BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

DATA WAREHOUSING

(SS G515)

PROJECT REPORT

Furniture Rental Company (Furlenco)

TEAM MEMBER NAME	ROLL NUMBER
ROHIT SINGHEE	2024H1120258P
SHREEDHAR SONI	2024H1120179P
MOHANEESH RAJ PRADHAN	2024H1120190P

SUBMITTED TO :

Dr. L. RAJYA LAKSHMI

CSIS DEPARTMENT

INDEX

PROBLEM DESCRIPTION

A furniture rental company seeks to optimize operations, profitability, and customer satisfaction by analyzing comprehensive data from rentals, inventory, customers, suppliers, and external factors. Key business challenges include tracking product performance, managing inventory efficiently, enhancing customer engagement, evaluating supplier effectiveness, and identifying reasons for product returns.

Leveraging collected data, the company aims to analyze:

1. **Profitability and Revenue Trends:** Evaluating income, discounts, and margins by product, category, and city.

2. **Inventory Health:** Monitoring inventory turnover, damaged goods, stock availability, and supplier inflows.

3. **Customer Insights:** Segmenting customers based on rental frequency, preferences, ratings, and churn rates.

4. **Returns Analysis:** Identifying frequently returned products, associated reasons, and related quality issues.

5. **Strategic Growth Opportunities:** Forecasting demand, analyzing year-over-year trends, and exploring new markets.

A star schema-based data warehouse has been designed, including dimension tables (Date, Month, City, Category, Product, Supplier, Customer, Return Reason), detailed fact tables (Rentals, Supplier Transactions, Product Returns, Inventory Snapshots), and aggregate tables for high-level business analytics. The goal is to generate actionable insights for strategic decision-making, enhance customer experiences, and drive profitable growth.

OVERVIEW

In the competitive and rapidly evolving furniture rental industry, leveraging data analytics is critical for operational excellence and enhanced customer experiences. Furniture rental operations involve multiple interrelated components such as products, customers, suppliers, rental transactions, inventory management, and geographic markets, each generating substantial and valuable data.

Systematic analysis of this data provides deep insights into product demand, customer preferences, supplier performance, inventory dynamics, and profitability trends. Effective data utilization allows the company to track high-performing products, manage inventory efficiently, reduce product returns, optimize supplier relationships, and better understand customer behavior and satisfaction.

A robust, star schema-based data warehouse facilitates this analytical capability by integrating diverse datasets into structured dimensions and facts. This architecture enables comprehensive, multidimensional analysis across various business aspects, empowering strategic decision-making.

Ultimately, through advanced data analytics, the furniture rental provider can significantly optimize operations, enhance customer satisfaction, drive profitability, and strategically position itself for sustained growth in a highly competitive market environment.

FURNITURE RENTAL DATA WAREHOUSE: SPECIFICATIONS

1. Purpose of the Data Warehouse:

- To serve as a centralized analytical repository for comprehensive data related to furniture rentals, inventory management, product returns, supplier performance, customer interactions, and profitability trends.

2. Stakeholders:

- Primary stakeholders include business management, inventory managers, suppliers, marketing analysts, customer relationship managers, financial analysts, and operations teams.

3. Key Performance Indicators (KPIs):

- Profitability metrics (net profit, revenue by category and city)
- Inventory turnover ratio and stock health
- Product return rates and reasons
- Customer satisfaction ratings and retention rates
- Supplier reliability and transaction volumes
- Impact of seasonal variations and promotions on rentals

4. Main Data Sources:

- Rental transaction systems
- Inventory management systems

- Supplier transaction databases
- Customer demographic and feedback systems
- Product return management databases
- External data sources (seasonality, festivals, promotions)

5. Granularity:

- Data stored at detailed granularity, including individual rental transactions, inventory snapshots by product and city, supplier transaction details, product return records, and specific customer interactions.

6. Analysis Types Supported:

- Supports detailed analysis across product categories (e.g., bedroom, living room furniture), customer segments (individual, corporate), geographical areas (cities), different seasonal trends, and reasons for product returns.

7. Reporting Requirements:

- Regular reporting on product and category performance, profitability analysis, detailed inventory and supplier reports, customer engagement insights, return analysis, and strategic growth opportunities. The system must also support flexible ad-hoc queries and multidimensional analysis.

8. Security Requirements:

- Strong access control mechanisms to restrict sensitive data access to authorized users only.

- Encryption standards for protecting data in transit and at rest.

- Comprehensive data backup and recovery strategies.

9. Data Integration Requirements:

- Seamless integration with rental transaction systems, inventory databases, supplier management systems, customer relationship platforms, and external data sources.

- Data cleansing, transformation, and standardization for consistency and accuracy.

10. Scalability Requirements:

- Scalable infrastructure to accommodate increasing volumes of rental transactions, customer interactions, and expanding product portfolios.

- Capability to handle growing analytical complexity, increased number of users, and evolving reporting requirements without performance issues.

REQUIREMENTS IDENTIFIED

The following requirements were identified to support the analytical and operational objectives of the furniture rental data warehouse project:

1. Business Requirements

- Monitor and analyze profitability across products, categories, and cities.
- Optimize inventory levels to prevent stock-outs, manage damaged goods, and minimize storage costs.
- Enhance customer engagement by analyzing preferences, satisfaction ratings, rental patterns, and churn rates.
- Evaluate supplier reliability, procurement costs, and overall supply-chain effectiveness.
- Identify products frequently associated with returns to reduce losses and improve product quality.

2. Data Requirements

- Rental transaction data (quantity rented, amount paid, discounts, ratings).
- Detailed inventory data (stock availability, reserved and damaged units).
- Supplier transaction details (quantity supplied, transaction costs, delivery timelines).
- Comprehensive customer demographics and rental history.
- Product return information (returned quantities, reasons, and frequencies).
- Temporal and contextual data (dates, months, festival seasons, weekends).

3. Functional Requirements

- Ability to generate periodic (daily, monthly, yearly) and ad-hoc reports.
- Support for multidimensional analysis (e.g., drill-down by city, product category, time).

- Capability to measure and report key performance indicators (KPIs) such as inventory turnover, net profitability, return rates, and customer satisfaction scores.

4. Technical Requirements

- Robust star schema design integrating fact tables (Rentals, Inventory Snapshots, Supplier Transactions, Product Returns) and dimension tables (Date, Month, City, Product, Category, Supplier, Customer, Return Reason).
- Creation of aggregate tables for quicker access to summarized data (Inventory Turnover, Product Performance, Profitability, Return Analysis).
- Seamless integration, cleansing, and transformation of data from various transactional and operational systems into the data warehouse.

5. Security and Privacy Requirements

- Controlled access to sensitive data through strict authentication and authorization procedures.
- Data encryption at rest and in transit.
- Regular backups and disaster recovery measures to ensure data availability and integrity.

6. Scalability and Performance Requirements

- Infrastructure capable of scaling with the business growth and increasing data volumes.
- Efficient query processing to handle complex and multidimensional analysis without performance degradation.

Meeting these requirements ensures the furniture rental company has a reliable, secure, and robust data warehouse capable of delivering actionable business intelligence to stakeholders.

INFORMATION PACKAGE DIAGRAM

INFORMATION PACKAGE DIAGRAM FOR RENTALS FACT TABLE

DATE	PRODUCT	CUSTOMER	CITY
Year	product_id	customer_id	city_id
Month	category_name	customer_name	country
Day	product_name	customer_type	state
Hour	rental_price	gender	city_name
Minute		age	
MEASURED FACTS	quantity, total_amount, discount_amount, rating		

INFORMATION PACKAGE DIAGRAM FOR PRODUCT RETURNS FACT TABLE

DATE	PRODUCT	CUSTOMER	CITY	RETURN REASON
Year	product_id	customer_id	city_id	reason_id
Month	category_name	customer_type	country	reason_name
Day	product_name		state	
Hour	rental_price		city_name	
Minute				
MEASURED FACTS	quantity_returned			

INFORMATION PACKAGE DIAGRAM FOR SUPPLIER TRANSACTIONS FACT TABLE

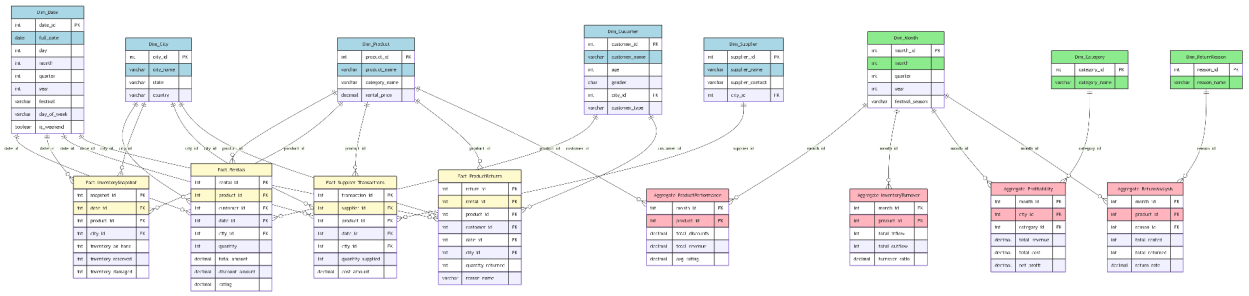
DATE	SUPPLIER	PRODUCT	CITY
Year	supplier_id	product_id	city_id
Month	supplier_name	category_name	country
Day	supplier_contact	product_name	state
Hour		rental_price	city_name
Minute			
MEASURED FACTS	quantity_supplied, cost_amount		

INFORMATION PACKAGE DIAGRAM FOR INVENTORY SNAPSHOT FACT TABLE

DATE	PRODUCT	CITY
Year	product_id	city_id
Month	category_name	country
Day	product_name	state
Hour	rental_price	city_name
Minute		
MEASURED FACTS	inventory_on_hand, inventory_reserved, inventory_damaged	

STAR SCHEMA

Diagram



[Star Schema Diagram Link](#) (Please use the link if above diagram is unclear)

Color	Table Type	Count
Yellow	Fact Table	4
Red	Aggregate Table	4
Blue	Dimension Table	5
Green	Derived Dimension	3

Dimension Tables

1. **Dim_Date**

Stores date-related information for time-based analysis.

- **date_id**(PK): Surrogate key for the date dimension.
- full_date: Full representation of the date (e.g., '2025-04-20').
- day: Day of the month.
- month: Month number or name.
- year: Year part of the date.
- quarter: Quarter of the year.
- festival: Signifies festival that might be going on.
- day_of_week: .signifies which day of the week it is.
- is_weekend: Boolean indicating if the date is a weekend.

2. **Dim_City**

Contains details about cities where business operates.

- **city_id**(PK): Surrogate key for the city.

- city_name: Name of the city.
- country: Country of the city.
- state: State or region of the city.

3. **Dim_Product**

Describes each product available for rental.

- **product_id**(PK): Surrogate key for the product.
- product_name: Name of the product.
- category_name: Category to which the product belongs.
- rental_price: Price for renting the product.

4. **Dim_Customer**

Holds customer demographic and type information.

- **customer_id**(PK): Surrogate key for the customer.
- customer_name: Name of the customer.
- age: Age of the customer.
- gender: Gender of the customer.
- city_id: Foreign key referencing the city of the customer.
- customer_type: Type of customer (e.g., Individual, Corporate).

5. **Dim_Supplier**

Stores supplier details for procurement analysis.

- **supplier_id**(PK): Surrogate key for the supplier.
- supplier_name: Name of the supplier.
- supplier_contact: Contact details for the supplier.
- city_id: Foreign key referencing the city of the supplier.

