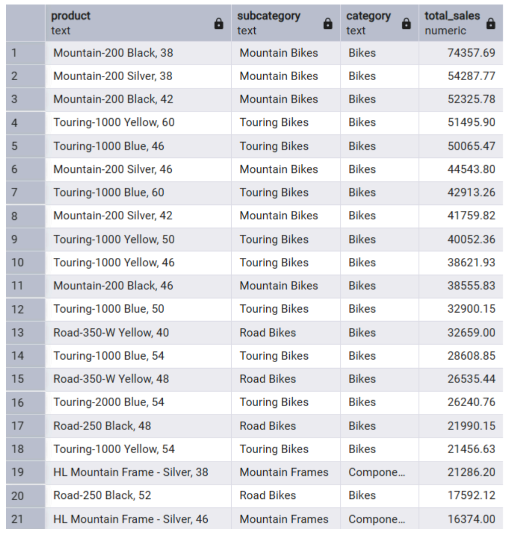
Amber Locasto
SQL Final Project
Analysis Report

Question 1: Finding the total sales by product in descending order:

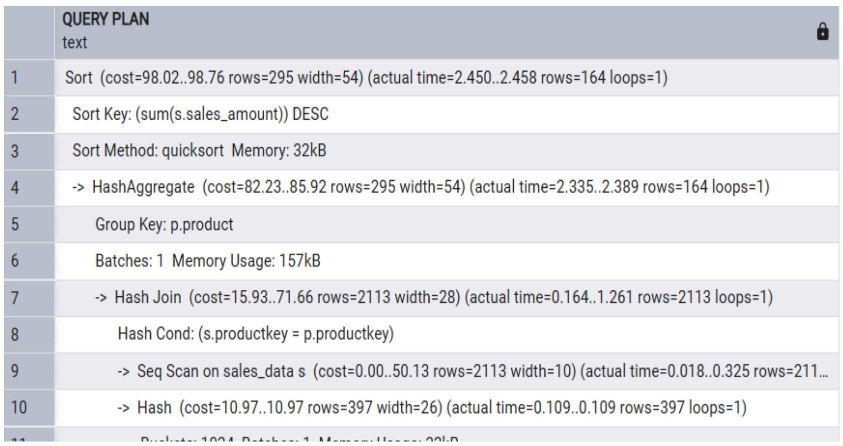
Code breakdown:

- SELECT→ choose the product from the product data table and sales amount as a sum amount alias as total sales from the sales data table
- FROM → sales data alias as s
- JOIN → product data as p using the product key column name from both tables
- GROUP BY → group the rows that have the same product name together so we can sum each group
- ORDER BY → total sales in descending order so the top amount should be the highest amount in total sales per product.

Analysis: The mountain-200 Black 38 product is the highest sales amount with a total sales of $74,357.69
Explain Analyze Query: I discuss these query plans in question 11. From the output we see some of our highest earners are mountain bikes, touring bikes, and road bikes generating a majority of our revenue.



	product text	subcategory text	category text	total_sales numeric
1	Mountain-200 Black, 38	Mountain Bikes	Bikes	74357.69
2	Mountain-200 Silver, 38	Mountain Bikes	Bikes	54287.77
3	Mountain-200 Black, 42	Mountain Bikes	Bikes	52325.78
4	Touring-1000 Yellow, 60	Touring Bikes	Bikes	51495.90
5	Touring-1000 Blue, 46	Touring Bikes	Bikes	50065.47
6	Mountain-200 Silver, 46	Mountain Bikes	Bikes	44543.80
7	Touring-1000 Blue, 60	Touring Bikes	Bikes	42913.26
8	Mountain-200 Silver, 42	Mountain Bikes	Bikes	41759.82
9	Touring-1000 Yellow, 50	Touring Bikes	Bikes	40052.36
10	Touring-1000 Yellow, 46	Touring Bikes	Bikes	38621.93
11	Mountain-200 Black, 46	Mountain Bikes	Bikes	38555.83
12	Touring-1000 Blue, 50	Touring Bikes	Bikes	32900.15
13	Road-350-W Yellow, 40	Road Bikes	Bikes	32659.00
14	Touring-1000 Blue, 54	Touring Bikes	Bikes	28608.85
15	Road-350-W Yellow, 48	Road Bikes	Bikes	26535.44
16	Touring-2000 Blue, 54	Touring Bikes	Bikes	26240.76
17	Road-250 Black, 48	Road Bikes	Bikes	21990.15
18	Touring-1000 Yellow, 54	Touring Bikes	Bikes	21456.63
19	HL Mountain Frame - Silver, 38	Mountain Frames	Compone...	21286.20
20	Road-250 Black, 52	Road Bikes	Bikes	17592.12
21	HL Mountain Frame - Silver, 46	Mountain Frames	Compone...	16374.00



