Advanced Concepts in Joins - II

IFNULL() VS COALESCE() ..... They both are used to handle the null values.

IFNULL(exp1,exp2)

COALESCE(exp1,exp2,exp3,.....)

ANNUALINCOME(null,"Unknown")

```
SELECT
    CustomerKey,
    FullName,
    Occupation,
    IFNULL(AnnualIncome,"Unknown") AS AnnualIncome
FROM Customers;
```

CustomerKey	FullName	Occupation	AnnualIncome
11000	JON YANG	Professional	90000
11001	EUGENE HUANG	Professional	60000
11002	RUBEN TORRES	Professional	60000
11003	CHRISTY ZHU	Professional	Unknown
11004	ELIZABETH JOHNSON	Professional	80000
11005	JULIO RUIZ	Professional	70000
11007	MARCO MEHTA	Professional	60000
11008	ROBIN VERHOEF	Professional	60000

```
SELECT
    CustomerKey,
    FullName,
    Occupation,
    COALESCE(AnnualIncome,0) AS AnnualIncome
FROM Customers;
```

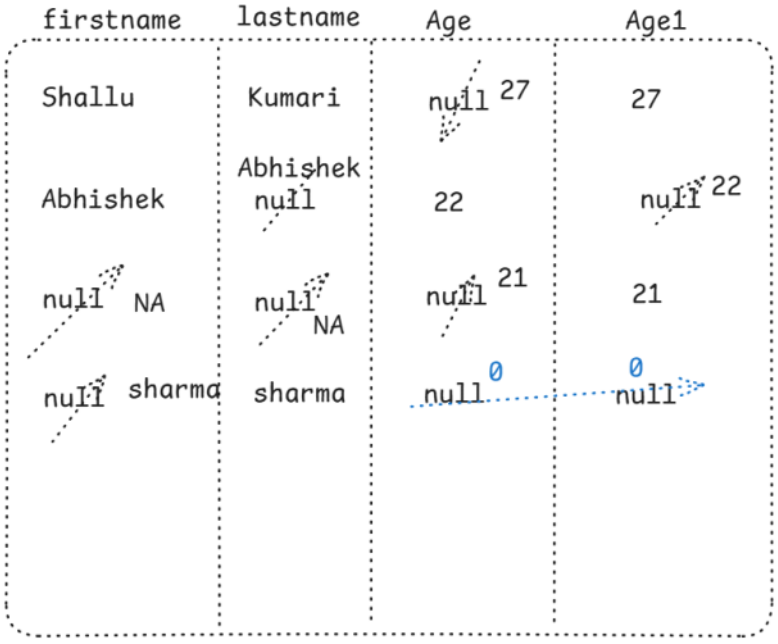
```
111 •  SELECT COUNT(*) FROM Customers WHERE AnnualIncome IS NULL;
112
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COUNT(*)
10

