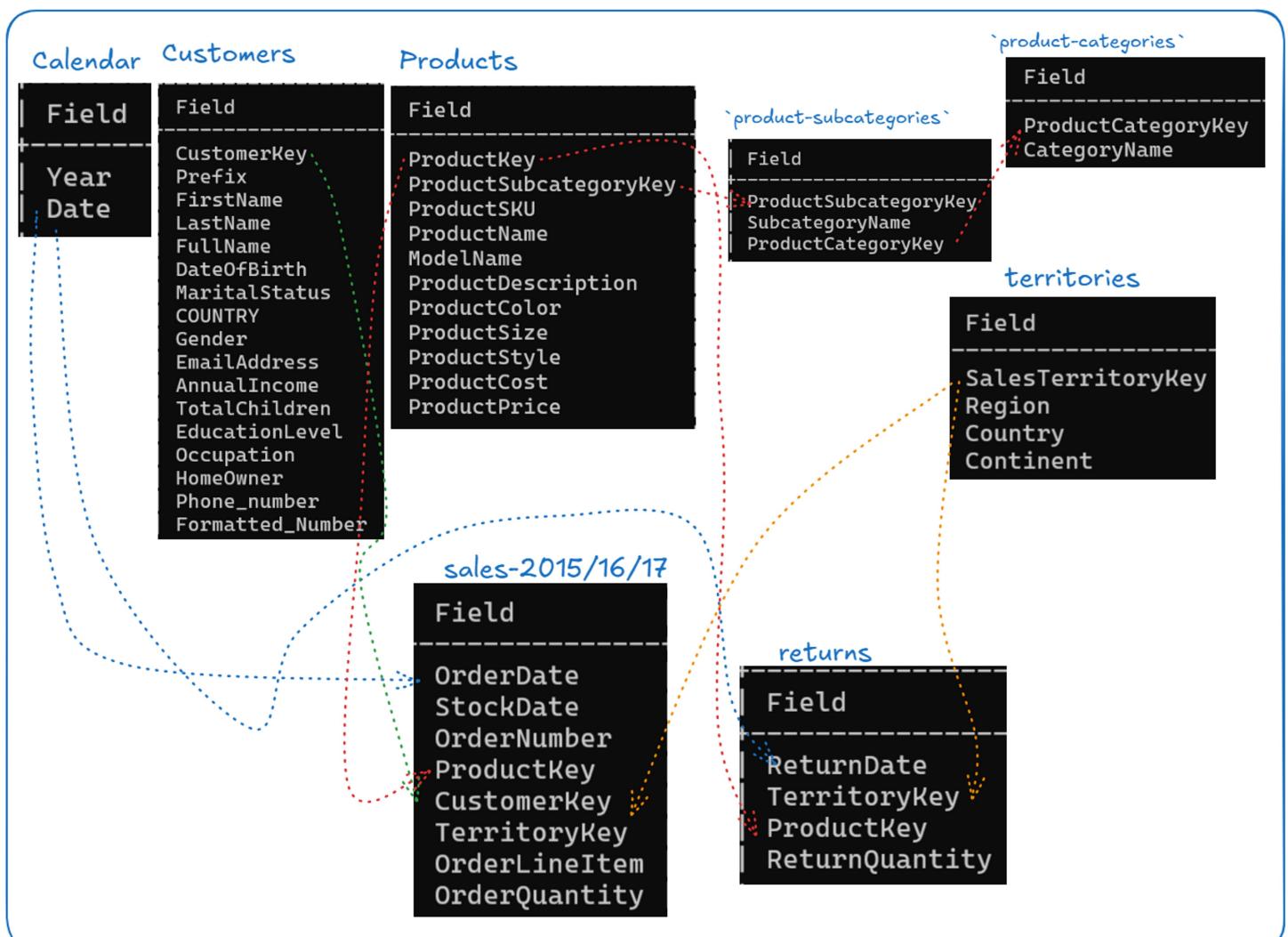
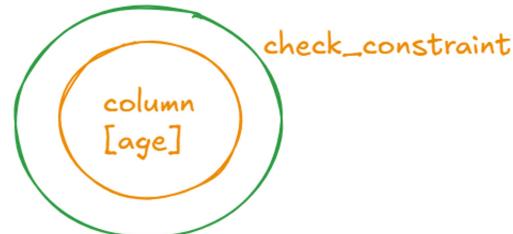


## Introduction To Joins - II



## Types of Relationship

1 : 1 [Person <-> Passport]

1 : M [Department -> Employees]

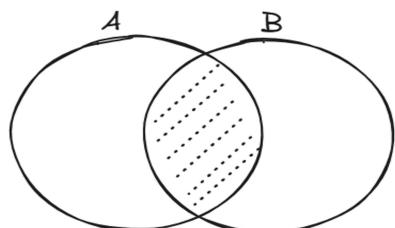
M : 1 [Student <- Course]

M : M [Student <-> Course]

## What is JOIN?

[Merge Queries] → Expanding the columns in horizontal direction.

