

Database Design Project

Oracle Baseball League Store Database

Project Scenario:

You are a small consulting company specializing in database development. You have just been awarded the contract to develop a data model for a database application system for a small retail store called Oracle Baseball League (OBL).

The Oracle Baseball League store serves the entire surrounding community selling baseball kit. The OBL has two types of customer, there are individuals who purchase items like balls, cleats, gloves, shirts, screen printed t-shirts, and shorts. Additionally customers can represent a team when they purchase uniforms and equipment on behalf of the team.

Teams and individual customers are free to purchase any item from the inventory list, but teams get a discount on the list price depending on the number of players. When a customer places an order we record the order items for that order in our database.

OBL has a team of three sales representatives that officially only call on teams but have been known to handle individual customer complaints.

Name : Eddy Koh Wei Hen
Section : 02

Database Lab 2 - Part 1

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.

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.

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.

Listing according to order

- 1) INVENTORY_LIST
- 2) ITEMS
- 3) PRICE_HISTORY
- 4) SALES_REPRESENTATIVES
- 5) SALES REP ADDRESSES
- 6) TEAMS
- 7) CUSTOMERS
- 8) CUSTOMERS_ADDRESSES
- 9) ORDERS
- 10) ORDERED_ITEMS

Recently Created Tables
ORDERED_ITEMS
ORDERS
CUSTOMERS_ADDRESSES
CUSTOMERS
TEAMS
SALES REP_ADDRESSES
SALES REPRESENTATIVES
PRICE_HISTORY
ITEMS
INVENTORY_LIST

Order to fill in 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.

Yes, the order of the data being added in the "sports data.sql" matches my list.

