

SECD2523 - Database

Lab 2: DML 1

SECTION: 08-SECJH

LECTURER NAME: DR NOOR HIDAYAH BINTI ZAKARIA

SUBMISSION DATE: 16 DECEMBER 2023

NO.	NAME	MATRIC NUMBER
1	LEW SHU BEI	A22EC8002

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.
 - a) inventory list
 - b) items
 - c) price history
 - d) sales representatives e) sales rep addresses f) teams
 - g) customers
 - h) customers_addresses i) orders
 - i) ordered items

Table created.

```
1 CREATE TABLE items (
2
       itm_number VARCHAR2(10) NOT NULL,
3
       name VARCHAR2 (20) NOT NULL,
4
       description VARCHAR2(50) NOT NULL,
5
        category VARCHAR2 (25) NOT NULL,
       color VARCHAR2 (15),
6
        "Size" CHAR (1),
7
8
        ilt_id VARCHAR2 (11) NOT NULL,
9
        CONSTRAINT item_pk PRIMARY KEY (itm_number )
10 )
```

Table created.

Table created.

```
1 CREATE TABLE sales_representatives (
        id VARCHAR2(4) NOT NULL,
2
        email VARCHAR2(50) NOT NULL,
3
         first_name VARCHAR2(20) NOT NULL,
4
        last_name VARCHAR2(30) NOT NULL,
5
        phone_number VARCHAR2(11) NOT NULL,
6
        commission_rate NUMBER(2) NOT NULL,
7
        supervisor_id VARCHAR2(4) NOT NULL,
8
        {\tt CONSTRAINT\ sales\_representative\_pk\ PRIMARY\ KEY\ (\ {\tt id\ ),}
9
10
        CONSTRAINT sre_email_uk UNIQUE (email)
11
   )
```

Table created.

Table created.

```
CREATE TABLE teams (
id VARCHAR2(4) NOT NULL,
name VARCHAR2 (20) NOT NULL,
number_of_players NUMBER(2) NOT NULL,
discount NUMBER (2),
CONSTRAINT team_pk PRIMARY KEY ( id )

)
```

Table created.

```
1 CREATE TABLE customers (

ctr_number VARCHAR2(6) NOT NULL,

email VARCHAR2 (50) NOT NULL,

first_name VARCHAR2 (20) NOT NULL,

last_name VARCHAR2 (30) NOT NULL,

phone_number VARCHAR2(11) NOT NULL,

current_balance NUMBER (6,2) NOT NULL,

sre_id VARCHAR2 (4),

tem_id VARCHAR2(4),

loyalty_card_number VARCHAR2(6),

CONSTRAINT customer_pk PRIMARY KEY ( ctr_number ), CONSTRAINT ctr_email_uk UNIQUE (email),

CONSTRAINT ctr_lcn_uk UNIQUE (loyalty_card_number)

)
```

Table created.

```
1 CREATE TABLE customers_addresses (
2
        id VARCHAR2(8) NOT NULL,
        address_line_1 VARCHAR2(30) NOT NULL,
3
4
        address_line_2 VARCHAR2 (30),
5
        city VARCHAR2(15) NOT NULL,
6
        zip_code VARCHAR2(7) NOT NULL,
7
        ctr_number VARCHAR2(6) NOT NULL,
8
        CONSTRAINT customer_address_pk PRIMARY KEY ( id )
9
Table created.
1 CREATE TABLE orders (
2
       id VARCHAR2 (9) NOT NULL,
3
       odr_date DATE NOT NULL,
4
       odr_time DATE NOT NULL,
5
       number_of_units NUMBER(2) NOT NULL,
6
       ctr_number VARCHAR2(6) NOT NULL,
7
       CONSTRAINT orders_pk PRIMARY KEY ( id )
8
Table created.
1 CREATE TABLE ordered_items (
       quantity_ordered NUMBER(3) NOT NULL,
2
3
       quantity_shipped NUMBER(3) NOT NULL,
       itm_number VARCHAR2(10) NOT NULL,
4
       odr_id VARCHAR2(9) NOT NULL,
       CONSTRAINT ordered_item_pk PRIMARY KEY ( itm_number,odr_id )
6
7
Table created.
1 ALTER TABLE customers_addresses ADD CONSTRAINT customer_address_customer_fk FOREIGN KEY ( ctr_number )
2 REFERENCES customers ( ctr_number )
Table altered.
1 V ALTER TABLE customers ADD CONSTRAINT customer_sales_rep_fk FOREIGN KEY ( sre_id )
2 REFERENCES sales_representatives ( id )
Table altered.
1 VALTER TABLE customers ADD CONSTRAINT customer_team_fk FOREIGN KEY ( tem_id )
2 REFERENCES teams ( id )
```

Table altered.