6. **Dim_Month**

Contains information about each month for use in time-based aggregation

- **month_id**(PK): Unique identifier for each month record in the table.
- month: Calendar month number (1–12).
- year: Calendar year corresponding to the month.
- quarter: Quarter of the year in which the month falls (1–4).
- festival_season: denotes any the kind of festive season going on

7. **Dim_Category**

Describes categories for grouping products.

- **category_id**(PK): Surrogate key for the category.

- category_name: Name of the category.

8. **Dim_RentalReason**

Lists possible reasons for renting furniture.

- **reason_id**(PK): Surrogate key for the rental reason.
- reason_name: Description of the rental reason.

Fact Tables

9. **Fact_InventorySnapshot**

Records periodic snapshots of inventory status at cities.

- snapshot_id: Surrogate key for the inventory snapshot.
- date_id: Foreign key to the date.
- product_id: Foreign key to the product.
- city_id: Foreign key to the city.
- inventory_on_hand: Quantity of product available.
- inventory_inflow: Quantity of product received.
- inventory_damaged: Quantity of damaged product.

10. **Fact_Rentals**

Captures details of each rental transaction.

- rental_id: Surrogate key for the rental.
- date_id: Foreign key to the date.
- product_id: Foreign key to the product.
- customer_id: Foreign key to the customer.
- city_id: Foreign key to the city.
- quantity: Number of products rented.
- total_amount: Total rental amount.
- discount_amount: Discount applied to the rental.
- rating: Customer rating for the rental.

11. **Fact_SupplierTransactions**

Tracks transactions between the company and suppliers.

- transaction_id: Surrogate key for the transaction.
- supplier_id: Foreign key to the supplier.
- product_id: Foreign key to the product.
- date_id: Foreign key to the date.
- city_id: Foreign key to the city.

- quantity_supplied: Quantity supplied in the transaction.
- cost_amount: Total amount of the transaction.

12. **Fact_ProductReturns**

Records details of product returns within the furniture rental business.

- return_id: Unique identifier for each product return event.
- rental_id: Links the return to the original rental transaction.
- product_id: Identifies the product being returned.
- customer_id: Identifies the customer making the return.
- date_id: Specifies the date the return was processed.
- city_id: Indicates the location where the return took place.
- quantity_returned: The number of units returned in this record.
- reason_name: States the reason provided for the return.

Aggregate Tables

13. **Aggregate_ProductPerformance**

Stores aggregated product performance metrics monthly.

- month_id: Surrogate key for the month.
- product_id: Foreign key to the product.
- total_discounts: Total discounts given for the products.
- total_revenue: Total revenue of the products.
- avg_rating: Average customer rating for the product.

14. **Aggregate_InventoryTurnover**

Contains monthly aggregated inventory data per product and branch.

- month_id: Surrogate key for the month.
- product_id: Foreign key to the product.
- turnover_ratio: Turnover ratio of the inventory.
- total_inflow: Total inventory received during the month.
- total_outflow: Total inventory rented out during the month.

15. **Aggregate_Profitability**

Tracks profitability by category and month.

- month_id: Surrogate key for the month.
- city_id: Foreign key to the cities.

- category_id: Foreign key to the product category.
- total_revenue: Total revenue generated for the category in the month.
- total_cost: Total cost for the category in the month.
- total_profit: Total profit for the category in the month.

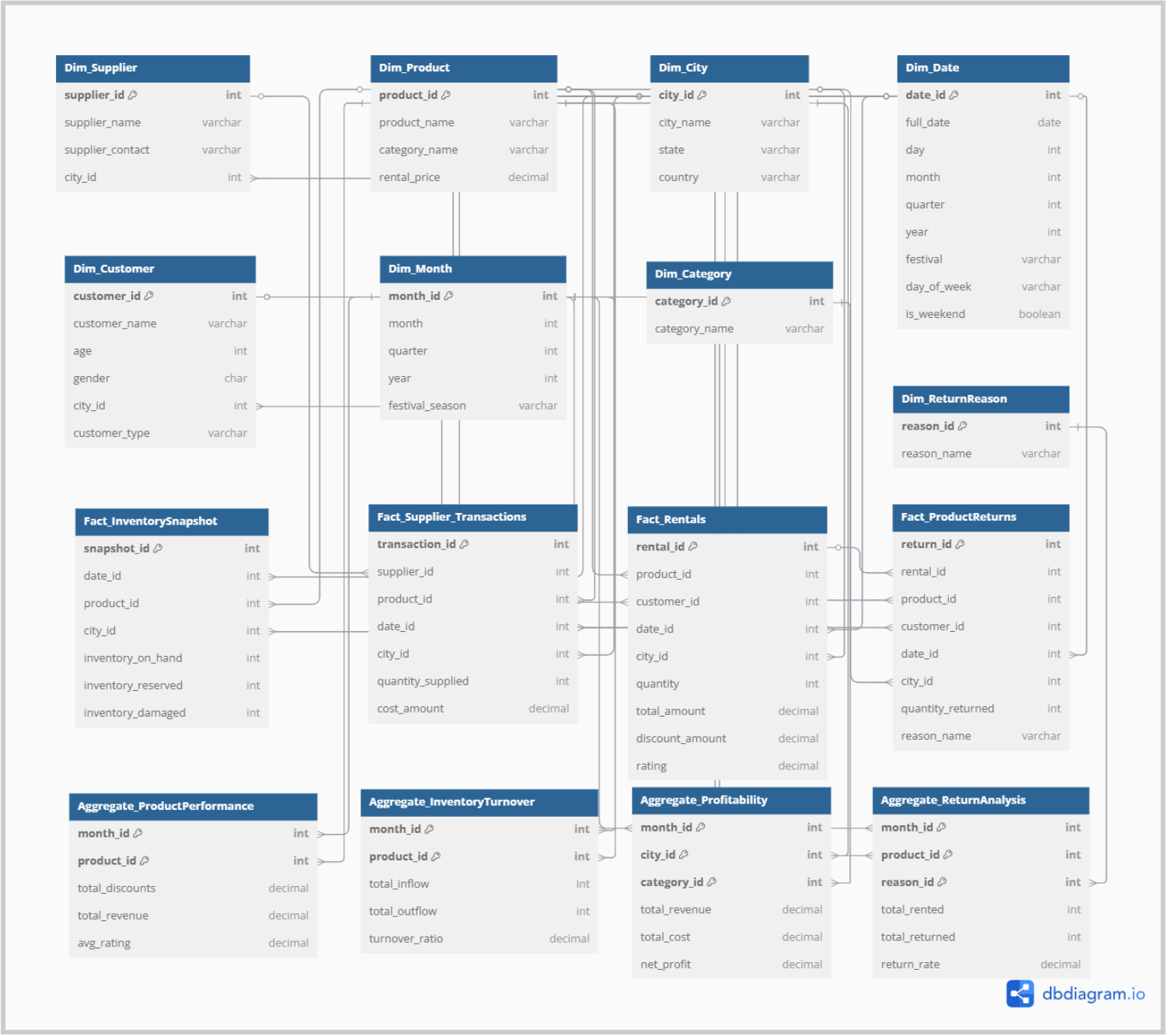
16. **Aggregate_ReturnsAnalysis**

Analyzes returns and rental reasons monthly per product.

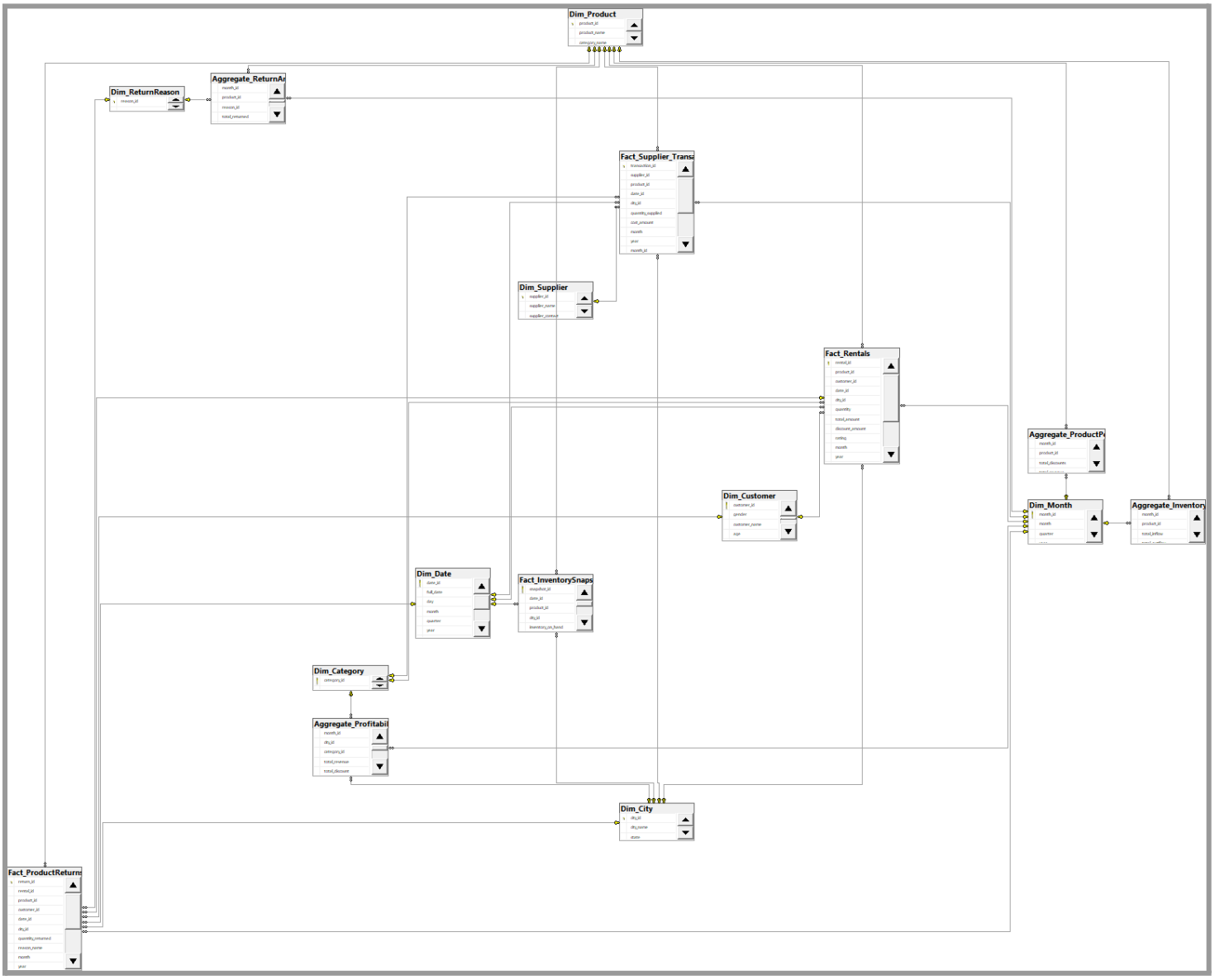
- month_id: Surrogate key for the month.
- product_id: Foreign key to the product.
- reason_id: Foreign key to the rental reason.
- total_rented: Total units rented.
- total_returned: Total units returned.
- return_rate: Rate of return of the products.

ENTITY RELATIONSHIP DIAGRAM OF THE STAR SCHEMA

[Live ER Diagram Link]



DATABASE DIAGRAM



REASON FOR DESIGN CHOICES

This section outlines the design decisions behind the furniture rental data warehouse. Following Kimball's dimensional modeling principles, the architecture is crafted for performance, scalability, and user-friendly analytics.

1. Architectural Approach and Methodology

Dimensional Modeling (Kimball Method):
A bottom-up approach using star schemas was adopted to enable intuitive analytics and support self-service BI. This simplifies ad hoc querying by minimizing complex joins.

Conformed Dimensions:
Shared dimensions—**Date**, **Product**, and **City**—are reused across all fact tables to ensure consistent filtering and seamless drill-across analysis.

