Advanced Concepts in Joins

territories

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

RETURN TABLE

TerritoryKey
9
10
8
4
6
1
7
5

2,3

RIGHT JOIN

Observing NULL values on Left Table.

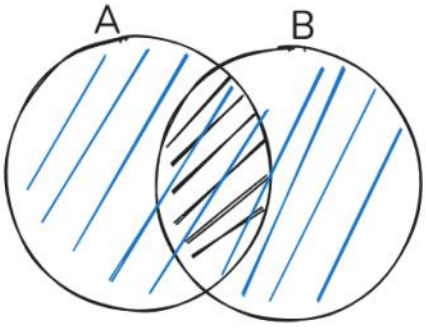
```
961 •  INSERT INTO Returns(ProductKey,ReturnQuantity)
962    VALUES(999991,1),
963    (999992,2),
964    (999993,3),
965    (999994,4),
966    (999995,5),
967    (999996,6),
```

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

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
NULL	NULL	NULL	45

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

Full JOIN ⟶ LEFT + UNION + RIGHT



FULL JOIN

LEFT JOIN

Union (U)

RIGHT JOIN

UNION - Append
[removes duplicates]
UNION ALL - Append

```
998 • SELECT t.*, r.*
999   FROM territories t
1000  LEFT JOIN returns r
1001  ON t.SalesTerritoryKey = r.TerritoryKey;
1002  -- UNION
1003 • SELECT t.*, r.*
1004  FROM territories t
1005  RIGHT JOIN returns r
1006  ON t.SalesTerritoryKey = r.TerritoryKey;
```

1

2

3

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

SalesTerritoryKey	Region	Country	Continent	ReturnDate	TerritoryKey	ProductKey	ReturnQuantity
1	Northwest	United States	North America	6/28/2017	1	587	1
1	Northwest	United States	North America	6/28/2017	1	480	1
1	Northwest	United States	North America	6/28/2017	1	214	1
1	Northwest	United States	North America	6/26/2017	1	472	1
1	Northwest	United States	North America	6/24/2017	1	487	1
1	Northwest	United States	North America	6/24/2017	1	485	1
1	Northwest	United States	North America	6/24/2017	1	480	1
1	Northwest	United States	North America	6/24/2017	1	479	1
1	Northwest	United States	North America	6/22/2017	1	485	1
1	Northwest	United States	North America	6/22/2017	1	232	1

SalesTerritoryKey	Region	Country	Continent	ReturnDate	TerritoryKey	ProductKey	ReturnQuantity
9	Australia	Australia	Pacific	1/18/2015	9	312	1
10	United Kingdom	United Kingdom	Europe	1/18/2015	10	310	1
8	Germany	Germany	Europe	1/21/2015	8	346	1
4	Southwest	United States	North America	1/22/2015	4	311	1
6	Canada	Canada	North America	2/2/2015	6	312	1
1	Northwest	United States	North America	2/15/2015	1	312	1
9	Australia	Australia	Pacific	2/19/2015	9	311	1
8	Germany	Germany	Europe	2/24/2015	8	314	1
8	Germany	Germany	Europe	3/8/2015	8	350	1
9	Australia	Australia	Pacific	3/13/2015	9	350	1

```
998 • SELECT t.*, r.*
999   FROM territories t
1000  LEFT JOIN returns r
1001  ON t.SalesTerritoryKey = r.TerritoryKey
1002  UNION
1003  SELECT t.*, r.*
1004  FROM territories t
1005  RIGHT JOIN returns r
1006  ON t.SalesTerritoryKey = r.TerritoryKey;
```

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

SalesTerritoryKey	Region	Country	Continent	ReturnDate	TerritoryKey	ProductKey	ReturnQuantity
1	Northwest	United States	North America	6/28/2017	1	587	1
1	Northwest	United States	North America	6/28/2017	1	480	1
1	Northwest	United States	North America	6/28/2017	1	214	1
1	Northwest	United States	North America	6/26/2017	1	472	1
1	Northwest	United States	North America	6/24/2017	1	487	1
1	Northwest	United States	North America	6/24/2017	1	485	1
1	Northwest	United States	North America	6/24/2017	1	480	1
1	Northwest	United States	North America	6/24/2017	1	479	1
1	Northwest	United States	North America	6/22/2017	1	485	1
1	Northwest	United States	North America	6/22/2017	1	232	1

