FACULTY OF COMPUTING
UTM Johor Bahru

LAB 2 DML 1

Name	Yap Jun Cheng
Matrix Number	A22EC0294
Course Code	SEDC2523
Course Name	Database
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

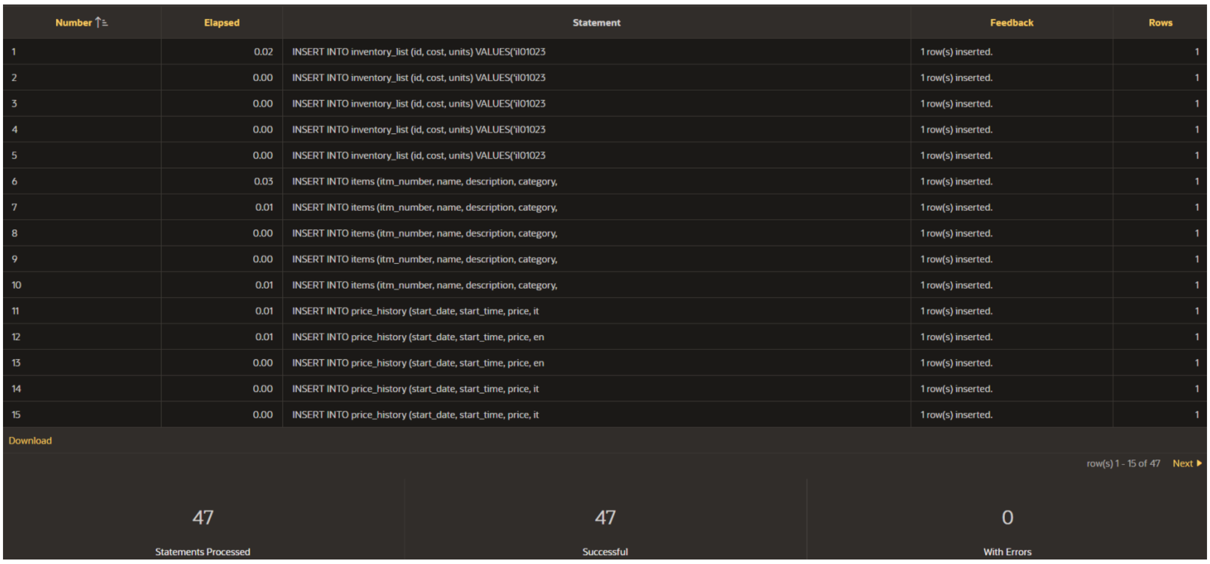
   i) orders

   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.

Yes. The list is matched.

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



Number	Elapsed	Statement	Feedback	Rows
1	0.02	INSERT INTO inventory_list (id, cost, units) VALUES('il01023	1 row(s) inserted.	1
2	0.00	INSERT INTO inventory_list (id, cost, units) VALUES('il01023	1 row(s) inserted.	1
3	0.00	INSERT INTO inventory_list (id, cost, units) VALUES('il01023	1 row(s) inserted.	1
4	0.00	INSERT INTO inventory_list (id, cost, units) VALUES('il01023	1 row(s) inserted.	1
5	0.00	INSERT INTO inventory_list (id, cost, units) VALUES('il01023	1 row(s) inserted.	1
6	0.03	INSERT INTO items (itm_number, name, description, category,	1 row(s) inserted.	1
7	0.01	INSERT INTO items (itm_number, name, description, category,	1 row(s) inserted.	1
8	0.00	INSERT INTO items (itm_number, name, description, category,	1 row(s) inserted.	1
9	0.00	INSERT INTO items (itm_number, name, description, category,	1 row(s) inserted.	1
10	0.01	INSERT INTO items (itm_number, name, description, category,	1 row(s) inserted.	1
11	0.01	INSERT INTO price_history (start_date, start_time, price, it	1 row(s) inserted.	1
12	0.01	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

