EDM Final Term Project Report

STEP 1: Topic/Domain Selection

Selected Topic: Business Insights Management System Introduction:

Our topic is focused on business and major enterprise insights. A Business Insights Management System serves as the backbone for businesses, acting as a sophisticated database management system tailored for major enterprises. It efficiently stores crucial information spanning products, markets, sales, and financial data. Beyond just storage, its true power lies in the ability to extract invaluable insights from this data, paving the way for strategic decisions that fuel business growth. This system empowers enterprises to not only manage their information effectively but also to unlock actionable intelligence that propels them forward in the competitive landscape.

In essence, a Business Insights Management System is a versatile tool that spans various aspects of business operations, providing a holistic approach to data management and intelligence extraction. It is instrumental in shaping organizational strategies, improving efficiency, and staying competitive in dynamic markets.

Advantages of Business Insights Management System:

- Strategic Decision-Making: BIMS provides decision-makers with real-time and historical insights, enabling strategic decisions based on a comprehensive understanding of the business landscape.
- 2. **Operational Efficiency:** By streamlining data storage and retrieval, BIMS enhances operational efficiency, reducing the time and effort required for data analysis.
- 3. **Competitive Edge:** Leveraging insights from BIMS allows businesses to stay ahead of the competition by adapting quickly to market trends and customer preferences.
- **4. Resource Optimization:** BIMS enables efficient allocation of resources by providing insights into areas such as inventory management, workforce planning, and marketing expenditure.
- **5. Data Security and Compliance:** Robust security measures in BIMS ensure the confidentiality and integrity of sensitive business data, ensuring compliance with data protection regulations.

Use Cases:

- Sales and Marketing Optimization: BIMS helps businesses analyze customer behavior, preferences, and buying patterns, enabling targeted marketing campaigns and optimizing sales strategies.
- **2. Supply Chain Management:** BIMS assists in optimizing the supply chain by providing insights into inventory levels, demand forecasting, and logistics, thereby minimizing costs and improving efficiency.
- **3. Financial Analytics:** BIMS supports financial decision-making by providing real-time insights into revenue, expenses, and profitability, aiding in budgeting and financial planning.

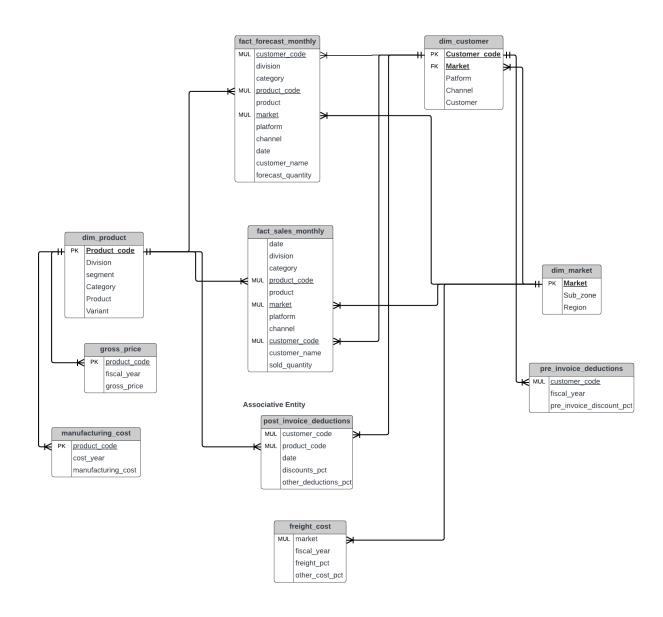
- **4.** Customer Relationship Management (CRM): BIMS helps in managing and analyzing customer data, improving customer satisfaction, and fostering long-term relationships through personalized services.
- **5. Risk Management:** BIMS aids in identifying and mitigating business risks by analyzing historical data, market trends, and external factors that could impact the organization.
- **6. Human Resources Analytics:** BIMS assists HR departments in workforce planning, talent management, and employee performance analysis, fostering a data-driven approach to human resources.
- 7. Quality Control and Process Improvement: BIMS supports quality control initiatives by analyzing production data, identifying inefficiencies, and facilitating continuous process improvement.

