



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECD2523 – 08 Database

Semester I , Session 2023/2024

Lab 2 – DML Part 1 and Part 2

Lecturer:

Dr. Noor Hidayah Zakaria

Name	Matrics Number
Teoh Zhi Yong	A22EC0125

Section 6 Lesson 4 Exercise 1: Data Manipulation Language

Use DML operations to manage database tables (S6L4 Objective 2)

In this exercise you will populate and work with the data that is stored in the database system tables.

Part 1: Reading information from a script

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.

- I. [inventory_list](#)
- II. [items](#)
- III. [price_history](#)
- IV. [sales_representatives](#)
- V. [sales_rep_addresses](#)
- VI. [teams](#)
- VII. [customers](#)
- VIII. [customers_addresses](#)
- IX. [orders](#)
- X. [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.

[Yes, the list is matched.](#)

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

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

1	0.06	INSERT INTO inventory_list (id, cost, units) VALUES(101023)	1 row(s) inserted.	1
2	0.00	INSERT INTO inventory_list (id, cost, units) VALUES(101023)	1 row(s) inserted.	1
3	0.00	INSERT INTO inventory_list (id, cost, units) VALUES(101023)	1 row(s) inserted.	1
4	0.00	INSERT INTO inventory_list (id, cost, units) VALUES(101023)	1 row(s) inserted.	1
5	0.00	INSERT INTO inventory_list (id, cost, units) VALUES(101023)	1 row(s) inserted.	1
6	0.03	INSERT INTO items (item_number, name, description, category,	1 row(s) inserted.	1
7	0.01	INSERT INTO items (item_number, name, description, category,	1 row(s) inserted.	1
8	0.00	INSERT INTO items (item_number, name, description, category,	1 row(s) inserted.	1
9	0.00	INSERT INTO items (item_number, name, description, category,	1 row(s) inserted.	1
10	0.00	INSERT INTO items (item_number, name, description, category,	1 row(s) inserted.	1
11	0.02	INSERT INTO price_history (start_date, start_time, price, it	1 row(s) inserted.	1
12	0.00	INSERT INTO price_history (start_date, start_time, price, en	1 row(s) inserted.	1
13	0.00	INSERT INTO price_history (start_date, start_time, price, en	1 row(s) inserted.	1
14	0.00	INSERT INTO price_history (start_date, start_time, price, it	1 row(s) inserted.	1
15	0.00	INSERT INTO price_history (start_date, start_time, price, it	1 row(s) inserted.	1

Download

row(s) 1 - 15 of 47 [Next](#)

47

Statements Processed

47

Successful

0

With Errors

[ashyong@graduateudemmy](#) [Tablespace](#) [en](#)

Copyright © 1999, 2023, Oracle and/or its affiliates.

