



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

DATA BASE (SECD2523-06)
LAB SQL 2 (part 1&2)

SECTION:	06
NAME:	Kwek Jia Cong (A22EC0122)
INSTRUCTORS:	Dr. Izyan Izzati Binti Kamsani

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.

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.

List according to order

1. INVENTORY_LIST
2. ITEMS
3. PRICE_HISTORY
4. SALES_REPRESENTATIVE
5. SALES_REP_ADDRESSES
6. TEAMS
7. CUSTOMERS
8. CUSTOMERS_ADDRESSES
9. ORDERS
10. ORDERED_ITEMS

ORDERED_ITEMS
ORDERS
CUSTOMERS_ADDRESSES
CUSTOMERS
TEAMS
SALES_REP_ADDRESSES
SALES_REPRESENTATIVES
PRICE_HISTORY
ITEMS
INVENTORY_LIST

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

Tables created after successfully running the "sports data.sql" script in Apex.

Table Name ↕	Rows	Last Analyzed	Type
CUSTOMERS	-	-	TABLE
CUSTOMERS_ADDRESSES	-	-	TABLE
HTMLDB_PLAN_TABLE	-	-	TABLE
INVENTORY_LIST	-	-	TABLE
ITEMS	-	-	TABLE
ORDERED_ITEMS	-	-	TABLE
ORDERS	-	-	TABLE
PRICE_HISTORY	-	-	TABLE
SALES_REPRESENTATIVES	-	-	TABLE
SALES_REP_ADDRESSES	-	-	TABLE
TEAMS	-	-	TABLE

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

Tables successfully created, no errors occurred.

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)
```

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

Results Explain Describe Saved SQL History

1 row(s) inserted.

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@hoootech.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@hoootech.com', 'Brian', 'Rogers', '01654564898', -5, 'lc4587');
```

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

Results Explain Describe Saved SQL History

ORA-02290: check constraint (HKSP_JCKWEKWORKSPACE.CUS_CURRENT_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.

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

```
153 INSERT INTO customers (ctr_number, email, first_name, last_name, phone_number, current_balance, loyalty_card_number)
154 VALUES('c02001', 'brianrog@hoootech.com', 'Brian', 'Rogers', '01654564898', 50, 'lc4587');
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

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.

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:SS')  
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.

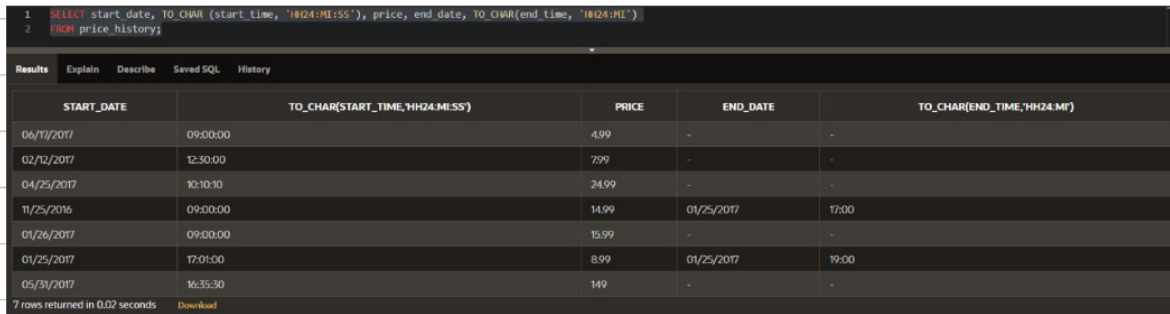
Part 2: Deleting rows from the system

1. Bob Thornberry has contacted Obl to ask that the 83 Barrhill Drive address be removed from the system as he can no 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

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;
```



The screenshot shows a SQL query window with the following SQL code:

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

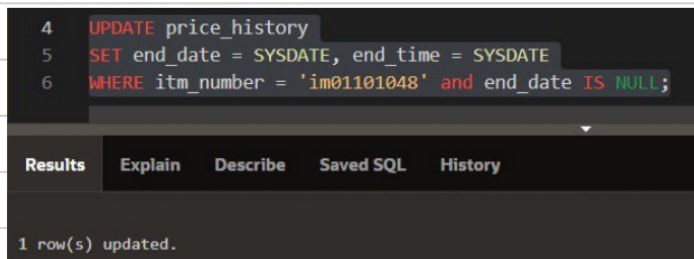
The results are displayed in a table with the following columns: START_DATE, TO_CHAR(START_TIME, 'HH24:MI:SS'), PRICE, END_DATE, and TO_CHAR(END_TIME, 'HH24:MI').

