



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SECD2523 - DATABASE

SECTION 10

SQL 3 : DML 2 PART 3

PREPARED FOR:

MADAM ROZILAWATI BINTI DOLLAH @ MD ZAIN

PREPARED BY:

NAME	MATRICS NO
NUR ALIA MAISARAH BINTI WAN ASMIRA	A22EC0242

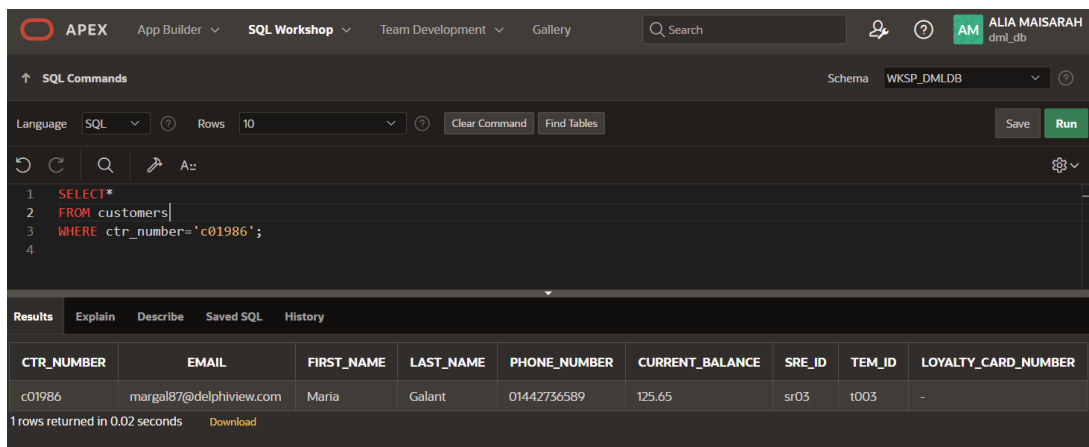
Section 6 Lesson 7 Exercise 1: Restricting Data Using WHERE

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 displays all columns for Maria Galant.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

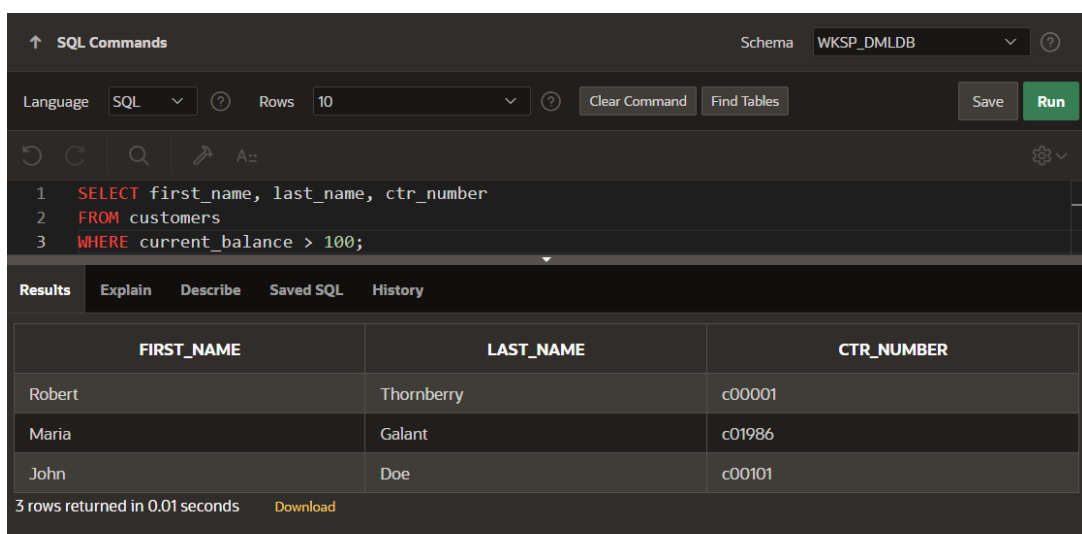
```
1 SELECT *
2 FROM customers
3 WHERE ctr_number='c01986';
4
```

The results table shows one row for Maria Galant with a current balance of 125.65.

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	tt003	-

1 rows returned in 0.02 seconds

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.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

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

The results table shows three rows for customers with a current balance greater than 100.

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

3 rows returned in 0.01 seconds

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.

The screenshot shows a SQL interface with the following query:

```
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 ('28-05-2019', 'DD-mm-YYYY');
```

The results table displays 5 rows of order data:

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

5 rows returned in 0.01 seconds

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.

The screenshot shows a SQL interface with the following query:

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

The results table displays 2 rows of inventory data:

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

2 rows returned in 0.03 seconds

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.

The screenshot shows the SQL Developer interface with the following components:

- SQL Commands** tab: Contains the query:

```
1 SELECT id AS "Inventory ID",  
2 cost AS "Cost", units AS "Number of Units"  
3 FROM inventory_list  
4 WHERE units IN ('50', '100', '150', '200');
```
- Results** tab: Shows the query results in a table with columns: Inventory ID, Cost, and Number of Units.

Inventory ID	Cost	Number of Units
il010230124	2.5	100

1 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.

The screenshot shows the SQL Developer interface with the following components:

- SQL Commands** tab: Contains the query:

```
1 SELECT id AS "Inventory ID",  
2 cost AS "Cost", units AS "Number of Units"  
3 FROM inventory_list  
4 WHERE units NOT IN ('50', '100', '150', '200');
```
- Results** tab: Shows the query results in a table with columns: Inventory ID, Cost, and Number of Units.

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

4 rows returned in 0.00 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.

The screenshot shows the SQL Developer interface with the following details:

- SQL Commands:** Schema: WKSP_DMLDB
- Language:** SQL
- Rows:** 10
- Buttons:** Clear Command, Find Tables, Save, Run
- Query:**

```
1 SELECT itm_number AS "Item Number",name AS "Item Name"
2 FROM items
3 WHERE name LIKE 'g%';
```
- Results:** Explain, Describe, Saved SQL, History
- Table:**

Item Number	Item Name
im01101044	gloves
im01101047	game top
- Footer:** 2 rows returned in 0.01 seconds, Download

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.

The screenshot shows the SQL Developer interface with the following details:

- SQL Commands:** Schema: WKSP_DMLDB
- Language:** SQL
- Rows:** 10
- Buttons:** Clear Command, Find Tables, Save, Run
- Query:**

```
1 SELECT itm_number AS "Item Number",name AS "Item Name"
2 FROM items
3 WHERE name LIKE '%o%';
```
- Results:** Explain, Describe, Saved SQL, History
- Table:**

Item Number	Item Name
im01101044	gloves
im01101046	socks
im01101047	game top
- Footer:** 3 rows returned in 0.01 seconds, Download