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

The diagram below show the script is run successfully

6 minutes ago	INSERT INTO items (itm_number, name, description, category, itl_id) VALUES('im0101048', 'premium b	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO items (itm_number, name, description, category, color, "Size", itl_id) VALUES('im01010	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO items (itm_number, name, description, category, color, "Size", itl_id) VALUES('im01010	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO items (itm_number, name, description, category, color, "Size", itl_id) VALUES('im01010	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO items (itm_number, name, description, category, color, "Size", itl_id) VALUES('im01010	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO items (itm_number, name, description, category, color, "Size", itl_id) VALUES('im01010	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO inventory_list (id, cost, units) VALUES('il010250128', 9746, 8);	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO inventory_list (id, cost, units) VALUES('il010250127', 1895, 65);	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO inventory_list (id, cost, units) VALUES('il010250126', 524, 87);	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO inventory_list (id, cost, units) VALUES('il010250125', 799, 250);	WKSP_EDDYKOHWORKSPACE
7 minutes ago	INSERT INTO inventory_list (id, cost, units) VALUES('il010250124', 25, 100); INSERT INTO inventory	WKSP_EDDYKOHWORKSPACE
7 minutes ago	INSERT INTO inventory_list (id, cost, units) VALUES('il010250124', 25, 100);	WKSP_EDDYKOHWORKSPACE
11 minutes ago	INSERT INTO inventory_list (id, cost, units) VALUES('il010250124', 25, 100); INSERT INTO inventory	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO sales_rep_addresses (id, address_line_1, address_line_2, city, zip_code) VALUES(sr02,	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO sales_rep_addresses (id, address_line_1, address_line_2, city, zip_code) VALUES(sr01,	WKSP_EDDYKOHWORKSPACE
4 minutes ago	INSERT INTO sales_representatives (id, email, first_name, last_name, phone_number, commission_rate,	WKSP_EDDYKOHWORKSPACE
4 minutes ago	INSERT INTO sales_representatives (id, email, first_name, last_name, phone_number, commission_rate,	WKSP_EDDYKOHWORKSPACE
4 minutes ago	INSERT INTO sales_representatives (id, email, first_name, last_name, phone_number, commission_rate,	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO price_history (start_date, start_time, price, itm_number) VALUES(TO_DATE('31-May-2017',	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO price_history (start_date, start_time, price, itm_number) VALUES(TO_DATE('25-Apr-2017',	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO price_history (start_date, start_time, price, itm_number) VALUES(TO_DATE('12-Feb-2017',	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO price_history (start_date, start_time, price, itm_number) VALUES(TO_DATE('26-Jan-2017',	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO price_history (start_date, start_time, price, itm_number) VALUES(TO_DATE('26-Jan-2017',	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO price_history (start_date, start_time, price, end_date, end_time, itm_number) VALUES(TO_	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO price_history (start_date, start_time, price, end_date, end_time, itm_number) VALUES(TO_	WKSP_EDDYKOHWORKSPACE
5 minutes ago	INSERT INTO price_history (start_date, start_time, price, end_date, end_time, itm_number) VALUES(TO_	WKSP_EDDYKOHWORKSPACE
6 minutes ago	INSERT INTO items (itm_number, name, description, category, itl_id) VALUES('im0101048', 'premium b	WKSP_EDDYKOHWORKSPACE
2 minutes ago	INSERT INTO customers_addresses (id, address_line_1, address_line_2, city, zip_code, ctr_number) VAL	WKSP_EDDYKOHWORKSPACE
2 minutes ago	INSERT INTO customers_addresses (id, address_line_1, address_line_2, city, zip_code, ctr_number) VAL	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO customers_addresses (id, address_line_1, address_line_2, city, zip_code, ctr_number) VAL	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, sre_	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, loya	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, sre_	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, loya	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, s	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO teams (id, name, number_of_players, discount) VALUES('t003', 'Rovers', 8, null);	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO teams (id, name, number_of_players, discount) VALUES('t002', 'Celtics', 42, 20);	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO teams (id, name, number_of_players, discount) VALUES('t001', 'Rockets', 25, 10);	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO sales_rep_addresses (id, address_line_1, address_line_2, city, zip_code) VALUES(sr02,	WKSP_EDDYKOHWORKSPACE
3 minutes ago	INSERT INTO sales_rep_addresses (id, address_line_1, address_line_2, city, zip_code) VALUES(sr02,	WKSP_EDDYKOHWORKSPACE
87 seconds ago	INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number) VALUES(10, 10, 'o	WKSP_EDDYKOHWORKSPACE
91 seconds ago	INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number) VALUES(18, 18, 'o	WKSP_EDDYKOHWORKSPACE
96 seconds ago	INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number) VALUES(5, 5, 'or0	WKSP_EDDYKOHWORKSPACE
100 seconds ago	INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number) VALUES(5, 5, 'or0	WKSP_EDDYKOHWORKSPACE
103 seconds ago	INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number) VALUES(5, 5, 'or0	WKSP_EDDYKOHWORKSPACE
109 seconds ago	INSERT INTO orders (id, odr_date, odr_time, number_of_units, ctr_number) VALUES('or0101750', TO_DATE	WKSP_EDDYKOHWORKSPACE
115 seconds ago	INSERT INTO orders (id, odr_date, odr_time, number_of_units, ctr_number) VALUES('or0101687', TO_DATE	WKSP_EDDYKOHWORKSPACE
117 seconds ago	INSERT INTO orders (id, odr_date, odr_time, number_of_units, ctr_number) VALUES('or0101425', TO_DATE	WKSP_EDDYKOHWORKSPACE
2 minutes ago	INSERT INTO orders (id, odr_date, odr_time, number_of_units, ctr_number) VALUES('or0101550', TO_DATE	WKSP_EDDYKOHWORKSPACE
2 minutes ago	INSERT INTO orders (id, odr_date, odr_time, number_of_units, ctr_number) VALUES('or0101250', TO_DATE	WKSP_EDDYKOHWORKSPACE
2 minutes ago	INSERT INTO customers_addresses (id, address_line_1, address_line_2, city, zip_code, ctr_number) VA	WKSP_EDDYKOHWORKSPACE
2 minutes ago	INSERT INTO customers_addresses (id, address_line_1, address_line_2, city, zip_code, ctr_number) VAL	WKSP_EDDYKOHWORKSPACE
2 minutes ago	INSERT INTO customers_addresses (id, address_line_1, address_line_2, city, zip_code, ctr_number) VAL	WKSP_EDDYKOHWORKSPACE