row(s) 1 - 15 of 47

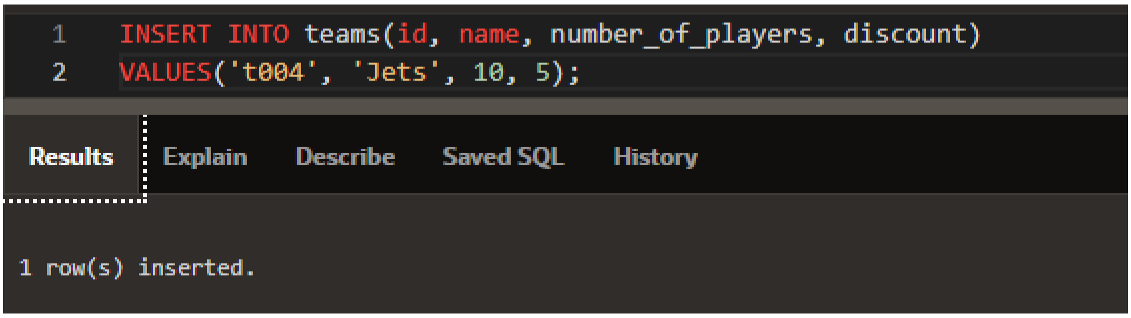
47 Statements Processed

47 Successful

0 With Errors

Part 2- Inserting rows to the system

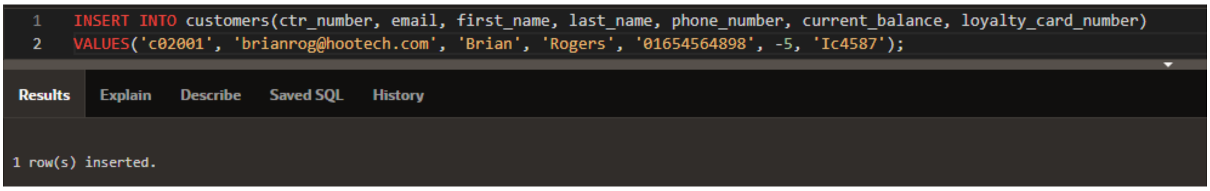
1. Add a new team to the system



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

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

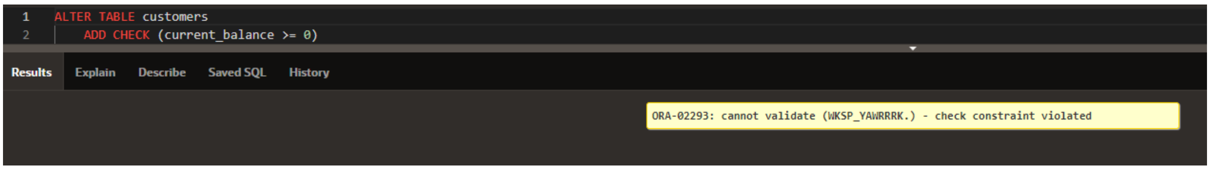
2. Add a new Customer with the following details to the system.



```
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', -5, 'Ic4587');
```

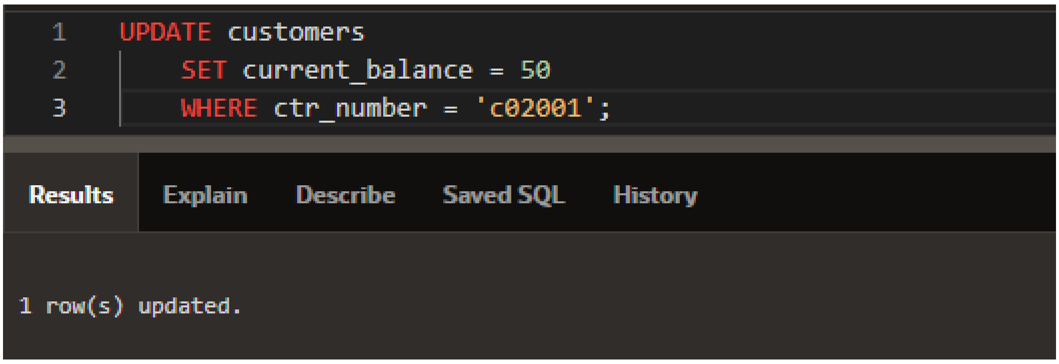
```
1 row(s) inserted.
```

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.



```
ALTER TABLE customers
  ADD CHECK (current_balance >= 0)
```

```
ORA-02293: cannot validate (WKSP_YAWRRRK.) - check constraint violated
```



```sql
UPDATE customers
    SET current_balance = 50
    WHERE ctr_number = 'c02001';
```

