



# Conditional Expressions and Procedures Lecture

---

- Section Overview
  - CASE
  - COALESCE
  - NULLIF
  - CAST
  - Views
  - Import and Export Functionality

- คำหลักและฟังก์ชันเหล่านี้จะช่วยให้เราสามารถเพิ่มตรรกะให้กับคำสั่งและเวิร์กโฟลว์ของเราใน SQL
- 

## CASE

- เราสามารถใช้คำสั่ง CASE เพื่อรันโค้ด SQL เมื่อตรงตามเงื่อนไขบางประการเท่านั้น
- ซึ่งคล้ายกับคำสั่ง IF/ELSE ในภาษาการเขียนโปรแกรมอื่นๆ
- มีสองวิธีหลักในการใช้คำสั่ง CASE ไม่ว่าจะเป็น general CASE หรือ CASE expression
- ทั้งสองวิธีสามารถนำไปสู่ผลลัพธ์เดียวกัน

- ชั้นแรกให้แสดงไวยากรณ์สำหรับ “general” CASE

- General Syntax

- CASE

WHEN condition1 THEN result1

WHEN condition2 THEN result2

ELSE some\_other\_result

END

- Simple Example

- SELECT a,  
CASE WHEN a = 1 THEN 'one'  
WHEN a = 2 THEN 'two'  
ELSE 'other'

| a |
|---|
| 1 |
| 2 |

- Simple Example
  - `SELECT a,`  
`CASE WHEN a = 1 THEN 'one'`  
`WHEN a = 2 THEN 'two'`  
`ELSE 'other' AS label`  
`END`  
`FROM test;`

| a | label |
|---|-------|
| 1 | one   |
| 2 | two   |



- syntax CASE จะประเมินนิพจน์ก่อน จากนั้นจึงเปรียบเทียบผลลัพธ์กับแต่ละค่าในคำสั่งย่อย WHEN ตามลำดับ

- *CASE Expression Syntax*
  - CASE expression

`WHEN value1 THEN result1`

`WHEN value2 THEN result2`

`ELSE some_other_result`

`END`

```
#CASE
SELECT customer_id,
CASE
  WHEN (customer_id <= 100) THEN 'Premium'
```

```

    WHEN (customer_id BETWEEN 100 and 200) THEN 'Plus'
    ELSE 'Normal'
END AS customer_class
FROM customer

```

```

#CASE expression
SELECT customer_id,
CASE customer_id
    WHEN 2 THEN 'Winner'
    WHEN 5 THEN 'Second Place'
    ELSE 'Normal'
END AS raffle_results
FROM customer
-- แทนที่จะตรวจสอบเงื่อนไข แต่เป็นการใช้ value แทน
-- CASE customer_id เรียกคอลัมน์ที่สนใจ
-- ไม่ยึดหยุ่นเท่าแต่ต่าง

```

```

SELECT rental_rate,
CASE rental_rate
    WHEN 0.99 THEN 1
    ELSE 0
END
FROM film
-- การตรวจสอบหา 0.99

```

```

SELECT
SUM(CASE rental_rate
    WHEN 0.99 THEN 1
    ELSE 0
END) AS number_of_bargains
FROM film
#การรวมผลของเงื่อนไขนี้สามารถนับว่ามี 0.99 ที่ตัวได้

```

```

SELECT
SUM(CASE rental_rate
    WHEN 0.99 THEN 1
    ELSE 0
END) AS bargains,
SUM(CASE rental_rate
    WHEN 2.99 THEN 1
    ELSE 0
END) AS regular,
SUM(CASE rental_rate
    WHEN 4.99 THEN 1
    ELSE 0
END) AS premium
FROM film
-- เปรียบเทียบราคาในแต่ละ rate เหมือนการกรองข้อมูล

```

# CASE

## Challenge Task

### Challenge

- เราต้องการทราบและเปรียบเทียบจำนวนภาพยนตร์ต่างๆ ที่เรามีการจัดเรตภาพยนตร์
- ใช้ CASE และฐานข้อมูล dvdrental เพื่อสร้างตารางนี้ใหม่:

```
SELECT
SUM(CASE rating
  WHEN 'R' THEN 1
  ELSE 0
END) AS r,
SUM(CASE rating
  WHEN 'PG-13' THEN 1
  ELSE 0
END) AS pg_13,
SUM(CASE rating
  WHEN 'PG' THEN 1
  ELSE 0
END) AS pg,
SUM(CASE rating
  WHEN 'NC-17' THEN 1
  ELSE 0
END) AS nc_17,
SUM(CASE rating
  WHEN 'G' THEN 1
  ELSE 0
END) AS g
FROM film
```

| Data Output |             | Messages        |              | Notifications   |             |
|-------------|-------------|-----------------|--------------|-----------------|-------------|
|             | r<br>bigint | pg_13<br>bigint | pg<br>bigint | nc_17<br>bigint | g<br>bigint |
| 1           | 195         | 223             | 194          | 210             | 178         |

