



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

LAB 2 DML Part 1

NAME	ADAM ISMAIL HASSAN AMER ABOURAYA
MATRIC NO.	A22EC0002
Course Code	SECD2523
Section	08

Section 6 Lesson 4 Exercise 1

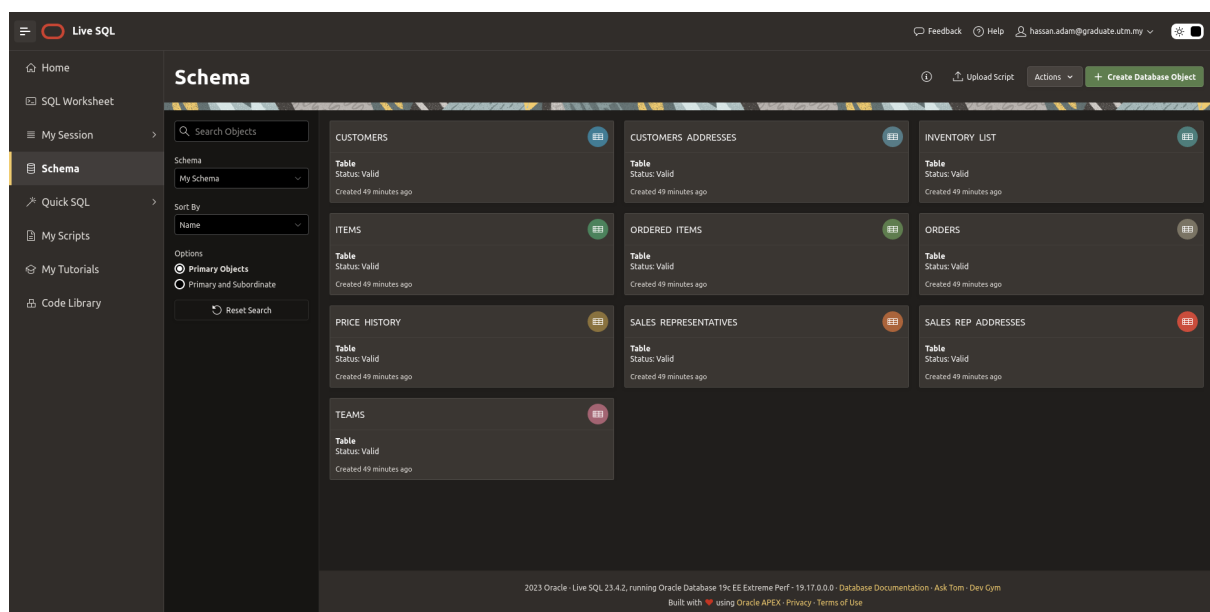
Part 1 : Running a script to populate the tables.

You have to consider the order of the tables when populating them. A table that has a foreign key field cannot be populated before the related table with the primary key.

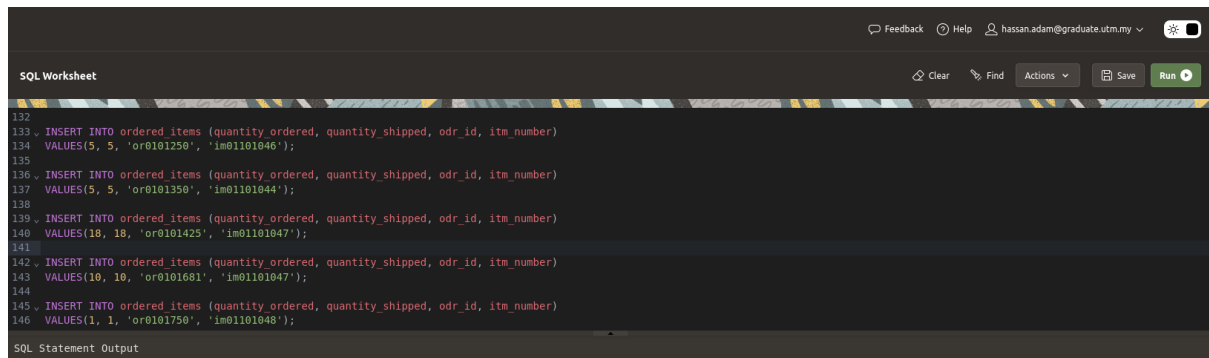
1. Use the table mapping document and list the order that you would use to populate

the tables.

