



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Lab 3: DML 2 Part 1

SECD2523 - 08 Database

SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Noor Hidayah Zakaria

TAN YUN XI	A22EC0282
------------	-----------

Section 6 Lesson 6 Exercise 1: Retrieving Data Using SELECT

Write and Execute SELECT statements (S6L6 Objective 2)

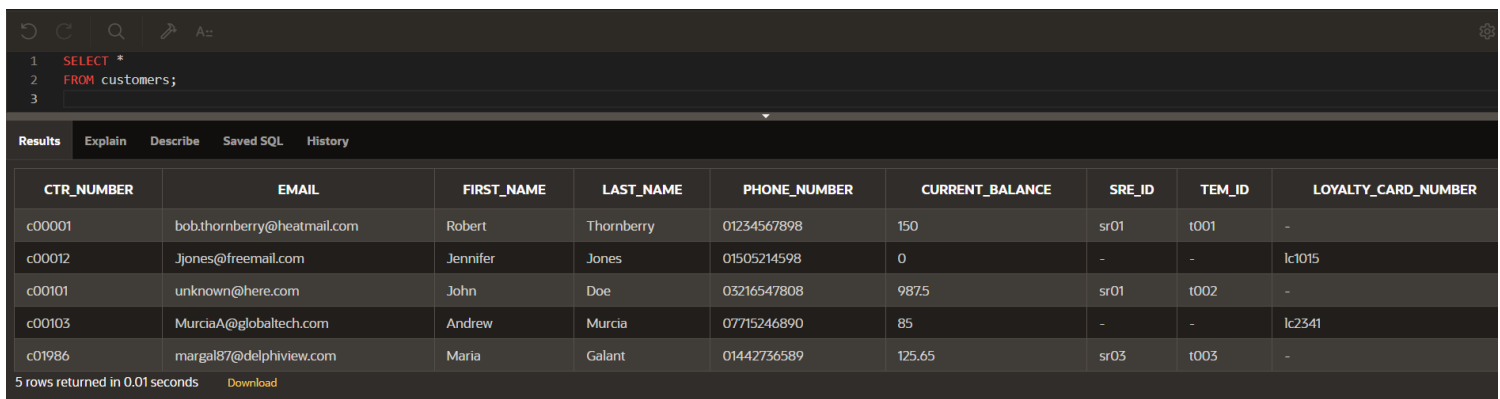
In this exercise you will retrieve data that is stored in the database system by using a SELECT statement.

Part 1: Retrieving all columns from a table.

Using the SELECT * statement show all data stored in the following tables:

1. customers.

```
SELECT *  
FROM customers;
```



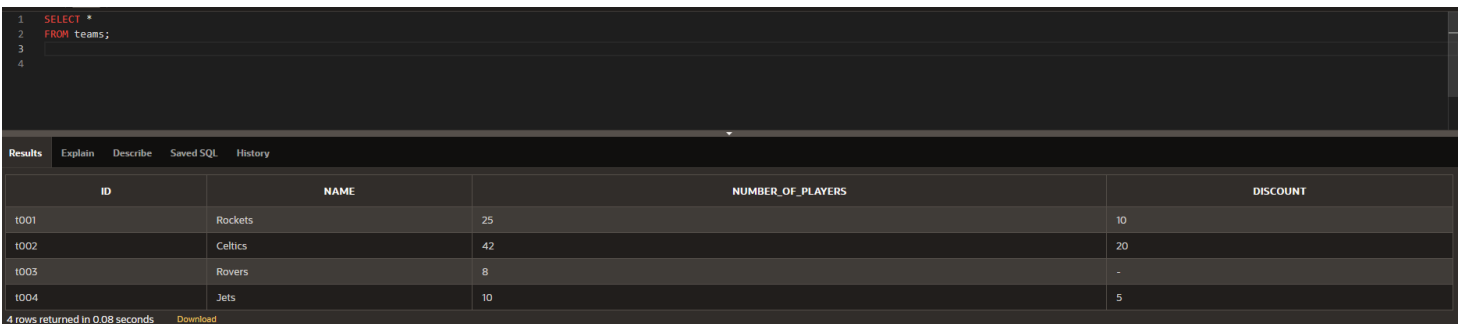
The screenshot shows a SQL query execution interface. The query entered is `SELECT * FROM customers;`. The results are displayed in a table with 9 columns: CTR_NUMBER, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, CURRENT_BALANCE, SRE_ID, TEM_ID, and LOYALTY_CARD_NUMBER. There are 5 rows of data returned.

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	03216547808	987.5	sr01	t002	-
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-

5 rows returned in 0.01 seconds [Download](#)

2. teams.

```
SELECT *  
FROM teams;
```



The screenshot shows a SQL query execution interface. The query entered is `SELECT * FROM teams;`. The results are displayed in a table with 4 columns: ID, NAME, NUMBER_OF_PLAYERS, and DISCOUNT. There are 4 rows of data returned.

ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT
t001	Rockets	25	10
t002	Celtics	42	20
t003	Rovers	8	-
t004	Jets	10	5

4 rows returned in 0.08 seconds [Download](#)

3. items

```
SELECT *  
FROM items;
```

```

1 SELECT *
2 FROM Items;
3

```

ITEM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID
im0101044	gloves	catcher mitt	clothing	brown	m	il010230124
im0101045	under shirt	top worn under the game top	clothing	white	s	il010230125
im0101046	socks	team socks with emblem	clothing	range	l	il010230126
im0101047	game top	team shirt with emblem	clothing	range	m	il010230127
im0101048	premium bat	high quality baseball bat	equipment	-	-	il010230128

5 rows returned in 0.09 seconds [Download](#)

Part 2: Selecting Specific Columns

1. Display the customer number, first name, last name, email and phone number of the customers.

SELECT ctr_number, first_name, last_name,email,phone_number
FROM customers;

```

1 SELECT ctr_number, first_name, last_name,email,phone_number
2 FROM customers;
3

```

CTR_NUMBER	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
c00001	Robert	Thornberry	bob.thornberry@heatmail.com	01234567898
c00012	Jennifer	Jones	Jjones@freemail.com	01505214598
c00101	John	Doe	unknown@here.com	03216547808
c00103	Andrew	Murcia	MurciaA@globaltech.com	07715246890
c01986	Maria	Galant	margal87@delphiview.com	01442736589

5 rows returned in 0.01 seconds [Download](#)

