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 : 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.

1) inventory_list
items

price_history

Sales_representatives

sales_rep_address

teams

customers

customers_addresses

orders

ordered_items

2) Yes, it is matched

3) Yes, it can successfully run

4) No, there are no errors.

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

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@hootech.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.

① INSERT INTO teams(id, name, number_of_players, discount)
VALUES('t004', 'Jets', 10, 5);

② INSEDT INTO customers(ctr_number, email, first_name, last_name, phone_number,
current_balance, loyalty_card_number)
VALUES('c02001', 'brianrog@hootech.com', 'Brian', 'Rogers', '01654564898',
-5, 'lc4587');

③ INSEDT 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');

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;
```

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.

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

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.

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

(1) Output:

	START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
1	17-JUN-17	09:00:00	4.99	(null)	(null)
2	25-NOV-16	09:00:00	14.99	25-JAN-17	17:00
3	25-JAN-17	17:01:00	8.99	25-JAN-17	19:00
4	26-JAN-17	09:00:00	15.99	(null)	(null)
5	12-FEB-17	12:30:00	7.99	(null)	(null)
6	25-APR-17	10:10:10	24.99	(null)	(null)
7	31-MAY-17	16:35:30	149	(null)	(null)

(2)
```
UPDATE price_history
SET end_date = current_date
WHERE itm_number = 'im0110104g' AND end_date IS NULL;

UPDATE price_history
SET end_time = current_timestamp;
WHERE itm_number = 'im0110104g';
```

(3)

	START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
2	25/11/2016	09:00:00	14.99	25/01/2017	17:00
3	25/01/2017	17:01:00	8.99	25/01/2017	19:00
4	26/01/2017	09:00:00	15.99	(null)	(null)
5	12/02/2017	12:30:00	7.99	(null)	(null)
6	25/04/2017	10:10:10	24.99	(null)	(null)
7	31/05/2017	16:35:30	149	09/11/2023	11:29

(4)
```
INSERT INTO price_history (start-date, start_time_price, itm_number)
VALUES( current-date, current_timestamp, 99.99 , 'im0110104g');
```

(5)

	START_DATE	TO_CHAR(START_TIME,'HH24:M...	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
3	25/01/2017	17:01:00	8.99	25/01/2017	19:00
4	26/01/2017	09:00:00	15.99	(null)	(null)
5	12/02/2017	12:30:00	7.99	(null)	(null)
6	25/04/2017	10:10:10	24.99	(null)	(null)
7	31/05/2017	16:35:30	149	09/11/2023	11:29
8	09/11/2023	11:41:33	99.99	(null)	(null)

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.

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

① DELETE FROM customers_addresses
WHERE address_line_1 = '83 Barhill Drive'.

② SELECT * FROM customers_addresses

	ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE	CTR_NUMBER
1	ca0102	17 Gartsquare Road	Starford	Liverpool	LP89JHK	c00001
2	ca0103	54 Ropehill Crescent	Georgetown	Star	ST45AGV	c00101
3	ca0104	36 Watercress Lane	(null)	Jump	JP23YTH	c01986
4	ca0105	63 Acacia Drive	Skins	Liverpool	LP83JHR	c00001