---

# COALESCE

- ฟังก์ชัน COALESCE ยอมรับอาร์กิวเมนต์ได้ไม่จำกัดจำนวน จะส่งกลับอาร์กิวเมนต์แรกที่ไม่ใช่ null หากอาร์กิวเมนต์ทั้งหมดเป็นค่า null ฟังก์ชัน COALESCE จะคืนค่าเป็นค่า null
- `COALESCE (arg_1, arg_2, ..., arg_n)`

- Example

- `SELECT COALESCE (1, 2)`
  - 1
- `SELECT COALESCE(NULL, 2, 3)`
  - 2

- ฟังก์ชัน COALESCE จะมีประโยชน์เมื่อ querying ตารางที่มีค่า Null และแทนที่ด้วยค่าอื่น

- Table of Products
  - What is the final price?

| Item | Price | Discount |
|------|-------|----------|
| A    | 100   | 20       |
| B    | 300   | null     |
| C    | 200   | 10       |

- `SELECT item,(price - discount) AS final`  
`FROM table`
- Doesn't work for item B, should be 300.

| Item | final |
|------|-------|
| A    | 80    |
| B    | null  |
| C    | 190   |

```
SELECT item,(price - COALESCE(discount,0))  
AS final FROM table
```

| Item | final |
|------|-------|
| A    | 80    |
| B    | 300   |
| C    | 190   |

PIERIAN  DATA



คำนึงถึงฟังก์ชัน COALESCE ไว้ในกรณีที่คุณพบตารางที่มีค่า Null ที่คุณต้องการดำเนินการ!

# CAST


- CAST จะแปลงข้อมูลประเภทหนึ่งเป็นอีกประเภทหนึ่ง
- โปรดทราบว่าไม่ใช่ทุก instance of a data type ที่สามารถ CAST ไปยังประเภทข้อมูลอื่นได้ การแปลงข้อมูลจะต้องมีเหตุผล เช่น '5' เป็นจำนวนเต็มจะใช้ได้ 'five' เป็นจำนวนเต็มจะใช้ไม่ได้



- Syntax for CAST function
  - SELECT CAST('5' AS INTEGER)
- PostgreSQL CAST operator
  - SELECT '5'::INTEGER

- โปรดทราบว่าคุณสามารถใช้สิ่งนี้ใน SELECT query ที่มีชื่อคอลัมน์แทน single instance
- `SELECT CAST(date AS TIMESTAMP) FROM table`
- อ่านเพิ่มเติม เพื่อเพิ่มลูกเล่นกับ string ได้!

String Functions and Operators

 <https://www.postgresql.org/docs/9.1/functions-string.html>



```
#INTEGER
SELECT CAST('5' AS INTEGER)
SELECT '5'::INTEGER

#VARCHAR
SELECT CAST(inventory_id AS VARCHAR) FROM rental
SELECT CHAR_LENGTH(CAST(inventory_id AS VARCHAR)) FROM rental
```

# NULLIF

- ฟังก์ชัน NULLIF รับ 2 อินพุตและส่งคืนค่า NULL หากทั้งคู่มีค่าเท่ากัน otherwise จะส่งคืนค่า first argument passed
- `NULLIF(arg1,arg2)`

```
#Example
NULLIF(10,10)
--> Returns NULL

NULLIF(10,12)
--> Returns 10
```

- สิ่งนี้มีประโยชน์มากในกรณีที่ค่า NULL อาจทำให้เกิดข้อผิดพลาดหรือผลลัพธ์ที่ไม่ต้องการ
- Let's jump to pgAdmin, quickly create this table and walk through solving the RATIO and why we may need NULLIF

