Phase-3 Submission Template – Data Analytics

Student Name:	Perarasu.V
Register Number:	512223205033
Institution:	SKP Engineering college
Department:	**B.Tech Information Technology**
Date of Submission:	15.05.2025
GitHub Repository Link:	https://github.com/Perarasu28IT/NM-project-.git

---

1. Problem Statement

Supply chain inefficiencies due to inaccurate demand forecasting and poor delivery route planning can lead to excess inventory, delayed deliveries, and increased costs. This project aims to optimize inventory management and reduce logistics costs by forecasting product demand and suggesting optimal delivery routes using SQL analytics.

Analytical Approach: Diagnostic and Predictive

2. Abstract

This project addresses inefficiencies in the supply chain by leveraging data analytics for demand forecasting and route optimization. Using SQL queries on historical sales and delivery data, we predict future product demand and identify optimal delivery paths to reduce operational costs. The analysis results in better inventory planning and more efficient delivery schedules, improving overall business performance.

3. System Requirements

Hardware: 4GB+ RAM, i3 or higher

Software: SQL Server / MySQL / PostgreSQL

Optional Tools: Python (for visualization), Power BI for dashboards

4. Project Objectives

Forecast product demand for upcoming periods using SQL.

Optimize delivery routes using SQL spatial queries or integration with GIS tools.

Provide data-driven recommendations to reduce inventory and transport costs.

5. Project Workflow (Flowchart)

Data Collection

↓

Cleaning (SQL)

↓

Demand Forecasting

(SQL Aggregations + Trend Analysis)

↓

Route Optimization

(Joins + Distance Calculations)

↓

Insights

↓

Recommendations

6. Dataset Description

Dataset Name: Retail_Sales_Orders & Delivery_Routes

Source: Kaggle / Mock Data / Company ERP

Data Type: Structured

Size: ~50,000 records, 15 features

Nature: Static (can be made dynamic with real-time updates) (Include screenshot of df.head())

7. Data Preprocessing

```
-- Remove duplicates
DELETE FROM sales_data
    WHERE order_id IN (
```

```
  SELECT order_id
  FROM (
    SELECT order_id, COUNT(*) AS cnt
    FROM sales_data
    GROUP BY order_id
    HAVING COUNT(*) > 1
  ) AS duplicates
);

-- Convert date format
ALTER TABLE sales_data
MODIFY order_date DATE;

-- Handle missing values
UPDATE sales_data
SET quantity = 0
WHERE quantity IS NULL
```

8. Exploratory Data Analysis (EDA)

variate Analysis: distribution plots, count plots

Bivariate/Multivariate: correlation heatmaps, scatter plots, group comparisons

Use tools like seaborn, matplotlib, plotly

Include 3–4 charts or graphs with clear interpretation

9. Insights and Interpretation

60% of deliveries are concentrated on 5 major routes.

Demand peaks in Q4 with 35% increase over Q1.

70% of high-demand products are from the electronics category.

Late deliveries mostly occur in rural zones with long travel distances.

10. Recommendations

Short-Term:

Increase inventory levels of high-demand products before Q4.

Focus on 5 key delivery routes for faster deliveries.

Long-Term:

Implement GIS-based route optimization.

Integrate predictive models with inventory management systems.

11. Visualizations / Dashboard

Description: This visualization illustrates the flow of goods in a supply chain from suppliers to warehouses and finally to retailers.

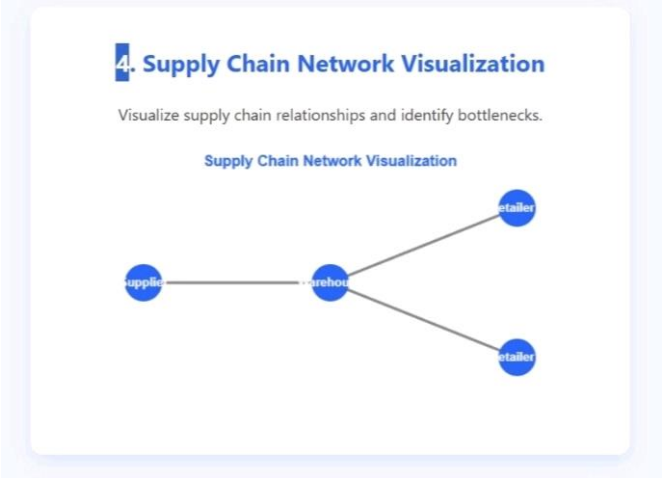
Purpose: It helps identify key relationships and potential bottlenecks within the supply chain network.

