Supply Chain Finance Management

**Introduction**

This project is focused on designing and implementing a database management system for sales, inventory, and cost tracking in a business environment. The goal is to efficiently track the movement of products, analyse sales data, calculate manufacturing and freight costs, and manage inventory levels. The system utilizes SQL queries, stored procedures, triggers, and user-defined functions to automate and optimize various business processes, providing insights into key performance indicators (KPIs) such as sales, profit margins, inventory turnover, and more.

The data model is based on a relational structure, connecting multiple tables such as products, sales, customers, cost factors, and freight costs. This enables complex analysis and reporting and facilitates inventory management and cost optimization.

**ERD Explanation**

The **Entity-Relationship Diagram (ERD)** represents the database schema and the relationships between the various entities involved in the sales and inventory management system. The key tables in the system are:

1. **dim\_product**: Contains product-related details, including product code, name, category, and variant.
2. **fact\_sales\_monthly**: Tracks monthly sales data, such as the quantity sold, fiscal year, and product code.
3. **fact\_forecast\_monthly**: Contains forecasted sales data for each product, allowing comparisons with actual sales.
4. **fact\_gross\_price**: Holds the gross prices of products for each fiscal year, used to calculate revenue.
5. **fact\_manufacturing\_cost**: Contains manufacturing costs for each product.
6. **fact\_freight\_cost**: Tracks freight costs by market and fiscal year.
7. **dim\_customer**: Contains customer-related information, such as customer code, market, and platform.
8. **fact\_post\_invoice\_deductions** and **fact\_pre\_invoice\_deductions**: Handle invoice-related deductions and discounts.

**Relationships**:

* **dim\_product** is connected to **fact\_sales\_monthly**, **fact\_forecast\_monthly**, **fact\_gross\_price**, and **fact\_manufacturing\_cost** via the product\_code.
* **fact\_sales\_monthly** is linked to **dim\_customer** by the customer\_code, enabling sales tracking per customer.
* **fact\_freight\_cost** and **fact\_manufacturing\_cost** are connected to the sales data for calculating costs based on fiscal year and product.
* **fact\_post\_invoice\_deductions** and **fact\_pre\_invoice\_deductions** handle product-level and customer-level financial adjustments.

The relationships are designed to facilitate the accurate and efficient flow of data between tables, ensuring that key business operations such as sales tracking, inventory management, and cost calculations can be automated and analyzed in real-time.

A screenshot of a computer

Description automatically generated

**Task-3**

**Q1) Assume calendar\_date is '2023-07-15'. Apply the function to this date and explain what value it will return as the fiscal year.**

**select year(date) as fiscal\_year from fact\_sales\_monthly**

update fact\_sales\_monthly

set fiscal\_year = YEAR(date)

**Q2) Analyzing Gross Sales: Monthly Product Transactions Report**

**Write a Query for making report on monthly product transactions, including details such as date, product code,**

**product name, variant, sold quantity, gross price, and gross price total. The query should involves joining several tables and**

**filtering results based on customer code and fiscal year.**

select

p.product\_code,

p.product,

p.variant,

fsm.sold\_quantity,

fgp.gross\_price,

(fsm.sold\_quantity \* fgp.gross\_price) as gross\_price\_total

from fact\_sales\_monthly fsm

join dim\_product p on p.product\_code = fsm.product\_code

join fact\_forecast\_monthly ffm on ffm.product\_code = p.product\_code

join fact\_gross\_price fgp on fgp.product\_code = p.product\_code

join dim\_customer c on c.customer\_code = ffm.customer\_code

**Task-4**

**Sales Trend Analysis:**

**Query the fact\_monthly\_sales table to identify the monthly sales trend for each product. How do the sales volumes fluctuate over time?**

select p.product, p.variant ,DATENAME(MONTH, fsm.date) as month, year(date) as year, sum(fsm.sold\_quantity) as total\_sold\_quantity from fact\_sales\_monthly fsm

join dim\_product p on p.product\_code = fsm.product\_code

group by p.product, p.variant, DATENAME(MONTH, fsm.date), year(date)

order by year,month asc, total\_sold\_quantity desc

**Customer Segmentation:**

**Utilizing the dim\_customer table, segment customers based on their purchasing behavior. Which customer segments contribute the most to sales revenue?**

