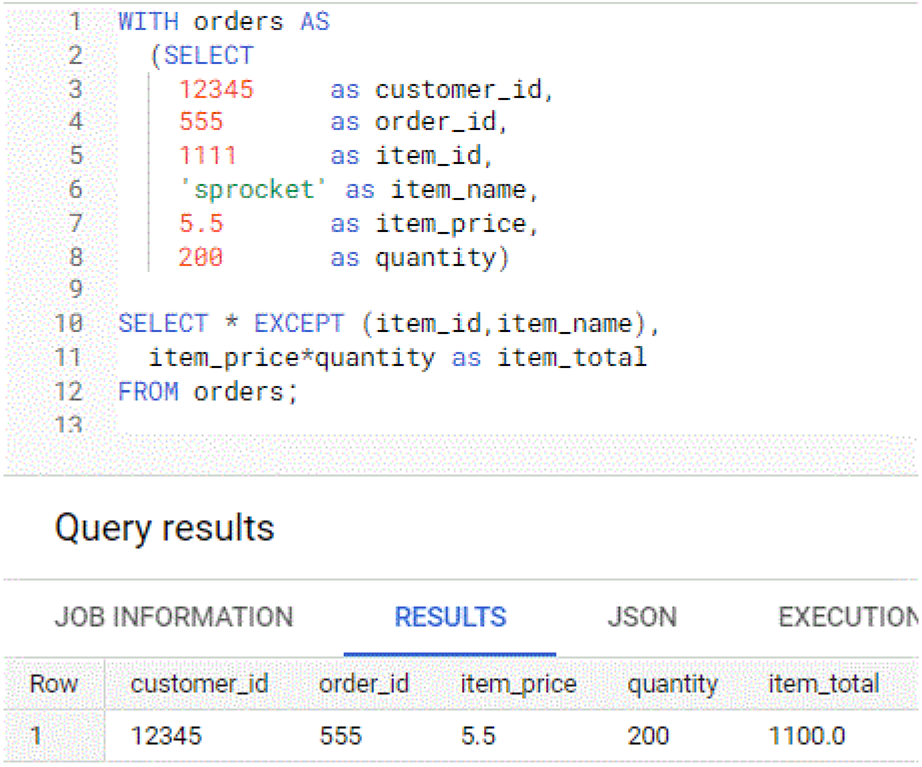
1. EXCEPT

Use EXCEPT in a SELECT * to select **all** fields from a table **except** the columns you don't want. In the example below, I created a CTE named **orders** and selected **all** of the columns **except** for **item_id** and **item_name**.



