

Query (1):

At first, I started to explore the total sales over years.

As seen we had a huge increase in sales 2011 comparing with 2010.

```
select distinct X.year, sum(sale) over(partition by X.year) total_sales
from (select invoiceno , Extract(year from cast(invoicedate as date)) as year,
      (unitprice * quantity) as sale
from online_retail
where quantity > 0 and unitprice > 0) X
order by 1
```

	year numeric	total_sales double precision
1	2010	823746
2	2011	9842938

Query (2):

We drilled down for more details in the country level to investigate the total sales per country

And determined the top 10 countries in selling

```
select distinct X.country, sum(sale) over(partition by X.country ) country_net_sales
from (select invoiceno , Extract(year from cast(invoicedate as date)) as year,
      (unitprice * quantity) as sale , country
from online_retail
where quantity > 0 and unitprice > 0) X
order by 2 desc
limit 10
```

	country text	country_net_sales double precision
1	United Kingdom	9025222
2	Netherlands	285446
3	EIRE	283454
4	Germany	228867
5	France	209715
6	Australia	138521
7	Spain	61577
8	Switzerland	57090
9	Belgium	41196
10	Sweden	38378

Query (3):

We drilled down for more details in both the country and years levels to investigate the total sales per country and the yearly sales in addition to calculating the yearly sales ratio over the total sales.

```

select Y.country , Y.year , Y.country_net_sales , Y.country_yearly_sales,
avg(Y.country_yearly_sales/Y.country_net_sales)
over(partition by Y.country order by Y.country_yearly_sales rows between current row and current
row)*100 as sales_pct
from(select distinct X.country, X.year ,sum(sale) over(partition by X.country ) country_net_sales
, sum(sale) over(partition by X.country ,X.year) country_yearly_sales
from (select invoiceno , Extract(year from cast(invoicedate as date)) as year,
(unitprice * quantity) as sale , country
from online_retail
where quantity > 0 and unitprice > 0) X ) Y
order by 1 , 2

```

	country text	year numeric	country_net_sales double precision	country_yearly_sales double precision	sales_pct double precision
1	United Kingdom	2010	9025222	748269	8.290865011801968
2	United Kingdom	2011	9025222	8276953	91.7091349881914
3	Netherlands	2010	285446	8784	3.07745406719875
4	Netherlands	2011	285446	276662	96.92254593280126
5	EIRE	2010	283454	10033	3.5396436161978504
6	EIRE	2011	283454	273421	96.46035638380195
7	Germany	2010	228867	15241	6.659383256154701
8	Germany	2011	228867	213626	93.3406167438456
9	France	2010	209715	9616	4.585415900647346
10	France	2011	209715	200099	95.4145840993529
11	Australia	2010	138521	1033	0.7456253481865006
12	Australia	2011	138521	137488	99.25437465181346

Query (4):

Simple query to calculate the number of sold items for each item in our stock.

Then we retrieve the top 5 sold items over all the world

```

select distinct stockcode, description, sum(quantity) over(partition by stockcode) quantity_sold
from online_retail
where quantity >= 0
order by 3 desc
limit 5

```

	stockcode text	description text	quantity_sold bigint
1	23843	PAPER CRAFT \, LITTLE BIRDIE	80995
2	23166	MEDIUM CERAMIC TOP STORAGE JAR	78033
3	22197	SMALL POPCORN HOLDER	56921
4	22197	POPCORN HOLDER	56921
5	84077	WORLD WAR 2 GLIDERS ASSTD DESIGNS	55047

Query (5):

We drilled down for more details in the country level to investigate quantity sold per country for each item and determined the top 3 sold items in each country

```
select X.country , X.stockcode, X.quantity_sold, X.Rank
from (select distinct S.country, S.stockcode , S.quantity_sold
, dense_rank() over(partition by country order by S.quantity_sold desc) Rank
from( select distinct stockcode,country, description, sum(quantity) over(partition by country,
stockcode) quantity_sold
from online_retail
where quantity >= 0) S
order by 1 ,3 desc ) X
where X.Rank in(1,2,3)
```

	country text	stockcode text	quantity_sold bigint	rank bigint
1	Australia	22492	2952	1
2	Australia	23084	1884	2
3	Australia	21915	1704	3
4	Austria	21918	288	1
5	Austria	22546	240	2
6	Austria	22584	144	3
7	Austria	22582	144	3
8	Bahrain	23076	96	1
9	Bahrain	23077	60	2
10	Bahrain	72802B	60	2
11	Bahrain	22693	24	3

Query (6):

I started to explore the total number of orders over years.

As seen we had a huge increase in orders in 2011 comparing with 2010.

```
select distinct X.year, count(X.invoiceno) over(partition by X.year) number_of_orders
from (select distinct invoiceno , Extract(year from cast(invoicedate as date)) as year
from online_retail) X
order by 1
```

	year numeric	number_of_orders bigint
1	2010	2025
2	2011	23875

Query (7):

We drilled down to the level of the country to show the number of orders in each year for each country.

```
select distinct X.country , X.year, count(*) over(partition by X.year, X.country)
number_of_orders
from (select distinct invoiceno , Extract(year from cast(invoicedate as date)) as year , country
from online_retail) X
order by 1 , 2
```

	country text	year numeric	number_of_orders bigint
1	Australia	2010	4
2	Australia	2011	65
3	Austria	2010	2
4	Austria	2011	17
5	Bahrain	2010	1
6	Bahrain	2011	3
7	Belgium	2010	5
8	Belgium	2011	114

Query (8):

Ranking the customers based on the purchasing frequency criteria.

```
select customerid , number_of_purchasing , rank() over(order by number_of_purchasing desc)
from
(with freq As (
select customerid , count(distinct invoiceno) as frequency
from online_retail
group by customerid
having customerid != '')
select distinct customerid, freq.frequency as number_of_purchasing
from online_retail O Join freq using(customerid)
order by 2 desc) X
```

	customerid text	number_of_purchasing bigint	rank bigint
1	14911	248	1
2	12748	224	2
3	17841	169	3
4	14606	128	4
5	13089	118	5
6	15311	118	5
7	12971	89	7
8	14527	86	8
9	13408	81	9
10	14646	77	10

Query (9):

Ranking the customers based on the total sales they did over the years.

```
select Y.customerid , round(Y.customer_net_sales) customer_net_sales , rank()  
over(order by Y.customer_net_sales desc)  
from (select distinct X.customerid, sum(sales) over(partition by X.customerid )  
customer_net_sales  
from (select invoiceno, (unitprice * quantity) as sales , customerid  
from online_retail  
where quantity > 0 and unitprice > 0 and customerid !='') X ) Y
```

	customerid text	customer_net_sales double precision	rank bigint
1	14646	280206	1
2	18102	259657	2
3	17450	194551	3
4	16446	168472	4
5	14911	143825	5
6	12415	124915	6
7	14156	117380	7
8	17511	91062	8
9	16029	81025	9
10	12346	77184	10