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

No error was found , tables successfully created

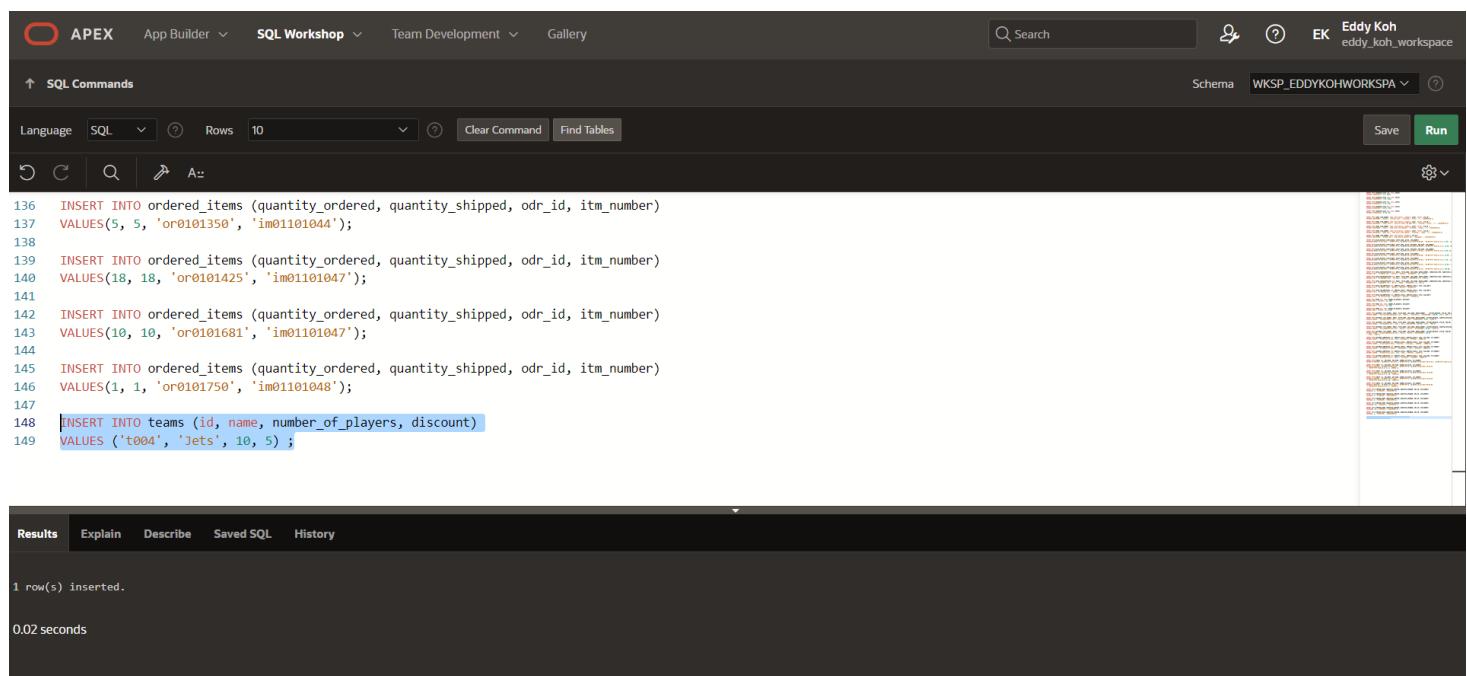
Part 2- Inserting rows to the system

1. Add a new team to the system

id	name	Number_of_players	discount
t004	Jets	10	5

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

Select * from teams;



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL code:

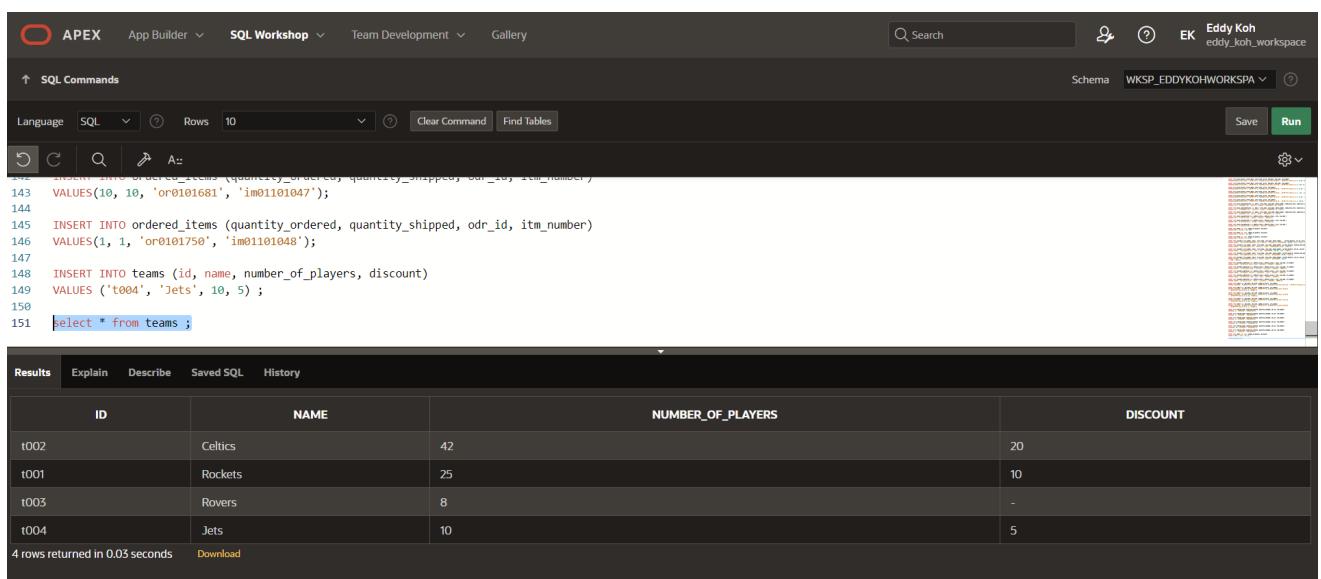
```

136  INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
137  VALUES(5, 5, 'or0101350', 'im01101044');
138
139  INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
140  VALUES(18, 18, 'or0101425', 'im01101047');
141
142  INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
143  VALUES(10, 10, 'or0101681', 'im01101047');
144
145  INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
146  VALUES(1, 1, 'or0101750', 'im01101048');
147
148  INSERT INTO teams (id, name, number_of_players, discount)
149  VALUES ('t004', 'Jets', 10, 5 );

```

The Results tab shows the output of the command:

1 row(s) inserted.
0.02 seconds



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL code:

```

142  INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
143  VALUES(10, 10, 'or0101681', 'im01101047');
144
145  INSERT INTO ordered_items (quantity_ordered, quantity_shipped, odr_id, itm_number)
146  VALUES(1, 1, 'or0101750', 'im01101048');
147
148  INSERT INTO teams (id, name, number_of_players, discount)
149  VALUES ('t004', 'Jets', 10, 5 );
150
151  select * from teams ;

```

The Results tab shows the output of the command:

4 rows returned in 0.03 seconds Download

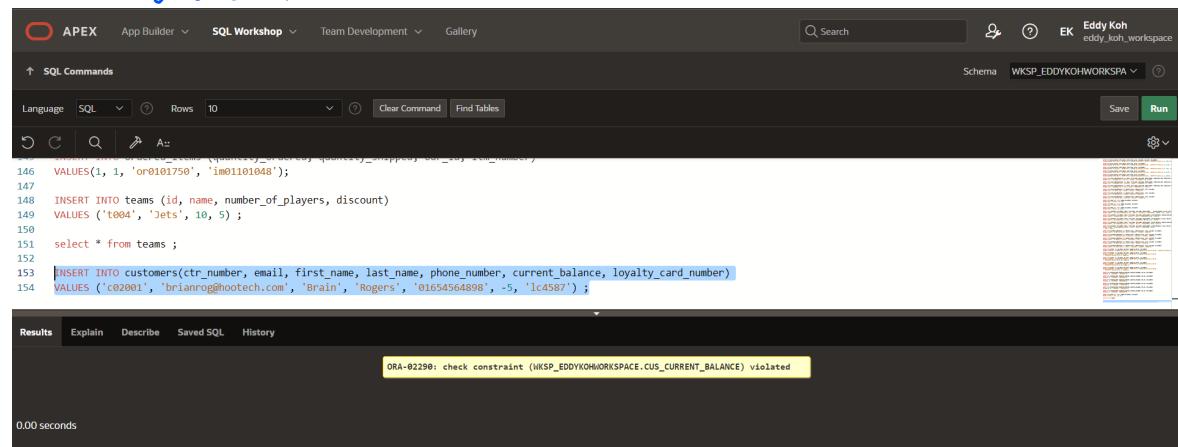
ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT
t002	Celtics	42	20
t001	Rockets	25	10
t003	Rovers	8	-
t004	Jets	10	5

A blue curly brace on the left side of the results table is labeled "Output table".