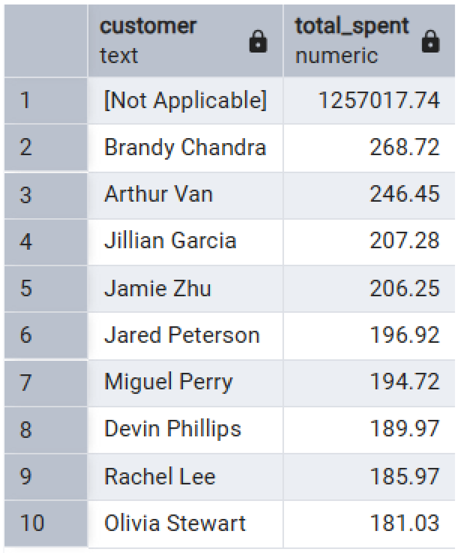
```
QUERY PLAN
text
Sort  (cost=98.02..98.76 rows=295 width=54) (actual time=2.450..2.458 rows=164 loops=1)
  Sort Key: (sum(s.sales_amount)) DESC
  Sort Method: quicksort  Memory: 32kB
  -> HashAggregate  (cost=82.23..85.92 rows=295 width=54) (actual time=2.335..2.389 rows=164 loops=1)
       Group Key: p.product
       Batches: 1  Memory Usage: 157kB
       -> Hash Join  (cost=15.93..71.66 rows=2113 width=28) (actual time=0.164..1.261 rows=2113 loops=1)
            Hash Cond: (s.productkey = p.productkey)
            -> Seq Scan on sales_data s  (cost=0.00..50.13 rows=2113 width=10) (actual time=0.018..0.325 rows=211...
            -> Hash  (cost=10.97..10.97 rows=397 width=26) (actual time=0.109..0.109 rows=397 loops=1)
```

Question 2: Top 10 customers by total purchase

Code breakdown:

- SELECT→ choose customer form customer data table and the sum sales amount from the sales data table alias as total spent (total each customer spent)
- FROM → sales data alias as s
- JOIN → customer data as c using the customer key matching the sales data customer key
- GROUP BY → summarizing all sales per customer
- ORDER BY → sorting customers by total spent with the highest amount first (descending)
- LIMIT 10: only show top 10 customers

Analysis: This query identifies which customers generated the highest total revenue. This is important to analyze since our top buyers help businesses with customer segmentation, loyalty strategies and target marketing. The very top shows the not applicable customer with over a million in total spent which would lead us back to our customer key -1. This seemed to be a data entry issue as I inspected the data in the sales data table and the -1 was found in every single customer key cell. Ignoring this issue, if we move to the second box we can see the highest total spent was by Brandy Chandra with $268.72.



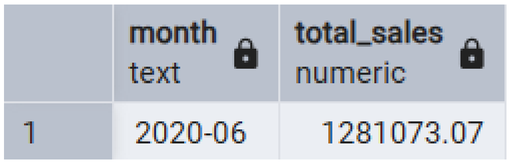
	customer text	total_spent numeric
1	[Not Applicable]	1257017.74
2	Brandy Chandra	268.72
3	Arthur Van	246.45
4	Jillian Garcia	207.28
5	Jamie Zhu	206.25
6	Jared Peterson	196.92
7	Miguel Perry	194.72
8	Devin Phillips	189.97
9	Rachel Lee	185.97
10	Olivia Stewart	181.03

Question 3: Monthly sales totals for the most recent year

Code breakdown:

- SELECT→ TO_CHAR is formatting the truncated date as a 06-2020 format and the DATE_TRUC('month', d."Date") is rounding each date down to the first day of its month. We also find the total sales for each month using a SUM aggregate function
- FROM → pulling from the sales data table aliased as s
- JOIN → joining two tables sales data and date data using datekey and orderdatekey
- WHERE → filtering only the latest year available in the dataset
- Subquery → finding the most recent year in the data
- GROUP BY → summarizing the sales by month
- ORDER BY →sorting the months in order

Analysis: Upon figuring out this query I needed to run some sanity checks like checking data types to look at dates and run a query to look at which years exist in the date table by using an EXTRACT(YEAR FROM "Date") method. This allowed me to look at the date range which was 2017-2021. This told me my query should show me some output from the 2021 year but this ended up not being the case as the year 2021 actually had no sales data to pull from. The year 2020 was our latest year with sufficient data for total sales we could pull from. In order to confirm that there was no data from 2020 I ran a simple query to pull sales data from the year 2020 and nothing showed in my output table. Analyzing this output was difficult and tricky as I was expecting to possibly see more months on our output but it shows only June 2020 which means we only have sales data from June of 2020 at a grand total of $1,281,073.07.



	month text	total_sales numeric
1	2020-06	1281073.07

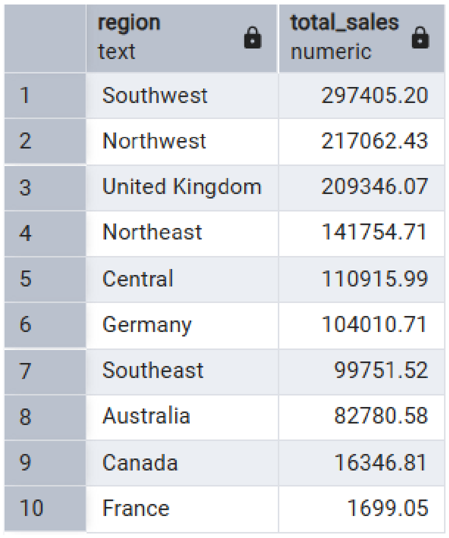
Amber Locasto
SQL Final Project
Analysis Report

Question 4: Sales by Region

Code breakdown:

- SELECT→ showing region and the sum of sales amount alias as total sales
- FROM → pulling from sales data table alias as s
- JOIN → joining sales territory data with sales data using the sales territory key
- GROUP BY → region level
- ORDER BY → sorting the highest revenue regions first with the descending key word

Analysis: Our output shows 10 different regions and their total sales per region. From our findings it shows that the Southwest region has the highest total sales amount at $297,405.20. Upon completing the roll up in the advanced analysis we can see how we can use group by to get the same output to determine who is the top market. The roll up will show us each category breakdown which is more useful for analytics as we can determine what category products are contributing to that region's revenue.



	region text	total_sales numeric
1	Southwest	297405.20
2	Northwest	217062.43
3	United Kingdom	209346.07
4	Northeast	141754.71
5	Central	110915.99
6	Germany	104010.71
7	Southeast	99751.52
8	Australia	82780.58
9	Canada	16346.81
10	France	1699.05

