BICYCLE SALES

Exploring Bicycle Sales

The Bicycle Sales database from Microsoft SQL Server was obtained through DataLab (DataCamp). There are 2 schemas in the Bicycle Sales database, namely 'production' and 'sales'. The following lists the data tables found in each schema and their respective columns.

Schemas	Table Name	Table Column Names (Data Types)
production	brands	brand_id (int), brand_name (varchar)
production	categories	category_id (int), category_name (varchar)
production	products	product_id (int), product_name (varchar), brand_id (int), category_id (int), model_year (smallint), list_price (decimal)
production	stocks	store_id (int), product_id (int), quantity (int)
sales	customers	customer_id (int), first_name (varchar), last_name (varchar), phone (varchar), email (varchar), street (varchar), city (varchar), state (varchar), zip_code (varchar)
sales	order_items	order_id (int), item_id (int), product_id (int), quantity (int), list_price (decimal), discount (decimal)
sales	orders	order_id (int), customer_id (int), order_status (tinyint), order_date (date), required_date (date), shipped_date (date), store_id (int), staff_id (int)
sales	staffs	staff_id (int), first_name (varchar), last_name (varchar), email (varchar), phone (varchar), active (tinyint), store_id (int), manager_id (int)
sales	stores	store_id (int), store_name (varchar), phone (varchar), phone (varchar), email (varchar), street (varchar), city (varchar), state (varchar), zip_code (varchar)

Please refer to the 'Bicycle Sales Data Schema Diagram' PNG file for how the tables are connected.

The database was queried using MS SQL to answer the questions posed in the subsequent section.

Questions of Interests

Q1. How many orders are there in the dataset? What is the minimum, maximum and average revenue per order?

Answer: There were 1615 orders. The minimum revenue of an order was $104.49, the maximum revenue of an order was $29,147.03, and the average revenue per order was $4,781.06.

Q2A. How many orders were not shipped by the required date?

Answer: There were 458 orders that were not shipped by the required date.

Q2B. How many days were they late by?

Answer: There were 305 orders late by 1 day, while there were 153 orders late by 2 days.

Q2C. Did the late orders contain a large quantity of items?

Answer: There is no evidence that the late orders contained a larger quantity of items compared to orders shipped on time. The following table shows a comparison of the frequency and percentage of late order quantities against the on-time order quantities. The 'Percentage Late' was calculated by dividing the frequency of late orders by the sum of the frequency of on-time orders and the frequency of late orders.

Quantity in Order	Frequency (On-time)	Frequency (Late)	Percentage Late (%)
1	93	40	30.08
2	145	69	32.24
3	130	57	30.48
4	165	82	33.20
5	143	58	28.86
6	135	66	32.84
7	95	42	30.66
8	61	30	32.97
9	18	12	40.00
10	2	2	50.00

Note: The analyses for Q2 were done based on the available records that contained both 'required date' and 'shipped date'. There were 170 records with missing 'shipped date' in the original dataset.

Q3. How many staff were there in each store and how was the sales performance (measured by revenue)?

Answer: The following table lists the number of staff of each store and the overall sales performance.

Store ID	Store Name	Number of staff	Overall Performance ($)
2	Baldwin Bikes	3	5,215,751.28
1	Santa Cruz Bikes	4	1,605,823.04
3	Rowlett Bikes	3	867,542.24

Q4. Get the employee-manager relationship. Who is the top manager?

Answer: The table lists the employee-manager relationship. Fabiola Jackson is the top manager as Fabiola is the manager to other store managers/employee.

Staff ID	Staff Name	Staff's Store ID	Manager Name	Hierarchy Level
1	Fabiola Jackson	1	NA	0
2	Mireya Copeland	1	Fabiola Jackson	1
5	Jannette David	2	Fabiola Jackson	1
8	Kali Vargas	3	Fabiola Jackson	1
6	Marcelene Boyer	2	Jannette David	2
7	Venita Daniel	2	Jannette David	2
3	Genna Serrano	1	Mireya Copeland	2
4	Virgie Wiggins	1	Mireya Copeland	2
9	Layla Terrell	3	Venita Daniel	3
10	Bernardine Houston	3	Venita Daniel	3

Q5. What is the third most popular brand of bicycles?

Answer: The third most popular brand of bicycle over the years 2016 to 2018 was Surly, with an order quantity of 908.

Q6. What is the most popular brand in each bicycle category?