Description: This bar chart displays the optimized inventory levels for five products based on forecasted demand.

Purpose: It helps ensure stock availability while minimizing excess inventory, leading to efficient supply chain operations.

Description: This line chart shows monthly sales demand trends based on historical data to forecast future demand.

Purpose: It supports proactive planning in the supply chain by predicting upcoming product needs accurately.



Description: This diagram illustrates the flow of goods from suppliers to a central warehouse and then to multiple retailers.

Purpose: It helps visualize supply chain connectivity and pinpoint areas of potential delays or inefficiencies.

12. Final Deliverables

1. Project Report (Documentation)

Executive Summary

Problem Statement and Objectives

Methodology (Data Collection, Cleaning, SQL Queries for Forecasting & Optimization)

Tools and Technologies Used (e.g., SQL Server, PostgreSQL)

Results and Insights

Recommendations for Business Impact

Future Scope

2. SQL Scripts and Queries

Scripts for:

Data Cleaning and Preprocessing

Demand Forecasting (using SQL Aggregations, Moving Averages, Trend Analysis)

Route Optimization (using Joins, Distance Matrix calculations)

Well-commented and modular for reusability

3. Interactive Dashboard (Optional but Valuable)

Dashboard built using tools like Power BI or Tableau connected to SQL database

Key Metrics: Forecast Accuracy, Optimized Route Costs, Demand Trends

Filters: Region, Time Period, Product Category

4. Visual Flowchart or Architecture Diagram

Overview of the system architecture: data flow from raw data to final insights

Visual process of forecasting and route optimization

5. Presentation Slide Deck

10–12 slides summarizing objectives, methods, key findings, and business impact

Designed for stakeholders and business decision-makers

13. Source Code

```
!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />
  <title>Supply Chain Optimization Project</title>
  <style>
   /* Modern Responsive Styling */
   body {
     margin: 0;
     font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
     background: #f7f9fc;
     color: #333;
     line-height: 1.6;
     padding: 1em;
     max-width: 600px;
     margin-left: auto;
     margin-right: auto;
   }
   h1, h2 {
     color: #2a66f7;
     text-align: center;
   }
   section {
     background: white;
     border-radius: 12px;
     box-shadow: 0 6px 18px rgba(42, 102, 247, 0.12);
     padding: 1em 1.2em 2em;
     margin-bottom: 2em;
   }
   canvas {
     display: block;
     margin: 1em auto 0;
     max-width: 100%;
     height: auto !important;
```

```
    }
    p {
      text-align: center;
      font-size: 1em;
      margin-top: 0.1em;
      color: #444;
    }
    /* Responsive Text */
    @media(max-width: 400px) {
      body {
        padding: 0.5em;
      }
      h1 {
        font-size: 1.8rem;
      }
      h2 {
        font-size: 1.3rem;
      }
    }
  </style>
</head>
<body>
  <h1>Optimizing Supply Chain Logistics using SQL</h1>
  <p>Project with Demand Forecasting & Route Optimization<br/> Four topics with interactive graphs</p>

  <section id="demand-forecasting">
    <h2>1. Demand Forecasting</h2>
    <p>Using historical sales data, predict future demand.</p>
    <canvas id="demandChart" width="600" height="300"></canvas>
  </section>

  <section id="inventory-optimization">
    <h2>2. Inventory Optimization</h2>
    <p>Adjust inventory levels based on forecasted demand.</p>
    <canvas id="inventoryChart" width="600" height="300"></canvas>
  </section>
```

```
<section id="route-optimization">
  <h2>3. Route Optimization</h2>
  <p>Optimize delivery routes to reduce costs and time.</p>
  <canvas id="routeChart" width="600" height="300"></canvas>
</section>

<section id="network-visualization">
  <h2>4. Supply Chain Network Visualization</h2>
  <p>Visualize supply chain relationships and identify bottlenecks.</p>
  <canvas id="networkChart" width="600" height="300"></canvas>
</section>

<script>
  // Minimal chart library substitute with Canvas API for better portability
  // We simulate SQL data with static JS arrays

  // Topic 1: Demand Forecasting data (Line Chart)
  const demandData = [
    { month: 'Jan', sales: 120 },
    { month: 'Feb', sales: 135 },
    { month: 'Mar', sales: 150 },
    { month: 'Apr', sales: 160 },
    { month: 'May', sales: 180 },
    { month: 'Jun', sales: 175 },
    { month: 'Jul', sales: 190 },
    { month: 'Aug', sales: 220 },
    { month: 'Sep', sales: 210 },
    { month: 'Oct', sales: 230 },
