

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.

LAB 2: DATA MANIPULATION LANGUAGE DML1 (PART 1 & PART 2)

NAME: ISKANDAR HAKIMI BIN ZULKIPPLI

MATRIC NUMBER: B23CS0040

SECTION: 8

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.
 - List of order 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 does match with the order of data added in “sports data.sql”.
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:

```
1 INSERT INTO teams (id, name, number_of_players, discount)
2 VALUES ('t004', 'Jets', 10, 5);
3
```

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		

Answer:

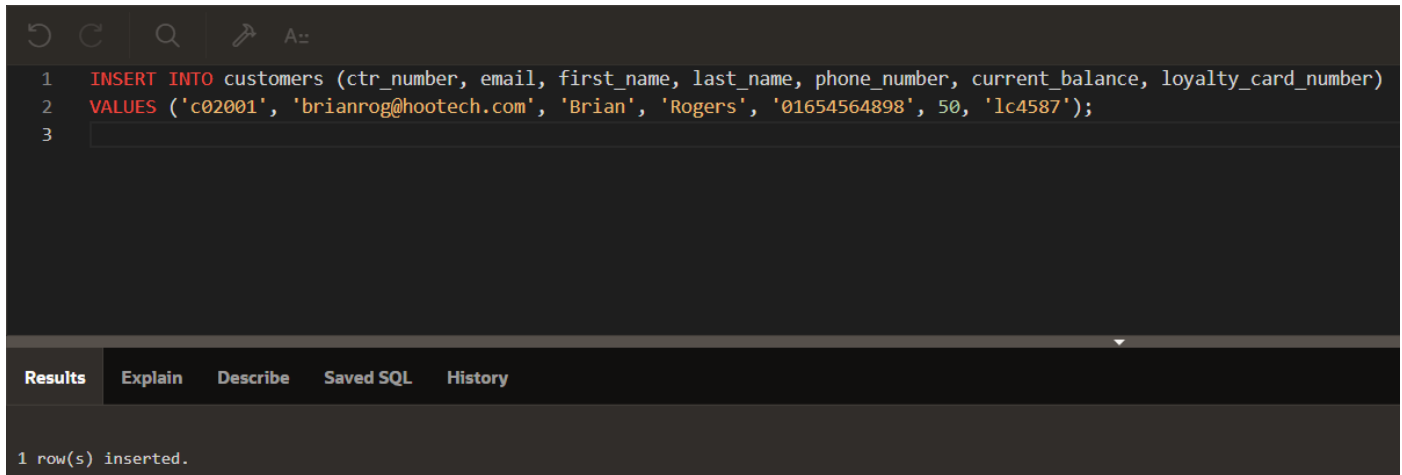
```
1 INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, loyalty_card_number)
2 VALUES ('c02001', 'brianrog@hootech.com', 'Brian', 'Rogers', '01654564898', -5, 'lc4587');
3
```

Results Explain Describe Saved SQL History

ORA-02290: check constraint (WKSP_ISKANDARS.CHECK_BALANCE) violated

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.

Answer:



The screenshot shows a SQL IDE interface. At the top, there is a toolbar with icons for undo, redo, search, and a command prompt. Below the toolbar, the SQL editor contains the following code:

```
1 INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, loyalty_card_number)
2 VALUES ('c02001', 'brianrog@hootech.com', 'Brian', 'Rogers', '01654564898', 50, 'lc4587');
3
```

Below the editor, there is a tabbed interface with the following tabs: Results, Explain, Describe, Saved SQL, and History. The 'Results' tab is selected, and it displays the following output:

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

Section 6 Lesson 4 Exercise 2: Data Manipulation Language

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

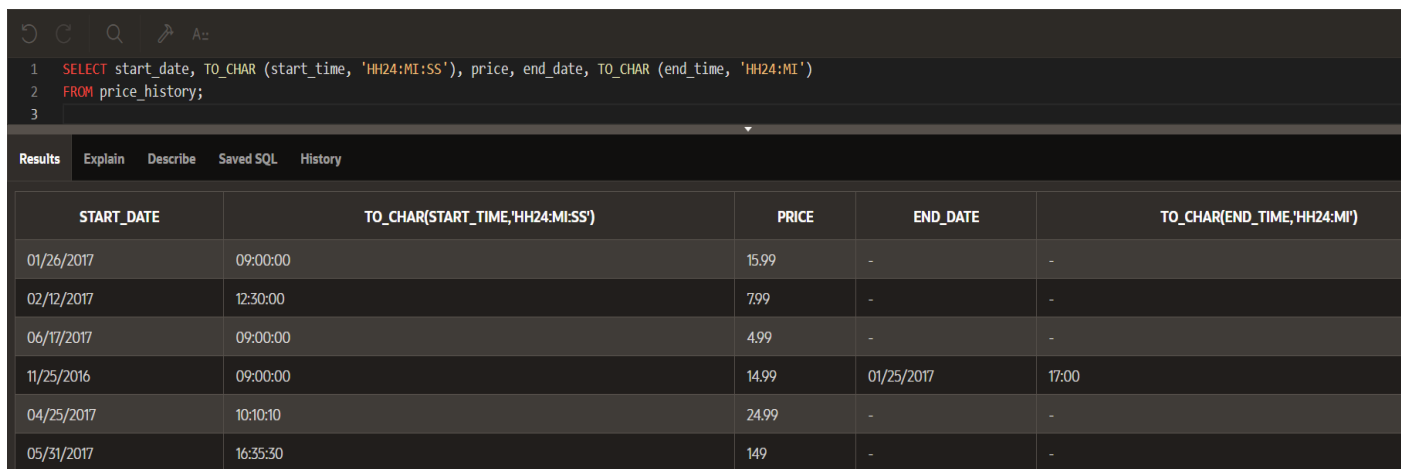
In this exercise you will populate and work with the data that is stored in the database system.