Final Table

9	21:22:53	SELECT t.*, r.* FROM territories t LEFT JOIN returns r ON t.SalesTerritoryKey = r.TerritoryKey LIMIT 0, 1000	1000 row(s) returned
10	21:24:44	SELECT t.*, r.* FROM territories t RIGHT JOIN returns r ON t.SalesTerritoryKey = r.TerritoryKey LIMIT 0, 1000	1000 row(s) returned
11	21:25:28	SELECT t.*, r.* FROM territories t LEFT JOIN returns r ON t.SalesTerritoryKey = r.TerritoryKey UNION SELECT...	1820 row(s) returned

Challenge 1: Display the customer full Name and what product they bought from all the sales records.


```sql
SELECT
    CONCAT(c.FirstName, ' ', c.LastName) AS FullName,
    ProductName,
    s.orderQuantity
FROM (
    SELECT * FROM `sales-2015`
    UNION ALL
    SELECT * FROM `sales-2016`
    UNION ALL
    SELECT * FROM `sales-2017`
) AS s
JOIN Customers c
ON c.CustomerKey = s.CustomerKey
JOIN Products p
ON p.ProductKey = s.ProductKey;
```

UNION
VS
UNION ALL

FullName	ProductName	orderQuantity
JON YANG	Sport-100 Helmet, Red	1
ELIZABETH JOHNSON	Sport-100 Helmet, Red	1
MARCO MEHTA	Sport-100 Helmet, Red	1
ROBIN VERHOFF	Sport-100 Helmet, Red	1
CAROL RAI	Sport-100 Helmet, Red	1
DEANNA MUNOZ	Sport-100 Helmet, Red	1
JON ZHOU	Sport-100 Helmet, Red	1
TODD GAO	Sport-100 Helmet, Red	1
LEAH YE	Sport-100 Helmet, Red	1
GINA MARTIN	Sport-100 Helmet, Red	1

FullName	ProductName	TotalQty
APRIL SHAN	Touring Tire Tube	28
SAMANTHA JENKINS	Patch Kit/8 Patches	21
DALTON PEREZ	Patch Kit/8 Patches	21
JENNIFER SIMMONS	Road Tire Tube	19
JENNIFER SIMMONS	Patch Kit/8 Patches	17
MASON ROBERTS	Patch Kit/8 Patches	17
HENRY GARCIA	Patch Kit/8 Patches	17
ASHLEY HENDERSON	Patch Kit/8 Patches	16
ASHLEY HENDERSON	Road Tire Tube	15
CHARLES JACKSON	Road Tire Tube	15

```
SELECT
        CONCAT(c.FirstName, ' ', c.LastName) AS FullName,
    ProductName,
    SUM(s.orderQuantity) AS TotalQty
FROM (
        SELECT * FROM `sales-2015`
    UNION ALL
    SELECT * FROM `sales-2016`
    UNION ALL
    SELECT * FROM `sales-2017`
) AS s
JOIN Customers c
ON c.CustomerKey = s.CustomerKey
JOIN Products p
ON p.ProductKey = s.ProductKey
GROUP BY 1,2
ORDER BY 3 DESC;
```

Challenge 2: Which Product is Sold all the three years?

```
1029 •  SELECT Count(DISTINCT ProductKey) FROM `sales-2015`;
```

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

Count(DISTINCT ProductKey)
44

```
1030 •  SELECT Count(DISTINCT ProductKey) FROM `sales-2016`;
```

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

Count(DISTINCT ProductKey)
117

```
1031 •  SELECT Count(DISTINCT ProductKey) FROM `sales-2017`;
```