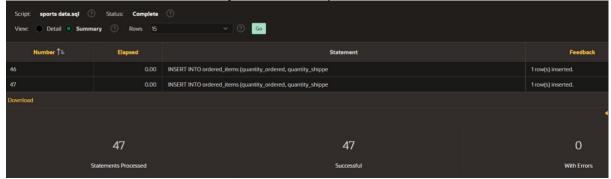
```
1 VALTER TABLE items ADD CONSTRAINT item_inventory_list_fk FOREIGN KEY ( ilt_id )
2 REFERENCES inventory_list ( id )
Table altered.
1 ALTER TABLE orders ADD CONSTRAINT order_customer_fk FOREIGN KEY ( ctr_number )
2 REFERENCES customers ( ctr_number )
Table altered.
1 VALTER TABLE ordered_items ADD CONSTRAINT ordered_item_item_fk FOREIGN KEY (itm_number )
2 REFERENCES items (itm_number )
Table altered.
1 VALTER TABLE ordered_items ADD CONSTRAINT ordered_item_order_fk FOREIGN KEY (odr_id )
2 REFERENCES orders (id )
Table altered.
1 VALTER TABLE sales_rep_addresses ADD CONSTRAINT sales_rep_add_sales_rep_fk FOREIGN KEY ( id )
2 REFERENCES sales_representatives ( id )
Table altered.
1 ALTER TABLE sales_representatives ADD CONSTRAINT sales_rep_sales_rep_fk FOREIGN KEY ( supervisor_id ) REFERENCES sales_representatives ( id )
1 CREATE OR REPLACE TRIGGER fkntm_orders BEFORE
         UPDATE OF ctr_number ON orders
2
3
    BEGIN
4
         raise_application_error(
5
             -20225,
              'Non Transferable FK constraint on table orders is violated'
6
7
8
    END;
```

Trigger created.

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.

Answer: 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.

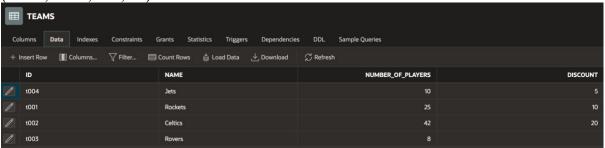


Part 2- Inserting rows to the system

1. Add a new team to the system

id	name	Number_of_players	discount	
t004	Jets	10	5	

Answer: 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		

Answer: INSERT INTO customers

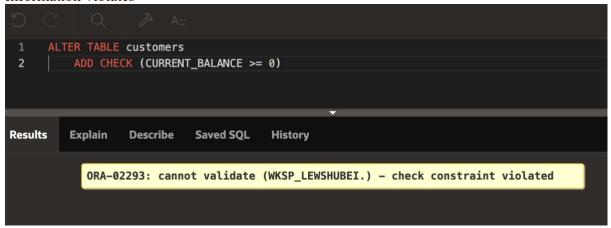
(ctr_number,email,first_name,last_name,phone_number,current_balance,loyalty_card_number)

VALUES ('c02001','brianrog@hootech.com','Brian','Rogers','01654564898','-5','lc4587')

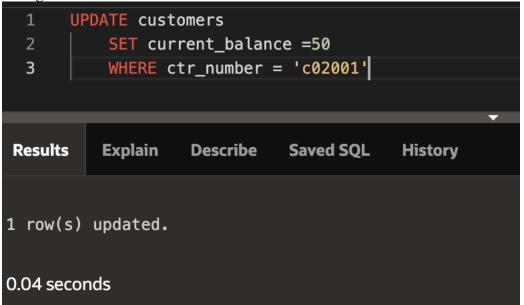
CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMI
c02001	brianrog@hoote	Brian	Rogers	01654564898	-5			lc4587
c00001	bob.thornberry	Robert	Thornberry	01234567898	150	sr01	t001	
c00012	Jjones@freemail	Jennifer	Jones	01505214598				lc1015
c00101	unknown@here	John	Doe	03216547808	987.5	sr01	t002	
c00103	MurciaA@global	Andrew	Murcia	07715246890	85			lc2341
c01986	margal87@delp	Maria	Galant	01442736589	125.65	sr03	t003	