2. Display the name and number of players for each team.

SELECT name, number_of_players
FROM teams;

```

1 SELECT name, number_of_players
2 FROM teams;

```

NAME	NUMBER_OF_PLAYERS
Rockets	25
Celtics	42
Rovers	8
Jets	10

4 rows returned in 0.00 seconds [Download](#)

3. Display the name, description and category for every item in the table.

```
SELECT name, description, category
FROM items;
```

1 SELECT name, description, category

2 FROM items;

3 |

Results

ExplainDescribeSaved SQLHistory

NAME	DESCRIPTION	CATEGORY
gloves	catcher mitt	clothing
under shirt	top worn under the game top	clothing
socks	team socks with emblem	clothing
game top	team shirt with emblem	clothing
premium bat	high quality baseball bat	equipment

5 rows returned in 0.01 secondsDownload



Lab 3: DML 2 Part 2

SECD2523 - 08 Database

SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Noor Hidayah Zakaria

TAN YUN XI	A22EC0282
------------	-----------

```
1 SELECT first_name, last_name, ctr_number, current_balance, current_balance-5
2 FROM customers;
```

Results Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME	CTR_NUMBER	CURRENT_BALANCE	CURRENT_BALANCE-5
Robert	Thornberry	c00001	150	145
Jennifer	Jones	c00012	0	-5
John	Doe	c00101	987.5	982.5
Andrew	Murcia	c00103	85	80
Maria	Galant	c01986	125.65	120.65

5 rows returned in 0.01 seconds [Download](#)

- The current balance cannot be zero value**

1. You previously wrote a query that display the customer's first name, last name, current balance and monthly payment. Rewrite the query to use First Name, Last Name, Balance and Monthly Repayments as the column aliases. The aliases are to be shown exactly as described (case sensitive).

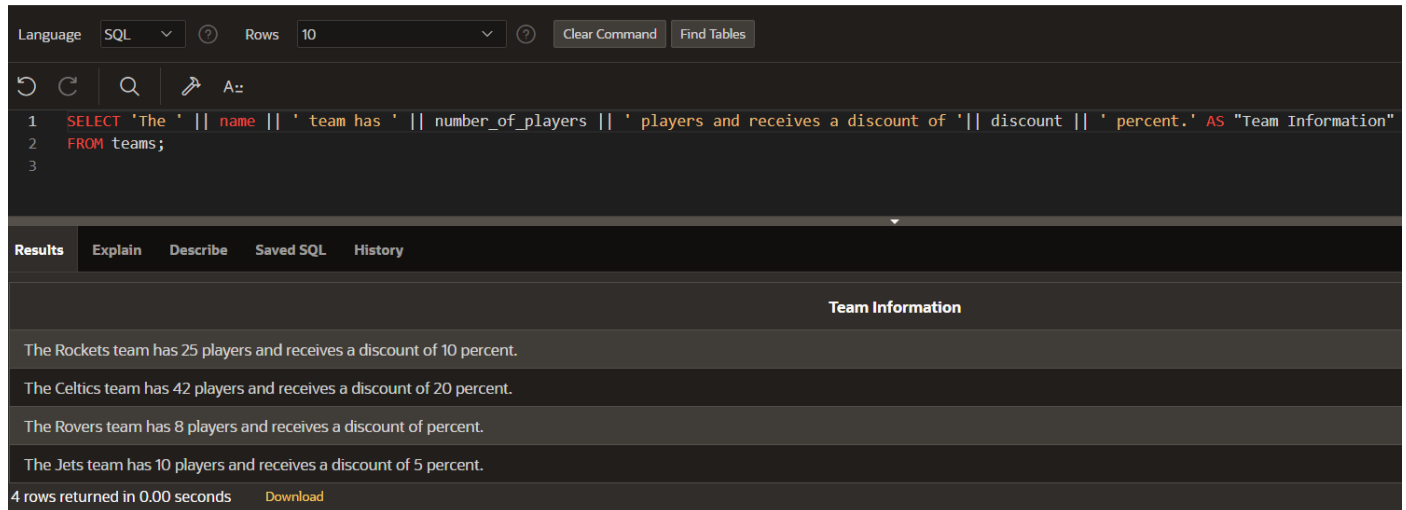
```
SELECT first_name AS "First Name", last_name AS "Last Name", current_balance AS "Balance",  
current_balance/12 AS "Monthly Repayments"  
FROM customers;
```

[illegible]