Surrogate Keys:
Integer-based surrogate keys (e.g., `date_id`, `product_id`) decouple warehouse timelines from source systems, enable historical tracking, and improve join performance.

---

2. Granularity and Fact Table Grain

Fact Table Design:
Each fact table captures a distinct business event in the rental lifecycle:

- **Fact_Rentals**: One record per product rented per customer per day.
- **Fact_Supplier_Transactions**: One record per product per supplier delivery per date.
- **Fact_InventorySnapshot**: Daily snapshot per product per location.
- **Fact_ProductReturns**: One record per returned unit linked to the original rental.

Design Rationale:

- **Business Process Alignment**: Fact tables reflect real-world processes—rentals, deliveries, inventory, and returns.
- **Analytical Completeness**: Enables full lifecycle visibility across revenue, cost, operations, and customer behavior.
- **Performance Isolation**: Separate workloads allow for targeted indexing and tuning per fact table.

- **Fine-Grained Granularity**: Supports detailed drill-downs, time-series trends, and accurate KPIs (e.g., return rates, rental counts).

---

3. Dimension Modeling and SCD Handling

Slowly Changing Dimensions (SCDs):

- **Type 2 SCDs** for **Dim_Product**, **Dim_Supplier**, and **Dim_Customer** to retain historical attribute changes (e.g., price changes, customer status).
- **Type 1 SCDs** for **Dim_City** and **Dim_Category** where historical changes are not critical.

---

4. Conformed and Derived Dimensions

Derived Dimensions:

- **Dim_Month**: Aggregates daily data into monthly/quarterly periods for simplified trend analysis and period-to-date metrics.
- **Dim_Category**: Captures hierarchical SKU structure, enabling roll-ups and category-level reporting.
- **Dim_ReturnReason**: Centralizes standardized return codes for root-cause analysis.

Benefits of Derived Dimensions:

- **Query Simplification**: Reduces on-the-fly logic like date formatting or category grouping.
- **Performance Gains**: Precomputed attributes (e.g., festival season) lower runtime transformation costs.
- **Consistency**: Centralized derived logic ensures uniform reporting and reduces risk of discrepancies.

---

5. Aggregates and Performance Optimization

Aggregate Tables:

- **Aggregate_ProductPerformance**: Monthly KPIs like revenue, discounts, ratings.
- **Aggregate_InventoryTurnover**: Measures inventory flow and turnover ratios.

- **Aggregate_Profitability**: Summarizes profit by city and category.
- **Aggregate_ReturnAnalysis**: Tracks return rates by product and reason.

Purpose:
Accelerate common analytics by avoiding full-table scans and enabling rapid access to summarized metrics.

6. Data Integrity, Governance, and Scalability

- **Referential Integrity:** Enforced via primary and foreign key constraints.
- **ETL Design:** Staging layers handle cleansing, deduplication, and SCD logic before loading.
- **Scalability:** Modular star schemas support future additions (e.g., maintenance logs) without redesign.
- **Metadata & Auditing:** ETL captures load timestamps and lineage for traceability and governance.

---

7. Summary

This design—rooted in dimensional modeling best practices—ensures:

- Clear grain definitions and surrogate key usage.
- Historical accuracy with appropriate SCD types.
- Performance via aggregation, indexing, and partitioning.
- Flexibility to evolve with business needs.

Together, these choices deliver a robust, performant, and scalable data warehouse that supports cross-functional analytics and business growth.

.

ANALYSIS AND RESULTS

Query Categories from a Stakeholder Perspective:

To support data-driven decision-making across various business functions, the following multidimensional query categories are defined. Each category aligns with a specific stakeholder objective:

1. Customer & Marketing (Objective: Business Expansion and Customer Satisfaction)

2. Profit & Revenue (Objective: Profit Maximization)

3. Inventory & Supply Chain (Objective: Loss Minimization and Logistics Efficiency)

4. Product & Returns (Objective: Quality Improvement and Loss Reduction)

5. Growth & Forecast (Objective: Strategic Planning and Market Expansion)

1. Customer & Marketing (Objective: Business Expansion and Customer Satisfaction)

Goal: Understand customer behavior, improve satisfaction, segment loyalty, reduce churn, and identify upsell/cross-sell opportunities.

Query 1: Top 5 Cities by Total Active Customers

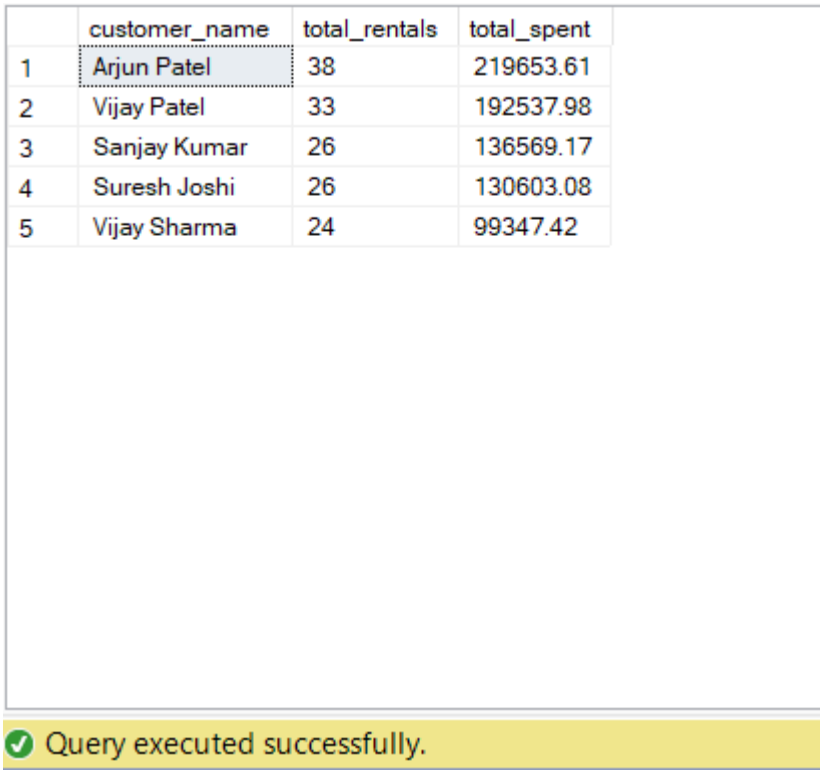
```
SELECT TOP 5
   c.city_name,
   COUNT(DISTINCT r.customer_id) AS total_active_customers
FROM Fact_Rentals r
JOIN Dim_City c ON r.city_id = c.city_id
GROUP BY c.city_name
ORDER BY total_active_customers DESC;
```

	city_name	total_active_customers
1	Bangalore	83
2	Hyderabad	77
3	Mumbai	76
4	Ahmedabad	76
5	Delhi	76

Query executed successfully.

Query 2: Top 5 Most Loyal Customers (Highest Rentals)

```
SELECT TOP 5
   c.customer_name,
   COUNT(r.rental_id) AS total_rentals,
   SUM(r.total_amount) AS total_spent
FROM Fact_Rentals r
JOIN Dim_Customer c ON r.customer_id = c.customer_id
GROUP BY c.customer_name
ORDER BY total_rentals DESC;
```



	customer_name	total_rentals	total_spent
1	Arjun Patel	38	219653.61
2	Vijay Patel	33	192537.98
3	Sanjay Kumar	26	136569.17
4	Suresh Joshi	26	130603.08
5	Vijay Sharma	24	99347.42

Query executed successfully.

Query 3: Average Rating by City and Product Category

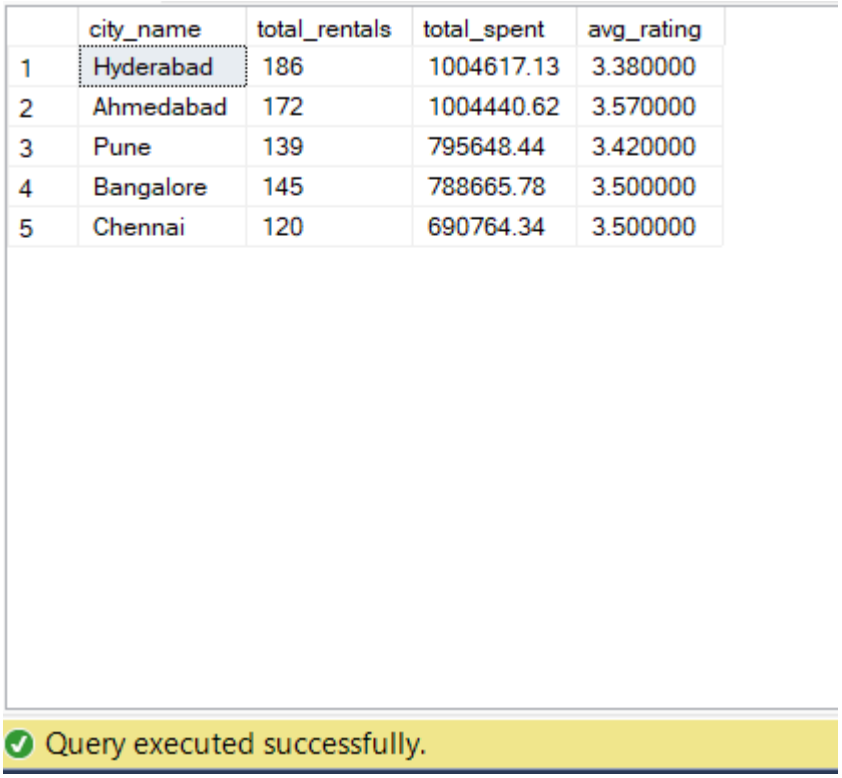
```
SELECT
   ci.city_name,
   cat.category_name,
   ROUND(AVG(r.rating), 2) AS avg_rating
FROM Fact_Rentals r
JOIN Dim_Customer cu ON r.customer_id = cu.customer_id
JOIN Dim_City ci ON r.city_id = ci.city_id
JOIN Dim_Product p ON r.product_id = p.product_id
JOIN Dim_Category cat ON p.category_name = cat.category_name
GROUP BY ci.city_name, cat.category_name
ORDER BY avg_rating DESC;
```

	city_name	category_name	avg_rating
1	Delhi	Mattresses & Bedding	4.280000
2	Delhi	Office Furniture	4.170000
3	Mumbai	Bedroom Furniture	3.930000
4	Pune	Kids Furniture	3.910000
5	Hyderabad	Home Office Essentials	3.850000
6	Mumbai	Kids Furniture	3.830000
7	Hyderabad	Outdoor Furniture	3.830000
8	Chennai	Dining Furniture	3.760000
9	Delhi	Bedroom Furniture	3.740000
10	Chennai	Mattresses & Bedding	3.740000
11	Delhi	Storage & Shelving	3.710000
12	Hyderabad	Office Furniture	3.670000
13	Ahmedabad	Kids Furniture	3.650000
14	Mumbai	Office Furniture	3.630000
15	Pune	Outdoor Furniture	3.630000
16	Hyderabad	Kids Furniture	3.590000
17	Hyderabad	Appliance Rentals	3.590000

Query executed successfully.