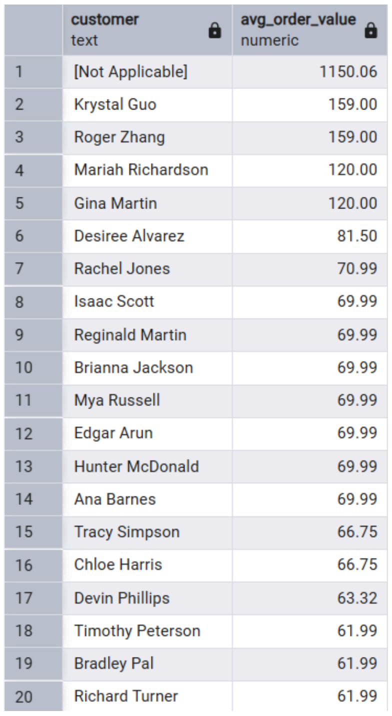
Question 5: Average Order Value Per Customer

Code breakdown:

- SELECT→ customer and average sales amount aliased as avg_order_value
- FROM → sales data table aliases as s
- JOIN → customer data with sales data using the customer key
- GROUP BY → separates for each customer
- ORDER BY → which customer has the highest average order size using descending

Analysis: Our goal in analyzing this output is to find the customers who tend to make larger purchases which we can see in our table Krystal Guo is the customer who makes the largest purchase. This suggests they may be making larger purchases like touring bikes or mountain bikes. There is a large majority of customers at $69.99 which may mean that these customers purchased a single item at this price point and is a strong indicator of a popular item priced at $69.99.

Amber Locasto
SQL Final Project
Analysis Report



	customer text	avg_order_value numeric
1	[Not Applicable]	1150.06
2	Krystal Guo	159.00
3	Roger Zhang	159.00
4	Mariah Richardson	120.00
5	Gina Martin	120.00
6	Desiree Alvarez	81.50
7	Rachel Jones	70.99
8	Isaac Scott	69.99
9	Reginald Martin	69.99
10	Brianna Jackson	69.99
11	Mya Russell	69.99
12	Edgar Arun	69.99
13	Hunter McDonald	69.99
14	Ana Barnes	69.99
15	Tracy Simpson	66.75
16	Chloe Harris	66.75
17	Devin Phillips	63.32
18	Timothy Peterson	61.99
19	Bradley Pal	61.99
20	Richard Turner	61.99

ADVANCED ANALYSIS:

Question 1: Top 3 Product Per Category

Code breakdown:

- CTE table → This is my first time using a CTE table (common table expression) we name is product_sales and we use it as a temporary table for our next query
- SELECT→ selecting the product categories like bikes, clothing and accessories, the product showing us the specific product name, and a sum of total revenue for each product across all sales rows.
- FROM → sales data table aliased as s
- JOIN →joining the product data and sales data tables using the product ket
- GROUP BY → grouping by product within each category

SECOND PART OF QUERY

- CTE table → Creating another CTE to compute rankings so we named it ranked
- SELECT → all columns from product sales, category, product and total sales
- DENSE RANK→ creating rankings based on windows function which performs calculations across a window of rows
- PARTITION BY→ restarting the ranking for every product category (1, 2, 3)
- ORDER BY→ highest selling products will get a rank of 1 and so on… naming it rnk

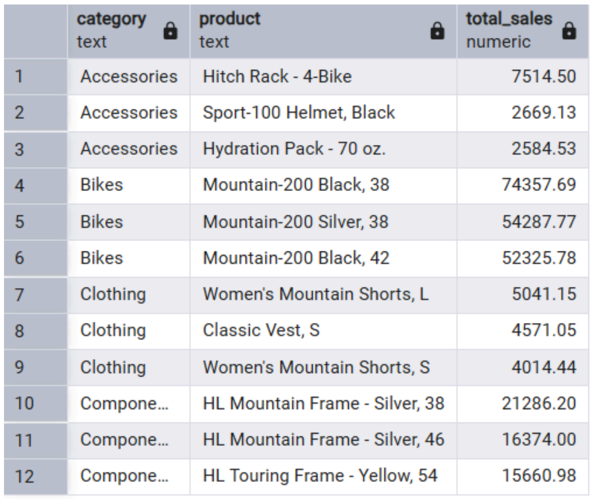
FINAL QUERY

- SELECT → showing category, product and total sales
- FROM → ranked CTE
- WHERE → filtering results to only keep the top three products in each category

Amber Locasto
SQL Final Project
Analysis Report

- ORDER BY → the list we provide will rank from top to bottom starting with category and ending with sales in descending order

Analysis: We have four different categories and the output shows top 3 products for each category giving us a total of 12 items This query identifies the three highest earning products within each category using the CTE for product sales with an aggregation for revenue and a window function dense rank to rank the products inside each category. The hitch rack 4 bike product massively outperforms the other accessories generating almost 3X the revenue of the second place item. This suggests a larger demand for this product along with a higher price tag. The black mountain 200 bike is taking the crown for revenue for bikes category which means it is our best selling bike and we may want to prioritize inventory for this product since it is popular and best selling. For the clothing category, the women's mountain shorts are best sellers and they actually appear twice in the output meaning the size large and small are the top clothing performers across all sizes and clothing.



