

Lab 3

Name: Lim Yu An

Matric Number: A22EC0186

Course Code: SECD2523

Course Name: Database

Section: 08

Part 1: Retrieving all columns from a table.

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

1. Customers.

```
1 SELECT * FROM customers
```

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	J.jones@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	-
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587

2. Teams.

```
1 SELECT * FROM teams
```

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

3. Items

```
1 SELECT * FROM items
```

ITM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID
im01101044	gloves	catcher mitt	clothing	brown	m	il010230124
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125
im01101046	socks	team socks with emblem	clothing	range	l	il010230126
im01101047	game top	team shirt with emblem	clothing	range	m	il010230127
im01101048	premium bat	high quality baseball bat	equipment	-	-	il010230128

Part 2: Selecting Specific Columns

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

```
1 SELECT ctr_number, first_name, last_name, email, phone_number
2 FROM customers
```

Results Explain Describe Saved SQL History

CTR_NUMBER	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
c00001	Robert	Thornberry	bob.thornberry@heatmail.com	01254567898
c00012	Jennifer	Jones	JJones@freemail.com	01505214598
c00101	John	Doe	unknown@here.com	03216547808
c00103	Andrew	Murcia	MurciaA@globaltech.com	0775246890
c01986	Maria	Gelant	margal87@delphiview.com	01442736589
c00001	Brian	Roevers	brianroe@bootech.com	01654564898

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

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

Results Explain Describe Saved SQL History

NAME	NUMBER_OF_PLAYERS
Rockets	25
Celtics	42
Rovers	8
Jets	10

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

```
1 SELECT name, description, category
2 FROM items
```

Results Explain Describe Saved SQL History

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 hat	high quality baseball hat	equipment

Part 1: 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.

[illegible]

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.

```
1 SELECT first_name, last_name, ctr_number, current_balance, current_balance - 5.00 AS discount_after_gift
2 FROM customers
```

Results Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME	CTR_NUMBER	CURRENT_BALANCE	DISCOUNT_AFTER_GIFT
Robert	Thornberry	c00001	150	145
Jennifer	Jones	c00012	0	-5
John	Doe	c00101	9875	987.5
Andrew	Murcia	c00103	85	80
Maria	Galant	c01986	125.65	120.65
Brian	Rogers	c02001	50	45

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

Jennifer with current_balance = 0 will have negative value. Her current balance after the deduction will become -5.

Part 2 : Using Column Aliases

1. You previously wrote a query that displayed 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]

Part 3: 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.

```
1 SELECT
2   The || name || 'team has ' || number_of_players || ' players and receives a discount of ' || discount || ' percent.' AS "Team Information"
3 FROM
4   Teams;
```

Results

Explain

Describe

Saved SQL

History

Team Information
The Rocketteam has 25 players and receives a discount of 10 percent.
The Celticteam has 42 players and receives a discount of 20 percent.
The Roverteam has 8 players and receives a discount of percent.
The Jetteam has 10 players and receives a discount of 5 percent.

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

The value of discount in Rovers team is null. A null value indicates that that particular column does not have any value or absence of value. Therefore, the discount value is missing.

Part 1: Using the WHERE Clause.

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

1 SELECT * FROM customers

2 WHERE ctr_number = 'c01986';

Results

Explain

Describe

Saved SQL

History

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
c01986	marc@127.0.1.1delphus.com	Maria	Galant	01447755589	725.65	c003	f003	

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.

