



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SECD2523 – DATABASE

SEMESTER 1/20232024

SECTION 08

LAB 2: DML1

LECTURER: DR. NOOR HIDAYAH BINTI ZAKARIA

NAME	MATRIC NO.
EILEEN YONG KAI QIN	A22EC0048

Exercise 1: Data Manipulation Language

Part 1: Running a script to populate the tables.

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

INVENTORY_LIST

ITEMS

PRICE_HISTORY

SALES_REPRESENTATIVES

SALES_REP_ADDRESSES

TEAMS

CUSTOMERS

CUSTOMERS_ADDRESSES

ORDERS

ORDERED_ITEMS

2. Open the “sports data.sql” and look at the order the data is being added there, does your list match?

Yes, the list is match.

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

There are no errors

Part 2- Inserting rows to the system

1. Add a new team to the system

```
SELECT * FROM teams;
```

Results	Explain	Describe	Saved SQL	History
ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT	
t002	Celtics	42	20	
t001	Rockets	25	10	
t003	Rovers	8	-	
3 rows returned in 0.03 seconds				Download

```
INSERT INTO teams (id, name, number_of_players, discount)
VALUES ('t004', 'Jets', 10, 5);
```

```
SELECT * FROM teams;
```

Results

Explain

Describe

Saved SQL

History

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

4 rows returned in 0.00 seconds

Download

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

```
SELECT * FROM customers;
```

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-

5 rows returned in 0.00 seconds [Download](#)

```
INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number,  
current_balance, sre_id, tem_id, loyalty_card_number)
```

```
VALUES ('c02001', 'brianrog@hoootech.com', 'Brian', 'Rogers', '01654564898', -5,  
'lc458');
```

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.

```
UPDATE customers
```

```
SET current_balance = 50
```

```
WHERE ctr_number = 'c02001';
```

```
SELECT * FROM customers;
```

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-
c02001	brianrog@hoootech.com	Brian	Rogers	01654564898	50	-	-	lc458
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-

6 rows returned in 0.00 seconds [Download](#)

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;
```

START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
11/25/2016	09:00:00	14.99	01/25/2017	17:00
05/31/2017	16:55:30	149	-	-
06/11/2017	09:00:00	4.99	-	-
01/25/2017	17:01:00	8.99	01/25/2017	19:00
04/25/2017	10:10:10	24.99	-	-
01/26/2017	09:00:00	15.99	-	-
02/12/2017	12:30:00	799	-	-

7 rows returned in 0.03 seconds [Download](#)

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 = CURRENT_DATE, end_time = CURRENT_TIMESTAMP

WHERE itm_number = 'im01101048' AND end_time IS NULL;

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

SELECT * FROM price_history;

START_DATE	START_TIME	PRICE	END_DATE	END_TIME	ITM_NUMBER
11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017	im01101045
05/31/2017	05/31/2017	149	12/10/2023	12/10/2023	im01101048
06/11/2017	06/11/2016	4.99	-	-	im01101044
01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017	im01101045
04/25/2017	04/25/2017	24.99	-	-	im01101047
01/26/2017	01/26/2017	15.99	-	-	im01101045
02/12/2017	02/12/2017	799	-	-	im01101046

7 rows returned in 0.00 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.

INSERT INTO price_history (start_date, start_time, price, itm_number)

VALUES (CURRENT_DATE, CURRENT_TIMESTAMP, 99.99, 'im01101048');

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

```
SELECT * FROM price_history;
```

START_DATE	START_TIME	PRICE	END_DATE	END_TIME	ITM_NUMBER
11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017	im01101045
05/31/2017	05/31/2017	149	12/10/2023	12/10/2023	im01101048
06/17/2017	06/17/2016	4.99	-	-	im01101044
01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017	im01101045
04/25/2017	04/25/2017	24.99	-	-	im01101047
12/10/2023	12/10/2023	99.99	-	-	im01101048
01/26/2017	01/26/2017	15.99	-	-	im01101045
02/12/2017	02/12/2017	79.9	-	-	im01101046

8 rows returned in 0.01 seconds [Download](#)

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.

```
DELETE FROM customers_addresses
```

```
WHERE address_line_1 = '83 Barrhill Drive';
```

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

```
SELECT * FROM customers_addresses;
```

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

4 rows returned in 0.03 seconds [Download](#)