CN-SQL-Lec12-DATE Manipulation & Window Functions

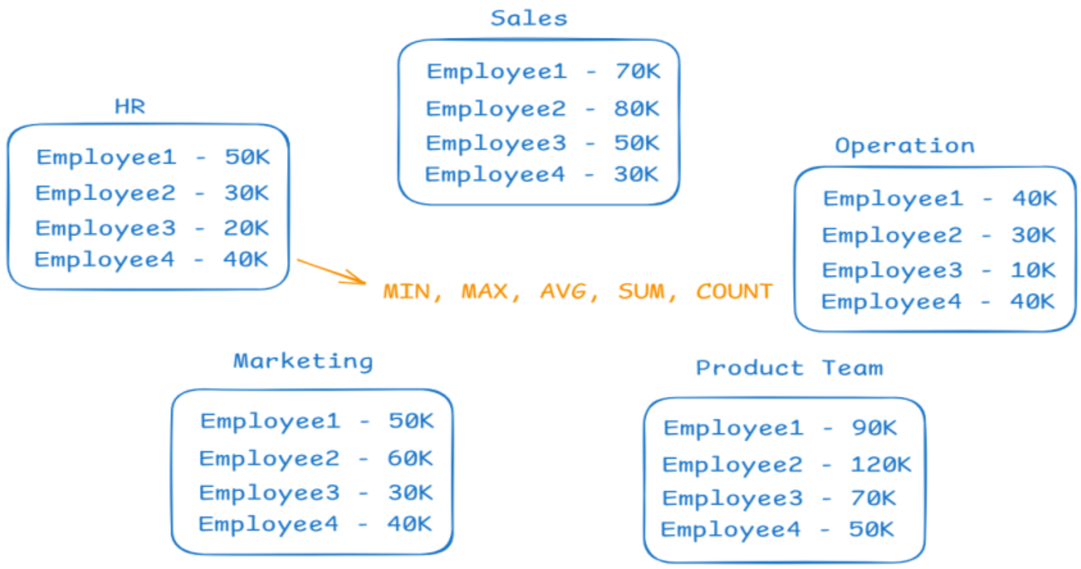
DATE Manipulation & Window Functions

🎯 Session Objectives:

- ✅ Understand the DATE data type in SQL (MySQL).
- ✅ Use SQL date functions to retrieve, analyse, and clean data.
- ✅ Apply date arithmetic and format transformations.
- ✅ Clean messy date columns for analysis.
- ✅ Solve real-world queries using date manipulation.
- ✅ Understand what window functions are and when to use them.
- ✅ Break down and apply the syntax of common window functions like ROW_NUMBER(), SUM(), AVG(), etc.
- ✅ Differentiate window functions from regular aggregate functions.

⚙️ Syntax:

```
SELECT
  window_function(...) OVER (
    PARTITION BY column_name
    ORDER BY column_name
    ROWS/RANGE ...
  ) AS result_column
FROM table_name;
```



Sales
Employee1 - 70K
Employee2 - 80K
Employee3 - 50K
Employee4 - 30K

HR
Employee1 - 50K
Employee2 - 30K
Employee3 - 20K
Employee4 - 40K

MIN, MAX, AVG, SUM, COUNT

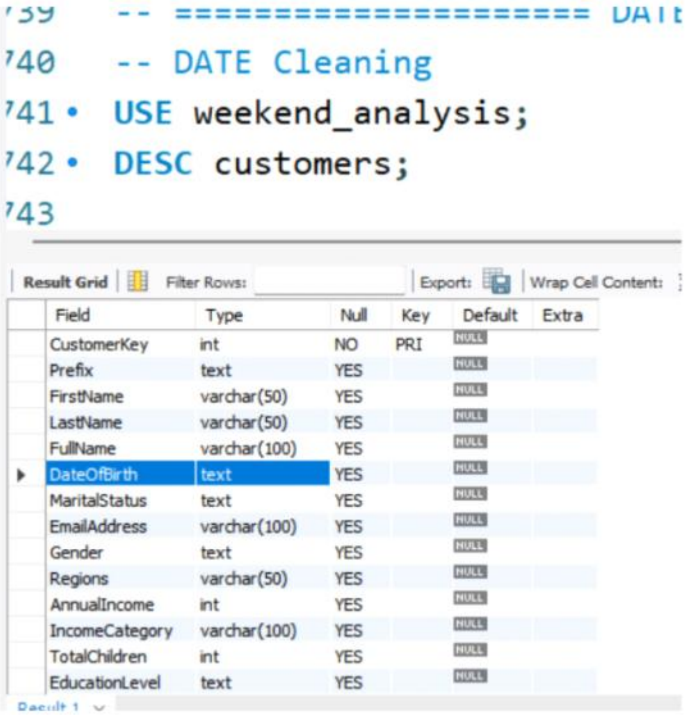
Operation
Employee1 - 40K
Employee2 - 30K
Employee3 - 10K
Employee4 - 40K

Marketing
Employee1 - 50K
Employee2 - 60K
Employee3 - 30K
Employee4 - 40K

Product Team
Employee1 - 90K
Employee2 - 120K
Employee3 - 70K
Employee4 - 50K

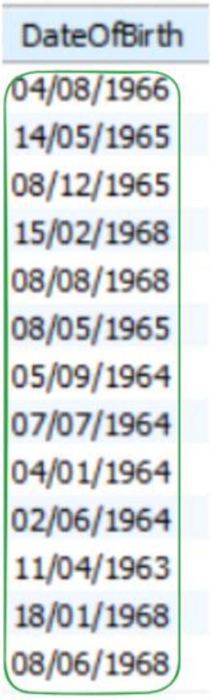
HR - 140K
Sales - 230K
Marketing - 180K
Operation - 120K
Product - 330K

With Group By & Aggregation



```
-- ===================== DATE
-- DATE Cleaning
USE weekend_analysis;
DESC customers;
```

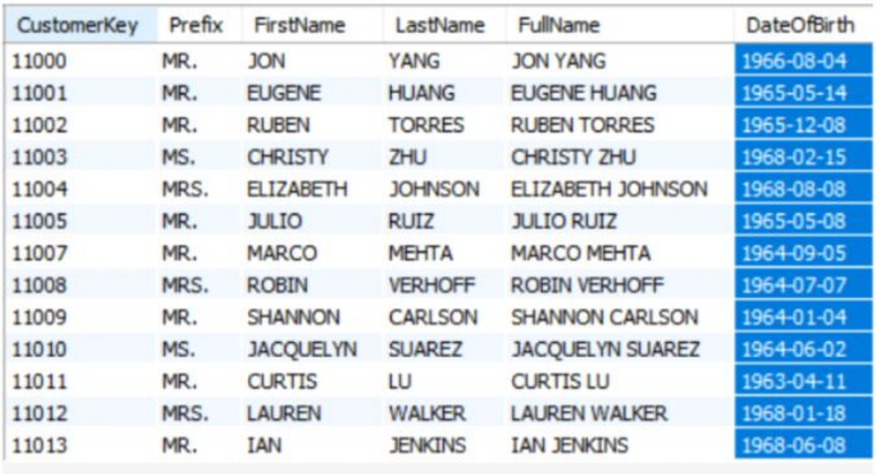
Field	Type	Null	Key	Default	Extra
CustomerKey	int	NO	PRI	NULL	
Prefix	text	YES		NULL	
FirstName	varchar(50)	YES		NULL	
LastName	varchar(50)	YES		NULL	
FullName	varchar(100)	YES		NULL	
DateOfBirth	text	YES		NULL	
MaritalStatus	text	YES		NULL	
EmailAddress	varchar(100)	YES		NULL	
Gender	text	YES		NULL	
Regions	varchar(50)	YES		NULL	
AnnualIncome	int	YES		NULL	
IncomeCategory	varchar(100)	YES		NULL	
TotalChildren	int	YES		NULL	
EducationLevel	text	YES		NULL	



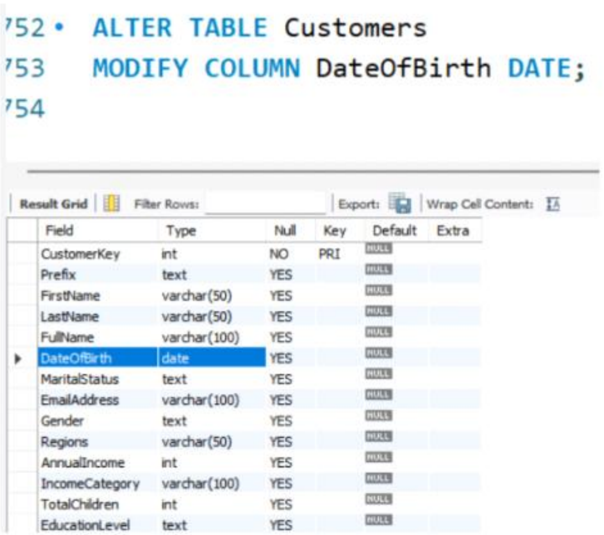
DateOfBirth
04/08/1966
14/05/1965
08/12/1965
15/02/1968
08/08/1968
08/05/1965
05/09/1964
07/07/1964
04/01/1964
02/06/1964
11/04/1963
18/01/1968
08/06/1968

dd/mm/yyyy -> "%d/%m/%Y"

```
-- DATE Cleaning
USE weekend_analysis;
DESC customers;

SELECT * FROM Customers;

SET SQL_SAFE_UPDATES = 0;
UPDATE Customers
SET DateofBirth = STR_TO_DATE(DateOfBirth , '%d/%m/%Y');
```



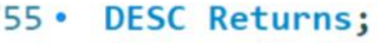
CustomerKey	Prefix	FirstName	LastName	FullName	DateOfBirth
11000	MR.	JON	YANG	JON YANG	1966-08-04
11001	MR.	EUGENE	HUANG	EUGENE HUANG	1965-05-14
11002	MR.	RUBEN	TORRES	RUBEN TORRES	1965-12-08
11003	MS.	CHRISTY	ZHU	CHRISTY ZHU	1968-02-15
11004	MRS.	ELIZABETH	JOHNSON	ELIZABETH JOHNSON	1968-08-08
11005	MR.	JULIO	RUIZ	JULIO RUIZ	1965-05-08
11007	MR.	MARCO	MEHTA	MARCO MEHTA	1964-09-05
11008	MRS.	ROBIN	VERHOFF	ROBIN VERHOFF	1964-07-07
11009	MR.	SHANNON	CARLSON	SHANNON CARLSON	1964-01-04
11010	MS.	JACQUELYN	SUAREZ	JACQUELYN SUAREZ	1964-06-02
11011	MR.	CURTIS	LU	CURTIS LU	1963-04-11
11012	MRS.	LAUREN	WALKER	LAUREN WALKER	1968-01-18
11013	MR.	IAN	JENKINS	IAN JENKINS	1968-06-08



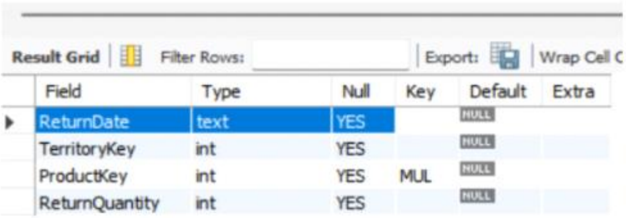
```sql
ALTER TABLE Customers
MODIFY COLUMN DateOfBirth DATE;
```

