Earlier known as

B. V. B. College of Engineering & Technology

Science and Engineering School of Computer

**DBA COURSE PROJECT**

**ON**

**FLEET MANAGEMENT SYSTEM**

**List of Team Members:**

|  |  |  |
| --- | --- | --- |
| **NAME** | **ROLL NO** | **USN** |
| Shakunthala Koti | 134 | 01FE19BCS039 |
| Rachita H | 138 | 01FE19BCS043 |
| Shreeya Goggi | 140 | 01FE19BCS045 |
| Rashmi Kiragi | 150 | 01FE19BCS057 |

# **CONTENTS**

1. Introduction

1.1 Fleet Management 3

1.2 Database Management System 3

1.3 Problem Statement 4

1.4 Objective 5

1.5 Motivation 5

2. Requirement Collection and Analysis

2.1 Introduction 6

2.2 Data Requirement 7

2.3 Functional Requirement 8

2.4 Non-Functional Requirement 8

3. Database Design

3.1 Introduction 9

3.2 Conceptual Design 9

4. Implementation and Results

4.1 Introduction 17

4.2 Database Tables 17

4.3 Results 24

References 25

# **1. Introduction**

## 1.1 Fleet Management

Fleet management has completely transformed over the last ten years as businesses start to see the impact that an efficient fleet can have on business operations. It now covers much more than procuring and maintaining vehicles, and fleet management teams are now responsible for a wide range of activities including driver management, compliance management, operational efficiency and environmental impact.

Any organization that needs commercial vehicles to function engages in some form of fleet operations and management. The purpose of fleet management is to oversee all fleet operations in order to increase productivity and help a business run as smoothly as possible.

Fleet managers are in charge of fleet operations and are tasked with responsibilities that include the delivery of the consignment to the clients, guiding their employees, managing all their branches and warehouses and the implementation of any programs that increase company productivity.

## 1.2 Database Management System

A database management system (DBMS) is system software for creating and managing databases. A DBMS makes it possible for end users to create, protect, read, update and delete data in a database. The most prevalent type of data management platform, the DBMS essentially serves as an interface between databases and end users or application programs, ensuring that [data is consistently organized and remains easily accessible](https://searchdatamanagement.techtarget.com/tip/Why-organizations-need-a-solid-data-governance-strategy).

The DBMS manages the data; the database engine allows data to be accessed, locked and modified; and the database [schema](https://searchsqlserver.techtarget.com/definition/schema) defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform data administration procedures. Typical database administration tasks the DBMS supports include change management, performance monitoring and tuning, security, and backup and recovery. The DBMS provides a centralized view of data that can be accessed by multiple users, from multiple locations, in a controlled manner.

The DBMS can offer both logical and physical data independence. This means it can protect users and applications from needing to know where data is stored or being concerned about changes to the physical structure of data. As long as programs use the application programming interface (API) for the database that the DBMS provides, developers won't have to modify programs just because changes have been made to the database. In a relational database management system ([RDBMS](https://searchdatamanagement.techtarget.com/definition/RDBMS-relational-database-management-system)), the most widely used type of DBMS, this API is SQL, a standard programming language for defining, protecting and accessing data.

## 1.3 Problem Statement

This project aims to design and implement the database for Fleet Management System to maintain the following activities.

* The fleet management has multiple branches around the country, each branch has unique id, name, location and manager name.
* The fleet management also has multiple warehouses, each warehouse is uniquely identified by id, branch id, name and address.
* Fleet maintains many types of vehicles, each vehicle has branch id, registration number, type of vehicle, status, permit and capacity of vehicle.
* Fleet has multiple clients, each client must have unique id, name, location and respective consignment id.
* The management has employees, each employee is uniquely identified by their id, name, dob, salary, doj, experience, branch id, address and contact number.
* It also has conveyance, which has its own id, source, destination, distance in km and vehicle registration number.
* It should contain consignments, each consignment has unique id, material, weight, delivery date, consigner, consignee, amount due, amount paid, flammability and warehouse id.
* A branch can have multiple of warehouses, employees and vehicles.
* One warehouse can have multiple consignment and many warehouses can belong to a single branch.
* One vehicle can belong to multiple employees and conveyance. And many vehicles can be associated with a single branch.
* A client can book multiple consignments.
* Many employees can be associated with a single vehicle or a single branch.
* Multiple conveyance can be undertaken by a single vehicle.
* Multiple consignments can be associated to a single client or a single vehicle.