Answer: The table below lists the most popular brands in each bicycle category.

Category	Brand	Quantity Sold
Children Bicycles	Electra	747
Comfort Bicycles	Electra	524
Cruisers Bicycles	Electra	1329
Cyclocross Bicycles	Surly	305
Electric Bikes	Trek	269
Mountain Bikes	Trek	752
Road Bikes	Trek	482

SQL Code

Bicycle Sales DataFrame as d

```
-- Q1 How many orders are there? And what is the minimum, maximum and average revenue per order?
SELECT COUNT(DISTINCT oi.order_id) AS num_order, ROUND(MIN(oi.rev_per_order),2) AS min_rev, ROUND(MAX(oi.rev_per_order),2) AS
max_rev, ROUND(AVG(oi.rev_per_order),2) AS avg_rev_per_order
FROM (
    SELECT order_id, SUM(quantity * list_price * (1 - discount)) AS rev_per_order
    FROM sales.order_items
    GROUP BY order_id
) AS oi;
```

	n.			avg_rev_per_...	
0	1615	104.49	29147.03	4761.06	

Rows: 1 Expand

Bicycle Sales DataFrame as d

```
-- 170 records with missing date in shipped_date
SELECT COUNT(*) AS num_missing
FROM sales.orders
WHERE (shipped_date IS NULL)
```

	nu...	
0	170	

Rows: 1 Expand

Bicycle Sales DataFrame as d

```
-- Q2A How many orders were not shipped by the required date?
SELECT COUNT(*) AS num_late
FROM sales.orders
WHERE shipped_date > required_date;
```

0	458	

Rows: 1 Expand

Bicycle Sales DataFrame as d

```sql
-- Q2B How many days were they late by?
SELECT o.days_late, COUNT(*) AS freq
FROM (
    SELECT order_id, DATEDIFF(day, required_date, shipped_date) AS days_late
    FROM sales.orders
    WHERE shipped_date > required_date
) AS o
GROUP BY o.days_late;
```

	d...	...
0	1	305
1	2	153

Rows: 2 Expand

Bicycle Sales DataFrame as d

```sql
-- Q2C Did the late orders contain a large quantity of items?
CREATE TABLE #order_quan(
    order_id INT,
    total_quan INT
);

INSERT INTO #order_quan
    SELECT order_id, SUM(quantity) AS total_quan
    FROM sales.order_items
    GROUP BY order_id;

SELECT in_time.total_quan,
       in_time.freq_in_time,
       late.freq_late, ROUND(100.0*late.freq_late/(in_time.freq_in_time + late.freq_late),2) AS percent_late
FROM (
    SELECT oi_sum.total_quan, COUNT(*) AS freq_in_time
    FROM sales.orders AS o
    LEFT JOIN #order_quan AS oi_sum
    ON o.order_id = oi_sum.order_id
    WHERE o.shipped_date <= o.required_date
    GROUP BY oi_sum.total_quan
) AS in_time
LEFT JOIN (
    SELECT l_oi_sum.total_quan, COUNT(*) AS freq_late
    FROM sales.orders AS l_o
    LEFT JOIN #order_quan AS l_oi_sum
    ON l_o.order_id = l_oi_sum.order_id
    WHERE l_o.shipped_date > l_o.required_date
    GROUP BY l_oi_sum.total_quan
) AS late
ON in_time.total_quan = late.total_quan;

DROP TABLE #order_quan;
```

	t...	freq...	f...	perc...
0	1	93	40	30.08
1	2	145	69	32.24
2	3	130	57	30.48
3	4	165	82	33.2
4	5	143	58	28.86
5	6	135	66	32.84
6	7	95	42	30.66
7	8	61	30	32.97
8	9	18	12	40
9	10	2	2	50

Rows: 10 Expand

Bicycle Sales DataFrame as

```
-- Q3 How many staff were there in each store and how was the sales performance (measured by revenue)?
SELECT s.store_id, s.store_name, COUNT(DISTINCT st.staff_id) AS num_staff, os.store_rev
FROM sales.stores AS s
LEFT JOIN (
    SELECT o.store_id, ROUND(SUM(oi.rev_per_order),2) AS store_rev
    FROM sales.orders AS o
    LEFT JOIN (
        SELECT order_id, SUM(quantity*list_price*(1 - discount)) AS rev_per_order
        FROM sales.order_items
        GROUP BY order_id
    ) AS oi
    ON o.order_id = oi.order_id
    GROUP BY o.store_id
) AS os
ON s.store_id = os.store_id
LEFT JOIN sales.staffs AS st
ON s.store_id = st.store_id
GROUP BY s.store_id, s.store_name, os.store_rev
ORDER BY os.store_rev DESC;
```

		store_name	n...	st...
0	2	Baldwin Bikes	3	5215751.28
1	1	Santa Cruz Bikes	4	1605823.04
2	3	Rowlett Bikes	3	867542.24