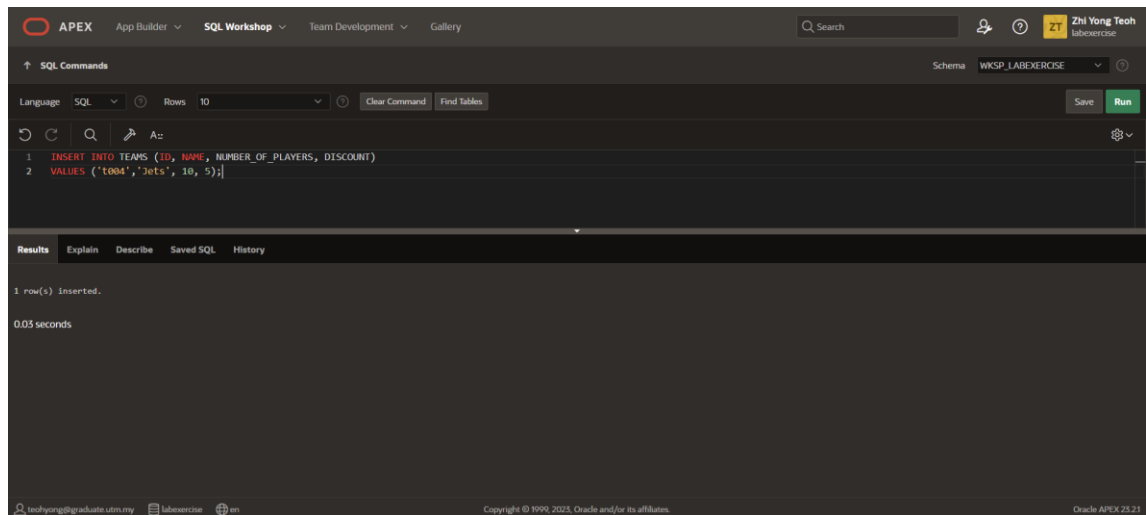
Oracle APEX 23.23

Part 2- Inserting rows to the system

1. Add a new team to the system

id	name	Number_of_players	discount
t004	Jets	10	5

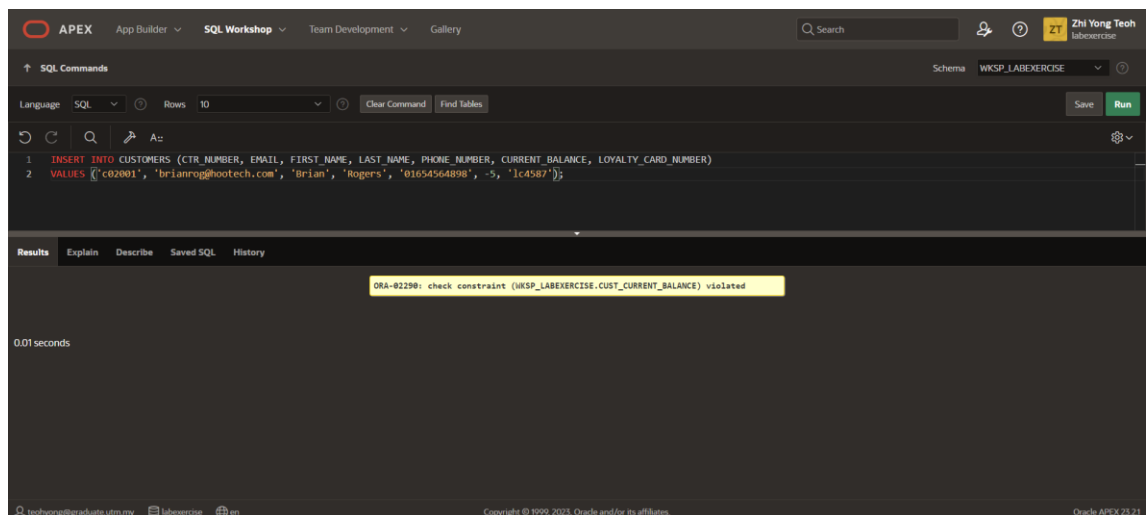
```
INSERT INTO TEAMS (ID, NAME, NUMBER_OF_PLAYERS, DISCOUNT)
VALUES ('t004','Jets', 10, 5);
```



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

ctr number	email	First name	Last name	Phone number	Current balance	Loyalty card number	tem id	sre id
c02001	brianrog@hoootech.com	Brian	Rogers	01654564898	-5	lc4587		

```
INSERT INTO CUSTOMERS (CTR_NUMBER, EMAIL, FIRST_NAME, LAST_NAME,
PHONE_NUMBER, CURRENT_BALANCE, LOYALTY_CARD_NUMBER)
VALUES ('c02001', 'brianrog@hoootech.com', 'Brian', 'Rogers', '01654564898', -5, 'lc4587');
```



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.

```
INSERT INTO CUSTOMERS (CTR_NUMBER, EMAIL, FIRST_NAME, LAST_NAME,  
PHONE_NUMBER, CURRENT_BALANCE, LOYALTY_CARD_NUMBER)  
VALUES ('c02001', 'brianrog@hootech.com', 'Brian', 'Rogers', '01654564898', 50, 'lc4587');
```

The screenshot shows the APEX SQL Workshop interface. The 'SQL Commands' tab is active, displaying the following SQL statement:

```
1 INSERT INTO CUSTOMERS (CTR_NUMBER, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, CURRENT_BALANCE, LOYALTY_CARD_NUMBER)  
2 VALUES ('c02001', 'brianrog@hootech.com', 'Brian', 'Rogers', '01654564898', 50, 'lc4587');
```

The 'Results' tab is selected, showing the output of the query:

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

The interface includes a search bar, a schema dropdown set to 'WKSP_LABEXERCISE', and a 'Run' button.

The screenshot shows the APEX Object Browser interface. The 'CUSTOMERS' table is selected, and the 'Data' tab is active, displaying the following data:

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUM
c00001	bob.thornberry@...	Robert	Thornberry	01234567898	150	sr01	td01	
c00012	j.jones@freemail...	Jennifer	Jones	01505214598	0			lc1015
c00101	unknown@here.c...	John	Doe	03216547808	9875	sr01	td02	
c00103	MurciaA@globalt...	Andrew	Murcia	0775246890	85			lc2541
c07986	margali7@delphi...	Maria	Galant	01442736589	125.65	sr03	td03	
c02001	brianrog@hootec...	Brian	Rogers	01654564898	50			lc4587

The interface includes a search bar, a schema dropdown set to 'WKSP_LABEXERCISE', and a 'Run' button.