STEP 2: Conceptual Data Modeling and Database Design:

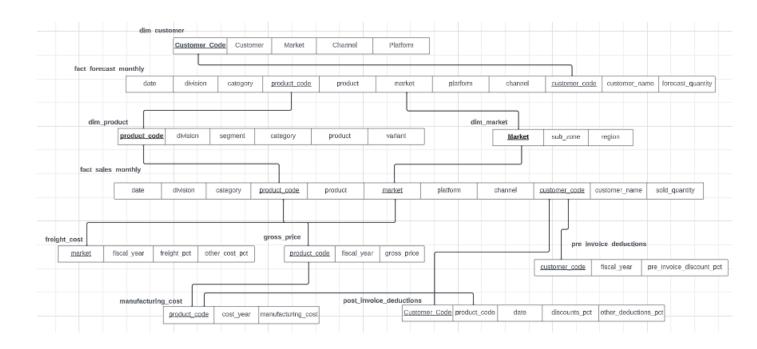
Business rules:

- 1. Ensures that each customer is uniquely identified within the system.
- 2. Ensures that each product is uniquely identified within the system.
- 3. Ensures that only valid dates are stored in the system.
- 4. Quantities and percentages cannot be negative in a real-world context.
- 5. Ensures that references to customers and products are valid.
- 6. Maintains consistency in representing fiscal years.
- 7. Ensures the accuracy and consistency of decimal values.
- 8. Ensures that essential information is captured for each record.
- 9. Ensures consistency and accuracy in recording platforms and channels.
- 10. Ensures that customers are associated with accurate geographical regions.

ER/EER Diagram:



Relational Model:



STEP 3: Database Implementation: SQL Commands to Create Tables for Database:

Below are the SQL commands that were used to create tables for the database The SQL commands used to create the tables for the database can also be found in an attached google drive text file.