Query 4: Top Cities with Highest Customer Lifetime Value

```
SELECT TOP 5
    ci.city_name,
    COUNT(r.rental_id) AS total_rentals,
    SUM(r.total_amount) AS total_spent,
    ROUND(AVG(r.rating), 2) AS avg_rating
FROM Fact_Rentals r
JOIN Dim_Customer c ON r.customer_id = c.customer_id
JOIN Dim_City ci ON c.city_id = ci.city_id
GROUP BY ci.city_name
ORDER BY total_spent DESC;
```

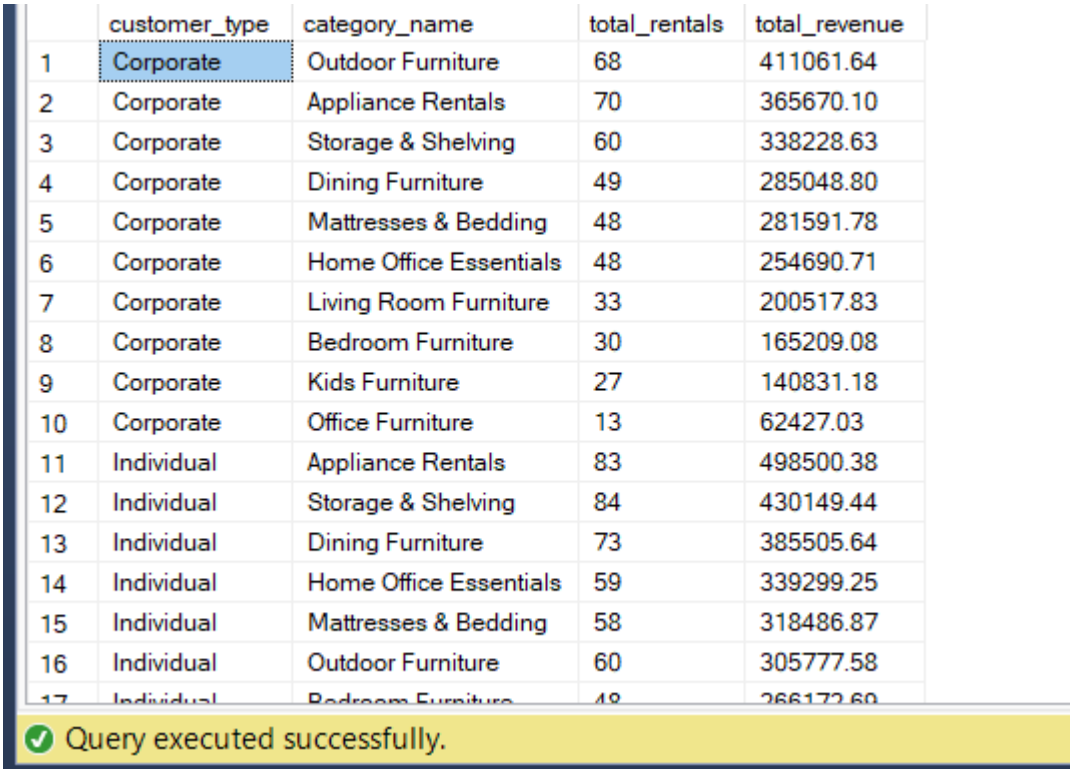


	city_name	total_rentals	total_spent	avg_rating
1	Hyderabad	186	1004617.13	3.380000
2	Ahmedabad	172	1004440.62	3.570000
3	Pune	139	795648.44	3.420000
4	Bangalore	145	788665.78	3.500000
5	Chennai	120	690764.34	3.500000

Query executed successfully.

Query 5: Popular Product Categories by Customer Segment

```
SELECT
    c.customer_type,
    p.category_name,
    COUNT(r.rental_id) AS total_rentals,
    ROUND(SUM(r.total_amount), 2) AS total_revenue
FROM Fact_Rentals r
JOIN Dim_Customer c ON r.customer_id = c.customer_id
JOIN Dim_Product p ON r.product_id = p.product_id
GROUP BY c.customer_type, p.category_name
ORDER BY c.customer_type, total_revenue DESC;
```



	customer_type	category_name	total_rentals	total_revenue
1	Corporate	Outdoor Furniture	68	411061.64
2	Corporate	Appliance Rentals	70	365670.10
3	Corporate	Storage & Shelving	60	338228.63
4	Corporate	Dining Furniture	49	285048.80
5	Corporate	Mattresses & Bedding	48	281591.78
6	Corporate	Home Office Essentials	48	254690.71
7	Corporate	Living Room Furniture	33	200517.83
8	Corporate	Bedroom Furniture	30	165209.08
9	Corporate	Kids Furniture	27	140831.18
10	Corporate	Office Furniture	13	62427.03
11	Individual	Appliance Rentals	83	498500.38
12	Individual	Storage & Shelving	84	430149.44
13	Individual	Dining Furniture	73	385505.64
14	Individual	Home Office Essentials	59	339299.25
15	Individual	Mattresses & Bedding	58	318486.87
16	Individual	Outdoor Furniture	60	305777.58
17	Individual	Bedroom Furniture	48	266172.69

Query executed successfully.

2. **Profit & Revenue (Objective: Profit Maximization)**

Understand revenue drivers, margins, discount patterns, and rating correlations.

Query 6: Monthly Profitability Breakdown by City & Category

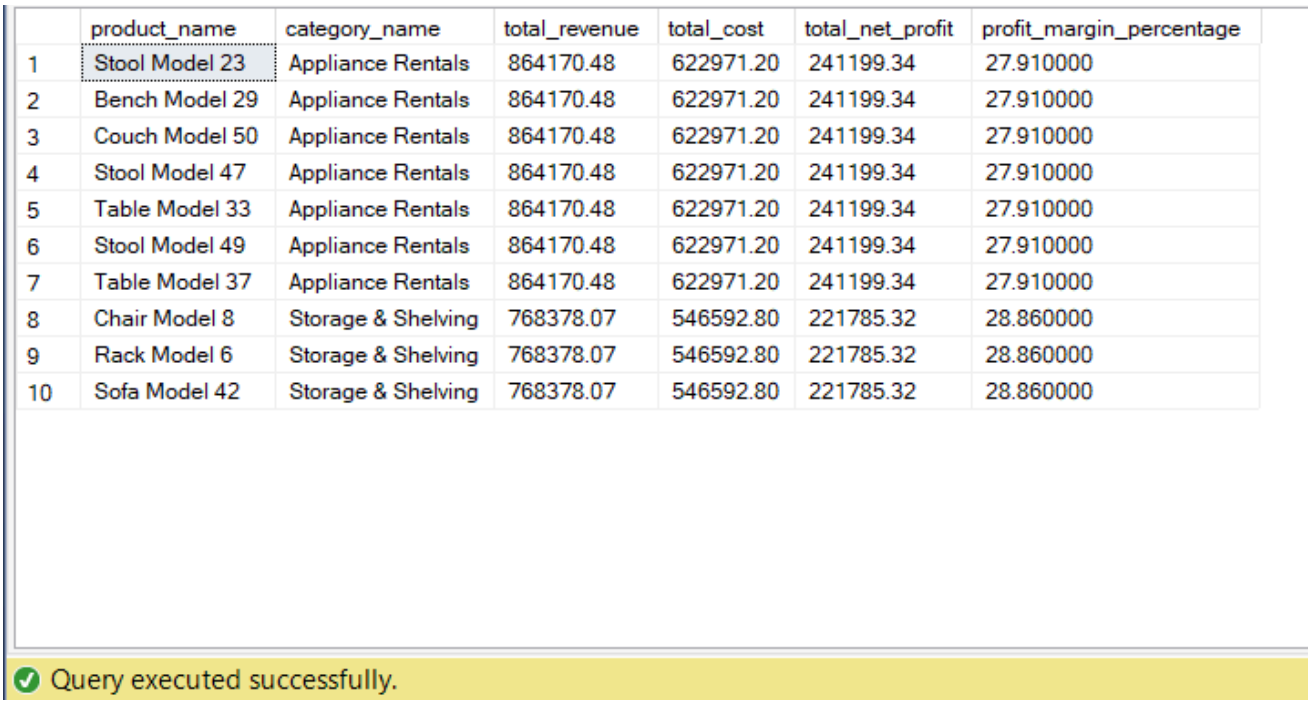
```
SELECT
    dm.year,
    dm.month,
    dc.city_name,
    dcat.category_name,
    SUM(ap.total_revenue) AS total_revenue,
    SUM(ap.total_cost) AS total_cost,
    SUM(ap.net_profit) AS net_profit,
    ROUND(SUM(ap.net_profit) * 100.0 / NULLIF(SUM(ap.total_revenue), 0), 2) AS
profit_margin_percentage
FROM Aggregate_Profitability ap
JOIN Dim_Month dm ON ap.month_id = dm.month_id
JOIN Dim_City dc ON ap.city_id = dc.city_id
JOIN Dim_Category dcat ON ap.category_id = dcat.category_id
GROUP BY dm.year, dm.month, dc.city_name, dcat.category_name
ORDER BY dm.year, dm.month, dc.city_name, dcat.category_name;
```

	year	month	city_name	category_name	total_revenue	total_cost	net_profit	profit_margin_percentage
1	2021	1	Ahmedabad	Appliance Rentals	17742.48	12419.74	5322.74	30.000000
2	2021	1	Ahmedabad	Dining Furniture	2278.04	1594.63	683.41	30.000000
3	2021	1	Ahmedabad	Kids Furniture	9203.09	6442.16	2760.93	30.000000
4	2021	1	Ahmedabad	Mattresses & Bedding	8952.99	6267.09	2685.90	30.000000
5	2021	1	Ahmedabad	Outdoor Furniture	17491.93	12244.35	5247.58	30.000000
6	2021	1	Ahmedabad	Storage & Shelving	0.00	0.00	0.00	NULL
7	2021	1	Bangalore	Appliance Rentals	10208.15	7145.70	3062.45	30.000000
8	2021	1	Bangalore	Dining Furniture	17811.86	12468.30	5343.56	30.000000
9	2021	1	Bangalore	Home Office Essentials	3274.16	2291.91	982.25	30.000000
10	2021	1	Bangalore	Living Room Furniture	0.00	0.00	0.00	NULL
11	2021	1	Bangalore	Office Furniture	8813.00	6169.10	2643.90	30.000000
12	2021	1	Bangalore	Outdoor Furniture	3723.93	2606.75	1117.18	30.000000
13	2021	1	Chennai	Appliance Rentals	5926.23	5333.61	592.62	10.000000
14	2021	1	Chennai	Home Office Essentials	0.00	0.00	0.00	NULL
15	2021	1	Chennai	Living Room Furniture	3301.28	2971.15	330.13	10.000000
16	2021	1	Delhi	Appliance Rentals	0.00	0.00	0.00	NULL
17	2021	1	Delhi	Home Office Essentials	8584.23	6009.03	2575.20	30.000000

Query executed successfully.