Part 3: Using Literal Character Strings

- Write a query that will display the team information 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;
```



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 command area with the SQL query entered. The query is: `SELECT 'The ' || name || ' team has ' || number_of_players || ' players and receives a discount of ' || discount || ' percent.' AS "Team Information" FROM teams;`. Below the command area is a results tab with sub-tabs: 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with one column titled 'Team Information'. The table contains four rows of data. The first row is 'The Rockets team has 25 players and receives a discount of 10 percent.', the second is 'The Celtics team has 42 players and receives a discount of 20 percent.', the third is 'The Rovers team has 8 players and receives a discount of percent.', and the fourth is 'The Jets team has 10 players and receives a discount of 5 percent.'. At the bottom of the results area, it says '4 rows returned in 0.00 seconds' and there is a 'Download' button.

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 Rovers team has 8 players and receives a discount of percent.
The Jets team has 10 players and receives a discount of 5 percent.

4 rows returned in 0.00 seconds [Download](#)

- Why does the last team not show a discount?
It contains the zero value, which means it does not get any discount



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Lab 3: DML 2 Part 3

SECD2523 - 08 Database

SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Noor Hidayah Zakaria

TAN YUN XI	A22EC0282
------------	-----------

Section 6 Lesson 7 Exercise 1: Restricting Data Using WHERE

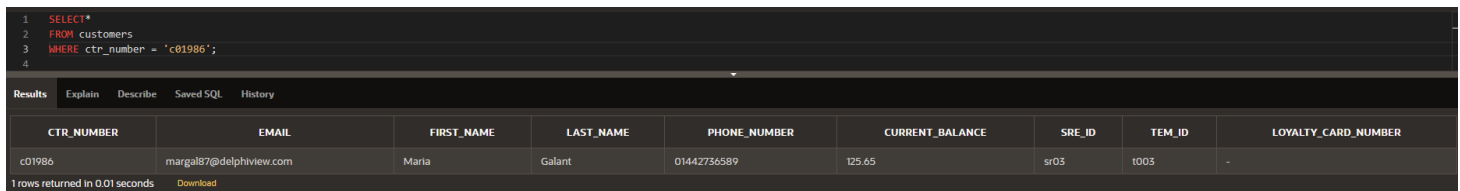
Limit rows using WHERE (S6L7 Objective 1)

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 query editor with the following query:

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

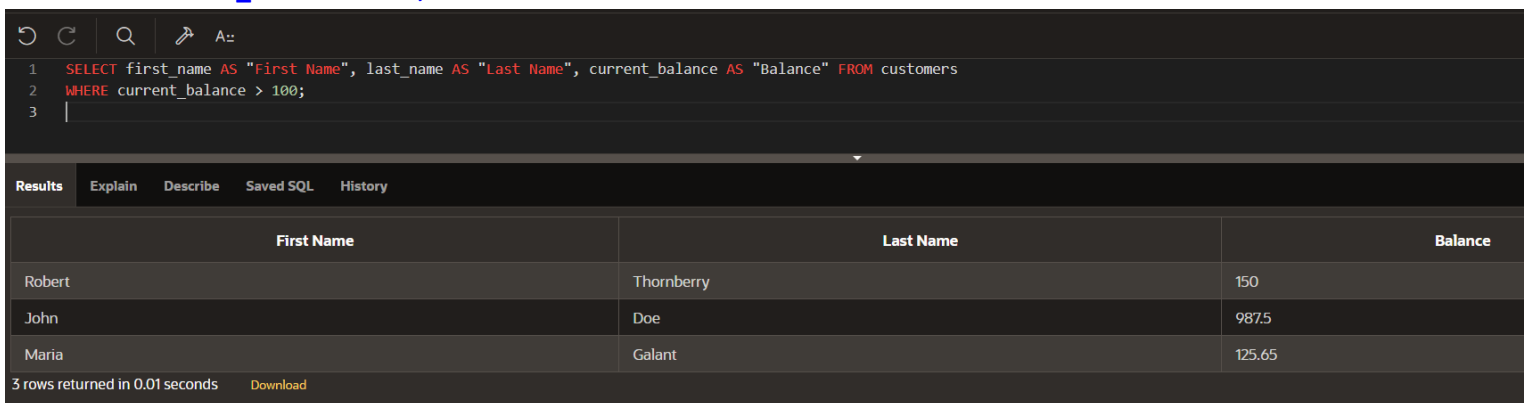
Below the query editor, the results are displayed in a table with the following columns: CTR_NUMBER, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, CURRENT_BALANCE, SRE_ID, TEM_ID, and LOYALTY_CARD_NUMBER. The results show one row for Maria Galant with CTR_NUMBER c01986.

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	t003	-

1 rows returned in 0.01 seconds [Download](#)

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 AS "First Name", last_name AS "Last Name", current_balance AS "Balance"  
FROM customers  
WHERE current_balance > 100;
```



The screenshot shows a SQL query editor with the following query:

```
1 SELECT first_name AS "First Name", last_name AS "Last Name", current_balance AS "Balance" FROM customers  
2 WHERE current_balance > 100;  
3
```

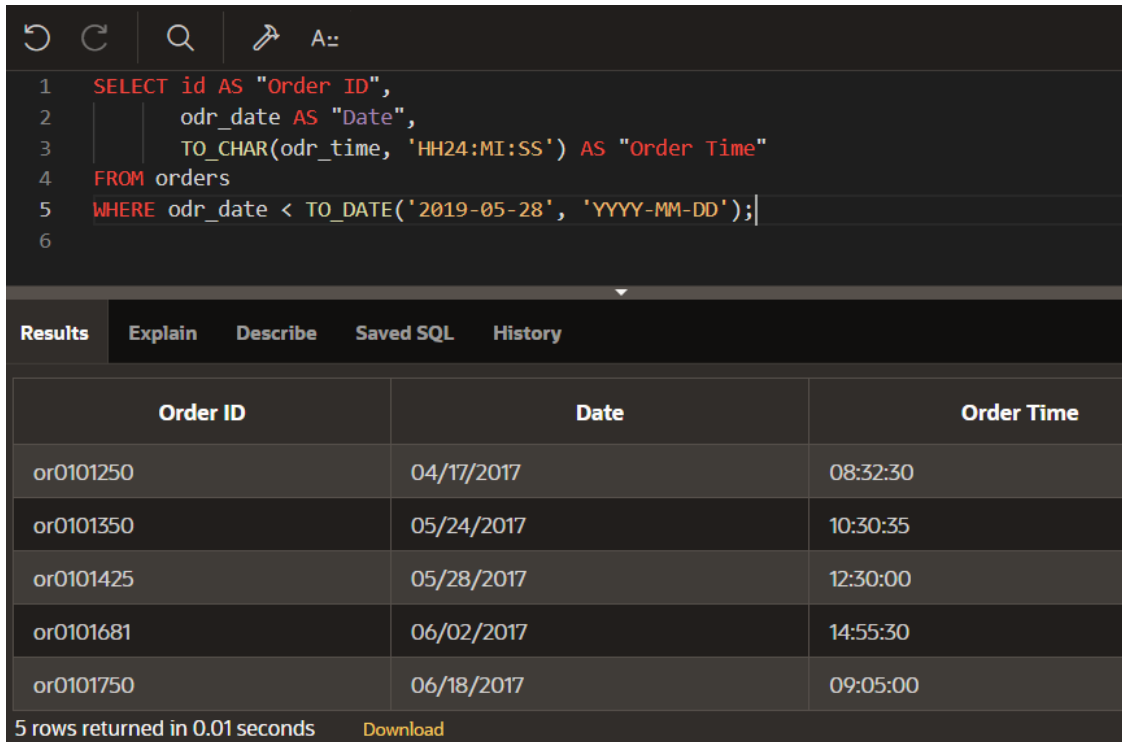
Below the query editor, the results are displayed in a table with the following columns: First Name, Last Name, and Balance. The results show three rows: Robert Thornberry, John Doe, and Maria Galant.