Field	Type	Null	Key	Default	Extra
CustomerKey	int	NO	PRI	NULL	
Prefix	text	YES		NULL	
FirstName	varchar(50)	YES		NULL	
LastName	varchar(50)	YES		NULL	
FullName	varchar(100)	YES		NULL	
DateOfBirth	date	YES		NULL	
MaritalStatus	text	YES		NULL	
EmailAddress	varchar(100)	YES		NULL	
Gender	text	YES		NULL	
Regions	varchar(50)	YES		NULL	
AnnualIncome	int	YES		NULL	
IncomeCategory	varchar(100)	YES		NULL	
TotalChildren	int	YES		NULL	
EducationLevel	text	YES		NULL	

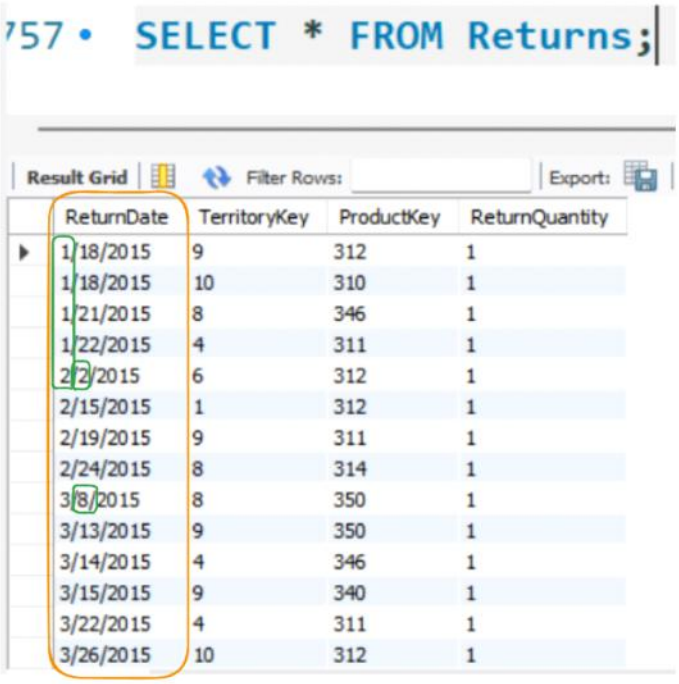
Returns



```sql
DESC Returns;
```



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

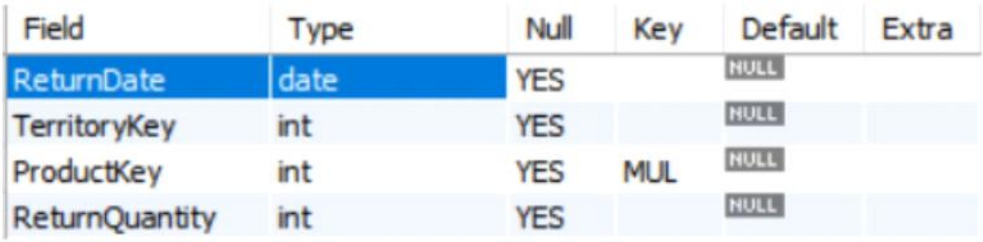


```sql
SELECT * FROM Returns;
```

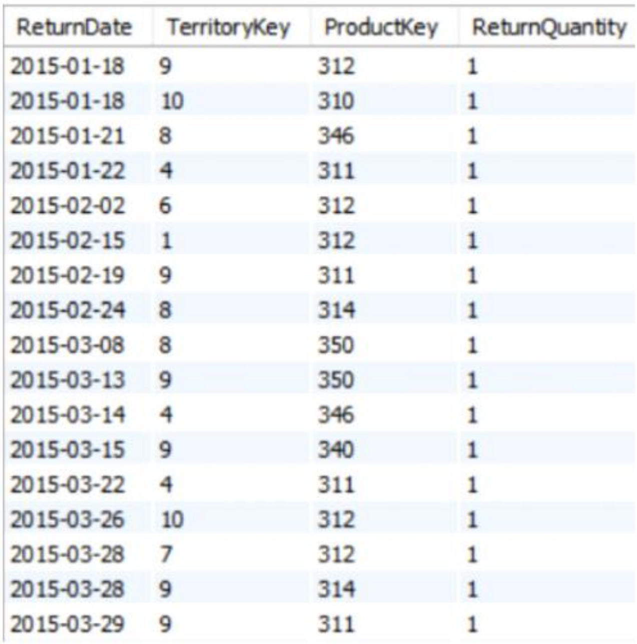
ReturnDate	TerritoryKey	ProductKey	ReturnQuantity
1/18/2015	9	312	1
1/18/2015	10	310	1
1/21/2015	8	346	1
1/22/2015	4	311	1
2/2/2015	6	312	1
2/15/2015	1	312	1
2/19/2015	9	311	1
2/24/2015	8	314	1
3/8/2015	8	350	1
3/13/2015	9	350	1
3/14/2015	4	346	1
3/15/2015	9	340	1
3/22/2015	4	311	1
3/26/2015	10	312	1

'm/d/yyyy'

```
-- Returns [Date Cleaning]
DESC Returns;

SELECT * FROM Returns;

UPDATE Returns
SET ReturnDate = STR_TO_DATE(ReturnDate , '%c/%e/%Y');

ALTER TABLE Returns
MODIFY COLUMN ReturnDate DATE;
```



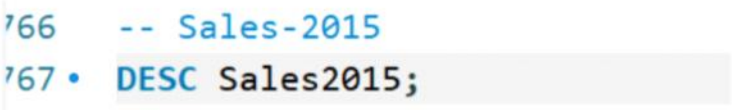
Field	Type	Null	Key	Default	Extra
ReturnDate	date	YES		NULL	
TerritoryKey	int	YES		NULL	
ProductKey	int	YES	MUL	NULL	
ReturnQuantity	int	YES		NULL	



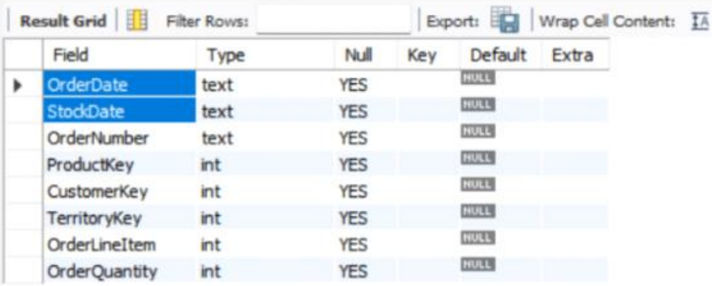
ReturnDate	TerritoryKey	ProductKey	ReturnQuantity
2015-01-18	9	312	1
2015-01-18	10	310	1
2015-01-21	8	346	1
2015-01-22	4	311	1
2015-02-02	6	312	1
2015-02-15	1	312	1
2015-02-19	9	311	1
2015-02-24	8	314	1
2015-03-08	8	350	1
2015-03-13	9	350	1
2015-03-14	4	346	1
2015-03-15	9	340	1
2015-03-22	4	311	1
2015-03-26	10	312	1
2015-03-28	7	312	1
2015-03-28	9	314	1
2015-03-29	9	311	1

YYYY-MM-DD

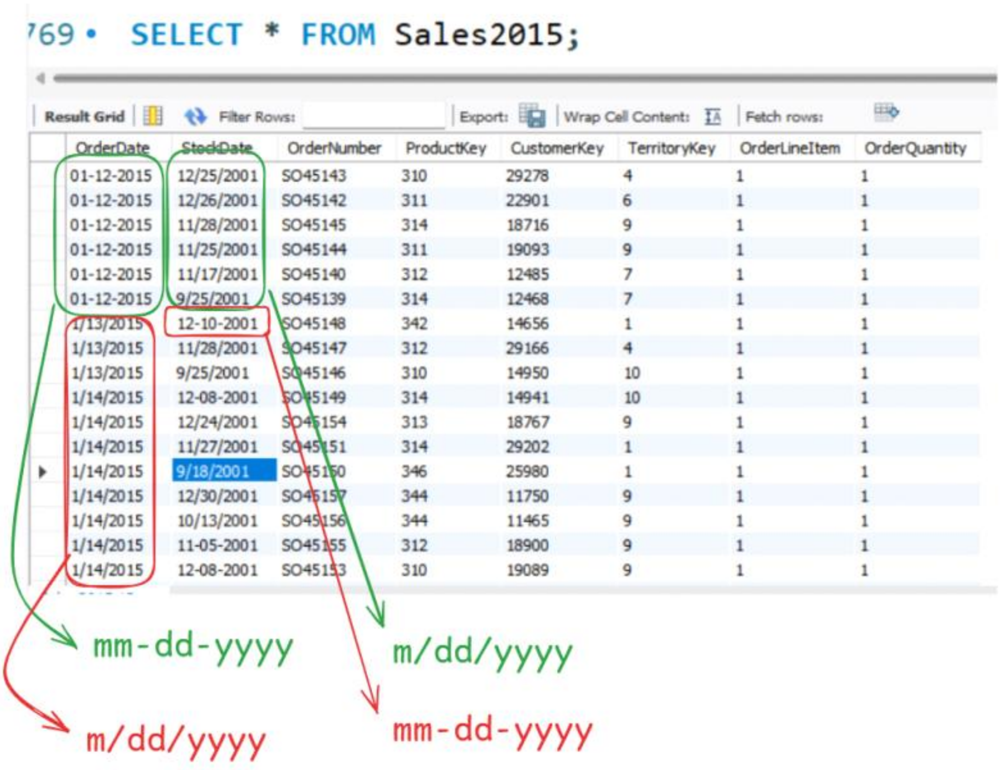
Sales2015



```
-- Sales-2015
DESC Sales2015;
```



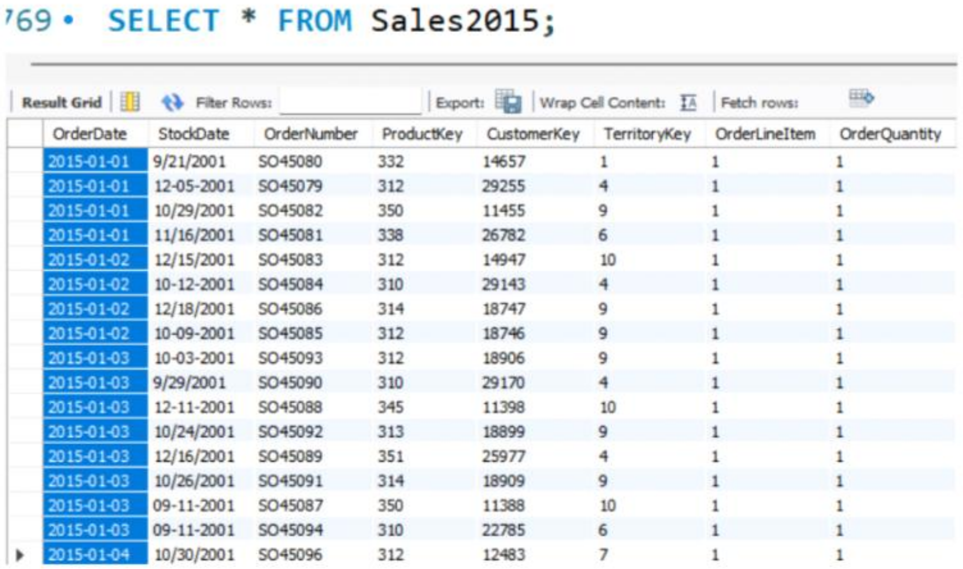
Field	Type	Null	Key	Default	Extra
OrderDate	text	YES		NULL	
StockDate	text	YES		NULL	
OrderNumber	text	YES		NULL	
ProductKey	int	YES		NULL	
CustomerKey	int	YES		NULL	
TerritoryKey	int	YES		NULL	
OrderLineItem	int	YES		NULL	
OrderQuantity	int	YES		NULL	



```sql
SELECT * FROM Sales2015;
```