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		

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



The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands tab, there is a code editor with the following SQL statement:

```

146 VALUES(1, 1, 'or0101750', 'im01101048');
147
148 INSERT INTO teams (id, name, number_of_players, discount)
149 VALUES ('t004', 'Jets', 10, 5);
150
151 select * from teams ;
152
153 INSERT INTO customers(ctr_number, email, first_name, last_name, phone_number, current_balance, loyalty_card_number)
154 VALUES ('c02001', 'brianrog@hootech.com', 'Brain', 'Rogers', '01654564898', -5, 'lc4587');

```

An error message is displayed in a yellow box: "ORA-02290: check constraint (WKSPEDDYKOHWORKSPACE.CUS_CURRENT_BALANCE) violated".

Error {

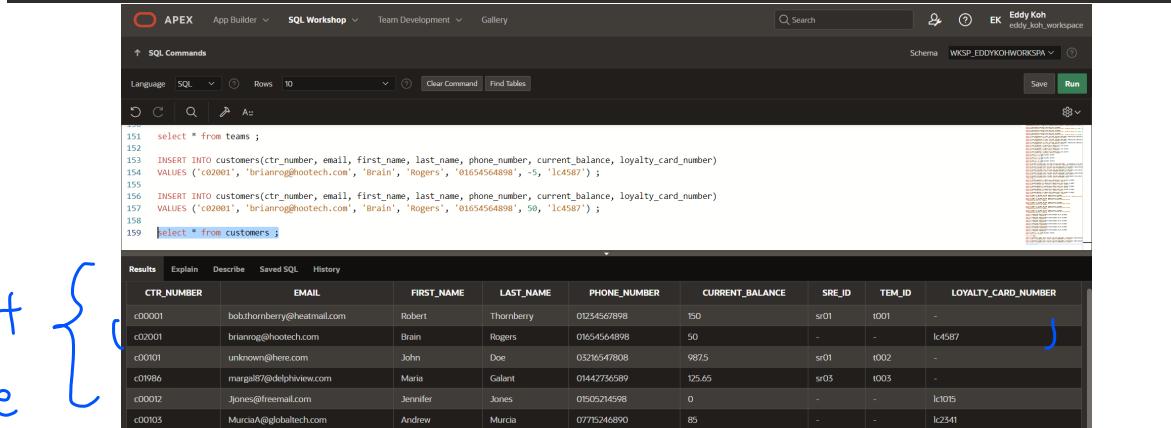
3. This information violates the check constraint that the current balance must not be less than zero. Change the current balance to 50 and rerun the query.

INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, loyalty_card_number)
VALUES ('c02001', 'brianrog@hootech.com', 'Brain', 'Rogers',
'01654564898', 50, 'lc4587');

Select * from customers;



The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands tab, the previous SQL statement is run again with a corrected current_balance value of 50. The message "1 row(s) inserted." is displayed in the results area.



The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands tab, the following SQL statements are run:

```

151 select * from teams ;
152
153 INSERT INTO customers(ctr_number, email, first_name, last_name, phone_number, current_balance, loyalty_card_number)
154 VALUES ('c02001', 'brianrog@hootech.com', 'Brain', 'Rogers', '01654564898', -5, 'lc4587');
155
156 INSERT INTO customers(ctr_number, email, first_name, last_name, phone_number, current_balance, loyalty_card_number)
157 VALUES ('c02001', 'brianrog@hootech.com', 'Brain', 'Rogers', '01654564898', 50, 'lc4587');
158
159 select * from customers ;

```

The results show the updated customer table with the new row added:

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
c00001	bobthompson@heatmail.com	Robert	Thompson	0234567898	150	sr01	t001	-
c02001	brianrog@hootech.com	Brain	Rogers	01654564898	50	-	-	lc4587
c01001	unknownhere.com	John	Doe	03216547808	987.5	sr01	t002	-
c01986	margal07@delphiview.com	Maria	Galant	01442736589	125.65	sr05	t003	-
c00012	jones@freemail.com	Jennifer	Jones	01950214958	0	-	-	lc1015
c00005	MariaA@globaltech.com	Andrew	Maria	07752146890	85	-	-	lc2541

Output table {

Database Design Project

Oracle Baseball League Store Database

Project Scenario:

You are a small consulting company specializing in database development. You have just been awarded the contract to develop a data model for a database application system for a small retail store called Oracle Baseball League (OBL).

