SUPER MARKET DECISION SYSTEM

Azuq SuperMarket decision making system

Prepared By:

Azharudin	0115-BSCS-20
Syed Zain Ali shah	0151-BSCS-20
Usman Shafique	0016-BSCS-20

Submitted to:

Professor Nadeem Zafar

Table of Contents

Executive summary

The Azuq SuperMarket Data Warehouse project is a comprehensive initiative aimed at enhancing data management and analytical capabilities for informed decision-making within our organization. This executive summary encapsulates the core aspects of our project, commencing with a collaborative approach between business and IT teams during the concept phase. The document delves into intricate analyses of key source entities, including dimensions such as DATE_Dm, PRODUCT_Dm, Depatement_Dm, CUSTOMER_Dm, Employee_Dm, and Vendor_Dm, alongside fact tables INVENTORY_Fct and Sales_Fct. Thoroughly documented requirements cover business needs, data specifications, query criteria, and technical functionalities, laying a solid foundation for subsequent development stages. The architectural design, aligned with the Kimball methodology, incorporates crucial technical components such as SSAS, SSMS, and business intelligence tools, ensuring a robust infrastructure. Conceptual, logical, and physical data models provide a detailed blueprint, culminating in illustrative snapshots showcasing the potential impact on cross-dimensional insights, customer-centric understanding, operational efficiency, data quality, and performance monitoring. This executive summary serves as a precise guide, steering stakeholders toward successful implementation and utilization of the Azuq SuperMarket Data Warehouse, empowering our organization with a data-driven approach to decision-making.

1.0 Introduction

The concept phase is about describing what the big idea is. The business may have a concept and the IT team will be able to describe the major component and concepts of a data warehouse.

***1.1* Analysis**

The goal of the analysis phase is to identify the sources of the information required to populate the physical data models. The main goal should be to populate the repository data model as this is used as the source for all data in the data marts. This is achieved in a number of steps:

1.1.1 Source Systems Analysis (SSA)

The hardware must be able to run SSMS, and SSDT softwares on Microsoft visual studio version 2019. For datawarehouse implementation these tools are required to be run on analyzing system of user.

1.1.2 Source Entity Analysis (SEA)

1.Source Entity Analysis - DATE_Dm Dimension:

- **DATE_Dm Table**:
  - Entity Name: `DATE_Dm`
  - Description: Represents the dimension table for date-related information.
  - Columns:
  - `Daily` (Primary Key): Daily date information for sales analyze
  - `Weekly`: Weekly date information. It is lower grain and primary key for Inventory Management.
  - `Monthly`: Monthly date information.
  - `Quarterly`: Quarterly date information.
  - `Yearly`: Yearly date information.

2.Source Entity Analysis - PRODUCT_Dm Dimension:

- **PRODUCT_Dm Table:**
  - Entity Name: `PRODUCT_Dm`
  - Description: Represents the dimension table for product-related information.
  - Columns:
  - `ProductID` (Primary Key): Unique identifier for the product.
  - `ProductName`: Name of the product.
  - `CategoryID` (Foreign Key): Foreign key referencing the product category.
  - `BrandName`: Brand name of the product.

3. Source Entity Analysis - Depatement_Dm Dimension:

- **Depatement_Dm Table:**
  - Entity Name: `Depatement_Dm`
  - Description: Represents the dimension table for department-related information.
  - Columns:
  - `DepartementID` (Primary Key): Identifier for the department.
  - `StoreID`: Store identifier.
  - `RegionID`: Region identifier.
  - Other store-related attributes such as `StoreName`, `StoreType`.

4.Source Entity Analysis - CUSTOMER_Dm Dimension:

- **CUSTOMER_Dm Table:**
  - Entity Name: `CUSTOMER_Dm`
  - Description: Represents the dimension table for customer-related information.
  - Columns:
  - `CustomerID` (Primary Key): Unique identifier for the customer.
  - `StoreID` (Foreign Key): Foreign key referencing the store.
  - Personal information such as `CustomerName`, `Email`, `Phone`.
  - Demographic information like `Country`, `Province`, `City`, `Address`.
  - Attributes like `Gender`, `MaritalStatus`, `Occupation`.

5.Source Entity Analysis - Employee_Dm Dimension:

- **Employee_Dm Table:**
  - Entity Name: `Employee_Dm`
  - Description: Represents the dimension table for employee-related information.
  - Columns:
  - `EmployeeID` (Primary Key): Unique identifier for the employee.
  - `ManagerID`: Manager identifier.
  - `StoreID` (Foreign Key): Foreign key referencing the store.
  - Employee attributes such as `JobID`, `Age`, `dutyTime`, `JobExperience`.
  - Additional details like `Country`, `Province`, `City`, `Address`.
  - Attributes like `Gender`, `MaritalStatus`, `Occupation`.