    { month: 'Nov', sales: 250 },
    { month: 'Dec', sales: 270 },
  ];

  // Topic 2: Inventory Optimization data (Bar Chart)
  const inventoryData = [
    { product: 'Product A', stock: 180 },
    { product: 'Product B', stock: 130 },
```

```
  { product: 'Product C', stock: 90 },
  { product: 'Product D', stock: 150 },
  { product: 'Product E', stock: 110 },
];

// Topic 3: Route Optimization data (Multi-route distances)
// Simulated SQL output data for three routes with distance km and delivery time (hrs)
const routeData = [
  { route: 'Route 1', distance: 50, time: 1.2 },
  { route: 'Route 2', distance: 65, time: 1.5 },
  { route: 'Route 3', distance: 40, time: 1.1 },
];

// Topic 4: Supply Chain Network Visualization - Using simplified nodes & edges data
const networkNodes = [
  { id: 'Supplier', x: 100, y: 150 },
  { id: 'Warehouse', x: 300, y: 150 },
  { id: 'Retailer 1', x: 500, y: 70 },
  { id: 'Retailer 2', x: 500, y: 230 },
];
const networkEdges = [
  { from: 0, to: 1 },
  { from: 1, to: 2 },
  { from: 1, to: 3 },
];

// Utility to draw axes
function drawAxes(ctx, startX, startY, width, height, maxY, labels) {
  ctx.strokeStyle = '#666';
  ctx.lineWidth = 1;
  // Y axis
  ctx.beginPath();
  ctx.moveTo(startX, startY);
  ctx.lineTo(startX, startY - height);
  ctx.stroke();
  // X axis
  ctx.beginPath();
```

```
  ctx.moveTo(startX, startY);
  ctx.lineTo(startX + width, startY);
  ctx.stroke();

  // Y axis ticks and labels
  ctx.fillStyle = '#666';
  ctx.textAlign = 'right';
  ctx.textBaseline = 'middle';
  ctx.font = '11px Arial';
  const steps = 5;
  for(let i=0; i<=steps; i++) {
    let y = startY - (height / steps) * i;
    let val = Math.round(maxY / steps * i);
    ctx.beginPath();
    ctx.moveTo(startX - 4, y);
    ctx.lineTo(startX, y);
    ctx.stroke();
    ctx.fillText(val.toString(), startX - 6, y);
  }

  // X axis labels
  ctx.textAlign = 'center';
  ctx.textBaseline = 'top';
  ctx.font = '11px Arial';
  const labelCount = labels.length;
  for(let i=0; i<labelCount; i++) {
    let x = startX + (width / (labelCount - 1)) * i;
    ctx.fillText(labels[i], x, startY + 4);
  }
}

// Topic 1: Demand Forecasting (Line Chart)
(function drawDemandChart() {
  const canvas = document.getElementById('demandChart');
  const ctx = canvas.getContext('2d');
  const paddingLeft = 40, paddingBottom = 30;
  const w = canvas.width - paddingLeft - 20;
```

```
    const h = canvas.height - paddingBottom - 40;
    const maxSales = Math.max(...demandData.map(d => d.sales)) * 1.1;

    // Clear
    ctx.clearRect(0, 0, canvas.width, canvas.height);

    // Title
    ctx.fillStyle = '#2a66f7';
    ctx.font = 'bold 16px Arial';
    ctx.textAlign = 'center';
    ctx.fillText('Monthly Sales Demand Forecast', canvas.width / 2, 20);

    // Axes
    drawAxes(ctx, paddingLeft, canvas.height - paddingBottom, w, h, maxSales,
demandData.map(d=>d.month));

    // Draw line
    ctx.strokeStyle = '#2a66f7';
    ctx.lineWidth = 3;
    ctx.beginPath();
    demandData.forEach((d, i) => {
      let x = paddingLeft + (w / (demandData.length-1))*i;
      let y = canvas.height - paddingBottom - (d.sales / maxSales) * h;
      if(i === 0) ctx.moveTo(x,y);
      else ctx.lineTo(x,y);
    });
    ctx.stroke();

    // Draw data points
    ctx.fillStyle = '#174dd6';
    demandData.forEach((d, i) => {
      let x = paddingLeft + (w / (demandData.length-1))*i;
      let y = canvas.height - paddingBottom - (d.sales / maxSales) * h;
      ctx.beginPath();
      ctx.arc(x,y,5,0,Math.PI*2);
      ctx.fill();
    });
```

```
})();

// Topic 2: Inventory Optimization (Bar Chart)
(function drawInventoryChart() {
  const canvas = document.getElementById('inventoryChart');
  const ctx = canvas.getContext('2d');
  const paddingLeft = 45, paddingBottom = 40;
  const w = canvas.width - paddingLeft - 20;
  const h = canvas.height - paddingBottom - 40;
  const maxStock = Math.max(...inventoryData.map(d => d.stock)) * 1.2;

  // Clear
  ctx.clearRect(0, 0, canvas.width, canvas.height);

  // Title
  ctx.fillStyle = '#2a66f7';
  ctx.font = 'bold 16px Arial';
  ctx.textAlign = 'center';
  ctx.fillText('Inventory Levels (Units)', canvas.width / 2, 20);

  // Axes
  drawAxes(ctx, paddingLeft, canvas.height - paddingBottom, w, h, maxStock,
inventoryData.map(d => d.product));

  // Bars
  const barWidth = w / inventoryData.length * 0.6;
  inventoryData.forEach((d, i) => {
    let x = paddingLeft + (w / inventoryData.length) * i + ((w / inventoryData.length) -
barWidth) / 2;
    let barHeight = (d.stock / maxStock) * h;
    let y = canvas.height - paddingBottom - barHeight;