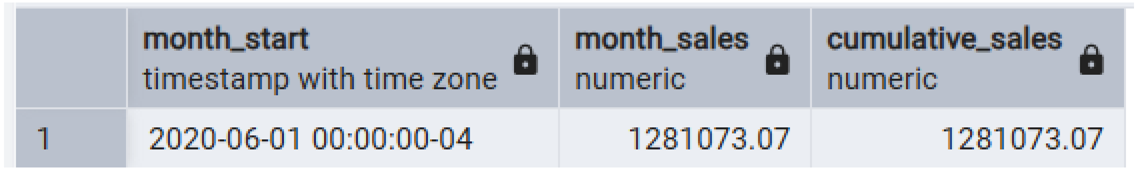
	category text	product text	total_sales numeric
1	Accessories	Hitch Rack - 4-Bike	7514.50
2	Accessories	Sport-100 Helmet, Black	2669.13
3	Accessories	Hydration Pack - 70 oz.	2584.53
4	Bikes	Mountain-200 Black, 38	74357.69
5	Bikes	Mountain-200 Silver, 38	54287.77
6	Bikes	Mountain-200 Black, 42	52325.78
7	Clothing	Women's Mountain Shorts, L	5041.15
8	Clothing	Classic Vest, S	4571.05
9	Clothing	Women's Mountain Shorts, S	4014.44
10	Compone...	HL Mountain Frame - Silver, 38	21286.20
11	Compone...	HL Mountain Frame - Silver, 46	16374.00
12	Compone...	HL Touring Frame - Yellow, 54	15660.98

Question 2: Running a Cumulative Total of Monthly Sales

Code breakdown:

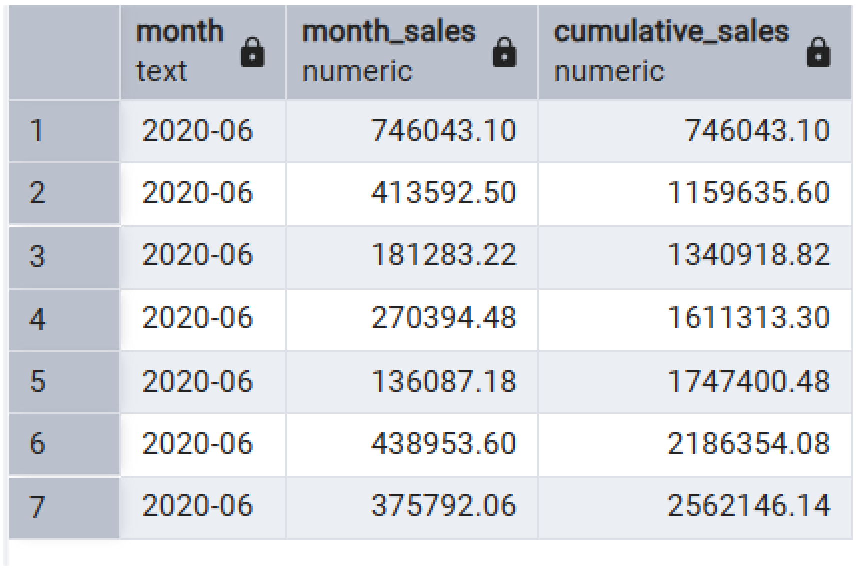
- SELECT→ DATE TRUNC is bucketing every sale into its month, SUM sales amount aggregates per month, SUM of SUM for sales amount windows over the monthly sums to accumulate a running total for the first month to the current month
- FROM → sales data table as d
- JOIN → joining sales data table and date date using the order data key
- GROUP BY → because we are aggregating monthly totals, every month becomes one row
- ORDER BY → chronological output for the first day of the month

Analysis: Like we discovered earlier June 2020 is the only month with sales data so we only get a running total for June. By using the SUM(SUM windows function we are able to create a running total for the month of interest.



	month_start timestamp with time zone	month_sales numeric	cumulative_sales numeric
1	2020-06-01 00:00:00-04	1281073.07	1281073.07

Amber Locasto
SQL Final Project
Analysis Report

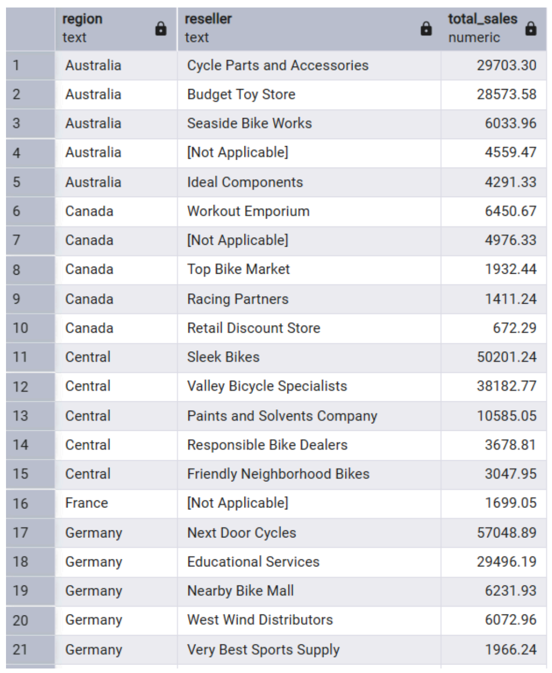


	month text	month_sales numeric	cumulative_sales numeric
1	2020-06	746043.10	746043.10
2	2020-06	413592.50	1159635.60
3	2020-06	181283.22	1340918.82
4	2020-06	270394.48	1611313.30
5	2020-06	136087.18	1747400.48
6	2020-06	438953.60	2186354.08
7	2020-06	375792.06	2562146.14

