



SECD2523 – DATABASE

SEMESTER 1/20232024

SECTION 08

LAB 3: DML2

LECTURER: DR. NOOR HIDAYAH BINTI ZAKARIA

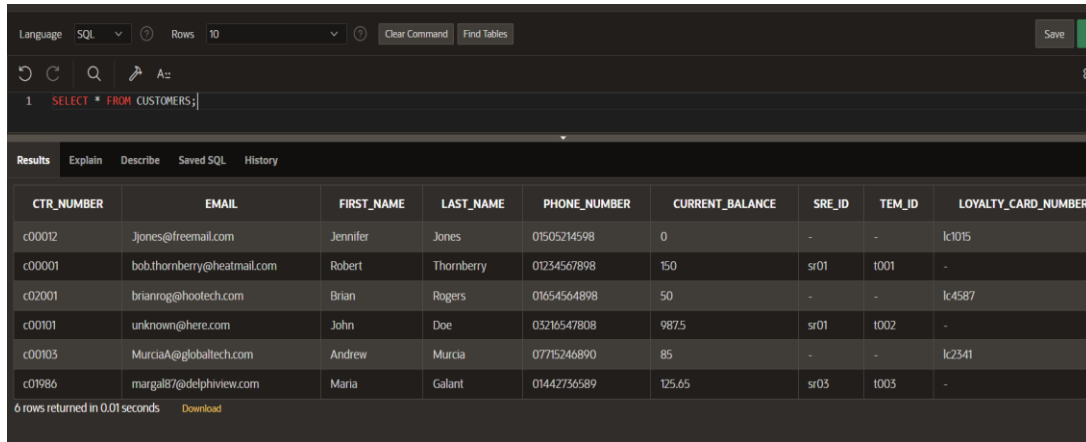
NAME	MATRIC NUMBER
<i>ABDULRAHMAN ABDULLAH AHMED DANAY</i>	<i>A22EC4001</i>

PART 1:

Retrieving all columns from a table.

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

1. customers.



Language: SQL Rows: 10 Clear Command Find Tables Save Run

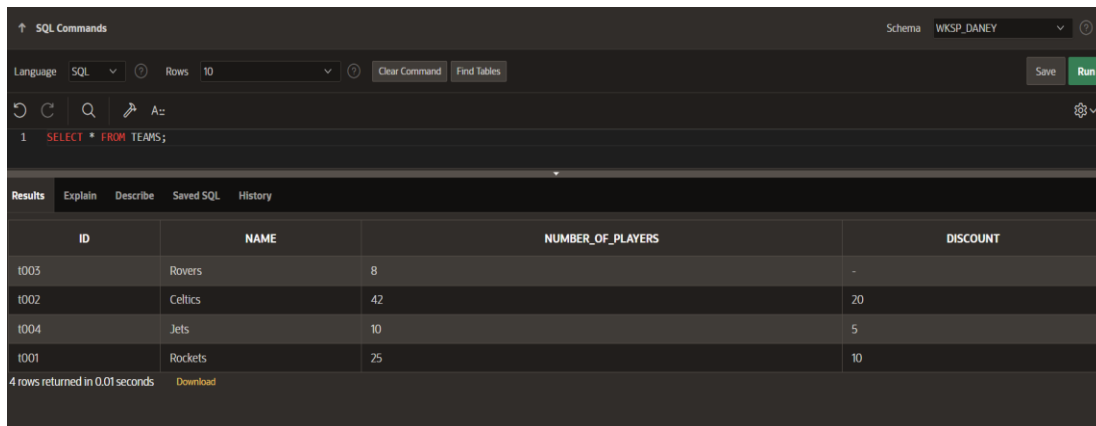
1 SELECT * FROM CUSTOMERS;

Results Explain Describe Saved SQL History

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	1001	-
c02001	brianrog@hooitech.com	Brian	Rogers	01654564898	50	-	-	lc4587
c00101	unknown@here.com	John	Doe	03216547808	9875	sr01	1002	-
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	1003	-

6 rows returned in 0.01 seconds Download

2. teams.



