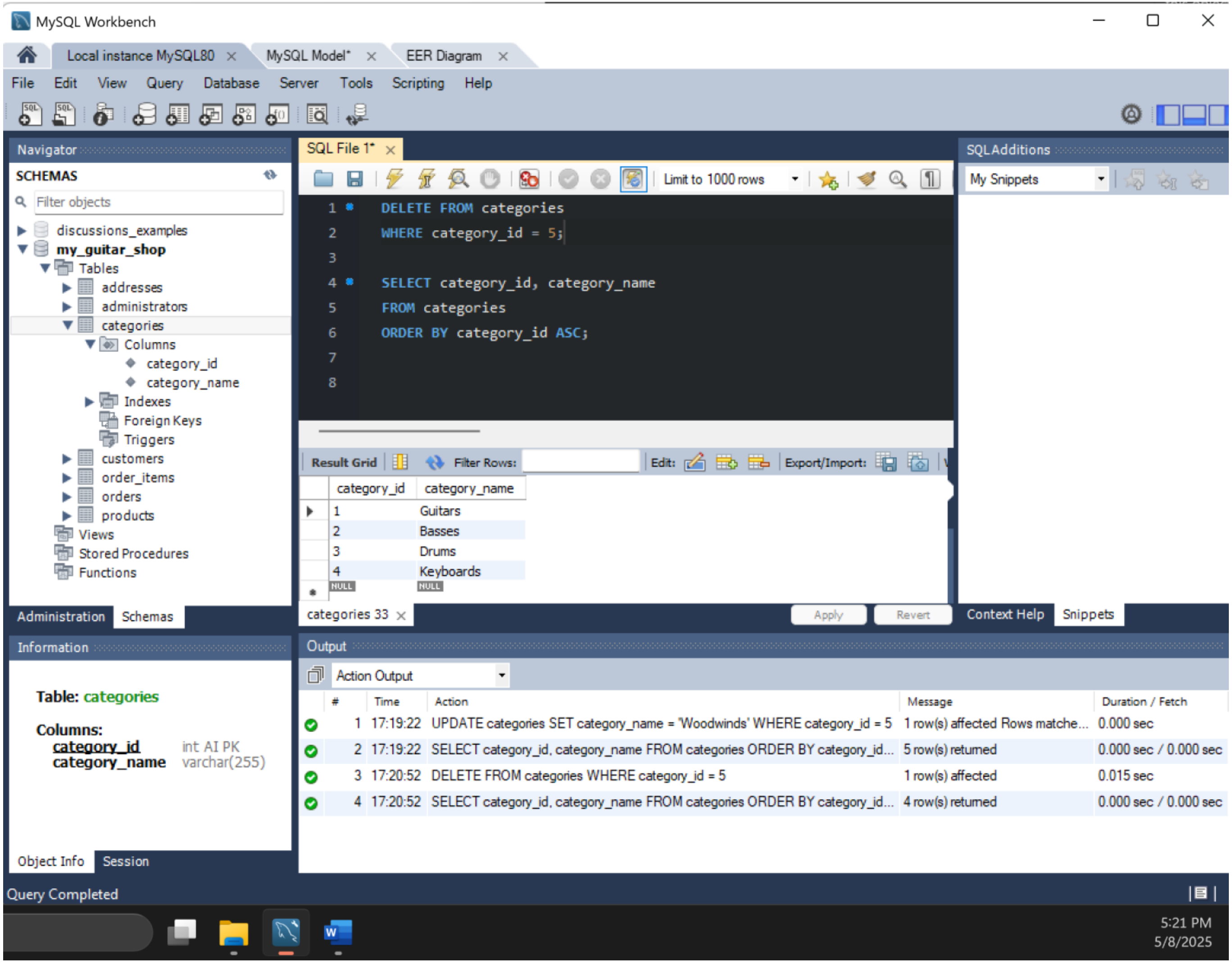
Note: The figure illustrates the MySQL Workbench result after performing steps 2.

Please see the next page.

Step 3: Writing a `DELETE` statement that deletes the row that was added to the `categories` table in exercise 1.

Figure 3

Assignment Step 3



```sql
DELETE FROM categories
WHERE category_id = 5;

SELECT category_id, category_name
FROM categories
ORDER BY category_id ASC;
```

category_id	category_name
1	Guitars
2	Basses
3	Drums
4	Keyboards
NULL	NULL

Table: categories

Columns:
category_id int AI PK
category_name varchar(255)

	#	Time	Action	Message	Duration / Fetch
✓	1	17:19:22	UPDATE categories SET category_name = 'Woodwinds' WHERE category_id = 5	1 row(s) affected Rows matche...	0.000 sec
✓	2	17:19:22	SELECT category_id, category_name FROM categories ORDER BY category_id...	5 row(s) returned	0.000 sec / 0.000 sec
✓	3	17:20:52	DELETE FROM categories WHERE category_id = 5	1 row(s) affected	0.015 sec
✓	4	17:20:52	SELECT category_id, category_name FROM categories ORDER BY category_id...	4 row(s) returned	0.000 sec / 0.000 sec