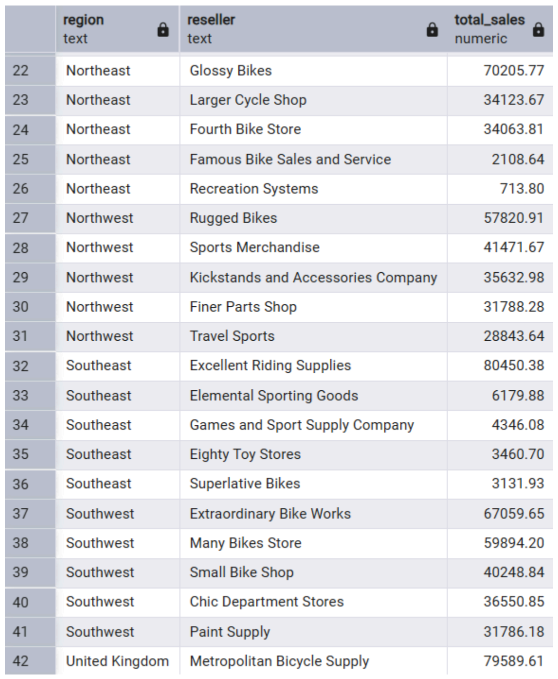
Question 3: Top 5 resellers per region
Code breakdown:
- CTE to store our reseller totals
- JOIN → joining sales to resellers and territories so we can roll up by region
- SUM → the total per region and reseller
- DENSE RANK→ ranking the resellers inside each region
- WHERE→ filtering to keep top 5 per region
- ORDER BY → regions and total sales in descending

Analysis: Our query ranks resellers within each region by their total sales and gives us the top 5 per region. France only has one reseller listed so this means our data for France may not exceed an amount able to output 5 top resellers. Our regional sales seem to vary strongly as we can see Central and Germany have the strongest reseller performances and regions like Canada and Australia have lower sales but still multiple resellers contributing. It is interesting to see that some distributions within regions are uneven showing one or two resellers may dominate the market for example our Central region the top reseller is around 50K and the 2nd is 38K and third is 10K down to fourth and fifth at 3K this is suggesting that there is a high concentration of revenue among the top resellers. France only having one reseller is important to analyze since this could mean there is a smaller market presence or we may have run into some missing or incomplete data for France.

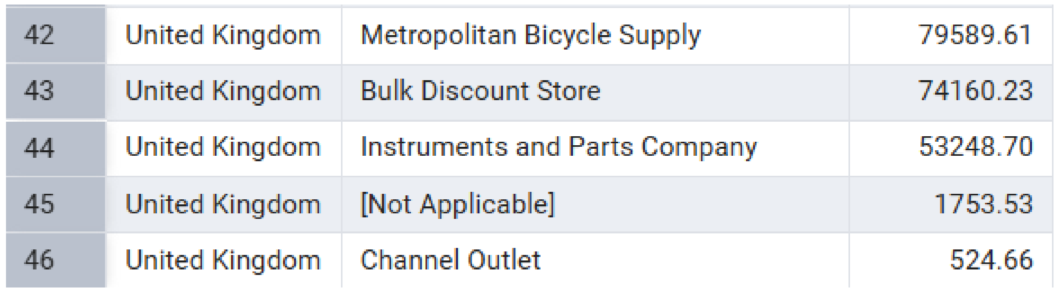
Amber Locasto
SQL Final Project
Analysis Report



	region text	reseller text	total_sales numeric
1	Australia	Cycle Parts and Accessories	29703.30
2	Australia	Budget Toy Store	28573.58
3	Australia	Seaside Bike Works	6033.96
4	Australia	[Not Applicable]	4559.47
5	Australia	Ideal Components	4291.33
6	Canada	Workout Emporium	6450.67
7	Canada	[Not Applicable]	4976.33
8	Canada	Top Bike Market	1932.44
9	Canada	Racing Partners	1411.24
10	Canada	Retail Discount Store	672.29
11	Central	Sleek Bikes	50201.24
12	Central	Valley Bicycle Specialists	38182.77
13	Central	Paints and Solvents Company	10585.05
14	Central	Responsible Bike Dealers	3678.81
15	Central	Friendly Neighborhood Bikes	3047.95
16	France	[Not Applicable]	1699.05
17	Germany	Next Door Cycles	57048.89
18	Germany	Educational Services	29496.19
19	Germany	Nearby Bike Mall	6231.93
20	Germany	West Wind Distributors	6072.96
21	Germany	Very Best Sports Supply	1966.24



	region text	reseller text	total_sales numeric
22	Northeast	Glossy Bikes	70205.77
23	Northeast	Larger Cycle Shop	34123.67
24	Northeast	Fourth Bike Store	34063.81
25	Northeast	Famous Bike Sales and Service	2108.64
26	Northeast	Recreation Systems	713.80
27	Northwest	Rugged Bikes	57820.91
28	Northwest	Sports Merchandise	41471.67
29	Northwest	Kickstands and Accessories Company	35632.98
30	Northwest	Finer Parts Shop	31788.28
31	Northwest	Travel Sports	28843.64
32	Southeast	Excellent Riding Supplies	80450.38
33	Southeast	Elemental Sporting Goods	6179.88
34	Southeast	Games and Sport Supply Company	4346.08
35	Southeast	Eighty Toy Stores	3460.70
36	Southeast	Superlative Bikes	3131.93
37	Southwest	Extraordinary Bike Works	67059.65
38	Southwest	Many Bikes Store	59894.20
39	Southwest	Small Bike Shop	40248.84
40	Southwest	Chic Department Stores	36550.85
41	Southwest	Paint Supply	31786.18
42	United Kingdom	Metropolitan Bicycle Supply	79589.61



42	United Kingdom	Metropolitan Bicycle Supply	79589.61
43	United Kingdom	Bulk Discount Store	74160.23
44	United Kingdom	Instruments and Parts Company	53248.70
45	United Kingdom	[Not Applicable]	1753.53
46	United Kingdom	Channel Outlet	524.66