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;
```

Answer:

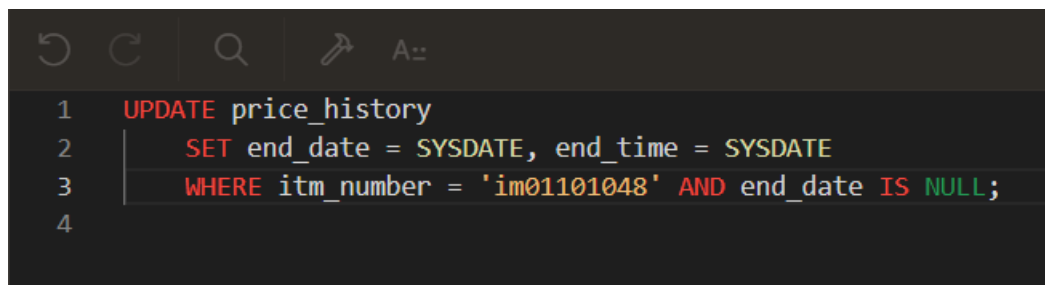


The screenshot shows a SQL query execution interface. The query is: `SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR (end_time, 'HH24:MI') FROM price_history;`. The results are displayed in a table with 5 columns: START_DATE, TO_CHAR(START_TIME, 'HH24:MI:SS'), PRICE, END_DATE, and TO_CHAR(END_TIME, 'HH24:MI').

START_DATE	TO_CHAR(START_TIME, 'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME, 'HH24:MI')
01/26/2017	09:00:00	15.99	-	-
02/12/2017	12:30:00	7.99	-	-
06/17/2017	09:00:00	4.99	-	-
11/25/2016	09:00:00	14.99	01/25/2017	17:00
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.

Answer:

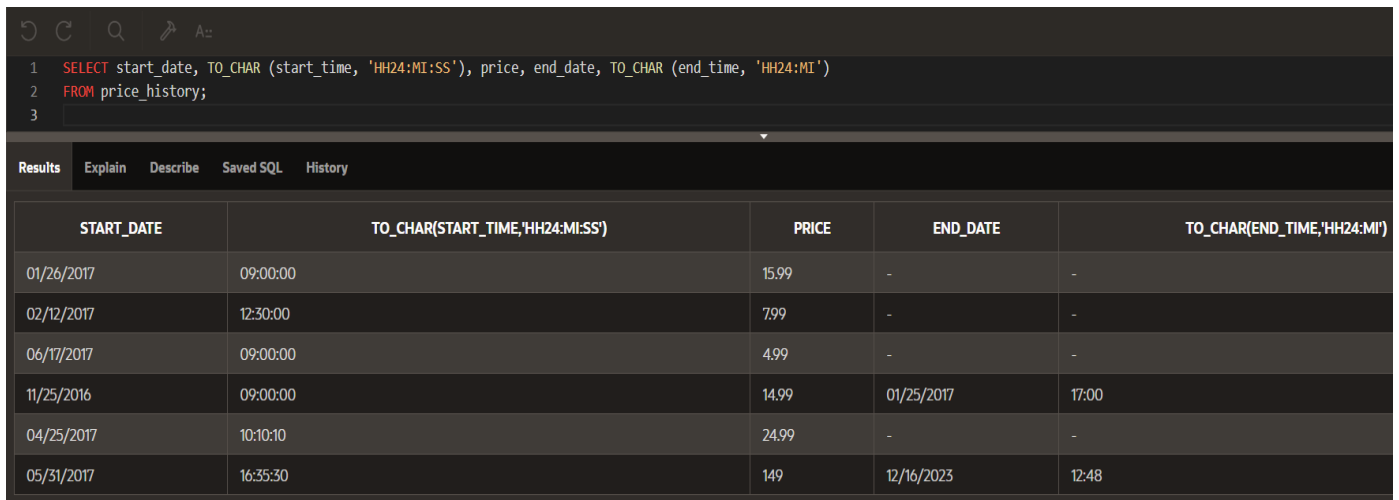


The screenshot shows a SQL query execution interface. The query is: `UPDATE price_history SET end_date = SYSDATE, end_time = SYSDATE WHERE itm_number = 'im01101048' AND end_date IS NULL;`