6.Source Entity Analysis - Vendor_Dm Dimension:

- **Vendor_Dm Table:**

- Entity Name: `Vendor_Dm`
- Description: Represents the dimension table for vendor-related information.
- Columns:
- `VendorID` (Primary Key): Vendor identifier.
- `VendorName`: Name of the vendor.
- Contact-related attributes such as `ContactPerson`, `ContactEmail`, `ContactPhone`.
    - Additional details like `Country`, `Province`, `City`, `Address`
- Attributes like `PerformanceHistory`, `QualityRating`.

7.Source Entity Analysis - INVENTORY_Fct Fact Table:

- **INVENTORY_Fct Table:**
    - *Entity Name:* `INVENTORY_Fct`
    - Description: Represents the fact table for inventory-related data.
      "Inventory refers to all the items, goods, merchandise, and materials held by a business for selling in the market to earn a profit."[4]
    - Columns:
    - *`DateID`* (Foreign Key): Foreign key referencing the `DATE_Dm` dimension.
    - *`ProductID`* (Foreign Key): Foreign key referencing the `PRODUCT_Dm` dimension.
    - *`DepartementID`* (Foreign Key): Foreign key referencing the `Depatement_Dm` dimension.
    - *`VendorID`* (Foreign Key): Foreign key referencing the `Vendor_Dm` dimension.
    - *`EmployeeID`* (Foreign Key): Foreign key referencing the `Employee_Dm` dimension.
    - Metrics:
    - *`BeginningInventory`*
    - *`QuantityPurchase`*
    - *`PurchaseUnitPrice`*
    - *`QuantitySold`*
        - *`EndingInventory`:* (Calculated Attribute: [BeginningInventory + QuantityPurchase - QuantitySold])
        - *`COGS`* (Calculated Attribute: [BeginningInventory + QuantityPurchase - EndingInventory])
        - *`InventoryTurnover`:* (Calculated Attribute: [Weekly COGS / averageTurnover(BeginningInventory + EndingInventory/2)])
          *"*The inventory turnover ratio is the number of times a company has sold and replenished its inventory over a specific amount of time*"[5]*
        - *`DaysofInventory`* (Calculated Attribute: [daysInWeeklyGrain / InventoryTurnover])
          *"*Days in inventory is the average time a company keeps its inventory before they sell it*"[6]*
    - *`ReorderStatus`* (Calculated Attribute: "Reorder" if Ending Inventory < Reorder Threshold; "OK" if Ending Inventory >= Reorder Threshold)

8.Source Entity Analysis - Sales_Fct Fact Table:

- **Sales_Fct Table:**
    - Entity Name: `Sales_Fct`
    - Description: Represents the fact table for sales-related information.

- Columns:
- `*DateID*` (Foreign Key): Foreign key referencing the `DATE_Dm` dimension.
- `*ProductID*` (Foreign Key): Foreign key referencing the `PRODUCT_Dm` dimension.
- *‘DepartementID`* (Foreign Key): Foreign key referencing the `Depatement_Dm` dimension.
- `*CustomerID*` (Foreign Key): Foreign key referencing the `CUSTOMER_Dm` dimension.
- *EmployeeID*` (Foreign Key): Foreign key referencing the `Employee_Dm` dimension.
- Metrics:
  - *`QuantitySold`*
  - *`SalesAmount`*
  - *`UnitPrice`*
  - *`CostPrice`*
  - *`DiscountApplied`*
  - Calculated Metrics:
    - `*GrossMargin*`: [(SalesAmount - CostPrice) / SalesAmount] * 100.

- *“*The portion of a company’s revenue left over after direct costs are subtracted.Gross margin is one of the most important indicators of a company’s financial performance. It’s the portion of business revenue left over after you subtract direct costs, such as labour and raw materials.*”[1]*

    - *`GrossProfit`:* SalesAmount - CostPrice.
    - `*NetProfit*`: SalesAmount - CostPrice - Operating Expenses.
      - *“*Net profit is the amount of money remaining after deducting a company's total expenses from its total revenue for a given accounting period*”[2]*

    - *GrossProfit*: SalesAmount - CostPrice.
      - **“**Gross profit measures the money your goods or services earned after subtracting the total costs to produce and sell them. The formula to calculate gross profit is the **total revenue** minus the **cost of goods sold.”**[3]

  - *`AverageDiscountPercentage`*
  - *`ReturnsAmount`*
  - *`ReturnsQuantity`*

These analyses provide a detailed breakdown of the structure and attributes for each dimension and the fact table in your schema. Make sure to adapt these notes to your specific business requirements and data characteristics.