The Oracle Baseball League store serves the entire surrounding community selling baseball kit. The OBL has two types of customer, there are individuals who purchase items like balls, cleats, gloves, shirts, screen printed t-shirts, and shorts. Additionally customers can represent a team when they purchase uniforms and equipment on behalf of the team.

Teams and individual customers are free to purchase any item from the inventory list, but teams get a discount on the list price depending on the number of players. When a customer places an order we record the order items for that order in our database.

OBL has a team of three sales representatives that officially only call on teams but have been known to handle individual customer complaints.

Name : Eddy Koh Wei Hen
Section : 02

Database Lab 2 - Part 2

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.
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.

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.

Part 1- Updating rows to the system

- 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;
```

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery' are visible. On the right, there's a user profile for 'Eddy Koh' and a schema dropdown set to 'WKSP_EDDYKOHWORKSPA'. Below the navigation is a search bar and a toolbar with icons for copy, clear, run, etc. The main area is titled 'SQL Commands' with a sub-section 'Results'. The results show a table with columns: START_DATE, TO_CHAR(START_TIME,'HH24:MI:SS'), PRICE, END_DATE, and TO_CHAR(END_TIME,'HH24:MI'). The data consists of seven rows from different dates in 2017, with various start times and prices. A note at the bottom says '7 rows returned in 0.04 seconds'.

START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
04/25/2017	10:10:10	24.99	-	-
05/31/2017	16:35:30	149	-	-
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