First Name	Last Name	Balance
Robert	Thornberry	150
John	Doe	987.5
Maria	Galant	125.65

3 rows returned in 0.01 seconds [Download](#)

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 "Date",  
       TO_CHAR(odr_time, 'HH24:MI:SS') AS "Order Time"  
FROM orders  
WHERE odr_date < TO_DATE('2019-05-28', 'YYYY-MM-DD');
```



The screenshot shows a SQL IDE interface. The top bar contains icons for undo, redo, search, and a command prompt. The main editor area displays the SQL query from the previous block. Below the editor is a tabbed interface with 'Results' selected. The results are shown in a table with three columns: 'Order ID', 'Date', and 'Order Time'. There are five rows of data. At the bottom of the results section, it says '5 rows returned in 0.01 seconds' and provides a 'Download' link.

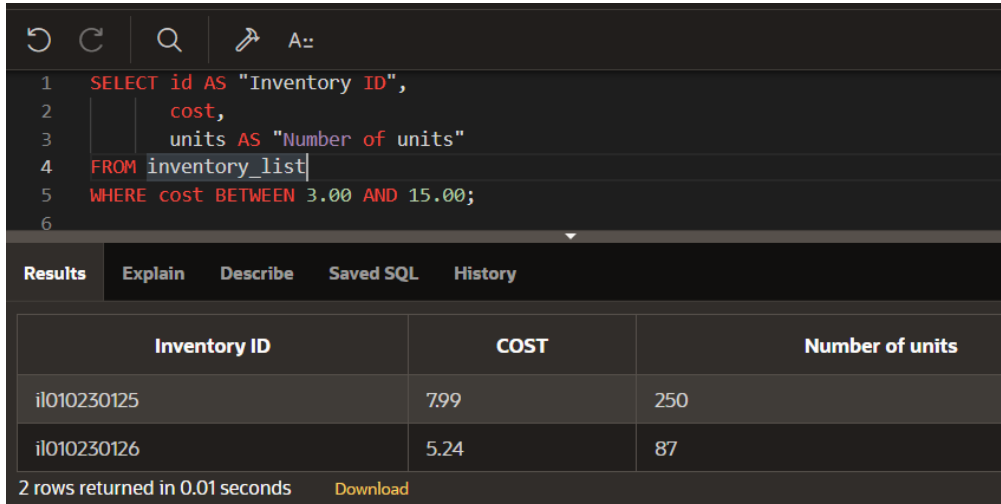
Order ID	Date	Order Time
or0101250	04/17/2017	08:32:30
or0101350	05/24/2017	10:30:35
or0101425	05/28/2017	12:30:00
or0101681	06/02/2017	14:55:30
or0101750	06/18/2017	09:05:00

5 rows returned in 0.01 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,  
       units AS "Number of units"  
FROM inventory_list  
WHERE cost BETWEEN 3.00 AND 15.00;
```



The screenshot shows a SQL query editor with a dark theme. The query is entered in the editor, and the results are displayed in a table below. The table has three columns: 'Inventory ID', 'COST', and 'Number of units'. There are two rows of data. The first row has 'il010230125' for the ID, '7.99' for the cost, and '250' for the units. The second row has 'il010230126' for the ID, '5.24' for the cost, and '87' for the units. The status bar at the bottom indicates '2 rows returned in 0.01 seconds' and provides a 'Download' link.

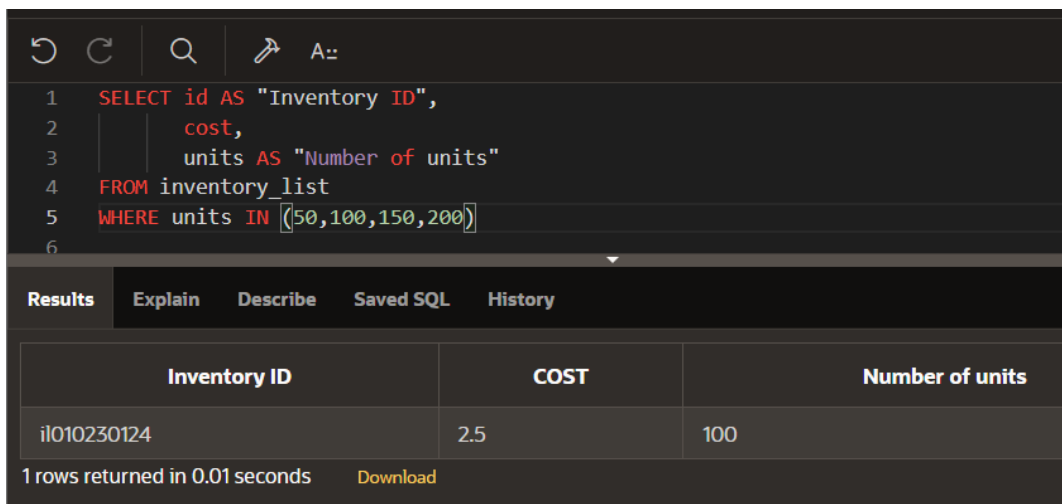
Inventory ID	COST	Number of units
il010230125	7.99	250
il010230126	5.24	87

