AERON GOH MING LUN No. Date

SQL 3

Part 1 : Retrieving all columns from a table

1. Customers

```
SELECT * FROM customers;
```

2. Teams

```
SELECT * FROM teams;
```

3. Items

```
SELECT * FROM items;
```

Part 2 : Selecting Specific Columns

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

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

3.
```
SELECT name, description, category
FROM items;
```

SQL 3 Part 2

Part 1 : Using Arithmetic Operators

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

No. Date

2 SELECT

first_name,

last_name,

ctr_number,

current_balance,

current_balance - 5.00 AS adjusted_balance

FROM customers;

3. When current balance is zero, the 'current_balance -5' becomes -5 which is illogical in real world.

Part 2 : Using Column Aliases

1. SELECT

first_name AS "First Name",

last_name AS "Last Name",

current_balance AS "Balance",

current_balance / 12 AS "Monthly Repayments"

FROM customers;

Part 3: Using Literal Character Strings

1. SELECT

'The' || name || 'team has' || number_of_players || 'players and receives a discount of' || discount || 'percent.' AS "Team Information"

2. It appears that the last team in the data has a null value in the discount column. When concatenating the string in the query, the null value is treated as unknown, and it does not display a specific discount in the result.

No. Date

SQL Part 3

1. Part 1 : Using the WHERE Clause

```
SELECT *
FROM customers
WHERE ctr_number = 'c01986';
```

2.
```
SELECT
    first_name AS "First Name",
    last_name AS "Last Name",
    ctr_number AS "Customer Number"
FROM customers
WHERE current_balance >100;
```

3.
```
SELECT
    id AS "Order ID",
    odr_date AS "Order Date",
    odr_time AS "Order Time"
From orders
WHERE odr_date < TO_DATE ('28-May-2019', 'DD-MM-YYYY');
```

Part 2: Range conditions : BETWEEN operator

```
SELECT
    il.id AS "Inventory ID",
    il.cost AS "Trade Cost",
    il.units AS "Number of Units"
FROM inventory_list il
WHERE il.cost BETWEEN 3.00 AND 15.00;
```

No. Date

PART 3: Membership Conditions: IN Operator

```
SELECT
    id AS "Inventory ID",
    cost AS "Trade Cost",
    units AS "Number of Units"
FROM inventory_list
WHERE units IN (50, 100, 150, 200);
```

PART 4: Membership Conditions: NOT IN Operator

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

PART 5: Pattern Matching: LIKE Operator

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

No. Date

SQL3

Part 4

Part 1 : Using the NULL Conditions

1.
```
SELECT
    'The' || name || 'team has' || number_of_players || 'players and does not receive a discount.'
    As "Team Information"

FROM teams
WHERE discount IS NULL;
```

2.
```
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;
```

Part 2 : Logical Operators : AND

1.
```
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';
```

Part 3 : Logical Operators : OR

1.
```
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';
```

No. Date

Part 4: Logical Operators : NOT Equal To

1.
```
SELECT
    ctr_number AS "Customer Number",
    address_line_1 AS "Street Address",
    zip_code AS "Postal Code"
FROM customers_addresses
WHERE city NOT LIKE 'Liverpool';
```

SQL 3 Part 5

Use the ORDER BY Clause to Sort SQL Results

1.
```
SELECT
    name AS "Team Name",
    number_of_players AS "Number of Players"
FROM teams
ORDER BY "Team Name" ASC;
```

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

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

No. Date

SQL3 Part 6

Part 1: TOP-N-ANALYSIS (SGL8 Objective 3)

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

Part 2: Using a substitution Variable (SGL8 Objective 4)

```
SELECT
   first_name AS "First Name",
   last_name AS "Last Name"
FROM sales_representatives
WHERE comission_rate = :input_comission_rate
ORDER BY last_name;
```

For the input = 5