SQL technique: Use DISTINCT *and* GROUP BY to avoid using CTEs or sub-queries

Problem statement:

“Return the maximum single-order sales amount by customer”

We have one row per product/order/customer

```sql
SELECT
    CustomerKey,
    OrderDateKey,
    SalesOrderLineNumber,
    ProductKey,
    SalesAmount
FROM FactInternetSales fis
WHERE CustomerKey = 18005
ORDER BY
    CustomerKey,
    OrderDateKey,
    SalesOrderLineNumber;
```

Use a single customer to explore the problem

	CustomerKey	OrderDateKey	SalesOrderLineNumber	ProductKey	SalesAmount
1	18005	20121220	1	371	2181.5625
2	18005	20130511	1	560	1214.85
3	18005	20130511	2	222	34.99
4	18005	20131119	1	363	2294.99
5	18005	20131119	2	478	9.99
6	18005	20131119	3	477	4.99
7	18005	20131119	4	472	63.50

This order has the largest sum of SalesAmount

Because each order can have more than one line, we must sum by order first

```sql
SELECT
    CustomerKey,
    OrderDateKey,
    SUM(SalesAmount) AS OrderSalesAmount
FROM FactInternetSales fis
WHERE CustomerKey = 18005
GROUP BY CustomerKey, OrderDateKey
ORDER BY CustomerKey, OrderDateKey;
```

	CustomerKey	OrderDateKey	SalesOrderLineNumber	ProductKey	SalesAmount
1	18005	20121220	1	371	2181.5625
2	18005	20130511	1	560	1214.85
3	18005	20130511	2	222	34.99
4	18005	20131119	1	363	2294.99
5	18005	20131119	2	478	9.99
6	18005	20131119	3	477	4.99
7	18005	20131119	4	472	63.50

	CustomerKey	OrderDateKey	OrderSalesAmount
1	18005	20121220	2181.5625
2	18005	20130511	1249.84
3	18005	20131119	2373.47

We can use a CTE for the maximum of the OrderSalesAmount values

```sql
WITH orders
AS
(
    SELECT
        CustomerKey,
        OrderDateKey,
        SUM(SalesAmount) AS OrderSalesAmount
    FROM FactInternetSales fis
    WHERE CustomerKey = 18005
    GROUP BY CustomerKey, OrderDateKey
)
SELECT
    CustomerKey,
    MAX(OrderSalesAmount) AS MaxOrderSalesAmount
FROM orders
GROUP BY CustomerKey;
```

	CustomerKey	OrderDateKey	OrderSalesAmount
1	18005	20121220	2181.5625
2	18005	20130511	1249.84
3	18005	20131119	2373.47

	CustomerKey	MaxOrderSalesAmount
1	18005	2373.47

But, we can also get the MAX *without* a CTE

We can wrap the SUM with a MAX window function, partitioned on CustomerKey

```sql
SELECT
    CustomerKey,
    OrderDateKey,
    SUM(SalesAmount) AS OrderSalesAmount,
    MAX(SUM(SalesAmount))
        OVER (PARTITION BY CustomerKey) AS MaxOrderSalesAmount
FROM FactInternetSales fis
WHERE CustomerKey = 18005
GROUP BY CustomerKey, OrderDateKey
ORDER BY CustomerKey, OrderDateKey;
```

	CustomerKey	OrderDateKey	OrderSalesAmount	MaxOrderSalesAmount
1	18005	20121220	2181.5625	2373.47
2	18005	20130511	1249.84	2373.47
3	18005	20131119	2373.47	2373.47

And if we remove OrderDateKey from the SELECT clause but leave it in GROUP BY...

```sql
SELECT
    CustomerKey,
    MAX(SUM(SalesAmount))
        OVER (PARTITION BY CustomerKey) AS MaxOrderSalesAmount
FROM FactInternetSales fis
WHERE CustomerKey = 18005
GROUP BY CustomerKey, OrderDateKey
ORDER BY CustomerKey;
```

OrderDateKey remains in GROUP BY so that the SUMs are still correct

	CustomerKey	MaxOrderSalesAmount
1	18005	2373.47
2	18005	2373.47
3	18005	2373.47

We now have duplicates!

And how do we get rid of duplicates?

We add DISTINCT!

```sql
SELECT DISTINCT
    CustomerKey,
    MAX(SUM(SalesAmount))
        OVER (PARTITION BY CustomerKey) AS MaxOrderSalesAmount
FROM FactInternetSales fis
GROUP BY CustomerKey, OrderDateKey
ORDER BY CustomerKey;
```

	CustomerKey	MaxOrderSalesAmount
1	18001	31.27
2	18002	27.28
3	18003	63.97
4	18004	819.48
5	18005	2373.47
6	18006	38.98
7	18007	38.98
8	18008	1759.97
9	18009	1735.98
10	18010	1785.97

We returned the max single-order sales amount by customer without sub-queries and without CTEs