

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.

Denies Wong Ke Ying A22EC0047

Section 6 Lesson 7 Exercise 1: Restricting Data Using WHERE

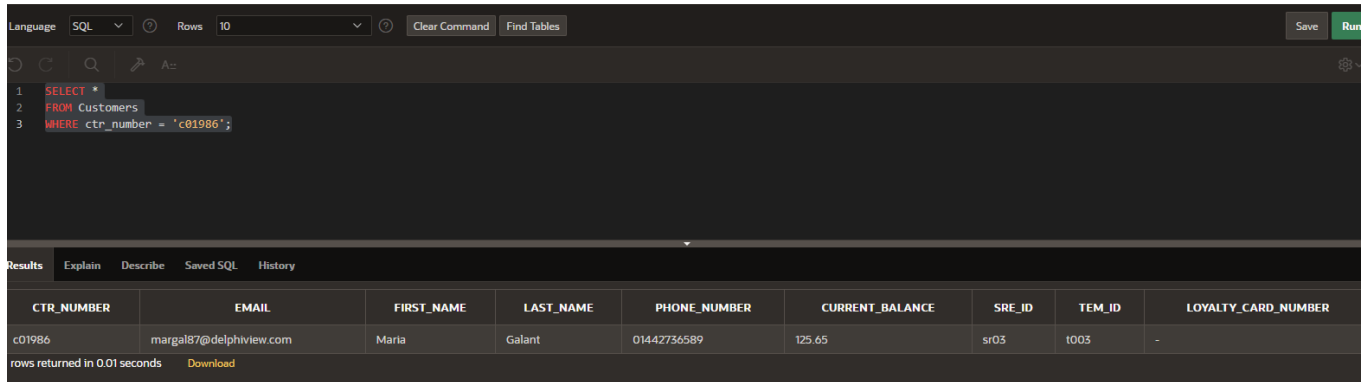
Limit rows using WHERE (S6L7 Objective 1)

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 WHERE Clause.

1. Using the unique customer number in the where clause display all columns for Maria Galant.

```
SELECT *  
FROM Customers  
WHERE ctr_number = 'c01986';
```



The screenshot shows a SQL IDE interface. The top bar includes 'Language SQL', 'Rows 10', and buttons for 'Clear Command' and 'Find Tables'. The main editor area contains the following SQL query:

```
1 SELECT *  
2 FROM Customers  
3 WHERE ctr_number = 'c01986';
```

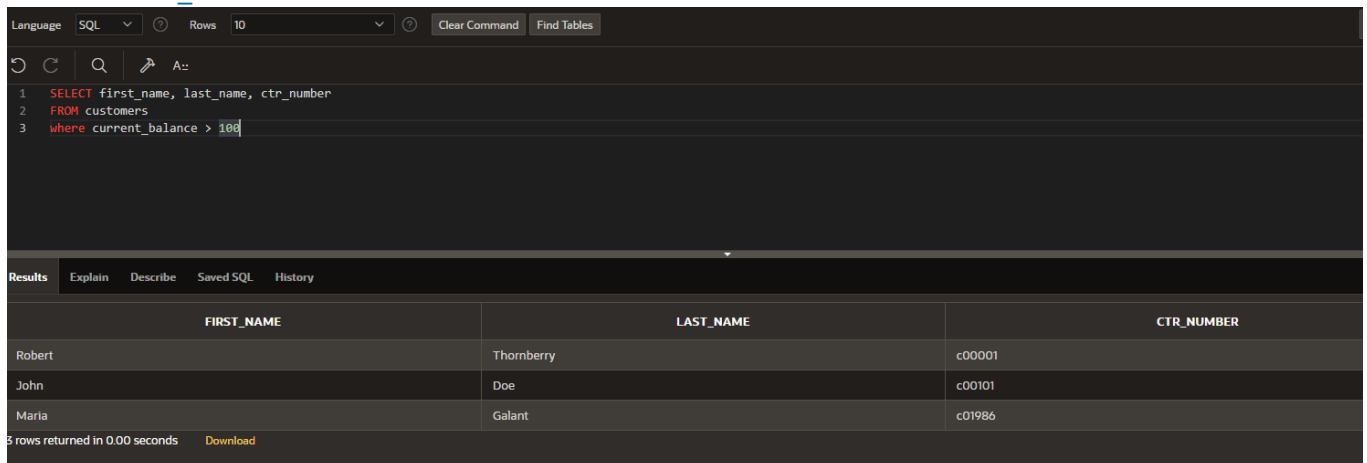
Below the editor, the 'Results' tab is active, displaying a table with the following data:

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	1003	-

At the bottom of the results section, it says 'rows returned in 0.01 seconds' and provides a 'Download' link.