| Name   | Department |
|--------|------------|
| Lauren | A          |
| Vinton | A          |
| Claire | B          |

```
CREATE TABLE depts(
  first_name VARCHAR(50),
  department VARCHAR(50)
)

INSERT INTO depts(
  first_name,
  department
```

```

)
VALUES
('Vinton', 'A'),
('Lauren', 'A'),
('Claire', 'B');

SELECT (
SUM(CASE WHEN department = 'A' THEN 1 ELSE 0 END)/
SUM(CASE WHEN department = 'B' THEN 1 ELSE 0 END)
) AS department_ratio
FROM depts

# ถ้าไม่มีแผนก B
DELETE FROM depts
WHERE department = 'B'

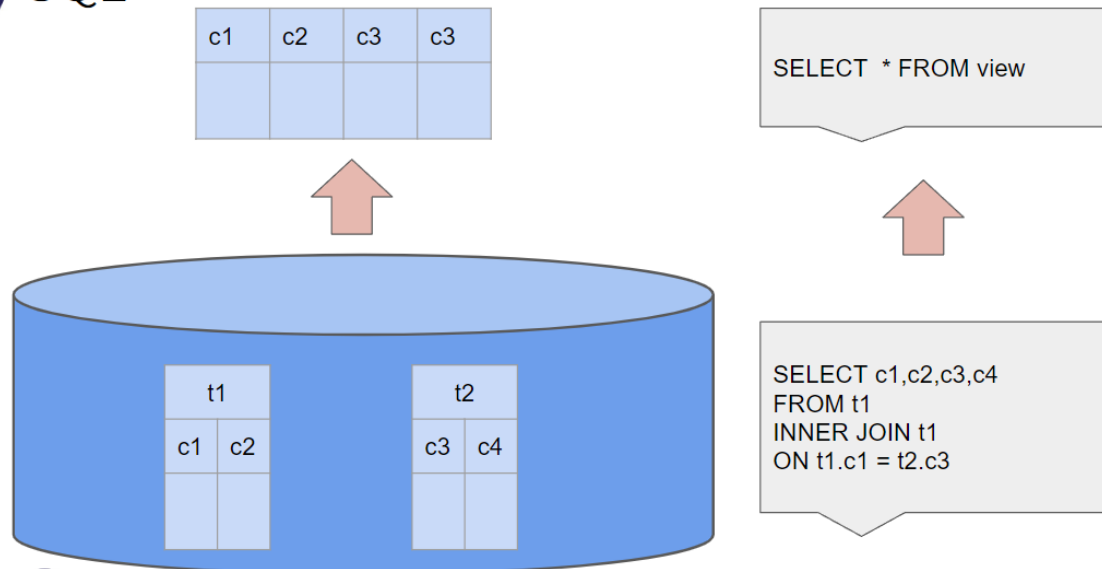
SELECT (
SUM(CASE WHEN department = 'A' THEN 1 ELSE 0 END)/
SUM(CASE WHEN department = 'B' THEN 1 ELSE 0 END)
) AS department_ratio
FROM depts
--> ERROR: division by zero
-- แผนก B เมื่อไม่มี B มันจะเป็นการหารด้วย 0 ซึ่งจะทำให้ ERROR
-- แก้โดยใช้ NULLIF
SELECT (
SUM(CASE WHEN department = 'A' THEN 1 ELSE 0 END)/
NULLIF(SUM(CASE WHEN department = 'B' THEN 1 ELSE 0 END),0)
) AS department_ratio
FROM depts

```

| Data Output   | Messages                   | Notifications |  |                            |   |   |  |        |
|---|----------------------------|---------------|--|----------------------------|---|---|--|--------|
| <div> <div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🔄</div> <div>⬇️</div> <div>📈</div> </div> <table> <tr> <td></td><td>department_ratio<br/>bigint</td><td>🔒</td></tr> <tr> <td>1</td><td></td><td>[null]</td></tr> </table> </div> |                            |               |  | department_ratio<br>bigint | 🔒 | 1 |  | [null] |
|   | department_ratio<br>bigint | 🔒             |  |                            |   |   |  |        |
| 1   |                            | [null]        |  |                            |   |   |  |        |

# VIEWS

- มักจะมีชุดของตารางและเงื่อนไขเฉพาะที่คุณพบว่าตัวเองใช้ค่อนข้างบ่อยสำหรับโครงการ
- แทนที่จะต้องทำการค้นหาเดิมซ้ำแล้วซ้ำอีกเป็นจุดเริ่มต้น คุณสามารถสร้าง VIEW เพื่อ query นี้ได้อย่างรวดเร็วด้วย simple call