→ Temporary Table to do the analysis while joining multiple columns from different tables using primary key & Foreign Key via "ON" Constraint.



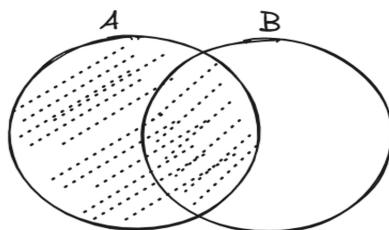
INNER JOIN

A intersection B.

SETA = [1,2,3,4,5,6,7,8,9,10]

SETB = [2,4,6,7,9,12,14,16,17,19,21]

SetA intersect SetB [2,4,6,7,9]



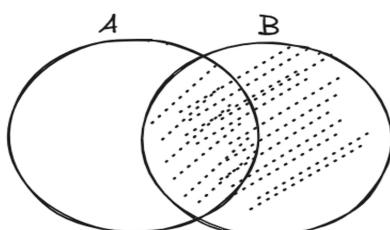
LEFT JOIN

A - B.

SETA = [1,2,3,4,5,6,7,8,9,10]

SETB = [2,4,6,7,9,12,14,16,17,19,21]

SetA - SetB = [1,2,3,4,5,6,7,8,9,10]



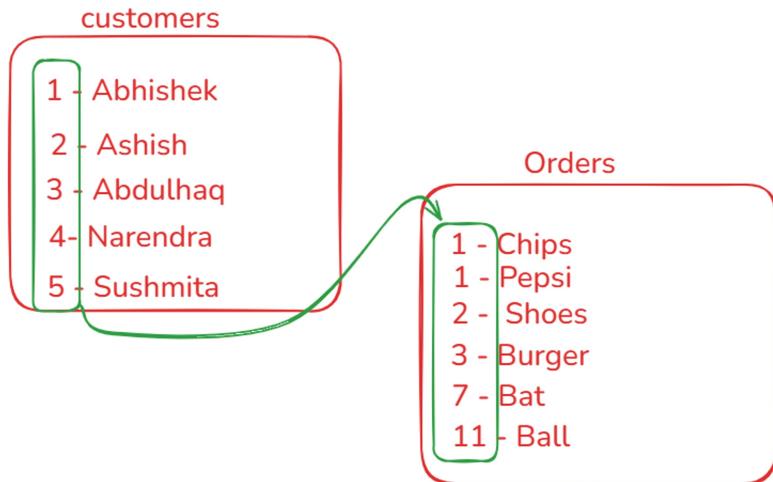
RIGHT JOIN

B - A

SETA = [1,2,3,4,5,6,7,8,9,10]

SETB = [2,4,6,7,9,12,14,16,17,19,21]

SetB - SetA = [2,4,6,7,9,12,14,16,17,19,21]



INNER JOIN -  
Abhishek - Chips, Pepsi  
Ashish - Shoes  
Abdulhaq - Burger

LEFT JOIN -  
Abhishek - Chips, Pepsi  
Ashish - Shoes  
Abdulhaq - Burger  
Narendra - NULL  
Sushmita - NULL

RIGHT JOIN -  
Abhishek - Chips, Pepsi  
Ashish - Shoes  
Abdulhaq - Burger  
NULL - Bat  
NULL - Ball

```
/* SELECT column1 , column2 .....
   FROM Table1
   [INNER/LEFT/RIGHT] JOIN Table2
   ON table1.key = table2.key
*/
```

/\* DESC territories;

Field	Type	Null	Key	Default	Extra
SalesTerritoryKey	int	YES		NULL	
Region	text	YES		NULL	
Country	text	YES		NULL	
Continent	text	YES		NULL	