2 rows returned in 0.01 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,  
       units AS "Number of units"  
FROM inventory_list  
WHERE units IN (50,100,150,200)
```



The screenshot shows a SQL query editor with a dark theme. The query is entered in the editor, and the results are displayed in a table below. The table has three columns: 'Inventory ID', 'COST', and 'Number of units'. There is one row of data. The first row has 'il010230124' for the ID, '2.5' for the cost, and '100' for the units. The status bar at the bottom indicates '1 rows returned in 0.01 seconds' and provides a 'Download' link.

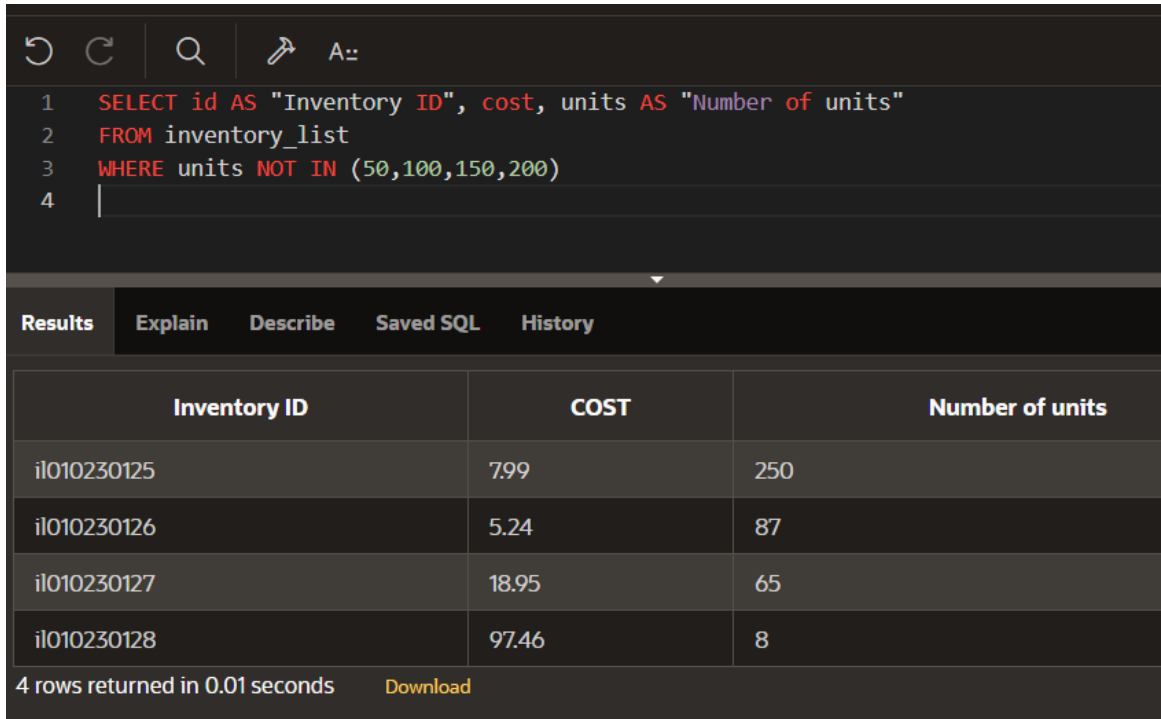
Inventory ID	COST	Number of units
il010230124	2.5	100

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.

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



The screenshot shows a SQL query editor with a dark theme. The query is: `SELECT id AS "Inventory ID", cost, units AS "Number of units" FROM inventory_list WHERE units NOT IN (50,100,150,200)`. Below the editor is a results table with three columns: "Inventory ID", "COST", and "Number of units". The table contains four rows of data. At the bottom, it says "4 rows returned in 0.01 seconds" and has a "Download" link.

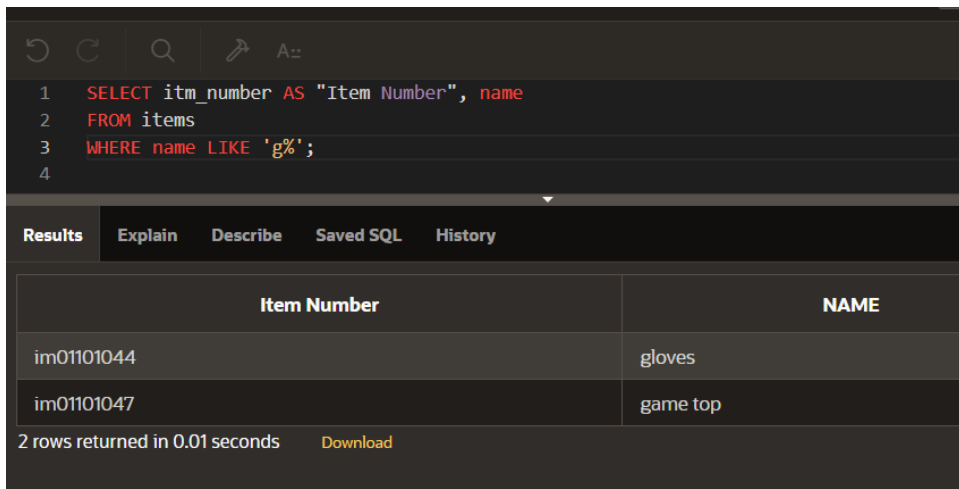
Inventory ID	COST	Number of units
il010230125	7.99	250
il010230126	5.24	87
il010230127	18.95	65
il010230128	97.46	8

4 rows returned in 0.01 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
FROM items
WHERE name LIKE 'g%';
```



The screenshot shows a SQL query editor with a dark theme. The query is: `SELECT itm_number AS "Item Number", name FROM items WHERE name LIKE 'g%';`. Below the editor is a results table with two columns: "Item Number" and "NAME". The table contains two rows of data. At the bottom, it says "2 rows returned in 0.01 seconds" and has a "Download" link.

Item Number	NAME
im01101044	gloves
im01101047	game top

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.

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

</



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Lab 3: DML 2 Part 4

SECD2523 - 08 Database

SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Noor Hidayah Zakaria

TAN YUN XI	A22EC0282
------------	-----------

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.

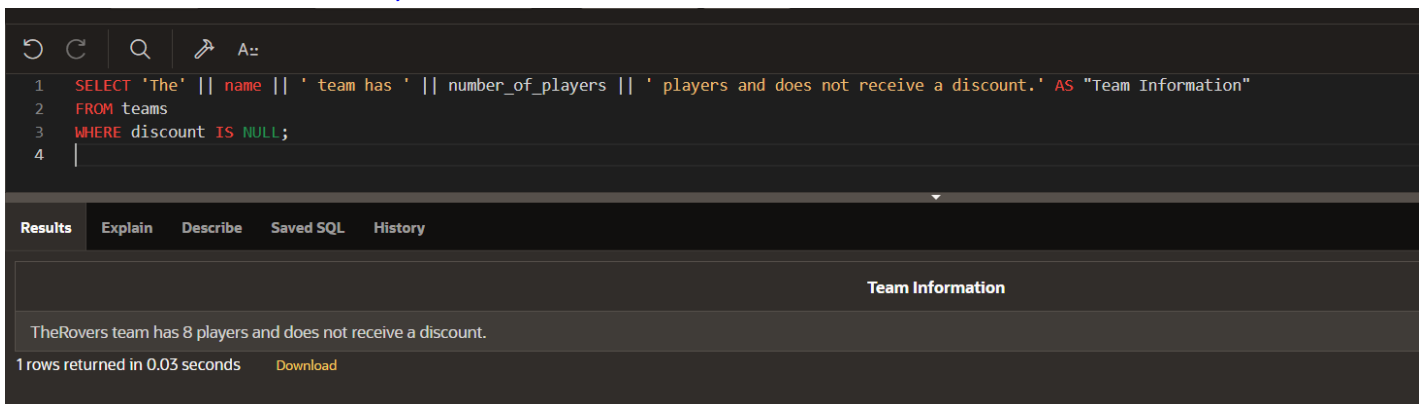
Part 1: Using the NULL Conditions

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;
```



The screenshot shows a SQL query editor with the following code:

```
1 SELECT 'The' || name || ' team has ' || number_of_players || ' players and does not receive a discount.' AS "Team Information"  
2 FROM teams  
3 WHERE discount IS NULL;  
4 |
```

Below the editor is a results viewer with tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is active, showing a single row of data:

Team Information
TheRovers team has 8 players and does not receive a discount.

At the bottom, it indicates "1 rows returned in 0.03 seconds" and provides a "Download" link.

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 10 percent.' AS "Team Information"
FROM teams
WHERE discount IS NOT NULL;
```

```
1 SELECT 'The ' || name || ' team has ' || number_of_players || ' players and receives a discount of 10 percent.' AS "Team Information"
2 FROM teams
3 WHERE discount IS NOT NULL;
4
```

Results	Explain	Describe	Saved SQL	History
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 10 percent.				
The Jets team has 10 players and receives a discount of 10 percent.				
3 rows returned in 0.01 seconds Download				

Part 2: Logical Operators: AND

1. 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 AS "Customer Number", Address_line_1 AS "Street Address", zip_code AS "Postal Code"
FROM customers_addresses
WHERE city = 'Liverpool' AND address_line_2 = 'Starford';
```

```
1 SELECT ctr_number AS "Customer Number", Address_line_1 AS "Street Address", zip_code AS "Postal Code"
2 FROM customers_addresses
3 WHERE city = 'Liverpool' AND address_line_2 = 'Starford';
4
```

Results

Explain

Describe

Saved SQL

History

Customer Number	Street Address	Postal Code
c00001	17 Gartsquare Road	LP89JHK

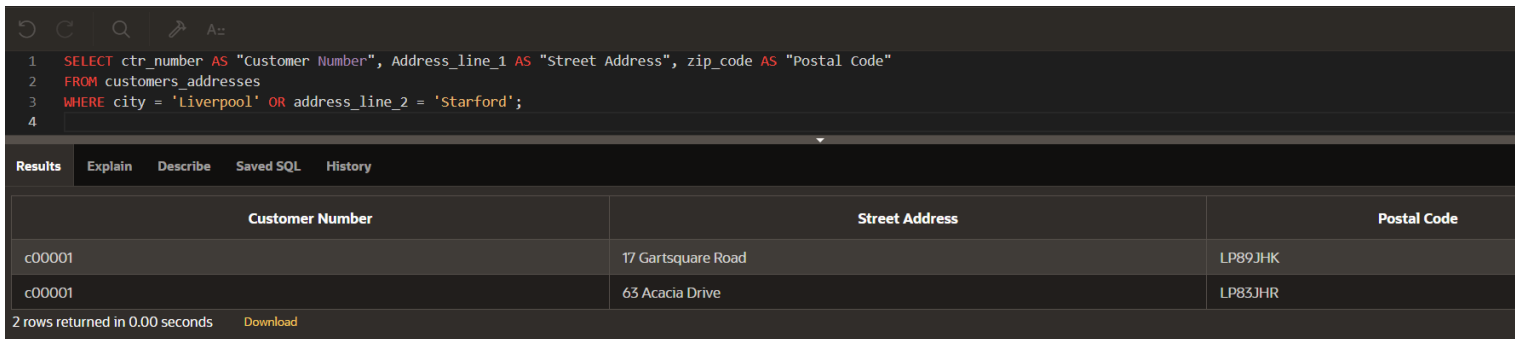
1 rows returned in 0.05 seconds

Download

Part 3: Logical Operators: OR

1. 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 AS "Customer Number", Address_line_1 AS "Street Address", zip_code AS "Postal Code"
FROM customers_addresses
WHERE city = 'Liverpool' OR address_line_2 = 'Starford';
```



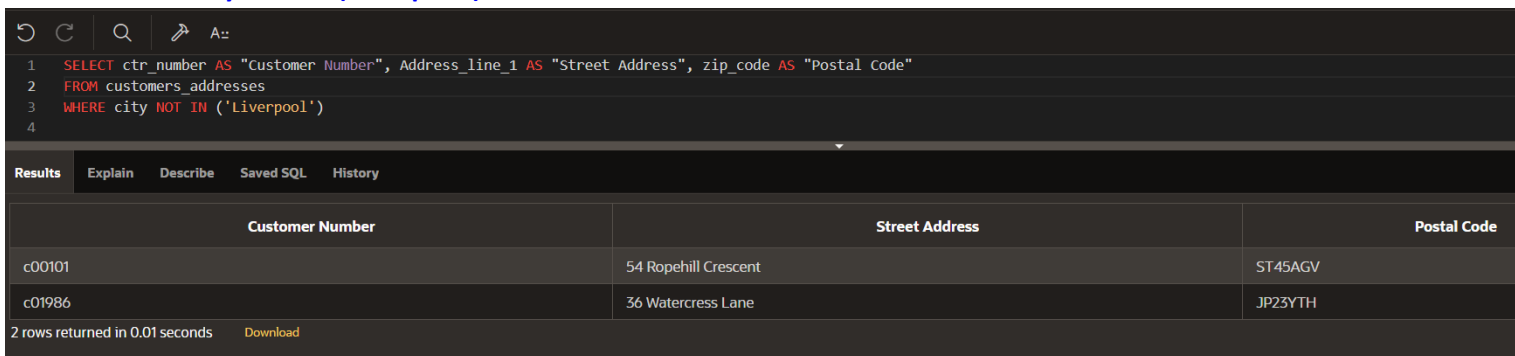
The screenshot shows the SQL Developer interface with the query from Part 3 executed. The 'Results' tab is active, displaying a table with three columns: Customer Number, Street Address, and Postal Code. Two rows of data are shown, corresponding to the two cities specified in the WHERE clause. The status bar at the bottom indicates '2 rows returned in 0.00 seconds'.