select c.channel,c.region,c.market,round(SUM(fsm.sold\_quantity \* fgp.gross\_price),2) as revenue from dim\_customer c

join fact\_sales\_monthly fsm on fsm.customer\_code = c.customer\_code

join fact\_gross\_price fgp on fgp.product\_code = fsm.product\_code

group by c.channel,c.region,c.market

**Product Performance Comparison:**

**Compare the performance of products in terms of sales quantity and revenue generated. Which products are the top performers, and which ones need improvement?**

select p.product,p.variant,SUM(fsm.sold\_quantity) AS total\_sold\_quantity,round(cast(SUM(fsm.sold\_quantity \* fgp.gross\_price) as float),2) as revenue from dim\_product p

join fact\_sales\_monthly fsm on p.product\_code = fsm.product\_code

join fact\_gross\_price fgp on fgp.product\_code = fsm.product\_code

group by p.product,p.variant

order by revenue desc, total\_sold\_quantity desc

**Market Expansion Opportunities:**

**Analyze the fact\_forecast\_monthly table to identify potential market expansion opportunities. Which markets show the highest forecasted demand growth?**

select c.market,ffm.fiscal\_year, SUM(ffm.forecast\_quantity) as forcast\_quantity from dim\_customer c

join fact\_forecast\_monthly ffm on ffm.customer\_code =c.customer\_code

group by c.market,ffm.fiscal\_year

order by ffm.fiscal\_year, forcast\_quantity desc

**Cost Analysis:**

**Calculate the total manufacturing cost for each product and compare it with the gross price to determine profitability. Which products have the highest profit margins?**

select p.product,p.variant, fmc.cost\_year,fgp.gross\_price,fmc.manufacturing\_cost,

(fgp.gross\_price-fmc.manufacturing\_cost) as profit\_per\_unit,

round(cast((fgp.gross\_price-fmc.manufacturing\_cost)/fgp.gross\_price \* 100 as float),2) as profit\_percentage

from dim\_product p

join fact\_manufacturing\_cost fmc on fmc.product\_code = p.product\_code

join fact\_gross\_price fgp on fgp.product\_code = p.product\_code

order by profit\_percentage desc

**Discount Impact Analysis:**

**Assess the impact of pre-invoice discounts on sales revenue.**

**How do varying discount levels affect overall revenue and customer retention?**

with revenue\_discount as (

select c.customer, c.customer\_code,fsm.fiscal\_year,p.product,p.product\_code,p.variant,fsm.sold\_quantity,fpid.pre\_invoice\_discount\_pct,

round(cast( (fsm.sold\_quantity \* fgp.gross\_price) as float),2) as Total\_revenue,

round(cast((fsm.sold\_quantity\* fgp.gross\_price \* (1 - fpid.pre\_invoice\_discount\_pct)) as float),2) as Revenue\_after\_discount

from dim\_customer c

join fact\_sales\_monthly fsm on fsm.customer\_code = c.customer\_code

join fact\_pre\_invoice\_deductions fpid on fpid.customer\_code = c.customer\_code

join dim\_product p on p.product\_code = fsm.product\_code

join fact\_gross\_price fgp on fgp.product\_code = p.product\_code

),

customerRetention as(

select customer\_code, count(distinct fiscal\_year) as retention\_years from fact\_sales\_monthly

group by customer\_code

)

select rd.customer,rd.fiscal\_year, cr.retention\_years ,

SUM(rd.Total\_revenue) as total\_gross\_revenue,

SUM(rd.Revenue\_after\_discount) as total\_revenue\_after\_discount,

avg(rd.pre\_invoice\_discount\_pct) \* 100 as avg\_dis\_percentage

from customerRetention cr

join revenue\_discount rd on rd.customer\_code = cr.customer\_code

group by rd.customer,rd.fiscal\_year,cr.retention\_years

**Market-specific Freight Costs:**

**Determine the average freight costs for different markets over the years. Are there any noticeable trends or outliers in freight expenses?**

select market, fiscal\_year, round(cast(AVG(freight\_pct)\*100 as float),0) as avg\_freight\_cost\_pct, round(cast(AVG(other\_cost\_pct)\*100 as float),0) as avg\_other\_cost\_pct,round(cast(AVG(freight\_pct+other\_cost\_pct) \*100 as float),0) as avg\_total\_cost\_pct

from fact\_freight\_cost

group by Market,fiscal\_year

**Seasonal Sales Patterns:**

**Explore the fact\_monthly\_sales table to identify seasonal sales patterns. How do sales volumes vary throughout the year, and are there any recurring trends?**

select DATENAME(MONTH, date) AS month\_name ,fiscal\_year,

SUM(sold\_quantity) as total\_sales\_volume

from fact\_sales\_monthly

group by fiscal\_year, DATENAME(MONTH, date);