## PIERIAN DATA

- view เป็น database object ที่เก็บ stored query
- view สามารถเข้าถึงได้เป็น virtual table ใน PostgreSQL
- โปรดสังเกตว่า view ไม่ได้จัดเก็บ data physically แต่จะเก็บ query เท่านั้น
- คุณยังสามารถ update and alter กับ view ที่มีอยู่เดิมได้

```
CREATE VIEW customer_info AS
SELECT first_name, last_name, address FROM customer
INNER JOIN address
ON customer.address_id = address.address_id
--> เหมือนว่าเราแทนคำสั่งที่แสดงทั้งหมดด้วย customer_info

# การเปลี่ยนแปลง หรือ แทนที่ VIEW เดิม
CREATE OR REPLACE VIEW customer_info AS
SELECT first_name, last_name, address, district FROM customer
INNER JOIN address
ON customer.address_id = address.address_id

# การลบ VIEW
DROP VIEW IF EXISTS customer_info --> IF EXISTS ตรวจสอบก่อนลบทิ้ง กันการ ERROR
DROP VIEW c_info

# ต้องการเปลี่ยนชื่อ VIEW
ALTER VIEW customer_info RENAME to c_info
```

# Importing and Exporting Data

- ในการบรรยายนี้ เราจะสำรวจฟังก์ชัน Import/Export ของ PgAdmin ซึ่งทำให้เราสามารถนำเข้าข้อมูลจากไฟล์ .csv ไปยังตารางที่มีอยู่แล้วได้
- มีหมายเหตุสำคัญบางประการที่ควรทราบเมื่อ Import/Export

## Important Note!

- ไม่ใช่ไฟล์ข้อมูลภายนอกทุกไฟล์จะทำงาน formatting, macros, data types, etc อาจทำให้คำสั่งนำเข้าไม่สามารถอ่านไฟล์ได้ ซึ่ง ณ จุดนี้ คุณต้องแก้ไขไฟล์ของคุณให้เข้ากันได้กับ SQL ก่อน
- รายละเอียดของประเภทไฟล์ที่เข้ากันได้และตัวอย่างมีอยู่ใน online documentation:

### COPY

COPY COPY — copy data between a file and a table Synopsis COPY table\_name [ ( column\_name [, ...] ) ...


 <https://www.postgresql.org/docs/12/sql-copy.html>



- คุณต้องระบุ path ไฟล์ที่ถูกต้อง 100% ไปยังไฟล์ภายนอกของคุณ มิฉะนั้น คำสั่ง Import จะค้นหาไฟล์ไม่พบ
  - ข้อผิดพลาดที่พบบ่อยที่สุดหากไม่สามารถระบุ path ของไฟล์ที่ถูกต้อง ให้ยืนยัน location under its properties
  - คำสั่ง Import ไม่ได้สร้างตารางให้คุณ
  - จะถือว่าตารางถูกสร้างขึ้นแล้ว (ต้องสร้างตารางมารองรับ)
  - ตอนนี้ (2020) ยังไม่มีวิธีอัตโนมัติภายใน pgAdmin ในการสร้างตารางโดยตรงจากไฟล์ .csv
- 
- อ่านเพิ่มเติมเกี่ยวกับการ Import/Export ข้อมูล

### How to import CSV file data into a PostgreSQL table


How can I write a stored procedure that imports data from a CSV file and populates the table?

 <https://stackoverflow.com/questions/2987433/how-to-import-csv-file-data-into-a-postgresql-table>




### How to import and export data using CSV files in PostgreSQL

This article explains how to import data from a CSV file into PostgreSQL and how to export it back from PostgreSQL to CSV. It includes an introduction to the CSV file format and some examples of its usage.

 <https://www.enterprisedb.com/postgres-tutorials/how-import-and-export-data-using-csv-files-postgresql>