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

Count(DISTINCT ProductKey)
102

ProductName
Mountain-200 Silver, 38
Mountain-200 Silver, 42
Mountain-200 Silver, 46
Mountain-200 Black, 38
Mountain-200 Black, 42
Mountain-200 Black, 46
Road-250 Red, 58
Road-250 Black, 44
Road-250 Black, 48
Road-250 Black, 52
Road-250 Black, 58
Road-550-W Yellow, 38
Road-550-W Yellow, 40
Road-550-W Yellow, 42
Road-550-W Yellow, 44
Road-550-W Yellow, 48

```
SELECT DISTINCT ProductName
FROM Products p
WHERE p.ProductKey IN (SELECT ProductKey FROM `sales-2015`)
AND p.ProductKey IN (SELECT ProductKey FROM `sales-2016`)
AND p.ProductKey IN (SELECT ProductKey FROM `sales-2017`);
```

```
101 IN (101,202,231,234,531,.....)

101 IN (101,202,231,234,531,.....)

101 IN (101,202,231,234,531,.....)
```

CROSS JOIN

`product-categories` ⟶ `product-subcategories`

```
-- CROSS JOIN
SELECT pc.categoryName, ps.subcategoryName FROM
`product-categories` pc
CROSS JOIN `product-subcategories` ps;
```

categoryName	subcategoryName
Accessories	Tires and Tubes
Bikes	Mountain Bikes
Bikes	Road Bikes
Bikes	Touring Bikes
Bikes	Handlebars
Bikes	Bottom Brackets
Bikes	Brakes
Bikes	Chains
Bikes	Cranksets
Bikes	Derailleurs
Bikes	Forks
Bikes	Headsets
Bikes	Mountain Frames
Bikes	Pedals

```
1055 •  SELECT COUNT(*) FROM `product-categories`;
```

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

COUNT(*)
4

```
1055 •  SELECT COUNT(*) FROM `product-subcategories`;
```

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

COUNT(*)
37

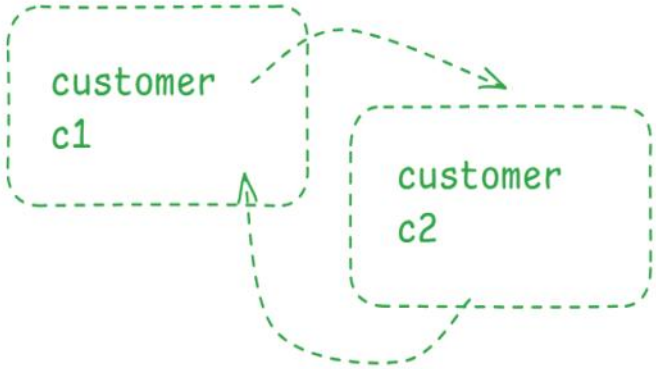
4 * 37 = 148.

148 row(s) returned

SELF JOIN

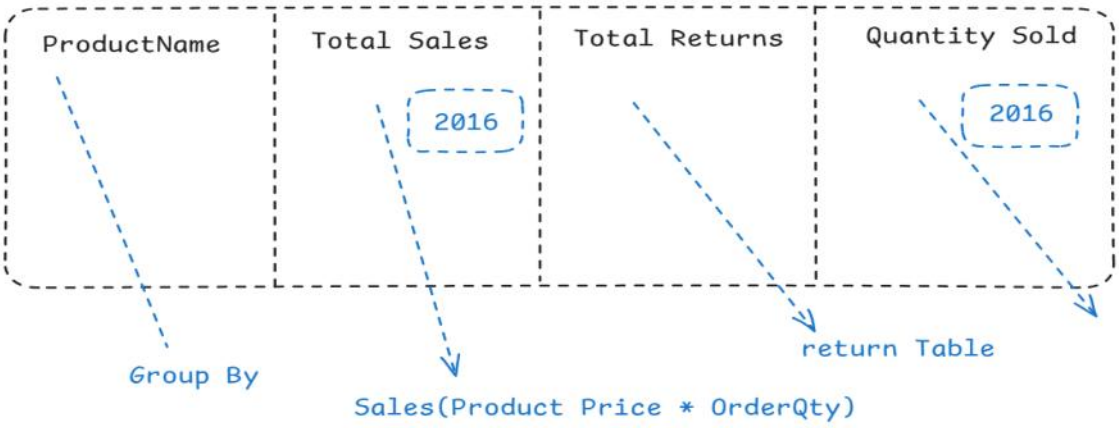
Using aliases c1 & c2 on customers helping us to perform self join.