REGION	TOP PERFORMER	RESELLER INSIGHTS
Australia	Cycle parts and accessories → $29K	Few strong partners and many low volume sellers
Canada	Workout emporium → $6.5K	Weak market, needs strategic expansion or increase marketing efforts
Central	Sleek bikes → $50K	Multiple healthy high value resellers, strong regional partnership network
France	NA → $1.6K	No strong reseller presence at all, market is likely undeveloped
Germany	Next door cycles → $57K	One major reseller and solid mid

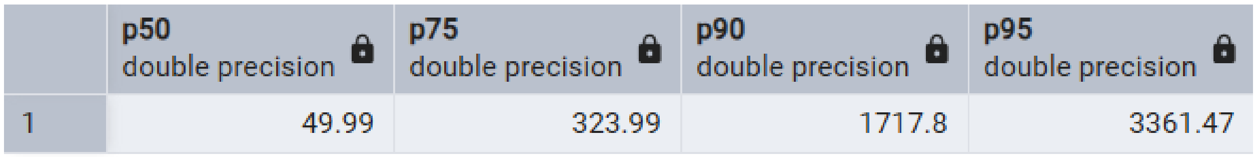
		tier partners, healthy moderate to strong market
NE	Glossy bikes → $70K	Huge value, region has multiple stable partners, a top performing territory!
NW	Rugged bikes → $57K	Very strong region with consistently high reseller performance
SE	Excellent riding supplies → $80K	One reseller dominates and the rest are much smaller, they only have one key partner they rely on, which may be risky!
SW	Extraordinary bike works → $67K	Excellent spread among resellers with multiple partners doing $30-67K
UK	Metropolitan bicycle supply → $79K	Three very strong partners, one of the most successful reseller regions

Question 4: Revenue percentiles

Code breakdown:

- PERCENTILE_CONT→ giving four different percentile arguments for continuous percentiles on sales amount
- WITHIN GROUP→ separating them into groups in output to show 50, 75, 90, and 95 percentiles
- ORDER BY→ sales amount alias as their percentiles

Analysis: the output tells us how sales amounts are distributed across all transactions for example, half of all sales amount are below $49.99, 75% of sales and under $323.99, 90% of sales are under $1717.80, and 95% of sales and under $3361.47. Analyzing this statistically, if our median is $49.99 and our 95th percentile is $3361.47 this is a huge gap and tells me that most of the customers spend very little while a few customers tend to splurge and contribute to the extremely high revenue amount. This is a useful business analysis as it helps us understand how much our customers are spending according to our entire dataset. We can use this insight to help us plan stock better for the low value items since they are bought more frequently and determine different pricing strategies as our high prices items are not selling as well.

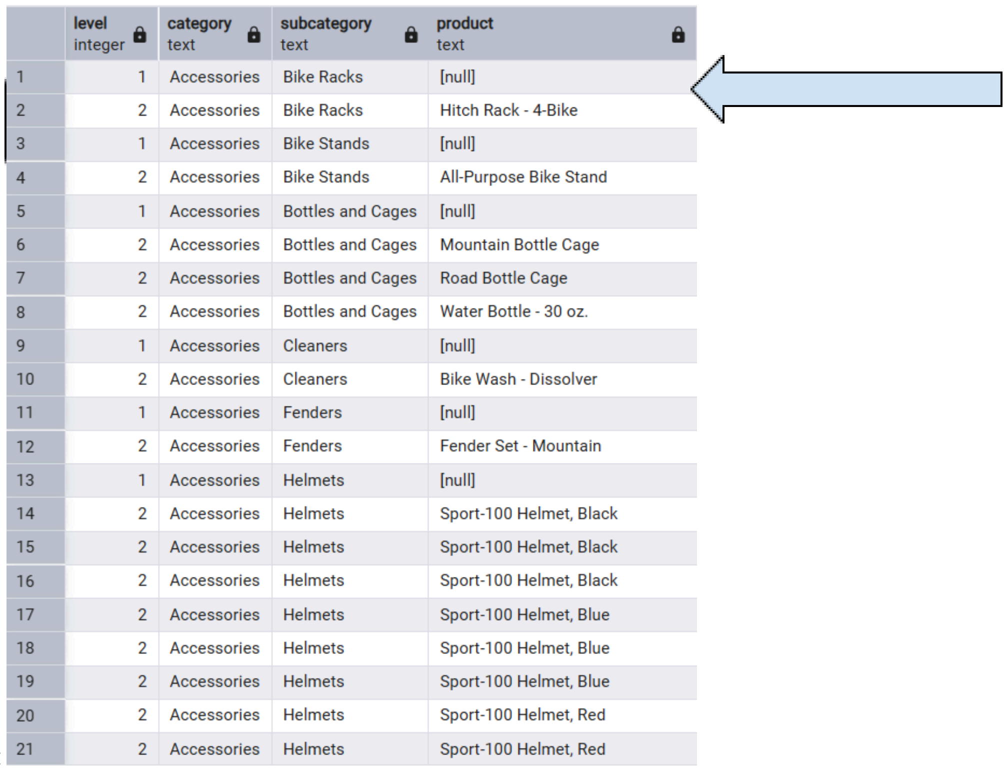


	p50 double precision	p75 double precision	p90 double precision	p95 double precision
1	49.99	323.99	1717.8	3361.47