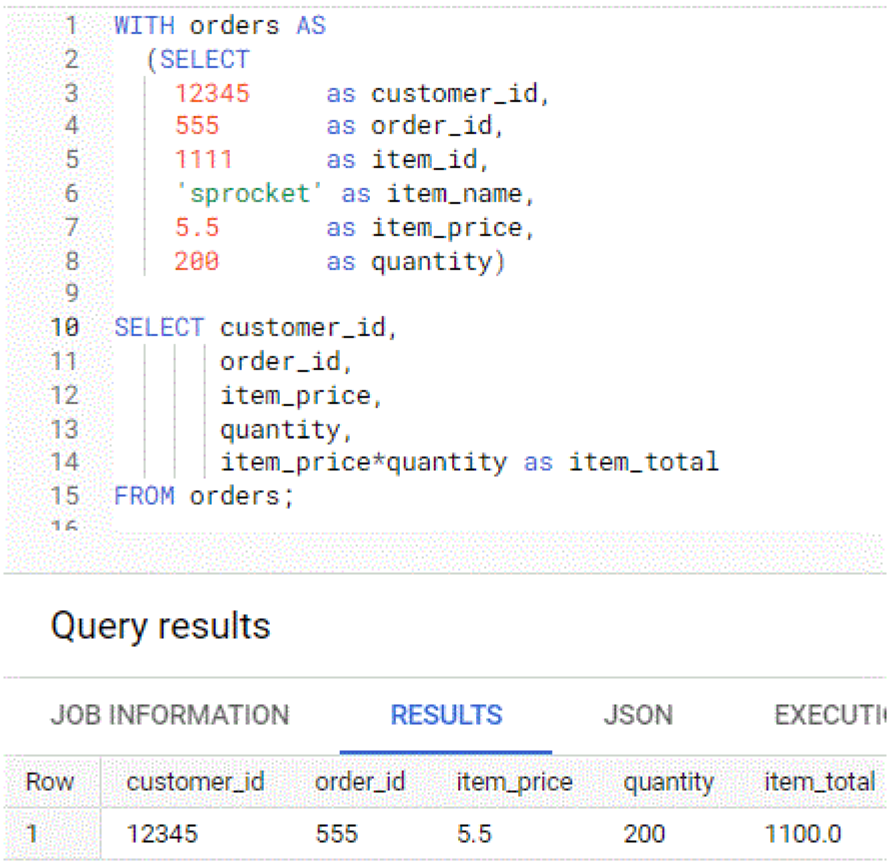
```
WITH orders AS
  (SELECT
   12345      as customer_id,
   555        as order_id,
   1111       as item_id,
   'sprocket'  as item_name,
   5.5        as item_price,
   200        as quantity)

SELECT * EXCEPT (item_id,item_name),
  item_price*quantity as item_total
FROM orders;
```

Query results

Row	customer_id	order_id	item_price	quantity	item_total
1	12345	555	5.5	200	1100.0

Screenshot with EXCEPT query example created by author

Prior to this, I would've written the SELECT below by listing **every column except** for **item_id** and **item_name**. This doesn't seem like a shortcut with 6 columns but imagine if you have a 20 column table and you just want to exclude 2 fields. Instead of typing 18 column names in a SELECT statement, you can simply use an EXCEPT to exclude the 2 columns you don't want.



```
WITH orders AS
  (SELECT
   12345      as customer_id,
   555        as order_id,
   1111       as item_id,
   'sprocket' as item_name,
   5.5        as item_price,
   200        as quantity)

SELECT customer_id,
       order_id,
       item_price,
       quantity,
       item_price*quantity as item_total
FROM orders;
```

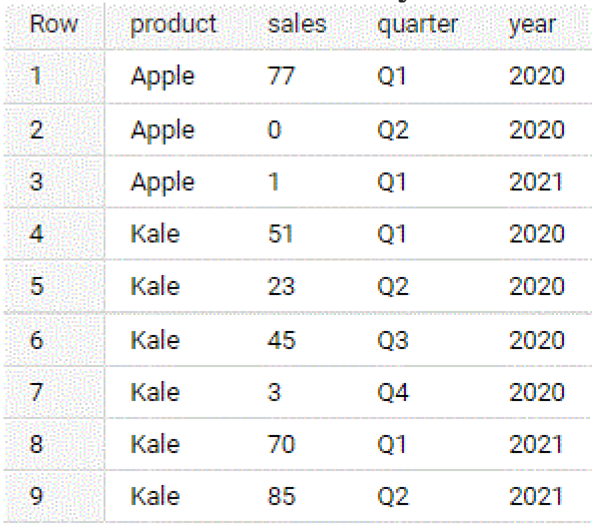
Query results

Row	customer_id	order_id	item_price	quantity	item_total
1	12345	555	5.5	200	1100.0

