SQL Level 3

Part 1

noble desktop

Conditionals

Adding If-Else Logic to a Query

IF-Else Logic

If-Else logic in SELECT statements is typically done using CASE.

SQL Server 2012 also introduced IIF( ) which is an inline IF.
IIF is a shorthand way for writing a basic CASE expression that accepts 3 arguments:

1. the expression to test
2. the value to use if it's true
3. the value to use if it's false

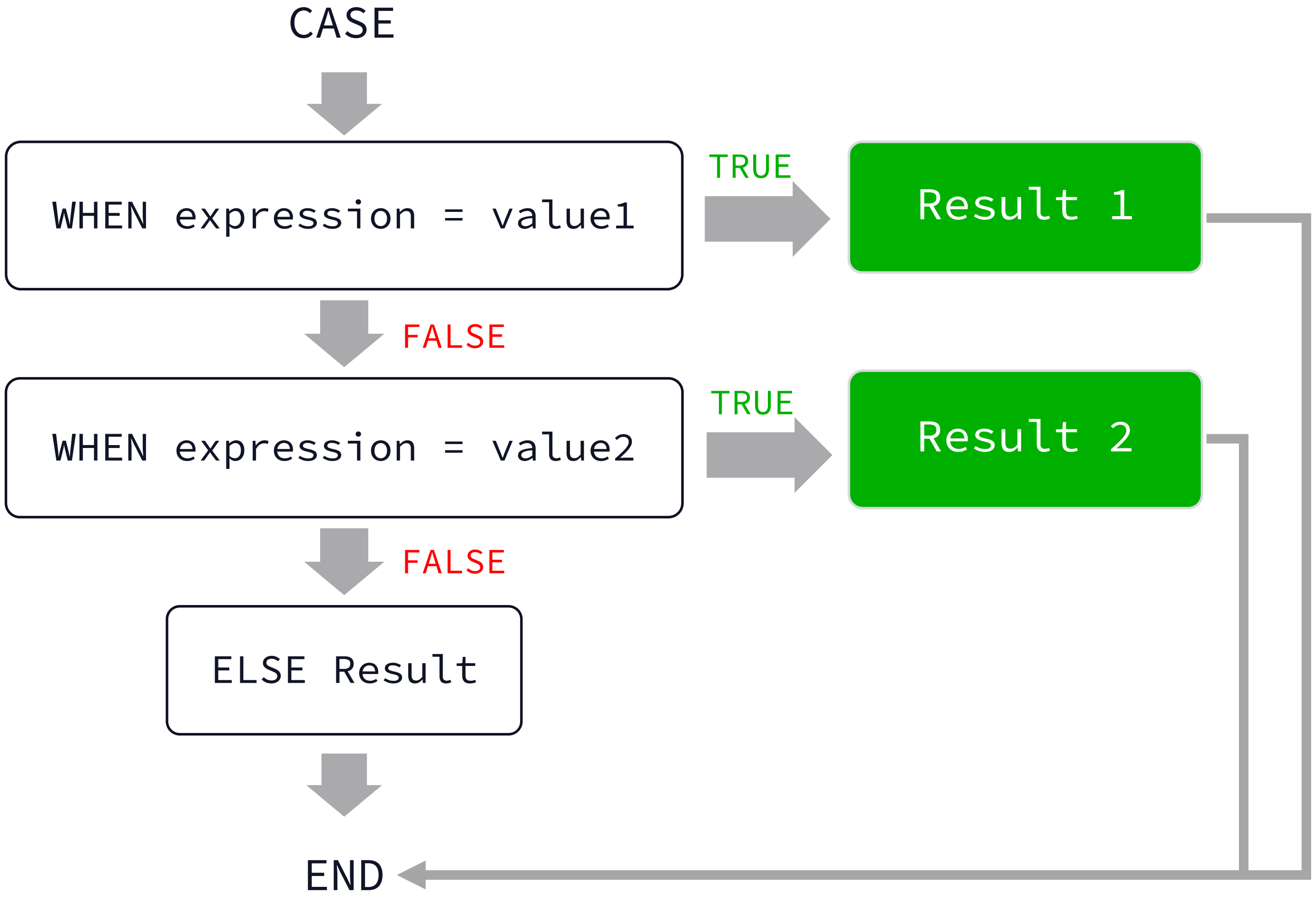
CASE

CASE allows you to evaluate a list of conditions and returns one of the possible results.

It has 2 formats:

- Simple CASE: performs a simple equality check
- Searched CASE: evaluates the conditions independently under each of the WHEN options.

Simple CASE Example



```
CASE bike_type
   WHEN 'R' THEN 'Road'
   WHEN 'M' THEN 'Mountain'
   ELSE 'Unclassified'
END
```

Simple Case:
performs a simple equality check

Searched CASE Example

```sql
SELECT title,
    CASE
        WHEN (price < 20) THEN 'cheap'
        WHEN (price < 50) THEN 'moderate'
        ELSE 'expensive'
    END AS pricing
FROM products
ORDER BY pricing;
```

Searched Case:
evaluates the conditions independently for each WHEN

Exercise

Open the file “1.0 CASE.sql” (in SQL Level 3 folder)

String Functions

Working with Text

String Functions

- A **string** is a sequence of characters.
- **String functions** take one or more characters (or numbers), perform an operation (such as convert to lowercase), and return a result.