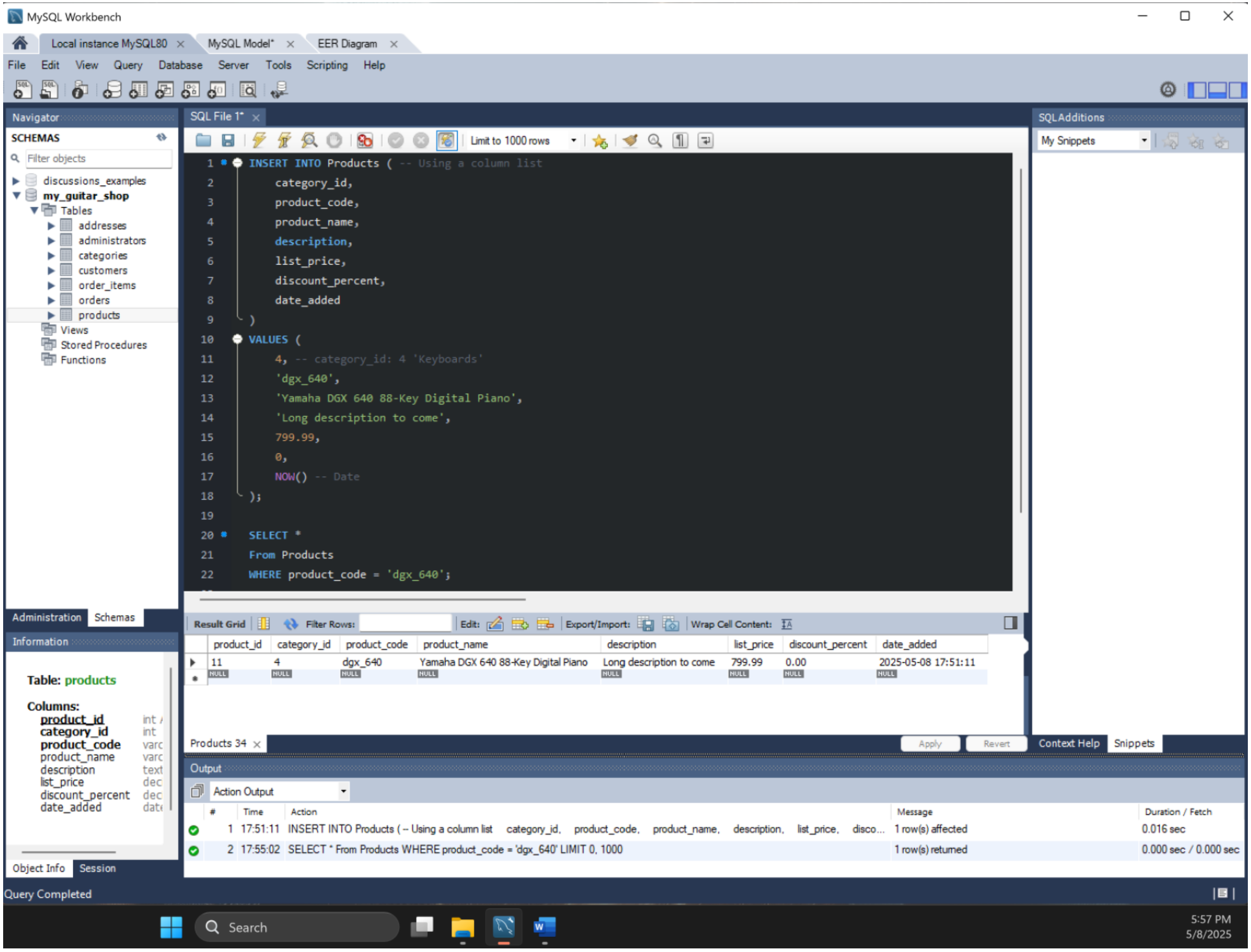
Note: The figure illustrates the MySQL Workbench result after performing steps 3.

Please see the next page.

Step 4: Writing an `INSERT` statement that adds this row to the `products` table. Similar to the `category_id` column in the `categories` table, the `product_id` is defined as: `int AI PK`

Figure 4

Assignment Step 4



```sql
INSERT INTO Products ( -- Using a column list
    category_id,
    product_code,
    product_name,
    description,
    list_price,
    discount_percent,
    date_added
)
VALUES (
    4, -- category_id: 4 'Keyboards'
    'dgx_640',
    'Yamaha DGX 640 88-Key Digital Piano',
    'Long description to come',
    799.99,
    0,
    NOW() -- Date
);

SELECT *
From Products
WHERE product_code = 'dgx_640';
```

product_id	category_id	product_code	product_name	description	list_price	discount_percent	date_added
11	4	dgx_640	Yamaha DGX 640 88-Key Digital Piano	Long description to come	799.99	0.00	2025-05-08 17:51:11
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
1	17:51:11	INSERT INTO Products (-- Using a column list category_id, product_code, product_name, description, list_price, disco...	1 row(s) affected	0.016 sec
2	17:55:02	SELECT * From Products WHERE product_code = 'dgx_640' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Note: The figure illustrates the MySQL Workbench result after performing steps 4.