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

7 rows returned in 0.02 seconds

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 a SQL query window with the following SQL code:

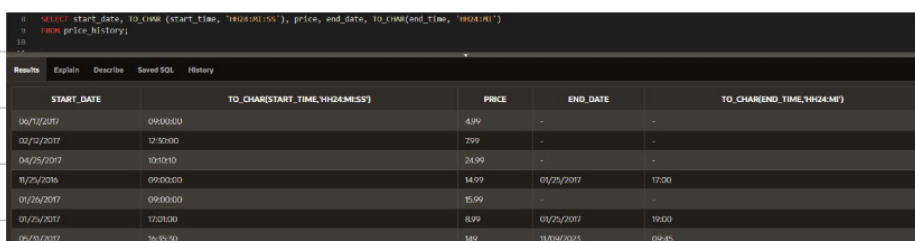
```
4 UPDATE price_history  
5 SET end_date = SYSDATE, end_time = SYSDATE  
6 WHERE itm_number = 'im01101048' and end_date IS NULL;
```

The results are displayed in a table with the following columns: Results, Explain, Describe, Saved SQL, History.

Results	Explain	Describe	Saved SQL	History
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 (start_time, 'HH24:MI:SS')  
FROM price_history;
```



The screenshot shows a SQL query window with the following SQL code:

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

The results are displayed in a table with the following 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')
06/11/2017	09:00:00	4.99	-	-
02/12/2017	12:30:00	7.99	-	-
04/25/2017	10:10:10	24.99	-	-
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
05/31/2017	16:35:30	149	11/09/2023	09:45

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')
```

```
11 INSERT INTO price_history (start_date, start_time, price, itm_number)
12 VALUES (SYSDATE, SYSDATE, 99.99, 'im01101048')
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

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:SS'), itm_number
FROM price_history;
```

```
14 SELECT start_date, TO_CHAR (start_time, 'HH24:MI:SS'), price, end_date, TO_CHAR(end_time, 'HH24:MI'),itm_number
15 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')	ITM_NUMBER
06/17/2017	09:00:00	4.99	-	-	im01101044
02/10/2017	12:30:00	7.99	-	-	im01101046
04/25/2017	10:10:10	24.99	-	-	im01101047
11/25/2016	09:00:00	14.99	01/25/2017	17:00	im01101045
01/26/2017	09:00:00	15.99	-	-	im01101045
01/25/2017	17:01:00	8.99	01/25/2017	19:00	im01101045
05/31/2017	16:35:30	149	11/09/2023	09:45	im01101048
11/09/2023	11:18:04	99.99	-	-	im01101048

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.

```
- DELETE FROM customer_addresses  
WHERE id = 'ca0101';
```

```
20 DELETE FROM customers_addresses  
21 WHERE id = 'ca0101';  
22  
Results Explain Describe Saved SQL Hi  
1 row(s) deleted.
```

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

```
- SELECT id, address_line_1, address_line_2, city, zip_code, ctr_number  
FROM customers_addresses;
```

```
23 SELECT id, address_line_1, address_line_2, city, zip_code, ctr_number  
24 FROM customers_addresses;
```

ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE	CTR_NUMBER
ca0025	65 Acacia Drive	Skins	Liverpool	L165JHR	c00001
ca0032	17 Cartersquare Road	Starford	Liverpool	L169JHK	c00009
ca0035	54 Rapshill Crescent	Georgeloom	Star	ST45AGV	c00010
ca0034	36 Materecra Lane	-	Jump	J023YTH	c00036

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