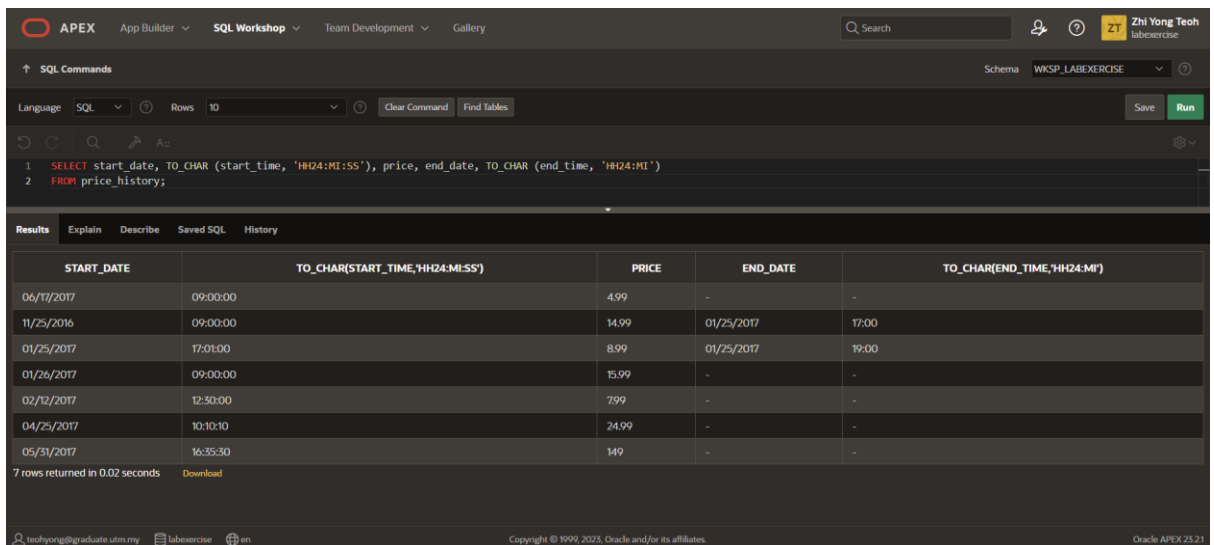
Section 6 Lesson 4 Exercise 1: Data Manipulation Language

Use DML operations to manage database tables (S6L4 Objective 2)

In this exercise you will populate and work with the data that is stored in the database system tables.

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 APEX SQL Workshop interface. The SQL Commands pane contains the query: `SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR (end_time, 'HH24:MI')` `FROM price_history;`. The Results pane shows 7 rows of data.

START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
06/11/2017	09:00:00	4.99	-	-
11/25/2016	09:00:00	14.99	01/25/2017	17:00
01/25/2017	17:01:00	8.99	01/25/2017	19:00
01/26/2017	09:00:00	15.99	-	-
02/12/2017	12:30:00	7.99	-	-
04/25/2017	10:10:10	24.99	-	-
05/31/2017	16:35:30	149	-	-

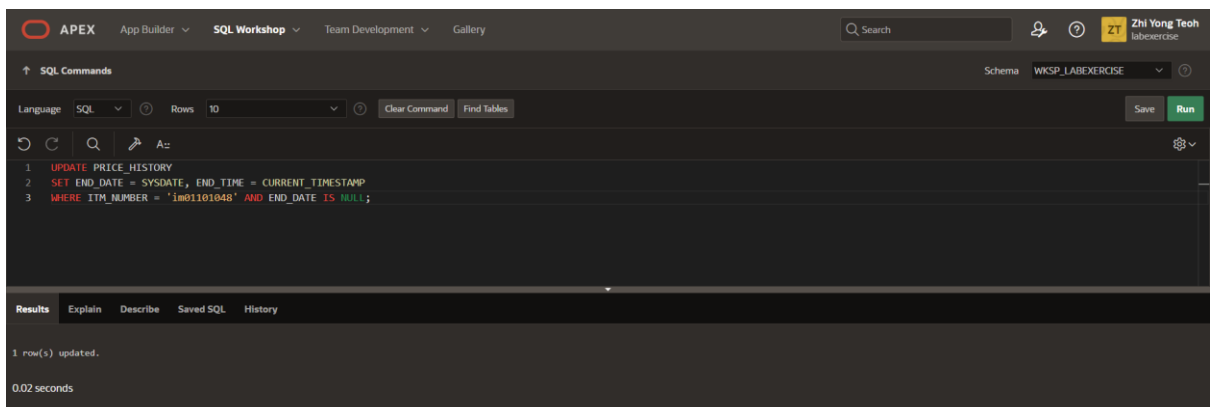
7 rows returned in 0.02 seconds

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.

