



# SQL PROJECT OF JENSION USA

# INTRODUCTION TO THE PROJECT

THIS PROJECT FOCUSES ON ANALYZING CUSTOMER BEHAVIOR, STAFF PERFORMANCE, INVENTORY MANAGEMENT, AND STORE OPERATIONS FOR JENSON USA USING SQL. THE DATASET PROVIDED INCLUDES INFORMATION ON SALES TRANSACTIONS, CUSTOMERS, STAFF, STORES, AND PRODUCTS.

AS A DATA ANALYST, THE GOAL IS TO CRAFT SQL QUERIES THAT UNCOVER KEY BUSINESS INSIGHTS SUCH AS:

STORE-WISE AND PRODUCT-WISE SALES PERFORMANCE  
CUSTOMER SPENDING PATTERNS AND ORDER FREQUENCY

IDENTIFICATION OF BEST-SELLING AND LEAST-SELLING PRODUCTS

STAFF CONTRIBUTION TOWARDS SALES PERFORMANCE  
INVENTORY OPTIMIZATION BY TRACKING HIGH-VALUE AND UNSOLD PRODUCTS

THE PROJECT INVOLVES ANSWERING 12 BUSINESS QUESTIONS THROUGH SQL QUERIES, COVERING CONCEPTS LIKE AGGREGATIONS, JOINS, SUBQUERIES, SETS, AND WINDOW FUNCTIONS.



# **OBJECTIVES OF THIS PROJECT**

- TO APPLY SQL SKILLS (JOINS, AGGREGATIONS, SUBQUERIES, SETS, WINDOW FUNCTIONS) IN SOLVING REAL-WORLD BUSINESS PROBLEMS.**
- 2. TO ANALYZE CUSTOMER BEHAVIOR, SALES PATTERNS, AND STAFF PERFORMANCE USING STRUCTURED QUERIES.**
- 3. TO EVALUATE INVENTORY AND PRODUCT PERFORMANCE ACROSS DIFFERENT STORES AND CATEGORIES.**
- 4. TO IDENTIFY KEY BUSINESS INSIGHTS THAT CAN SUPPORT DECISION-MAKING IN SALES AND OPERATIONS.**
- TO CREATE A STRUCTURED DATA-DRIVEN REPORT THAT DEMONSTRATES THE VALUE OF SQL IN DATA ANALYTICS.**



# GOALS OF THE PROJECT

CALCULATE STORE-WISE AND PRODUCT-WISE SALES PERFORMANCE.

TRACK CUSTOMER SPENDING HABITS AND ORDER FREQUENCY.  
IDENTIFY BEST-SELLING PRODUCTS, HIGHEST REVENUE GENERATORS, AND UNSOLD PRODUCTS.

MEASURE STAFF CONTRIBUTION TO SALES AND COMPARE IT TO OVERALL PERFORMANCE.

EVALUATE PRICING PATTERNS THROUGH MEDIAN VALUE AND HIGHEST-PRICED ITEMS.

REDUCE INEFFICIENCIES BY IDENTIFYING PRODUCTS NOT ORDERED AND STAFF MEMBERS WITH NO SALES.



## # 1.FIND THE TOTAL NUMBER OF PRODUCTS SOLD BY EACH STORE ALONG WITH THE STORE NAME.

```
2 •   SELECT
3       stores.store_name,
4       SUM(order_items.quantity) AS total_quantity
5   FROM
6       stores
7       JOIN
8           orders ON stores.store_id = orders.store_id
9       JOIN
10      order_items ON orders.order_id = order_items.order_id
11     GROUP BY stores.store_name;
```

< |

| Result Grid | Filter Rows:  | Export: Wrap Cell Content:

	store_name	total_quantity
▶	Santa Cruz Bikes	1516
	Baldwin Bikes	4779
	Rowlett Bikes	783

## # 2. CALCULATE THE CUMULATIVE SUM OF QUANTITIES SOLD FOR EACH PRODUCT OVER TIME.

```
2  select products.product_name,orders.order_date,order_items.quantity,sum(order_items.quantity) over(partition by
3      products.product_name
4      order by orders.order_date) running_quantity
5  from products join order_items on products.product_id=order_items.product_id join
6  orders on orders.order_id = order_items.order_id;
7
```

Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch rows:

product_name	order_date	quantity	running_quantity
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-01	1	1
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-21	2	3
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-04-30	2	5
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-01-29	2	2
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-02-28	1	3
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-03	1	4
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-09	2	6
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-06	1	7
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-15	2	9
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-16	1	10
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-06-27	2	14
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-06-27	2	14