String Functions

Refer to these website links for a list of string functions:

- PostgreSQL String Functions

  PostgreSQL String Functions and Operators Documentation

- SQL Server String Functions

  Microsoft’s String Functions Documentation

Convert to Lowercase

```
SELECT LOWER(email)
FROM users;
```

Convert to Uppercase

```
SELECT UPPER(user_state)
FROM users;
```

Get Part of a String

```
SELECT SUBSTRING(ship_zipcode,1,5)
FROM orders;
```

Zipcodes are sometimes written 12345-1234

Starting with the the 1st character, this returns

5 characters (the part before the -)

PostgreSQL: Splitting a String

```
SELECT SPLIT_PART(ship_zipcode,'-',1)
FROM orders;


SELECT SPLIT_PART(ship_zipcode,'-',2)
FROM orders;
```

This splits a zipcode at the - character, then returns the first or second part (before/after the - character).

PostgreSQL: Splitting a String

```
SELECT SPLIT_PART(email,'@',2)
FROM users;
```

This splits an email at the @ character, then returns the second part (the part after the @ character).

Exercise

Open the file “1.1 String Functions.sql” (in SQL Level 3 folder)

Subqueries

Writing a Query Inside of a Query

What is a Subquery?

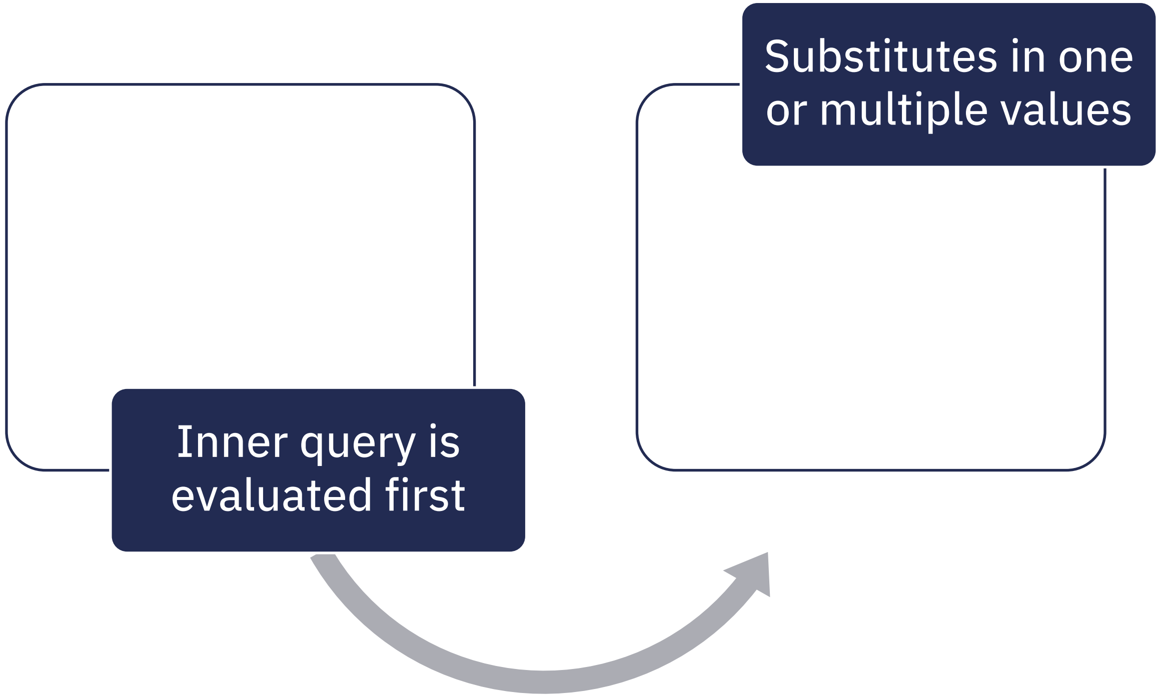
- Subqueries are commonly used to filter the results of one table, based on the results of a query on another table.
- The **subquery** is called an ***inner query*** or ***inner select***, while the **statement containing the subquery** is called an ***outer query*** or ***outer select***.

Single-Value Subquery

```
SELECT * FROM line_items
WHERE price = (
    SELECT MAX(price) FROM products
);
```

Subquery Syntax

The inner query executes first (before its parent query), so the results of an inner query can be passed to the outer query.



Substitutes in one or multiple values

Inner query is evaluated first

Subqueries

- Joins let you include columns from multiple tables (building a combined data set that includes columns from those tables).
- Subqueries can refer to another table without having to join it.
- You can use both joins and subqueries in the same SQL query.
- Subqueries can be nested inside other subqueries.

Exercise

Open the file “1.2 Subqueries.sql” (in SQL Level 3 folder)