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.

a) Information violates



b) Change the current balance to 50



c) Data updated

Data apaatea								
CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMI
c02001	brianrog@hoote	Brian	Rogers	01654564898	50			lc4587
c00001	bob.thornberry	Robert	Thornberry	01234567898	150	sr01	t001	
c00012	Jjones@freemail	Jennifer	Jones	01505214598				lc1015
c00101	unknown@here	John	Doe	03216547808	987.5	sr01	t002	
c00103	MurciaA@global	Andrew	Murcia	07715246890	85			lc2341
c01986	margal87@delp	Maria	Galant	01442736589	125.65	sr03	t003	

Section 6 Lesson 4 Exercise 2: Data Manipulation Language

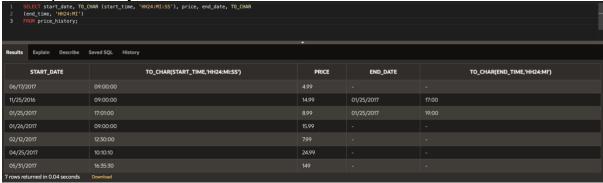
Use DML operations to manage database tables (S6L4 Objective 2)

In this exercise you will populate and work with the data that is stored in the database system.

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.

Answer:

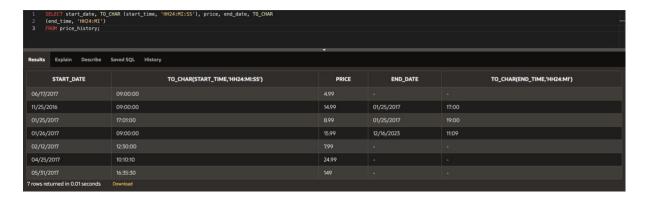
UPDATE price_history

SET end_date = SYSDATE, end_time = SYSDATE

WHERE itm_number = 'im01101045' AND end_date is null;

1	UPDATE price_history									
2	<pre>SET end_date = SYSDATE, end_time = SYSDATE</pre>									
3	<pre>WHERE itm_number = 'im01101045' AND end_date is null;</pre>									
	_									
Results	Explain	Describe	Saved SQL	History						
1 row(s) updated.										
0.01 sec	onds									

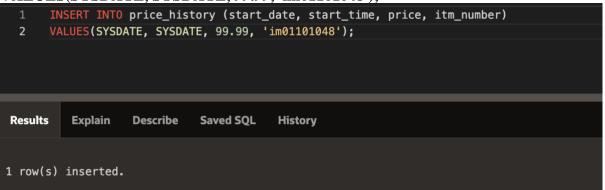
3. Rerun the select statement on the price_history table to ensure that the statement has been executed.



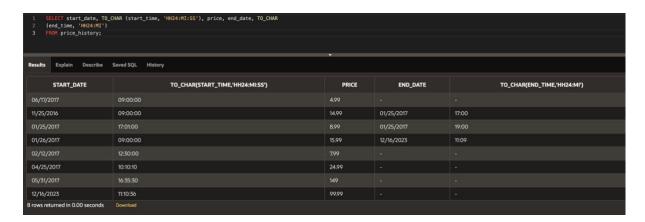
4. Insert a new row that will use the current date and time to set the new price of the premium bat to be 99.99.

Answer:

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

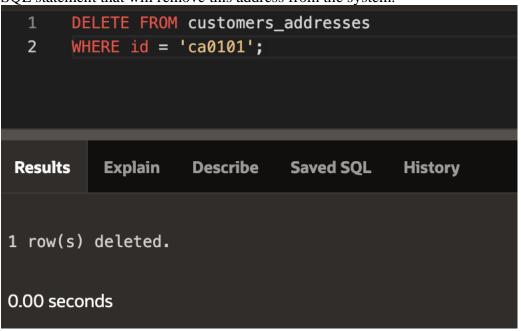


5. Rerun the select statement on the price_history table to ensure that the statement has been executed.



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.