2. Display the first name, last name and customer number for all customers who have a current balance of greater than 100. Use an appropriate alias for your column headings.

```
SELECT first_name, last_name, ctr_number  
FROM customers  
where current_balance > 100
```



The screenshot shows a SQL IDE interface. The top bar includes 'Language SQL', 'Rows 10', and buttons for 'Clear Command' and 'Find Tables'. The main editor area contains the following SQL query:

```
1 SELECT first_name, last_name, ctr_number  
2 FROM customers  
3 where current_balance > 100
```

Below the editor, the 'Results' tab is active, displaying a table with the following data:

FIRST_NAME	LAST_NAME	CTR_NUMBER
Robert	Thornberry	c00001
John	Doe	c00101
Maria	Galant	c01986

At the bottom of the results section, it says '5 rows returned in 0.00 seconds' and provides a 'Download' link.

3. Display the order id, date and time of all orders that were placed before the 28th of May 2019. Use an appropriate alias for your column headings.

```
SELECT id AS "Order ID",  
       odr_date AS "Order Date",  
       odr_time AS "Order Time"  
FROM orders  
WHERE odr_date < TO_DATE('28-May-2019', 'DD-Mon-YYYY');
```

Language: SQL Rows: 10 Clear Command Find Tables

```

1 SELECT id AS "Order ID",
2       odr_date AS "Order Date",
3       odr_time AS "Order Time"
4 FROM orders
5 WHERE odr_date < TO_DATE('28-May-2019', 'DD-Mon-YYYY');

```

Results Explain Describe Saved SQL History

Order ID	Order Date	Order Time
or0101250	04/17/2017	04/17/2017
or0101350	05/24/2017	05/24/2017
or0101425	05/28/2017	05/28/2017
or0101681	06/02/2017	06/02/2017
or0101750	06/18/2017	06/18/2017

5 rows returned in 0.02 seconds Download

Part 2: Range Conditions: BETWEEN Operator

1. Display the inventory id, cost and number of units using appropriate aliases for all items that have a trade cost of between 3.00 and 15.00.

```

SELECT id AS "Inventory ID",
       cost AS "Cost",
       units AS "Number of Units"
FROM inventory_list
WHERE cost BETWEEN 3.00 AND 15.00;

```

Language: SQL Rows: 10 Clear Command Find Tables

```

1 SELECT id AS "Inventory ID",
2       cost AS "Cost",
3       units AS "Number of Units"
4 FROM inventory_list
5 WHERE cost BETWEEN 3.00 AND 15.00;

```

Results Explain Describe Saved SQL History

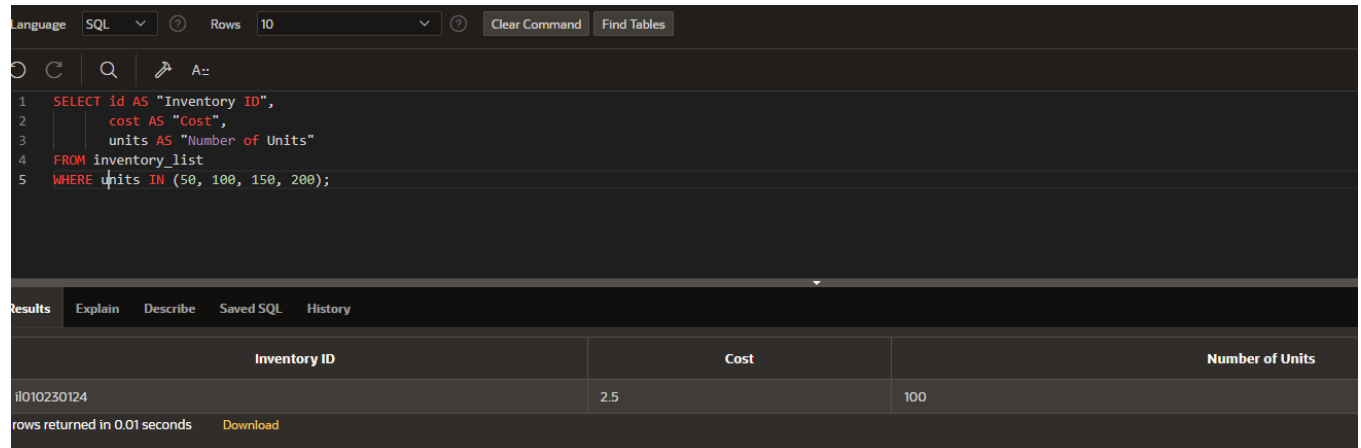
Inventory ID	Cost	Number of Units
il010230125	7.99	250
il010230126	5.24	87