**Customer Loyalty Analysis:**

**Analyze customer purchase frequency and retention rates over time. Which customers exhibit the highest levels of loyalty, and how can their behavior be leveraged for targeted marketing campaigns?**

with CustomerPurchaseFrequency as (

select customer\_code, COUNT(distinct date) as Purchase\_frequency, MIN(date) as first\_purchase, MAX(date) as last\_purchase,

DATEDIFF(DAY,MIN(date),MAX(date)) as customer\_life\_span\_days, COUNT(distinct fiscal\_year) as active\_years

from fact\_sales\_monthly

group by customer\_code

)

select c.customer,c.customer\_code,c.market,cpf.Purchase\_frequency,cpf.active\_years

from dim\_customer c

join CustomerPurchaseFrequency cpf on cpf.customer\_code = c.customer\_code

order by cpf.purchase\_frequency DESC;

**Forecast Accuracy Evaluation:**

**Evaluate the accuracy of sales forecasts by comparing forecasted quantities with actual sales data. Are there any significant discrepancies, and how can forecast models be improved?**

with ForecastComparison as(

select ffm.date, ffm.product\_code, p.product, ffm.customer\_code, c.customer, ffm.forecast\_quantity,

coalesce(SUM(fsm.sold\_quantity),0) as total\_sold\_quantity,

(coalesce(SUM(fsm.sold\_quantity),0) -ffm.forecast\_quantity) as forecast\_error

from fact\_forecast\_monthly ffm

left join fact\_sales\_monthly fsm on ffm.customer\_code = fsm.customer\_code and ffm.date = fsm.date and ffm.product\_code = fsm.product\_code

join dim\_customer c on c.customer\_code = ffm.customer\_code

join dim\_product p on p.product\_code = ffm.product\_code

group by ffm.date, ffm.product\_code, p.product, ffm.customer\_code, c.customer, ffm.forecast\_quantity

),

ForecastError as(

select product, product\_code,customer\_code, customer,

round(SUM(case

when total\_sold\_quantity >0 then ABS(cast(forecast\_error as float))/total\_sold\_quantity

else null

end)/count(\*) \* 100,2) as error\_pct

from ForecastComparison

group by product, product\_code,customer\_code, customer

)

select fc.product, fc.product\_code,fc.customer\_code, fc.customer,fc.total\_sold\_quantity,fc.forecast\_quantity,fc.forecast\_error, fe.error\_pct

from ForecastComparison fc

join ForecastError fe on fc.customer\_code=fe.customer\_code and fc.product\_code = fe.product\_code

**Channel Performance Assessment:**

**Compare sales performance across different sales channels (e.g., E-Commerce vs. Brick & Mortar). Which channels are most effective in driving sales, and are there any opportunities for**

**optimization?**

select c.channel as sales\_channel, count(distinct fsm.customer\_code) as total\_customers, SUM(fsm.sold\_quantity) as total\_sold\_units,

cast(SUM(fsm.sold\_quantity \* fgp.gross\_price)as float) as total\_revenue,

round(cast(SUM(fsm.sold\_quantity \* fgp.gross\_price)as float)/SUM(fsm.sold\_quantity),2) as avg\_price\_per\_unit

from dim\_customer c

join fact\_sales\_monthly fsm on fsm.customer\_code = c.customer\_code

join fact\_gross\_price fgp on fgp.product\_code = fsm.product\_code and fgp.fiscal\_year = fsm.fiscal\_year

group by c.channel;

**Geographical Sales Distribution:**

**Analyze sales distribution across different geographical regions. How does sales performance vary by region, and are there any emerging markets worth focusing on?**

with TotalSales as (

select SUM(fsm.sold\_quantity \* fgp.gross\_price) as total\_revenue,

SUM(fsm.sold\_quantity) as total\_sold\_unit

from fact\_sales\_monthly fsm

join fact\_gross\_price fgp on fgp.fiscal\_year = fsm.fiscal\_year and fgp.product\_code = fsm.product\_code