## 1.4 Objectives

To operate efficiently and to remain competitive, it is essential for a fleet company to manage its data appropriately. Databases enable quick access to critical information, provide secure storage of company’s data, and offer analysis/reporting tools for real-time decision making. Databases have the ability to store information pertaining to company’s infrastructure, its clients, employees, consignments and so on. This database will include tables that store the most common types of information pertinent to fleet management:

\*Branch, \*Warehouse, \*Vehicle, \*Client, \*Employee, \*Conveyance and \*Consignment. In addition to the tables, you will also setup the relationships between tables, and design input forms and queries.

## 1.5 Motivation

Since SQL is used to view, manage and access that data, it is well suited for Data Manipulation. It gives the users an easier time to test and manipulate the data. SQL has many advantages which makes it popular and highly demanded. It is reliable and efficient language used for communicating with the database. Some of the advantages of SQL are-

* [Simple Troubleshooting](https://www.edureka.co/blog/top-10-reasons-to-learn-sql/#troubleshoot)
* [Combine Data from Multiple Sources](https://www.edureka.co/blog/top-10-reasons-to-learn-sql/#combinedata)
* [Data Manipulation](https://www.edureka.co/blog/top-10-reasons-to-learn-sql/#datamanipulation)
* [Quick Access to Data](https://www.edureka.co/blog/top-10-reasons-to-learn-sql/#quickaccess)
* [Client-Server Access](https://www.edureka.co/blog/top-10-reasons-to-learn-sql/#clientserver)
* [Manage Humongous Amounts of Data](https://www.edureka.co/blog/top-10-reasons-to-learn-sql/#managedata)
* [Perform Data Mining](https://www.edureka.co/blog/top-10-reasons-to-learn-sql/#datamining)
* [Standardization](https://www.edureka.co/blog/top-10-reasons-to-learn-sql/#standardization)

# **2. Requirement Collection and Analysis**

## 2.1 Introduction

The most critical aspect of specification is the gathering and compilation of system and

user requirements. This process is normally done in conjunction with managers and

users. The major goal in requirements gathering process is to:

· Collect the data used by the organization

· Identify relationships/conditions to be applied on the data

· Identify future data needs

· Determine how the data is used and generated

· Identify the functions that are performed on the data

The starting place for data collection is gathering existing forms and reviewing policies

and systems. Then, ask users what the data means, and determine their daily

processes

Following subsections discuss the data requirements and functional & non functional

requirements identified based on the following activities collected from the fleet management system which is restricted to road transport.

* The system has branches which in turn can have any number of warehouses that can collect any number of consignments.
* Each branch has a manager who is an employee in the respective branch.
* Branches, warehouses have names, unique id’s and locations.
* Each employee has a unique identification number, name, date of birth, date of joining, contact number, salary, experience, branch id and address. He / She can work in only one branch but a branch can have any number of employees.
* Only one employee can drive a vehicle for a particular delivery. A vehicle can be ridden by an employee any number of times.
* Goods / Consignments are delivered via different types of carriers (closed body truck, light open body, heavy open body, reefer truck or tanker).
* Every vehicle is registered under a branch with unique registration number. It can carry only one type of material for a particular delivery. Every type of vehicle has insurance, vehicle permit (state or all India), carrier capacity (ton).
* Each client has a unique id, consignment id and location. It can order similar / different material as many times as it wants to.
* Each consignment has a unique id, material name, weight (ton), flammability, delivery date, names of consigner & consignee, paid and due amount (Lakhs).