MIS 686 Project DB Tables.txt

```
MIS 686 Project DB Tables - Notepad
<u>File Edit Format View Help</u>
CREATE DATABASE IF NOT EXISTS gdb041
USE gdb041;
CREATE TABLE dim_customer (
  customer varchar(255) NOT NULL,
  market varchar(255) NOT NULL,
  platform varchar(255) NOT NULL,
  channel varchar(255) NOT NULL,
  customer_code varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE dim market (
 market varchar(255) NOT NULL,
  sub_zone varchar(255) NOT NULL,
  region varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE dim_product (
  product_code varchar(255) NOT NULL,
  division varchar(255) NOT NULL,
  segment varchar(255) NOT NULL,
  category varchar(255) NOT NULL,
  product varchar(255) NOT NULL,
 variant varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE fact_forecast_monthly (
  date datetime NOT NULL,
  division varchar(255) NOT NULL,
  category varchar(255) NOT NULL,
  product_code varchar(255) NOT NULL,
  product varchar(255) NOT NULL,
```

```
MIS 686 Project DB Tables - Notepad
<u>File Edit Format View Help</u>
CREATE TABLE fact_forecast_monthly (
  date datetime NOT NULL,
  division varchar(255) NOT NULL,
  category varchar(255) NOT NULL,
  product_code varchar(255) NOT NULL,
  product varchar(255) NOT NULL,
  market varchar(255) NOT NULL,
  platform varchar(255) NOT NULL,
  channel varchar(255) NOT NULL,
  customer_code varchar(255) NOT NULL,
  customer_name varchar(255) NOT NULL,
  forecast quantity int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE fact_sales_monthly (
  date datetime NOT NULL,
  division varchar(255) NOT NULL,
  category varchar(255) NOT NULL,
  product_code varchar(255) NOT NULL,
  product varchar(255) NOT NULL,
  market varchar(255) NOT NULL,
  platform varchar(255) NOT NULL,
  channel varchar(255) NOT NULL,
  customer_code varchar(255) NOT NULL,
  customer_name varchar(255) NOT NULL,
  sold_quantity int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE freight_cost (
  market varchar(255) NOT NULL,
  fiscal_year varchar(10) NOT NULL,
  freight_pct decimal(15,10) NOT NULL,
  other_cost_pct decimal(15,10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
MIS 686 Project DB Tables - Notepad
<u>File Edit Format View Help</u>
CREATE TABLE freight_cost (
  market varchar(255) NOT NULL,
  fiscal year varchar(10) NOT NULL,
  freight_pct decimal(15,10) NOT NULL,
  other_cost_pct decimal(15,10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE gross_price (
  product_code varchar(255) NOT NULL,
  fiscal_year varchar(10) NOT NULL,
  gross_price decimal(15,10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE manufacturing_cost (
  product_code varchar(255) NOT NULL,
  cost_year varchar(255) NOT NULL,
  manufacturing_cost decimal(15,10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE post invoice deductions (
  customer_code varchar(255) NOT NULL,
  product_code varchar(255) NOT NULL,
  date date NOT NULL,
  discounts_pct decimal(15,10) NOT NULL,
  other_deductions_pct decimal(15,10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE pre invoice deductions (
  customer_code varchar(255) NOT NULL,
  fiscal_year varchar(10) NOT NULL,
  pre_invoice_discount_pct decimal(25,20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

SQL Commands to Insert Data Into Database:

A copy of all the insert into commands can be found in the attached text file

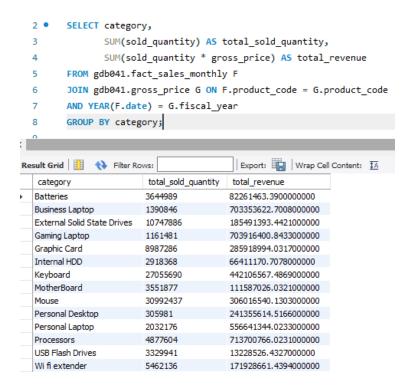
MIS 686 Project DB Tables Data.txt

Analytical Questions and SQL Queries:

1. Sales Performance Metrics:

Calculate the total sold quantity and revenue for a specific product category across all markets and channels.

```
SELECT category,
SUM(sold_quantity) AS total_sold_quantity,
SUM(sold_quantity * gross_price) AS total_revenue
FROM gdb041.fact_sales_monthly F
JOIN gdb041.gross_price G ON F.product_code = G.product_code
AND YEAR(F.date) = G.fiscal_year
GROUP BY category;
```



2. Demand Analysis:

What is the total forecasted quantity for a specific product in a given market and platform for the each year?

