RailUK Database Development Case Study

You have been appointed as a Database Developer for *RailUK Logistics Ltd*, a national freight transport and logistics company based in the United Kingdom. RailUK manages freight trains, warehouses, and delivery hubs across the country, working with both large corporate clients (e.g., supermarkets, manufacturers) and SMEs. The company has expanded rapidly into multi-modal logistics (rail, road, last-mile delivery), and now faces fragmented records across regions, inefficiencies in cargo tracking, lack of integrated client billing and scheduling, and security lapses in staff access to sensitive data.

RailUK's IT Department has been tasked with developing a new database system using Oracle Application Express (APEX) **(**preferred) or Oracle SQL Developer**.** The database will support enterprise-level operations including shipment scheduling, cargo tracking, warehouse management, and customer billing.

Stage One: Database Design

You are required to:

1. **Design the database schema** with a clear structure including all entities, attributes, relationships, data types, and constraints while ensuring referential integrity.
2. **Normalise the data** (to 3NF) to avoid redundancy and anomalies.
3. **Implement the tables** in Oracle (APEX or SQL Developer), ensuring all constraints such as primary keys, foreign keys, and unique constraints are defined.

Entities for this case study include (minimum):

- **Clients** (corporate or SME; contact details, industry sector)
- **Shipments** (unique shipment ID, cargo details, weight, delivery deadlines)
- **TransportModes** (rail, lorry, last-mile van)
- **Schedules** (routes, departure/arrival times, associated shipment IDs)
- **Warehouses** (location, storage capacity, manager details)
- **Staff** (roles: drivers, logistics officers, admin)
- **Invoices** (linked to shipments and clients)
- **Payments** (status, amounts, method)
- **Suppliers** (e.g., fuel, maintenance contractors)

You are encouraged to expand with **sub-entities** such as *CargoType*, *StaffRoles*, *MaintenanceLogs* if justified. Please complete the weekly labs to get familiar with the necessary SQL commands.

Stage Two: Data Population & Queries

Once the schema is built:

1. **Populate** each table with at least 20 realistic records suitable for this kind of company.
2. Demonstrate **INSERT, UPDATE, DELETE** operations with appropriate examples.

3. Run **basic queries** (SELECT, filters, joins) and **advanced queries** (aggregates, GROUP BY, HAVING, complex filters, nested subqueries).
4. Demonstrate at least three **real-world reporting** queries, e.g.:
    - List all shipments delayed beyond deadline, grouped by region.
    - Calculate total monthly revenue per client sector.

Stage Three: Optimisation, Security & Research

You will optimise and secure the database and do some research on optimisation techniques.

Optimisation:

- Create **indexes** on frequently queried attributes (e.g., shipment ID, client ID).
- Use **partitioning** on large tables (e.g., Shipments by year, Invoices by quarter).
- Explore **query optimisation techniques** (such as cost-based optimisation and indexing strategies) to demonstrate performance impact and implement improvements.
- Implement **stored procedures/functions** for recurring operations (e.g., invoice generation, shipment tracking queries).

Security: (*For this part, you will need to write an **SQL script** using your favourite SQL environment/editor as it is not possible to perform these actions in APEX)*

- Create **users** with role-based access (Admin, Logistics Officer, Warehouse Staff).
- Apply **GRANT/REVOKE** to enforce least-privilege access.
- Demonstrate **SQL injection awareness** with mitigations.

Research Component — Literature Review (600 words)

Drawing upon **at least 6** peer-reviewed academic sources plus reputable industry white papers (e.g., Oracle, ACM SIGMOD, VLDB), you are required to produce a literature review that:

1. Surveys academic and industry research on optimisation in RDBMS (query processing, indexing, partitioning, clustering, cost-based optimisation, parallel execution).
2. Critically evaluates **recent advances** (e.g., adaptive query optimisation in Oracle 19c/23c, in-memory column stores, autonomous indexing).
3. Discusses **open challenges** in the field (e.g., balancing OLTP vs OLAP workloads, hybrid relational/NoSQL optimisation, cloud cost/performance trade-offs).
4. Connects insights back to **your implementation in RailUK Logistics** (**important!**)

Assessment Notes:

- You should demonstrate the **impact of database optimisation techniques** by highlighting performance metrics (e.g., costs, cardinality).
- Additional database optimisation and security measures will be rewarded with extra credit in the assessment.
- **Remember**, you will have to write an SQL script for the security part because it cannot be implemented in APEX.