2 rows returned in 0.02 seconds Download

Part 3: Membership Conditions: IN Operator

1. Display the inventory id, cost and number of units using appropriate aliases for all items that have 50, 100, 150 or 200 units in stock.

```
SELECT id AS "Inventory ID",  
       cost AS "Cost",  
       units AS "Number of Units"  
FROM inventory_list  
WHERE units IN (50, 100, 150, 200);
```



The screenshot shows a SQL IDE interface. The top bar indicates the language is SQL, with 10 rows returned. The command area contains the SQL query for Part 3. The results pane shows a single row of data.

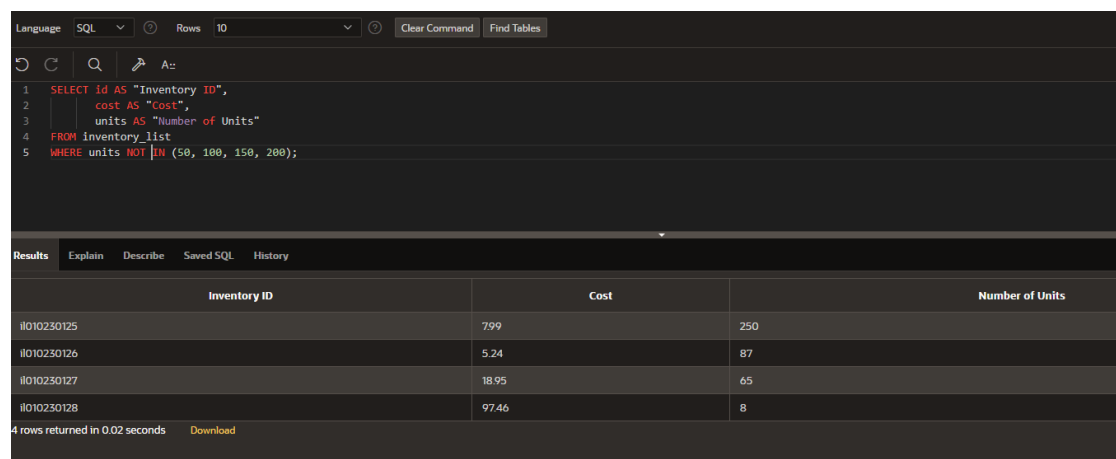
Inventory ID	Cost	Number of Units
il010230124	2.5	100

rows returned in 0.01 seconds [Download](#)

Part 4: Membership Conditions: NOT IN Operator

1. Display the inventory id, cost and number of units using appropriate aliases for all items that do not have 50, 100, 150 or 200 units in stock.

```
SELECT id AS "Inventory ID",  
       cost AS "Cost",  
       units AS "Number of Units"  
FROM inventory_list  
WHERE units NOT IN (50, 100, 150, 200);
```



The screenshot shows a SQL IDE interface. The top bar indicates the language is SQL, with 10 rows returned. The command area contains the SQL query for Part 4. The results pane shows four rows of data.

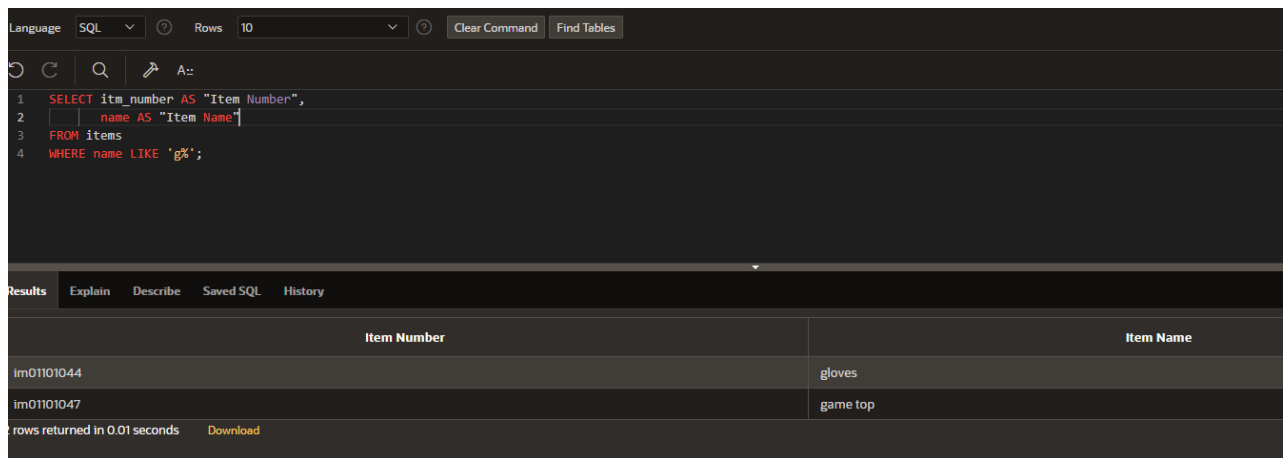
Inventory ID	Cost	Number of Units
il010230125	7.99	250
il010230126	5.24	87
il010230127	18.95	65
il010230128	97.46	8