Query 7: Top 10 Most Profitable Products Overall

```
SELECT TOP 10
    dp.product_name,
    dp.category_name,
    SUM(ap.total_revenue) AS total_revenue,
    SUM(ap.total_cost) AS total_cost,
    SUM(ap.net_profit) AS total_net_profit,
    ROUND(SUM(ap.net_profit) * 100.0 / NULLIF(SUM(ap.total_revenue), 0), 2) AS
profit_margin_percentage
FROM Aggregate_Profitability ap
JOIN Dim_Category dc ON ap.category_id = dc.category_id
JOIN Dim_Product dp ON dp.category_name = dc.category_name
GROUP BY dp.product_name, dp.category_name
ORDER BY total_net_profit DESC;
```

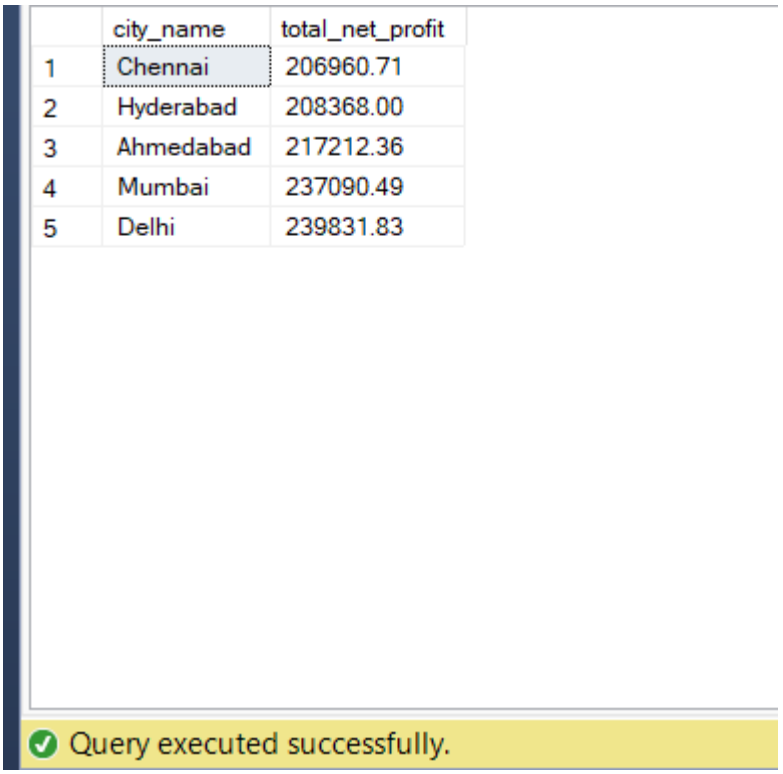


	product_name	category_name	total_revenue	total_cost	total_net_profit	profit_margin_percentage
1	Stool Model 23	Appliance Rentals	864170.48	622971.20	241199.34	27.910000
2	Bench Model 29	Appliance Rentals	864170.48	622971.20	241199.34	27.910000
3	Couch Model 50	Appliance Rentals	864170.48	622971.20	241199.34	27.910000
4	Stool Model 47	Appliance Rentals	864170.48	622971.20	241199.34	27.910000
5	Table Model 33	Appliance Rentals	864170.48	622971.20	241199.34	27.910000
6	Stool Model 49	Appliance Rentals	864170.48	622971.20	241199.34	27.910000
7	Table Model 37	Appliance Rentals	864170.48	622971.20	241199.34	27.910000
8	Chair Model 8	Storage & Shelving	768378.07	546592.80	221785.32	28.860000
9	Rack Model 6	Storage & Shelving	768378.07	546592.80	221785.32	28.860000
10	Sofa Model 42	Storage & Shelving	768378.07	546592.80	221785.32	28.860000

Query executed successfully.

Query 8: Top 5 Cities with Lowest Profit (All Years)

```
SELECT TOP 5
    ci.city_name,
    SUM(p.net_profit) AS total_net_profit
FROM Aggregate_Profitability p
JOIN Dim_City ci ON p.city_id = ci.city_id
JOIN Dim_Month m ON p.month_id = m.month_id
GROUP BY ci.city_name
ORDER BY total_net_profit ASC;
```

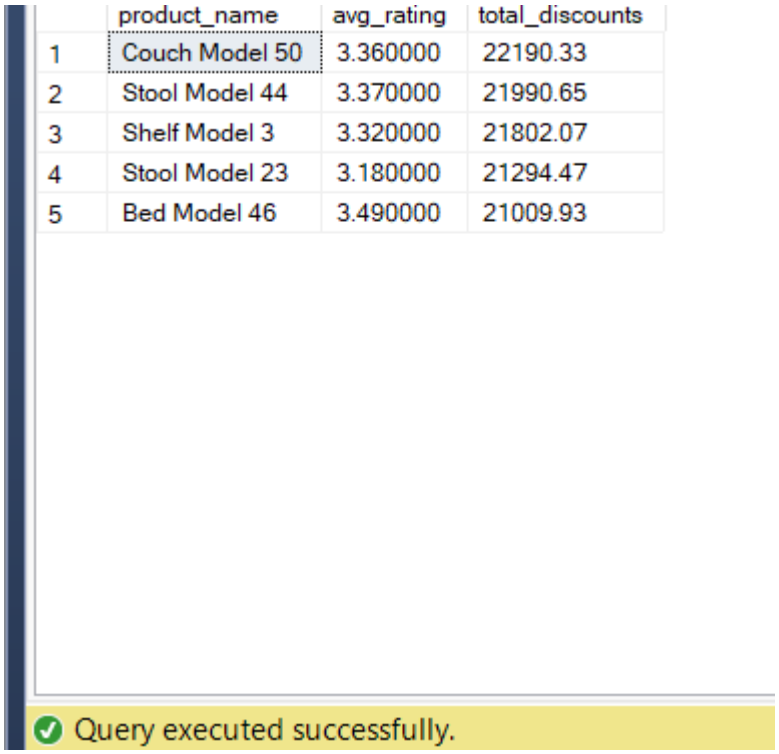


	city_name	total_net_profit
1	Chennai	206960.71
2	Hyderabad	208368.00
3	Ahmedabad	217212.36
4	Mumbai	237090.49
5	Delhi	239831.83

Query executed successfully.

Query 9 : Top 5 Products with High Discounts and Low Ratings

```
SELECT TOP 5
    p.product_name,
    ROUND(AVG(a.avg_rating), 2) AS avg_rating,
    ROUND(SUM(a.total_discounts), 2) AS total_discounts
FROM Aggregate_ProductPerformance a
JOIN Dim_Product p ON a.product_id = p.product_id
GROUP BY p.product_name
HAVING AVG(a.avg_rating) < 3.5
ORDER BY total_discounts DESC;
```



	product_name	avg_rating	total_discounts
1	Couch Model 50	3.360000	22190.33
2	Stool Model 44	3.370000	21990.65
3	Shelf Model 3	3.320000	21802.07
4	Stool Model 23	3.180000	21294.47
5	Bed Model 46	3.490000	21009.93

Query executed successfully.

3. Inventory & Supply Chain (Objective: Loss Minimization and Logistics Efficiency)

Minimize loss, improve logistics efficiency, and understand stock/inflow/outflow trends

Query 10: Monthly Inventory Turnover Ratio by Product (Top 5)

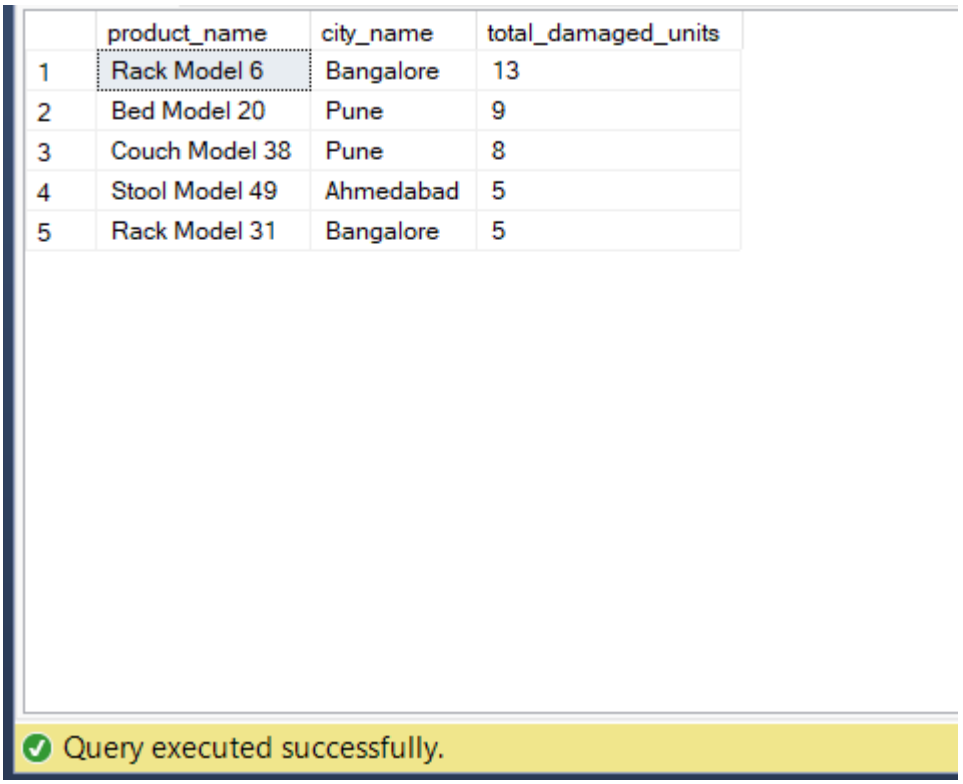
```
SELECT TOP 5
    p.product_name,
    m.year, m.month,
    i.turnover_ratio
FROM Aggregate_InventoryTurnover i
JOIN Dim_Product p ON i.product_id = p.product_id
JOIN Dim_Month m ON i.month_id = m.month_id
WHERE i.turnover_ratio > 0
ORDER BY i.turnover_ratio DESC;
```

	product_name	year	month	turnover_ratio
1	Stool Model 36	2021	1	1.33
2	Sofa Model 42	2021	6	1.00
3	Stool Model 49	2023	5	1.00
4	Couch Model 38	2024	10	1.00
5	Stool Model 30	2021	9	0.89

Query executed successfully.

Query 11: Top 5 Products with Highest Inventory Damage by City (Last Year)

```
SELECT TOP 5
    p.product_name,
    c.city_name,
    SUM(f.inventory_damaged) AS total_damaged_units
FROM Fact_InventorySnapshot f
JOIN Dim_Product p ON f.product_id = p.product_id
JOIN Dim_Date d ON f.date_id = d.date_id
JOIN Dim_City c ON f.city_id = c.city_id
WHERE d.year = YEAR(GETDATE()) - 1
GROUP BY p.product_name, c.city_name
ORDER BY total_damaged_units DESC;
```

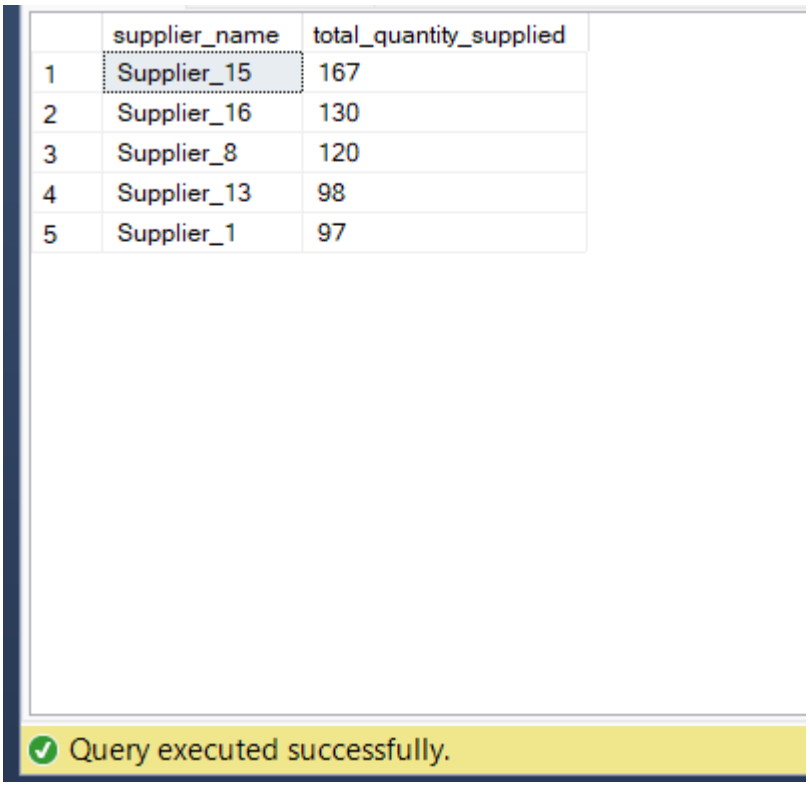


	product_name	city_name	total_damaged_units
1	Rack Model 6	Bangalore	13
2	Bed Model 20	Pune	9
3	Couch Model 38	Pune	8
4	Stool Model 49	Ahmedabad	5
5	Rack Model 31	Bangalore	5

Query executed successfully.