- a) inventory_list
- b) items
- c) price_history
- d) sales_representatives
- e) sales_rep_addresses
- f) teams
- g) customers
- h) customers_addresses
- i) orders
- j) ordered_items



2. Open the “sports data.sql” and look at the order the data is being added there, does your list match? This file can be found in the Section 6 Lesson 4 interaction (sports data.zip) and must first be extracted.



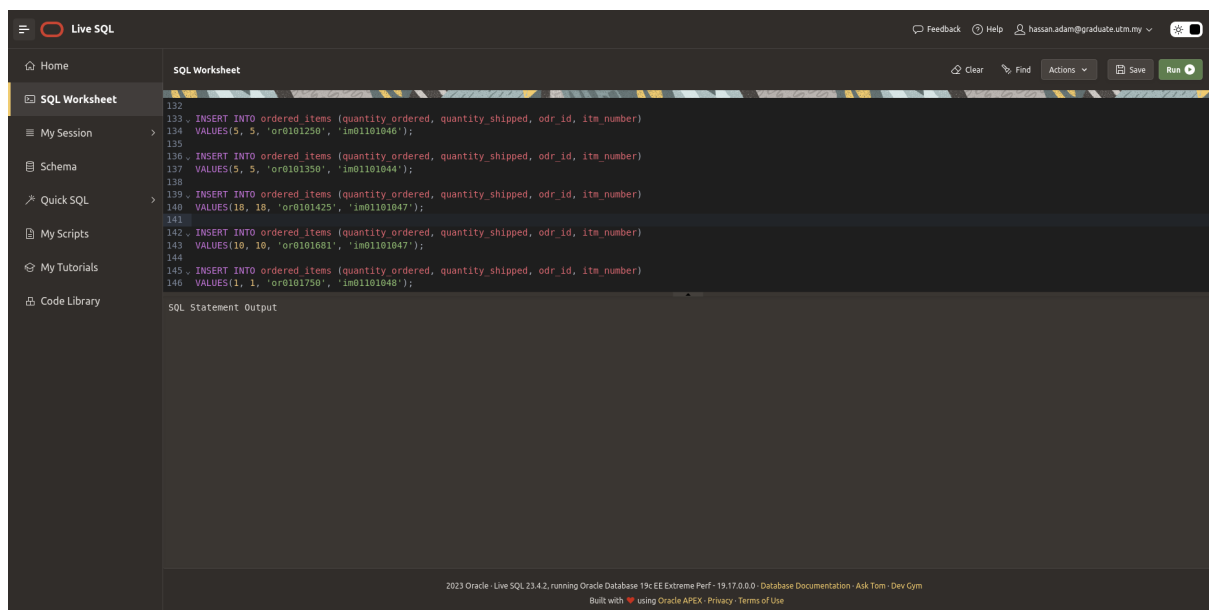
The screenshot shows an SQL Worksheet interface with a dark theme. The top bar includes a 'Feedback' icon, a 'Help' icon, a user profile 'hassan.adam@graduate.utm.my', and a 'Run' button. Below the top bar, there are buttons for 'Clear', 'Find', 'Actions', 'Save', and 'Run'. The main area contains a list of SQL statements, numbered 132 to 146. The statements are INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number) VALUES(...). The statements are as follows:

```
132
133. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
134 VALUES(5, 5, 'or0101250', 'im01101046');
135
136. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
137 VALUES(5, 5, 'or0101350', 'im01101044');
138
139. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
140 VALUES(18, 18, 'or0101425', 'im01101047');
141
142. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
143 VALUES(10, 10, 'or0101681', 'im01101047');
144
145. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
146 VALUES(1, 1, 'or0101750', 'im01101048');
```

Below the statements, there is a section labeled 'SQL Statement Output'.

Yes, it matched.