Field	Type	Null	Key	Default	Extra
ReturnDate	text	YES		NULL	
TerritoryKey	int	YES		NULL	
ProductKey	int	YES		NULL	
ReturnQuantity	int	YES		NULL	

```

mysql> SELECT
->   Country,                               TotalReturns
->   SUM(ReturnQuantity) AS TotalRetruns
->   FROM territories
->   INNER JOIN returns
->   ON territories.SalesTerritoryKey = returns.TerritoryKey
->   GROUP BY Country
->   ORDER BY 2 DESC;
+-----+-----+
| Country | TotalRetruns |
+-----+-----+
| United States | 633 |
| Australia | 404 |
| Canada | 238 |
| United Kingdom | 204 |
| France | 186 |
| Germany | 163 |
+-----+-----+
6 rows in set (0.01 sec)

```

```

mysql> SELECT * FROM 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 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

```

```

mysql> SELECT
->   Region,                               TotalReturns
->   SUM(ReturnQuantity) AS TotalRetruns
->   FROM territories
->   INNER JOIN returns
->   ON territories.SalesTerritoryKey = returns.TerritoryKey
->   GROUP BY Region
->   ORDER BY 2 DESC;
+-----+-----+
| Region | TotalRetruns |
+-----+-----+
| Australia | 404 |
| Southwest | 362 |
| Northwest | 270 |
| Canada | 238 |
| United Kingdom | 204 |
| France | 186 |
| Germany | 163 |
| Southeast | 1 |
+-----+-----+
8 rows in set (0.01 sec)

```

```

mysql> SELECT
    -> Region,                               TotalReturns
    -> SUM(ReturnQuantity) AS TotalRetruns
    -> FROM territories
    -> LEFT JOIN returns
    -> ON territories.SalesTerritoryKey = returns.TerritoryKey
    -> GROUP BY Region
    -> ORDER BY 2 DESC;
+-----+-----+
| Region | TotalRetruns |
+-----+-----+
| Australia | 404 |
| Southwest | 362 |
| Northwest | 270 |
| Canada | 238 |
| United Kingdom | 204 |
| France | 186 |
| Germany | 163 |
| Southeast | 1 |
| Northeast | NULL |
| Central | NULL |
+-----+-----+
10 rows in set (0.00 sec)

```

```

mysql> SELECT
    -> SUM(ReturnQuantity) AS TotalReturns,
    -> Region
    -> FROM returns
    -> RIGHT JOIN territories
    -> ON territories.SalesTerritoryKey = returns.TerritoryKey
    -> GROUP BY Region
    -> ORDER BY 1 DESC;
+-----+-----+
| TotalReturns | Region |
+-----+-----+
| 404 | Australia |
| 362 | Southwest |
| 270 | Northwest |
| 238 | Canada |
| 204 | United Kingdom |
| 186 | France |
| 163 | Germany |
| 1 | Southeast |
| NULL | Northeast |
| NULL | Central |
+-----+-----+
10 rows in set (0.01 sec)

```

## # Quick Challenge [Multiple Table Joins + Use of Aliases]

Category Name	Total Returns

```

mysql> SELECT
    --> CategoryName,
    --> SUM(ReturnQuantity) AS TotalReturns
    --> FROM `product-categories` pc
    --> JOIN `product-subcategories` ps
    --> ON pc.ProductCategoryKey = ps.ProductCategoryKey
    --> JOIN Products p
    --> ON ps.ProductSubcategoryKey = p.ProductSubcategoryKey
    --> JOIN Returns r
    --> ON p.ProductKey = r.ProductKey
    --> GROUP BY 1
    --> ORDER BY 2 DESC;
+-----+-----+
| CategoryName | TotalReturns |
+-----+-----+
| Accessories   |      1130 |
| Bikes         |      429  |
| Clothing       |      269  |
+-----+-----+
3 rows in set (0.03 sec)

```

```

mysql> SELECT
    --> CategoryName,
    --> SUM(ReturnQuantity) AS TotalReturns
    --> FROM `product-categories`
    --> JOIN `product-subcategories`
    --> ON `product-categories`.ProductCategoryKey = `product-subcategories`.ProductCategoryKey
    --> JOIN Products
    --> ON `product-subcategories`.ProductSubcategoryKey = Products.ProductSubcategoryKey
    --> JOIN Returns
    --> ON Products.ProductKey = Returns.ProductKey
    --> GROUP BY 1
    --> ORDER BY 2 DESC;
+-----+-----+
| CategoryName | TotalReturns |
+-----+-----+
| Accessories   |      1130 |
| Bikes         |      429  |
| Clothing       |      269  |
+-----+-----+
3 rows in set (0.01 sec)