## 2.2 Data Requirement

Data requirement describes the data to be stored in the database pertaining to activities

of the fleet management system requirement as described in section 2.1. Details of the data stored in the database is shown in the table 2.1 and Table 2.2.

2.1 Data to be stored in database

|  |  |  |
| --- | --- | --- |
| S. No. | Group | Data related to each group |
| 1 |  | Location ID (unique), area, city, state and pin |
| 2 | Branch | Branch ID, name, location ID, manager ID |
| 3 | Employee | Employee ID, name, date of birth, date of joining, salary, experience, contact number, branch ID and address ID |
| 4 | Warehouse | Warehouse ID, name, location ID and branch ID |
| 5 | Consignment | Consignment ID, material name, names of consigner and consignee, delivery date, weight, flammability, paid and due amount and warehouse ID |
| 6 | Client | Client ID, name, location ID and consignment ID |
| 7 | Vehicle | Registration number (unique), insurance, type, status, carrying capacity, vehicle permit, consignment ID and branch ID |
| 8 | Conveyance | Source, destination, distance (km), employee ID and vehicle registration number |

2.2 Conditions on Data

|  |  |
| --- | --- |
| S. No. | Conditions |
| 1 | * A branch can have any number of employees. * A branch can have any number of warehouses. |
| 2 | * An employee can work in only one branch. * An employee can drive any number of vehicles. |
| 3 | * A warehouse can collect any number of consignments. * Any number of warehouses can be present in a branch. |
| 4 | * A consignment can be delivered by only one vehicle. * Any number of consignments can be delivered to a particular warehouse. * Any number of consignments can be booked by the client. |
| 5 | A client can book similar / different material any number of times. |
| 6 | * Any number of vehicles can belong to a particular branch. * A vehicle can deliver one material per order. * A vehicle can be driven by any number of employees any number of times. * A vehicle can have any number of conveyances / journeys. |
| 7 | Any number of conveyances / journeys can be made by a vehicle. |

\*Locations’ group has data of all locations of client, employee, branch or warehouse, with unique ID’s.

## 2.3 Functional Requirement

Functional requirements are product features or functions that developers must

implement to enable users to accomplish their tasks. So, it’s important to make them

clear both for the development team and the stakeholders (clients). Table 2.3 shows

the different types users of fleet system and their respective responsibilities (tasks). Table 2.4 shows the different functions user can perform on the database.

Table 2.3: Categories of Users and their tasks

|  |  |  |
| --- | --- | --- |
| S. No. | User | Responsibility |
| 1 | Administration | Data administration functionalities |
| 2 | Branch manager | Viewing and modifying respective branch and warehouse data |

Table 2.4: Functions of each user

|  |  |  |
| --- | --- | --- |
| S. No. | Function | User |
| 1 | Inserting delivery records into database | Administration, branch manager |
| 2 | Delete records from database | Administration, branch manager |

## 2.4 Non-Functional Requirement

Non-functional Requirements (NFRs) define system attributes such as security,

reliability, performance, maintainability, scalability, and usability. They serve as

constraints or restrictions on the design of the system across the different backlogs. In

this project we are addressing the security and reliability in order to secure the data and accessibility and to rely on the statistics.

# **3. Database Design**

## 3.1 Introduction

The requirements gathering and specification provides you with a high-level understanding of the organization, its data, and the processes that you must model in the database. Database design involves constructing a suitable model of for the information. Since the design process is complicated, especially for large databases, database design is divided into three phases:

* + - Conceptual database design
    - Logical database design
* Physical database design.

In our project work we are addressing the conceptual database design using ER modeling and logical database design using the implementation data model called Relational model.

### Conceptual Database Design

Conceptual database design involves modelling the collected information at a high-level of abstraction without using a particular data model or DBMS. This model allows for easy communication between end-users and database developers and has a clear method to convert from high-level model to relational model. The most popular model for conceptual database design is the Entity Relationship model which describes data as attribute, entity and relationship.

