

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 7 Exercise 2: Restricting Data Using WHERE

Limit rows using WHERE (S6L7 Objective 1)

In this exercise you will refine the data that is returned in your query by adding a WHERE clause to your SELECT statement.

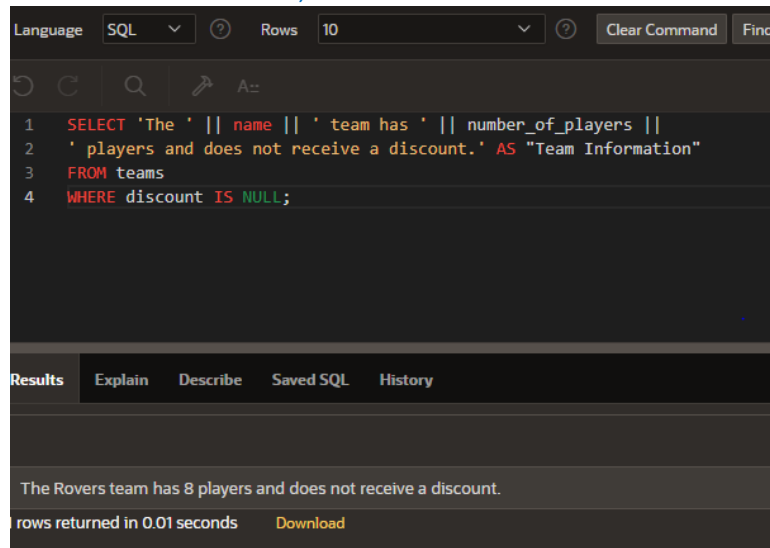
Part 1: Using the NULL Conditions

1. Write a query that will display information for teams that don't receive a discount in the following format:

The Rovers team has 25 players and does not receive a discount.

Use **Team Information** as the column alias.

```
SELECT 'The ' || name || ' team has ' || number_of_players ||  
' players and does not receive a discount.' AS "Team Information"  
FROM teams  
WHERE discount IS NULL;
```



2. Write a query that will display information for only teams that receive a discount in the following format:

The Rockets team has 25 players and receives a discount of 10 percent.

Use **Team Information** as the column alias.

```
SELECT 'The ' || name || ' team has ' || number_of_players ||  
' players and receives a discount of ' || discount || ' percent.' AS "Team Information"  
FROM teams  
WHERE discount IS NOT NULL;
```

The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command' and 'Find Tables'. Below the toolbar is a search bar and a command area. The command area contains the following SQL query:

```

1 SELECT 'The ' || name || ' team has ' || number_of_players ||
2 ' players and receives a discount of ' || discount || ' percent.' AS "Team Information"
3 FROM teams
4 WHERE discount IS NOT NULL;

```

Below the command area, there's a 'Results' tab with sub-tabs for 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with one column, 'Team Information'. The table contains three rows of data:

Team Information
The Rockets team has 25 players and receives a discount of 10 percent.
The Celtics team has 42 players and receives a discount of 20 percent.
The Jets team has 10 players and receives a discount of 5 percent.

At the bottom of the results area, it says '3 rows returned in 0.00 seconds' and there's a 'Download' button.

Part 2: Logical Operators: AND

- Write a query that will display the customer number, address line 1 and postal code for customers that live in the starford area of Liverpool. Use Customer Number, Street Address and Postal Code as the column aliases.

```

SELECT ctr_number "Customer Number",
       address_line_1 "Street Address",
       zip_code "Postal Code"
FROM customers_addresses
WHERE city = 'Liverpool' AND address_line_2 = 'Starford';

```

The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command' and 'Find Tables'. Below the toolbar is a search bar and a command area. The command area contains the following SQL query:

```

1 SELECT ctr_number "Customer Number",
2        address_line_1 "Street Address",
3        zip_code "Postal Code"
4 FROM customers_addresses
5 WHERE city = 'Liverpool' AND address_line_2 = 'Starford';

```

Below the command area, there's a 'Results' tab with sub-tabs for 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with three columns: 'Customer Number', 'Street Address', and 'Postal Code'. The table contains one row of data:

Customer Number	Street Address	Postal Code
c00001	17 Gartsquare Road	LP89JHK

At the bottom of the results area, it says 'rows returned in 0.04 seconds' and there's a 'Download' button.