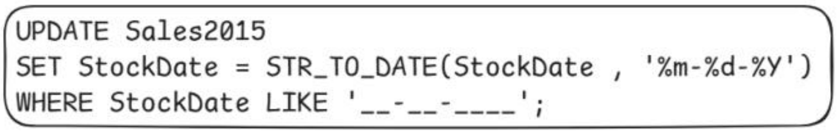
OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity
01-12-2015	12/25/2001	SO45143	310	29278	4	1	1
01-12-2015	12/26/2001	SO45142	311	22901	6	1	1
01-12-2015	11/28/2001	SO45145	314	18716	9	1	1
01-12-2015	11/25/2001	SO45144	311	19093	9	1	1
01-12-2015	11/17/2001	SO45140	312	12485	7	1	1
01-12-2015	9/25/2001	SO45139	314	12468	7	1	1
1/13/2015	12-10-2001	SO45148	342	14656	1	1	1
1/13/2015	11/28/2001	SO45147	312	29166	4	1	1
1/13/2015	9/25/2001	SO45146	310	14950	10	1	1
1/14/2015	12-08-2001	SO45149	314	14941	10	1	1
1/14/2015	12/24/2001	SO45154	313	18767	9	1	1
1/14/2015	11/27/2001	SO45151	314	29202	1	1	1
1/14/2015	9/18/2001	SO45150	346	25980	1	1	1
1/14/2015	12/30/2001	SO45157	344	11750	9	1	1
1/14/2015	10/13/2001	SO45156	344	11465	9	1	1
1/14/2015	11-05-2001	SO45155	312	18900	9	1	1
1/14/2015	12-08-2001	SO45153	310	19089	9	1	1

```sql
-- Sales-2015
DESC Sales2015;

SELECT * FROM Sales2015;

UPDATE Sales2015
SET OrderDate =
    CASE
        WHEN OrderDate Like '__-__-____' THEN STR_TO_DATE(OrderDate , '%m-%d-%Y')
        ELSE STR_TO_DATE(OrderDate , '%c/%d/%Y')
    END;
```

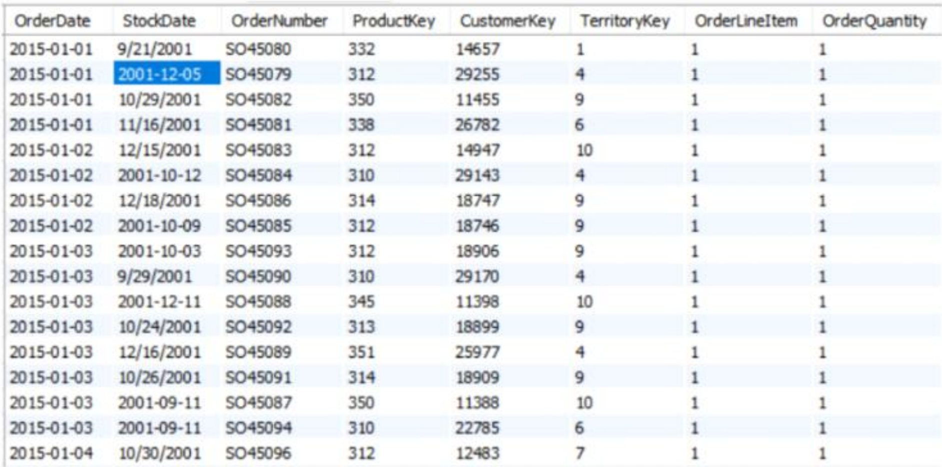


```sql
SELECT * FROM Sales2015;
```

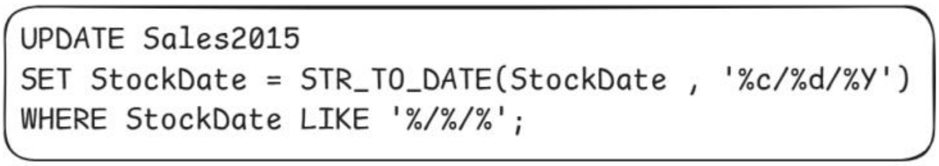
OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity
2015-01-01	9/21/2001	SO45080	332	14657	1	1	1
2015-01-01	12-05-2001	SO45079	312	29255	4	1	1
2015-01-01	10/29/2001	SO45082	350	11455	9	1	1
2015-01-01	11/16/2001	SO45081	338	26782	6	1	1
2015-01-02	12/15/2001	SO45083	312	14947	10	1	1
2015-01-02	10-12-2001	SO45084	310	29143	4	1	1
2015-01-02	12/18/2001	SO45086	314	18747	9	1	1
2015-01-02	10-09-2001	SO45085	312	18746	9	1	1
2015-01-03	10-03-2001	SO45093	312	18906	9	1	1
2015-01-03	9/29/2001	SO45090	310	29170	4	1	1
2015-01-03	12-11-2001	SO45088	345	11398	10	1	1
2015-01-03	10/24/2001	SO45092	313	18899	9	1	1
2015-01-03	12/16/2001	SO45089	351	25977	4	1	1
2015-01-03	10/26/2001	SO45091	314	18909	9	1	1
2015-01-03	09-11-2001	SO45087	350	11388	10	1	1
2015-01-03	09-11-2001	SO45094	310	22785	6	1	1
2015-01-04	10/30/2001	SO45096	312	12483	7	1	1



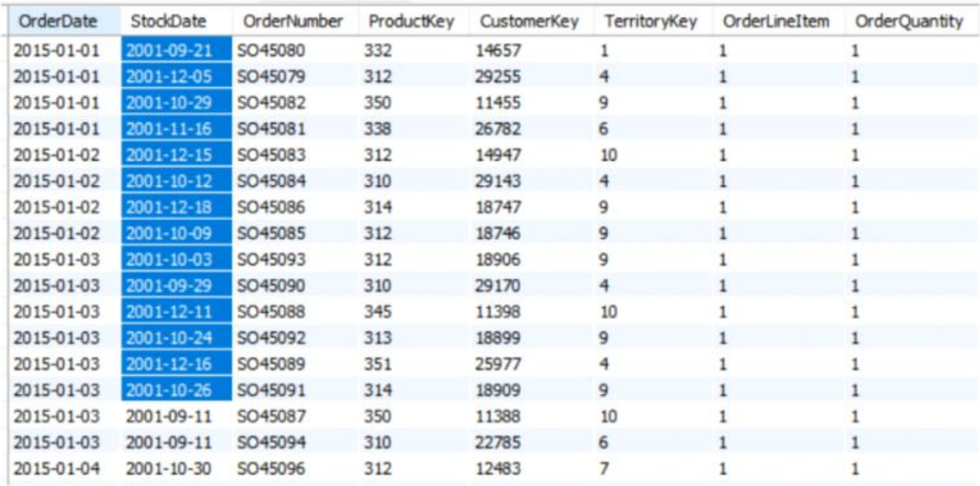
```
UPDATE Sales2015
SET StockDate = STR_TO_DATE(StockDate , '%m-%d-%Y')
WHERE StockDate LIKE '__-__-____';
```



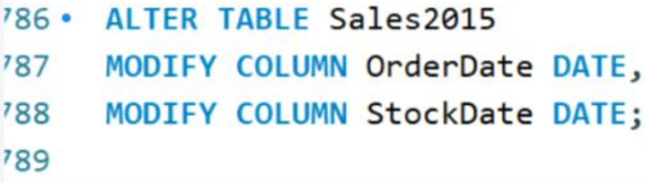
OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity
2015-01-01	9/21/2001	SO45080	332	14657	1	1	1
2015-01-01	2001-12-05	SO45079	312	29255	4	1	1
2015-01-01	10/29/2001	SO45082	350	11455	9	1	1
2015-01-01	11/16/2001	SO45081	338	26782	6	1	1
2015-01-02	12/15/2001	SO45083	312	14947	10	1	1
2015-01-02	2001-10-12	SO45084	310	29143	4	1	1
2015-01-02	12/18/2001	SO45086	314	18747	9	1	1
2015-01-02	2001-10-09	SO45085	312	18746	9	1	1
2015-01-03	2001-10-03	SO45093	312	18906	9	1	1
2015-01-03	9/29/2001	SO45090	310	29170	4	1	1
2015-01-03	2001-12-11	SO45088	345	11398	10	1	1
2015-01-03	10/24/2001	SO45092	313	18899	9	1	1
2015-01-03	12/16/2001	SO45089	351	25977	4	1	1
2015-01-03	10/26/2001	SO45091	314	18909	9	1	1
2015-01-03	2001-09-11	SO45087	350	11388	10	1	1
2015-01-03	2001-09-11	SO45094	310	22785	6	1	1
2015-01-04	10/30/2001	SO45096	312	12483	7	1	1



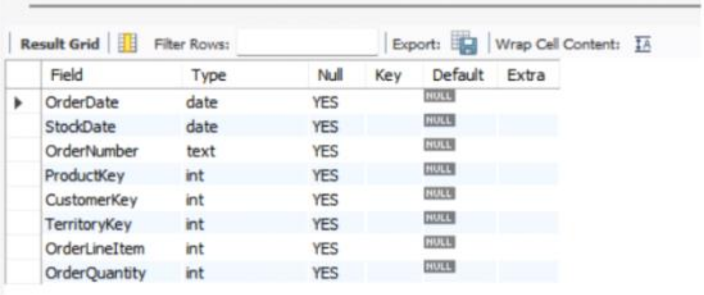
```
UPDATE Sales2015
SET StockDate = STR_TO_DATE(StockDate , '%c/%d/%Y')
WHERE StockDate LIKE '%/%/%';
```



OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity
2015-01-01	2001-09-21	SO45080	332	14657	1	1	1
2015-01-01	2001-12-05	SO45079	312	29255	4	1	1
2015-01-01	2001-10-29	SO45082	350	11455	9	1	1
2015-01-01	2001-11-16	SO45081	338	26782	6	1	1
2015-01-02	2001-12-15	SO45083	312	14947	10	1	1
2015-01-02	2001-10-12	SO45084	310	29143	4	1	1
2015-01-02	2001-12-18	SO45086	314	18747	9	1	1
2015-01-02	2001-10-09	SO45085	312	18746	9	1	1
2015-01-03	2001-10-03	SO45093	312	18906	9	1	1
2015-01-03	2001-09-29	SO45090	310	29170	4	1	1
2015-01-03	2001-12-11	SO45088	345	11398	10	1	1
2015-01-03	2001-10-24	SO45092	313	18899	9	1	1
2015-01-03	2001-12-16	SO45089	351	25977	4	1	1
2015-01-03	2001-10-26	SO45091	314	18909	9	1	1
2015-01-03	2001-09-11	SO45087	350	11388	10	1	1
2015-01-03	2001-09-11	SO45094	310	22785	6	1	1
2015-01-04	2001-10-30	SO45096	312	12483	7	1	1