Rows: 3 Expand

Bicycle Sales DataFrame as

```
-- Q4 Get the employee-manager relationship. Who is the top manager?

WITH getManager AS (
    SELECT st.staff_id, st.first_name, st.last_name, st.store_id, st.manager_id, mg.first_name AS manager_first_name,
mg.last_name AS manager_last_name, mg.store_id AS manager_store_id, 0 AS hierarchy_level
    FROM sales.staffs as st
    LEFT JOIN sales.staffs AS mg
    ON mg.staff_id = st.manager_id
    WHERE st.manager_id IS NULL

    UNION ALL

    SELECT st.staff_id, st.first_name, st.last_name, st.store_id, st.manager_id, mg.first_name, mg.last_name, mg.store_id,
mg.hierarchy_level + 1 AS hierarchy_level
    FROM sales.staffs as st
    INNER JOIN getManager AS mg
    ON mg.staff_id = st.manager_id
    WHERE mg.hierarchy_level < 5
)

SELECT *
FROM getManager
ORDER BY hierarchy_level ASC;
```

		fi...	l...		m...	manager_first_...	manager_last...	manager_st...
0	1	Fabiola	Jackson	1		null	null	
1	2	Mireya	Copeland	1	1	Fabiola	Jackson	
2	5	Jannette	David	2	1	Fabiola	Jackson	
3	8	Kali	Vargas	3	1	Fabiola	Jackson	
4	6	Marcelene	Boyer	2	5	Jannette	David	
5	7	Venita	Daniel	2	5	Jannette	David	
6	3	Genna	Serrano	1	2	Mireya	Copeland	
7	4	Virgie	Wiggins	1	2	Mireya	Copeland	
8	9	Layla	Terrell	3	7	Venita	Daniel	
9	10	Bernardine	Houston	3	7	Venita	Daniel	

Rows: 10 Expand

Bicycle Sales DataFrame as

```sql
-- Q5 What is the third most popular brand of bikes?
SELECT pb.brand_id, b.brand_name, pb.brand_total_quan
FROM production.brands AS b
RIGHT JOIN (
    SELECT p.brand_id, SUM(oi.total_quan) AS brand_total_quan, DENSE_RANK() OVER(ORDER BY SUM(oi.total_quan) DESC) AS rank
    FROM production.products AS p
    RIGHT JOIN (
        SELECT product_id, SUM(quantity) AS total_quan
        FROM sales.order_items
        GROUP BY product_id
    ) AS oi
    ON p.product_id = oi.product_id
    GROUP BY p.brand_id
) AS pb
ON pb.brand_id = b.brand_id
WHERE pb.rank = 3;
```

		b...	brand_total...
0	8	Surly	908

Rows: 1 Expand

Bicycle Sales DataFrame as

```sql
-- Q6 What is the most popular brand in each bike category?
SELECT c.category_name, b.brand_name, summ.total
FROM (
    SELECT p.category_id, p.brand_id, SUM(oi.total_quan) AS total, DENSE_RANK() OVER(PARTITION BY p.category_id ORDER BY
SUM(oi.total_quan) DESC) AS rank
    FROM production.products AS p
    RIGHT JOIN (
        SELECT product_id, SUM(quantity) AS total_quan
        FROM sales.order_items
        GROUP BY product_id
    ) AS oi
    ON p.product_id = oi.product_id
    GROUP BY p.category_id, p.brand_id
) AS summ
LEFT JOIN production.categories AS c
ON c.category_id = summ.category_id
LEFT JOIN production.brands AS b
ON b.brand_id = summ.brand_id
WHERE summ.rank = 1;
```

	category_name	b...	
0	Children Bicycles	Electra	747
1	Comfort Bicycles	Electra	524
2	Cruisers Bicycles	Electra	1329
3	Cyclocross Bicycles	Surly	305
4	Electric Bikes	Trek	269
5	Mountain Bikes	Trek	752
6	Road Bikes	Trek	482

Rows: 7 Expand