Amber Locasto
SQL Final Project
Analysis Report

Question 5: Recursive CTE

The goal was to build a three level hierarchical structure from our product data with the category being level 1, subcategory being level 2, and product being level 3. The 1st level is the skeleton of our hierarchy level 2 adds each product to each subcategory making this the child. So our CTE should show category→ subcategory → product so for example below in our output we can see that a category bike racks is a level 1 and a product that belongs in the bike racks like hitch rack for 4 bikes product will be level 2.



	level integer	category text	subcategory text	product text
1	1	Accessories	Bike Racks	[null]
2	2	Accessories	Bike Racks	Hitch Rack - 4-Bike
3	1	Accessories	Bike Stands	[null]
4	2	Accessories	Bike Stands	All-Purpose Bike Stand
5	1	Accessories	Bottles and Cages	[null]
6	2	Accessories	Bottles and Cages	Mountain Bottle Cage
7	2	Accessories	Bottles and Cages	Road Bottle Cage
8	2	Accessories	Bottles and Cages	Water Bottle - 30 oz.
9	1	Accessories	Cleaners	[null]
10	2	Accessories	Cleaners	Bike Wash - Dissolver
11	1	Accessories	Fenders	[null]
12	2	Accessories	Fenders	Fender Set - Mountain
13	1	Accessories	Helmets	[null]
14	2	Accessories	Helmets	Sport-100 Helmet, Black
15	2	Accessories	Helmets	Sport-100 Helmet, Black
16	2	Accessories	Helmets	Sport-100 Helmet, Black
17	2	Accessories	Helmets	Sport-100 Helmet, Blue
18	2	Accessories	Helmets	Sport-100 Helmet, Blue
19	2	Accessories	Helmets	Sport-100 Helmet, Blue
20	2	Accessories	Helmets	Sport-100 Helmet, Red
21	2	Accessories	Helmets	Sport-100 Helmet, Red

:

Question 6: Explain Analyze plus index suggestions

INDEXES:

1. **CREATE INDEX idx_sales_productkey ON sales_data(ProductKey); → Any query that groups sales by product, subcategory or category, top 3 products per category, total sales per category and any product level aggregations or filters will be improved with this index. Instead of scanning all the rows in sales_data to find the matching ProductKeys it will use a tree look up which is mimicking a table of contents.**
2. **CREATE INDEX idx_sales_customerkey ON sales_data(CustomerKey); → improves top 10 customers by spending, average order value per customer, and any customer level segmentation. Allows us to jump to rows for a customer instead of scanning the entire fact table.**
3. **CREATE INDEX idx_sales_resellerkey ON sales_data(ResellerKey); → improves locating all sales that are tied to one reseller, region level reseller summaries and filtering or ranking resellers.**

Amber Locasto
SQL Final Project
Analysis Report

4. **CREATE INDEX idx_sales_orderdatekey ON sales_data(OrderDateKey); → we have a lot of date based queries and filters so this index will help improve monthly sales totals, cumulative monthly sales, latest year queries, time analytics and date filtering.**
5. **CREATE INDEX idx_sales_salesterritorykey ON sales_data(SalesTerritoryKey); → Made this index because we join sales territory data on a sales territory key so this will help improve sales my region, region level revenue analysis, any geographical analysis, and it will help the materialized views.**

Why did we use these indexes? → The table sales_data is the fact table which is huge and connects to every table. In a lot of my queries I used a join to this table specifically so I decided these indexes would be appropriate for improving the join speeds.

Q1 ANALYZE REPORT:

- What did Q1 do? → join sales_data with product_data, aggregate sales per product, sort by highest sales, and returns the products ranked by total sales
- Sorting shows no issues
- Aggregations and groupings are fast no issues
- Hash join = fastest join for this size table GOOD! This is telling me my keys are appropriate and no indexing issues I can find yet.
- Seq Scan spotted if we were using millions of rows of data like in the real world this may be an issue and we would want to suggest indexes

Index recommendations:

- CREATE INDEX inx_sales_productkey ON sales_data(ProductKey);
- CREATE INDEX inx_product_productkey ON product_data(ProductKey);
- Why? → seeing Seq Scan tells me that postgreSQL scanned entire tables during out join and although our data is relatively small, if it was larger this would be insufficient so we could create indexes in order to improve join performances using index look ups

Question 7: Creating a materialized view of monthly sales by region

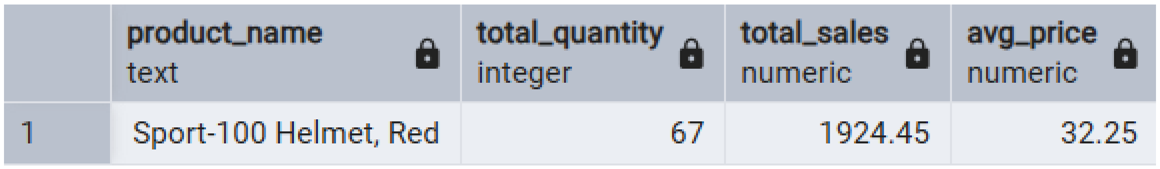
- I learned about the two types of views asked in our project requirements and I will break down the differences for each one and when to use each one below:

View type:	Storing data	speed	updates	storage	dashboard	Okay for frequent changes?
VIEW	NO	Slower	automatically	minimal	Not ideal	yes
MVIEW	YES	fast	manually	Great	Great	Refresh often