2.0 Requirements

The requirements gathering phase of any data warehouse is one of the most difficult. The objective of these templates is to give breadth and depth to the requirements. Breadth is the ability to ensure that all truly required information would be covered, whilst depth is the amount of detail that is specified in the requirements to ensure that the developers have sufficient, unambiguous, detail with which to develop. Requirements should have a programme-long life cycle. After the initial version of the requirements is developed a project can start the build, however the business moves on and therefore whilst the build phase is occurring it is important that new versions of the requirements are also being developed. A project within a programme of work should have a fixed version of the requirements; however each project may work with a different version of the requirements.

2.1 Data Warehouse Business Requirements (WBR)
Azuq SuperMarket takes decision of inventory management system based on following KPIs:

QuantitySold,EndingInventory,COGS,InventoryTurnover,DaysofInventory,and ReorderStatus. The InventoryTurnover and DaysofInventory KPIs helps the stakeholder to take decision that in a weekly grain how many times the prodcuts is solved and how many days the product remain unsolved in a weekly grain.

Azuq SuperMarket takes decision of Sales management system based on following KPIs:

GrossMargin, GrossProfit, and profit NetProfit. The GrossMargin indicates the manager that what percentage of revenue remains after subtracting the COGS of product and Netprofit indicates to manager that subtracting the expenses on products also from revenue with COGS how much profit is gained.**2.2 Data Warehouse Data Requirements (WDR)**
For sales and Inventory analyzing system the Azuq SuperMarket's following data is required.

Data Warehouse Data Requirements (WDR) - Dimensions:

I. **DATE_Dm Dimension:**

Data Elements:
1. `DepartementID`: Identifier for the department.
2. `Weekly`: Weekly date information.
3. `Monthly`: Monthly date information.
4. `Quarterly`: Quarterly date information.
5. `Yearly`: Yearly date information.

II. **PRODUCT_Dm Dimension:**

Data Elements:
1. `ProductID`: Unique identifier for the product.
2. `ProductCategory`: Category to which the product belongs.
3. `BrandName`: Brand name of the product.

III. **Depatement_Dm Dimension:**

Data Elements:
1. `DepartementID`: Identifier for the department.
2. `StoreID`: Store identifier.
3. `RegionID`: Region identifier.
4. Other store-related attributes such as `StoreName`, `StoreType`.

IV. **Vendor_Dm Dimension:**

Data Elements:
1. `VendorID`: Vendor identifier.
2. `VendorName`: Name of the vendor.
3. Contact-related attributes such as `ContactPerson`, `ContactEmail`, `ContactPhone`.
4. QualityPerformance,and History.

V. **Employee_Dm Dimension**:

Data Elements:

1. `EmployeeID`: Employee identifier.
2. `ManagerID`: Manager identifier.
3. `JobID`: Foreign key referencing the job information.
4. Employee attributes such as `dutyTime`, `JobExperience`.
5. Additional details like `Country`, `Province`, `City`, `Address`.
6. Attributes like `PerformanceHistory`, `QualityRating`.

VI. Data Warehouse Data Requirements (WDR) -SALES_ Fact Tables:

Data Elements:

1. `DateID`: Foreign key referencing the `DATE_Dm` dimension.
2. `ProductID`: Foreign key referencing the `PRODUCT_Dm` dimension.
3. `DepartementID`: Foreign key referencing the `Depatement_Dm` dimension.
4. `CustomerID`: Foreign key referencing the `CUSTOMER_Dm` dimension.
5. `EmployeeID`: Foreign key referencing the `Employee_Dm` dimension.
6. Metrics:
7. `QuantitySold`
8. `SalesAmount`
9. `UnitPrice`
10. `CostPrice`
11. `DiscountApplied`
12. Calculated Metrics:
13. `GrossMargin`
14. `GrossProfit`
15. `NetProfit`
16. `AverageDiscountPercentage`
17. `ReturnsAmount`
18. `ReturnsQuantity`

VII. Data Warehouse Data Requirements (WDR) – INVENTORY_fct Tables:

Data Elements:

1. `DateID`: Foreign key referencing the `DATE_Dm` dimension.
2. `ProductID`: Foreign key referencing the `PRODUCT_Dm` dimension.
3. `DepartementID`: Foreign key referencing the `Depatement_Dm` dimension.
4. `VendorID`: Foreign key referencing the `Vendor_Dm` dimension.
5. `EmployeeID`: Foreign key referencing the `Employee_Dm` dimension.
6. Metrics:
7. `BeginningInventory`
8. `QuantityPurchase`
9. `PurchaseUnitPrice`
10. `QuantitySold`
11. `EndingInventory`
12. `COGS`
13. `InventoryTurnover`
14. `DaysofInventory`
15. `ReorderStatus`

2.3 Data Warehouse Query Requirements (WQR)

The Data Warehouse Query Requirements (WQR) for a supermarket data warehouse outline the specific queries and analyses that stakeholders might perform to extract valuable insights. Here's an overview of query requirements for the mentioned dimensions and fact tables:

Data Warehouse Query Requirements (WQR) - Dimensions:

- 1.**DATE_Dm Dimension:**
  - ***Queries:***
  - Retrieve weekly, monthly, quarterly, and yearly sales trends.
  - Analyze temporal patterns for specific departments or regions.

