

## 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 9 Exercise 1: Joining Tables Using JOIN

##### Write SELECT Statements Using Data From Multiple Tables Using Equijoins and Non-Equijoins (S6L9 Objective 1)

In this exercise you will write SELECT statements to access data from more than one table.

## Part 1: Creating Natural Joins.

1. Display all of the information about sales representatives and their addresses using a natural join.
2. Adapt the query from the previous question to only show the id, first name, last name, address line 1, address line 2, city, email and phone\_number for the sales representatives.

## Part 2: Creating Joins with the USING Clause

1. Adapt the previous query answer to use the USING clause instead of a natural join.
2. Display all of the information about items and their price history by joining the items and price\_history tables.

## Part 3: Creating Joins with the ON Clause

1. Use an ON clause to join the customer and sales representative table so that you display the customer number, customer first name, customer last name, customer phone number, customer email, sales representative id, sales representative first name, sales representative last name and sales representative email. You will need to use a table alias in your answer as both tables have columns with the same name.

## Part 4- Creating Three-Way Joins with the ON Clause

1. Using the answer to Task 3 add a join that will allow the team name that the customer represents to be included in the results.

## Part 5: Applying Additional Conditions to a Join

1. Using the answer to Task 4 add an additional condition to only show the results for the customer that has the number - c00001.

## Part 6: Retrieving Records with Nonequijoins

1. Write a query that will display name and cost of the item with the number im01101045 on the 12<sup>th</sup> of December 2016. The output of the query should look like this:

The cost of the under shirt on this day was 14.99

## PART 1 (creating Natural Joins)

1) SELECT \* FROM sales\_representatives  
NATURAL JOIN sales\_rep\_addresses;

2) SELECT id, first\_name, last\_name, address\_line\_1, address\_line\_2, city, email,  
phone\_name  
FROM sales\_representative  
NATURAL JOIN sales\_rep\_addresses;

## PART 2 (creating Joins with the USING clause)

1) SELECT sr.id, sr.first\_name, sr.last\_name, sra.address\_line\_1, sra.address\_line\_2,  
sr.email, sr.phone\_number  
FROM sales\_representative sr JOIN sales\_rep\_addresses sra  
USING (id);

2) SELECT \* FROM items  
JOIN price\_history  
USING (itm\_number);

## PART 3 (creating Joins with the ON clause)

1) SELECT c.ctr\_number, c.first\_name, c.last\_name, c.phone\_number, c.email, sr.id,  
sr.first\_name, sr.last\_name, sr.email  
FROM customer c JOIN sales\_representative sr  
ON (c.sre\_id = sr.id);

## PART 4 (creating Three-way Joins with the ON clause)

1) SELECT c.ctr\_number, c.first\_name, c.last\_name, c.phone\_number, c.email, sr.id,  
sr.first\_name, sr.last\_name, sr.email  
FROM customer c JOIN sales\_representative sr  
ON (c.sre\_id = sr.id)  
JOIN teams t  
ON (t.id = c.tem\_id);

#### PART 5 (Additional condition to a Joins)

```
1) SELECT c.ctr_number, c.first_name, c.last_name, c.phone_number, c.email, sr.id,  
        sr.first_name, sr.last_name, sr.email  
FROM customer c JOIN sales_representative sr  
ON (c.sre_id = sr.id)  
JOIN teams t  
ON (t.id = c.tem_id)  
WHERE c.ctr_number = "C00001";
```

#### PART 6 (retrieving record with nonequi joins)

```
1) SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price AS "Items and  
        cost information"  
FROM i itm_number = p.itm_number  
WHERE itm_number = 'im01101045'  
AND TO_DATE ('2016-12-12', 'YYYY-MM-DD')  
BETWEEN p.start_date AND p.end_date;
```