```
COALESCE(AnnualIncome,PreviousYearIncome, 0)
```

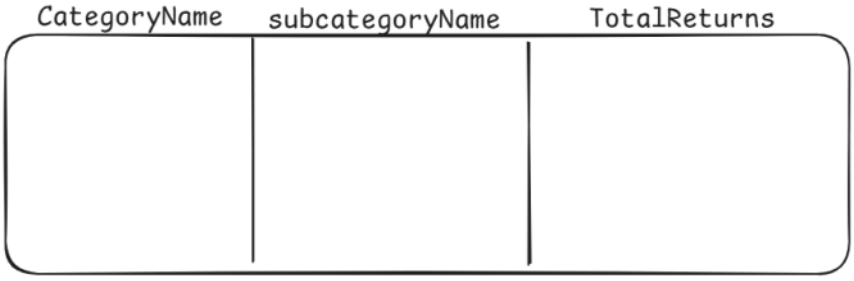
if it checks multiple columns in order, the first non-Null wins.

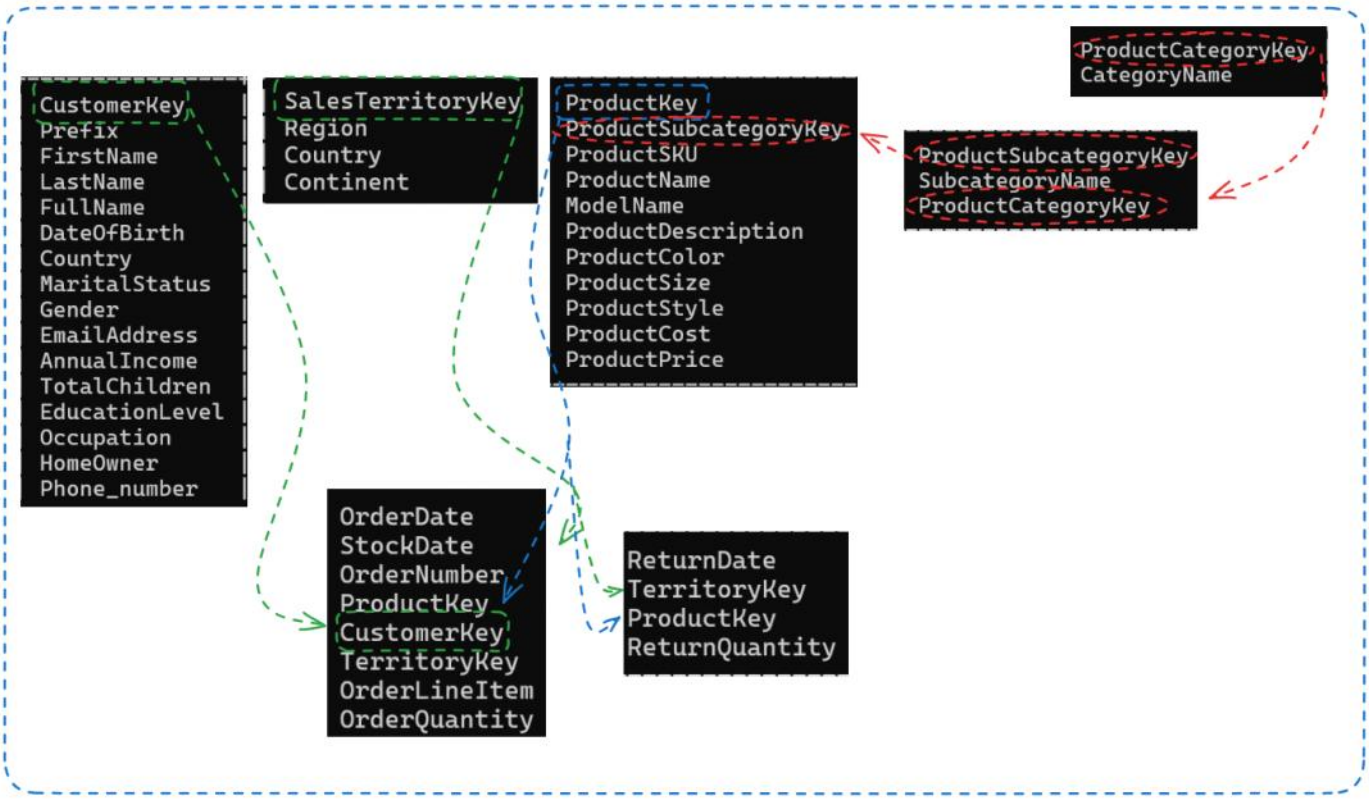



```
SELECT
COALESCE(firstname,lastname,"NA") AS firstname,
COALESCE(lastname,firstName,"NA") AS lastname,
COALESCE(age,age1,0) AS age,
COALESCE(age1,age,0) AS age1
```

Challenge 1

Calculate the Totalreturns by categoryName, SubcategoryName.