4 rows returned in 0.02 seconds [Download](#)

Part 5: Pattern Matching: LIKE Operator

1. Display item number and name of all items that have a name that begins with g. Use an appropriate alias for your column headings.

```
SELECT itm_number AS "Item Number",  
       name AS "Item Name"  
FROM items  
WHERE name LIKE 'g%';
```



The screenshot shows a SQL IDE interface. The top bar indicates the language is SQL and the number of rows is 10. The command window contains the following SQL query:

```
1 SELECT itm_number AS "Item Number",  
2     name AS "Item Name"  
3 FROM items  
4 WHERE name LIKE 'g%';
```

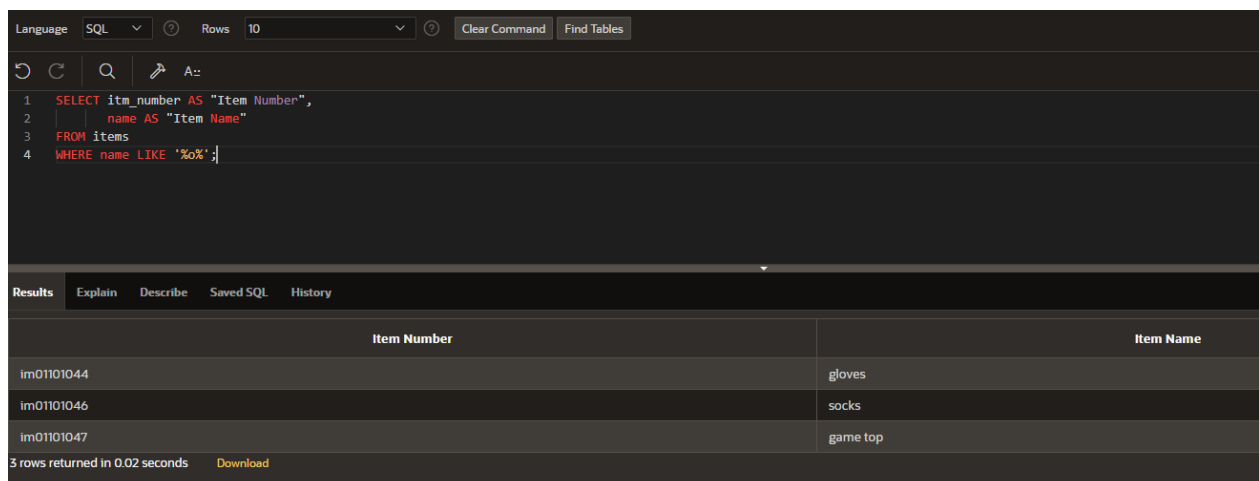
The Results tab is active, displaying a table with two columns: Item Number and Item Name. The results show two rows: one with item number im01101044 and name gloves, and another with item number im01101047 and name game top. The status bar at the bottom indicates that 2 rows were returned in 0.01 seconds.

Item Number	Item Name
im01101044	gloves
im01101047	game top

Part 6 : Pattern Matching: Combining Wildcard Characters with the LIKE Operator

1. Display item number and name of all items that have a name that contain a lowercase o. Use an appropriate alias for your column headings.

```
SELECT itm_number AS "Item Number",  
       name AS "Item Name"  
FROM items  
WHERE name LIKE '%o%';
```



The screenshot shows a SQL IDE interface. The top bar indicates the language is SQL and the number of rows is 10. The command window contains the following SQL query:

```
1 SELECT itm_number AS "Item Number",  
2     name AS "Item Name"  
3 FROM items  
4 WHERE name LIKE '%o%';
```

The Results tab is active, displaying a table with two columns: Item Number and Item Name. The results show three rows: one with item number im01101044 and name gloves, one with item number im01101046 and name socks, and another with item number im01101047 and name game top. The status bar at the bottom indicates that 3 rows were returned in 0.02 seconds.

Item Number	Item Name
im01101044	gloves
im01101046	socks
im01101047	game top