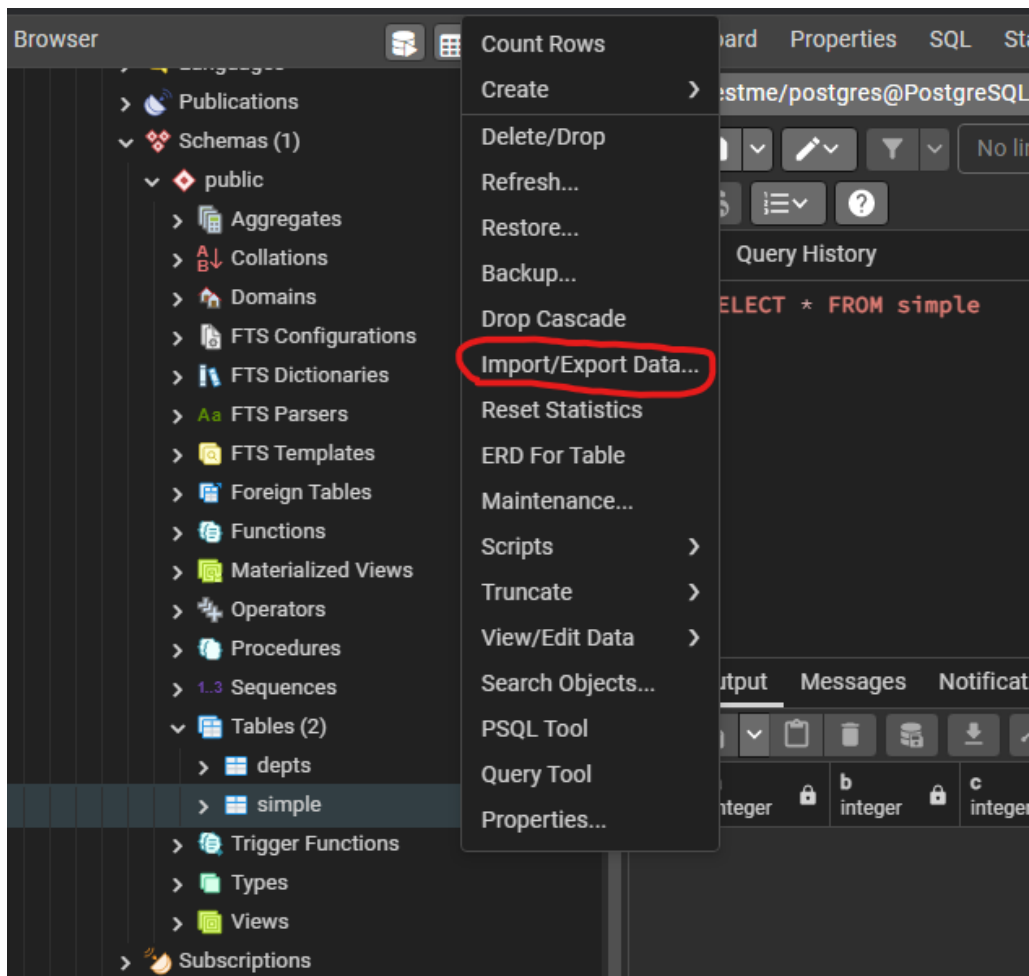
### Can I automatically create a table in PostgreSQL from a csv file with headers?

I'm running PostgreSQL 9.2.6 on OS X 10.6.8. I would like to import data from a CSV file with column headers into a database. I can do this with the COPY statement, but only if I first manually cre...

 <https://stackoverflow.com/questions/21018256/can-i-automatically-create-a-table-in-postgresql-from-a-csv-file-with-headers>



```
# สร้างตารางรองรับข้อมูลจากการ Import
CREATE TABLE simple(
a INTEGER,
b INTEGER,
c INTEGER
)
```



- เอา Header ออกเนื่องจากเราสร้างไว้แล้ว ไม่อยากทำซ้ำ

Import/Export data - table 'simple'

General Options Columns

OID ☐

Header ☒

Delimiter  | v

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Quote  X | v

Specifies the quoting character to be used when a data value is quoted. The default is double-quote. This must be a single one-byte character. This option is allowed only when using CSV format.

Escape  X | v

Specifies the character that should appear before a data character that matches the QUOTE value. The default is the same as the QUOTE value (so that the quoting character is doubled if it appears in the data). This must

i ? X Close ↺ Reset ✓ OK

- หากมีคอลัมน์ที่นอกเหนือจากคอลัมน์ที่เราสร้างมาในตารางให้ไปเอาออก



Import/Export data - table 'simple'

General

Options

Columns

Columns to import

a x

b x

c x

An optional list of columns to be copied. If no column list is specified, all columns of the table will be copied.

NOT NULL columns

Not null columns...

Do not match the specified column values against the null string. In the default case where the null string is empty, this means that empty values will be read as zero-length strings rather than nulls, even when they are not quoted. This option is allowed only in import, and only when using CSV format.

i

?

Close

Reset

OK

Data Output

Messages

Notifications

|   | <div>a</div> <div>integer</div> <div></div> | <div>b</div> <div>integer</div> <div></div> | <div>c</div> <div>integer</div> <div></div> |  |
|---|---|---|---|--|
| 1 | 1   | 4   | 7   |  |
| 2 | 2   | 5   | 8   |  |
| 3 | 3   | 6   | 9   |  |

- ส่วน export ก็ทำง่ายมากในเมนูเดียวกันเลย
-