Query 12: Top 5 Suppliers by Volume in Latest Available Year

```
SELECT TOP 5
    s.supplier_name,
    SUM(f.quantity_supplied) AS total_quantity_supplied
FROM Fact_Supplier_Transactions f
JOIN Dim_Supplier s ON f.supplier_id = s.supplier_id
JOIN Dim_Date d ON f.date_id = d.date_id
WHERE d.year = (SELECT MAX(year) FROM Dim_Date)
GROUP BY s.supplier_name
ORDER BY total_quantity_supplied DESC;
```

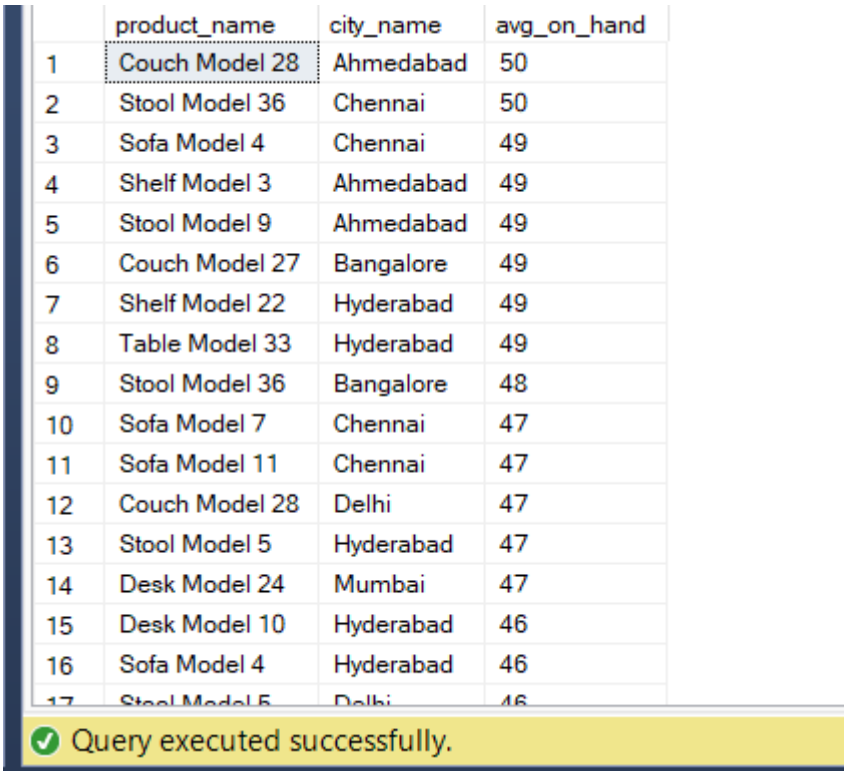


	supplier_name	total_quantity_supplied
1	Supplier_15	167
2	Supplier_16	130
3	Supplier_8	120
4	Supplier_13	98
5	Supplier_1	97

Query executed successfully.

Query 13: Products with High On-Hand Inventory (Potential Overstock)

```
SELECT
    p.product_name,
    c.city_name,
    AVG(f.inventory_on_hand) AS avg_on_hand
FROM Fact_InventorySnapshot f
JOIN Dim_Product p ON f.product_id = p.product_id
JOIN Dim_City c ON f.city_id = c.city_id
GROUP BY p.product_name, c.city_name
HAVING AVG(f.inventory_on_hand) > 25
ORDER BY avg_on_hand DESC;
```



	product_name	city_name	avg_on_hand
1	Couch Model 28	Ahmedabad	50
2	Stool Model 36	Chennai	50
3	Sofa Model 4	Chennai	49
4	Shelf Model 3	Ahmedabad	49
5	Stool Model 9	Ahmedabad	49
6	Couch Model 27	Bangalore	49
7	Shelf Model 22	Hyderabad	49
8	Table Model 33	Hyderabad	49
9	Stool Model 36	Bangalore	48
10	Sofa Model 7	Chennai	47
11	Sofa Model 11	Chennai	47
12	Couch Model 28	Delhi	47
13	Stool Model 5	Hyderabad	47
14	Desk Model 24	Mumbai	47
15	Desk Model 10	Hyderabad	46
16	Sofa Model 4	Hyderabad	46
17	Stool Model 5	Delhi	46

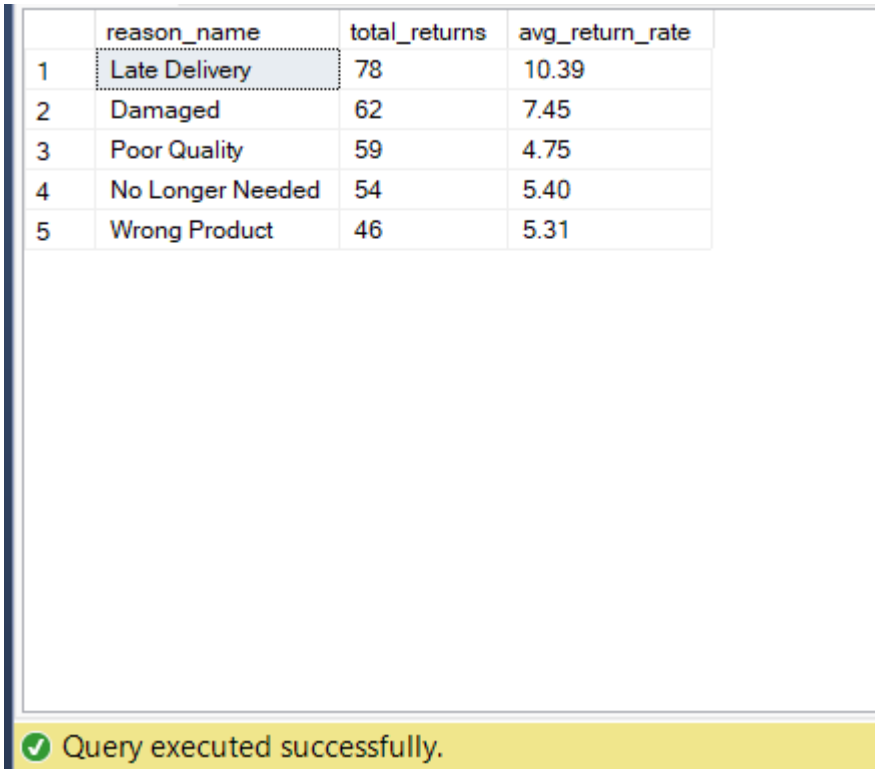
Query executed successfully.

4. Product & Returns (Objective: Quality Improvement and Loss Reduction)

Improve product quality, reduce returns, and identify systemic issues across customer, category, city, and time.

Query 14 : Top 5 Return Reasons Across All Products

```
SELECT TOP 5
   r.reason_name,
   SUM(a.total_returned) AS total_returns,
   ROUND(SUM(a.return_rate), 2) AS avg_return_rate
FROM Aggregate_ReturnAnalysis a
JOIN Dim_ReturnReason r ON a.reason_id = r.reason_id
GROUP BY r.reason_name
ORDER BY total_returns DESC;
```

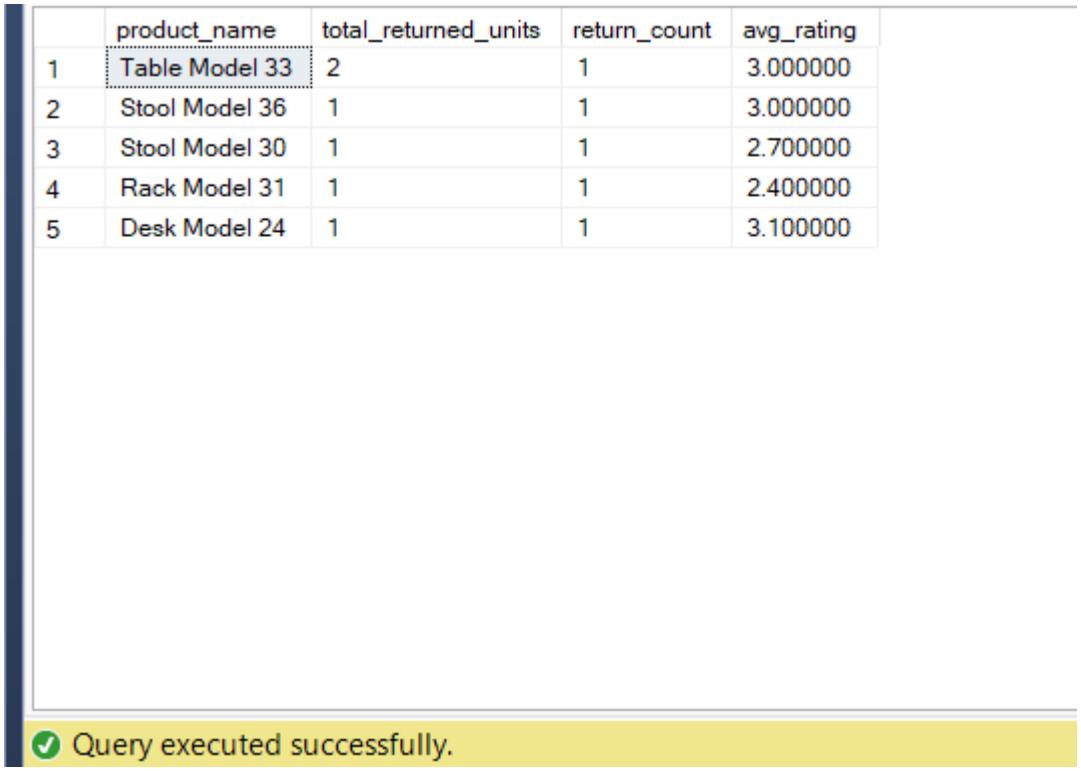


	reason_name	total_returns	avg_return_rate
1	Late Delivery	78	10.39
2	Damaged	62	7.45
3	Poor Quality	59	4.75
4	No Longer Needed	54	5.40
5	Wrong Product	46	5.31

Query executed successfully.

Query 15: Top 5 High-Return Products with Low Customer Ratings (Last 6 Months)

```
SELECT TOP 5
    p.product_name,
    SUM(pr.quantity_returned) AS total_returned_units,
    COUNT(DISTINCT pr.return_id) AS return_count,
    ROUND(AVG(r.rating), 2) AS avg_rating
FROM Fact_ProductReturns pr
JOIN Fact_Rentals r ON pr.rental_id = r.rental_id
JOIN Dim_Product p ON pr.product_id = p.product_id
JOIN Dim_Date d ON pr.date_id = d.date_id
WHERE d.full_date >= DATEADD(MONTH, -6, GETDATE())
GROUP BY p.product_name
HAVING AVG(r.rating) < 3.5
ORDER BY total_returned_units DESC;
```



	product_name	total_returned_units	return_count	avg_rating
1	Table Model 33	2	1	3.000000
2	Stool Model 36	1	1	3.000000
3	Stool Model 30	1	1	2.700000
4	Rack Model 31	1	1	2.400000
5	Desk Model 24	1	1	3.100000

Query executed successfully.