UPDATE PRICE_HISTORY

SET END_DATE = SYSDATE, END_TIME = CURRENT_TIMESTAMP

WHERE ITM_NUMBER = 'im01101048' AND END_DATE IS NULL;



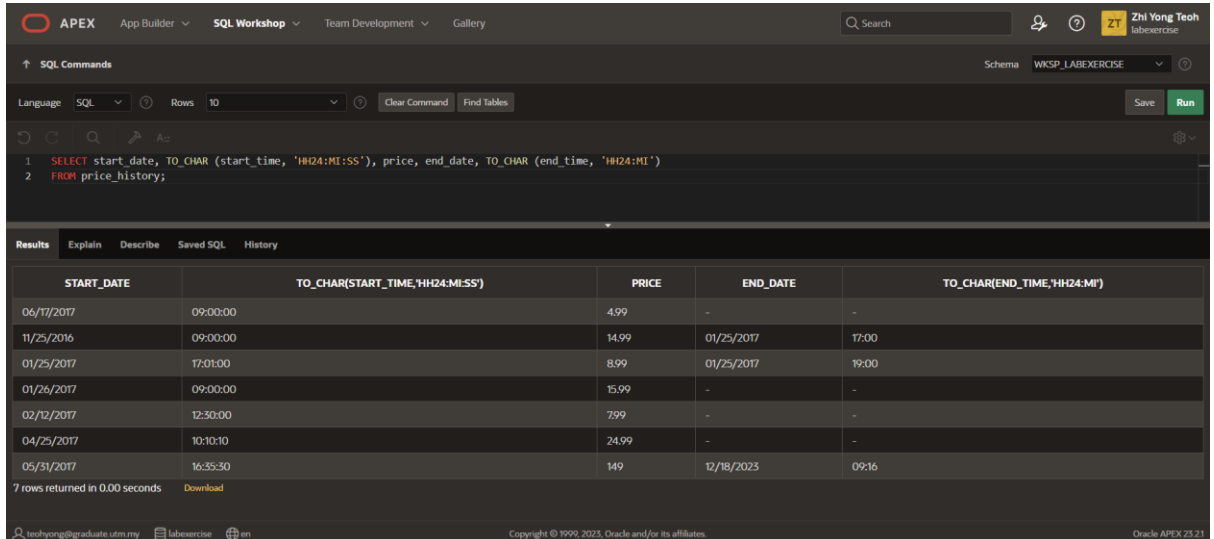
The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the query: `UPDATE PRICE_HISTORY` `SET END_DATE = SYSDATE, END_TIME = CURRENT_TIMESTAMP` `WHERE ITM_NUMBER = 'im01101048' AND END_DATE IS NULL;`. The Results pane shows 1 row(s) updated.

Results
1 row(s) updated.

0.02 seconds

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

```
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 APEX SQL Workshop interface. The SQL Commands pane contains the following 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;
```

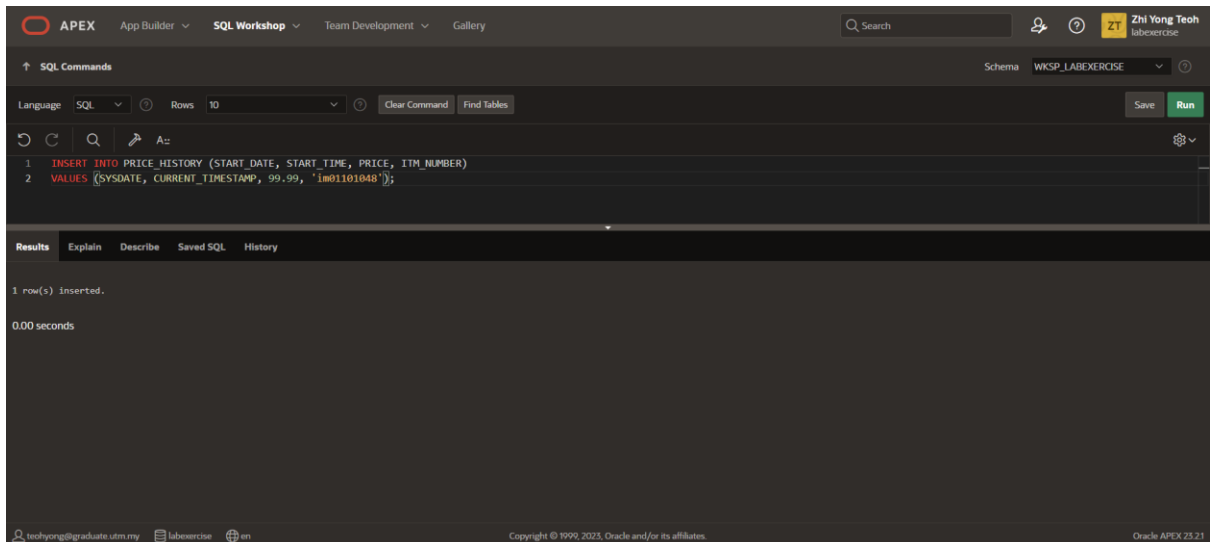
The Results pane shows the output of the query, which is a table with 5 columns: START_DATE, TO_CHAR(START_TIME,'HH24:MI:SS'), PRICE, END_DATE, and TO_CHAR(END_TIME,'HH24:MI'). The table contains 7 rows of data.