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

Results

Explain

Describe

Saved SQL

History

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

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.

```
1 SELECT id, odr_date, odr_time FROM orders
2 WHERE odr_date < TO_DATE('28-May-2019', 'DD-MON-YYYY');
```

Results	Explain	Describe	Saved SQL	History
ID	ODR_DATE	ODR_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		

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.

```
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		

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.

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

Results	Explain	Describe	Saved SQL	History
Inventory ID		Cost	Number of units	
il010230124		2.5	100	

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.

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

Results	Explain	Describe	Saved SQL	History
Inventory ID		Cost	Number of units	
il010230125		7.99	250	
il010230126		5.24	87	
il010230127		18.95	65	
il010230128		97.46	8	

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.

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

Results	Explain	Describe	Saved SQL	History
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.

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

Results	Explain	Describe	Saved SQL	History
Item Number		Item Name		
im01101044		gloves		
im01101046		socks		
im01101047		game top		

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.

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

Results Explain Describe Saved SQL History

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

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.

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

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

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 stratford area of Liverpool. Use Customer Number, Street Address and Postal Code as the column aliases.

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

Results

Explain

Describe

Saved SQL

History

Customer Number	Street Address	Postal Code
c00001	17 Gartsquare Road	LP89JHK

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 stratford or Liverpool in general. Use Customer Number, Street Address and Postal Code as the column aliases.

