

## 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

```
CREATE TABLE inventory_list(
    id VARCHAR2(30),
    cost NUMBER(6,2),
    units NUMBER(4),
    CONSTRAINT id_id_pk PRIMARY KEY (id)
);
```

```
CREATE TABLE items(
    item_number VARCHAR2(30),
    name VARCHAR2(20),
    description VARCHAR2(30),
    category VARCHAR2(20),
    color VARCHAR2(10),
    "Size" VARCHAR2(5),
    ilt_id VARCHAR2(30) CONSTRAINT itm_ilt_id_uk UNIQUE,
    CONSTRAINT item_number_pk PRIMARY KEY (item_number)
);
```

```
CREATE TABLE price_history(
    start_date DATE,
    start_time TIMESTAMP,
    price NUMBER(8,2),
    end_date DATE,
    end_time TIMESTAMP,
    item_number VARCHAR2(20),
    CONSTRAINT item_number_fk FOREIGN KEY (item_number)
        REFERENCES items(item_number)
);
```

```
CREATE TABLE sales_rep_addresses(
    id VARCHAR2(5),
    address_line_1 VARCHAR2(30),
    address_line_2 VARCHAR2(30),
    city VARCHAR2(20),
    zip_code VARCHAR2(20),
    CONSTRAINT cra_id_pk PRIMARY KEY (id)
);
```

```
CREATE TABLE sales_representatives (
    id VARCHAR2(5),
    email VARCHAR2(40),
    first_name VARCHAR2(20),
    last_name VARCHAR2(20),
    phone_number VARCHAR2(20),
    commission_rate VARCHAR2(3),
    supervisor_id VARCHAR2(5),
    CONSTRAINT srs_id_pk PRIMARY KEY (id),
    CONSTRAINT srs_supervisor_id_fk FOREIGN KEY (supervisor_id)
        REFERENCES sales_rep_addresses(id)
);
```

```
CREATE TABLE teams (
    id VARCHAR2(5),
    name VARCHAR2(20),
    number_of_players NUMBER(4),
    discount NUMBER(3),
    CONSTRAINT teams_id_pk PRIMARY KEY (id)
);
```

```
CREATE TABLE customers (
    ctr_number VARCHAR2(10),
    email VARCHAR2(30) CONSTRAINT cus_email_uk UNIQUE,
    first_name VARCHAR2(20),
    last_name VARCHAR2(20),
    phone_number VARCHAR2(20),
    current_balance NUMBER(8,2),
    sre_id VARCHAR2(5),
    tum_id VARCHAR2(6) CONSTRAINT cus_tum_id_uk UNIQUE,
    loyalty_card_number VARCHAR2(7) CONSTRAINT cus_loy_card_num_uk UNIQUE,
    CONSTRAINT cus_ctr_number_pk PRIMARY KEY (ctr_number)
);
```

```
CREATE TABLE customer_addresses(
    id VARCHAR2(8),
    address_line_1 VARCHAR2(30),
    address_line_2 VARCHAR2(30),
    city VARCHAR2(20),
    zip_code VARCHAR2(10),
    ctr_number VARCHAR2(10),
    CONSTRAINT cus_id_pk PRIMARY KEY(id)
    CONSTRAINT cus_ctr_number_fk FOREIGN KEY (ctr_number)
        REFERENCES customers(ctr_number)
```

```
CREATE TABLE orders(
    id      VARCHAR2(10),
    odr_date DATE,
    odr_time TIMESTAMP,
    number_of_units NUMBER(4),
    ctr_number VARCHAR(10)
    CONSTRAINT ord_id_pk PRIMARY KEY(id)
    CONSTRAINT ord_ctr_number_fk FOREIGN KEY (ctr_number)
        REFERENCES customers(ctr_number)
);
```

```
CREATE TABLE ordered_items(
    quantity_ordered NUMBER(3),
    quantity_shipped NUMBER(3),
    odr_id VARCHAR2(20),
    itm_number VARCHAR(20),
    CONSTRAINT ordi_odr_id_fk FOREIGN KEY (odr_id)
        REFERENCES orders(id),
    CONSTRAINT ordi_itm_number_fk FOREIGN KEY (itm_number)
        REFERENCES items(itm_number)
);
```

1. `INSERT INTO teams  
VALUES ('t004', 'Jets', 10, 5);`
2. `INSERT INTO customers(ctr_number, email, first_name, last_name, phone_number, current_balance,  
loyalty_card_number)  
VALUES ('c02001', 'brianrog@hootech.com', 'Brian', 'Rogers', '0165456698', -5, 'lc5597');`
3. `UPDATE customers  
SET current_balance = 50  
WHERE current_balance = -5;  
  
SELECT * FROM customers;`

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	-	-	lc1015
c00101	unknown@here.com	John	Doe	03216547888	987.5	sr01	t002	-
c00103	Murcia@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587

## 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.

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	-	-

## 2. UPDATE price\_history

```
SET end_date = CURRENT_DATE,
    end_time = CURRENT_TIMESTAMP
WHERE itm_number = 'im01101048' AND end_time IS NULL;
```

3.

START_DATE	START_TIME	PRICE	END_DATE	END_TIME	ITM_NUMBER
17-JUN-17	17-JUN-16 09.00.00.000000 AM	4.99	-	-	im01101044
25-NOV-16	25-NOV-16 09.00.00.000000 AM	14.99	25-JAN-17	25-JAN-17 05.00.00.000000 PM	im01101045
25-JAN-17	25-JAN-17 05.01.00.000000 PM	8.99	25-JAN-17	25-JAN-17 07.00.00.000000 PM	im01101045
26-JAN-17	26-JAN-17 09.00.00.000000 AM	15.99	-	-	im01101045
12-FEB-17	12-FEB-17 12.30.00.000000 PM	7.99	-	-	im01101046
25-APR-17	25-APR-17 10.10.10.000000 AM	24.99	-	-	im01101047
31-MAY-17	31-MAY-17 04.35.30.000000 PM	149	09-NOV-23	09-NOV-23 12.23.56.432503 PM	im01101048

Download CSV

## 4. INSERT INTO price\_history(start\_date, start\_time, price, itm\_number) VALUES (CURRENT\_DATE, CURRENT\_TIMESTAMP, 99.99, 'im01101048');

5.

START_DATE	START_TIME	PRICE	END_DATE	END_TIME	ITEM_NUMBER
17-JUN-17	17-JUN-16 09.00.00.000000 AM	4.99	-	-	im01101044
25-NOV-16	25-NOV-16 09.00.00.000000 AM	14.99	25-JAN-17	25-JAN-17 05.00.00.000000 PM	im01101045
25-JAN-17	25-JAN-17 05.01.00.000000 PM	8.99	25-JAN-17	25-JAN-17 07.00.00.000000 PM	im01101045
26-JAN-17	26-JAN-17 09.00.00.000000 AM	15.99	-	-	im01101045
12-FEB-17	12-FEB-17 12.30.00.000000 PM	7.99	-	-	im01101046
25-APR-17	25-APR-17 10.10.10.000000 AM	24.99	-	-	im01101047
31-MAY-17	31-MAY-17 04.35.30.000000 PM	149	09-NOV-23	09-NOV-23 12.36.03.910265 PM	im01101048
09-NOV-23	09-NOV-23 12.36.03.913338 PM	99.99	-	-	im01101048

Download CSV

## PART 2

1. DELETE FROM customers\_addresses  
 WHERE address\_line\_1 = '83 Barrhill Drive';

2.

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

Download CSV