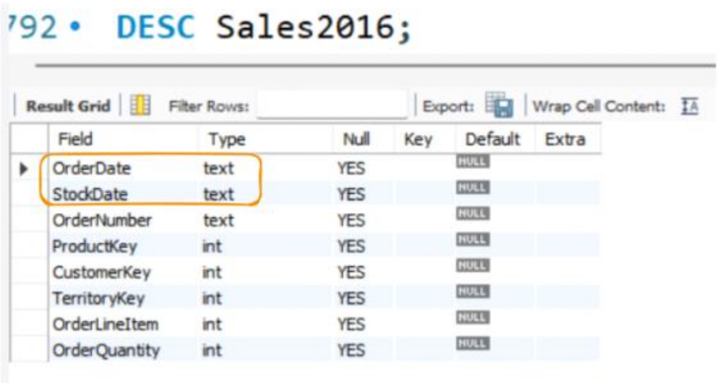


```
ALTER TABLE Sales2015
MODIFY COLUMN OrderDate DATE,
MODIFY COLUMN StockDate DATE;
```



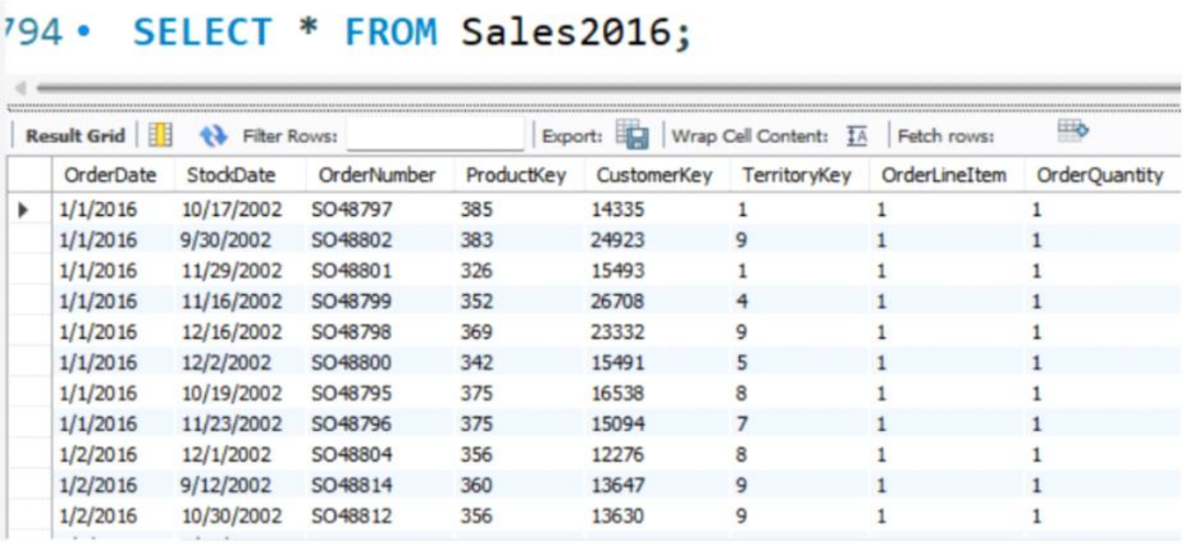
Field	Type	Null	Key	Default	Extra
OrderDate	date	YES		NULL	
StockDate	date	YES		NULL	
OrderNumber	text	YES		NULL	
ProductKey	int	YES		NULL	
CustomerKey	int	YES		NULL	
TerritoryKey	int	YES		NULL	
OrderLineItem	int	YES		NULL	
OrderQuantity	int	YES		NULL	

Sales2016



```
DESC Sales2016;
```

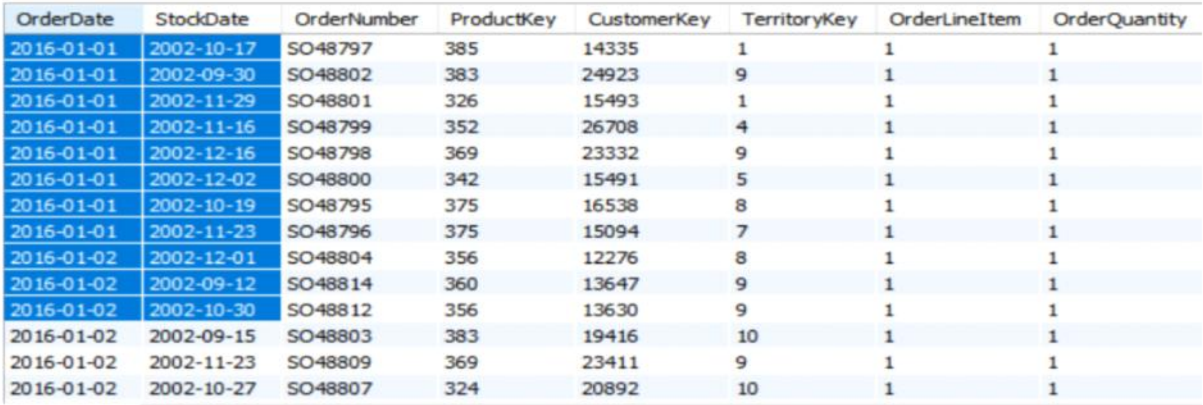
Field	Type	Null	Key	Default	Extra
OrderDate	text	YES		NULL	
StockDate	text	YES		NULL	
OrderNumber	text	YES		NULL	
ProductKey	int	YES		NULL	
CustomerKey	int	YES		NULL	
TerritoryKey	int	YES		NULL	
OrderLineItem	int	YES		NULL	
OrderQuantity	int	YES		NULL	



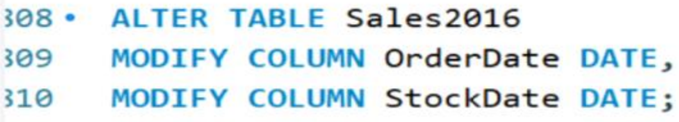
```
SELECT * FROM Sales2016;
```

OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity
1/1/2016	10/17/2002	SO48797	385	14335	1	1	1
1/1/2016	9/30/2002	SO48802	383	24923	9	1	1
1/1/2016	11/29/2002	SO48801	326	15493	1	1	1
1/1/2016	11/16/2002	SO48799	352	26708	4	1	1
1/1/2016	12/16/2002	SO48798	369	23332	9	1	1
1/1/2016	12/2/2002	SO48800	342	15491	5	1	1
1/1/2016	10/19/2002	SO48795	375	16538	8	1	1
1/1/2016	11/23/2002	SO48796	375	15094	7	1	1
1/2/2016	12/1/2002	SO48804	356	12276	8	1	1
1/2/2016	9/12/2002	SO48814	360	13647	9	1	1
1/2/2016	10/30/2002	SO48812	356	13630	9	1	1

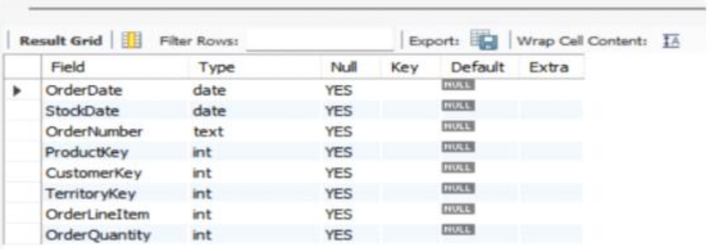
```
-- ======================= Sales2016  ==============================

DESC Sales2016;

SELECT * FROM Sales2016 WHERE OrderDate LIKE '%/%/%'; -- 23935 row(s) returned

SELECT * FROM Sales2016 WHERE StockDate LIKE '%/%/%'; -- 23935 row(s) returned

UPDATE Sales2016
SET OrderDate = STR_TO_DATE(OrderDate , '%c/%e/%Y')
WHERE OrderDate LIKE '%/%/%';

UPDATE Sales2016
SET StockDate = STR_TO_DATE(StockDate , '%c/%e/%Y')
WHERE StockDate LIKE '%/%/%';

SELECT * FROM Sales2016;
```



OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity
2016-01-01	2002-10-17	SO48797	385	14335	1	1	1
2016-01-01	2002-09-30	SO48802	383	24923	9	1	1
2016-01-01	2002-11-29	SO48801	326	15493	1	1	1
2016-01-01	2002-11-16	SO48799	352	26708	4	1	1
2016-01-01	2002-12-16	SO48798	369	23332	9	1	1
2016-01-01	2002-12-02	SO48800	342	15491	5	1	1
2016-01-01	2002-10-19	SO48795	375	16538	8	1	1
2016-01-01	2002-11-23	SO48796	375	15094	7	1	1
2016-01-02	2002-12-01	SO48804	356	12276	8	1	1
2016-01-02	2002-09-12	SO48814	360	13647	9	1	1
2016-01-02	2002-10-30	SO48812	356	13630	9	1	1
2016-01-02	2002-09-15	SO48803	383	19416	10	1	1
2016-01-02	2002-11-23	SO48809	369	23411	9	1	1
2016-01-02	2002-10-27	SO48807	324	20892	10	1	1

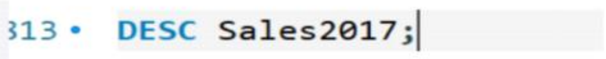


```
ALTER TABLE Sales2016
MODIFY COLUMN OrderDate DATE,
MODIFY COLUMN StockDate DATE;
```

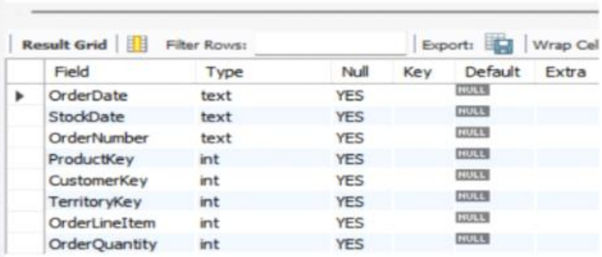


Field	Type	Null	Key	Default	Extra
OrderDate	date	YES		NULL	
StockDate	date	YES		NULL	
OrderNumber	text	YES		NULL	
ProductKey	int	YES		NULL	
CustomerKey	int	YES		NULL	
TerritoryKey	int	YES		NULL	
OrderLineItem	int	YES		NULL	
OrderQuantity	int	YES		NULL	