Part 3: Logical Operators: OR

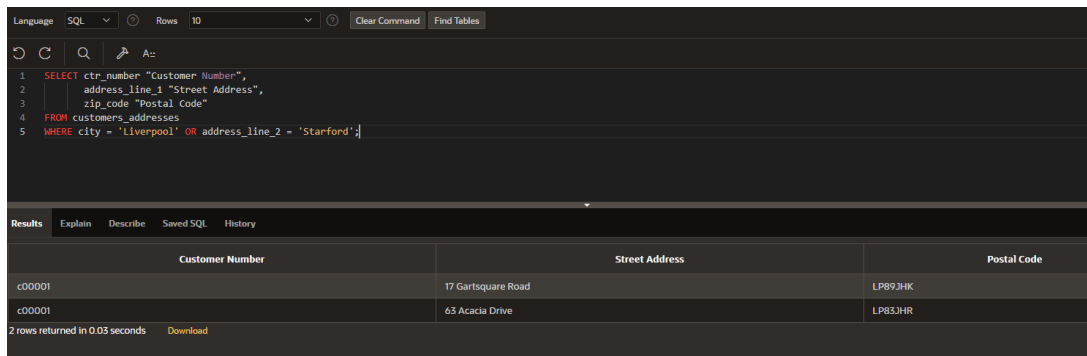
- Write a query that will display the customer number, address line 1 and postal code for customers that live in either starford or Liverpool in general. Use Customer Number, Street Address and Postal Code as the column aliases.

```

SELECT ctr_number "Customer Number",
       address_line_1 "Street Address",
       zip_code "Postal Code"
FROM customers_addresses

```

WHERE city = 'Liverpool' OR address_line_2 = 'Starford';



The screenshot shows the SQL Developer interface. The top bar indicates the language is SQL, with 10 rows returned. The command window contains the following SQL query:

```
1 SELECT ctr_number "Customer Number",
2       address_line_1 "Street Address",
3       zip_code "Postal Code"
4 FROM customers_addresses
5 WHERE city = 'Liverpool' OR address_line_2 = 'Starford';
```

The Results tab is active, displaying a table with 2 rows. The table has three columns: Customer Number, Street Address, and Postal Code.

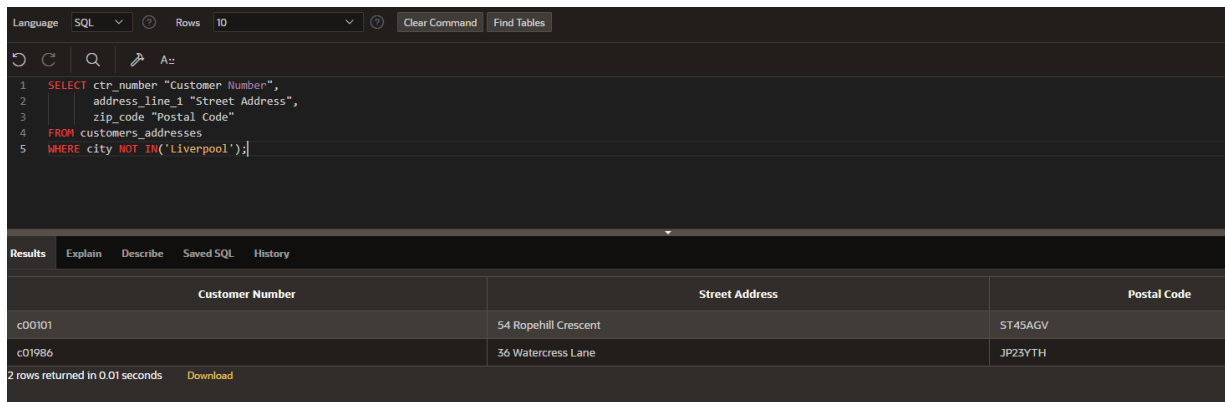
Customer Number	Street Address	Postal Code
c00001	17 Gartsquare Road	LP89JHK
c00001	65 Acacia Drive	LP83JHR

2 rows returned in 0.03 seconds. A Download button is visible at the bottom of the results pane.

Part 4: Logical Operators: NOT Equal To

1. Write a query that will display the customer number, address line 1 and postal code for customers that do not live in Liverpool. Use Customer Number, Street Address and Postal Code as the column aliases.

```
SELECT ctr_number "Customer Number",
       address_line_1 "Street Address",
       zip_code "Postal Code"
FROM customers_addresses
WHERE city NOT IN('Liverpool');
```



The screenshot shows the SQL Developer interface. The top bar indicates the language is SQL, with 10 rows returned. The command window contains the following SQL query:

```
1 SELECT ctr_number "Customer Number",
2       address_line_1 "Street Address",
3       zip_code "Postal Code"
4 FROM customers_addresses
5 WHERE city NOT IN('Liverpool');
```

The Results tab is active, displaying a table with 2 rows. The table has three columns: Customer Number, Street Address, and Postal Code.

Customer Number	Street Address	Postal Code
c00101	54 Ropehill Crescent	ST45AGV
c01986	36 Watercress Lane	JP23YTH

2 rows returned in 0.01 seconds. A Download button is visible at the bottom of the results pane.