**Project Title**

**Logistics Management System for Ship and Cargo Operations**

**Objective**

To create a robust database system for managing ship and cargo logistics, including tracking shipments, managing cargo storage, scheduling routes, monitoring maintenance, and handling staff. The system ensures streamlined operations, real-time analytics, and effective resource utilization.

**Database Features**

**Core SQL Concepts:**

* CRUD operations
* Joins (INNER, LEFT, RIGHT, FULL OUTER)
* Subqueries
* Union and Union All
* Aggregate Functions (SUM, AVG, COUNT, etc.)

**Advanced SQL Concepts:**

* Common Table Expressions (CTE)
* PIVOT/UNPIVOT
* Window Functions (e.g., ROW\_NUMBER, RANK, NTILE)
* User-Defined Functions (UDF)
* Stored Procedures (SP) and Wrapper SP
* Triggers
* Conditional Statements (IF, CASE)

**Real-Time Functionalities:**

* Cargo tracking and inventory
* Ship scheduling and route management
* Maintenance tracking for ships
* Employee task assignments
* Real-time shipment updates
* Analytics for operational efficiency

**Database Schema**

**1. Tables**

**Ships Table**  
Tracks details of the ships in the logistics system.

CREATE TABLE Ships (

ShipID INT PRIMARY KEY,

ShipName NVARCHAR(100) NOT NULL,

Capacity INT,

Status NVARCHAR(20) -- Operational, Under Maintenance, Docked

);

**Cargo Table**  
Stores information about cargo and its tracking.

CREATE TABLE Cargo (

CargoID INT PRIMARY KEY,

CargoName NVARCHAR(100),

Weight DECIMAL(10, 2),

DestinationPort NVARCHAR(100),

ShipID INT,

Status NVARCHAR(20), -- Pending, Shipped, Delivered

FOREIGN KEY (ShipID) REFERENCES Ships(ShipID)

);

**Ports Table**  
Contains details about the ports involved.

sql

CopyEdit

CREATE TABLE Ports (

PortID INT PRIMARY KEY,

PortName NVARCHAR(100),

Location NVARCHAR(100)

);

**ShipRoutes Table**  
Manages ship routes and schedules.

CREATE TABLE ShipRoutes (

RouteID INT PRIMARY KEY,

ShipID INT,

DeparturePortID INT,

ArrivalPortID INT,

DepartureDate DATETIME,

ArrivalDate DATETIME,

Status NVARCHAR(20), -- Scheduled, En Route, Completed

FOREIGN KEY (ShipID) REFERENCES Ships(ShipID),

FOREIGN KEY (DeparturePortID) REFERENCES Ports(PortID),

FOREIGN KEY (ArrivalPortID) REFERENCES Ports(PortID)

);

**Employees Table**  
Tracks ship crew and logistics staff.

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FullName NVARCHAR(100),

Role NVARCHAR(50), -- Captain, Crew Member, Logistics Manager

ShipID INT,

Salary DECIMAL(10, 2),

FOREIGN KEY (ShipID) REFERENCES Ships(ShipID)

);

**MaintenanceLog Table**  
Tracks maintenance records for ships.

sql

CopyEdit

CREATE TABLE MaintenanceLog (

MaintenanceID INT PRIMARY KEY,

ShipID INT,

MaintenanceDate DATETIME,

Description NVARCHAR(MAX),

Status NVARCHAR(20), -- Completed, Pending

FOREIGN KEY (ShipID) REFERENCES Ships(ShipID)

);

**Advanced SQL Functionalities**

**1. Joins**

Query to fetch cargo details along with ship and destination port:

SELECT

c.CargoName,

c.Weight,

s.ShipName,

p.PortName AS DestinationPort,

c.Status

FROM Cargo c

INNER JOIN Ships s ON c.ShipID = s.ShipID

INNER JOIN Ports p ON c.DestinationPort = p.PortName;

**2. CTE and PIVOT**

**CTE** for tracking cargo status:

WITH CargoStatus AS (

SELECT

CargoID,

CargoName,

Weight,

Status,

CASE

WHEN Status = 'Pending' THEN 'Not Shipped'

WHEN Status = 'Shipped' THEN 'In Transit'

WHEN Status = 'Delivered' THEN 'Delivered'

END AS CurrentStatus

FROM Cargo

)

SELECT \* FROM CargoStatus;

**PIVOT** for ship route status:

SELECT \*

FROM (

SELECT

ShipID,

Status,

COUNT(RouteID) AS RouteCount

FROM ShipRoutes

GROUP BY ShipID, Status

) AS SourceTable

PIVOT (

SUM(RouteCount)

FOR Status IN ([Scheduled], [En Route], [Completed])

) AS PivotTable;

**3. User-Defined Function**

UDF to calculate route duration in hours:

CREATE FUNCTION CalculateRouteDuration(@DepartureDate DATETIME, @ArrivalDate DATETIME)

RETURNS INT

AS

BEGIN

RETURN DATEDIFF(HOUR, @DepartureDate, @ArrivalDate);

END;

**4. Stored Procedure**

Wrapper SP to add a new ship route:

CREATE PROCEDURE AddShipRoute

@ShipID INT,

@DeparturePortID INT,

@ArrivalPortID INT,

@DepartureDate DATETIME,

@ArrivalDate DATETIME

AS

BEGIN

INSERT INTO ShipRoutes (ShipID, DeparturePortID, ArrivalPortID, DepartureDate, ArrivalDate, Status)

VALUES (@ShipID, @DeparturePortID, @ArrivalPortID, @DepartureDate, @ArrivalDate, 'Scheduled');

END;

**5. Trigger**

Trigger to update ship status after route completion:

CREATE TRIGGER trg\_UpdateShipStatusAfterRoute

ON ShipRoutes

AFTER UPDATE

AS

BEGIN

UPDATE Ships

SET Status = 'Docked'

WHERE ShipID IN (

SELECT ShipID

FROM inserted

WHERE Status = 'Completed'

);

END;

**Flowchart**

1. **Cargo Management**: Tracks cargo → Updates Cargo table.
2. **Ship Scheduling**: Assign routes → Updates ShipRoutes table.
3. **Port Management**: Stores port info → Updates Ports table.
4. **Maintenance Tracking**: Logs maintenance → Updates MaintenanceLog table.
5. **Employee Allocation**: Assign staff → Updates Employees table.

**Deliverables**

1. **SQL Scripts**: Full schema, sample data, and queries.
2. **Base Paper**: Comprehensive explanation of the system, its objectives, methodologies, and SQL features.
3. **Flowchart**: Visual representation of the workflow.
4. **Sample Data**: Pre-populated tables with realistic values for testing.