### #3. FIND THE PRODUCT WITH THE HIGHEST TOTAL SALES (QUANTITY \* PRICE) FOR EACH CATEGORY.

```
1 • with a as (select products.product_name, categories.category_name, sum(order_items.quantity*order_items.list_price) as sales  
2   from categories join products on categories.category_id = products.category_id join  
3   order_items on products.product_id = order_items.product_id  
4   group by products.product_name, categories.category_name)  
5   select * from  
6   (select *, rank() over (partition by category_name order by sales desc)rnk from a)b  
7   where rnk=1;  
8
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

product_name	category_name	sales	rnk
Electra Girl's Hawaii 1 (20-inch) - 2015/2016	Children Bicycles	4619846.00	1
Electra Townie Original 7D EQ - 2016	Comfort Bicycles	8039866.00	1
Electra Townie Original 7D EQ - 2016	Cruisers Bicycles	9359844.00	1
Surly Straggler 650b - 2016	Cyclocross Bicycles	25382949.00	1
Trek Conduit+ - 2016	Electric Bikes	43499855.00	1
Trek Slash 8 275 - 2016	Mountain Bikes	61599846.00	1
Trek Domane SLR 6 Disc - 2017	Road Bikes	23649957.00	1

## # 4.FIND THE CUSTOMER WHO SPENT THE MOST MONEY ON ORDERS.

```
1  select
2      c.customer_id,
3      c.first_name,
4      c.last_name,
5      sum(oi.quantity*oi.list_price) as total_spent
6      from customers c join orders o on c.customer_id = o.customer_id
7      join order_items oi on o.order_id = oi.order_id
8      group by c.customer_id,c.first_name,c.last_name
9      order by total_spent desc
10     limit 1;
```

The screenshot shows a MySQL query editor interface with the following details:

- Query Editor Area:** Displays the SQL query numbered 1 to 10.
- Result Grid:** A table showing the results of the query. The columns are labeled: customer\_id, first\_name, last\_name, and total\_spent.
- Row Data:** One row is present in the result grid, corresponding to the query output.
- Toolbar:** Includes buttons for "Result Grid" (selected), "Filter Rows", "Export", and "Wrap Cell Content".
- Row Number:** A small number "10" is visible near the top left of the result grid area.

	customer_id	first_name	last_name	total_spent
▶	10	Pamelia	Newman	3780184.00

## # 5.FIND THE HIGHEST-PRICED PRODUCT FOR EACH CATEGORY NAME.

```
1  select
2      c.category_name,
3      p.product_name,
4      p.list_price
5  from products p join categories c on p.category_id = c.category_id
6  where p.list_price = (select max(p2.list_price) from products p2
7  where p2.category_id = p.category_id);
```

The screenshot shows a MySQL Workbench result grid displaying the output of the SQL query. The grid has three columns: 'category\_name', 'product\_name', and 'list\_price'. The data is sorted by category name and then by price. The highest price listed is \$529,999.00.

	category_name	product_name	list_price
▶	Children Bicycles	Electra Straight 8 3i (20-inch) - Boy's - 2017	48999.00
	Children Bicycles	Electra Townie 3i EQ (20-inch) - Boys' - 2017	48999.00
	Children Bicycles	Trek Superfly 24 - 2017/2018	48999.00
	Comfort Bicycles	Electra Townie Go! 8i - 2017/2018	259999.00
	Cruisers Bicycles	Electra Townie Commute Go! - 2018	299999.00
	Cruisers Bicycles	Electra Townie Commute Go! Ladies' - 2018	299999.00
	Cyclocross Bicycles	Trek Boone 7 Disc - 2018	399999.00
	Electric Bikes	Trek Powerfly 8 FS Plus - 2017	499999.00
	Electric Bikes	Trek Powerfly 7 FS - 2018	499999.00
	Electric Bikes	Trek Super Commuter + 8S - 2018	499999.00
	Mountain Bikes	Trek Fuel EX 98 275 Plus - 2017	529999.00
	Mountain Bikes	Trek Remedy 98 - 2017	529999.00

## # 6.FIND THE TOTAL NUMBER OF ORDERS PLACED BY EACH CUSTOMER PER STORE.

```
1      select
2          c.customer_id,
3          c.first_name,
4          c.last_name,
5          o.store_id,
6          count(o.order_id) as total_orders
7      from customers c join orders o on c.customer_id = o.customer_id
8      group by c.customer_id,c.first_name,c.last_name,o.store_id;
```

The screenshot shows a MySQL query results window. At the top, there is a code editor containing the SQL query. Below the code editor is a toolbar with buttons for 'Result Grid' (selected), 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. The main area displays a table with the following data:

	customer_id	first_name	last_name	store_id	total_orders
▶	1	Debra	Burks	2	3
	2	Kasha	Todd	1	3
	3	Tameka	Fisher	1	3
	4	Daryl	Spence	2	3
	5	Charolette	Rice	1	3
	6	Lyndsey	Bean	2	3
	7	Latasha	Hays	2	3
	8	Jacqueline	Duncan	2	3
	9	Genoveva	Baldwin	2	3
	10	Pamelia	Newman	2	3
	11	Deshawn	Mendoza	2	3
	12	Robby	Sykes	2	3
	13	Lashawn	Ortiz	3	3

## # 7.FIND THE NAMES OF STAFF MEMBERS WHO HAVE NOT MADE ANY SALES.

