



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**FACULTY OF COMPUTING**  
UTM Johor Bahru

**SECD2523**  
**SECTION 10**

**LECTURER: ROZILAWATI BINTI DOLLAH @ MD ZAIN**

**LAB 2**

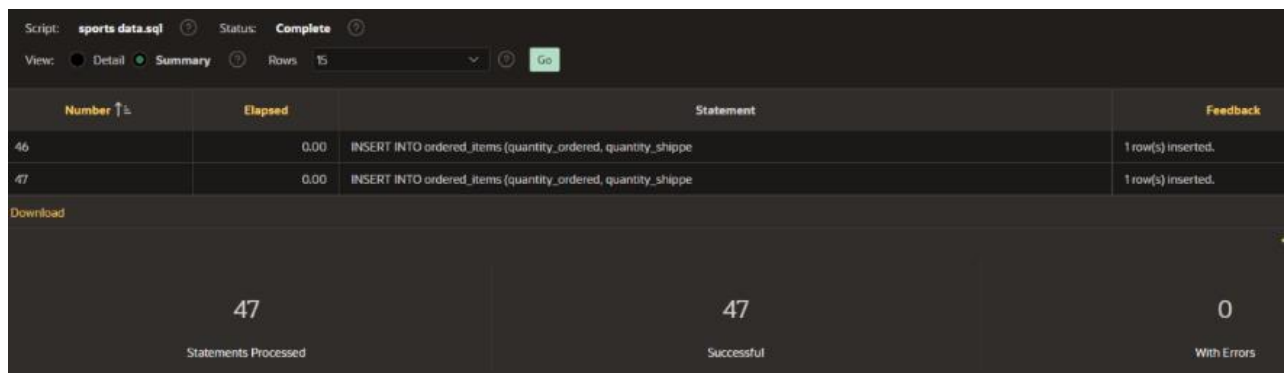
**Name: ENG JUN XIANG**  
**Matric No: A22EC0049**

## DML1 PART 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.
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.
3. Run the “sports data.sql” script in APEX to populate your tables
4. Check that no errors occurred when you ran the script



The screenshot shows the APEX SQL script execution interface. At the top, the script name is 'sports data.sql' and the status is 'Complete'. Below this, there are tabs for 'Detail' and 'Summary', and a 'Go' button. The main table displays the execution results for two statements. The first statement is an INSERT INTO ordered\_items (quantity\_ordered, quantity\_shippe) and the second is an INSERT INTO ordered\_items (quantity\_ordered, quantity\_shippe). Both statements were successful, with 1 row(s) inserted each. At the bottom, a summary bar shows 47 statements processed, 47 successful, and 0 with errors.

Number ↑	Elapsed	Statement	Feedback
46	0.00	INSERT INTO ordered_items (quantity_ordered, quantity_shippe	1 row(s) inserted.
47	0.00	INSERT INTO ordered_items (quantity_ordered, quantity_shippe	1 row(s) inserted.





Download

47	47	0
Statements Processed	Successful	With Errors







No error.

## Part 2- Inserting rows to the system

1. Add a new team to the system

EDIT	ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT
	t001	Rockets	25	10
	t002	Celtics	42	20
	t003	Rovers	8	-
	t004	Jets	10	5

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







EDIT	CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
	c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-
	c00012	J.jones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015
	c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-
	c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-	lc2341
	c01986	margali7@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
	c02001	brianrog@hootech.com	Brian	Rogers	01654564898	-5	-	-	lc4587

3. Change the current balance to 50 and rerun the query.

```
1 UPDATE customers
2 SET current_balance = 50
3 WHERE ctr_number = 'c02001'
```

Results Explain Describe Saved SQL History

1 row(s) updated.

Query	Count Rows	Insert Row	Load Data						
EDIT	CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
	c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-
	c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015
	c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-
	c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-	lc2341
	c01986	margali7@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
	c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587

## DML2 PART 2

### 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;
```

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	-	-

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.

Columns

Data

Indexes

Constraints

Grants

Statistics

Triggers

Dependencies

DDL

Sample Queries

+ Insert Row

Columns...

Filter...

Count Rows

Load Data

Download

Refresh

	START_DATE	START_TIME	PRICE	END_DATE	END_TIME	ITEM_NUMBER
	06/17/2017	06/17/2016	4.99			im01101044
	11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017	im01101045
	01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017	im01101045
	01/26/2017	01/26/2017	15.99			im01101045
	02/12/2017	02/12/2017	7.99			im01101046
	04/25/2017	04/25/2017	24.99			im01101047
	05/31/2017	05/31/2017	149	12/18/2023	12/18/2023	im01101048

3. Rerun the select statement on the price\_history table to ensure that the statement has been executed.

Results

Explain

Describe

Saved SQL

History

START_DATE	START_TIME	PRICE	END_DATE	END_TIME	ITM_NUMBER
06/17/2017	06/17/2016	4.99	-	-	im01101044
11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017	im01101045
01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017	im01101045
01/26/2017	01/26/2017	15.99	-	-	im01101045
02/12/2017	02/12/2017	7.99	-	-	im01101046
04/25/2017	04/25/2017	24.99	-	-	im01101047
05/31/2017	05/31/2017	149	12/18/2023	12/18/2023	im01101048

7 rows returned in 0.01 seconds

Download

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.

```
1 INSERT INTO price_history (itm_number, price, start_date, start_time)
2 VALUES ('im01101048', 99.99, CURRENT_DATE, CURRENT_TIMESTAMP);
```

Results	Explain	Describe	Saved SQL	History
1 row(s) inserted.				

5. Rerun the select statement on the price\_history table to ensure that the statement has been executed

Results	Explain	Describe	Saved SQL	History	
START_DATE	START_TIME	PRICE	END_DATE	END_TIME	ITEM
12/18/2023	12/18/2023	99.99	-	-	im01101
06/17/2017	06/17/2016	4.99	-	-	im01101
11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017	im01101
01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017	im01101
01/26/2017	01/26/2017	15.99	-	-	im01101
02/12/2017	02/12/2017	7.99	-	-	im01101
04/25/2017	04/25/2017	24.99	-	-	im01101
05/31/2017	05/31/2017	149	12/18/2023	12/18/2023	im01101
8 rows returned in 0.01 seconds <a href="#">Download</a>					

## 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 longer receive parcels at this address. Write a SQL statement that will remove this address from the system.

1	DELETE FROM customers		
2	WHERE id = 'ca0101';		
Results	Explain	Describe	Save
1 row(s) deleted.			

2. Run a select statement on the customers\_addresses table to ensure that the statement has been executed

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	JP23YTH	c01986					
ca0105	63 Acacia Drive	Skins	Liverpool	LP83JHR	c00001					