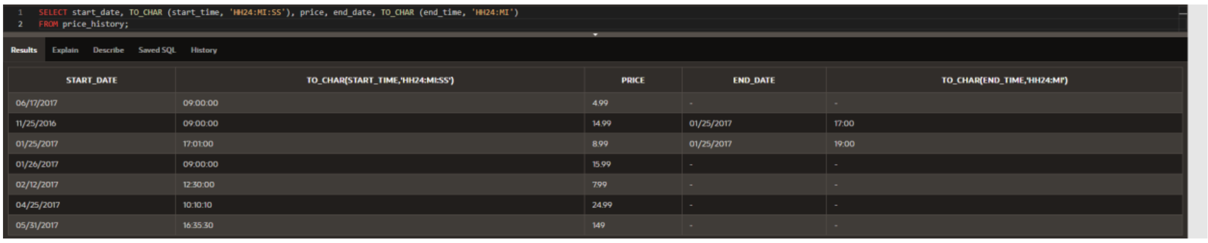
```
1 row(s) updated.
```

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:

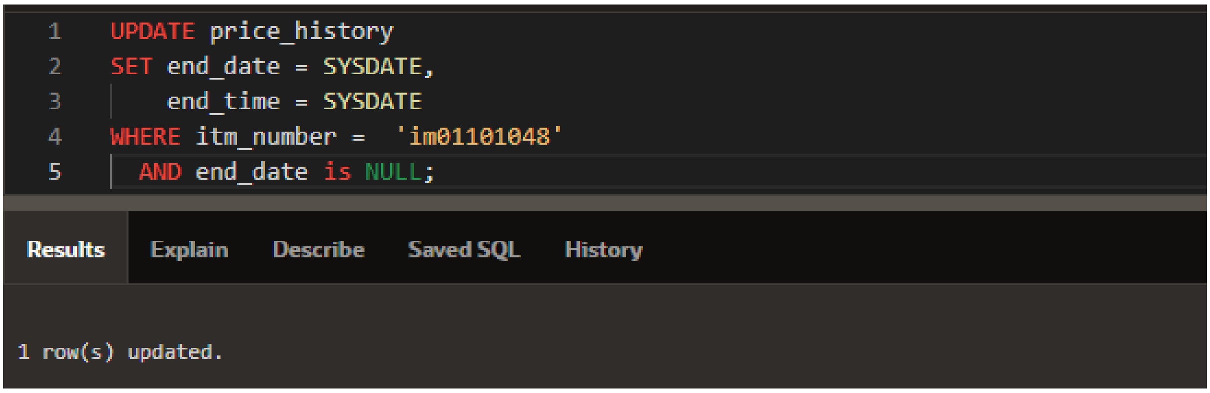
**SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR (end_time, 'HH24:MI')
FROM price_history;**



```
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	-	-

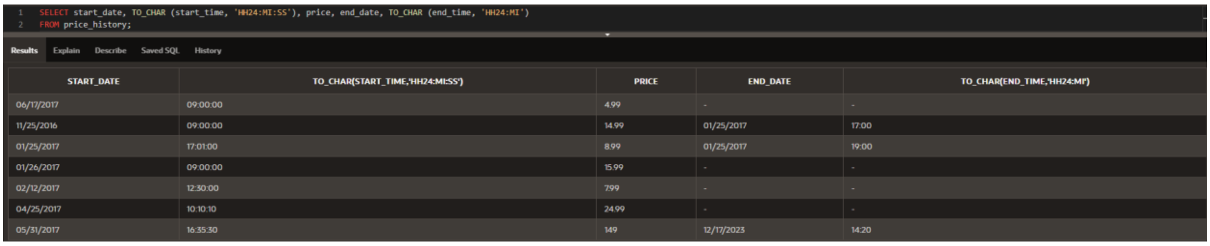
2. Obl is going to update the price of the premium bat so you will need to write a query that will close off the current price by adding the system date values to the end_date and end_time fields. To run this query you will need to both match the item number and identify that the end date is null. This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



```
UPDATE price_history
SET end_date = SYSDATE,
    end_time = SYSDATE
WHERE itm_number =  'im01101048'
  AND end_date is NULL;
```

