CASE Statements

Session Objectives

- Understand how SQL CASE statements work.
- Use CASE WHEN to create new flags/segments.
- Apply conditional logic for aggregation.
- Use CASE in SELECT, UPDATE, and JOIN.

SYNTAX:

```
SELECT column1,
       CASE
           WHEN condition1 THEN result1
           WHEN condition2 THEN result2
           ELSE default_result
       END AS alias_name
FROM table_name;
```

This will create a new column based on some conditions.

Categorizing the AnnualIncome -> By creating a new column of income Category

```
SELECT
        CustomerKey,
        AnnualIncome,
    CASE
        WHEN AnnualIncome IS NULL THEN 'Not Available'
        WHEN AnnualIncome < 50000 THEN 'Low Income'
        WHEN AnnualIncome BETWEEN 50000 AND 100000 THEN 'Moderate Income'
        ELSE 'High Income'
    END AS IncomeCategory
FROM Customers;
```

CustomerKey	AnnualIncome	IncomeCategory
11000	90000	Moderate Income
11001	60000	Moderate Income
11002	60000	Moderate Income
11003	NULL	Not Available
11004	80000	Moderate Income
11005	70000	Moderate Income
11007	60000	Moderate Income
11008	60000	Moderate Income

```
SELECT
        CustomerKey,
        EducationLevel,
    CASE
        WHEN EducationLevel IN ("High School", "Partial High School") THEN "High School"
        WHEN EducationLevel IN ("Partial College", "Bachelors") THEN "Bachelors"
        ELSE 'Graduate Degree'
    END AS NewEducationLevel
FROM Customers;
```

CustomerKey	EducationLevel	NewEducationLevel
11013	Bachelors	Bachelors
11014	Bachelors	Bachelors
11015	Partial College	Bachelors
11016	Partial College	Bachelors
11017	High School	High School
11018	Partial College	Bachelors
11019	High School	High School
11020	High School	High School

```
-- UPDATE THE IncomeCategory in Customer Table.
ALTER TABLE Customers
ADD COLUMN IncomeCategory VARCHAR(100)
AFTER AnnualIncome;
SELECT * FROM Customers;
```

AnnualIncome	IncomeCategory
90000	NULL
60000	NULL
60000	NULL
NULL	NULL
80000	NULL
70000	NULL
60000	NULL

```
SET SQL_SAFE_UPDATES = 0;

UPDATE Customers
SET IncomeCategory = CASE
    WHEN AnnualIncome IS NULL THEN 'Not Available'
    WHEN AnnualIncome < 50000 THEN 'Low Income'
    WHEN AnnualIncome BETWEEN 50000 AND 100000 THEN 'Moderate Income'
    ELSE 'High Income'
END;
```

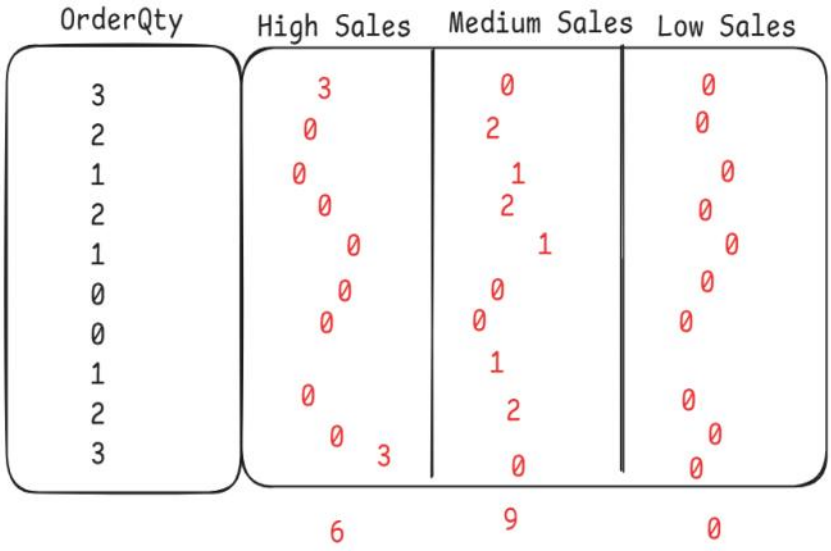
Gender	EmailAddress	AnnualIncome	IncomeCategory
M	jon24@learnsector.com	90000	Moderate Income
M	eugene10@learnsector.com	60000	Moderate Income
M	ruben35@learnsector.com	60000	Moderate Income
F	christy12@learnsector.com	NULL	Not Available
F	elizabeth5@learnsector.com	80000	Moderate Income
M	julio1@learnsector.com	70000	Moderate Income
M	marco14@learnsector.com	60000	Moderate Income