START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
06/11/2017	09:00:00	4.99	-	-
11/25/2016	09:00:00	14.99	01/25/2017	17:00
01/25/2017	17:01:00	8.99	01/25/2017	19:00
01/26/2017	09:00:00	15.99	-	-
02/12/2017	12:30:00	7.99	-	-
04/25/2017	10:10:10	24.99	-	-
05/31/2017	16:35:30	149	12/18/2023	09:16

7 rows returned in 0.00 seconds

4. Insert a new row that will use the current date and time to set the new price of the premium bat to be 99.99.

```
INSERT INTO PRICE_HISTORY (START_DATE, START_TIME, PRICE, ITM_NUMBER)  
VALUES (SYSDATE, CURRENT_TIMESTAMP, 99.99, 'im01101048');
```



The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following query:

```
1 INSERT INTO PRICE_HISTORY (START_DATE, START_TIME, PRICE, ITM_NUMBER)  
2 VALUES (SYSDATE, CURRENT_TIMESTAMP, 99.99, 'im01101048');
```

The Results pane shows the output of the query, which is a single row indicating that 1 row(s) were inserted. The execution time is 0.00 seconds.

Results
1 row(s) inserted.
0.00 seconds

5. Rerun the select statement on the price_history table to ensure that the statement has been executed.

```
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 Oracle APEX SQL Workshop interface. The SQL command window contains the following 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;
```

The Results tab is active, displaying a table with 8 rows. The columns are: START_DATE, TO_CHAR(START_TIME, 'HH24:MI:SS'), PRICE, END_DATE, and TO_CHAR(END_TIME, 'HH24:MI').

START_DATE	TO_CHAR(START_TIME, 'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME, 'HH24:MI')
06/17/2017	09:00:00	4.99	-	-
11/25/2016	09:00:00	14.99	01/25/2017	17:00
01/25/2017	17:01:00	8.99	01/25/2017	19:00
01/26/2017	09:00:00	15.99	-	-
02/12/2017	12:30:00	7.99	-	-
04/25/2017	10:10:10	24.99	-	-
05/31/2017	16:35:30	149	12/18/2023	09:16
12/18/2023	09:21:44	99.99	-	-

8 rows returned in 0.01 seconds. A Download button is visible below the results.

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.

```
DELETE FROM CUSTOMERS_ADDRESSES  
WHERE ADDRESS_LINE_1 = '83 Barrhill Drive'
```

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command window contains the following query:

```
1 DELETE FROM CUSTOMERS_ADDRESSES  
2 WHERE ADDRESS_LINE_1 = '83 Barrhill Drive'
```

The Results tab is active, displaying the message: 1 row(s) deleted. 0.01 seconds.

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

`SELECT * FROM CUSTOMERS_ADDRESSES;`

APEX App Builder SQL Workshop Team Development Gallery

Search

Schema WKSP_LABEXERCISE

Language SQL Rows 10 Clear Command Find Tables Save Run

1 `SELECT * FROM CUSTOMERS_ADDRESSES;`

Results Explain Describe Saved SQL History

ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE	CTR_NUMBER
ca0102	17 Gartsquare Road	Starford	Liverpool	LP89JHK	c00001
ca0103	54 Ropehill Crescent	Georgetown	Star	ST45AGV	c00101
ca0104	36 Watercress Lane	-	Jump	JP25YTH	c01986
ca0105	63 Acacia Drive	Skins	Liverpool	LP83JHR	c00001

4 rows returned in 0.03 seconds Download

teohyong@graduate.utm.my labexercise en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.21