```sql
-- Calculate the Totalreturns by categoryName, SubcategoryName.
SELECT
    pc.CategoryName,
    ps.SubcategoryName,
    SUM(r.ReturnQuantity) AS TotalReturns
FROM `product-categories` pc
JOIN `product-subcategories` ps
ON pc.ProductCategoryKey = ps.ProductCategoryKey
JOIN Products p
ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
JOIN Returns r
ON p.ProductKey = r.ProductKey
GROUP BY 1,2
ORDER BY 3 DESC
LIMIT 10;
```

CategoryName	SubcategoryName	TotalReturns
Accessories	Tires and Tubes	534
Accessories	Bottles and Cages	288
Bikes	Road Bikes	223
Accessories	Helmets	188
Bikes	Mountain Bikes	136
Clothing	Jerseys	93
Bikes	Touring Bikes	70
Accessories	Fenders	54
Clothing	Gloves	49
Clothing	Caps	46

Challenge 2

Find the territories with high returns(>200)

```
131 •  SELECT * FROM territories;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

SalesTerritoryKey	Region	Country	Continent
1	Northwest	United States	North America
2	Northeast	United States	North America
3	Central	United States	North America
4	Southwest	United States	North America
5	Southeast	United States	North America
6	Canada	Canada	North America
7	France	France	Europe
8	Germany	Germany	Europe
9	Australia	Australia	Pacific
10	United Kingdom	United Kingdom	Europe

(returnQty)

```
132 •  SELECT DISTINCT TerritoryKey FROM returns;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TerritoryKey
9
10
8
4
6
1
7
5
NULL

```
SELECT
    SalesTerritoryKey,
    region,
    Country,
    SUM(ReturnQuantity) AS TotalReturns
FROM territories t
JOIN returns r
ON t.SalesTerritoryKey = r.TerritoryKey
GROUP BY 1,2,3;
```

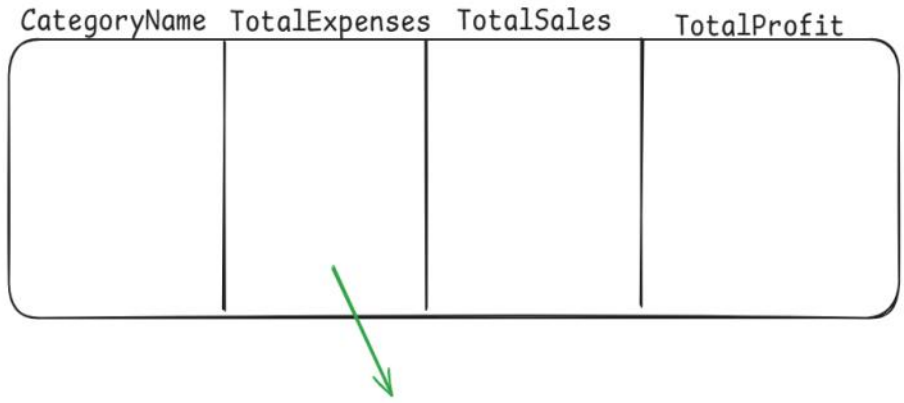
>200

SalesTerritoryKey	region	Country	TotalReturns
9	Australia	Australia	404
10	United Kingdom	United Kingdom	204
8	Germany	Germany	163
4	Southwest	United States	362
6	Canada	Canada	238
1	Northwest	United States	270
7	France	France	186
5	Southeast	United States	1