Screenshot without EXCEPT query example created by author

2. PIVOT

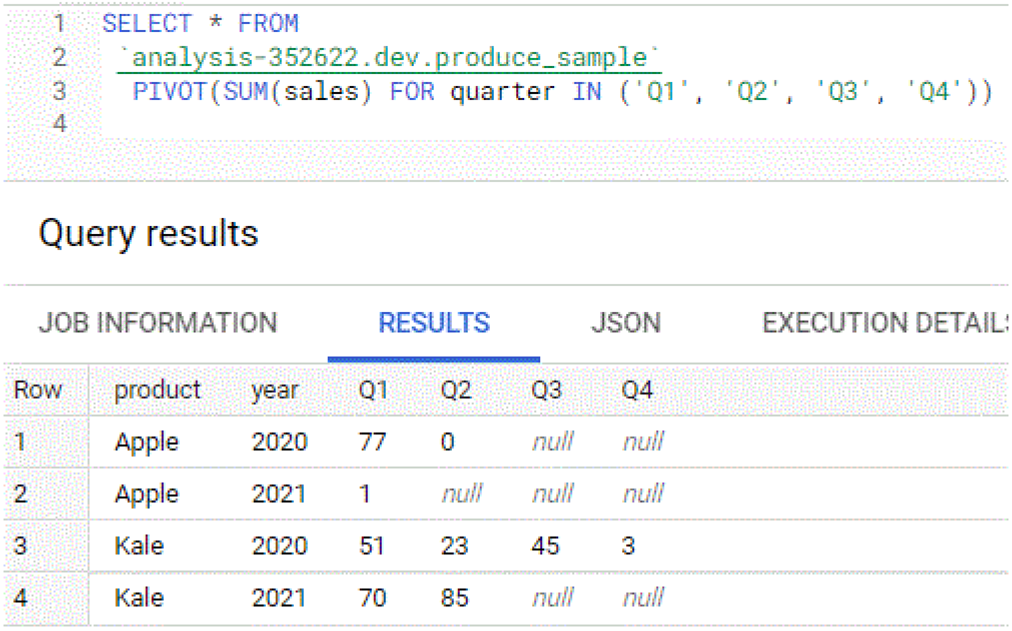
I lost track of how many times I queried data from a table to put into Excel to get subtotals by different time periods using a pivot table. Now I can do this easily with a PIVOT statement and bypass Excel.



Row	product	sales	quarter	year
1	Apple	77	Q1	2020
2	Apple	0	Q2	2020
3	Apple	1	Q1	2021
4	Kale	51	Q1	2020
5	Kale	23	Q2	2020
6	Kale	45	Q3	2020
7	Kale	3	Q4	2020
8	Kale	70	Q1	2021
9	Kale	85	Q2	2021

Screenshot sample produce data created by author

To get sales pivoted by quarter using the sample data above you just need to specify the quarters in your PIVOT statement.



```
SELECT * FROM
 `analysis-352622.dev.produce_sample`
  PIVOT(SUM(sales) FOR quarter IN ('Q1', 'Q2', 'Q3', 'Q4'))

```

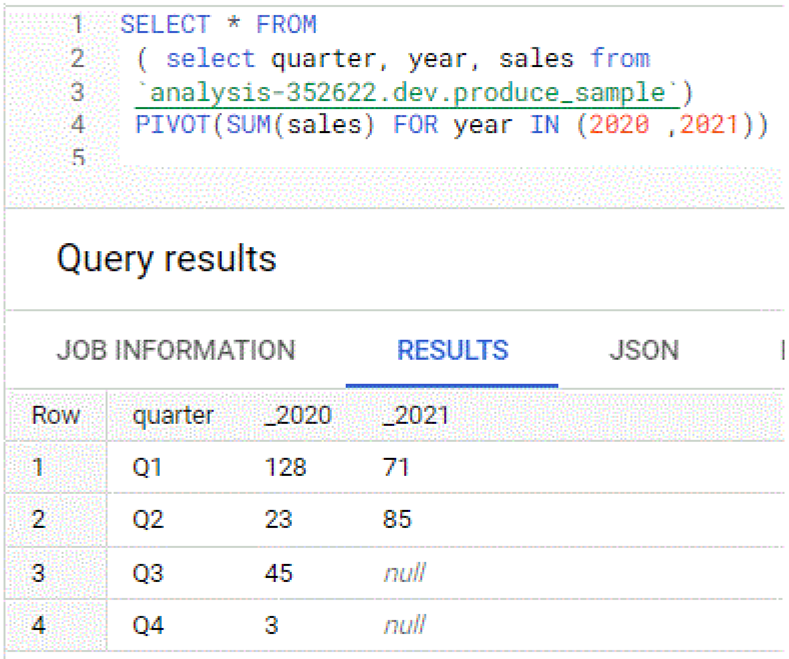
Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	product	year	Q1	Q2	Q3	Q4
1	Apple	2020	77	0	*null*	*null*
2	Apple	2021	1	*null*	*null*	*null*
3	Kale	2020	51	23	45	3
4	Kale	2021	70	85	*null*	*null*