Query 16: High-Rated Products Still Getting Returned

```
SELECT
   p.product_name,
   ROUND(AVG(pp.avg_rating), 2) AS avg_rating,
   ROUND(SUM(a.total_returned) * 1.0 / NULLIF(SUM(a.total_rented), 0), 2) AS return_rate
FROM Aggregate_ReturnAnalysis a
JOIN Dim_Product p ON a.product_id = p.product_id
JOIN Aggregate_ProductPerformance pp ON p.product_id = pp.product_id AND a.month_id =
pp.month_id
GROUP BY p.product_name
HAVING AVG(pp.avg_rating) >= 4.0 AND SUM(a.total_returned) > 0
ORDER BY return_rate DESC;
```

	product_name	avg_rating	return_rate
1	Sofa Model 16	4.200000	2.000000000000000000000000
2	Stool Model 47	4.500000	2.000000000000000000000000
3	Stool Model 49	4.600000	1.000000000000000000000000
4	Rack Model 26	4.200000	0.550000000000000000000000
5	Sofa Model 18	4.230000	0.500000000000000000000000
6	Sofa Model 42	5.000000	0.500000000000000000000000
7	Table Model 12	4.650000	0.500000000000000000000000
8	Chair Model 17	4.800000	0.400000000000000000000000
9	Sofa Model 7	4.900000	0.330000000000000000000000
10	Couch Model 19	4.700000	0.330000000000000000000000
11	Sofa Model 11	4.180000	0.270000000000000000000000
12	Stool Model 9	4.200000	0.250000000000000000000000
13	Chair Model 8	4.700000	0.200000000000000000000000

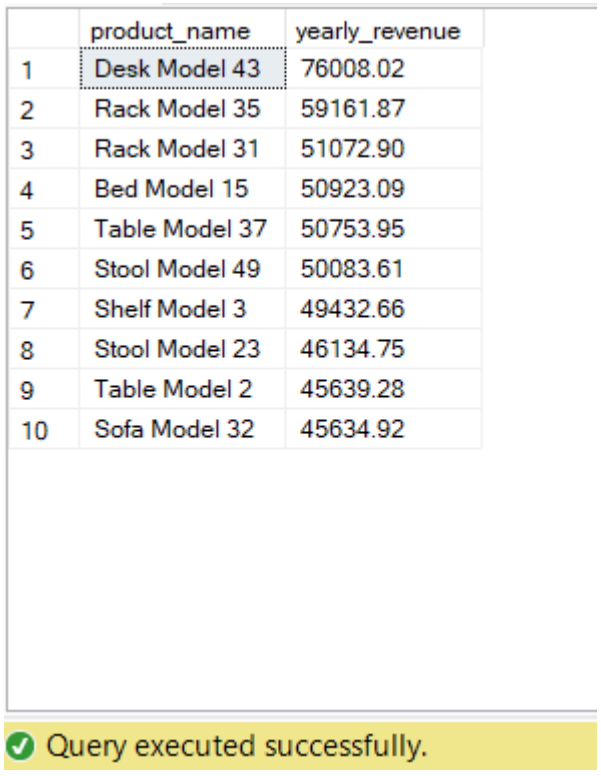
Query executed successfully.

5. Growth & Forecast (Objective: Strategic Planning and Market Expansion)

Identify year-over-year growth, forecast demand, uncover city/category potential, and support business expansion decisions.

Query 17: Top Products Driving Growth in Last Year

```
SELECT TOP 10
   p.product_name,
   SUM(r.total_amount) AS yearly_revenue
FROM Fact_Rentals r
JOIN Dim_Product p ON r.product_id = p.product_id
JOIN Dim_Date d ON r.date_id = d.date_id
WHERE d.year = YEAR(GETDATE()) - 1
GROUP BY p.product_name
ORDER BY yearly_revenue DESC;
```



	product_name	yearly_revenue
1	Desk Model 43	76008.02
2	Rack Model 35	59161.87
3	Rack Model 31	51072.90
4	Bed Model 15	50923.09
5	Table Model 37	50753.95
6	Stool Model 49	50083.61
7	Shelf Model 3	49432.66
8	Stool Model 23	46134.75
9	Table Model 2	45639.28
10	Sofa Model 32	45634.92

Query executed successfully.

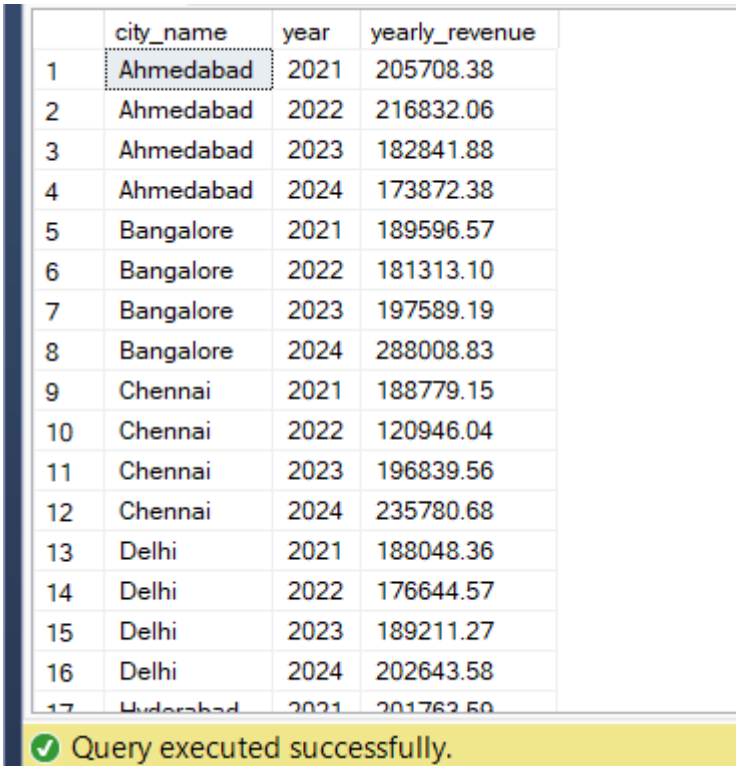
Query 18:City-Wise Customer Growth Over Time

```
SELECT
   ci.city_name,
   d.year,
   COUNT(DISTINCT c.customer_id) AS new_customers
FROM Dim_Customer c
JOIN Dim_City ci ON c.city_id = ci.city_id
JOIN Fact_Rentals r ON c.customer_id = r.customer_id
JOIN Dim_Date d ON r.date_id = d.date_id
GROUP BY ci.city_name, d.year
ORDER BY d.year, new_customers DESC;
```

	city_name	year	new_customers
1	Hyderabad	2021	17
2	Ahmedabad	2021	16
3	Pune	2021	13
4	Bangalore	2021	12
5	Chennai	2021	12
6	Mumbai	2021	12
7	Delhi	2021	11
8	Ahmedabad	2022	16
9	Hyderabad	2022	16
10	Pune	2022	14
11	Bangalore	2022	13
12	Mumbai	2022	12
13	Chennai	2022	11
14	Delhi	2022	10
15	Hyderabad	2023	18
16	Ahmedabad	2023	17
17	Pune	2023	14

Query 19: City-Wise Revenue Growth Trend Over the Years

```
SELECT
   ci.city_name,
   d.year,
   ROUND(SUM(r.total_amount), 2) AS yearly_revenue
FROM Fact_Rentals r
JOIN Dim_City ci ON r.city_id = ci.city_id
JOIN Dim_Date d ON r.date_id = d.date_id
GROUP BY ci.city_name, d.year
ORDER BY ci.city_name, d.year;
```



	city_name	year	yearly_revenue
1	Ahmedabad	2021	205708.38
2	Ahmedabad	2022	216832.06
3	Ahmedabad	2023	182841.88
4	Ahmedabad	2024	173872.38
5	Bangalore	2021	189596.57
6	Bangalore	2022	181313.10
7	Bangalore	2023	197589.19
8	Bangalore	2024	288008.83
9	Chennai	2021	188779.15
10	Chennai	2022	120946.04
11	Chennai	2023	196839.56
12	Chennai	2024	235780.68
13	Delhi	2021	188048.36
14	Delhi	2022	176644.57
15	Delhi	2023	189211.27
16	Delhi	2024	202643.58
17	Hyderabad	2021	201763.59

Query 20: Top 5 Cities with Lowest Return Rate

```
SELECT TOP 5
   c.city_name,
   ROUND(SUM(p.total_revenue), 2) AS total_revenue,
   ROUND(SUM(r.quantity_returned) * 1.0 / NULLIF(SUM(rental.quantity), 0), 2) AS return_rate
FROM Aggregate_Profitability p
JOIN Dim_City c ON p.city_id = c.city_id
JOIN Fact_ProductReturns r ON r.city_id = c.city_id
JOIN Fact_Rentals rental ON rental.city_id = c.city_id
GROUP BY c.city_name
ORDER BY return_rate ASC;
```

	city_name	total_revenue	return_rate
1	Hyderabad	3071849733.81	0.440000000000
2	Ahmedabad	1991775013.20	0.440000000000
3	Bangalore	3297554606.50	0.450000000000
4	Mumbai	3982256007.68	0.470000000000
5	Delhi	2983824444.32	0.490000000000

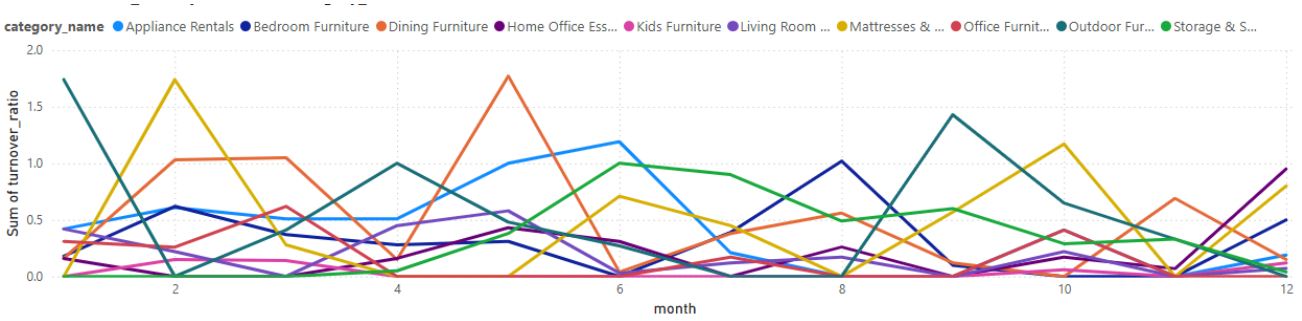
Query executed successfully.