```
1 UPDATE price_history
2   SET end_date = SYSDATE, end_time = SYSDATE
3   WHERE itm_number = 'im01101048' AND end_date IS NULL;
4
```

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

Answer:



The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains the following SQL statement:

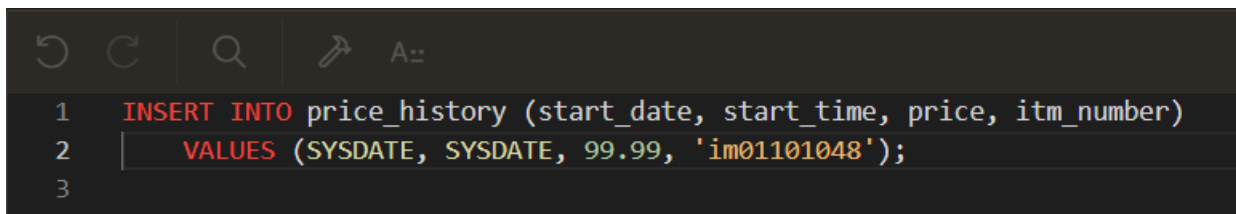
```
1 SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR (end_time, 'HH24:MI')
2 FROM price_history;
3
```

The results pane shows the following table:

START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
01/26/2017	09:00:00	15.99	-	-
02/12/2017	12:30:00	7.99	-	-
06/17/2017	09:00:00	4.99	-	-
11/25/2016	09:00:00	14.99	01/25/2017	17:00
04/25/2017	10:10:10	24.99	-	-
05/31/2017	16:35:30	149	12/16/2023	12:48

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:

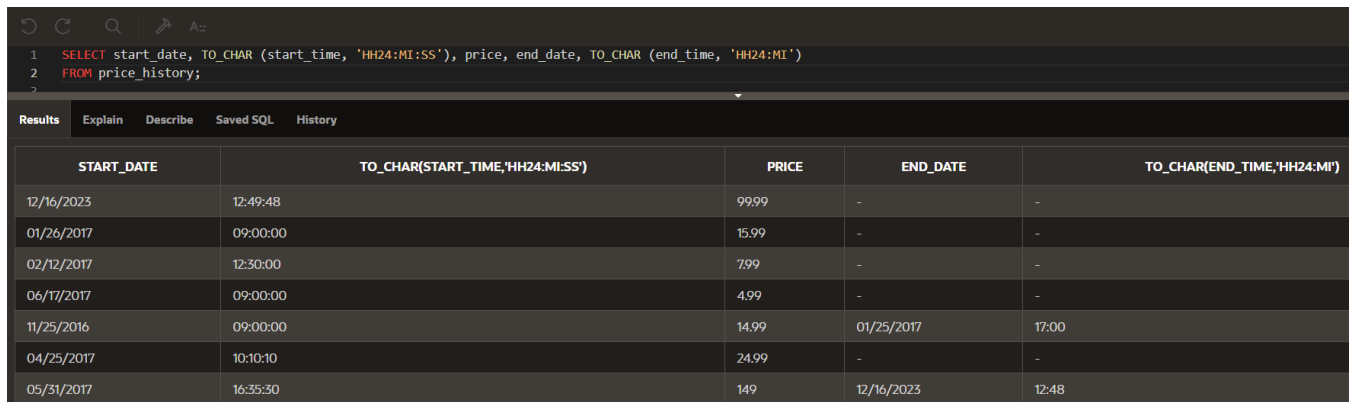


The screenshot shows a SQL IDE with a query editor containing the following SQL statement:

```
1 INSERT INTO price_history (start_date, start_time, price, itm_number)
2 VALUES (SYSDATE, SYSDATE, 99.99, 'im01101048');
3
```

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

Answer:



The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains the following SQL statement:

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

The results pane shows the following table:

START_DATE	TO_CHAR(START_TIME,'HH24:MI:SS')	PRICE	END_DATE	TO_CHAR(END_TIME,'HH24:MI')
12/16/2023	12:49:48	99.99	-	-
01/26/2017	09:00:00	15.99	-	-
02/12/2017	12:30:00	7.99	-	-
06/17/2017	09:00:00	4.99	-	-
11/25/2016	09:00:00	14.99	01/25/2017	17:00
04/25/2017	10:10:10	24.99	-	-
05/31/2017	16:35:30	149	12/16/2023	12:48

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 no longer receive parcels at this address. Write a SQL statement that will remove this address from the system.

Answer:

```
1 DELETE FROM customers_addresses
2 WHERE address_line_1 = '83 Barrhill Drive';
3
```

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

Answer:

```
1 SELECT address_line_1
2 FROM customers_addresses;
3
```

Results	Explain	Describe	Saved SQL	History
ADDRESS_LINE_1				
54 Ropehill Crescent				
17 Gartsquare Road				
36 Watercress Lane				
63 Acacia Drive				
83 Barrhill Drive				