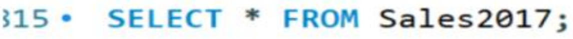
Sales2017



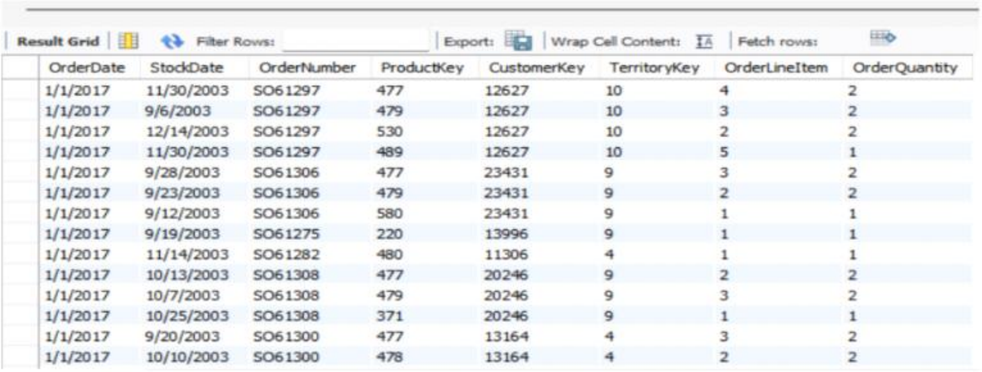
```
DESC Sales2017;
```



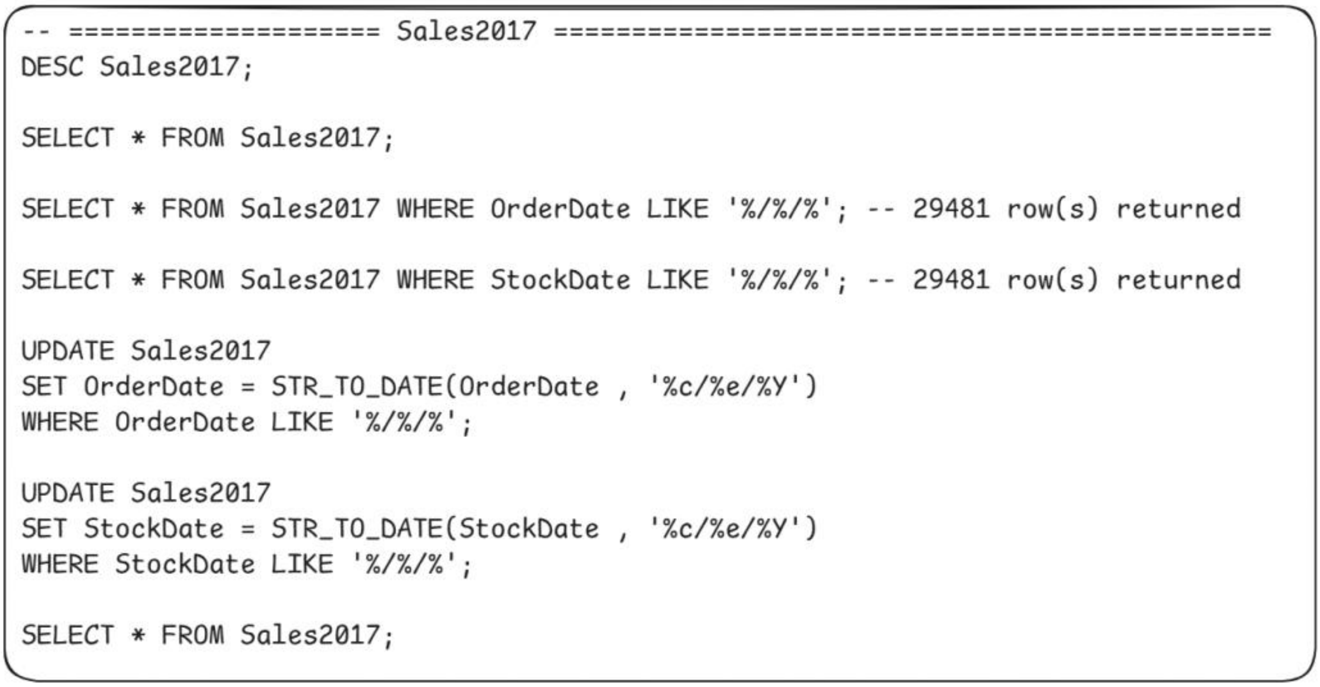
Field	Type	Null	Key	Default	Extra
OrderDate	text	YES		NULL	
StockDate	text	YES		NULL	
OrderNumber	text	YES		NULL	
ProductKey	int	YES		NULL	
CustomerKey	int	YES		NULL	
TerritoryKey	int	YES		NULL	
OrderLineItem	int	YES		NULL	
OrderQuantity	int	YES		NULL	



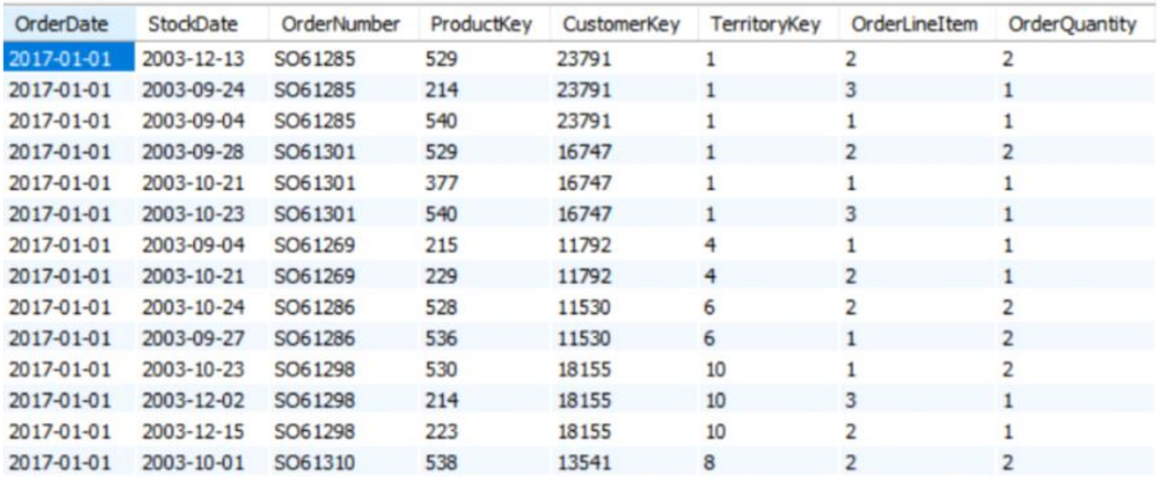
```
SELECT * FROM Sales2017;
```



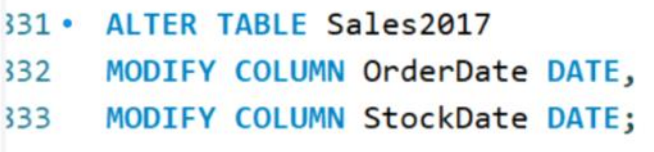
OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity
1/1/2017	11/30/2003	SO61297	477	12627	10	4	2
1/1/2017	9/6/2003	SO61297	479	12627	10	3	2
1/1/2017	12/14/2003	SO61297	530	12627	10	2	2
1/1/2017	11/30/2003	SO61297	489	12627	10	5	1
1/1/2017	9/28/2003	SO61306	477	23431	9	3	2
1/1/2017	9/23/2003	SO61306	479	23431	9	2	2
1/1/2017	9/12/2003	SO61306	580	23431	9	1	1
1/1/2017	9/19/2003	SO61275	220	13996	9	1	1
1/1/2017	11/14/2003	SO61282	480	11306	4	1	1
1/1/2017	10/13/2003	SO61308	477	20246	9	2	2
1/1/2017	10/7/2003	SO61308	479	20246	9	3	2
1/1/2017	10/25/2003	SO61308	371	20246	9	1	1
1/1/2017	9/20/2003	SO61300	477	13164	4	3	2
1/1/2017	10/10/2003	SO61300	478	13164	4	2	2



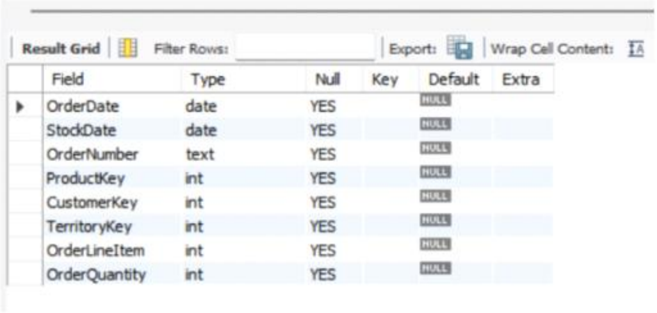
```
-- =================== Sales2017 ==============================================
DESC Sales2017;

SELECT * FROM Sales2017;

SELECT * FROM Sales2017 WHERE OrderDate LIKE '%/%/%'; -- 29481 row(s) returned

SELECT * FROM Sales2017 WHERE StockDate LIKE '%/%/%'; -- 29481 row(s) returned

UPDATE Sales2017
SET OrderDate = STR_TO_DATE(OrderDate , '%c/%e/%Y')
WHERE OrderDate LIKE '%/%/%';

UPDATE Sales2017
SET StockDate = STR_TO_DATE(StockDate , '%c/%e/%Y')
WHERE StockDate LIKE '%/%/%';

SELECT * FROM Sales2017;
```



OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity
2017-01-01	2003-12-13	SO61285	529	23791	1	2	2
2017-01-01	2003-09-24	SO61285	214	23791	1	3	1
2017-01-01	2003-09-04	SO61285	540	23791	1	1	1
2017-01-01	2003-09-28	SO61301	529	16747	1	2	2
2017-01-01	2003-10-21	SO61301	377	16747	1	1	1
2017-01-01	2003-10-23	SO61301	540	16747	1	3	1
2017-01-01	2003-09-04	SO61269	215	11792	4	1	1
2017-01-01	2003-10-21	SO61269	229	11792	4	2	1
2017-01-01	2003-10-24	SO61286	528	11530	6	2	2
2017-01-01	2003-09-27	SO61286	536	11530	6	1	2
2017-01-01	2003-10-23	SO61298	530	18155	10	1	2
2017-01-01	2003-12-02	SO61298	214	18155	10	3	1
2017-01-01	2003-12-15	SO61298	223	18155	10	2	1
2017-01-01	2003-10-01	SO61310	538	13541	8	2	2