```
1 row(s) updated.
```

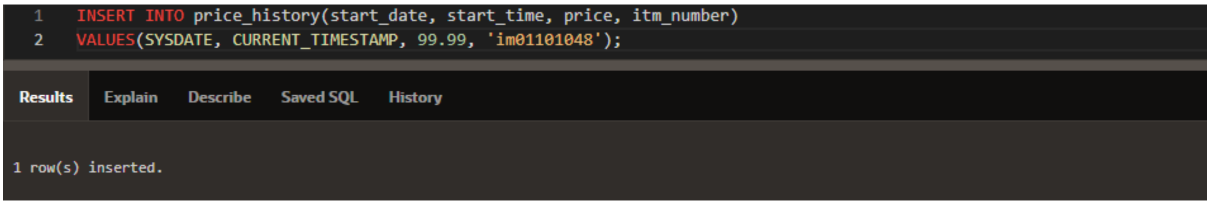
3. Rerun the select statement on the price_history table to ensure that the statement has been executed.



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

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	12/17/2023	14:20

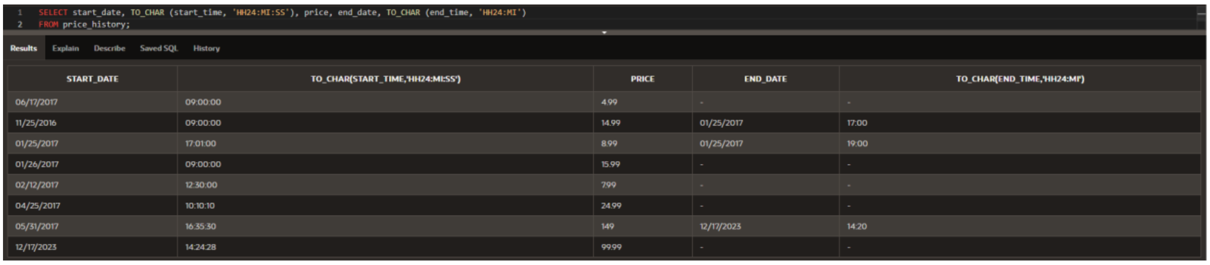
4. Insert a new row that will use the current date and time to set the new price of the premium bat to be 99.99.



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

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

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



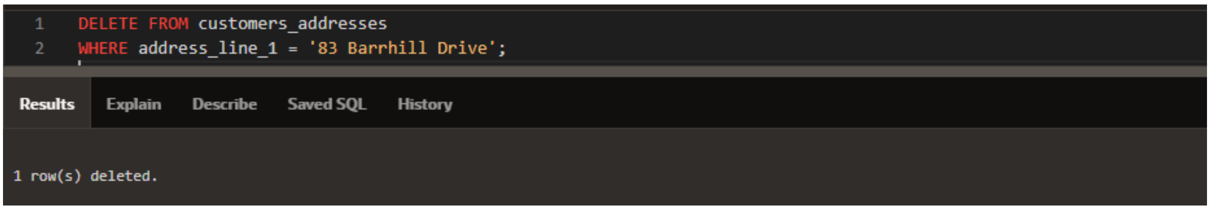
```
SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR (end_time, 'HH24:MI')
FROM price_history;
```

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	12/17/2023	14:20
12/17/2023	14:24:28	99.99	-	-

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.

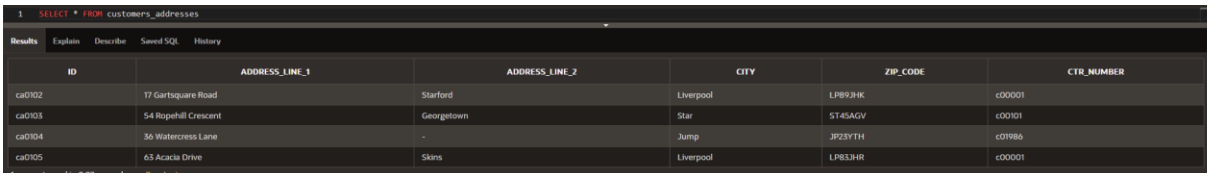
DELETE FROM customers_addresses
WHERE address_line_1 = '83 Barrhill Drive';



```
DELETE FROM customers_addresses
WHERE address_line_1 = '83 Barrhill Drive';
```

```
1 row(s) deleted.
```

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
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