```
SELECT Year(date), SUM(forecast_quantity) AS total_forecast
FROM gdb041.fact_forecast_monthly
WHERE product_code = 'A6218160101'
AND market = 'Australia'
 AND platform = 'E-Commerce'
group by YEAR(date);
        SELECT Year(date), SUM(forecast_quantity) AS total_forecast
 10 •
        FROM gdb041.fact forecast monthly
 11
        WHERE product_code = 'A6218160101'
 12
          AND market = 'Australia'
 13
 14
          AND platform = 'E-Commerce'
 15
         group by YEAR(date);
                                         Export: Wrap Cell Content: TA
Year(date) total_forecast
  2017
            1086
  2018
            3345
  2019
            6953
  2020
            4811
  2021
            2337
```

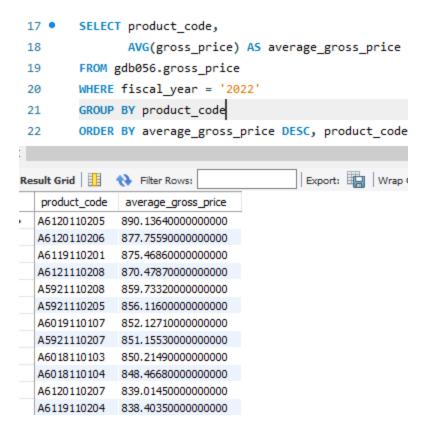
3. Pricing Strategy:

What is the average gross price for each product in the year 2022? SELECT product_code, AVG(gross_price) AS average_gross_price FROM gdb056.gross_price

WHERE fiscal_year = '2022'

GROUP BY product_code

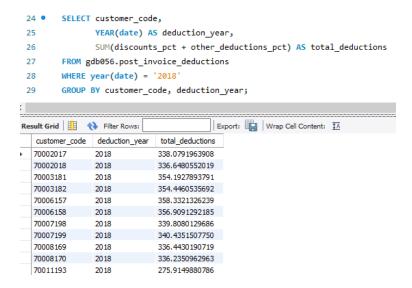
ORDER BY average_gross_price DESC, product_code;



4. Customer Deductions:

Identify the total deductions and discounts for customers in the fiscal year 2018.

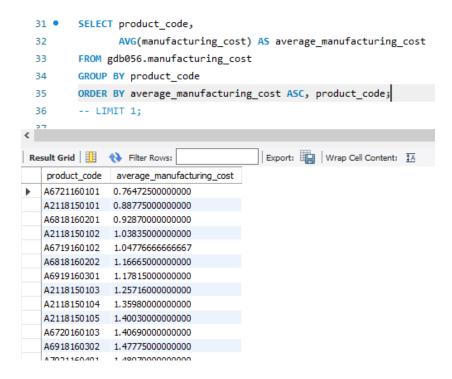
```
SELECT customer_code,
YEAR(date) AS deduction_year,
SUM(discounts_pct + other_deductions_pct) AS total_deductions
FROM gdb056.post_invoice_deductions
WHERE year(date) = '2018'
GROUP BY customer_code, deduction_year;
```



5. Supply Chain Optimization:

Analyze the manufacturing costs for each product and identify products with the highest and lowest manufacturing costs.

SELECT product_code,
 AVG(manufacturing_cost) AS average_manufacturing_cost
FROM gdb056.manufacturing_cost
GROUP BY product_code
ORDER BY average_manufacturing_cost ASC, product_code;



If we want to see product with the lowest manufacturing cost.

```
31 •
         SELECT product_code,
                 AVG(manufacturing cost) AS average manufacturing cost
 32
         FROM gdb056.manufacturing cost
 33
         GROUP BY product code
 34
         ORDER BY average_manufacturing_cost ASC, product_code
 35
 36
         LIMIT 1;
Result Grid Filter Rows:
                                             Export: Wrap Cell Content: $\overline{A}$ Fetch rows:
   product code | average manufacturing cost
 A6721160101 0.76472500000000
```

6. Freight Cost Analysis:

Calculate the total freight costs for each market in the current fiscal year 2022?

```
SELECT market,

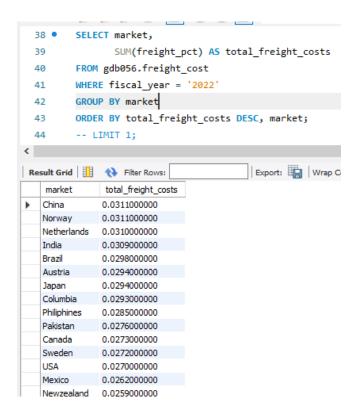
SUM(freight_pct) AS total_freight_costs

FROM gdb056.freight_cost

WHERE fiscal_year = '2022'