Customer Number	Street Address	Postal Code
c00001	17 Gartsquare Road	LP89JHK
c00001	63 Acacia Drive	LP83JHR

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 AS "Customer Number", Address_line_1 AS "Street Address", zip_code AS "Postal Code"
FROM customers_addresses
WHERE city NOT IN ('Liverpool')
```



The screenshot shows the SQL Developer interface with the query from Part 4 executed. The 'Results' tab is active, displaying a table with three columns: Customer Number, Street Address, and Postal Code. Two rows of data are shown, representing customers who do not live in Liverpool. The status bar at the bottom indicates '2 rows returned in 0.01 seconds'.

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



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Lab 3: DML 2 Part 5

SECD2523 - 08 Database

SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Noor Hidayah Zakaria

TAN YUN XI	A22EC0282
------------	-----------

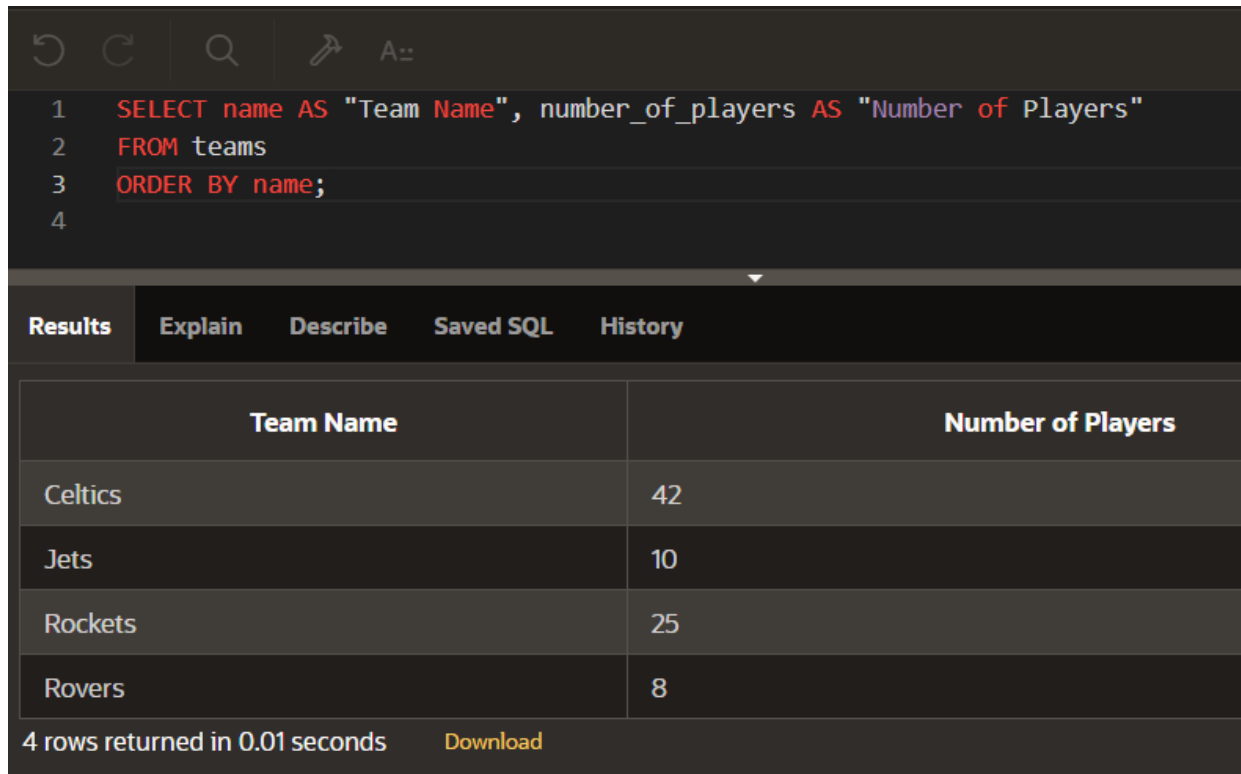
Section 6 Lesson 8 Exercise 1: Sorting Data Using ORDER BY

Use the ORDER BY Clause to Sort SQL Results (S6L8 Objective 1)

In this exercise you will sort the order of the data that is returned in your query by adding an ORDER BY clause to the end of your SELECT statement.

1. Display the team name and number of players alphabetically in order of team name. Use an appropriate alias for your column headings.

```
SELECT name AS "Team Name", number_of_players AS "Number of Players"  
FROM teams  
ORDER BY name;
```



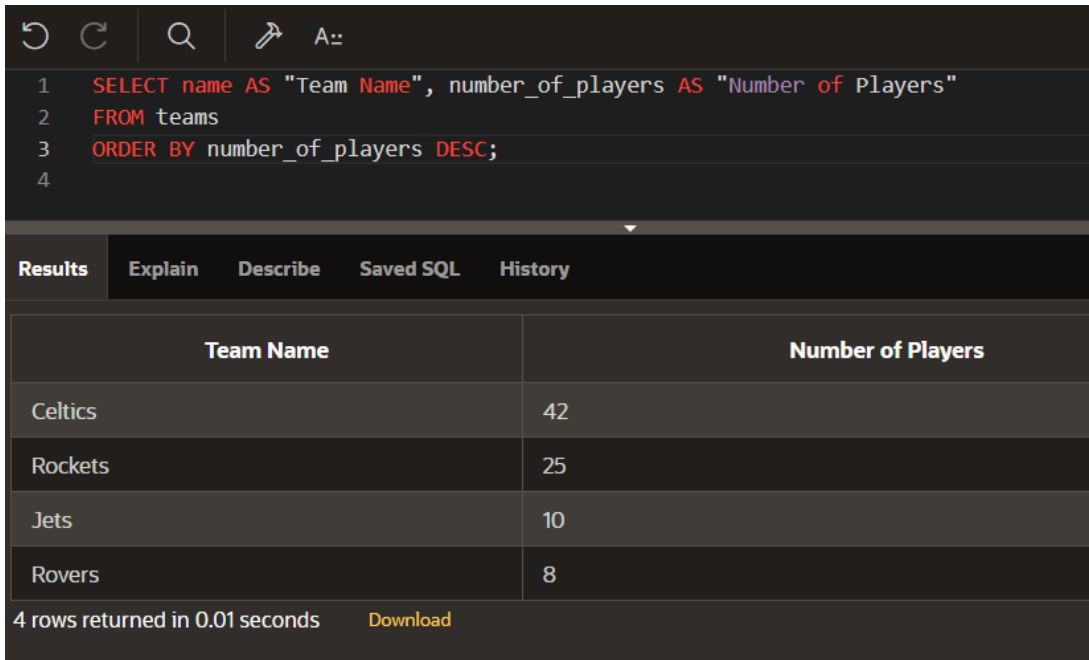
The screenshot shows a SQL IDE interface. At the top, there is a toolbar with icons for undo, redo, search, and a dropdown menu. Below the toolbar, the SQL query is entered in a text area. The query is: `SELECT name AS "Team Name", number_of_players AS "Number of Players" FROM teams ORDER BY name;`. Below the query area, there is a tabbed interface with four tabs: "Results", "Explain", "Describe", and "History". The "Results" tab is selected, and it displays a table with two columns: "Team Name" and "Number of Players". The table contains four rows of data: "Celtics" with 42 players, "Jets" with 10 players, "Rockets" with 25 players, and "Rovers" with 8 players. At the bottom of the "Results" tab, it says "4 rows returned in 0.01 seconds" and there is a "Download" button.

Team Name	Number of Players
Celtics	42
Jets	10
Rockets	25
Rovers	8

4 rows returned in 0.01 seconds [Download](#)

2. Display the team name and number of players in descending order of number of players. Use an appropriate alias for your column headings.