3. Run the “sports data.sql” script in APEX to populate your tables



The screenshot shows the Oracle Live SQL interface. The top bar includes a 'Feedback' icon, a 'Help' icon, a user profile 'hassan.adam@graduate.utm.my', and a 'Run' button. Below the top bar, there are buttons for 'Clear', 'Find', 'Actions', 'Save', and 'Run'. The main area contains a list of SQL statements, numbered 132 to 146. The statements are INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number) VALUES(...). The statements are as follows:

```
132
133. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
134 VALUES(5, 5, 'or0101250', 'im01101046');
135
136. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
137 VALUES(5, 5, 'or0101350', 'im01101044');
138
139. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
140 VALUES(18, 18, 'or0101425', 'im01101047');
141
142. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
143 VALUES(10, 10, 'or0101681', 'im01101047');
144
145. INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
146 VALUES(1, 1, 'or0101750', 'im01101048');
```

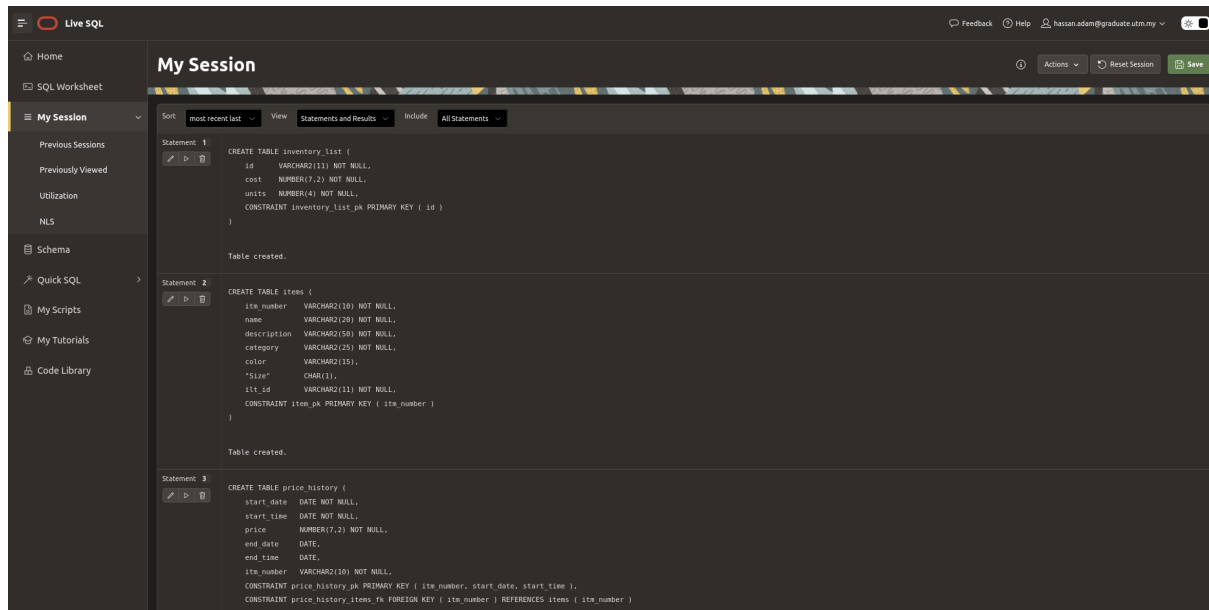
Below the statements, there is a section labeled 'SQL Statement Output'.

The sidebar on the left contains the following navigation options:

- Home
- SQL Worksheet
- My Session
- Schema
- Quick SQL
- My Scripts
- My Tutorials
- Code Library

