

Network Design for Faculty of Computing

DML1

Group YGZ

Members: - AHMAD ALYASIN |  
A21EC4030

Lecturer: - Dr.Izzyan izzati

## 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.

### Script Results





Script **Table**

**Success**  
20 statements ran successfully. 24 objects created.

Statement  
85

```
CREATE TABLE inventory_list (  
  id      VARCHAR2(11) NOT NULL,  
  cost    NUMBER(7,2) NOT NULL,  
  units   NUMBER(4) NOT NULL,  
  CONSTRAINT inventory_list_pk PRIMARY KEY ( id )  
)
```

Table created.

  Replace Script  My Session  SQL Worksheet

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

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, Check that no errors occurred when you ran the script.

The screenshot displays the 'Script Results' window in an APEX environment. At the top right, a green notification banner states 'Script uploaded.' with a checkmark icon. Below this, the script is identified as 'Sport\_Data'. A large green checkmark icon and the word 'Success' indicate that the script ran without errors. The message '47 statements ran successfully. 0 objects created.' is displayed. Below the success message, two specific SQL statements are shown as examples:

Statement	SQL Statement	Result
132	<pre>INSERT INTO inventory_list (id, cost, units) VALUES('11010230124', 2.5, 100)</pre>	1 row(s) inserted.
131	<pre>INSERT INTO inventory_list (id, cost, units) VALUES('11010230125', 7.99, 250)</pre>	

At the bottom of the window, there are three buttons: 'Replace Script' (with a circular arrow icon), 'My Session', and 'SQL Worksheet' (highlighted in green). An information icon (i) is also present on the left.

## 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

- a.** Data inserted:

SQL Worksheet

Clear

Find

Actions

Save

Run

```
1 select
2     "ID",
3     "NAME",
4     "NUMBER_OF_PLAYERS",
5     "DISCOUNT"
6 from "TEAMS";
```

ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT
t004	Jets	10	5

Download CSV

- 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		

- a.** New data inserted as shown

```
b. 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');
```

SQL Worksheet

Clear
 Find
 Actions
 Save
 Run

```

1 select
2     "CTR_NUMBER",
3     "EMAIL",
4     "FIRST_NAME",
5     "LAST_NAME",
6     "PHONE_NUMBER",
7     "CURRENT_BALANCE",
8     "SRE_ID",
9     "TEM_ID",
10    "LOYALTY_CARD_NUMBER"
11 from "CUSTOMERS";

```

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	-5	-	-	1c4597

Download CSV

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 violated

```
SQL Worksheet
1 ALTER TABLE CUSTOMERS
2 ADD CHECK (CURRENT_BALANCE >= 0);

ORA-02293: cannot validate (SQL_FMVRI0XJHFESWAGTVYBDJUCSZ.) - check constraint violated
```

b. Command used

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

c. Data updated

```
SQL Worksheet
1 select
2 "CTR_NUMBER",
3 "EMAIL",
4 "FIRST_NAME",
```

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	1c1015
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	1c2341
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	1c4587

## 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;
```

**SQL Worksheet**  Clear  Find **Actions**  Save **Run** 

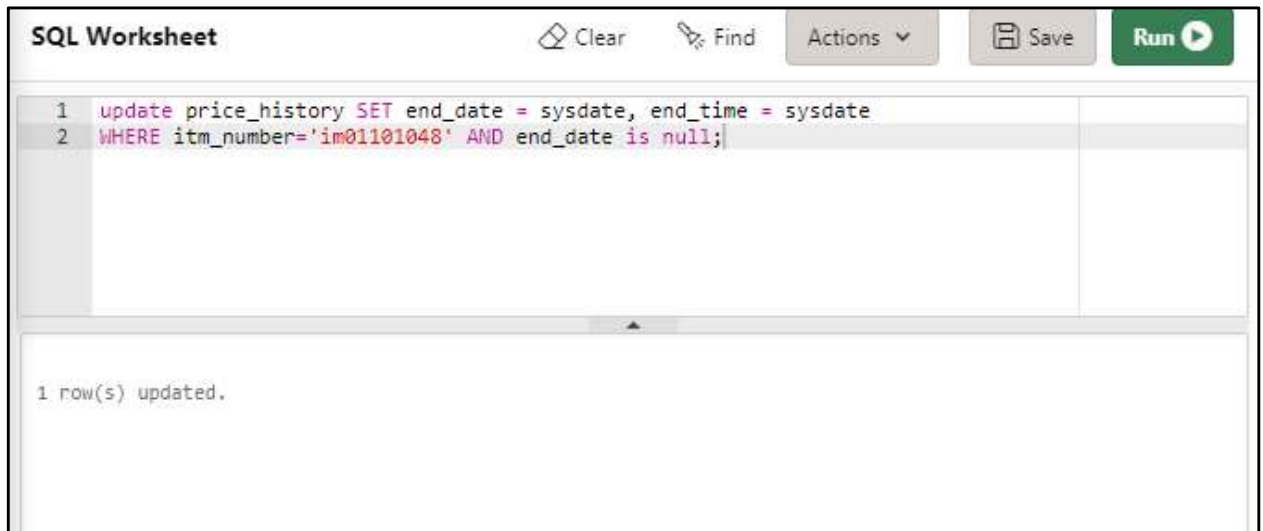
```
1 SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR  
2 (end_time, 'HH24:MI')  
3 FROM price_history;
```

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

[Download CSV](#)

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;
```