Table 3.1 shows the list of attributes Table 3.2 shows the list of entity types and table

3.3 shows the list of relationship types identified for the requirement discussed in the section 2.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.  No. | Attribute Name | Attribute Type | Justification | Entity Type |
| 1 | Branch\_ ID | Simple | It is indivisible | Branch |
| 2 | Branch\_name | Simple | It is indivisible | Branch |
| 3 | Location\_Id | Simple | It is indivisible | Branch |
| 4 | Branch\_manager\_id | Simple | It is indivisible | Branch |
| 5 | Employee\_Id | Simple | It is indivisible | Employee |
| 6 | Employee\_Name | Simple | It is indivisible | Employee |
| 7 | Date of birth | Simple | It is indivisible | Employee |
| 8 | Salary | Simple | It is indivisible | Employee |
| 9 | Experience | Simple | It is indivisible | Employee |
| 10 | Contact number | Simple | It is indivisible | Employee |
| 11 | Date of joining | Simple | It is indivisible | Employee |
| 12 | Warehouse id | Simple | It is indivisible | warehouse |
| 13 | Warehouse name | Simple | It is indivisible | warehouse |
| 14 | Warehouse location id | Simple | It is indivisible | warehouse |
| 15 | Consignment id | Simple | It is indivisible | consignment |
| 16 | Material | Simple | It is indivisible | consignment |
| 17 | Consigner | Simple | It is indivisible | consignment |
| 18 | Consignee | Simple | It is indivisible | consignment |
| 19 | Delivery date | Simple | It is indivisible | consignment |
| 20 | Amount paid | Simple | It is indivisible | consignment |
| 21 | Amount due | Simple | It is indivisible | consignment |
| 22 | Weight | Simple | It is indivisible | consignment |
| 23 | Flammability | Simple | It is indivisible | consignment |
| 24 | Client id | Simple | It is indivisible | Client |
| 25 | Client name | Simple | It is indivisible | Client |
| 26 | Client location | Simple | It is indivisible | Client |
| 27 | Vehicle registration number | Simple | It is indivisible | Vehicle |
| 28 | Vehicle insurance | Simple | It is indivisible | Vehicle |
| 29 | Vehicle type | Simple | It is indivisible | Vehicle |
| 30 | Vehicle status | Simple | It is indivisible | Vehicle |
| 31 | Carrier capacity | Simple | It is indivisible | Vehicle |
| 32 | Vehicle permit | Simple | It is indivisible | Vehicle |
| 33 | Source | Simple | It is indivisible | Conveyance |
| 34 | Destination | Simple | It is indivisible | Conveyance |
| 35 | Distance | Simple | It is indivisible | Conveyance |
| 36 | Location id | Simple | It is indivisible | Location |
| 37 | pin | Simple | It is indivisible | Location |
| 38 | area | Simple | It is indivisible | Location |
| 39 | city | Simple | It is indivisible | Location |
| 40 | state | Simple | It is indivisible | Location |

Table 3.2: List of Entity Types

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Entity Type Name | Type of Entity Type | Justification |
| 1 | Branch | Strong | It has key attribute |
| 2 | Warehouse | Strong | It has key attribute |
| 3 | Vehicle | Strong | It has key attribute |
| 4 | Employee | Strong | It has key attribute |
| 5 | Consignment | Strong | It has key attribute |
| 6 | Client | Strong | It has key attribute |
| 7 | Conveyance | Strong | It does not have key attribute |
| 8 | Location | Strong | It has key attribute |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl.No. | Relationship Type  Name | Type of Relationship Type | Justification | Participating Entity Type with cardinality ratio | Participation |
| 1 | Has | Simple |  | Branch  Warehouse  Vehicle  Client  Employee  Location  1:n | total |
| partial |
| partial |
| 2 | Delivers | Simple |  | Consignment  Vehicle  1:1 | partial |
| 3 | Booked by | Simple |  | Consignment  Client  1:n | Total |
| 4 | Driven by | Simple |  | Vehicle  Employee  1:n | Partial |
| 5 | Undertakes | Simple |  | Conveyance  Vehicle  1:n | Partial |
| 6 | Collect | Simple |  | Warehouse  Consignment  1:n | Partial |

E-R Diagram, Schema Diagram and Normalization

Entity relationship diagram of the proposed system as described in the requirement analysis is shown in the figure 3.1and figure 3.2 shows the schema diagram obtained after converting ER diagram to relational model.