- 2. **PRODUCT_Dm Dimension:**
  - ***Queries:***
  - Identify top-selling products within a specific category.
  - Analyze the performance of products based on brands.

- 3. **Depatement_Dm Dimension:**
  - ***Queries:***
  - Assess sales performance by department and store.
  - Analyze store-specific metrics for regional insights.

- **Vendor_Dm Dimension:**
  - ***Queries:***
  - Evaluate vendor performance and identify top-performing vendors.
  - Analyze vendor-specific metrics such as purchase patterns.

- **Employee_Dm Dimension:**
  - ***Queries:***
  - Assess employee performance and productivity.
  - Analyze employee-related metrics for workforce optimization.

- Data Warehouse Query Requirements (WQR) - Fact Tables:

- **Sales_Fct Fact Table:**
  - ***Queries:***
  - Retrieve total sales amounts for specific time periods.
  - Analyze gross profit margins and net profits.
  - Identify products with the highest returns.

- **NVENTORY_Fct Fact Table:**
  - ***Queries:***
  - Monitor inventory turnover for specific products and departments.
  - Analyze days of inventory and assess reorder status.
  - Identify products with low stock levels for reordering.

Cross-Dimensional Queries:

- **Cross-Dimensional Analysis:**

  Queries:
  - Evaluate the correlation between sales and employee performance.
  - Analyze the impact of vendor performance on inventory turnover.

- **Customer-Centric Insights:**

Queries:

- Identify purchasing patterns for specific customer segments.
- Analyze customer demographics for targeted marketing.

- 3.**Operational Efficiency:**

  Queries:

  - Assess employee scheduling efficiency.
  - Identify areas for improvement in inventory management.

- **Data Quality and Consistency:**

  Queries:

  - Validate data quality across dimensions and fact tables.
  - Identify and resolve any inconsistencies in the data.

- **Performance Monitoring:**

  Queries:

  - Monitor KPIs for sales, inventory, and employee performance.
  - Assess the overall performance of the supermarket based on key metrics.

These query requirements serve as a guide for developing queries and analyses to extract meaningful insights from the supermarket data warehouse. Stakeholders can use these queries to make informed decisions and optimize various aspects of supermarket operations.

2.4 Data Warehouse Technical Requirements (WTR)

The fourth document details the functional and non-functional requirements that are expected of the solution. Again, these requirements are stated from the business perspective rather than the technical perspective. The document should include topics such:

• The functionality required of the query tools. (SSAS,SSMS,MDX,EXCEL)

Functional requirements for a supermarket data warehouse using SQL Server Analysis Services (SSAS) and SQL Server Management Studio (SSMS) would specify the functionalities and capabilities needed to meet the business objectives. Here are some functional requirements:

- **SQL Server Analysis Services (SSAS):**

- **1.Dimensional Modeling:**

  Requirement:

  - Implement a star or snowflake schema for effective dimensional modeling in SSAS, allowing for efficient querying and analysis.

- **2. Cube Design:**

  Requirement:

  - Design SSAS cubes that encapsulate key business metrics, allowing for multidimensional analysis of sales, inventory, and other relevant data.

- 3**.Hierarchies and Aggregations:**

  Requirement:

  - Define hierarchies within dimensions to facilitate drill-down analysis. Implement aggregations to optimize query performance.

- 4**. KPIs (Key Performance Indicators):**

Requirement:

- ➢ Incorporate KPIs in SSAS cubes to monitor and measure performance against predefined business goals.

- ❖ **5. MDX (Multidimensional Expressions) Queries:**

  Requirement:

- ➢ Enable users to write and execute MDX queries within SSAS for custom analysis and reporting.

- ❖ **6. Security and Roles**

  Requirement:

- ➢ Implement role-based security in SSAS to restrict access to sensitive data based on user roles and responsibilities.

- ❖ **7.Processing and Refresh**

  Requirement:

- ➢ Schedule automated cube processing and refreshing to ensure that the data in SSAS reflects the latest changes in the underlying data warehouse.

- ❖ **8. Partitioning:**

  Requirement:

- ➢ Implement partitioning strategies in SSAS cubes to manage large datasets efficiently and improve query performance.

- ❖ **SQL Server Management Studio (SSMS):**

- ❖ 1**. Database Administration:**

  Requirement:

- ➢ Utilize SSMS for database administration tasks, including backup, restore, and monitoring of the supermarket data warehouse.

- ❖ 2**. Querying and Scripting:**

  Requirement:

- ➢ Enable users to write and execute T-SQL queries within SSMS for ad-hoc analysis and troubleshooting.

- ❖ **3. Indexing and Optimization:**

  Requirement:

- ➢ Leverage SSMS for managing indexes and optimizing query performance through execution plans and statistics analysis.