```

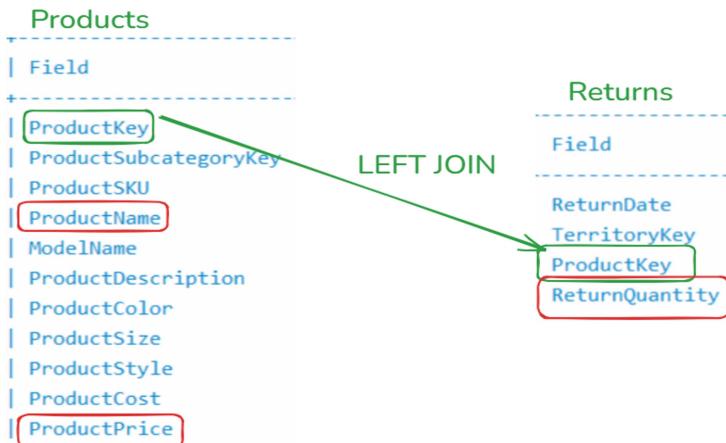
```

mysql> SELECT
    --> CategoryName,
    --> SUM(ReturnQuantity) AS TotalReturns
    --> FROM `product-categories` AS pc
    --> JOIN `product-subcategories` AS ps
    --> ON pc.ProductCategoryKey = ps.ProductCategoryKey
    --> JOIN Products AS p
    --> ON ps.ProductSubcategoryKey = p.ProductSubcategoryKey
    --> JOIN Returns AS r
    --> ON p.ProductKey = r.ProductKey
    --> GROUP BY 1
    --> ORDER BY 2 DESC;
+-----+-----+
| CategoryName | TotalReturns |
+-----+-----+
| Accessories   |      1130 |
| Bikes         |      429  |
| Clothing       |      269  |
+-----+-----+
3 rows in set (0.01 sec)

```

## Challenge - 1

List product names and prices for all products that have never been returned.



```

mysql> SELECT
-->     p.ProductKey,
-->     p.ProductName,
-->     p.ProductPrice,
-->     r.ReturnQuantity
--> FROM Products p
--> LEFT JOIN Returns r
--> ON p.ProductKey = r.ProductKey
--> WHERE r.ProductKey IS NULL;
+-----+-----+-----+-----+
| ProductKey | ProductName | ProductPrice | ReturnQuantity |
+-----+-----+-----+-----+
| 218 | Mountain Bike Socks, M | 9.5 | NULL |
| 219 | Mountain Bike Socks, L | 9.5 | NULL |
| 238 | HL Road Frame - Red, 62 | 1263.4598 | NULL |
| 241 | HL Road Frame - Red, 44 | 1263.4598 | NULL |
| 244 | HL Road Frame - Red, 48 | 1263.4598 | NULL |
| 247 | HL Road Frame - Red, 52 | 1263.4598 | NULL |
| 250 | HL Road Frame - Red, 56 | 1263.4598 | NULL |
| 253 | LL Road Frame - Black, 58 | 297.6346 | NULL |
| 256 | LL Road Frame - Black, 60 | 297.6346 | NULL |
| 259 | LL Road Frame - Black, 62 | 297.6346 | NULL |
| 262 | LL Road Frame - Red, 44 | 306.5636 | NULL |
| 264 | LL Road Frame - Red, 48 | 306.5636 | NULL |
| 266 | LL Road Frame - Red, 52 | 306.5636 | NULL |
+-----+-----+-----+-----+

```

-- Challenge 1

```

SELECT
    p.ProductKey,
    p.ProductName,
    p.ProductPrice,
    r.ReturnQuantity
FROM Products p
LEFT JOIN Returns r
ON p.ProductKey = r.ProductKey
WHERE r.ProductKey IS NULL;

```

## Challenge - 2

List products that were sold in 2015, 2016, and 2017.  
Ensure the product appears in all 3 sales tables.

```
mysql> SELECT
-> DISTINCT p.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`);
```

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

16 rows in set (0.06 sec)

Short Circuit

T || F || T || F || T || F || F || F || F

F && ( F || T || F || T || F || F || F || F )

ProductKey  
[293]

→ [sales-2015 ]

```
mysql> SELECT COUNT(DISTINCT ProductKey) ProductKey FROM `sales-2015`;
```

ProductKey
44

```
mysql> SELECT COUNT(DISTINCT ProductKey) ProductKey FROM `sales-2016`;
```

ProductKey
117

1 row in set (0.01 sec)

```
mysql> SELECT COUNT(DISTINCT ProductKey) ProductKey FROM `sales-2017`;
```

ProductKey
102

1 row in set (0.02 sec)