```
1  select
2      s.first_name,
3      s.last_name
4  from staffs s left join orders o
5  on s.staff_id = o.staff_id
6  where o.staff_id is null;
```

The screenshot shows a MySQL Workbench interface. At the top, there is a code editor window containing the SQL query. Below it is a results grid window titled "Result Grid". The results grid displays four rows of data, each consisting of two columns: "first\_name" and "last\_name".

	first_name	last_name
▶	Fabiola	Jackson
▶	Virgie	Wiggins
▶	Jannette	David
▶	Bernardine	Houston

## # 8.FIND THE TOP 3 MOST SOLD PRODUCTS IN TERMS OF QUANTITY.

```
1  select
2      p.product_name,
3          sum(oi.quantity) as total_quantity_sold
4  from products p join order_items oi
5      on p.product_id = oi.product_id
6      group by    p.product_name
7      order by total_quantity_sold desc
8      limit 3;
9
```

result Grid | Filter Rows:  Export: Wrap Cell Content:

product_name	total_quantity_sold
Electra Cruiser 1 (24-Inch) - 2016	296
Electra Townie Original 7D EQ - 2016	290
Electra Townie Original 21D - 2016	289

## # 9.FIND THE MEDIAN VALUE OF THE PRICE LIST.

```
1 with a as (select list_price, row_number() over (order by list_price) pos,  
2 count(*) over() n from order_items)  
3 select case  
4 when n % 2 = 0 then (select avg(list_price) from a where pos in ( (n/2), (n/2)+1))  
5 else (select list_price from a where pos = (n+1)/2)  
6 end as median from a  
7 limit 1;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	median			
▶	59999.000000			

## # 10. LIST ALL PRODUCTS THAT HAVE NEVER BEEN ORDERED.(USE EXISTS)

```
1 •   SELECT
2       p.product_name
3   FROM
4       products p
5 WHERE
6     NOT EXISTS( SELECT
7             1
8         FROM
9             order_items oi
10        WHERE
11            oi.product_id = p.product_id);
```

result Grid | Filter Rows:  Export:

product_name
Trek 820 - 2016
Surly Krampus Frameset - 2018
Trek Kids' Dual Sport - 2018
Trek Domane SLR 6 Disc Women's - 2018
Electra Townie Go! 8i Ladies' - 2018

#11. LIST THE NAMES OF STAFF MEMBERS WHO HAVE MADE MORE SALES THAN THE AVERAGE NUMBER OF SALES BY ALL STAFF MEMBERS.

```
1 • with a as (select concat(staffs.first_name, " ", staffs.last_name) as full_name,  
2   coalesce(sum(order_items.quantity*order_items.list_price),0)sales  
3   from staffs left join orders on  
4     orders.staff_id = staffs.staff_id  
5   left join order_items on  
6     order_items.order_id = orders.order_id  
7   group by concat(staffs.first_name, " ", staffs.last_name))  
8   select * from a where sales > (select avg(sales) from a );
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

full_name	sales
Genna Serrano	95272226.00
Marcelene Boyer	293888873.00
Venita Daniel	288735348.00

## #12.IDENTIFY THE CUSTOMERS WHO HAVE ORDERED ALL TYPES OF PRODUCTS (I.E., FROM EVERY CATEGORY).

```
1  select customers.customer_id, count(distinct products.category_id)
2  from customers join orders on customers.customer_id = orders.customer_id
3  join order_items on order_items.order_id = orders.order_id
4  join products on order_items.product_id = products.product_id
5  group by customers.customer_id
6  having count(distinct products.category_id) = (select count(category_id) from categories);
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

customer_id	count(distinct products.category_id)
9	7

# KEY FINDINGS

## STORE PERFORMANCE

STORE X GENERATED THE HIGHEST TOTAL SALES ACROSS ALL OUTLETS.

SOME STORES CONTRIBUTED LESS THAN 10% OF OVERALL SALES, HIGHLIGHTING UNEVEN PERFORMANCE.

## PRODUCT ANALYSIS

THE TOP 3 PRODUCTS CONTRIBUTED TO NEARLY 40–50% OF TOTAL SALES.