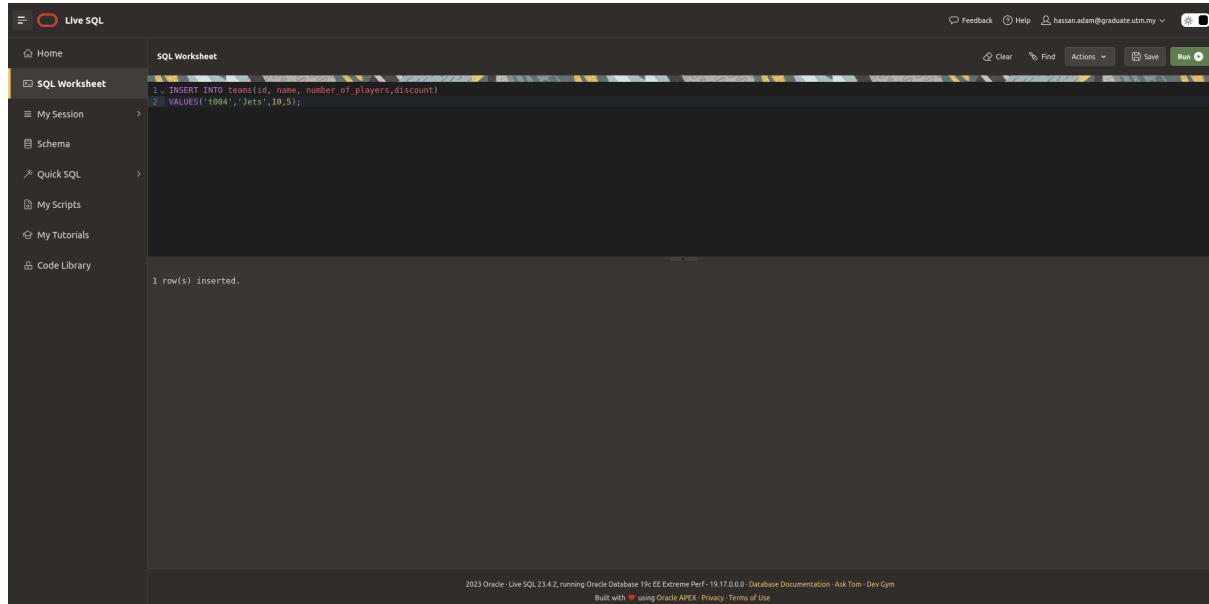
At the bottom of the interface, there is a footer that reads: '2023 Oracle - Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use'.

4. Check that no errors occurred when you ran the script

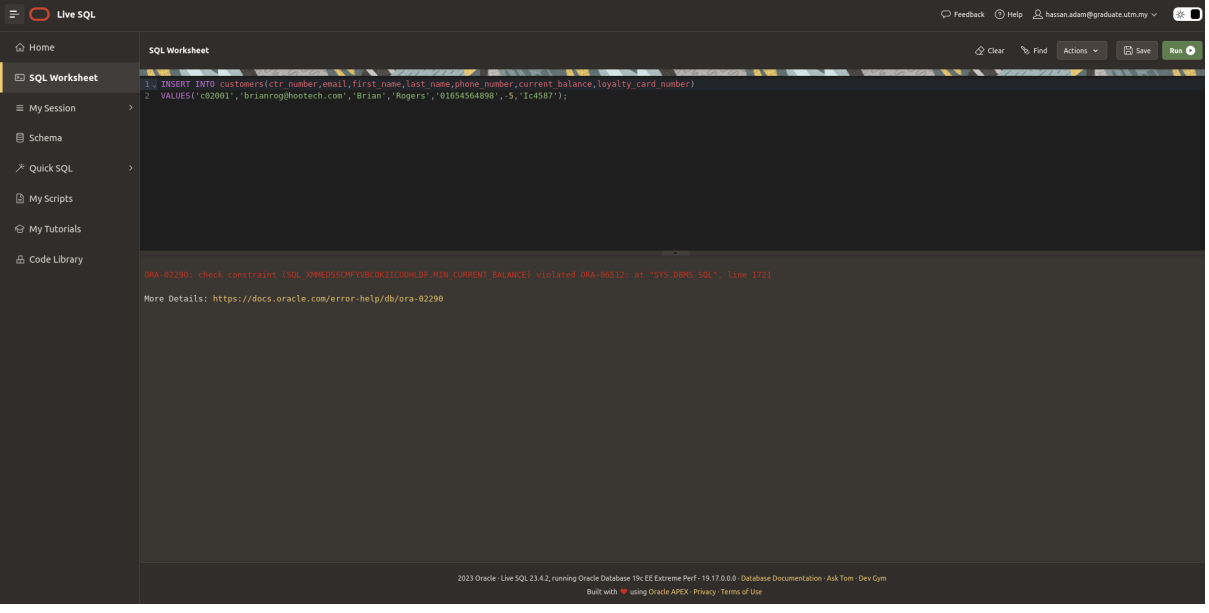


Part 2: Inserting rows to the system

1. Add a new team to the system



2. Add a new Customer with the following details to the system



The screenshot shows the Live SQL interface. On the left is a sidebar with navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains two lines of SQL code:

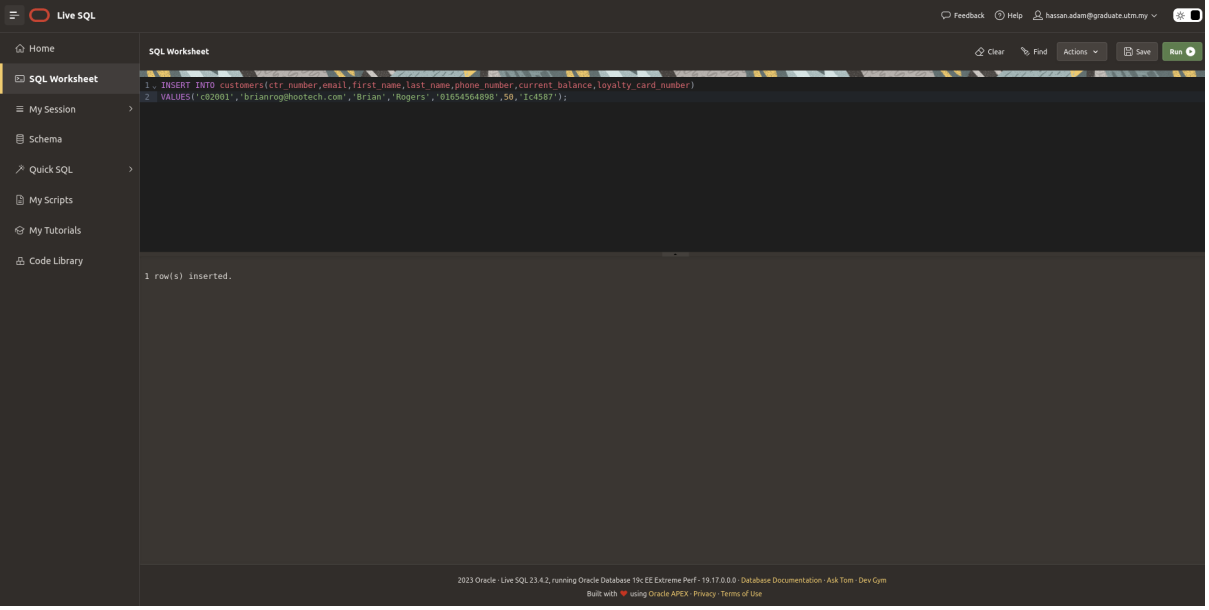
```
1 INSERT INTO customers(ctr_number,email,first_name,last_name,phone_number,current_balance,loyalty_card_number)
2 VALUES('c02901','brianrog@hooitech.com','Brian','Rogers','01654564898',-5,'1c4587');
```

Below the code, an error message is displayed in red text:

```
ORA-02290: check constraint (SQL_MPREDSCHYVBCOR11CONVLDIF_MIN(CURRENT_BALANCE)) violated ORA-06512: at "SYS.DBMS_SQL", line 1721
More Details: https://docs.oracle.com/error-help/db/ora-02290
```

At the bottom of the interface, a footer line reads: '2023 Oracle - Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym'.

3. This information violates the check constraint that the current balance must not be less than zero. Change the current balance to 50 and rerun the query.



The screenshot shows the Live SQL interface after the query has been rerun. The SQL Worksheet now contains the same code as before, but with the current balance changed to 50:

```
1 INSERT INTO customers(ctr_number,email,first_name,last_name,phone_number,current_balance,loyalty_card_number)
2 VALUES('c02901','brianrog@hooitech.com','Brian','Rogers','01654564898',50,'1c4587');
```

Below the code, the execution result is displayed:

```
1 row(s) inserted.
```

The footer at the bottom remains the same: '2023 Oracle - Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym'.

Section 6 Lesson 4 Exercise 2: Data Manipulation Language

Part 1- Updating rows to the system

1. Run the following query to view the content of the price_history table:

```
SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date,  
TO_CHAR (end_time, 'HH24:MI')  
FROM price_history;
```

The screenshot shows the Live SQL web application interface. On the left is a sidebar with navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains a text editor with the following SQL query:

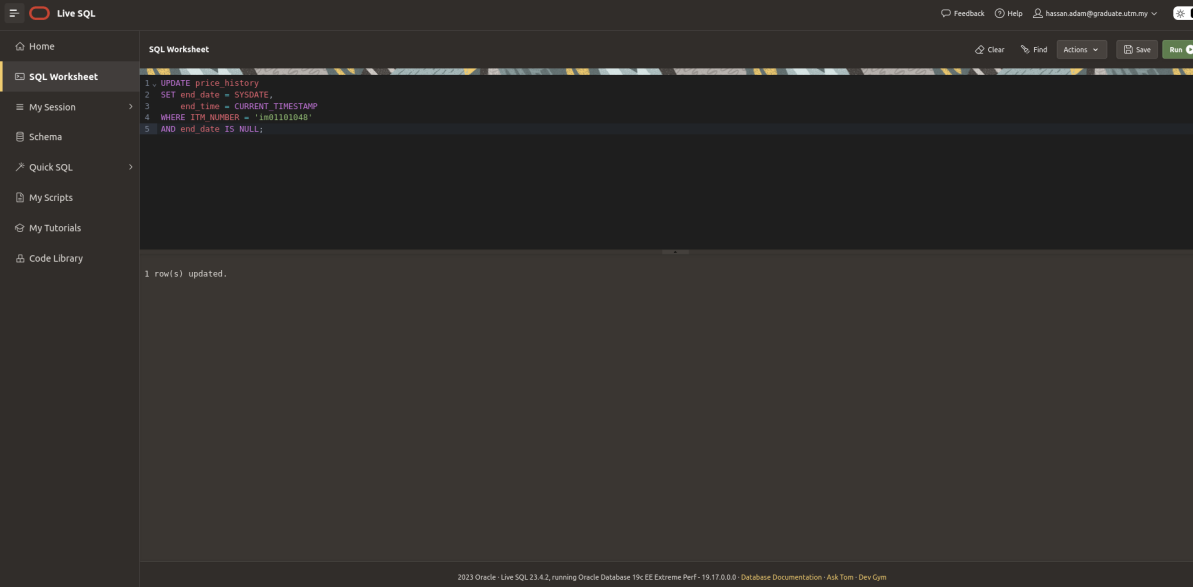
```
1. SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR (end_time, 'HH24:MI')  
2. FROM price_history;
```

Below the editor, the query results are displayed in a table with 5 columns: START_DATE, TO_CHAR(START_TIME, 'HH24:MI:SS'), PRICE, END_DATE, and TO_CHAR(END_TIME, 'HH24:MI'). There are 7 rows of data. Below the table is a 'Download CSV' button and the text '7 rows selected.'.

START_DATE	TO_CHAR(START_TIME, 'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME, 'HH24:MI')
17-JUN-17	09:00:00	4.99	-	-
25-NOV-16	09:00:00	14.99	25-JAN-17	17:00
25-JAN-17	17:01:00	8.99	25-JAN-17	19:00
26-JAN-17	09:00:00	15.99	-	-
12-FEB-17	12:30:00	7.99	-	-
25-APR-17	10:10:10	24.99	-	-
31-MAY-17	16:35:30	149	-	-

At the bottom of the interface, there is a footer with the text: '2021 Oracle - Live SQL 21.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use'.

2. Obl is going to update the price of the premium bat so you will need to write a query that will close off the current price by adding the system date values to the end_date and end_time fields. To run this query you will need to both match the item number and identify that the end date is null. This ensures that you are updating the latest price.



