RETAIL INVENTORY & SALES MANAGEMENT DATABASE PROJECT

Tech : SQL

PROJECT OBJECTIVE

The objective of this project is to design and implement a relational database to:

- Manage product inventory
- Maintain category information
- Record sales transactions
- Ensure data integrity and validation
- Generate analytical business reports

DATABASE SCHEMA OVERVIEW

1. **Categories Table**
- Stores product category information.
- Category_Id – Primary Key
- CategoryName – Category description

	category_id [PK] integer	categoryname character varying (50)
1	101	Electronics
2	102	Mobile Devices
3	103	Audio Accessories
4	104	Televisions

2. **Product Table**
- Maintains product inventory details.
- Product_Id – Primary Key
- Product_Name
- Category_Id – Foreign Key → Categories.Category_Id
- Unit_Price
- Units_In_Stock

	product_id [PK] integer	product_name character varying (55)	category_id integer	unit_price numeric (10,2)	units_in_stock integer
1	2	Smartphone	102	800.00	100
2	3	Tablet	102	500.00	75
3	4	Desktop Computer	101	1500.00	30
4	6	Portable Speaker	103	150.00	150

3. **Sales Table**
- Stores product sales transactions.
- Sale_ID – Primary Key
- Product_Id – Foreign Key → Product.Product_Id
- Sale_Date
- Quantity_Sold

	sale_id [PK] integer	product_id integer	sale_date date	quantity_s integer
1	1	1	2024-01-01	
2	2	5	2024-01-02	
3	3	10	2024-01-03	
4	4	17	2024-01-04	

DATA INTEGRITY IMPLEMENTATION

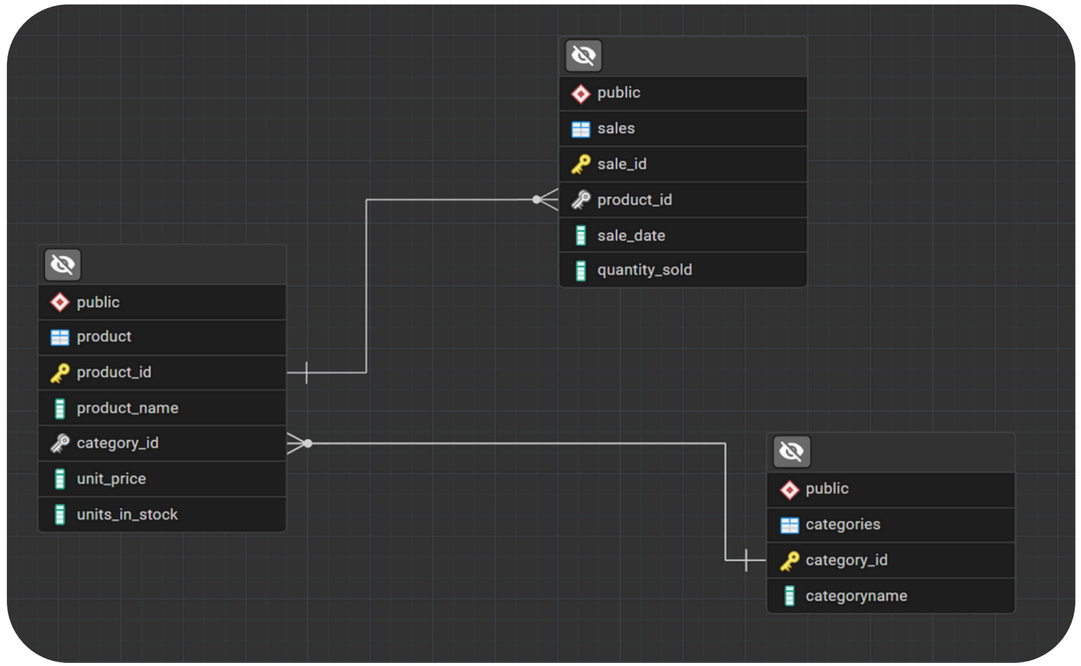
To maintain data consistency:

Foreign Key Constraints

- Product.Category_Id → Categories.Category_Id
- Sales.Product_Id → Product.Product_Id

This ensures:

- No product exists without a valid category
- No sale occurs for a non-existing product



DATA VALIDATION RULES

To ensure valid data entry:

Constraint	Purpose
Unit price must be positive	Prevents negative pricing
Units in stock must be non-negative	Avoid invalid inventory
Quantity sold must be ≥ 0	Prevents invalid sales
Foreign key validations	Prevents orphan records

These validations prevent incorrect, inconsistent, or illogical data from entering the database

BUSINESS REPORTS GENERATED

1) Total Sales by Product

Identifies best-selling products and revenue contribution.

	product_id [PK] integer	product_name character varying (55)	total_sales numeric
1	31	Patio Furniture Set	10400.00
2	1	Laptop	9600.00
3	79	Gaming Console	3200.00
4	22	Dining Table	2000.00

2) Total Sales by Category

Shows performance of each product category.

	[PK] integer	character varying (50)	numeric
1	115	Outdoor Living	10400.00
2	101	Electronics	9600.00
3	128	Gaming	3200.00
4	111	Furniture	2000.00

3) Current Inventory Status

Displays remaining stock of each product.

	product_id [PK] integer	product_name character varying (55)	units_in_stock integer
1	31	Patio Furniture Set	14
2	21	Bed	20
3	33	Treadmill	20
4	50	Projector	25
5	20	Sofa	25

4) Out of Stock Products

Helps in restocking and supply chain planning.

	product_id [PK] integer	product_name character varying (55)

TESTING & VALIDATION

To verify the robustness of the database:
✔ Test Cases Executed

- Insert negative price → **Rejected**
  “ insert into Product values (999, 'Test', 101, -10, 5) ; ”

- Insert invalid category → **Rejected**
  “ insert into Product values (998, 'Test', 999, 100, 5) ; ”

- Insert negative quantity sold → **Rejected**
  “ insert into Sales values (500, 1, CURRENT_DATE, -5) ; ”

- Insert valid sales entry → **Accepted**
  “ insert into Sales values (501, 1, CURRENT_DATE, 5) ; ”

This confirms database constraints and validations are working correctly.

CONCLUSION

This database successfully:

- Maintains structured retail data
- Ensures high data integrity
- Prevents invalid entries
- Supports inventory management
- Generates meaningful analytical reports

THANK YOU