Amber Locasto
SQL Final Project
Analysis Report

Question 8: summarizing sales for a product and date range

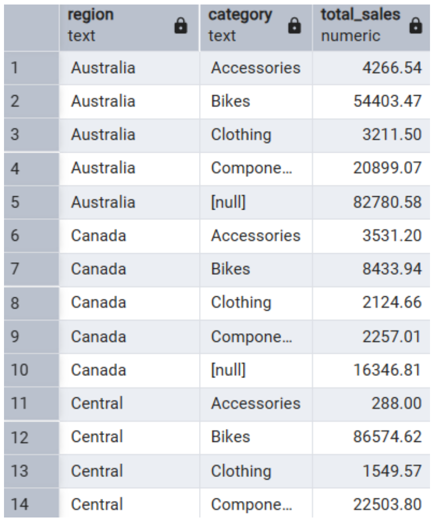
I enjoyed learning how to do this as it reminds me of writing a python function. Basically I am able to create a function and tell it to do what I want to return an output of the function parameters. For this question I created a function that takes a product name and returns the total quantity of the item, total sales of the item, and average price of the item between a certain date range you set in the parameters when you call the function. As an example I ran the function call with parameters set to filter product key 214, through the dates '2020-06-01', '2020-12-31'. Returning the following output below. Analyzing this we can see it returns a sports helmet red, with 67 total products, 1,924.45 sales and an average price of 32.25. So between June 1st 2020 and December 31st 2020, customers bought a total of 67 helmets with total sales at $1,924.45. This is important to analyze as a business outlook because we can see how a product is selling during a specific time of the year and determine if that product is doing well on our market or not.



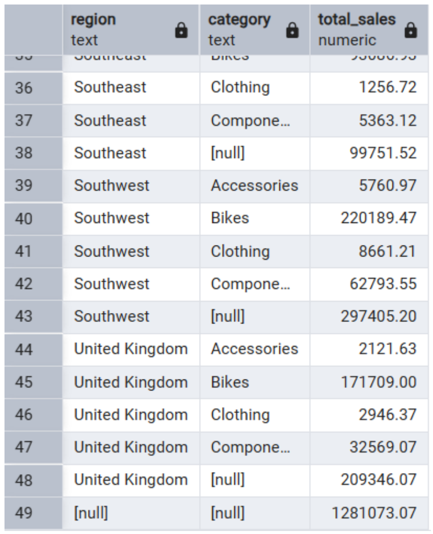
	product_name text	total_quantity integer	total_sales numeric	avg_price numeric
1	Sport-100 Helmet, Red	67	1924.45	32.25

Question 9: grouping sets and rollup for subtotals and grand totals

This query produces a hierarchy of subtotal rows by region, by category and a grand total for each region. Group by roll up (region, category) will return regular rows where we see total sales for each category within each region, subtotal rows where we see subtotals for all categories within that region and a grand total row where we see total sales for the entire dataset. Where you see the null after the regions are finished listing out, is the roll up total. I made a table to easily understand the rankings among total sales for the regions below:



	region text	category text	total_sales numeric
1	Australia	Accessories	4266.54
2	Australia	Bikes	54403.47
3	Australia	Clothing	3211.50
4	Australia	Compone...	20899.07
5	Australia	[null]	82780.58
6	Canada	Accessories	3531.20
7	Canada	Bikes	8433.94
8	Canada	Clothing	2124.66
9	Canada	Compone...	2257.01
10	Canada	[null]	16346.81
11	Central	Accessories	288.00
12	Central	Bikes	86574.62
13	Central	Clothing	1549.57
14	Central	Compone...	22503.80



	region text	category text	total_sales numeric
36	Southeast	Clothing	1256.72
37	Southeast	Compone...	5363.12
38	Southeast	[null]	99751.52
39	Southwest	Accessories	5760.97
40	Southwest	Bikes	220189.47
41	Southwest	Clothing	8661.21
42	Southwest	Compone...	62793.55
43	Southwest	[null]	297405.20
44	United Kingdom	Accessories	2121.63
45	United Kingdom	Bikes	171709.00
46	United Kingdom	Clothing	2946.37
47	United Kingdom	Compone...	32569.07
48	United Kingdom	[null]	209346.07
49	[null]	[null]	1281073.07

Table view for easier interpretation:

REGION	TOTAL SALES($)	Ranking
Australia	$82K	8
Canada	$16K	9

Amber Locasto
SQL Final Project
Analysis Report

Central	$110K	5
France	$1.6K	10
Germany	$104K	6
NE	$141K	4
NW	$217K	2
SE	$99K	7
SW	$297K	1
UK	$209K	3
GRAND TOTAL	$1,281,073.07	

Question 10: creating a read-only report_user with select only privileges

In the query we first create a new database user called BIalinlocasto with a password. This creates a user account I can use to log in with power BI like a real analyst would. Then we allow the user to connect to the correct database and then we grant access to our schema. We then give the user read only access to all current tables so we cannot modify, delete or insert data through our tools. We automatically grant select on all future tables so we can use them in the future.

POWER BI INFORMATION:

I created my first ever power BI dashboard as a visual for this project, page one is an executive summary which provides a high-level business overview for those who want quick insights into the company performance. This page mirrors my SQL queries and turns them into visuals. Page 1 serves as an overview page so we can answer important analysis questions like, how much money are we making, what products are bringing in the most revenue, which region performs better and how do the individual orders vary. Page 2 is a product and regional performance page, the purpose of this page is to dive deeper into performance patterns across product categories, regions, resellers, and product hierarchies. This page allows us to answer questions like which products dominate each category, which regions have the higher performing partners and what segments can we improve on. This mimics supply chain, marketing and product management. Our last page is a customer insights page that focuses on customer behavior and spending. This page allows us to dig deeper into understanding who our most valuable customers are, proportions of high spenders versus low spenders, how the customer behavior changes by region and where our marketing teams should focus. This analysis will help us with creating loyalty programs for qualified spenders and understand where our revenue is coming from to focus on inventory in that realm or better marketing strategies in our lower revenue zones.

Amber Locasto
SQL Final Project
Analysis Report