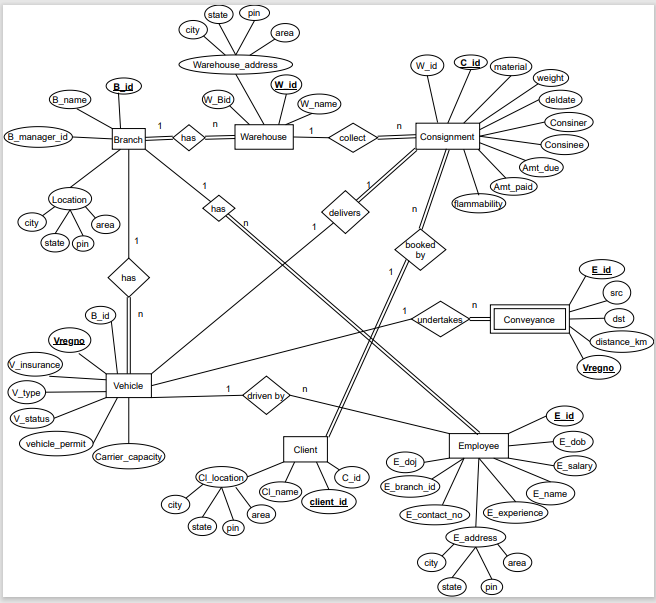


Fig.3.1: ER Diagram

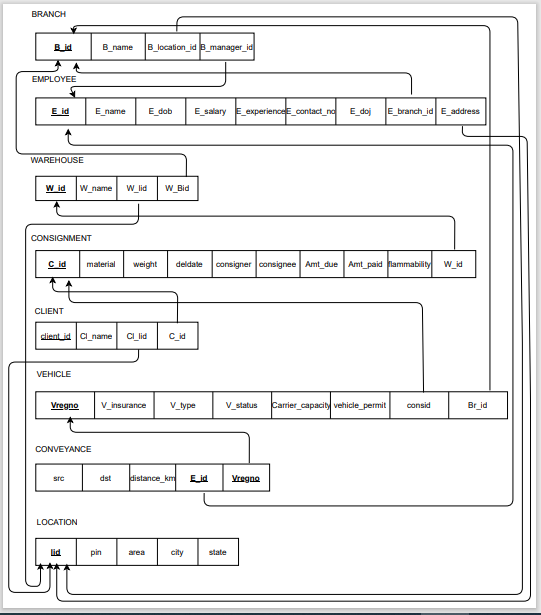


Fig 3.2: Schema diagram of the Fig.3.1

BRANCH

(B\_id,B\_name,B\_location\_id,B\_manager\_id)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF as it does not have partial dependency
      * The Relation is in 3NF as no non prime attribute determines prime attribute.i.e there is no transitive dependency.

EMPLOYEE

(e\_id , e\_name,e\_dob ,e\_salary ,e\_experience ,e\_contact\_no, e\_doj date,e\_branch\_id, address )

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF as it does not have partial dependency
      * The Relation is in 3NF as no non prime attribute determines prime attribute.i.e there is no transitive dependency.

WAREHOUSE

(W\_id , W\_name ,W\_lid ,W\_Bid)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF as it does not have partial dependency
      * The Relation is in 3NF as no non prime attribute determines prime attribute.i.e there is no transitive dependency.