```
SELECT
    SalesTerritoryKey,
    region,
    Country,
    SUM(ReturnQuantity) AS TotalReturns
FROM territories t
JOIN returns r
ON t.SalesTerritoryKey = r.TerritoryKey
GROUP BY 1,2,3
HAVING TotalReturns > 200;
```

SalesTerritoryKey	region	Country	TotalReturns
9	Australia	Australia	404
10	United Kingdom	United Kingdom	204
4	Southwest	United States	362
6	Canada	Canada	238
1	Northwest	United States	270

Challenge 3

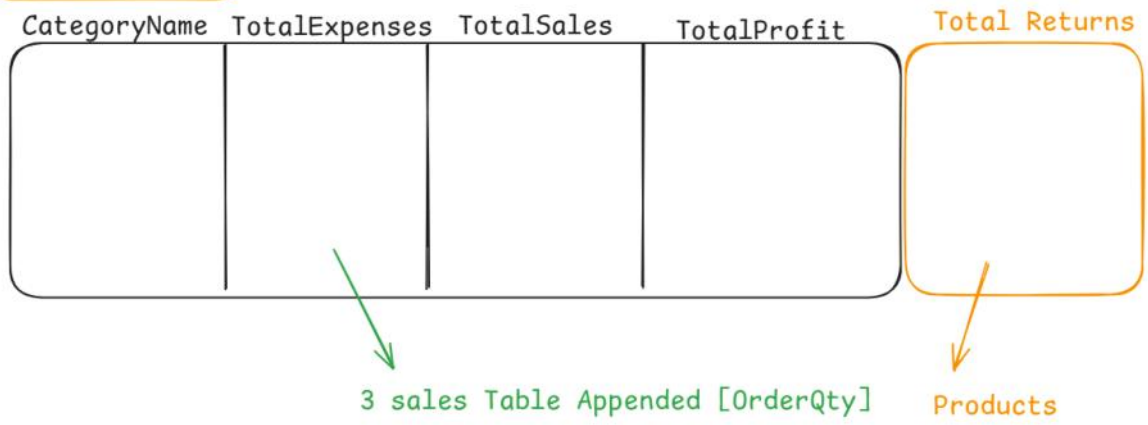


CategoryName	TotalExpenses	TotalSales	TotalProfit
Accessories	336913	906673	569760
Clothing	203632	365419	161787
Bikes	13916327	23642495	9726168

```
-- Find the CategoryName with TotalExpenses, TotalSales & TotalProfit
SELECT
    pc.CategoryName,
    ROUND(SUM(p.ProductCost * s.OrderQuantity),0) AS TotalExpenses,
    ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalSales,
    ROUND((SUM(p.ProductPrice * s.OrderQuantity) - SUM(p.ProductCost * s.OrderQuantity)),0)
    AS TotalProfit
FROM(
    SELECT * FROM `sales-2015`
    UNION ALL
    SELECT * FROM `sales-2016`
    UNION ALL
    SELECT * FROM `sales-2017`
) AS s
JOIN Products p
ON s.ProductKey = p.ProductKey
JOIN `product-subcategories` ps
ON ps.ProductSubcategoryKey = p.ProductSubcategoryKey
JOIN `product-categories` pc
ON pc.ProductCategoryKey = ps.ProductCategoryKey
GROUP BY pc.CategoryName;
```

Sales Appended Table

Challenge 4

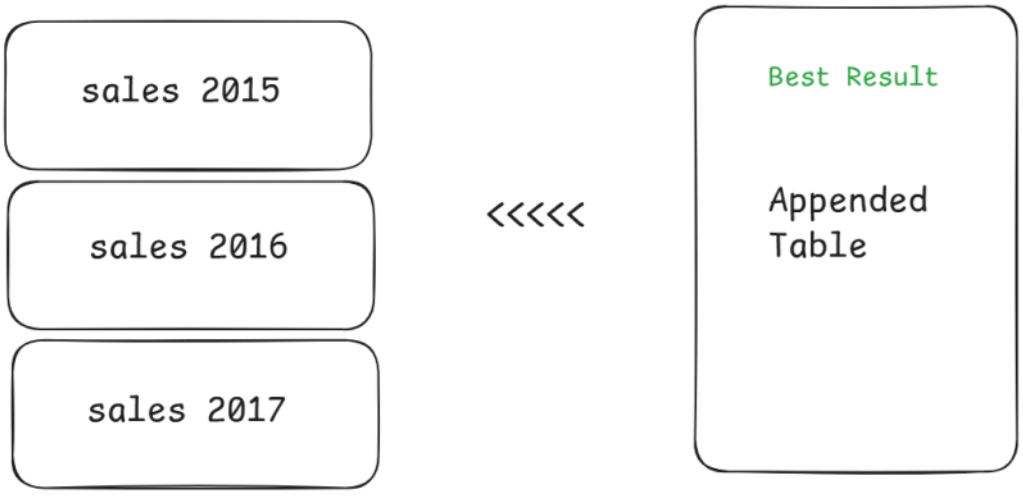


CategoryName	TotalExpenses	TotalSales	TotalProfit	TotalReturns
Bikes	123822283	215882898	92060615	121650
Accessories	16969493	45717781	28748288	2366328
Clothing	3241572	5669837	2428265	174888