Screenshot of PIVOT by quarter created by author

Alternatively, to compare year over year total sales by quarter you can drop the product column and specify the years in the PIVOT statement.



```
SELECT * FROM
 ( select quarter, year, sales from
 `analysis-352622.dev.produce_sample`)
 PIVOT(SUM(sales) FOR year IN (2020 ,2021))

```

Query results

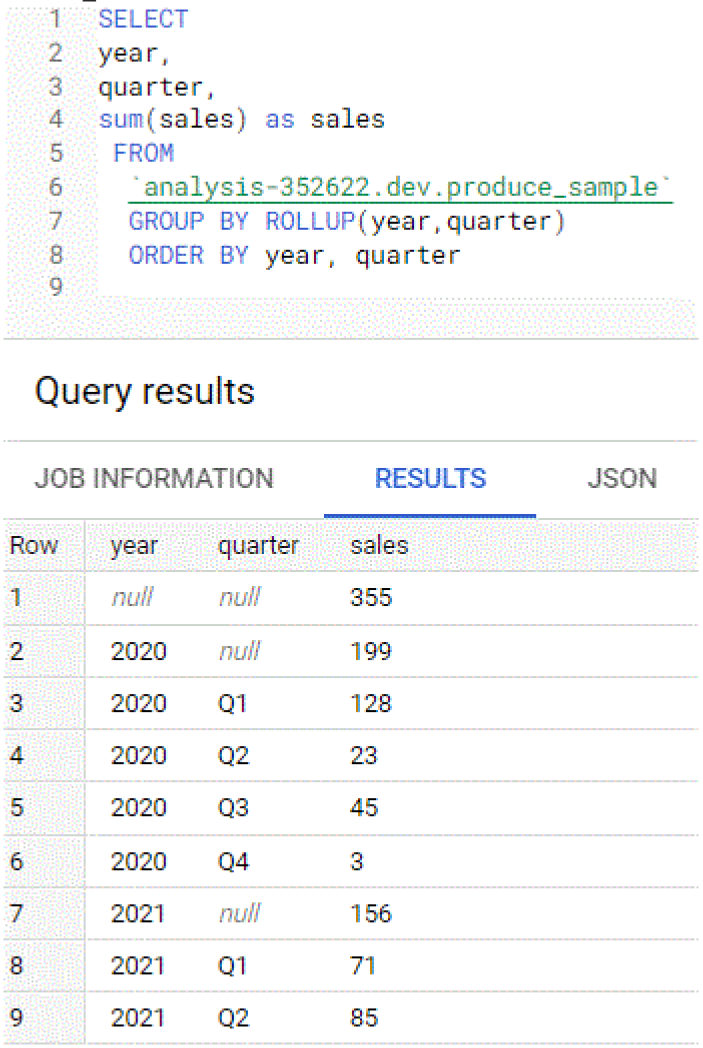
JOB INFORMATION RESULTS JSON

Row	quarter	_2020	_2021
1	Q1	128	71
2	Q2	23	85
3	Q3	45	*null*
4	Q4	3	*null*

Screenshot of PIVOT by year created by author

3. ROLLUP

In the past, I used an Excel pivot table or ran multiple queries to get the total and subtotals but now I can use ROLLUP with GROUP BY instead. Using the product sample shown above we can get total sales and subtotals by year using **ROLLUP (year, quarter)** after the GROUP BY. Total sales for **2020** and **2021** are **355** denoted by the **null** values in the **year** and **quarter** columns. **2020** sales are **199** and **2021** is **156** denoted by the **null** values in the **quarter** column.



```
SELECT
year,
quarter,
sum(sales) as sales
 FROM
  `analysis-352622.dev.produce_sample`
  GROUP BY ROLLUP(year,quarter)
  ORDER BY year, quarter