GROUP BY market

ORDER BY total_freight_costs DESC, market;
```



7. Top Products by Revenue:

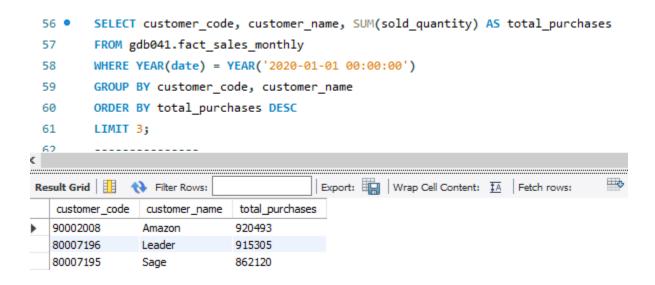
```
What are the top 10 products based on total revenue in the year 2021?
SELECT F.product_code,
    SUM(F.sold_quantity * G.gross_price) AS total_revenue
FROM gdb041.fact_sales_monthly F
JOIN gdb041.gross_price G ON F.product_code = G.product_code
AND G.fiscal_year = YEAR(F.date)
WHERE YEAR(date) = YEAR('2021-01-01 00:00:00')
GROUP BY product_code
ORDER BY total revenue DESC
LIMIT 10:
 46
         SELECT F.product code,
 47
                 SUM(F.sold_quantity * G.gross_price) AS total_revenue
         FROM gdb041.fact_sales_monthly F
 48
         JOIN gdb041.gross_price G ON F.product_code = G.product_code
 49
         AND G.fiscal year = YEAR(F.date)
 50
 51
         WHERE YEAR(date) = YEAR('2021-01-01 00:00:00')
 52
         GROUP BY product code
         ORDER BY total revenue DESC
 53
 54
         LIMIT 10;
 55
                                           Export: Wrap Cell Content: A Fe
Result Grid
              Filter Rows:
   product_code | total_revenue
   A1521150601 48218625.8778000000
   A1521150602 47276510.4085000000
   A1421150503 46042613.1495000000
   A1420150502 43221627.0005000000
   A1420150501 42344631.2912000000
   A1320150403 41853308.5696000000
   A1320150402 38423656.3000000000
   A1219150303 38247224.0150000000
   A1219150302 37343668.2194000000
   A1319150401 36002357.8606000000
```

8. Top 3 Customers by Purchases:

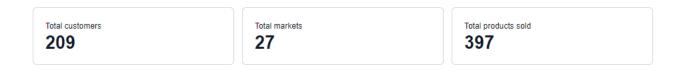
Who are the top 3 customers based on total purchases in the year 2020?

```
SELECT customer_code, customer_name, SUM(sold_quantity) AS total_purchases FROM gdb041.fact_sales_monthly WHERE YEAR(date) = YEAR('2020-01-01 00:00:00') GROUP BY customer_code, customer_name
```

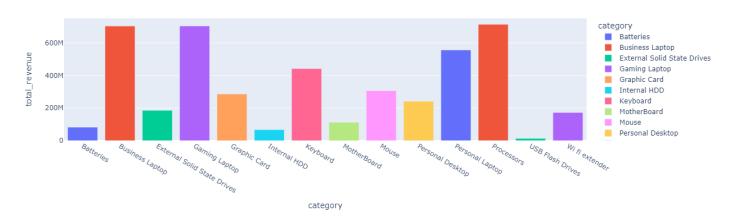
ORDER BY total_purchases DESC LIMIT 3;



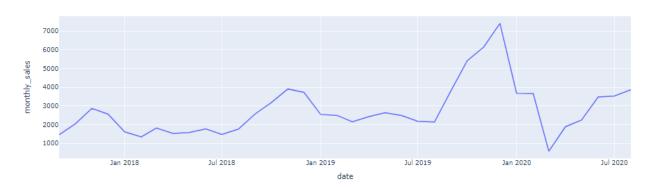
STEP 4: Enterprise (web) Database Dashboard:



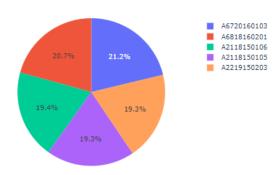
Total Sales by Product Category



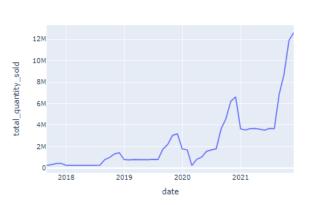
Monthly Sales Trends for a Specific Product



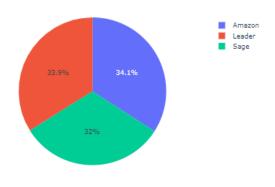
Market Share of Top Products



Sales Trends Over Time



Top 3 Customers by Purchases



Distribution of Manufacturing Costs by Product