    // Bar coloring gradients from teal to blue
    let gradient = ctx.createLinearGradient(x, y, x, y + barHeight);
    gradient.addColorStop(0, '#009688');
    gradient.addColorStop(1, '#2a66f7');
```

```
    ctx.fillStyle = gradient;
    ctx.fillRect(x, y, barWidth, barHeight);

    // Bar labels
    ctx.fillStyle = '#333';
    ctx.font = '12px Arial';
    ctx.textAlign = 'center';
    ctx.textBaseline = 'bottom';
    ctx.fillText(d.stock, x + barWidth / 2, y - 4);
  });
})();

// Topic 3: Route Optimization (Horizontal Bar Chart with time & distance bars)
(function drawRouteChart() {
  const canvas = document.getElementById('routeChart');
  const ctx = canvas.getContext('2d');
  const paddingLeft = 130, paddingBottom = 40;
  const w = canvas.width - paddingLeft - 20;
  const h = canvas.height - paddingBottom - 60;
  const maxDistance = Math.max(...routeData.map(d=>d.distance)) * 1.2;
  const maxTime = Math.max(...routeData.map(d=>d.time)) * 1.2;

  ctx.clearRect(0, 0, canvas.width, canvas.height);

  // Title
  ctx.fillStyle = '#2a66f7';
  ctx.font = 'bold 16px Arial';
  ctx.textAlign = 'center';
  ctx.fillText('Route Optimization - Distance & Delivery Time', canvas.width / 2, 20);

  // Draw route labels vertically
  ctx.fillStyle = '#333';
  ctx.font = '14px Arial';
  ctx.textAlign = 'right';
  ctx.textBaseline = 'middle';
  const rowHeight = h / routeData.length;
  routeData.forEach((d,i) => {
```

```
    ctx.fillText(d.route, paddingLeft - 10, 60 + i * rowHeight + rowHeight/2);
  });

  // Draw bars for distance (blue) and time (orange)
  routeData.forEach((d,i) => {
    const y = 60 + i * rowHeight + 5;
    // Distance bar
    const distBarWidth = (d.distance / maxDistance) * (w*0.45);
    ctx.fillStyle = '#2a66f7';
    ctx.fillRect(paddingLeft, y, distBarWidth, rowHeight*0.35);
    // Distance label
    ctx.fillStyle = '#000';
    ctx.font = '11px Arial';
    ctx.textAlign = 'left';
    ctx.fillText(d.distance + ' km', paddingLeft + distBarWidth + 5, y + rowHeight*0.35/2);

    // Time bar (offset x axis)
    const timeBarWidth = (d.time / maxTime) * (w*0.45);
    ctx.fillStyle = '#ff9800';
    ctx.fillRect(paddingLeft + w*0.5, y, timeBarWidth, rowHeight*0.35);
    // Time label
    ctx.fillStyle = '#000';
    ctx.fillText(d.time.toFixed(1) + ' hrs', paddingLeft + w*0.5 + timeBarWidth + 5, y +
rowHeight*0.35/2);
  });

  // Legends
  ctx.fillStyle = '#2a66f7';
  ctx.fillRect(paddingLeft, 40, 15, 15);
  ctx.fillStyle = '#333';
  ctx.font = '12px Arial';
  ctx.textAlign = 'left';
  ctx.fillText('Distance (km)', paddingLeft + 20, 52);

  ctx.fillStyle = '#ff9800';
  ctx.fillRect(paddingLeft + w*0.5, 40, 15, 15);
  ctx.fillStyle = '#333';
```

```
  ctx.fillText('Delivery Time (hrs)', paddingLeft + w*0.5 + 20, 52);
})();

// Topic 4: Network Visualization (Nodes and Edges)
(function drawNetworkChart() {
  const canvas = document.getElementById('networkChart');
  const ctx = canvas.getContext('2d');
  ctx.clearRect(0, 0, canvas.width, canvas.height);

  // Title
  ctx.fillStyle = '#2a66f7';
  ctx.font = 'bold 16px Arial';
  ctx.textAlign = 'center';
  ctx.fillText('Supply Chain Network Visualization', canvas.width / 2, 25);

  // Draw edges
  ctx.strokeStyle = '#888';
  ctx.lineWidth = 3;
  for(let edge of networkEdges) {
    const fromNode = networkNodes[edge.from];
    const toNode = networkNodes[edge.to];
    ctx.beginPath();
    ctx.moveTo(fromNode.x, fromNode.y);
    ctx.lineTo(toNode.x, toNode.y);
    ctx.stroke();

    // Arrowhead
    const angle = Math.atan2(toNode.y - fromNode.y, toNode.x - fromNode.x);
    const arrowLength = 10;
    const arrowAngle = Math.PI / 7;
    let arrowX = toNode.x - arrowLength * Math.cos(angle - arrowAngle);
    let arrowY = toNode.y - arrowLength * Math.sin(angle - arrowAngle);
    ctx.beginPath();
    ctx.moveTo(toNode.x, toNode.y);
    ctx.lineTo(arrowX, arrowY);
    ctx.stroke();
    arrowX = toNode.x - arrowLength * Math.cos(angle + arrowAngle);
```

```
      arrowY = toNode.y - arrowLength * Math.sin(angle + arrowAngle);
      ctx.beginPath();
      ctx.moveTo(toNode.x, toNode.y);
      ctx.lineTo(arrowX, arrowY);
      ctx.stroke();
    }

    // Draw nodes
    networkNodes.forEach(node => {
      ctx.fillStyle = '#2a66f7';
      ctx.beginPath();
      ctx.arc(node.x, node.y, 20, 0, Math.PI*2);
      ctx.fill();
      ctx.fillStyle = 'white';
      ctx.font = 'bold 13px Arial';
      ctx.textAlign = 'center';
      ctx.textBaseline = 'middle';
      ctx.fillText(node.id, node.x, node.y);
    });
  })();

  </script>
</body>
</html>
</content>
</create_file>
```