Conditional Aggregation with CASE :

```
SELECT column,
       SUM(CASE WHEN condition THEN value ELSE 0 END) AS label
FROM table
GROUP BY column;
```

Find the territorykey wise orderquantity:

1. OrderQty > 2 -> High Sales
2. OrderQty [1,2] Inclusive -> Medium Sales
3. OrderQty < 1 -> Low Sales



Qty > 2 -> Qty Else 0

Qty[1,2] -> Qty Else 0

Qty < 1 -> Qty Else 0

```
SELECT
    TerritoryKey,
    SUM(CASE WHEN OrderQuantity>2 THEN OrderQuantity ELSE 0 END) AS HighSales,
    SUM(CASE WHEN OrderQuantity BETWEEN 1 AND 2 THEN OrderQuantity ELSE 0 END) AS MediumSales,
    SUM(CASE WHEN OrderQuantity < 1 THEN OrderQuantity ELSE 0 END) AS LowSales
FROM `sales-2017`
GROUP BY 1;
```

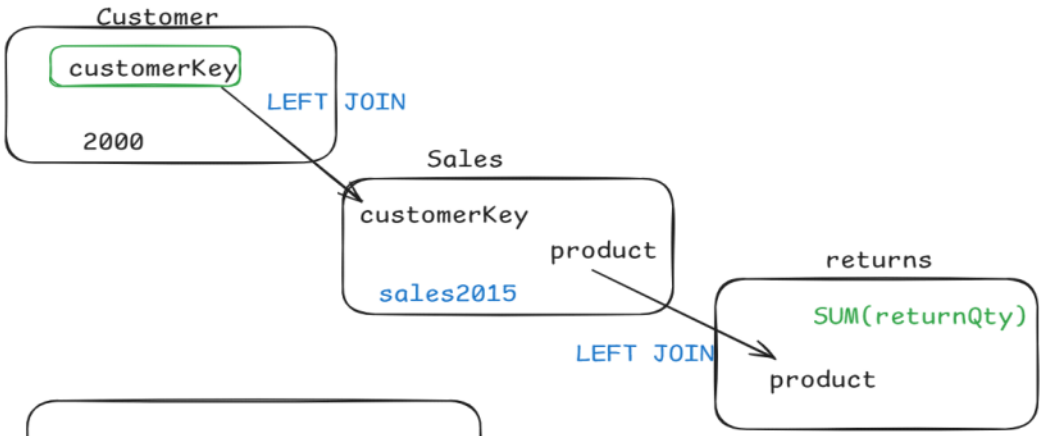
TerritoryKey	HighSales	MediumSales	LowSales
6	852	5008	0
10	630	4463	0
8	600	3786	0
9	1185	8302	0
7	498	3738	0
5	0	35	0
3	0	14	0
2	3	21	0

```
SELECT
        TerritoryKey,
    SUM(CASE WHEN OrderQuantity>3 THEN OrderQuantity ELSE 0 END) AS HighSales,
    SUM(CASE WHEN OrderQuantity BETWEEN 2 AND 3 THEN OrderQuantity ELSE 0 END) AS MediumSales,
    SUM(CASE WHEN OrderQuantity < 2 THEN OrderQuantity ELSE 0 END) AS LowSales
FROM `sales-2017`
GROUP BY 1;
```