```

Query results

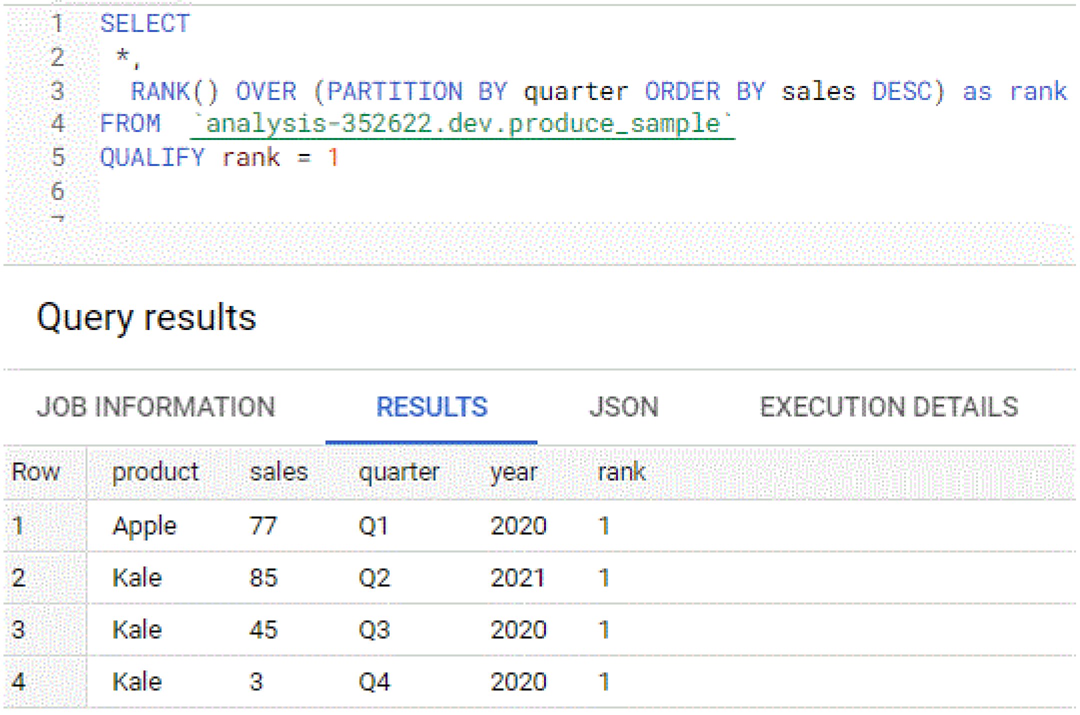
Row	year	quarter	sales
1	*null*	*null*	355
2	2020	*null*	199
3	2020	Q1	128
4	2020	Q2	23
5	2020	Q3	45
6	2020	Q4	3
7	2021	*null*	156
8	2021	Q1	71
9	2021	Q2	85

Screenshot of sales by year and quarter using ROLLUP

4. QUALIFY

QUALIFY allows you to apply it like a WHERE condition on a column created in your SELECT statement because it’s **evaluated after the GROUP BY, HAVING, and WINDOW statements.**

You can use QUALIFY to get the product with the highest sales by quarter calculated with a rank window function by simply using QUALIFY = 1.