- ❖ 4**. ETL (Extract, Transform, Load) Monitoring:**

  Requirement:

- ➢ Monitor and manage ETL processes using SSMS to ensure timely and accurate data loading into the data warehouse.

- ❖ **5. Security Management:**

  Requirement:

- ➢ Administer security settings in SSMS, including user roles, permissions, and access control to the underlying database.

- ❖ **6.Integration with Version Control:**

  Requirement:

  - ➢ Enable integration with version control systems through SSMS for managing changes and updates to database schemas and scripts.

- ❖ **7.Collaboration and Documentation:**

  Requirement:

  - ➢ Facilitate collaboration among database administrators and developers through SSMS. Maintain comprehensive documentation of database schema and configurations.

- ❖ Cross-Tool Requirements:

- ❖ **1.Compatibility and Versioning:**

  Requirement:

  - ➢ Ensure compatibility and version consistency between SSAS and SSMS to maintain a seamless and integrated development environment.

- ❖ **2. Training and Support:**

  Requirement:

  - ➢ Provide training and support for users and administrators to effectively use SSAS and SSMS functionalities for supermarket data analysis and management.

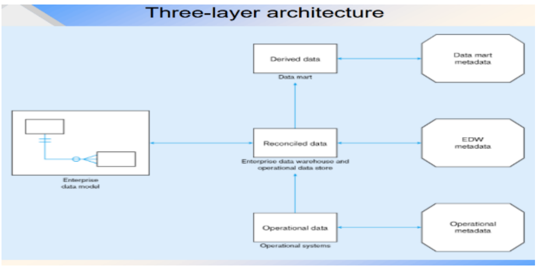
These functional requirements aim to define the necessary features and capabilities for using SSAS and SSMS in a supermarket data warehouse, ensuring efficient analysis, reporting, and administration.

• The performance characteristics.:

The supermarket data warehouse should exhibit high query responsiveness and efficient ETL processing, ensuring timely analysis and reporting while maintaining optimal system performance under varying workloads.

3.0 Architecture

For Azuq SuperMarket Datawarehouse Architecture the Kimball approach is Adapted. In which data is moved to staging area and from staging area data is loaded into data marts which unions and make centralized datawarehouse. Following the Kimball's Approach the Three Layer architecture is adapted in which data is taken from operational database.



3.1 Technical Architecture

The technical architecture of a supermarket data warehouse involves the integration of various components and technologies to facilitate data storage, processing, and analysis. Here's a detailed breakdown of the technical architecture:

- Components:

- Data Sources:
  - Transactional databases, external systems, and other data sources provide raw data for the data warehouse.

- ETL (Extract, Transform, Load) Processes:
  - ETL processes extract, clean, transform, and load data from source systems into the data warehouse. Tools like Apache NiFi, Talend, or Microsoft SQL Server Integration Services (SSIS) can be used.

- Data Warehouse:
  - The central repository for structured, historical data optimized for analytical queries. It may be implemented using relational databases like Microsoft SQL Server, Oracle, or cloud-based solutions like Amazon Redshift or Snowflake.

- Dimensional Modeling:
  - Dimensions and fact tables are designed using dimensional modeling techniques (star or snowflake schema) to support efficient querying. This is often implemented using SQL databases.

- SQL Server Analysis Services (SSAS):
  - SSAS is used for creating multidimensional cubes, providing OLAP capabilities for in-depth data analysis. It enables features such as aggregations, hierarchies, and KPIs.

- SQL Server Management Studio (SSMS):
  - SSMS is utilized for database administration, querying, and scripting. It provides a user interface for managing the data warehouse and optimizing queries.