)

select c.region,c.market,SUM(fsm.sold\_quantity) as total\_sold\_units, SUM(fsm.sold\_quantity \* fgp.gross\_price) as total\_revenue,

round(cast(SUM(fsm.sold\_quantity \* fgp.gross\_price) as float) \* 100.0 / (select total\_revenue from totalSales),2) as revenue\_pct,

round(cast(SUM(fsm.sold\_quantity)as float) \*100.0/(select total\_sold\_unit from totalSales),2) as quantity\_pct

from dim\_customer c

join fact\_sales\_monthly fsm on fsm.customer\_code = c.customer\_code

join fact\_gross\_price fgp on fgp.product\_code = fsm.product\_code and fsm.fiscal\_year = fgp.fiscal\_year

group by c.region,c.market

order by total\_revenue desc

**Customer Acquisition Cost Analysis:**

**Calculate the customer acquisition cost (CAC) for each market and channel. Which acquisition channels provide the highest return on investment (ROI), and where should resources be**

**allocated for customer acquisition?**

with customer\_first\_purchase as(

select customer\_code, min(date) as first\_purchase from fact\_sales\_monthly

group by customer\_code

),

new\_customer as(

select c.customer\_code,c.market,c.channel from dim\_customer c

join customer\_first\_purchase cfp on c.customer\_code = cfp.customer\_code

),

new\_customer\_revenue as (

select nc.market,nc.channel,

SUM(fsm.sold\_quantity \* fgp.gross\_price) AS revenue\_by\_new\_customers,

COUNT(nc.customer\_code) as new\_customer\_count

from new\_customer nc

join fact\_sales\_monthly fsm on fsm.customer\_code = nc.customer\_code

join fact\_gross\_price fgp on fgp.product\_code = fsm.product\_code and fsm.fiscal\_year = fgp.fiscal\_year

group by nc.market, nc.channel

)

select market,channel,revenue\_by\_new\_customers, new\_customer\_count,

ROUND(cast(revenue\_by\_new\_customers \* 0.2 as float), 2) AS estimated\_acquisition\_cost, Assume 20% of revenue for acquisition

ROUND(cast(revenue\_by\_new\_customers \* 0.2 / new\_customer\_count as float), 2) AS customer\_acquisition\_cost,

ROUND(cast((revenue\_by\_new\_customers - (revenue\_by\_new\_customers \* 0.2)) \* 100.0 as float) / (revenue\_by\_new\_customers \* 0.2), 2) AS roi\_percentage

from new\_customer\_revenue

order by roi\_percentage desc

**Product Mix Optimization:**

**Determine the optimal product mix based on sales volume, profitability, and market demand.**

**How can product portfolios be adjusted to maximize overall revenue?**

with productSales as(

select p.product\_code, p.product, p.variant,fsm.fiscal\_year ,c.market,c.region,c.channel ,

round(cast(SUM(fsm.sold\_quantity)as float),2) as total\_sold\_quantity,

round(cast(SUM(fsm.sold\_quantity \* fgp.gross\_price)as float),2) AS total\_revenue,

round(cast(SUM(fsm.sold\_quantity \* fmc.manufacturing\_cost)as float),2) AS total\_cost

from dim\_product p

join fact\_sales\_monthly fsm on fsm.product\_code = p.product\_code

join fact\_gross\_price fgp on fgp.product\_code = p.product\_code and fgp.fiscal\_year = fsm.fiscal\_year

join fact\_manufacturing\_cost fmc on fmc.product\_code = p.product\_code and fmc.cost\_year = fsm.fiscal\_year

join dim\_customer c on c.customer\_code = fsm.customer\_code

group by p.product\_code, p.product, p.variant,fsm.fiscal\_year ,c.market,c.region,c.channel

)

Select product\_code , product,fiscal\_year,variant,total\_sold\_quantity,total\_revenue,total\_cost ,region,market,channel ,

(total\_revenue - total\_cost) AS total\_profit,

ROUND(cast((total\_revenue - total\_cost) \* 100.0 / total\_revenue as float), 2) AS profit\_margin

