

LAB 2 DATA MANIPULATION LANGUAGE (DML 1)

SECD2523-Database
SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Noor Hidayah

Name	Matric No
LEE XIN HUI	A22EC0066

Section: 08

DML 1 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.

```
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? This file can be found in the Section 6 Lesson 4 interaction (sports data.zip) and must first be extracted.

Yes, it matched.

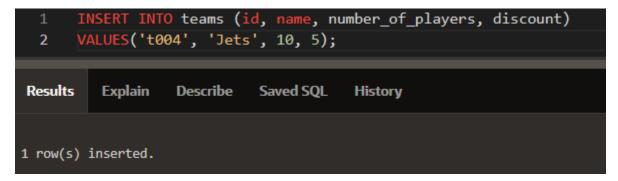
- 3. Run the "sports data.sql" script in APEX to populate your tables
- 4. Check that no errors occurred when you ran the script.

Part 2- Inserting rows to the system

1. Add a new team to the system

id name		Number_of_players	discount	
t004	t004 Jets		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@hoote ch.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@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.

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');



DML 1 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;

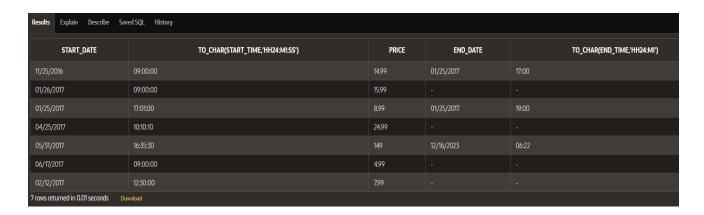
Results Explain Describe Saved SQL History								
START_DATE	TO_CHAR(START_TIME, HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME, HH24:MI')				
11/25/2016	09:00:00	14.99	01/25/2017	17:00				
01/26/2017	09:00:00	15.99						
01/25/2017	17:01:00	8.99	01/25/2017	19:00				
04/25/2017	10:10:10	24.99						
05/31/2017	16:35:30	149						
06/17/2017	09:00:00	4.99						
02/12/2017	12:30:00	7.99						
7 rows returned in 0.01 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 = SYSDATE,
end_time = SYSDATE
WHERE itm number = 'im01101048' AND end date IS NULL;
```

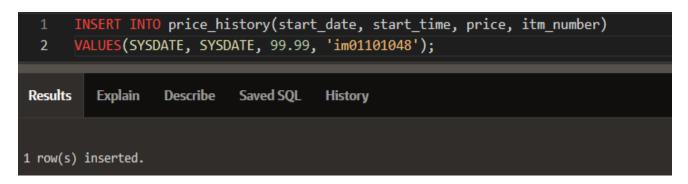
3. Rerun the select statement on the price history table to ensure that the statement has been executed.

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



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, SYSDATE, 99.99, 'im01101048');



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

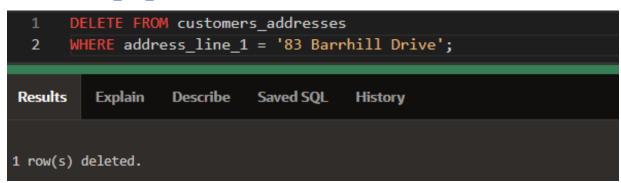
SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR (end_time, 'HH24:MI') FROM price_history;

Results Explain Describe Saved SQL 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				
01/26/2017	09:00:00	15.99						
01/25/2017	17:01:00	8.99	01/25/2017	19:00				
04/25/2017	10:10:10	24.99						
05/31/2017	16:35:30	149	12/16/2023	06:22				
06/17/2017	09:00:00	4.99						
12/16/2023	06:36:53	99.99		-				
02/12/2017	12:30:00	7.99		-				
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

Results Explain Describe Saved SQL History								
D	ADDRESS_LINE_1	ADDRESS_LINE_2	СІТҮ	ZIP_CODE	CTR_NUMBER			
ca0104	36 Watercress Lane		Jump	JP23YTH	c01986			
ca0102	17 Gartsquare Road	Starford	Liverpool	LP89JHK	c00001			
ca0105	63 Acacia Drive	Skins	Liverpool	LP83JHR	c00001			
ca0103	54 Ropehill Crescent	Georgetown	Star	ST45AGV	c00101			
4 rows returned in 0.02 seconds Download								