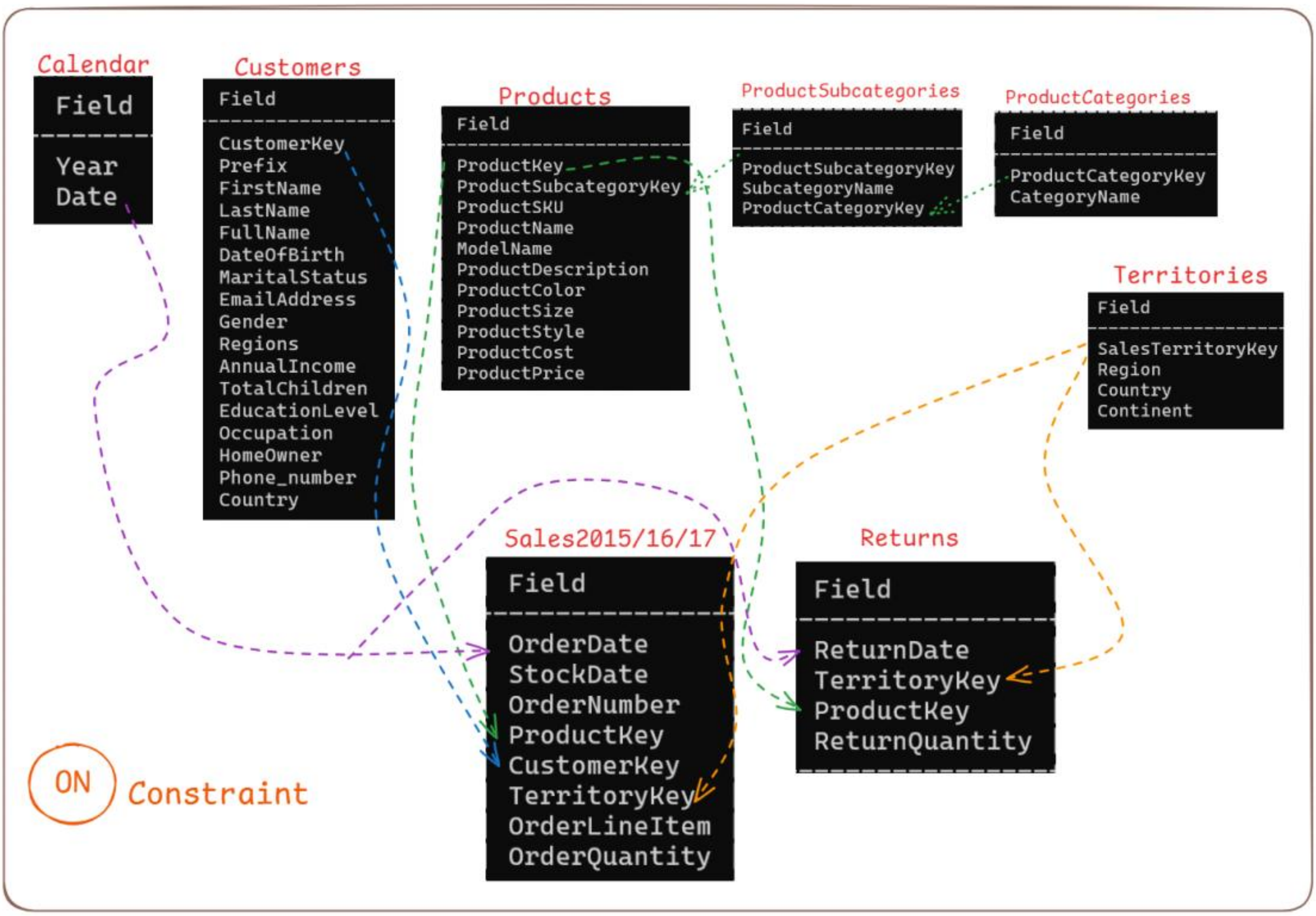
```
ALTER TABLE Sales2017
MODIFY COLUMN OrderDate DATE,
MODIFY COLUMN StockDate DATE;
```



Field	Type	Null	Key	Default	Extra
OrderDate	date	YES		NULL	
StockDate	date	YES		NULL	
OrderNumber	text	YES		NULL	
ProductKey	int	YES		NULL	
CustomerKey	int	YES		NULL	
TerritoryKey	int	YES		NULL	
OrderLineItem	int	YES		NULL	
OrderQuantity	int	YES		NULL	

Challenge-1

Total Return Quantity based on Category [Return Year == 2017]



Calendar

Field

Year
Date

Customers

Field

CustomerKey
Prefix
FirstName
LastName
FullName
DateOfBirth
MaritalStatus
EmailAddress
Gender
Regions
AnnualIncome
TotalChildren
EducationLevel
Occupation
HomeOwner
Phone_number
Country

Products

Field

ProductKey
ProductSubcategoryKey
ProductSKU
ProductName
ModelName
ProductDescription
ProductColor
ProductSize
ProductStyle
ProductCost
ProductPrice

ProductSubcategories

Field

ProductSubcategoryKey
SubcategoryName
ProductCategoryKey

ProductCategories

Field

ProductCategoryKey
CategoryName

Territories

Field

SalesTerritoryKey
Region
Country
Continent

Sales2015/16/17

Field

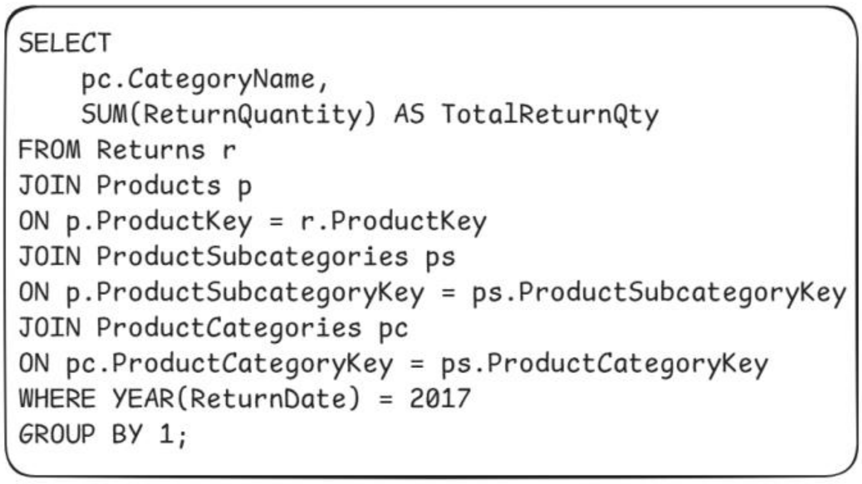
OrderDate
StockDate
OrderNumber
ProductKey
CustomerKey
TerritoryKey
OrderLineItem
OrderQuantity

Returns

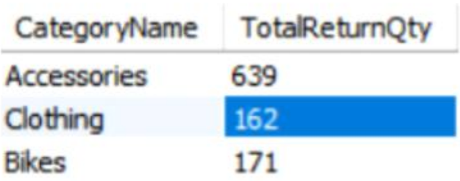
Field

ReturnDate
TerritoryKey
ProductKey
ReturnQuantity

ON Constraint



```
SELECT
    pc.CategoryName,
    SUM(ReturnQuantity) AS TotalReturnQty
FROM Returns r
JOIN Products p
ON p.ProductKey = r.ProductKey
JOIN ProductSubcategories ps
ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
JOIN ProductCategories pc
ON pc.ProductCategoryKey = ps.ProductCategoryKey
WHERE YEAR(ReturnDate) = 2017
GROUP BY 1;
```



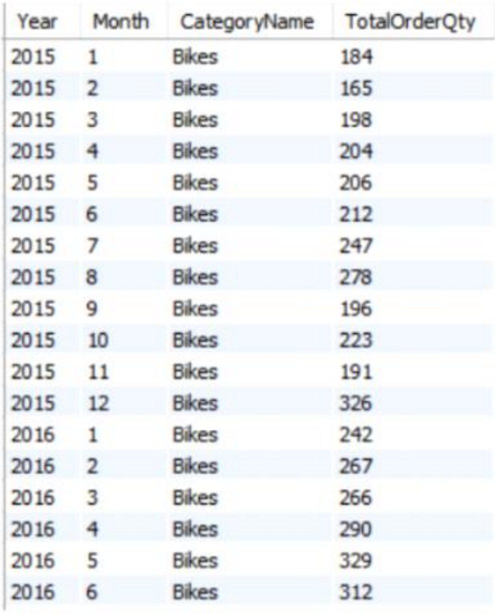
CategoryName	TotalReturnQty
Accessories	639
Clothing	162
Bikes	171

Challenge 2 : Calculate the Sum of Return Qty per Year.

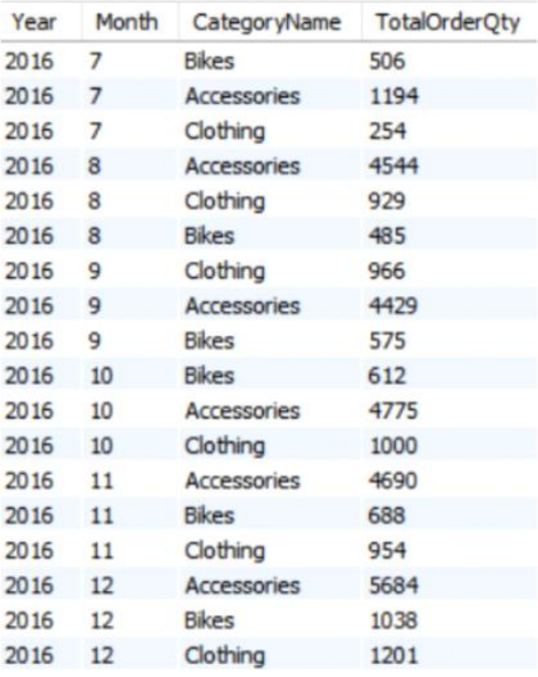
```
SELECT
    YEAR(ReturnDate) AS Year,
    SUM(ReturnQuantity) AS TotalReturnQty
FROM Returns
GROUP BY 1;
```

Challenge 3 : Seasonal Sales Trends [Year/Month Wise for each Category] for Sales2015/16/17

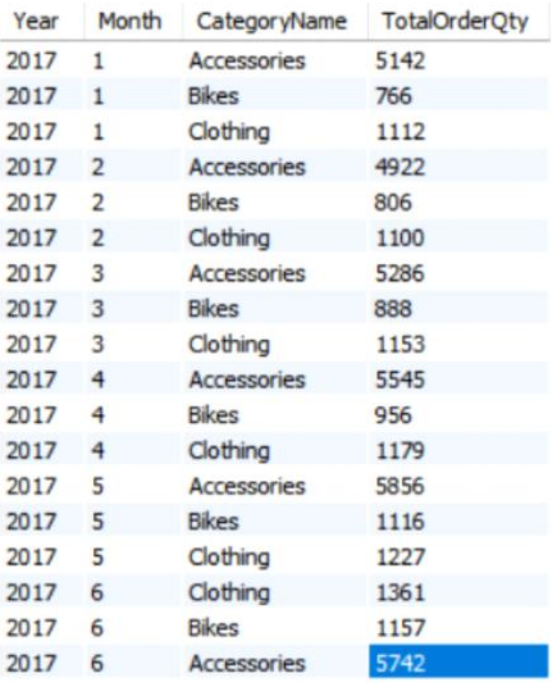
```
WITH AllSales AS (
    SELECT * FROM Sales2015
    UNION
    SELECT * FROM Sales2016
    UNION
    SELECT * FROM Sales2017
)
SELECT
    Year(s.OrderDate) AS Year,
    Month(s.OrderDate) AS Month,
    pc.CategoryName,
    SUM(s.OrderQuantity) AS TotalOrderQty
FROM AllSales s
JOIN Products p
ON p.ProductKey = s.ProductKey
JOIN ProductSubcategories ps
ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
JOIN ProductCategories pc
ON pc.ProductCategoryKey = ps.ProductCategoryKey
GROUP BY 1,2,3;
```



Year	Month	CategoryName	TotalOrderQty
2015	1	Bikes	184
2015	2	Bikes	165
2015	3	Bikes	198
2015	4	Bikes	204
2015	5	Bikes	206
2015	6	Bikes	212
2015	7	Bikes	247
2015	8	Bikes	278
2015	9	Bikes	196
2015	10	Bikes	223
2015	11	Bikes	191
2015	12	Bikes	326
2016	1	Bikes	242
2016	2	Bikes	267
2016	3	Bikes	266
2016	4	Bikes	290
2016	5	Bikes	329
2016	6	Bikes	312