FROM ProductSales

order by fiscal\_year,total\_profit desc

**Customer Lifetime Value Calculation:**

**Calculate the customer lifetime value (CLV) for each customer segment. Which segments are the most valuable in terms of long-term revenue generation?**

with customer\_revenue as (

select c.customer,c.customer\_code,c.market,c.region,

round(cast(SUM(fsm.sold\_quantity \* fgp.gross\_price)as float),2) AS total\_revenue,

COUNT(DISTINCT YEAR(fsm.date)) AS active\_years

from dim\_customer c

join fact\_sales\_monthly fsm on fsm.customer\_code = c.customer\_code

join fact\_gross\_price fgp on fgp.product\_code = fsm.product\_code and fgp.fiscal\_year =fsm.fiscal\_year

GROUP BY c.customer,c.customer\_code, c.market, c.region

)

select market,region,

round(cast(avg(total\_revenue) as float),2) as Average\_revenue\_per\_cus,

ROUND(cast(sum(total\_revenue) as float),2) as Total\_market\_revenue,

ROUND(AVG(total\_revenue / NULLIF(active\_years, 0)), 2) AS avg\_revenue\_per\_year,

COUNT(DISTINCT customer\_code) AS total\_customers

from customer\_revenue

GROUP BY market, region

ORDER BY total\_market\_revenue DESC, Average\_revenue\_per\_cus DESC;

**Task-5**

**1. Define a user-defined function to calculate the total forecasted quantity for a given product and fiscal year.**

CREATE FUNCTION fn\_total\_forecast\_quantity (

@product\_id varchar(10), Product ID for which forecast is calculated

@fiscal\_year INT Fiscal year to filter the forecasted data

)

returns int

as

begin

declare @total\_forecast int;

select @total\_forecast = SUM(forecast\_quantity)

from fact\_forecast\_monthly

where product\_code = @product\_id and fiscal\_year= @fiscal\_year

RETURN @total\_forecast;

END;

select dbo.fn\_total\_forecast\_quantity('P001', 2022) as forcasted\_result

**2. Write a query to find the customers who made purchases exceeding the average monthly sales quantity across all products.**

with avg\_montly\_sales\_cte as(

select avg(monthly\_sales) as avg\_monthly\_sales from

(select fiscal\_year, MONTH(date) as month, SUM(sold\_quantity) as monthly\_sales from fact\_sales\_monthly group by fiscal\_year, MONTH(date)) AS MonthlySales)

select c.customer,c.customer\_code,

SUM(fsm.sold\_quantity) as total\_purchase\_quantity,

(select avg\_monthly\_sales from avg\_montly\_sales\_cte) as average\_monthly\_sales

from dim\_customer c

join fact\_sales\_monthly fsm on fsm.customer\_code = c.customer\_code

group by c.customer,c.customer\_code

having SUM(fsm.sold\_quantity) > (select avg\_monthly\_sales from avg\_montly\_sales\_cte)

**3. Create a stored procedure to update the gross price of a product for a specific fiscal year.**

CREATE PROCEDURE UpdateGrossPrice

@product\_id VARCHAR(10),

@fiscal\_year INT,

@new\_gross\_price DECIMAL(18, 2)

AS

begin

update fact\_gross\_price

set gross\_price = @new\_gross\_price

where fiscal\_year = @fiscal\_year and product\_code = @product\_id

end;

EXEC UpdateGrossPrice @product\_id = 'P001', @fiscal\_year = 2022, @new\_gross\_price = 150.75;

**4. Implement a trigger that automatically inserts a record into the audit log table whenever a new entry is added to the sales table.**

**5. Use a window function to rank products based on their monthly sales quantity, partitioned by fiscal year.**

select p.product,p.variant ,p.product\_code,fsm.fiscal\_year ,sum(fsm.sold\_quantity)as total\_sold\_quantity,

rank() over ( partition by fsm.fiscal\_year order by sum(fsm.sold\_quantity) desc) as rank

from dim\_product p

join fact\_sales\_monthly fsm on fsm.product\_code = p.product\_code

group by p.product,p.variant ,p.product\_code, fsm.fiscal\_year

**6. Utilize the STRING\_AGG function to concatenate the names of all customers who purchased a specific product within a given timeframe.**

select p.product, STRING\_AGG(c.customer,',') as customers from dim\_customer c

join fact\_sales\_monthly fsm on fsm.customer\_code = c.customer\_code

join dim\_product p on p.product\_code = fsm.product\_code

where p.product\_code = 'P002' and date between '2022-01-01' and '2022-01-04'

group by p.product

**7. Develop a user-defined function that calculates the total manufacturing cost for a product over a specified range of years, using a subquery to retrieve the necessary data.**

