A black background with red letters

Description automatically generated

SECD2523 - DATABASE

SEMESTER 1/20232024

SECTION 08

**LAB 3 – DML PART 2**

**Lecturer: Dr. Noor Hidayah Binti Zakaria**

|  |  |
| --- | --- |
| **NAME** | **MATRIC NO.** |
| JESSIE CHANG | A22EC0054 |

**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

A screenshot of a computer

Description automatically generated

2. teams.

SELECT \* FROM teams

A screenshot of a computer

Description automatically generated

3. items

SELECT \* FROM items

A screenshot of a computer

Description automatically generated

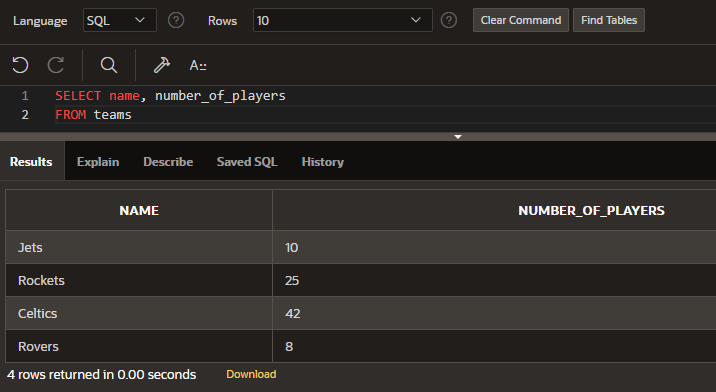
**Part 2: Selecting Specific Columns**

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

A screenshot of a computer

Description automatically generated

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



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

A screenshot of a computer

Description automatically generated

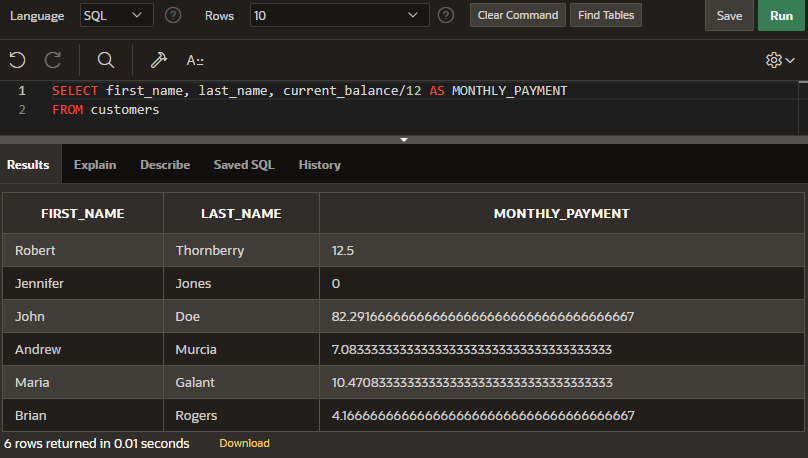
**Section 6 Lesson 6 Exercise 2: Retrieving Data Using SELECT**

**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: 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.



1. 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.

A screenshot of a computer

Description automatically generated

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

**The customer whose first\_name is Jennifer, have a current balance of 0. Her current value after minus the gift value turn into negative value, which is -5. Hence, the current\_balance could not be less than 5 as it is violating the rules.**

**Part 2 : 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).

**SELECT first\_name AS "First Name", last\_name AS "Last Name", current\_balance AS "Balance", current\_balance/12 AS "Monthly Repayments"**

**FROM customers;**

A screenshot of a computer

Description automatically generated

**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.

**SELECT 'The ' || name || ' team has ' || number\_of\_players || ' players and receives a discount of ' || discount || ' percent.' AS "Team Information"**

**FROM teams;**

A screenshot of a computer

Description automatically generated

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

**The last team which is Rovers team does not show a discount because the value insert into the discount column is null, which defines having no value. Thus, no value is shown.**

**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';**

A screenshot of a computer

Description automatically generated

1. 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;**

A screenshot of a computer

Description automatically generated

1. 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", odr\_time AS "Time"**

**FROM orders**

**WHERE odr\_date < '05/28/2019';**

A screenshot of a computer

Description automatically generated

**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;**

A screenshot of a computer

Description automatically generated

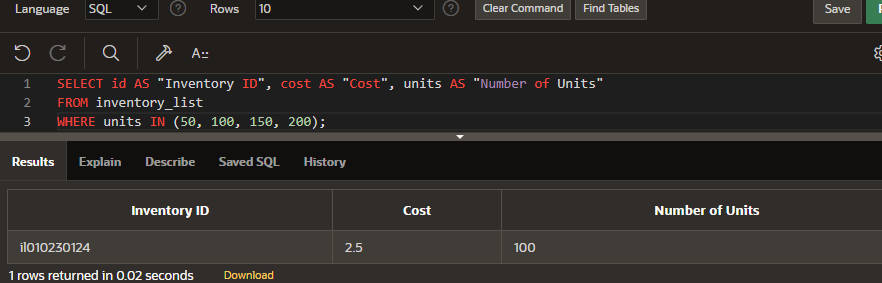
**Part 3: Membership Conditions: IN Operator**

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);**



**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);**

A screenshot of a computer

Description automatically generated

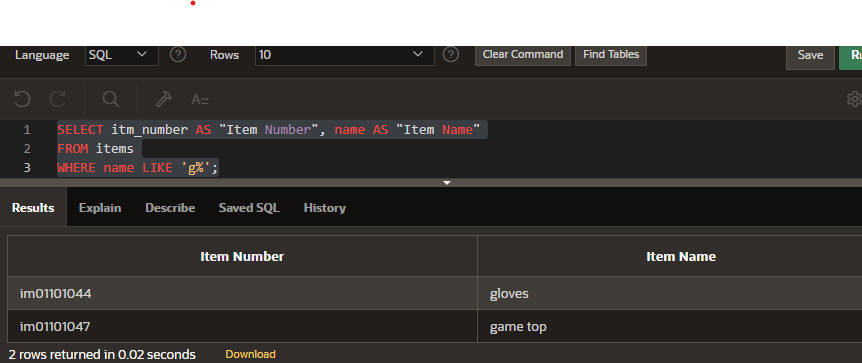
**Part 5: Pattern Matching: LIKE Operator**

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%';**



**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%';**

A screenshot of a computer

Description automatically generated

**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;**

A screenshot of a computer

Description automatically generated

1. 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 ' || discount || ' percent.' AS "Team Information"**

**FROM teams**

**WHERE discount IS NOT NULL;**

A screenshot of a computer

Description automatically generated

**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 address\_line\_2 = 'Starford'**

**AND city = 'Liverpool';**

A screenshot of a computer

Description automatically generated

**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 address\_line\_2 = 'Starford'**

**OR city = 'Liverpool';**

A screenshot of a computer

Description automatically generated

**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');**

A screenshot of a computer

Description automatically generated

**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;**

A screenshot of a computer

Description automatically generated

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

**SELECT name AS "Team Name", number\_of\_players AS "Number of Players"**

**FROM teams**

**ORDER BY number\_of\_players DESC;**

A screenshot of a computer

Description automatically generated

1. 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;**

A screenshot of a computer

Description automatically generated

**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**

**)**

**WHERE ROWNUM <= 3;**

A screenshot of a computer

Description automatically generated

**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 "Commisiion Rate"**

**FROM sales\_representatives**

**WHERE commission\_rate = :commission\_rate**

**ORDER BY last\_name;**

**A screenshot of a computer

Description automatically generated**