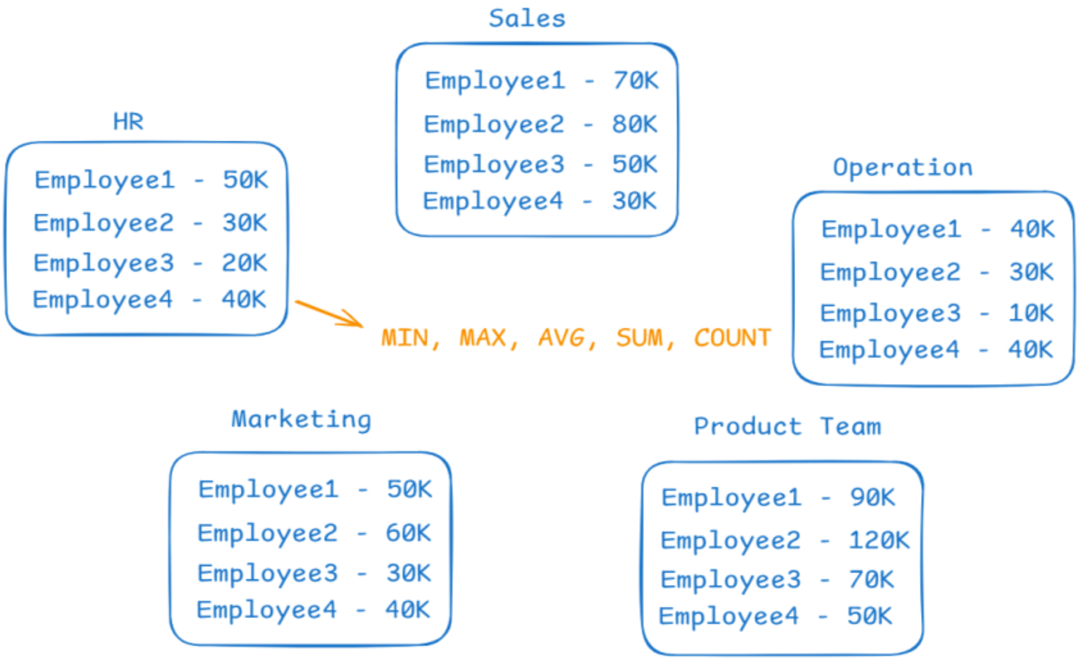
Year	Month	CategoryName	TotalOrderQty
2016	7	Bikes	506
2016	7	Accessories	1194
2016	7	Clothing	254
2016	8	Accessories	4544
2016	8	Clothing	929
2016	8	Bikes	485
2016	9	Clothing	966
2016	9	Accessories	4429
2016	9	Bikes	575
2016	10	Bikes	612
2016	10	Accessories	4775
2016	10	Clothing	1000
2016	11	Accessories	4690
2016	11	Bikes	688
2016	11	Clothing	954
2016	12	Accessories	5684
2016	12	Bikes	1038
2016	12	Clothing	1201



Year	Month	CategoryName	TotalOrderQty
2017	1	Accessories	5142
2017	1	Bikes	766
2017	1	Clothing	1112
2017	2	Accessories	4922
2017	2	Bikes	806
2017	2	Clothing	1100
2017	3	Accessories	5286
2017	3	Bikes	888
2017	3	Clothing	1153
2017	4	Accessories	5545
2017	4	Bikes	956
2017	4	Clothing	1179
2017	5	Accessories	5856
2017	5	Bikes	1116
2017	5	Clothing	1227
2017	6	Clothing	1361
2017	6	Bikes	1157
2017	6	Accessories	5742

Syntax:

```
SELECT
  window_function(...) OVER (
    PARTITION BY column_name
    ORDER BY column_name
    ROWS/RANGE ...
  ) AS result_column
FROM table_name;
```



Sales

Employee1 - 70K
Employee2 - 80K
Employee3 - 50K
Employee4 - 30K

HR

Employee1 - 50K
Employee2 - 30K
Employee3 - 20K
Employee4 - 40K

MIN, MAX, AVG, SUM, COUNT

Operation

Employee1 - 40K
Employee2 - 30K
Employee3 - 10K
Employee4 - 40K

Marketing

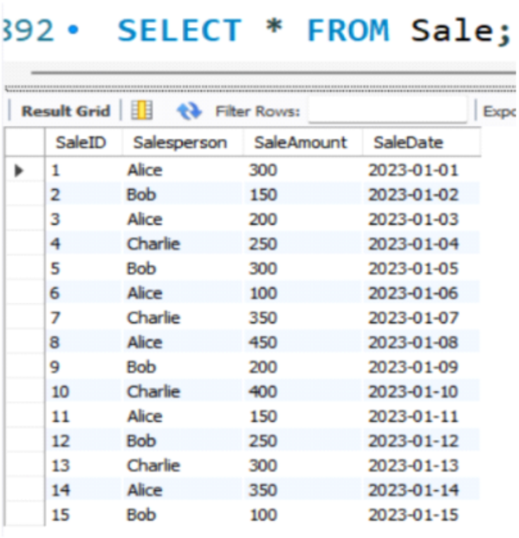
Employee1 - 50K
Employee2 - 60K
Employee3 - 30K
Employee4 - 40K

Product Team

Employee1 - 90K
Employee2 - 120K
Employee3 - 70K
Employee4 - 50K

HR - 140K
Sales - 230K
Marketing - 180K
Operation - 120K
Product - 330K

With Group By & Aggregation

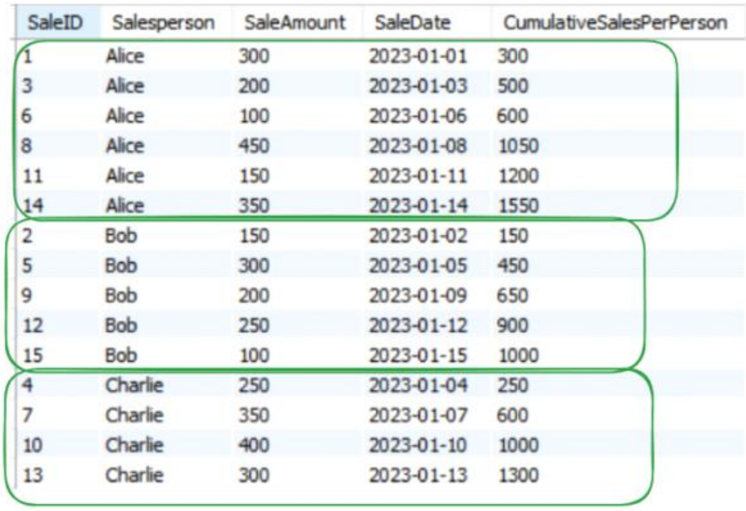


```
392 •  SELECT * FROM Sale;
```

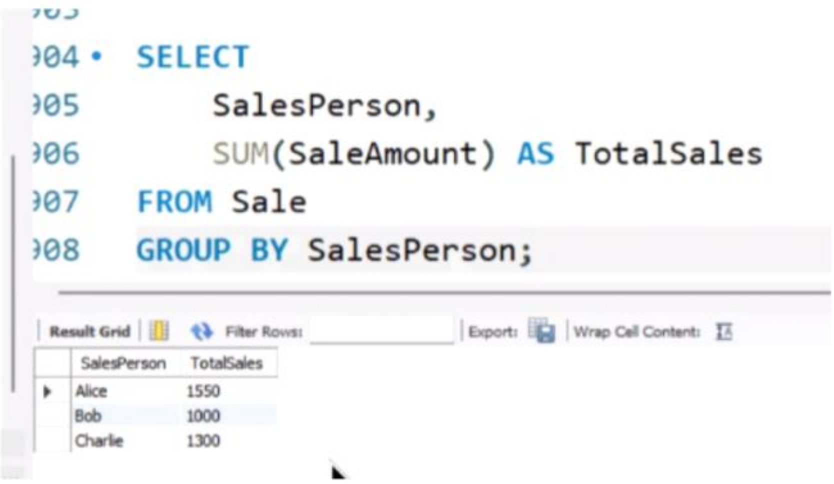
SaleID	Salesperson	SaleAmount	SaleDate
1	Alice	300	2023-01-01
2	Bob	150	2023-01-02
3	Alice	200	2023-01-03
4	Charlie	250	2023-01-04
5	Bob	300	2023-01-05
6	Alice	100	2023-01-06
7	Charlie	350	2023-01-07
8	Alice	450	2023-01-08
9	Bob	200	2023-01-09
10	Charlie	400	2023-01-10
11	Alice	150	2023-01-11
12	Bob	250	2023-01-12
13	Charlie	300	2023-01-13
14	Alice	350	2023-01-14
15	Bob	100	2023-01-15

Challenge 1 : Find the Cumulative Total Sum by SalesPerson

```
SELECT
    *,
    SUM(SaleAmount) OVER(
        PARTITION BY SalesPerson
        ORDER BY SaleDate
    ) AS CumulativeSalesPerPerson
FROM Sale;
```



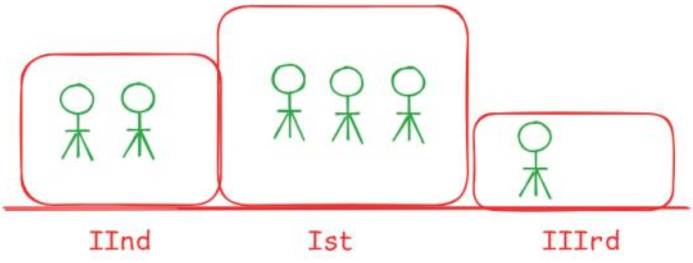
SaleID	Salesperson	SaleAmount	SaleDate	CumulativeSalesPerPerson
1	Alice	300	2023-01-01	300
3	Alice	200	2023-01-03	500
6	Alice	100	2023-01-06	600
8	Alice	450	2023-01-08	1050
11	Alice	150	2023-01-11	1200
14	Alice	350	2023-01-14	1550
2	Bob	150	2023-01-02	150
5	Bob	300	2023-01-05	450
9	Bob	200	2023-01-09	650
12	Bob	250	2023-01-12	900
15	Bob	100	2023-01-15	1000
4	Charlie	250	2023-01-04	250
7	Charlie	350	2023-01-07	600
10	Charlie	400	2023-01-10	1000
13	Charlie	300	2023-01-13	1300



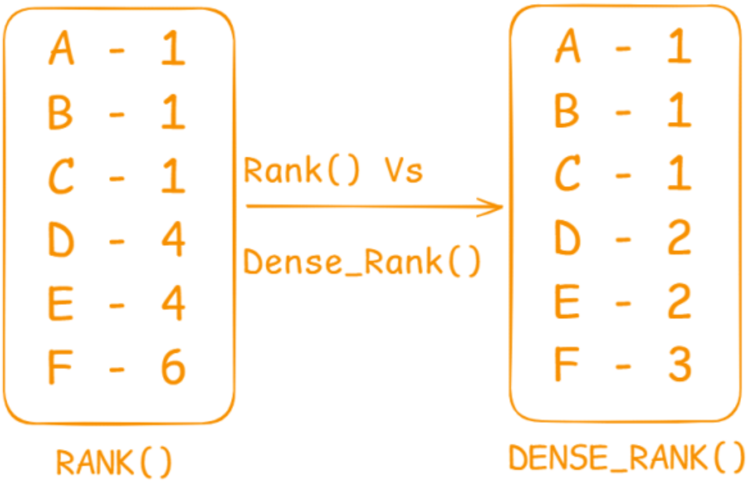
```
SELECT
    SalesPerson,
    SUM(SaleAmount) AS TotalSales
FROM Sale
GROUP BY SalesPerson;
```

SalesPerson	TotalSales
Alice	1550
Bob	1000
Charlie	1300

Rank the SalesPerson by SalesAmount



RANK()
DENSE_RANK()



A - 1
B - 1
C - 1
D - 4
E - 4
F - 6

RANK()

Rank() Vs Dense_Rank()

A - 1
B - 1
C - 1
D - 2
E - 2
F - 3

DENSE_RANK()

Rank() skips when ties , Dense_Rank() not.....