CREATE FUNCTION fn\_total\_manufacturing\_cost (

@product\_code VARCHAR(10),

@start\_year INT,

@end\_year INT

)

RETURNS DECIMAL(10,2)

AS

BEGIN

DECLARE @total\_cost DECIMAL(10,2);

SELECT @total\_cost = SUM(manufacturing\_cost)

FROM fact\_manufacturing\_cost

WHERE product\_code = @product\_code

AND cost\_year BETWEEN @start\_year AND @end\_year;

RETURN @total\_cost;

END;

SELECT dbo.fn\_total\_manufacturing\_cost('P001', 2020, 2024) AS total\_cost;

**8. Design a stored procedure to insert new records into the sales table and use a trigger to enforce constraints on the quantity sold, ensuring it doesn't exceed the available inventory.**

**9. Apply the LEAD or LAG function to compare monthly sales quantities of a product with the previous month's sales.**

SELECT fsm.fiscal\_year,

MONTH(fsm.date) AS month,

fsm.product\_code,

SUM(fsm.sold\_quantity) AS current\_month\_sales,

LAG(SUM(fsm.sold\_quantity), 1) OVER (PARTITION BY fsm.product\_code ORDER BY fsm.fiscal\_year, MONTH(fsm.date)) AS previous\_month\_sales,

SUM(fsm.sold\_quantity) - LAG(SUM(fsm.sold\_quantity), 1) OVER (PARTITION BY fsm.product\_code ORDER BY fsm.fiscal\_year, MONTH(fsm.date)) AS sales\_diff

FROM fact\_sales\_monthly fsm

GROUP BY fsm.fiscal\_year,MONTH(fsm.date),fsm.product\_code

ORDER BY fsm.product\_code, fsm.fiscal\_year, MONTH(fsm.date);

**10. Create a query to identify the top-selling products in each market based on their total sales quantity, utilizing subqueries and window functions.**

select product\_code, product, variant, market, total\_sales\_quantity, sales\_rank

from (

select p.product\_code, p.product, p.variant, c.market,

sum(fsm.sold\_quantity) as total\_sales\_quantity,

rank() over (partition by c.market order by sum(fsm.sold\_quantity) desc) as sales\_rank

from dim\_product p

join fact\_sales\_monthly fsm on p.product\_code = fsm.product\_code

join dim\_customer c on c.customer\_code = fsm.customer\_code

group by p.product\_code, p.product, p.variant, c.market

) as ranked\_products

where sales\_rank = 1

order by market, sales\_rank;

**11. Develop a user-defined function to calculate the total freight cost for a product based on its market and fiscal year. Then, integrate this function into a stored procedure to**

**update the overall cost.**

create function dbo.fn\_total\_freight\_cost (

@market varchar(45),

@fiscal\_year int

)

returns decimal(18, 4)

as

begin

declare @total\_freight\_cost decimal(18, 4);

select @total\_freight\_cost = sum(freight\_pct + other\_cost\_pct)

from fact\_freight\_cost

where market = @market

and fiscal\_year = @fiscal\_year;

return isnull(@total\_freight\_cost, 0);

end;

create procedure dbo.sp\_update\_total\_cost

@product\_code varchar(50),

@market varchar(45),

@fiscal\_year int,

@base\_cost decimal(18, 4)

as

begin

declare @freight\_cost decimal(18, 4);

declare @total\_cost decimal(18, 4);

set @freight\_cost = dbo.fn\_total\_freight\_cost(@market, @fiscal\_year);

set @total\_cost = @base\_cost + @freight\_cost;

update fact\_manufacturing\_cost

set manufacturing\_cost = @total\_cost

where product\_code = @product\_code

and cost\_year = @fiscal\_year;

select @total\_cost as updated\_total\_cost;

end;

exec dbo.sp\_update\_total\_cost 'P001', 'north america', 2024, 1200.00;

select \* from fact\_manufacturing\_cost

where product\_code = 'P001' and cost\_year = 2024;