CONSIGNMENT

(cid ,material ,consigner ,consignee ,deldate ,Amt\_paid decimal, Amt\_due decimal)

weight decimal, flammability ,c\_wid ,

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF as it does not have partial dependency
      * The Relation is in 3NF as no non prime attribute determines prime attribute.i.e there is no transitive dependency.

CLIENT

(client\_id ,cl\_name ,cl\_lid ,c\_id )

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF as it does not have partial dependency
      * The Relation is in 3NF as no non prime attribute determines prime attribute.i.e there is no transitive dependency.

VEHICLE

(vregno ,vinsurance , vtype ,vstatus ,carrier\_capacity ,vehicle\_permit,consid, branch )

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF as it does not have partial dependenc
      * The Relation is in 3NF as no non prime attribute determines prime attribute.i.e there is no transitive dependency

CONVEYANCE

(src ,dst ,distance\_km ,e\_id ,vregno)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF as it does not have partial dependenc
      * The Relation is in 3NF as no non prime attribute determines prime attribute.i.e there is no transitive dependency

LOCATIONS

(lid,pin,area,city,state)

* + - * The Relation is in 1NF as it has atomic valued attributes
      * The Relation is in 2NF as it does not have partial dependenc
      * The Relation is in 3NF as no non prime attribute determines prime attribute.i.e there is no transitive dependency

# **4. Implementation and Results**

## 4.1 Introduction

Implementation involves the construction of a database according to the specification of a logical schema. This will include the specification of an appropriate storage schema, security enforcement, external schema and so on. Implementation is influenced by the choice of available DBMSs, database tools and operating environment. There are additional tasks beyond simply creating a database schema and implementing the constraints such as data must be entered into the tables, issues relating to the users and user processes need to be addressed, and the management activities associated with wider aspects of corporate data management need to be supported. In practice, implementation of the logical schema in a given DBMS requires a very detailed knowledge of the specific features and facilities that the DBMS has to offer. In an ideal world, and in keeping with good software engineering practice, the first stage of implementation would involve matching the design requirements with the best available implementing tools and then using those tools for the implementation. In database terms, this might involve choosing vendor products with DBMS and SQL variants most suited to the database which is to be implemented. There are many relational DBMSs, available such as Oracle Database, Microsoft SQL Server, MySQL, IBM DB2, IBM Informix and Microsoft Access, use SQL. In this project we used Oracle SQL developer create the following tables of Fleet Management System database. 4.2 Database Tables Following tables table 4.1 to table 4.6 are the tables created for the schema diagram shown in figure 3.2.

## 4.2 Database Tables

Following tables table 4.1 to table 4.8 are the tables created for the schema diagram shown in figure 3.2.

4.1 Locations Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| lid | int | Primary Key |
| pin | Char (6) | Not Null |
| area | Varchar (70) | Not Null |
| city | Varchar (20) | Not Null |
| state | Varchar (20) | Not Null |

4.2 Branch Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| B\_id | int | Primary Key |
| B\_name | Varchar (40) | Not Null |
| B\_location\_id | int | Foreign Key References Locations |
| B\_manager\_id | int | Foreign Key References Employee |

4.3 Employee Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| e\_id | int | Primary Key |
| e\_name | Varchar (20) | Not Null |
| e\_dob | date | Not Null |
| E\_salary | int | Not Null |
| E\_experience | int | Not Null |
| E\_contact\_no | int | Not Null |
| E\_doj | date | Not Null |
| E\_branch\_id | int | Foreign Key References Branch |

4.4 Warehouse Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| W\_id | int | Primary Key |
| W\_name | Varchar (40) | Not Null |
| W\_lid | int | Foreign Key References Locations |
| W\_Bid | int | Foreign Key References Branch |

4.5 Consignment Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| cid | number | Primary Key |
| material | Varchar (30) | Not Null |
| consigner | Varchar (40) | Not Null |
| consignee | Varchar (40) | Not Null |
| deldate | date | Not Null |
| Amt\_paid | Decimal (10,2) | Not Null |
| Amt\_due | Decimal (10,2) | Not Null |
| weight | Decimal (10,2) | Not Null |
| flammability | Char (1) | Not Null |
| c\_wid | number | Foreign Key References Warehouse |