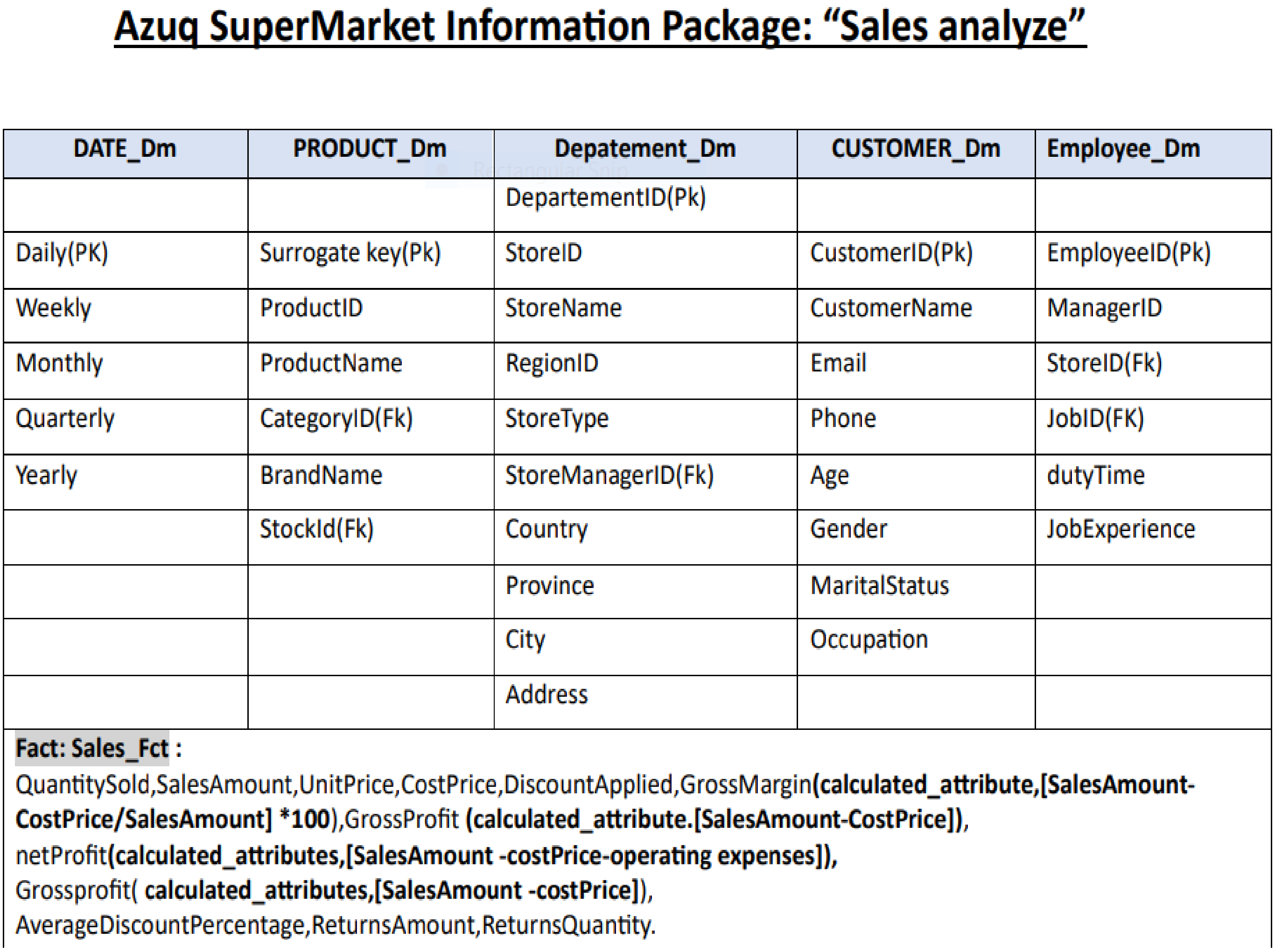
- Business Intelligence Tools:
  - BI tools like Tableau, Power BI, or others connect to the data warehouse, enabling users to create interactive reports and dashboards for data visualization.

- Security Layer:
  - Role-based access controls are implemented to ensure data security. This layer may include encryption, authentication, and authorization mechanisms to protect sensitive information.

- Monitoring and Logging Tools:
  - Monitoring tools track system performance, resource usage, and query execution times. Logging mechanisms capture events, errors, and user activities for analysis and troubleshooting.

- Metadata Management:
  - A metadata repository stores information about data definitions, transformations, and lineage. This supports data governance and ensures transparency.

- Scalability and Redundancy:



Azuq SuperMarket Information Package: "Sales analyze"

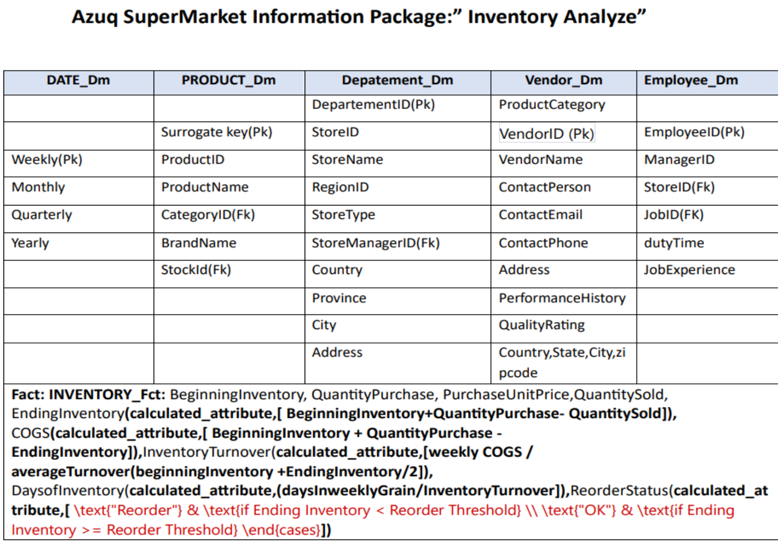
DATE_Dm	PRODUCT_Dm	Depatement_Dm	CUSTOMER_Dm	Employee_Dm
		DepartementID(Pk)		
Daily(PK)	Surrogate key(Pk)	StoreID	CustomerID(Pk)	EmployeeID(Pk)
Weekly	ProductID	StoreName	CustomerName	ManagerID
Monthly	ProductName	RegionID	Email	StoreID(Fk)
Quarterly	CategoryID(Fk)	StoreType	Phone	JobID(FK)
Yearly	BrandName	StoreManagerID(Fk)	Age	dutyTime
	StockId(Fk)	Country	Gender	JobExperience
		Province	MaritalStatus	
		City	Occupation	
		Address		