POWER-BI VISUALIZATIONS

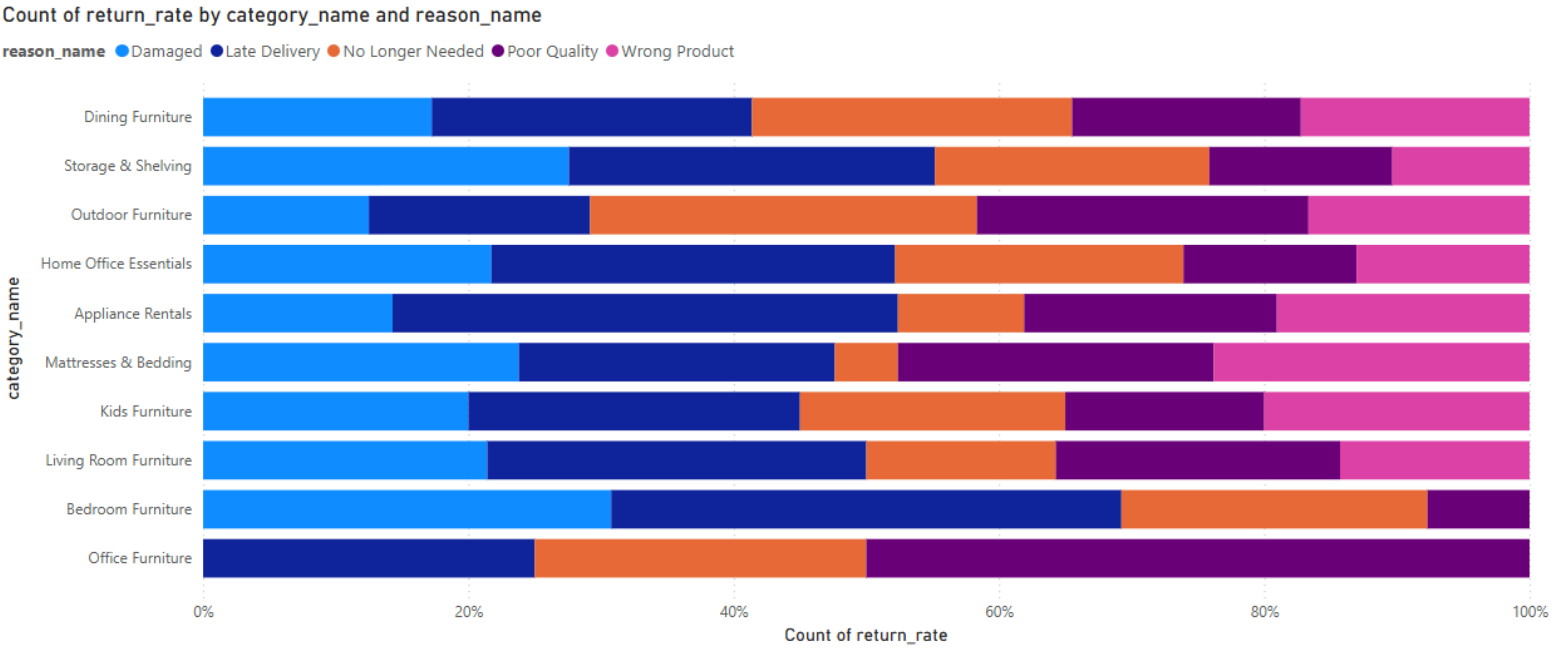
Discount and Revenue Comparison per category per month

city_name	Appliance Rentals	Bedroom Furniture	Dining Furniture	Home Office Essentials	Kids Furniture	Living Room Furniture	Mattresses & Bedding	Office Furniture	Outdoor Furniture	Storage & Shelving	**Total**
Ahmedabad	29,770.06	23,752.61	20,810.17	19,351.84	10,915.55	15,257.72	23,235.84	1,696.33	43,046.54	29,375.67	**2,17,212.32**
Bangalore	54,381.81	14,198.04	26,075.89	30,639.65	19,309.70	25,202.66	17,725.21	9,662.63	24,668.89	22,289.13	**2,44,153.60**
Chennai	22,467.14	15,207.77	31,831.13	26,760.59	17,131.01	13,796.87	20,932.34	4,303.50	34,356.51	20,173.73	**2,06,960.59**
Delhi	29,488.52	33,807.68	20,889.50	18,964.19	17,756.83	16,189.61	11,405.42	3,741.77	27,471.77	60,116.50	**2,39,831.80**
Hyderabad	24,034.08	14,074.52	27,860.98	20,479.84	10,791.83	18,574.21	38,072.10	5,877.22	24,203.93	24,399.28	**2,08,368.00**
Mumbai	34,120.17	14,276.70	29,246.74	33,239.41	13,313.89	23,780.24	33,390.19	3,594.89	15,593.14	36,535.04	**2,37,090.40**
Pune	46,937.51	13,818.99	26,917.96	25,751.32	19,177.03	14,404.30	28,623.55	5,874.76	31,708.38	28,895.95	**2,42,109.74**
Total	**2,41,199.29**	**1,29,136.32**	**1,83,632.36**	**1,75,186.84**	**1,08,395.84**	**1,27,205.59**	**1,73,384.64**	**34,751.10**	**2,01,049.15**	**2,21,785.31**	**15,95,726.45**

Revenue per product category per city

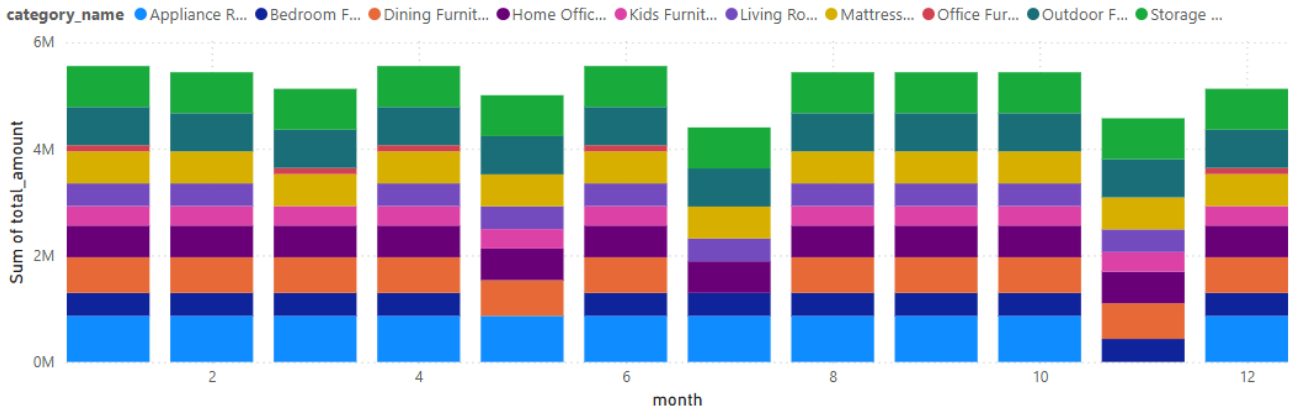


Turnover Ratio of inventory per category per month



Return rate with return reason per category

Return rate per category



Revenue per month per category

city_name	Appliance Rentals	Bedroom Furniture	Dining Furniture	Home Office Essentials	Kids Furniture	Living Room Furniture	Mattresses & Bedding	Office Furniture	Outdoor Furniture	Storage & Shelving	**Total**
Ahmedabad	3.26	3.45	3.14	3.42	3.65	3.46	3.53	2.85	3.49	3.47	**3.40**
Bangalore	3.38	3.17	3.46	3.41	3.42	3.48	3.56	3.03	3.44	3.55	**3.42**
Chennai	3.32	3.54	3.76	3.47	3.47	3.30	3.74	3.10	3.56	3.56	**3.53**
Delhi	3.48	3.74	3.38	2.84	3.40	3.51	4.28	4.17	3.47	3.71	**3.56**
Hyderabad	3.59	3.58	3.43	3.85	3.59	3.27	3.58	3.67	3.83	3.25	**3.55**
Mumbai	3.15	3.93	3.29	3.32	3.83	3.19	3.51	3.63	3.55	3.49	**3.44**
Pune	3.51	3.34	3.46	3.06	3.91	3.13	3.42	3.20	3.63	3.29	**3.42**
Total	**3.39**	**3.54**	**3.42**	**3.38**	**3.60**	**3.34**	**3.61**	**3.41**	**3.55**	**3.49**	**3.47**

Rating per city per category

SYNTHETIC DATA GENERATION

Methodology & Script Structure for data generation

To generate our synthetic Furlenco dataset, we structured the Python script into four main phases: environment setup, dimension table creation, fact table generation, and aggregate computation. Below is a refined outline with representative code snippets.

1. Environment Setup

Set up imports, RNG seeds for reproducibility, and ensure the output directory exists.

python

```
import pandas as pd
import numpy as np
import random
from datetime import datetime, timedelta
import os

# Reproducibility
random.seed(123)
np.random.seed(123)

# Prepare output folder
OUTPUT_DIR = "FURLENCO_CSV_Output"
os.makedirs(OUTPUT_DIR, exist_ok=True)
```

2.2 Dimension Tables

Each dimension follows the schema, populated with realistic placeholders.

python

```
# Dim_City: Seven major Indian metros
cities = [("Mumbai","Maharashtra"), ("Delhi","Delhi"), ...]
dim_city = pd.DataFrame([
    {"city_id": i+1, "city_name": name, "state": st, "country": "India"}
    for i,(name,st) in enumerate(cities)
])

# Dim_Date: Daily from 2021-01-01 to 2024-12-31
date_range = pd.date_range("2021-01-01","2024-12-31",freq="D")
dim_date = pd.DataFrame({
    "date_id": np.arange(1,len(date_range)+1),
    "full_date": date_range,
    "day": date_range.day,
    "month": date_range.month,
    "quarter": date_range.quarter,
    "year": date_range.year,
    "festival": np.random.choice(
        ["Diwali","Holi","Eid","Christmas",""], len(date_range)
    ),
    "day_of_week": date_range.day_name(),
    "is_weekend": (date_range.weekday >= 5).astype(int)
})
```

Additional dimensions—**Dim_Month**, **Dim_Category**, **Dim_Product**, **Dim_Supplier**, **Dim_Customer**, **Dim_ReturnReason**—are built similarly by sampling from small "fake" name and category pools and joining on the base date or city tables.

2.3 Fact Tables

We generate transactional facts by randomly sampling foreign keys and numeric values within realistic bounds:

python

```
# Example: Fact_Rentals
```

```
fact_rentals = pd.DataFrame([{
    "rental_id": i+1,
    "product_id": random.randint(1, len(dim_product)),
    "customer_id": random.randint(1, len(dim_customer)),
    "date_id":    random.randint(1, len(dim_date)),
    "city_id":    random.randint(1, len(dim_city)),
    "quantity":   random.randint(1,5),
    "total_amount": round(
        random.uniform(1000,10000), 2
    ),
    "discount_amount": round(
        random.uniform(100,1500), 2
    ),
    "rating": round(random.uniform(2.0,5.0),1)
} for i in range(1000)])
```

Tables **Fact_InventorySnapshot**, **Fact_Supplier_Transactions**, and **Fact_ProductReturns** follow the same pattern with appropriate columns and ranges.

2.4 Aggregate Tables

To compute monthly aggregates:

1. **Join to Dim_Month** via the day-level date_id → month/year.
2. **Group-by** the relevant keys.
3. **Aggregate** using sum(), mean(), or custom formulas.

python

```
# Product Performance: sum discounts, revenue; avg rating
fact_rentals = fact_rentals.merge(
    dim_date[["date_id","month","year"]],
    on="date_id", how="left"
```

```
).merge(dim_month, on=["month","year"], how="left")

agg_product_perf = (
    fact_rentals
    .groupby(["month_id","product_id"])
    .agg(
        total_discounts=("discount_amount","sum"),
        total_revenue=("total_amount","sum"),
        avg_rating=("rating","mean")
    )
    .reset_index()
)
```

Similar workflows produce **Aggregate_InventoryTurnover**, **Aggregate_Profitability**, and **Aggregate_ReturnAnalysis**.

2.5 Export to CSV

Finally, each of the 16 tables is written out with:

python

```
for name, df in tables.items():
    df.to_csv(f"{OUTPUT_DIR}/{name}.csv", index=False)
```

This modular structure keeps the code clear, reusable, and easy to adapt for other synthetic-data scenarios.

FULL CODE - Added to Zip folder

REFERENCES

The following references were utilized in the design, development, implementation, and analysis phases of the furniture rental data warehouse project:

1. Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd Edition)*. Wiley.

2. Inmon, W.H. (2005). *Building the Data Warehouse (4th Edition)*. Wiley Publishing.

3. Golfarelli, M., & Rizzi, S. (2009). *Data Warehouse Design: Modern Principles and Methodologies*. McGraw-Hill Education.

4. Adamson, C. (2010). *Star Schema: The Complete Reference*. McGraw-Hill Education.

5. Rainardi, V. (2008). *Building a Data Warehouse: With Examples in SQL Server*. Apress.

6. Microsoft SQL Server Documentation. (2024). Retrieved from: https://learn.microsoft.com/sql/sql-server/

7. Microsoft Power BI Documentation. (2024). Retrieved from: https://learn.microsoft.com/power-bi/

8. Pandas Library Documentation. (2024). Retrieved from: https://pandas.pydata.org/pandas-docs/stable/