TerritoryKey	HighSales	MediumSales	LowSales
1	0	4471	2318
4	0	6108	3282
6	0	4092	1768
10	0	3386	1707
8	0	2870	1516
9	0	5949	3538
7	0	2794	1442
5	0	20	15

Challenge 1

Classify the customer based on how frequently they return product.




```
>10 -> Frequent Returner

1 - 10 -> Moderate Returner

ELSE -> NON Returner.
```

```
-- Classify the customer based on how frequently they return product.
SELECT
    c.CustomerKey,
    c.FullName,
    SUM(COALESCE(r.ReturnQuantity,0)) AS TotalReturns,
    CASE
        WHEN SUM(COALESCE(r.ReturnQuantity,0)) > 10 THEN 'Frequent Returner'
        WHEN SUM(COALESCE(r.ReturnQuantity,0)) BETWEEN 1 AND 10 THEN 'Occasional Returner'
        ELSE 'Non-Returner'
    END AS ReturnBehaviour
FROM Customers c
LEFT JOIN `sales-2015` s
ON c.CustomerKey = s.CustomerKey
LEFT JOIN returns r
ON r.ProductKey = s.ProductKey
GROUP BY 1,2
ORDER BY 3 DESC;
```

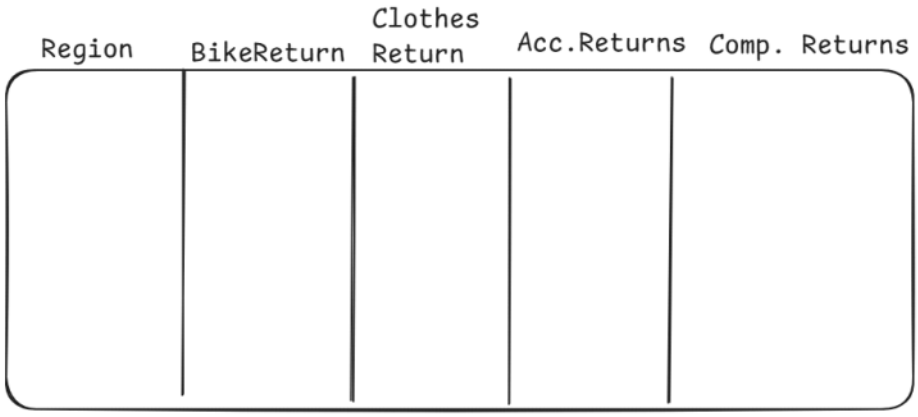
CustomerKey	FullName	TotalReturns	ReturnBehaviour
11605	HEIDI SUBRAM	21	Frequent Returner
11612	KEITH ANDERSEN	21	Frequent Returner
12123	WESLEY LIANG	21	Frequent Returner
12128	KRISTY MUNOZ	21	Frequent Returner
12214	JULIE RAJE	21	Frequent Returner

CustomerKey	FullName	TotalReturns	ReturnBehaviour
13068	ALEXANDRIA MO...	11	Frequent Returner
13069	ADAM GREEN	11	Frequent Returner
13055	PATRICK GRAY	9	Occasional Returner
13062	EDUARDO WILLI...	9	Occasional Returner
13083	JORDYN BUTLER	9	Occasional Returner
11270	ROBERT LEE	8	Occasional Returner
11831	CHRISTIAN ROSS	8	Occasional Returner

CustomerKey	FullName	TotalReturns	ReturnBehaviour
12253	JENNY SUN	1	Occasional Returner
12337	DUSTIN GOLDST...	1	Occasional Returner
11418	RAFAEL HU	0	Non-Returner
11419	KYLE SCOTT	0	Non-Returner
11420	JORDAN TURNER	0	Non-Returner
11421	AMY SUN	0	Non-Returner
11422	DUSTIN DENG	0	Non-Returner