```
SELECT name AS "Team Name", number_of_players AS "Number of Players"
FROM teams
ORDER BY number_of_players DESC;
```



The screenshot shows a SQL query editor with the following query:

```
1 SELECT name AS "Team Name", number_of_players AS "Number of Players"
2 FROM teams
3 ORDER BY number_of_players DESC;
4
```

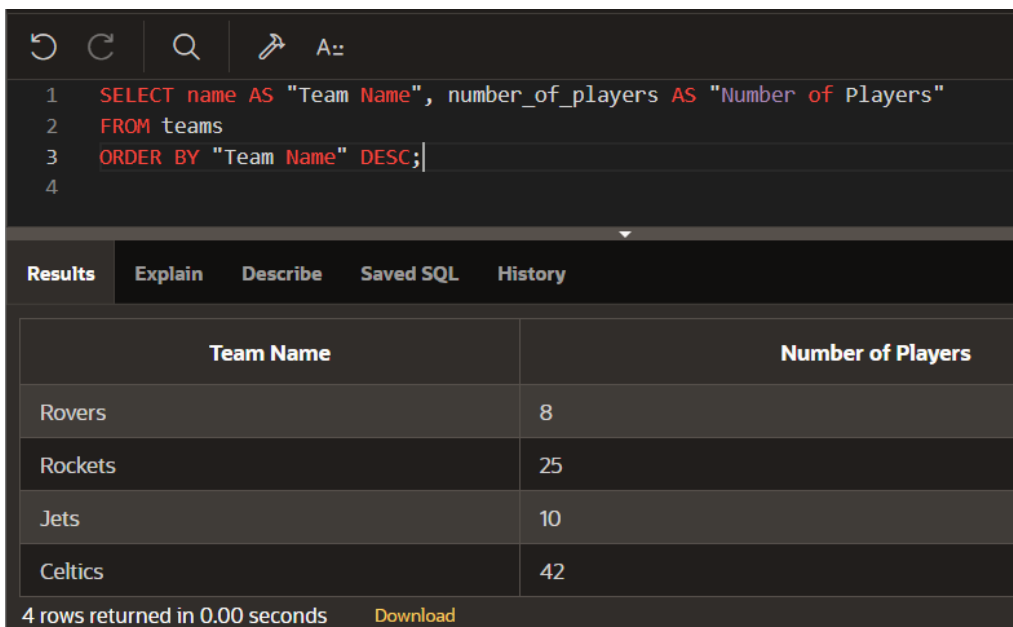
Below the query editor is a results table with the following structure:

Team Name	Number of Players
Celtics	42
Rockets	25
Jets	10
Rovers	8

At the bottom of the results table, it says "4 rows returned in 0.01 seconds" and there is a "Download" link.

3. Display the team name and number of players alphabetically in order of team name. Use Team Name for the name alias and Players for the number of players. Sort the output in descending order of name using the alias in the ORDER BY clause.

```
SELECT name AS "Team Name", number_of_players AS "Number of Players"
FROM teams
ORDER BY "Team Name" DESC;
```



The screenshot shows a SQL query editor with the following query:

```
1 SELECT name AS "Team Name", number_of_players AS "Number of Players"
2 FROM teams
3 ORDER BY "Team Name" DESC;
4
```

Below the query editor is a results table with the following structure:

Team Name	Number of Players
Rovers	8
Rockets	25
Jets	10
Celtics	42

At the bottom of the results table, it says "4 rows returned in 0.00 seconds" and there is a "Download" link.



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Lab 3: DML 2 Part 6

SECD2523 - 08 Database

SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Noor Hidayah Zakaria

TAN YUN XI	A22EC0282
------------	-----------

Section 6 Lesson 8 Exercise 2: Sorting Data Using ORDER BY

Part 1 : TOP-N-ANALYSIS (S6L8 Objective 3)

1. The customers are numbered sequentially with each new customer being assigned a higher customer number.
Use TOP-N-ANALYSIS to only show the **First and last name** of the **first three customers**. Show the customers first and last name in the same column using Customer Name as the column alias.

```
SELECT first_name || ' ' || last_name AS "Customer Name"
FROM (
    SELECT
        first_name,
        last_name
    FROM customers
    ORDER BY ctr_number DESC
)
WHERE ROWNUM <= 3;
```

```
1  SELECT first_name || ' ' || last_name AS "Customer Name"
2  FROM (
3      SELECT
4          first_name,
5          last_name
6      FROM customers
7      ORDER BY ctr_number DESC
8  )
9  WHERE ROWNUM <= 3;
10
```

Customer Name
Maria Galant
Andrew Murcia
John Doe

3 rows returned in 0.01 seconds [Download](#)

Part 2 : Using a Substitution Variable (S6L8 Objective 4)

1. Use a substitution variable that will allow you to **enter the commission rate for the sales representatives**. The **first and last names** should be displayed to screen for any sales representatives that earn that commission rate and the output should be **ordered by their last name**. Use an appropriate alias for your column headings.

```
SELECT first_name AS "First Name", last_name AS "Last Name", commission_rate AS "Commision Rate"
FROM sales_representatives
WHERE commission_rate= :commission_rate
ORDER BY last_name;
```

Bind Variable	Value
:COMMISSION_RATE	<input type="text" value="5"/>

SQL Commands

Schema

Language SQL Rows 10 Clear Command Find Tables

```
1 SELECT first_name AS "First Name", last_name AS "Last Name", commission_rate AS "Commision Rate"
2 FROM sales_representatives
3 WHERE commission_rate= :commission_rate
4 ORDER BY last_name;
5
```

Results Explain Describe Saved SQL History

First Name	Last Name	Commision Rate
Barry	Speed	5
Victoria	Wright	5

2 rows returned in 0.01 seconds Download