```
SELECT
    pc.CategoryName,
    ROUND(SUM(p.ProductCost * s.OrderQuantity),0) AS TotalExpenses,
    ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalSales,
    ROUND((SUM(p.ProductPrice * s.OrderQuantity) - SUM(p.ProductCost * s.OrderQuantity)),0)
    AS TotalProfit,
    SUM(r.ReturnQuantity) AS TotalReturns
FROM(
    SELECT * FROM `sales-2015`
    UNION ALL
    SELECT * FROM `sales-2016`
    UNION ALL
    SELECT * FROM `sales-2017`
) AS s
JOIN Products p
ON s.ProductKey = p.ProductKey
JOIN `product-subcategories` ps
ON ps.ProductSubcategoryKey = p.ProductSubcategoryKey
JOIN `product-categories` pc
ON pc.ProductCategoryKey = ps.ProductCategoryKey
JOIN Returns r
ON p.ProductKey = r.ProductKey
GROUP BY pc.CategoryName;
```

Challenge 5

Top 5 Customers By Sales Quantity



CustomerKey	CustomerName	TotalSalesQ
11262	JENNIFER SIMMONS	74
11300	FERNANDO BARNES	74
11185	ASHLEY HENDERSON	72
11331	SAMANTHA JENKINS	62
11223	HAILEY PATTERSON	59

<<<<<

CustomerKey	CustomerName	TotalSalesQty
11262	JENNIFER SIMMONS	106
11300	FERNANDO BARNES	106
11331	SAMANTHA JENKINS	102
11185	ASHLEY HENDERSON	100
11566	APRIL SHAN	99

```
SELECT
    c.CustomerKey,
    CONCAT(c.FirstName, " ", c.LastName) AS CustomerName,
    SUM(s2015.OrderQuantity) AS TotalSalesQty
FROM `sales-2015` s2015
JOIN Customers c
ON s2015.CustomerKey = c.CustomerKey
GROUP BY 1,2
UNION ALL
SELECT
    c.CustomerKey,
    CONCAT(c.FirstName, " ", c.LastName) AS CustomerName,
    SUM(s2016.OrderQuantity) AS TotalSalesQty
FROM `sales-2016` s2016
JOIN Customers c
ON s2016.CustomerKey = c.CustomerKey
GROUP BY 1,2
UNION ALL
SELECT
    c.CustomerKey,
    CONCAT(c.FirstName, " ", c.LastName) AS CustomerName,
    SUM(s2017.OrderQuantity) AS TotalSalesQty
FROM `sales-2017` s2017
JOIN Customers c
ON s2017.CustomerKey = c.CustomerKey
GROUP BY 1,2
ORDER BY TotalSalesQty DESC
LIMIT 5;
```

```sql
-- APPENDED TABLE
SELECT
        c.CustomerKey,
    CONCAT(c.FirstName, " ", c.LastName) AS CustomerName,
    SUM(s.OrderQuantity) AS TotalSalesQty
FROM(
        SELECT * FROM `sales-2015`
    UNION ALL
    SELECT * FROM `sales-2016`
    UNION ALL
    SELECT * FROM `sales-2017`
) s
JOIN Customers c
ON s.CustomerKey = c.CustomerKey
GROUP BY 1,2
ORDER BY 3 DESC
LIMIT 5;
```