Fact: Sales_Fct : QuantitySold,SalesAmount,UnitPrice,CostPrice,DiscountApplied,GrossMargin**(calculated_attribute,[SalesAmount-CostPrice/SalesAmount] *100)**,GrossProfit **(calculated_attribute.[SalesAmount-CostPrice])**, netProfit**(calculated_attributes,[SalesAmount -costPrice-operating expenses]),** Grossprofit(**calculated_attributes,[SalesAmount -costPrice]**), AverageDiscountPercentage,ReturnsAmount,ReturnsQuantity.

- The architecture is designed to scale horizontally or vertically to accommodate growing data volumes and user loads. Redundancy mechanisms, such as load balancing and failover, ensure high availability.

4.0 Data Models

4.1: Information Packages



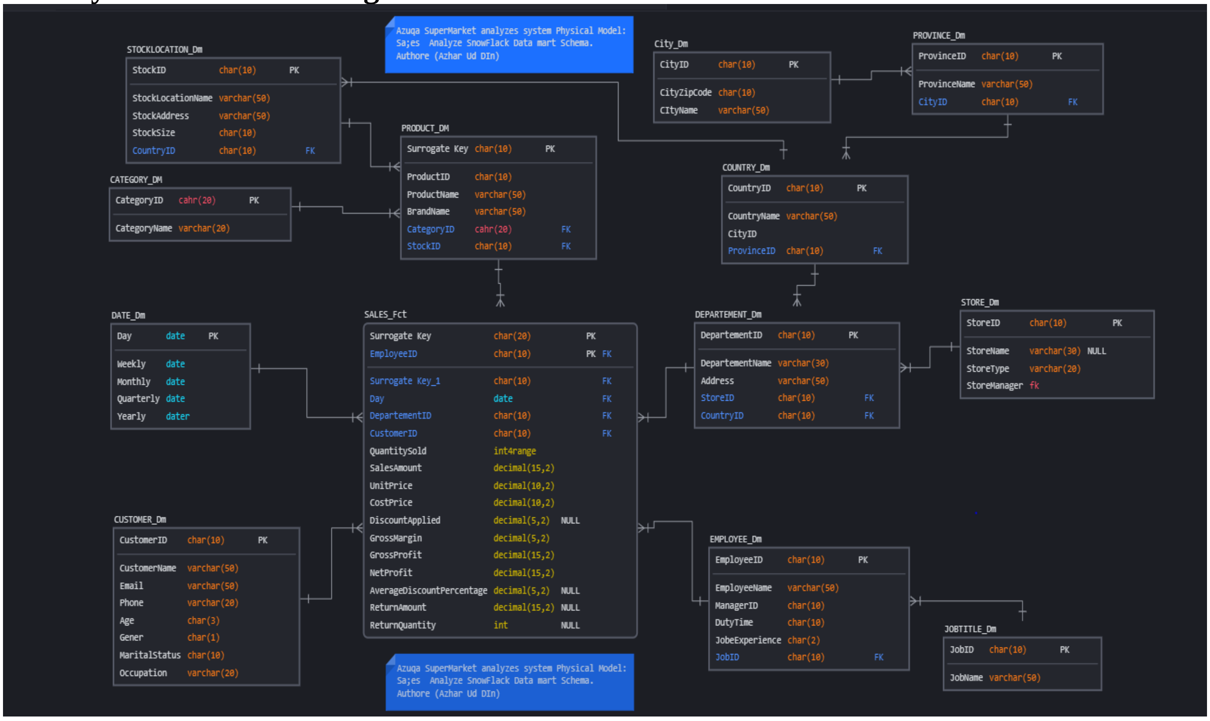
Azuq SuperMarket Information Package:" Inventory Analyze"

