

PART 1

1. SELECT * FROM customers
2. SELECT * FROM teams
3. SELECT * FROM items

PART 2

1. SELECT first_name, last_name, email, phone_number
FROM customers;

2. SELECT name, number_of_players
FROM teams;

3. SELECT name, description, category
FROM items

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. `SELECT first_name, last_name, current_balance, current_balance / 12
FROM customers;`

2. Obi 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.

3. `SELECT first_name, last_name, phone_number, current_balance, current_balance - 5
FROM customers;`

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

- invalid values
- could lead to errors.

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).

1. `SELECT first_name, last_name, current_balance, current_balance / 12 AS monthly_repayment
FROM customers;`

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 'The' || name || 'team has a' || number_of_players || 'and receives a discount of ' || discount || 'percent.'
AS "Team Information"
FROM teams;`

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

because it has NULL values

Part 1: Using the WHERE Clause.

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

```
SELECT ctr-number, email, first-name, last-name, phone-number, current-balance, stc-id, tem-id, loyalty-card-number  
FROM customers  
WHERE ctr-number = 'c01986';
```

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, last-name, current-balance AS "balance > 100"  
FROM customers  
WHERE current-balance > 100;
```

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, odr-date, odr-time  
FROM orders  
WHERE odr-date < ' 28 - MAY - 2019 ';
```

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, cost, units  
FROM inventory-list  
WHERE units BETWEEN 3 AND 15;
```

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, cost, 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, cost, units  
FROM inventory-list  
WHERE units NOT IN ( 50, 100, 150, 200 );
```

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, 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, name  
FROM items  
WHERE name LIKE "- o %";
```

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 'The' || name || 'team has a' || number_of_players || 'and receives a discount of' || discount || 'percent.'  
AS "Team Information"  
  
FROM teams  
  
WHERE discount IS NULL;
```

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 'The' || name || 'team has a' || number_of_players || 'and receives a discount of' || discount || 'percent.'  
AS "Team Information"  
  
FROM teams  
  
WHERE discount = 10;
```

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';
```

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';
```

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 NOT IN city = 'Liverpool';
```

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 team"  
FROM teams  
ORDER BY name;
```

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 "First Name", last_name "Last Name"  
FROM sales_representative WHERE commission_rate = commission;
```

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 team"  
FROM teams  
ORDER BY number_of_players DESC;
```

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 "Players"  
FROM teams  
ORDER BY name DESC;
```

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;
```