Percentage
Vs Percentile

Percentage[100]

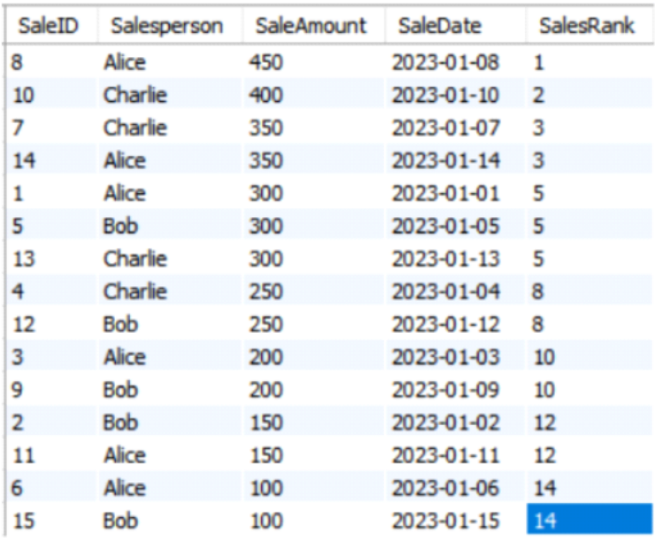
PersonA - 75
PersonB - 90
PersonC - 85
PersonD - 95

Percentile[High Score] [95]

PersonA - 75
PersonB - 90
PersonC - 85
PersonD - 95

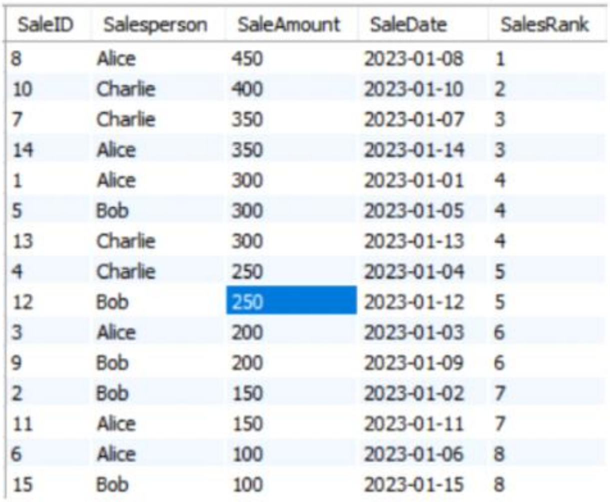
```
SELECT
    *,
    RANK() OVER(
        ORDER BY SaleAmount DESC
) AS SalesRank
FROM Sale;
```

Counter = 1
Counter+=1



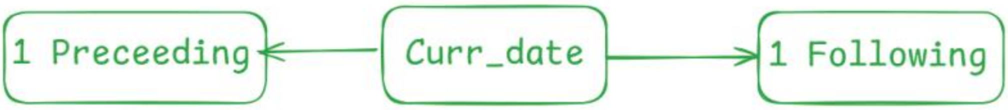
SaleID	Salesperson	SaleAmount	SaleDate	SalesRank
8	Alice	450	2023-01-08	1
10	Charlie	400	2023-01-10	2
7	Charlie	350	2023-01-07	3
14	Alice	350	2023-01-14	3
1	Alice	300	2023-01-01	5
5	Bob	300	2023-01-05	5
13	Charlie	300	2023-01-13	5
4	Charlie	250	2023-01-04	8
12	Bob	250	2023-01-12	8
3	Alice	200	2023-01-03	10
9	Bob	200	2023-01-09	10
2	Bob	150	2023-01-02	12
11	Alice	150	2023-01-11	12
6	Alice	100	2023-01-06	14
15	Bob	100	2023-01-15	14

```
-- Rank() VS Dense_Rank()

SELECT
        *,
    RANK() OVER(
        ORDER BY SaleAmount DESC
) AS SalesRank
FROM Sale;

SELECT
        *,
    DENSE_RANK() OVER(
        ORDER BY SaleAmount DESC
) AS SalesRank
FROM Sale;
```



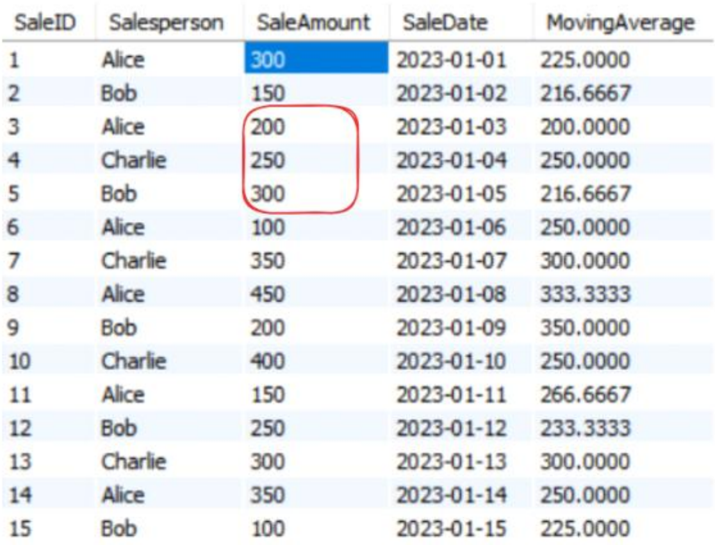
SaleID	Salesperson	SaleAmount	SaleDate	SalesRank
8	Alice	450	2023-01-08	1
10	Charlie	400	2023-01-10	2
7	Charlie	350	2023-01-07	3
14	Alice	350	2023-01-14	3
1	Alice	300	2023-01-01	4
5	Bob	300	2023-01-05	4
13	Charlie	300	2023-01-13	4
4	Charlie	250	2023-01-04	5
12	Bob	250	2023-01-12	5
3	Alice	200	2023-01-03	6
9	Bob	200	2023-01-09	6
2	Bob	150	2023-01-02	7
11	Alice	150	2023-01-11	7
6	Alice	100	2023-01-06	8
15	Bob	100	2023-01-15	8

3 Days Moving Average

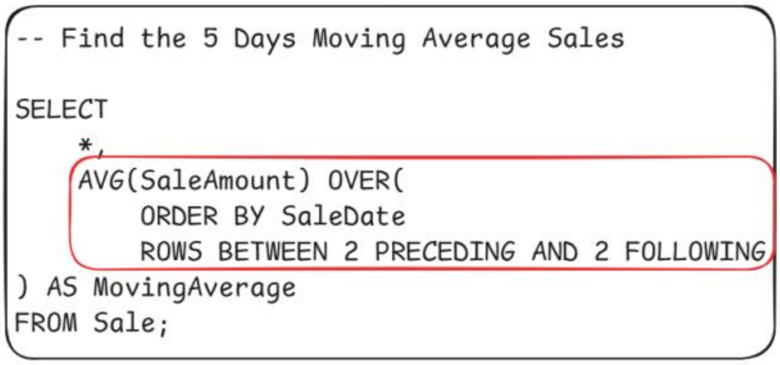


```
-- Find the 3 Days Moving Average Sales
```

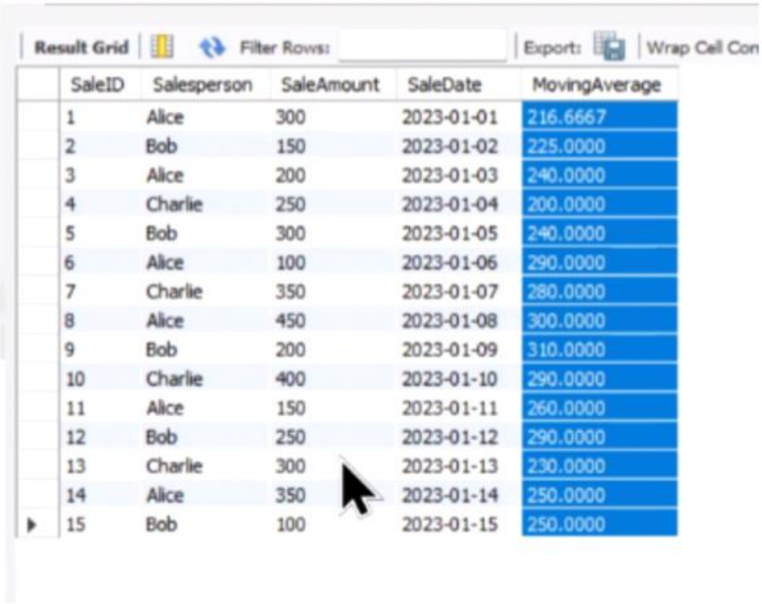
```
SELECT
        *,
    AVG(SaleAmount) OVER(
        ORDER BY SaleDate
    ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING
) AS MovingAverage
FROM Sale;
```



SaleID	Salesperson	SaleAmount	SaleDate	MovingAverage
1	Alice	300	2023-01-01	225.0000
2	Bob	150	2023-01-02	216.6667
3	Alice	200	2023-01-03	200.0000
4	Charlie	250	2023-01-04	250.0000
5	Bob	300	2023-01-05	216.6667
6	Alice	100	2023-01-06	250.0000
7	Charlie	350	2023-01-07	300.0000
8	Alice	450	2023-01-08	333.3333
9	Bob	200	2023-01-09	350.0000
10	Charlie	400	2023-01-10	250.0000
11	Alice	150	2023-01-11	266.6667
12	Bob	250	2023-01-12	233.3333
13	Charlie	300	2023-01-13	300.0000
14	Alice	350	2023-01-14	250.0000
15	Bob	100	2023-01-15	225.0000



```
-- Find the 5 Days Moving Average Sales

SELECT
    *,
    AVG(SaleAmount) OVER(
        ORDER BY SaleDate
        ROWS BETWEEN 2 PRECEDING AND 2 FOLLOWING
) AS MovingAverage
FROM Sale;
```



SaleID	Salesperson	SaleAmount	SaleDate	MovingAverage
1	Alice	300	2023-01-01	216.6667
2	Bob	150	2023-01-02	225.0000
3	Alice	200	2023-01-03	240.0000
4	Charlie	250	2023-01-04	200.0000
5	Bob	300	2023-01-05	240.0000
6	Alice	100	2023-01-06	290.0000
7	Charlie	350	2023-01-07	280.0000
8	Alice	450	2023-01-08	300.0000
9	Bob	200	2023-01-09	310.0000
10	Charlie	400	2023-01-10	290.0000
11	Alice	150	2023-01-11	260.0000
12	Bob	250	2023-01-12	290.0000
13	Charlie	300	2023-01-13	230.0000
14	Alice	350	2023-01-14	250.0000
15	Bob	100	2023-01-15	250.0000