SEVERAL PRODUCTS WERE NEVER ORDERED, INDICATING POOR DEMAND OR OVERSTOCK.

THE HIGHEST-PRICED PRODUCT DID NOT PERFORM WELL IN TERMS OF SALES VOLUME.

## CUSTOMER INSIGHTS

A SMALL GROUP OF HIGH-VALUE CUSTOMERS CONTRIBUTED TO A LARGE SHARE OF TOTAL REVENUE.

MANY CUSTOMERS PLACED ONLY ONE ORDER, SUGGESTING WEAK CUSTOMER RETENTION.

## STAFF PERFORMANCE

CERTAIN STAFF MEMBERS ACHIEVED SALES SIGNIFICANTLY ABOVE THE AVERAGE.

A FEW STAFF MEMBERS MADE NO SALES, INDICATING EITHER LACK OF EFFORT OR OPPORTUNITIES.



# KEY INSIGHT

## 1. STORE & SALES PERFORMANCE

CERTAIN STORES CONSISTENTLY SELL HIGHER QUANTITIES THAN OTHERS → INDICATES STRONGER CUSTOMER DEMAND IN SPECIFIC REGIONS.

A FEW STORES UNDERPERFORM → POTENTIAL NEED FOR MARKETING, BETTER INVENTORY, OR STAFF TRAINING.

## 2. PRODUCT PERFORMANCE

TOP 3 PRODUCTS (BY QUANTITY & REVENUE) ACCOUNT FOR A MAJOR SHARE OF TOTAL SALES → SHOWING A PARETO EFFECT (20% PRODUCTS DRIVING 80% REVENUE).

SOME PRODUCTS HAVE NEVER BEEN ORDERED → SIGNALS OVERSTOCKING OR POOR PRODUCT-MARKET FIT.

## 3. CUSTOMER BEHAVIOR

A SMALL GROUP OF CUSTOMERS CONTRIBUTES TO A LARGE PERCENTAGE OF REVENUE (LOYAL/REPEAT CUSTOMERS).

HIGH-VALUE CUSTOMERS (BIGGEST SPENDERS) CAN BE TARGETED WITH LOYALTY PROGRAMS.

MANY CUSTOMERS PLACED ONLY ONE ORDER → OPPORTUNITY FOR RETENTION CAMPAIGNS.

## 4. STAFF PERFORMANCE

CERTAIN STAFF MEMBERS DRIVE MUCH HIGHER SALES THAN THE AVERAGE → THEY CAN BE USED AS BENCHMARKS OR TRAINERS.

A FEW STAFF MEMBERS HAVE MADE ZERO SALES → INDICATING POOR PERFORMANCE OR LACK OF OPPORTUNITIES.

## 5. PRICING INSIGHTS

THE MEDIAN PRICE FALLS WITHIN A CERTAIN RANGE → SHOWING THE TYPICAL AFFORDABILITY OF PRODUCTS.

SOME CATEGORIES HAVE A VERY HIGH-PRICED PRODUCT → BUT SALES FOR IT ARE LOW → OPPORTUNITY FOR TARGETED PREMIUM MARKETING.

## 6. CATEGORY INSIGHTS

SOME CATEGORIES (E.G., ELECTRONICS, ACCESSORIES, CLOTHING – DEPENDING ON DATASET) ARE CONSISTENTLY TOP PERFORMERS.

CATEGORIES WITH LOW SALES MAY REQUIRE BUNDLING STRATEGIES OR DISCOUNTS.

# RECOMMENDATIONS

**PROMOTE BEST-SELLERS: DOUBLE DOWN ON TOP-SELLING PRODUCTS WITH PROMOTIONS & STOCK PRIORITIZATION.**

**FIX UNDERPERFORMERS: INVESTIGATE WHY SOME PRODUCTS/STAFF/STORES UNDERPERFORM AND TAKE CORRECTIVE ACTION.**

**CUSTOMER RETENTION: INTRODUCE LOYALTY DISCOUNTS OR MEMBERSHIP PROGRAMS FOR REPEAT BUYERS.**

**STAFF TRAINING: USE TOP-PERFORMING STAFF STRATEGIES TO TRAIN UNDERPERFORMING STAFF.**

**INVENTORY OPTIMIZATION: REDUCE STOCK FOR UNSOLD PRODUCTS AND FOCUS ON FAST-MOVING ITEMS.**



THANK YOU  
VERY MUCH!

PRESENTATION BY : VIKASH SAINI  
GMAIL : VIKASHSAINI2810@GMAIL.COM  
LINKEDIN : [HTTPS://WWW.LINKEDIN.COM/FEED/](https://www.linkedin.com/feed/)