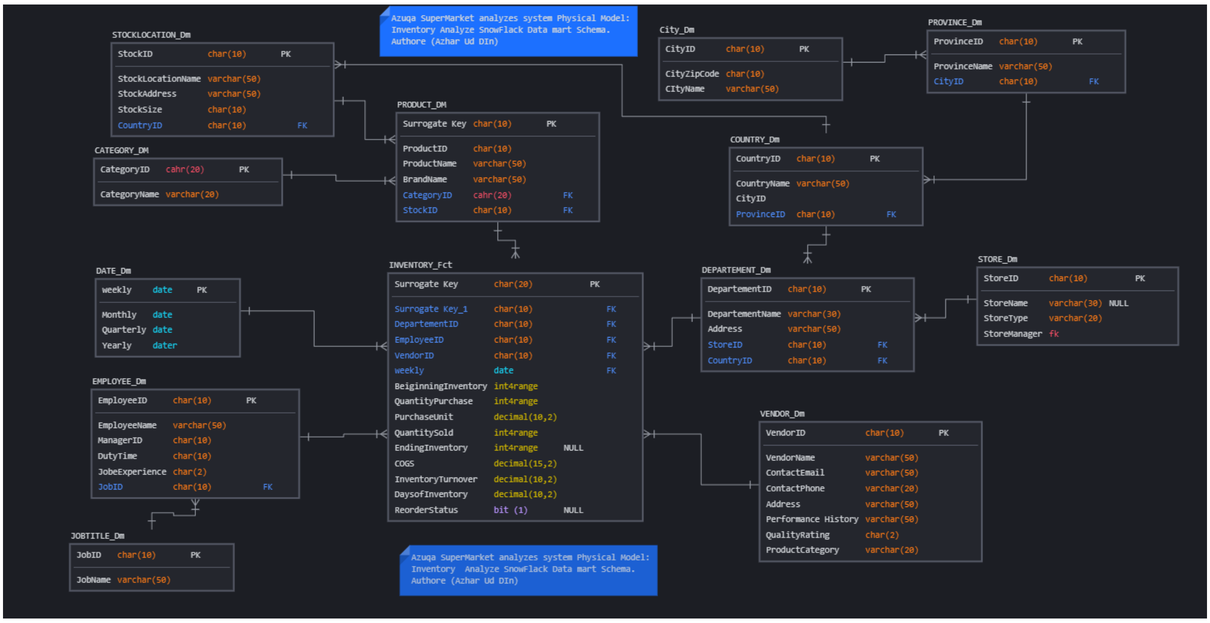
DATE_Dm	PRODUCT_Dm	Depatement_Dm	Vendor_Dm	Employee_Dm
		DepartementID(Pk)	ProductCategory	
	Surrogate key(Pk)	StoreID	VendorID (Pk)	EmployeeID(Pk)
Weekly(Pk)	ProductID	StoreName	VendorName	ManagerID
Monthly	ProductName	RegionID	ContactPerson	StoreID(Fk)
Quarterly	CategoryID(Fk)	StoreType	ContactEmail	JobID(FK)
Yearly	BrandName	StoreManagerID(Fk)	ContactPhone	dutyTime
	StockId(Fk)	Country	Address	JobExperience
		Province	PerformanceHistory	
		City	QualityRating	
		Address	Country,State,City,zipcode	

Fact: INVENTORY_Fct: BeginningInventory, QuantityPurchase, PurchaseUnitPrice,QuantitySold, EndingInventory**(calculated_attribute,[BeginningInventory+QuantityPurchase- QuantitySold]),** COGS**(calculated_attribute,[BeginningInventory + QuantityPurchase - EndingInventory]),**InventoryTurnover(**calculated_attribute,[weekly COGS / averageTurnover(beginningInventory +EndingInventory/2]),** DaysofInventory(**calculated_attribute,(daysInweeklyGrain/InventoryTurnover]),**ReorderStatus(**calculated_attribute,[** \text{"Reorder"} & \text{if Ending Inventory < Reorder Threshold} \\ \text{"OK"} & \text{if Ending Inventory >= Reorder Threshold} \end{cases}**])**

4.2 Conceptual Data Modling:

4.3:Physical Data Modeling:





5.0 Analysis snapshots

Here 5 to 6 types of analysis and its results are displayed by using snapshots of the created data warehouse.

Bibliography

[1]: bdc.ca. (n.d.). Retrieved from https://www.bdc.ca/en/articles-tools/entrepreneur-toolkit/templates-business-guides/glossary/gross-margin#:~:text=Gross%20margin%20definition,as%20labour%20and%20raw%20materials

capitalone.com. (n.d.). Retrieved from https://www.capitalone.com/learn-grow/business-resources/what-is-gross-profit/

netsuite.com. (n.d.). Retrieved from https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-turnover-ratio.shtml#:~:text=Inventory%20turnover%20is%20the%20rate,lower%20one%20to%20weak%20sales.

paddle.com. (n.d.). Retrieved from https://www.paddle.com/resources/net-profit-formula

zoho.com. (n.d.). Retrieved from https://www.zoho.com/inventory/guides/inventory-definition-meaning-types.html#:~:text=Inventory%20refers%20to%20all%20the,newspaper%20will%20be%20considered%20inventory.