SQL Commands Schema: WKSP_DANEY

Language: SQL Rows: 10 Clear Command Find Tables Save Run

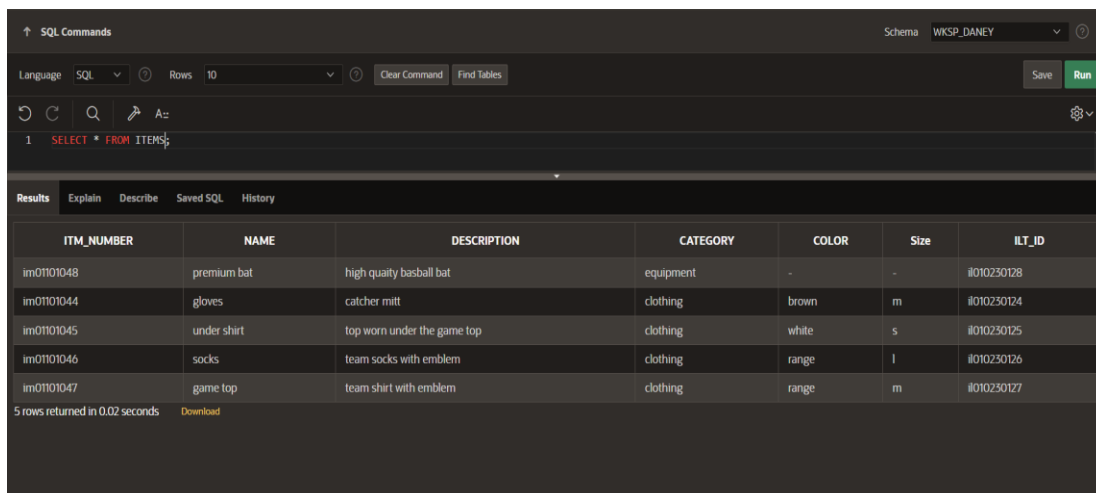
1 SELECT * FROM TEAMS;

Results Explain Describe Saved SQL History

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

4 rows returned in 0.01 seconds Download

3. items



SQL Commands Schema: WKSP_DANEY

Language: SQL Rows: 10 Clear Command Find Tables Save Run

1 SELECT * FROM ITEMS;

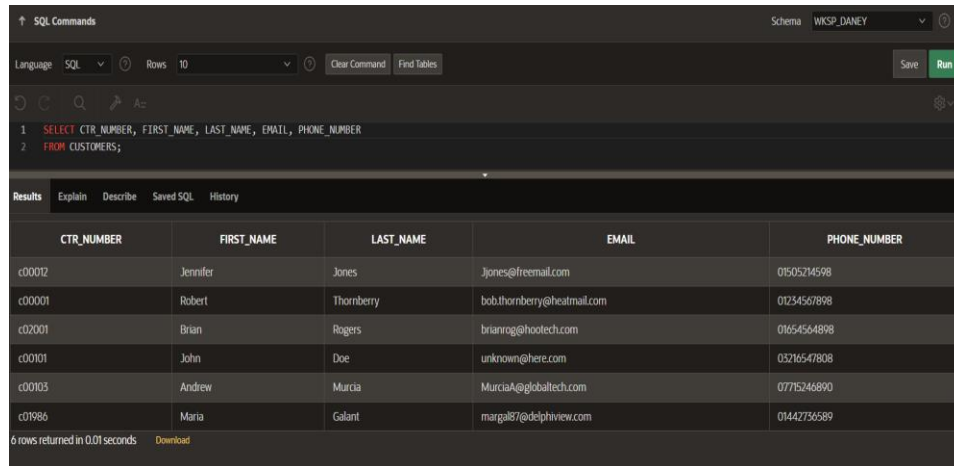
Results Explain Describe Saved SQL History

ITM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID
im0101048	premium bat	high quaiity baseball bat	equipment	-	-	#010230128
im0101044	gloves	catcher mitt	clothing	brown	m	#010230124
im0101045	under shirt	top worn under the game top	clothing	white	s	#010230125
im0101046	socks	team socks with emblem	clothing	range	l	#010230126
im0101047	game top	team shirt with emblem	clothing	range	m	#010230127

5 rows returned in 0.02 seconds Download

Selecting Specific Columns

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



The screenshot shows a SQL interface with the following query:

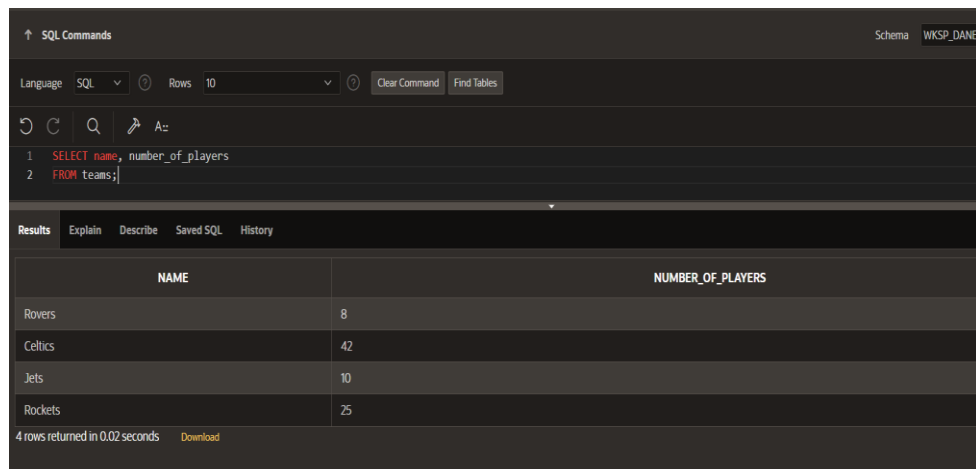
```
1 SELECT CTR_NUMBER, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER
2 FROM CUSTOMERS;
```

The results table displays the following data:

CTR_NUMBER	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
c00012	Jennifer	Jones	J.jones@freemail.com	01505214598
c00001	Robert	Thornberry	bob.thornberry@heatmail.com	01234567898
c02001	Brian	Rogers	brianrog@hootech.com	01654564898
c00101	John	Doe	unknown@here.com	03216547808
c00103	Andrew	Murcia	MurciaA@globaltech.com	07715246890
c01986	Maria	Gallant	margal87@delphiview.com	01442736589

6 rows returned in 0.01 seconds

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



The screenshot shows a SQL interface with the following query:

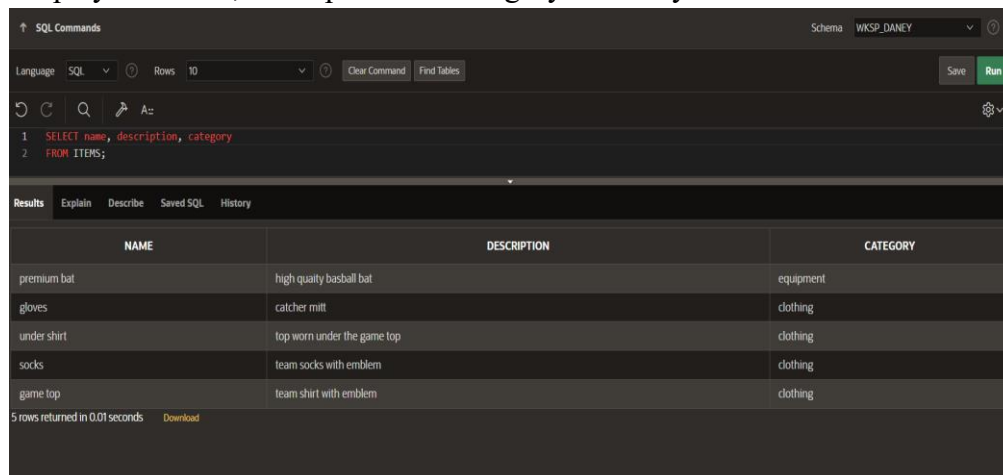
```
1 SELECT name, number_of_players
2 FROM teams;
```

The results table displays the following data:

NAME	NUMBER_OF_PLAYERS
Rovers	8
Celtics	42
Jets	10
Rockets	25

4 rows returned in 0.02 seconds

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



The screenshot shows a SQL interface with the following query:

```
1 SELECT name, description, category
2 FROM ITEMS;
```

The results table displays the following data:

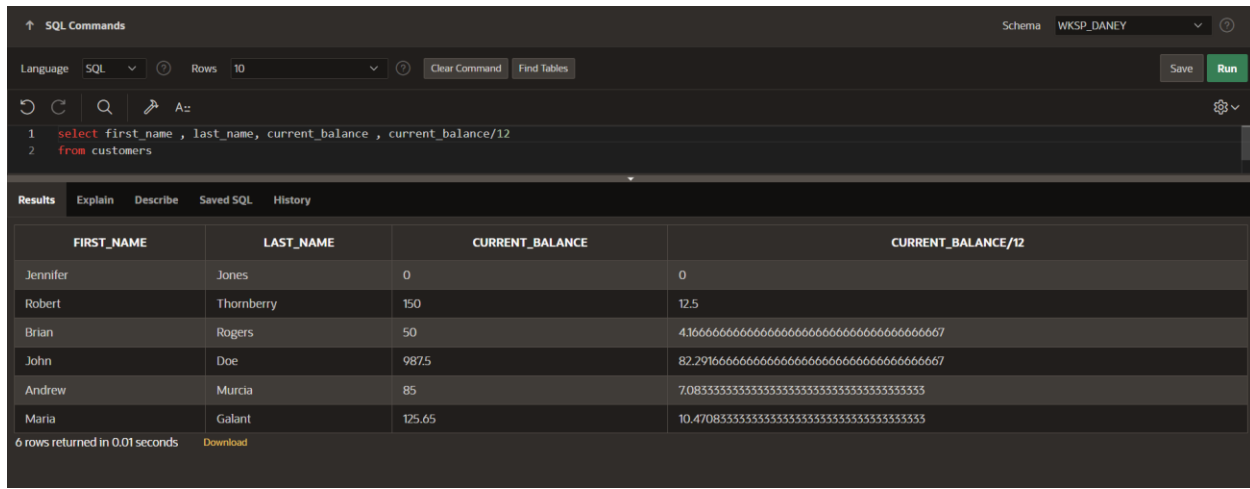
NAME	DESCRIPTION	CATEGORY
premium bat	high quality baseball bat	equipment
gloves	catcher mitt	clothing
under shirt	top worn under the game top	clothing
socks	team socks with emblem	clothing
game top	team shirt with emblem	clothing

5 rows returned in 0.01 seconds

PART 2:

Using Arithmetic Operators

1. Every customer has been told they can pay off their current balance over a 12 month period. Display the customer's first name, last name, current balance and monthly payment.



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

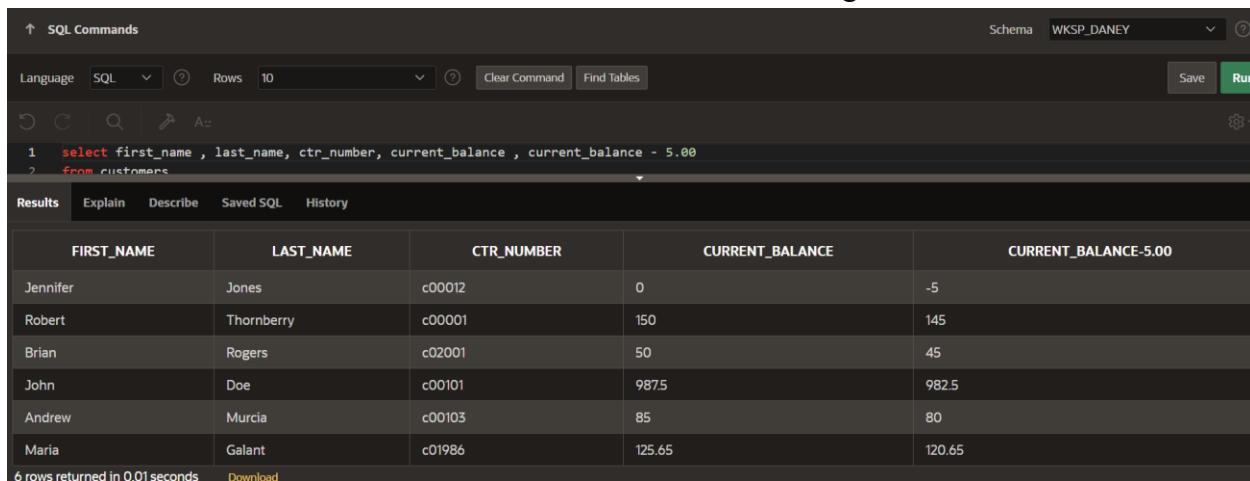
```
1 select first_name , last_name, current_balance , current_balance/12
2 from customers
```

The results table displays the following data:

FIRST_NAME	LAST_NAME	CURRENT_BALANCE	CURRENT_BALANCE/12
Jennifer	Jones	0	0
Robert	Thornberry	150	12.5
Brian	Rogers	50	4.166666666666666666666666666667
John	Doe	987.5	82.2916666666666666666666666667
Andrew	Murcia	85	7.083333333333333333333333333333
Maria	Galant	125.65	10.4708333333333333333333333333

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

2. Obl is considering giving a gift card to all its customers of 5.00 that can be used to reduce their current balance. Write a query that will show the customers first name, last name, customer number, current balance and the value of their balance minus the gift value



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

```
1 select first_name , last_name, ctr_number, current_balance , current_balance - 5.00
2 from customers
```

The results table displays the following data:

FIRST_NAME	LAST_NAME	CTR_NUMBER	CURRENT_BALANCE	CURRENT_BALANCE-5.00
Jennifer	Jones	c00012	0	-5
Robert	Thornberry	c00001	150	145
Brian	Rogers	c02001	50	45
John	Doe	c00101	987.5	982.5
Andrew	Murcia	c00103	85	80
Maria	Galant	c01986	125.65	120.65

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

3. **What would be the problem with implementing this scheme?**

These operations on the current data and the new columns that appear on the screen actually don't reflect on the database, so it's just for viewing.

Using Column Aliases

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).

[illegible]

Using Literal Character Strings

1. 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.

↑ SQL Commands

Schema WKSP_DANEY

Language SQL Rows 10 Clear Command Find Tables Save Run

↶ ↷ 🔍 ↵ A:

```
1 select 'The ' || name || ' team has ' || number_of_players || ' and receives a dicount of ' || discount || ' percent.'
2 from teams;
```

Results Explain Describe Saved SQL History

```
'THE'||NAME||'TEAMHAS'||NUMBER_OF_PLAYERS||'ANDRECEIVSADICOUNTOF'||DISCOUNT||'PERCENT.'
```

```
The Rovers team has 8 and receives a discount of percent.
The Celtics team has 42 and receives a discount of 20 percent.
The Jets team has 10 and receives a discount of 5 percent.
The Rockets team has 25 and receives a discount of 10 percent.
```

4 rows returned in 0.01 seconds Download

2. Why does the last team not show a discount?

Because The discount column for the Rovers team doesn't have an assigned value.

PART 3:

Using the WHERE Clause.

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

The screenshot shows the SQL Commands interface with the following query:

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

The results table is displayed below the query:

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

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 SQL Commands interface with the following query:

```
1 SELECT first_name "First Name", last_name "Last Name", ctr_number "Customer Number"
2 FROM customers
3 WHERE current_balance > 100
```

The results table is displayed below the query:

First Name	Last Name	Customer Number
Robert	Thornberry	c00001
John	Doe	c00101
Maria	Galant	c01986

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 the SQL Commands interface with the following query:

```
1 SELECT id "Order ID", odr_date "Order Date", odr_time "Order Time"
2 FROM orders
3 WHERE odr_date < DATE '2019-05-28'
```

The results table is displayed below the query:

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

5 rows returned in 0.02 seconds

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 the SQL Developer interface. The 'SQL Commands' tab is active, displaying the following query:

```
1 select id "Inventory ID", cost "Inventory Cost", units "Inventory Unit"
2 from inventory_list
3 where cost between 3.00 and 15.00
```

The 'Results' tab shows the output of the query:

Inventory ID	Inventory Cost	Inventory Unit
il010230126	5.24	87
il010230125	7.99	250

2 rows returned in 0.01 seconds

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. The 'SQL Commands' tab is active, displaying the following query:

```
1 select id "Inventory ID", cost "Inventory Cost", units "Inventory Unit"
2 from inventory_list
3 where units IN ( 50, 100, 150 , 200 )
```

The 'Results' tab shows the output of the query:

Inventory ID	Inventory Cost	Inventory Unit
il010230124	2.5	100

1 rows returned in 0.03 seconds

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. The 'SQL Commands' tab is active, displaying the following query:

```
1 select id "Inventory ID", cost "Inventory Cost", units "Inventory Unit"
2 from inventory_list
3 where units NOT IN ( 50, 100, 150 , 200 )
```

The 'Results' tab shows the output of the query:

Inventory ID	Inventory Cost	Inventory Unit
il010230126	5.24	87
il010230128	97.46	8
il010230125	7.99	250
il010230127	18.95	65

4 rows returned in 0.01 seconds

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:** The command window contains the following SQL query:

```
1 select item_number "Item Number", name "Item Name"
2 from Items
3 where name like 'g%'
4
```
- Schema:** WKSP_DANEY
- Language:** SQL
- Rows:** 10
- Buttons:** Clear Command, Find Tables, Save, Run
- Results:** The results tab is active, showing a table with two columns: Item Number and Item Name. The table contains two rows of data.