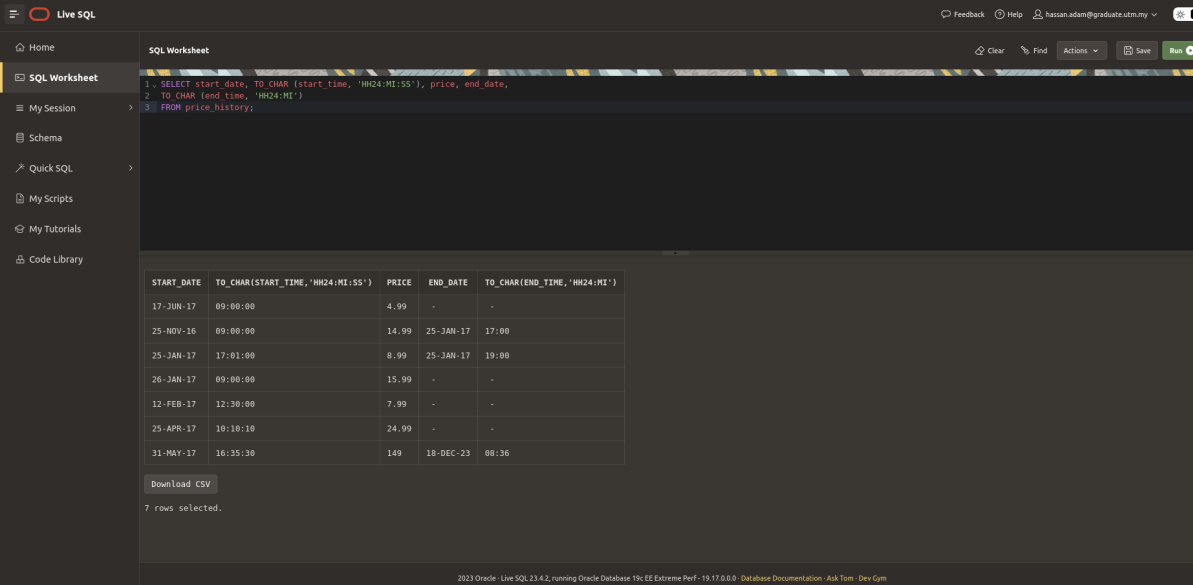
The screenshot shows the Live SQL interface with the following SQL query in the editor:

```
1. UPDATE price_history
2. SET end_date = SYSDATE,
3.     end_time = CURRENT_TIMESTAMP
4. WHERE ITEM_NUMBER = '1401101048'
5. AND end_date IS NULL;
```

The execution result shows: 1 row(s) updated.

Footer text: 2023 Oracle Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

3. rerun the select statement on the price_history table to ensure that the statement has been executed.



The screenshot shows the Live SQL interface with the following SQL query in the editor:

```
1. SELECT start_date, TO_CHAR(start_time, 'HH24:MI:SS'), price, end_date,
2.        TO_CHAR(end_time, 'HH24:MI:')
3. FROM price_history;
```

The execution result displays a table with 7 rows selected:

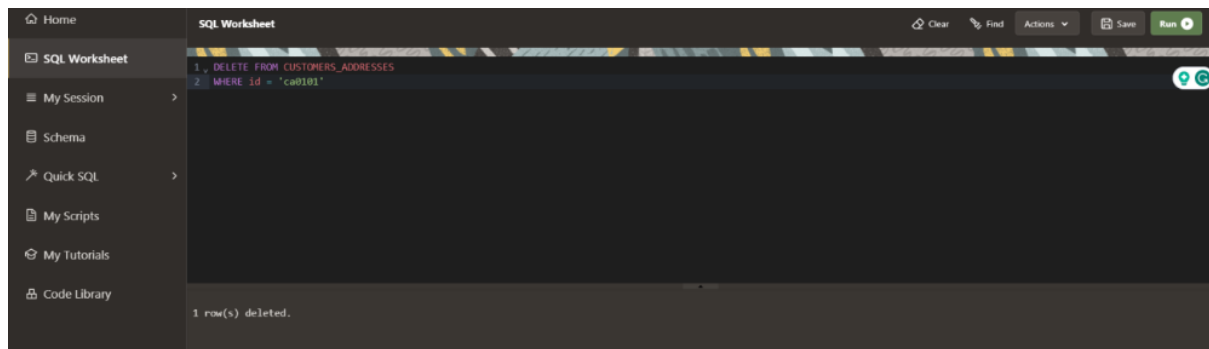
START_DATE	TO_CHAR(START_TIME, 'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME, 'HH24:MI')
17-JUN-17	09:00:00	4.99	-	-
25-NOV-16	09:00:00	14.99	25-JAN-17	17:00
25-JAN-17	17:01:00	8.99	25-JAN-17	19:00
26-JAN-17	09:00:00	15.99	-	-
12-FEB-17	12:30:00	7.99	-	-
25-APR-17	10:10:10	24.99	-	-
31-MAY-17	16:35:30	149	18-DEC-23	08:36

Download CSV
7 rows selected.

Footer text: 2023 Oracle Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

Part 2: Deleting rows from the system

1. Bob Thornberry has contacted Obl to ask that the 83 Barrhill Drive address be removed from the system as he can no longer receive parcels at this address. Write a SQL statement that will remove this address from the system.



2. Run a select statement on the customers_addresses table to ensure that the statement has been executed