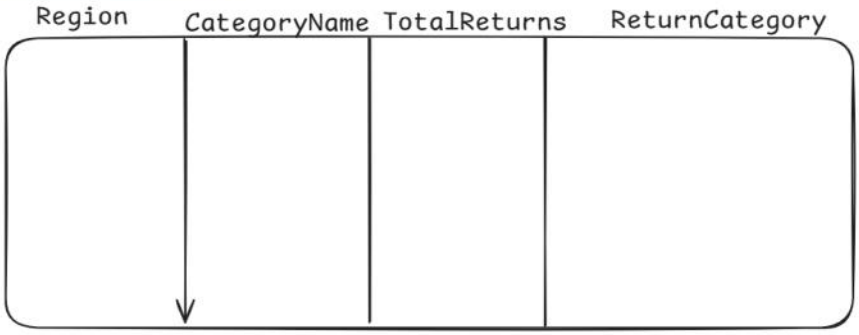
Challenge 2

- Show the Total return quantity for Each Product Category break down by region.




```
-- - Show the Total return quantity for Each Product Category break down by region.
SELECT
    Region,
    SUM(CASE WHEN CategoryName = "Bikes" THEN returnQuantity ELSE 0 END ) AS BikeReturns,
    SUM(CASE WHEN CategoryName = "Clothing" THEN returnQuantity ELSE 0 END ) AS ClothingReturns,
    SUM(CASE WHEN CategoryName = "Accessories" THEN returnQuantity ELSE 0 END ) AS AccessoriesReturns,
    SUM(CASE WHEN CategoryName = "Components" THEN returnQuantity ELSE 0 END ) AS ComponentsReturns
FROM `product-categories` pc
JOIN `product-subcategories` ps
ON ps.ProductCategoryKey = pc.ProductCategoryKey
JOIN Products p
ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
JOIN returns r
ON r.ProductKey = p.ProductKey
JOIN territories t
ON r.TerritoryKey = t.SalesTerritoryKey
GROUP BY 1;
```

Challenge 3



>50 -> High Returns

>25 - Moderate Return

ELSE - No Return

```
-- Another CASE Query
SELECT
    Region,
    CategoryName,
    SUM(ReturnQuantity) AS TotalReturns,
    CASE
        WHEN SUM(ReturnQuantity) > 50 THEN 'High Returns'
        WHEN SUM(ReturnQuantity) > 25 THEN 'Moderate Returns'
        ELSE 'Low Returns'
    END AS ReturnCategory
FROM `product-categories` pc
JOIN `product-subcategories` ps
ON ps.ProductCategoryKey = pc.ProductCategoryKey
JOIN Products p
ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
JOIN returns r
ON r.ProductKey = p.ProductKey
JOIN territories t
ON r.TerritoryKey = t.SalesTerritoryKey
GROUP BY 1,2;
```

Region	CategoryName	TotalReturns	ReturnCategory
Australia	Accessories	223	High Returns
United Kingdom	Accessories	129	High Returns
France	Accessories	121	High Returns
Southwest	Accessories	231	High Returns
Northwest	Accessories	172	High Returns
Germany	Accessories	88	High Returns
Canada	Accessories	165	High Returns
Southeast	Accessories	1	Low Returns
Australia	Bikes	125	High Returns
United Kingdom	Bikes	51	High Returns
Germany	Bikes	53	High Returns
Southwest	Bikes	78	High Returns
Canada	Bikes	22	Low Returns
Northwest	Bikes	58	High Returns
France	Bikes	42	Moderate Returns
Germany	Clothing	22	Low Returns
Northwest	Clothing	40	Moderate Returns
Australia	Clothing	56	High Returns
Canada	Clothing	51	High Returns
United Kingdom	Clothing	24	Low Returns
Southwest	Clothing	53	High Returns
France	Clothing	23	Low Returns