Item Number	Item Name
im01101044	gloves
im01101047	game top

2 rows returned in 0.03 seconds [Download](#)

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:** The command window contains the following SQL query:

```
1 select item_number "Item Number", name "Item Name"
2 from Items
3 where name like '%o%'
4
```
- Schema:** WKSP_DANEY
- Language:** SQL
- Rows:** 10
- Buttons:** Clear Command, Find Tables, Save, Run
- Results:** The results tab is active, showing a table with two columns: Item Number and Item Name. The table contains three rows of data.

Item Number	Item Name
im01101044	gloves
im01101046	socks
im01101047	game top

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

PART 4:

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.

The screenshot shows a SQL editor with the following query:

```
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
```

The results tab shows one row:

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

1 rows returned in 0.00 seconds

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.

The screenshot shows a SQL editor with the following query:

```
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 NOT NULL
```

The results tab shows three rows:

Team Information
The Celtics team has 42 players and does not receive a discount.
The Jets team has 10 players and does not receive a discount.
The Rockets team has 25 players and does not receive a discount.

3 rows returned in 0.03 seconds

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.

The screenshot shows a SQL editor with the following query:

```
1 select ctr_number "Customer Number", address_line_1 "Street Address", zip_code "Postal Code"
2 from customers_addresses
3 where address_line_1 LIKE 'Starford' AND city LIKE 'Liverpool'
```

The results tab shows "no data found".

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.

The screenshot shows a SQL interface with the following query:

```
1 select ctr_number "Customer Number", address_line_1 "Street Address", zip_code "Postal Code"
2 from customers_addresses
3 where address_line_1 LIKE 'Starford' OR city LIKE 'Liverpool'
```

The results tab shows 3 rows returned in 0.01 seconds. The data is as follows:

Customer Number	Street Address	Postal Code
c00001	83 Barrhill Drive	LP79HJK
c00001	63 Acacia Drive	LP83JHR
c00001	17 Gartsquare Road	LP89JHK

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.

The screenshot shows a SQL interface with the following query:

```
1 select ctr_number "Customer Number", address_line_1 "Street Address", zip_code "Postal Code"
2 from customers_addresses
3 where city NOT IN ('Liverpool')
```

The results tab shows 2 rows returned in 0.01 seconds. The data is as follows:

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

PART 5:

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

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

The screenshot shows a SQL interface with the following query:

```
1 select name "Team Name", number_of_players "Number of players"
2 from teams
3 ORDER BY name ASC
```

The results are displayed in a table with the following data:

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

4 rows returned in 0.01 seconds

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

The screenshot shows a SQL interface with the following query:

```
1 select name "Team Name", number_of_players "Number of players"
2 from teams
3 ORDER BY number_of_players DESC
```

The results are displayed in a table with the following data:

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

4 rows returned in 0.01 seconds

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.

The screenshot shows a SQL interface with the following query:

```
1 select name "Team Name", number_of_players "Players"
2 from teams
3 ORDER BY "Team Name" DESC
```

The results are displayed in a table with the following data:

Team Name	Players
Rovers	8
Rockets	25
Jets	10
Celtics	42

4 rows returned in 0.01 seconds

PART 6:

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

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

```
1 SELECT ROWNUM AS "N", first_name || ' ' || last_name "customer name"
2 FROM customers
3 where ROWNUM <= 3
4 ORDER BY ctr_number
```

The results are displayed in a table with two columns: "N" and "customer name".

N	customer name
2	Robert Thornberry
1	Jennifer Jones
3	Brian Rogers

3 rows returned in 0.01 seconds

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

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

```
1 SELECT first_name , last_name
2 FROM sales_representatives
3 WHERE commission_rate = :Commission_Rate
4 ORDER BY last_name
```

A dialog box titled "Enter Bind Variables - Google Chrome" is open, showing the substitution variable `:COMMISSION_RATE` and a text input field for its value. The dialog also includes a "Submit" button.