02/12/2017	12:30:00	7.99	-	-
06/17/2017	09:00:00	4.99	-	-
01/26/2017	09:00:00	15.99	-	-

7 rows returned in 0.04 seconds Download

Output table

- 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 = SYSTIMESTAMP
WHERE item_number = 'im01101048' and end_date IS NULL
and end_time IS NULL ;

The screenshot shows the Oracle SQL Workshop interface with the same navigation and search bar as the previous screenshot. The main area has a toolbar with icons for copy, clear, run, etc. The results table shows a single row updated, with a note at the bottom saying '0.03 seconds'.

Result
1 row(s) updated.

0.03 seconds

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;`

Output Table {

START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
04/25/2017	10:10:10	24.99	-	-
05/31/2017	16:35:30	149	11/09/2025	14:14
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
02/12/2017	12:30:00	799	-	-
06/17/2017	09:00:00	4.99	-	-
01/26/2017	09:00:00	15.99	-	-

7 rows returned in 0.00 seconds [Download](#)

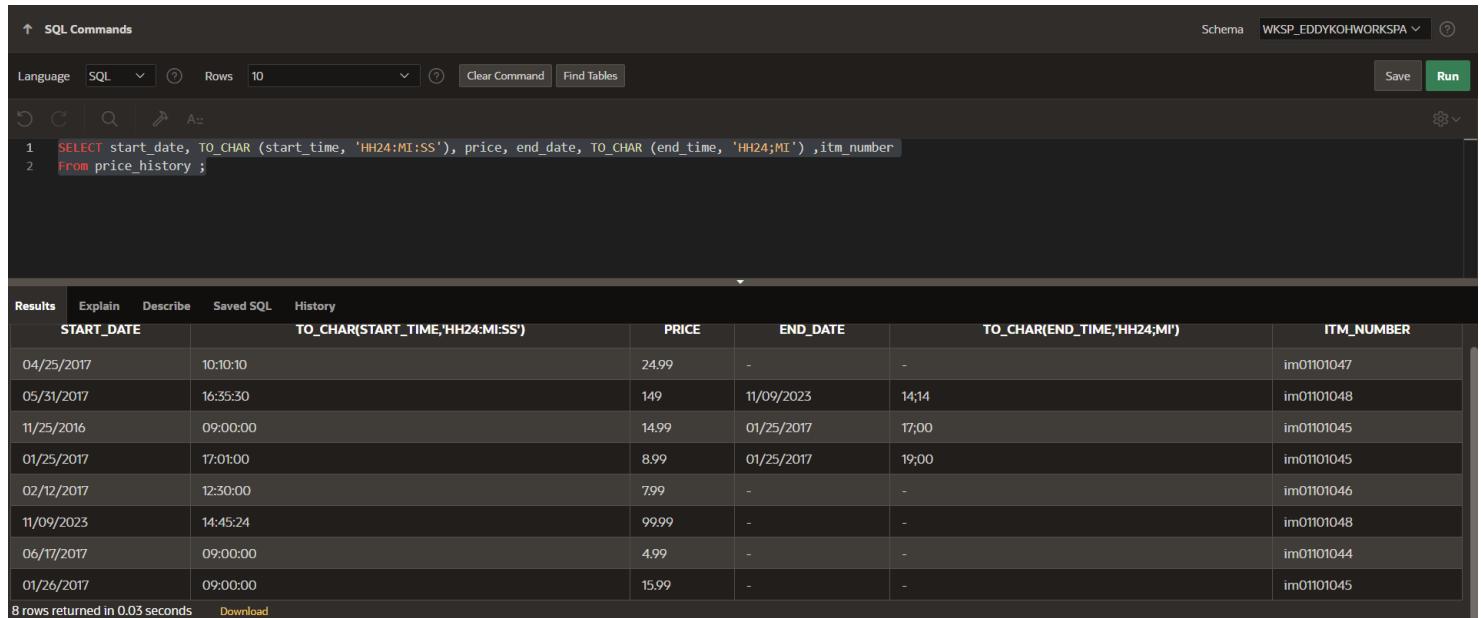
4. Insert a new row that will use the current date and time to set the new price of the premium bat to be 99.99.

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

1 row(s) inserted.
 0.03 seconds

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'), item_number
 FROM price_history;*



The screenshot shows a SQL command window with the following details:

- Language:** SQL
- Rows:** 10
- Clear Command** and **Find Tables** buttons
- Run** button
- Schema:** WKSP_EDDYKOHWORKSPA

The SQL code entered is:

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

The results table displays 8 rows of data:

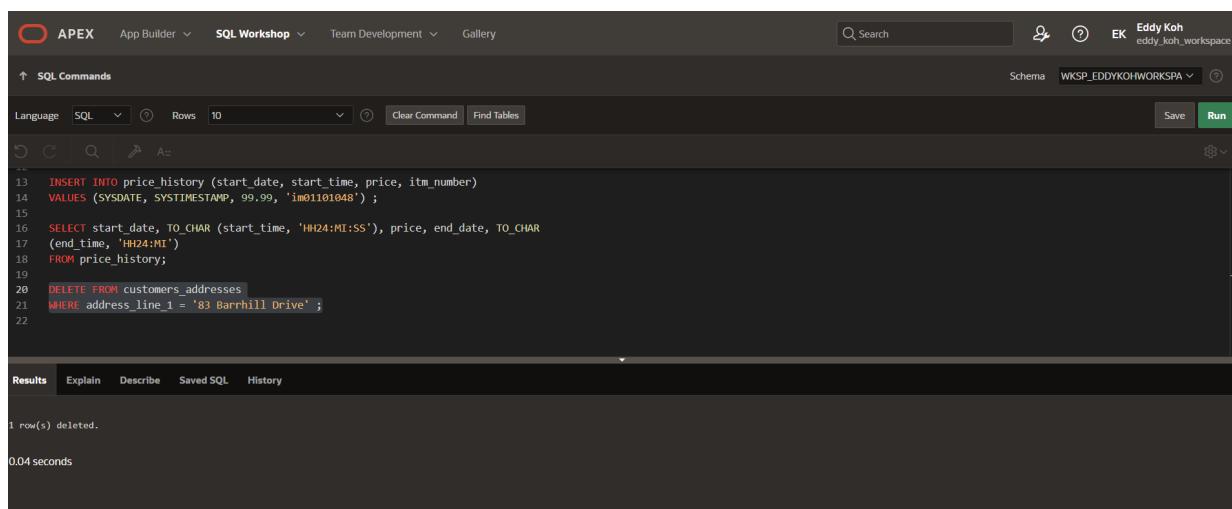
START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')	ITEM_NUMBER
04/25/2017	10:10:10	24.99	-	-	im01101047
05/31/2017	16:55:30	149	11/09/2023	14:14	im01101048
11/25/2016	09:00:00	14.99	01/25/2017	17:00	im01101045