Find the customer pairs who shares the same last name.

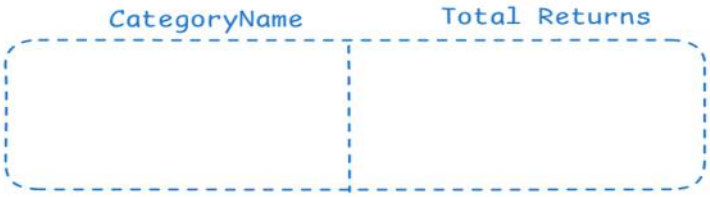


customer1_ID	Customer1_Name	customer2_ID	Customer2_Name	LastName
11463	ALISHA BECK	11579	CLAYTON BECK	BECK
11025	ALEJANDRO BECK	11579	CLAYTON BECK	BECK
11014	SYDNEY BENNETT	11199	JACQUELINE BEN...	BENNETT
11199	JACQUELINE BENNETT	11631	ANTONIO BENNETT	BENNETT
11014	SYDNEY BENNETT	11631	ANTONIO BENNETT	BENNETT
11164	DEVIN BROOKS	11617	SEAN BROOKS	BROOKS
11086	RYAN BROWN	11192	JAMES BROWN	BROWN
11192	JAMES BROWN	11218	OLIVIA BROWN	BROWN
11086	RYAN BROWN	11218	OLIVIA BROWN	BROWN
11218	OLIVIA BROWN	11334	NICOLE BROWN	BROWN
11192	JAMES BROWN	11334	NICOLE BROWN	BROWN
11086	RYAN BROWN	11334	NICOLE BROWN	BROWN
11334	NICOLE BROWN	11506	NICHOLAS BROWN	BROWN
11218	OLIVIA BROWN	11506	NICHOLAS BROWN	BROWN
11192	JAMES BROWN	11506	NICHOLAS BROWN	BROWN
11086	RYAN BROWN	11506	NICHOLAS BROWN	BROWN
11090	TREVOR BRYANT	11190	CARSON BRYANT	BRYANT
11190	CARSON BRYANT	11232	HAILEY BRYANT	BRYANT
11090	TREVOR BRYANT	11232	HAILEY BRYANT	BRYANT
11232	HAILEY BRYANT	11412	SYDNEY BRYANT	BRYANT

```
-- SELF JOIN
SELECT
    c1.customerKey AS customer1_ID,
    CONCAT(c1.FirstName, ' ', c1.LastName) AS Customer1_Name,
    c2.customerKey AS customer2_ID,
    CONCAT(c2.FirstName, ' ', c2.LastName) AS Customer2_Name,
    c1.LastName
FROM Customers c1
JOIN Customers c2
ON c1.LastName = c2.LastName
AND c1.customerKey < c2.customerKey;
```




```
SELECT
    p.ProductName,
    ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalSales,
    SUM(r.ReturnQuantity) AS TotalReturns,
    SUM(s.OrderQuantity) AS QuantitySold
FROM Products p
JOIN `sales-2016` s
ON p.ProductKey = s.ProductKey
JOIN returns r
ON p.productKey = r.ProductKey
GROUP BY 1;
```




```
SELECT
    CategoryName,
    SUM(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 r.ProductKey = p.ProductKey
GROUP BY 1;
```

CategoryName	TotalReturns
Bikes	429
Accessories	1130
Clothing	269