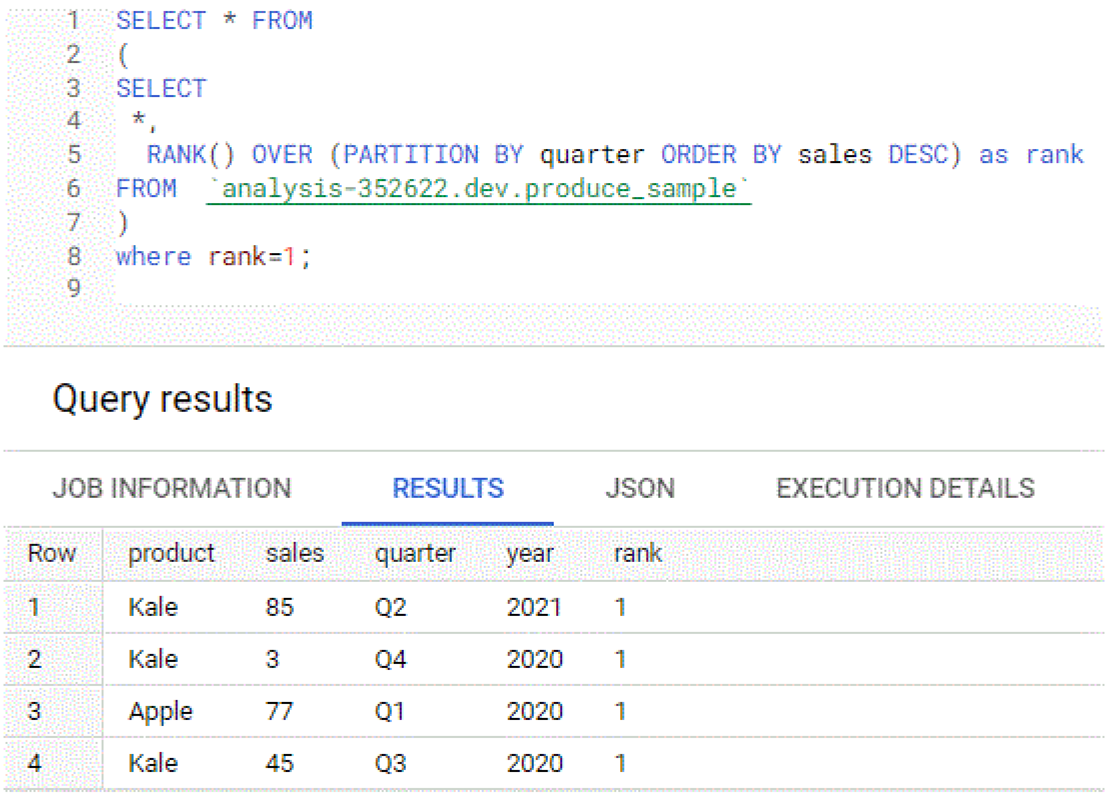
```
SELECT
 *,
  RANK() OVER (PARTITION BY quarter ORDER BY sales DESC) as rank
FROM  `analysis-352622.dev.produce_sample`
QUALIFY rank = 1

```

Query results

Row	product	sales	quarter	year	rank
1	Apple	77	Q1	2020	1
2	Kale	85	Q2	2021	1
3	Kale	45	Q3	2020	1
4	Kale	3	Q4	2020	1

Screenshot using QUALIFY = 1

The traditional way without using QUALIFY is to have the SQL statement as a subquery and then apply a WHERE rank = 1 like I have below. Seems a lot simpler with QUALIFY right?



```sql
SELECT * FROM
(
SELECT
 *,
  RANK() OVER (PARTITION BY quarter ORDER BY sales DESC) as rank
FROM  `analysis-352622.dev.produce_sample`
)
where rank=1;
```

Query results

JOB INFORMATION   RESULTS   JSON   EXECUTION DETAILS

Row	product	sales	quarter	year	rank
1	Kale	85	Q2	2021	1
2	Kale	3	Q4	2020	1
3	Apple	77	Q1	2020	1
4	Kale	45	Q3	2020	1

Screenshot using WHERE rank =1