01/25/2017	17:01:00	8.99	01/25/2017	19:00	im01101045
02/12/2017	12:30:00	7.99	-	-	im01101046
11/09/2023	14:45:24	99.99	-	-	im01101048
06/17/2017	09:00:00	4.99	-	-	im01101044
01/26/2017	09:00:00	15.99	-	-	im01101045

8 rows returned in 0.03 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';*



The screenshot shows a SQL command window with the following details:

- Language:** SQL
- Rows:** 10
- Clear Command** and **Find Tables** buttons
- Run** button
- Schema:** WKSP_EDDYKOHWORKSPA

The SQL code entered is:

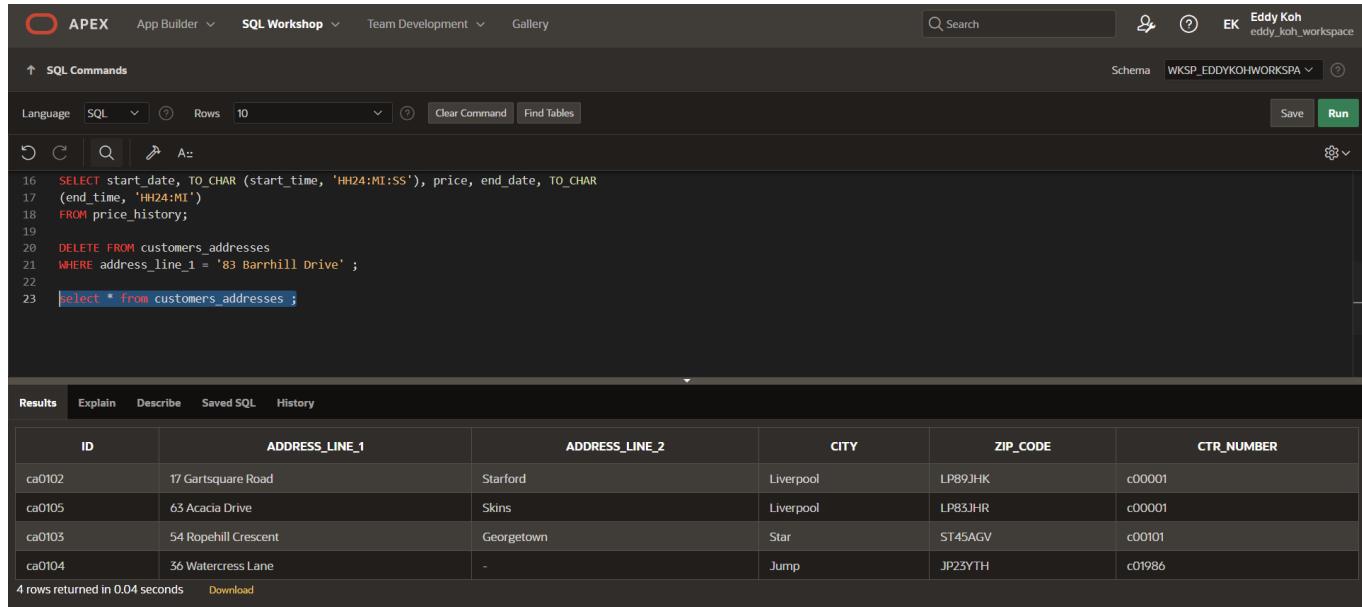
```
13 INSERT INTO price_history (start_date, start_time, price, item_number)
14 VALUES (SYSDATE, SYSTIMESTAMP, 99.99, 'im01101048') ;
15
16 SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR
17 (end_time, 'HH24:MI')
18 FROM price_history;
19
20 DELETE FROM customers_addresses
21 WHERE address_line_1 = '83 Barrhill Drive' ;
22
```

The results table displays the message:

1 row(s) deleted.
 0.04 seconds

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

Select * from customers_addresses;



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user Eddy Koh (eddy_koh.workspace). The main area is titled 'SQL Commands' with tabs for Language (SQL selected), Rows (10), Clear Command, Find Tables, Save, and Run. The code editor contains the following SQL:

```
16 SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR
17 (end_time, 'HH24:MI')
18 FROM price_history;
19
20 DELETE FROM customers_addresses
21 WHERE address_line_1 = '83 Barrhill Drive';
22
23 select * from customers_addresses ;
```

The results tab is active, displaying a table with columns ID, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, ZIP_CODE, and CTR_NUMBER. The data is as follows:

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

4 rows returned in 0.04 seconds. There is a 'Download' link at the bottom.