The screenshot shows an SQL Worksheet interface. At the top, there is a title bar with the text "SQL Worksheet" and several icons: a trash can for "Clear", a magnifying glass for "Find", a dropdown menu for "Actions", a floppy disk for "Save", and a green button with a play icon for "Run". Below the title bar, the SQL query is entered in a text area, with line numbers 1 and 2 on the left. The query is: `1 update price_history SET end_date = sysdate, end_time = sysdate` and `2 WHERE itm_number='im01101048' AND end_date is null;`. Below the query area, the result of the query is displayed: "1 row(s) updated."

```
SQL Worksheet
```





```
1 update price_history SET end_date = sysdate, end_time = sysdate
2 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;
```

**SQL Worksheet**

 Clear  Find Actions ▾  Save Run 

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

START_DATE	TO_CHAR(START_TIME, 'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME, 'HH24:MI')
17-JUN-17	09:00:00	4.99	-	-
25-NOV-16	09:00:00	14.99	25-JAN-17	17:00
25-JAN-17	17:01:00	8.99	25-JAN-17	19:00
26-JAN-17	09:00:00	15.99	-	-
12-FEB-17	12:30:00	7.99	-	-
25-APR-17	10:10:10	24.99	-	-
31-MAY-17	16:35:30	149	21-DEC-20	05:43


[Download CSV](#)  
7 rows selected.



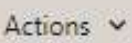




4. Insert a new row that will use the current date and time to set the new price of the premium bat to be 99.99.

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

```
VALUES(SYSDATE, SYSDATE, 99.99, 'im01101048');
```

 **Live SQL**

**SQL Worksheet**  Clear  Find  Actions  Save  Run



```
1 SELECT * from price_history;  
2 |
```

START_DATE	START_TIME	PRICE	END_DATE	END_TIME	ITM_NUMBER
21-DEC-20	21-DEC-20	99.99	-	-	im01101048
17-JUN-17	17-JUN-16	4.99	-	-	im01101044
25-NOV-16	25-NOV-16	14.99	25-JAN-17	25-JAN-17	im01101045
25-JAN-17	25-JAN-17	8.99	25-JAN-17	25-JAN-17	im01101045
26-JAN-17	26-JAN-17	15.99	-	-	im01101045
12-FEB-17	12-FEB-17	7.99	-	-	im01101046
25-APR-17	25-APR-17	24.99	-	-	im01101047
31-MAY-17	31-MAY-17	149	21-DEC-20	21-DEC-20	im01101048

[Download CSV](#)  
8 rows selected.

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;
```

  **Live SQL** Feedback Help mnizzuddin2@graduate.utm.my

**SQL Worksheet** Clear Find Actions Save Run

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

START_DATE	TO_CHAR(START_TIME, 'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME, 'HH24:MI')
21-DEC-20	05:58:04	99.99	-	-
17-JUN-17	09:00:00	4.99	-	-
25-NOV-16	09:00:00	14.99	25-JAN-17	17:00
25-JAN-17	17:01:00	8.99	25-JAN-17	19:00
26-JAN-17	09:00:00	15.99	-	-
12-FEB-17	12:30:00	7.99	-	-
25-APR-17	10:10:10	24.99	-	-
31-MAY-17	16:35:30	149	21-DEC-20	05:43

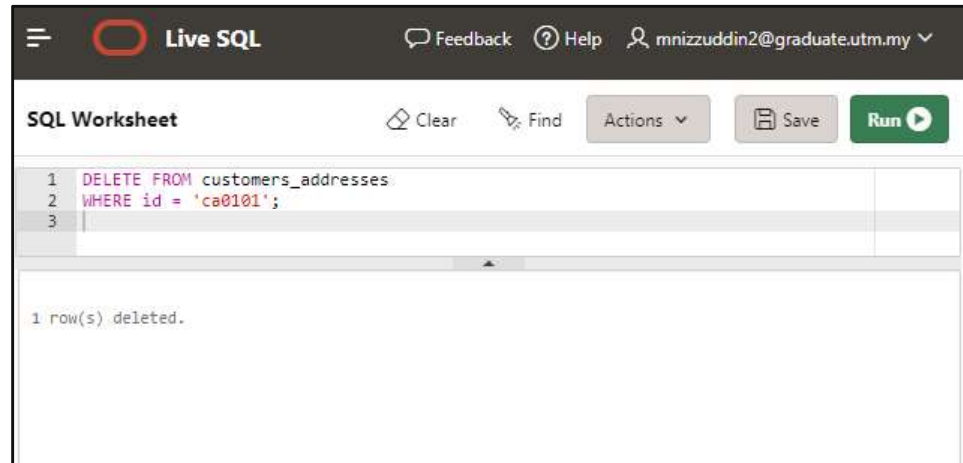
START_DATE	TO_CHAR(START_TIME, 'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME, 'HH24:MI')
21-DEC-20	05:58:04	99.99	-	-
17-JUN-17	09:00:00	4.99	-	-
25-NOV-16	09:00:00	14.99	25-JAN-17	17:00
25-JAN-17	17:01:00	8.99	25-JAN-17	19:00
26-JAN-17	09:00:00	15.99	-	-
12-FEB-17	12:30:00	7.99	-	-
25-APR-17	10:10:10	24.99	-	-
31-MAY-17	16:35:30	149	21-DEC-20	05:43

## Part 2: Deleting rows from the system

1. Bob Thornberry has contacted Obl to ask that the 83 Barr hill 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 id = 'ca0101'



2. Run a select statement on the customers\_addresses table to ensure that the statement has been executed.

SELECT \* FROM customers\_addresses;