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

Results	Explain	Describe	Saved SQL	History
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.

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

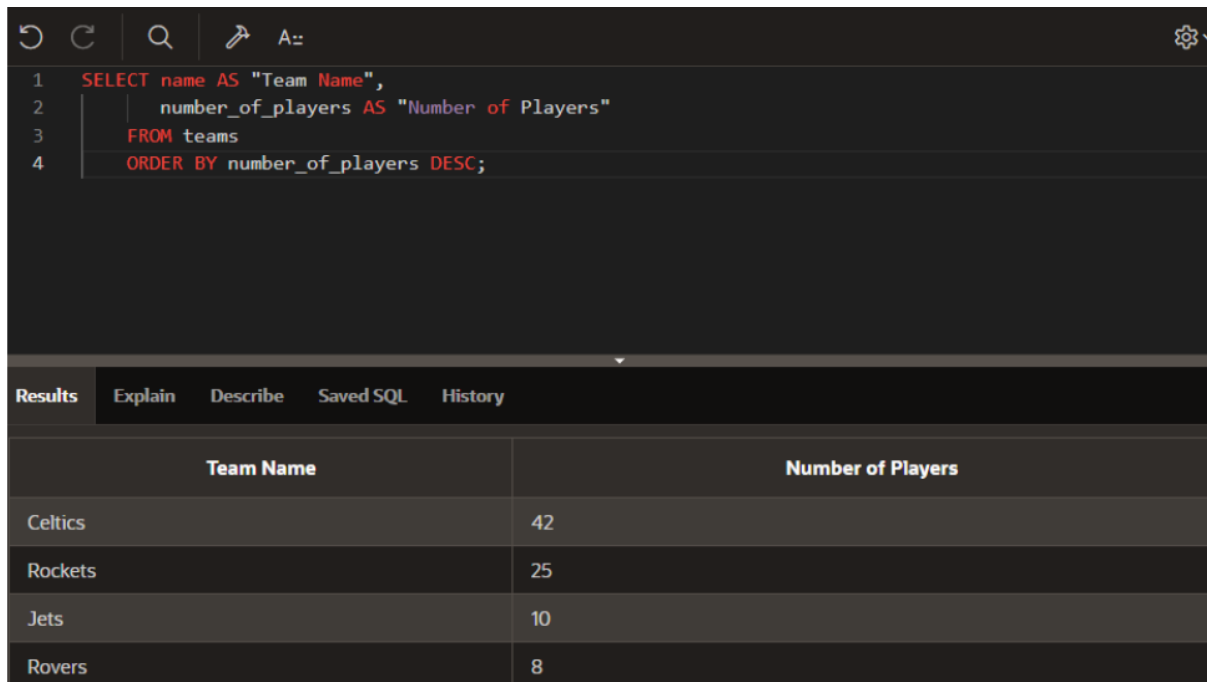
Results	Explain	Describe	Saved SQL	History
Customer Number		Street Address		Postal Code
c00101		54 Ropehill Crescent		ST45AGV
c01986		36 Watercress Lane		JP23YTH

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

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

Results	Explain	Describe	Saved SQL	History
Team Name		Number of Players		
Celtics		42		
Jets		10		
Rockets		25		
Rovers		8		

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



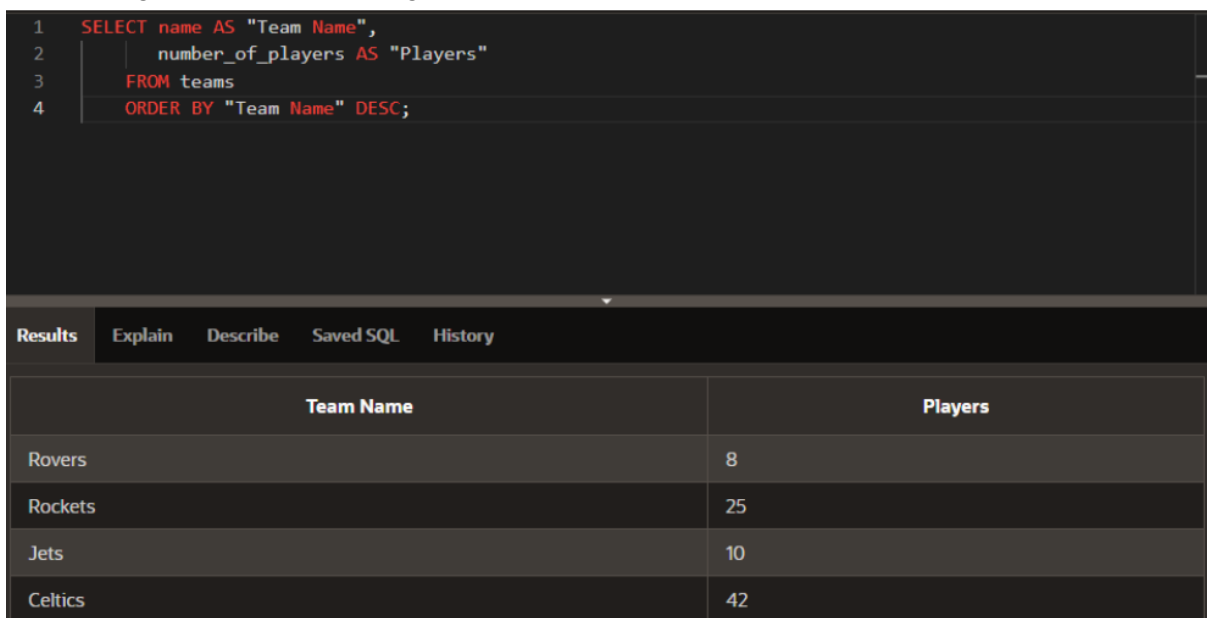
The screenshot shows a SQL IDE interface. The top toolbar includes icons for undo, redo, search, and a keyboard shortcut 'A:'. The SQL editor contains the following query:

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

Below the editor, the 'Results' tab is active, displaying the query output in a table format:

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

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 IDE interface. The top toolbar includes icons for undo, redo, search, and a keyboard shortcut 'A:'. The SQL editor contains the following query:

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

Below the editor, the 'Results' tab is active, displaying the query output in a table format:

Team Name	Players
Rovers	8
Rockets	25
Jets	10
Celtics	42

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.

```

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

```

Results Explain Describe Saved SQL History

Customer Name
Robert Thornberry
Jennifer Jones
John Doe

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.

Bind Variable	Value
:COMMISSION_RATE	5

```

1 SELECT
2     first_name AS "First Name",
3     last_name AS "Last Name",
4     commission_rate AS "Commission Rate"
5 FROM
6     sales_representatives
7 WHERE
8     commission_rate = :commission_rate
9 ORDER BY
10    last_name;

```

Results Explain Describe Saved SQL History

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