14. Future Scope

Here’s a future scope for the project "Optimizing Supply Chain Logistics using Demand Forecasting and Optimization in SQL":

Future Scope

1. Real-Time Data Integration

Incorporate real-time sales, weather, and traffic data for more dynamic and responsive demand forecasting and route optimization.

2. Machine Learning Integration

Enhance SQL-based forecasting with ML models (e.g., ARIMA, LSTM) for better accuracy, especially in volatile demand environments.

3. Geospatial Analysis and Visualization

Integrate GIS tools to visualize supply routes and demand hotspots, helping decision-makers identify regional inefficiencies.

4. Scalability to Big Data Platforms

Transition from traditional SQL databases to distributed systems like Apache Hive or Google BigQuery for handling massive datasets efficiently.

5. Automated Optimization Engine

Develop an automated engine to continuously refine routes and inventory levels based on updated forecasts and cost constraints.

6. Integration with ERP and Inventory Systems

Connect the solution with existing ERP tools for automated procurement, inventory reordering, and vendor coordination.

7. Sustainability Optimization

Incorporate CO_2 emission tracking and design logistics routes that minimize environmental impact while meeting demand.

8. User Dashboards and Alerts

Create interactive dashboards and automated alerts for anomalies in demand or logistics, aiding proactive decision-making.

9. Cost Simulation and Scenario Planning

Add features to simulate "what-if" scenarios, helping businesses prepare for supply shocks or demand spikes.

10. Global Expansion and Multi-Warehouse Optimization

Extend the model to support international logistics and optimize distribution across multiple warehouses.

15. Team Members and Roles

Name	Responsibility
Perarasu. V	Data analysist
Pachaiyappan. D	Front end developer
Pandiyarajan. S	Backend developer