4.6 Client Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| client\_id | int | Primary Key |
| cl\_name | Varchar (50) | Not Null |
| cl\_lid | int | Foreign Key References Locations |
| c\_id | int | Foreign Key References Consignment |

4.7 Vehicle Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| vregno | Varchar (20) | Primary Key |
| vinsurance | Varchar (30) | Not Null |
| vtype | Varchar (50) | Not Null |
| vstatus | Varchar (20) | Not Null |
| carrier\_capacity | Decimal (10,2) | Not Null |
| vehicle\_permit | Varchar (20) | Not Null |
| consid | number | Foreign Key References |
| branch | int | Foreign Key References |

4.8 Conveyance Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| src | Varchar (50) | Not Null |
| dst | Varchar (50) | Not Null |
| distance\_km | Decimal (10,2) | Not Null |
| E\_id | int | Foreign Key References |
| vregno | Varchar (20) | Foreign Key References |

Following syntax shows for creating database table shown with an example for the

**Table 4.1 Database Creation, locations**

create table locations

(

lid int constraint plid primary key,

pin char(6),

area varchar(70),

city varchar(20),

state varchar(20) };

**Table 4.2 Database Creation, Branch**

create table branch

(

B\_id int,

B\_name varchar(40),

B\_location\_id int,

B\_manager\_id int,

primary key(B\_id),

foreign key(B\_location\_id) references locations(lid)

);

**Table 4.3 Database Creation, Employee**

create table employee

( e\_id int,

e\_name varchar(20),

e\_dob date,

e\_salary int,

e\_experience int,

e\_contact\_no int,

e\_doj date,

e\_branch\_id int,

address int constraint fa references locations(lid),

primary key(e\_id),

foreign key (e\_branch\_id) references branch(B\_id)

);

**Table 4.4 Database Creation, Warehouse**

create table warehouse

(

W\_id int,

W\_name varchar(20),

W\_lid int,

W\_Bid int,

primary key(W\_id),

foreign key (W\_Bid) references branch(B\_id),

foreign key (W\_lid) references locations(Lid)

);

**Table 4.5 Database Creation, Consignment**

create table consignment

(

cid number,

material varchar(30),

consigner varchar(40),

consignee varchar(40),

deldate date,

Amt\_paid decimal(10, 2),--in Lakhs

Amt\_due decimal(10, 2),

weight decimal(10, 2), --ton

flammability char(1),

c\_wid number,

primary key(cid),

foreign key(c\_wid) references warehouse(w\_id)

);

**Table 4.6 Database Creation, Client**

create table client

(

client\_id int,

cl\_name varchar(50),

cl\_lid int,

c\_id int,

primary key(client\_id),

Foreign Key(cl\_lid) references locations(lid),

Foreign key (c\_id) references consignment(cid)

);

**Table 4.7 Database Creation, Vehicle**

create table vehicle

(

vregno varchar(20),

vinsurance varchar(30),

vtype varchar(50),

vstatus varchar(20),

carrier\_capacity decimal(10,2),

vehicle\_permit varchar(20),

consid number constraint fc references consignment(cid),

branch int constraint fbid references branch(B\_id),

primary key (vregno)

);

**Table 4.8 Database Creation, Conveyance**

create table conveyance

(

src varchar(50),

dst varchar(50),

distance\_km decimal(10,2),

e\_id int constraint feid references employee(e\_id),

vregno varchar(20) constraint fno references vehicle(vregno),

primary key(e\_id,vregno)

);

## 4.3 Results

Figures Fig.4.3.1, 4.3.2, 4.3.3, 4.3.4 show the sample reports of the proposed system. Tables 4.3.1, 4.3.2, 4.3.3, 4.3.4 show consignments having delivery date 11-June-2021, client details who ordered such consignments, vehicles carrying them and employees who deliver the same.

Table 4.3.1 (a)

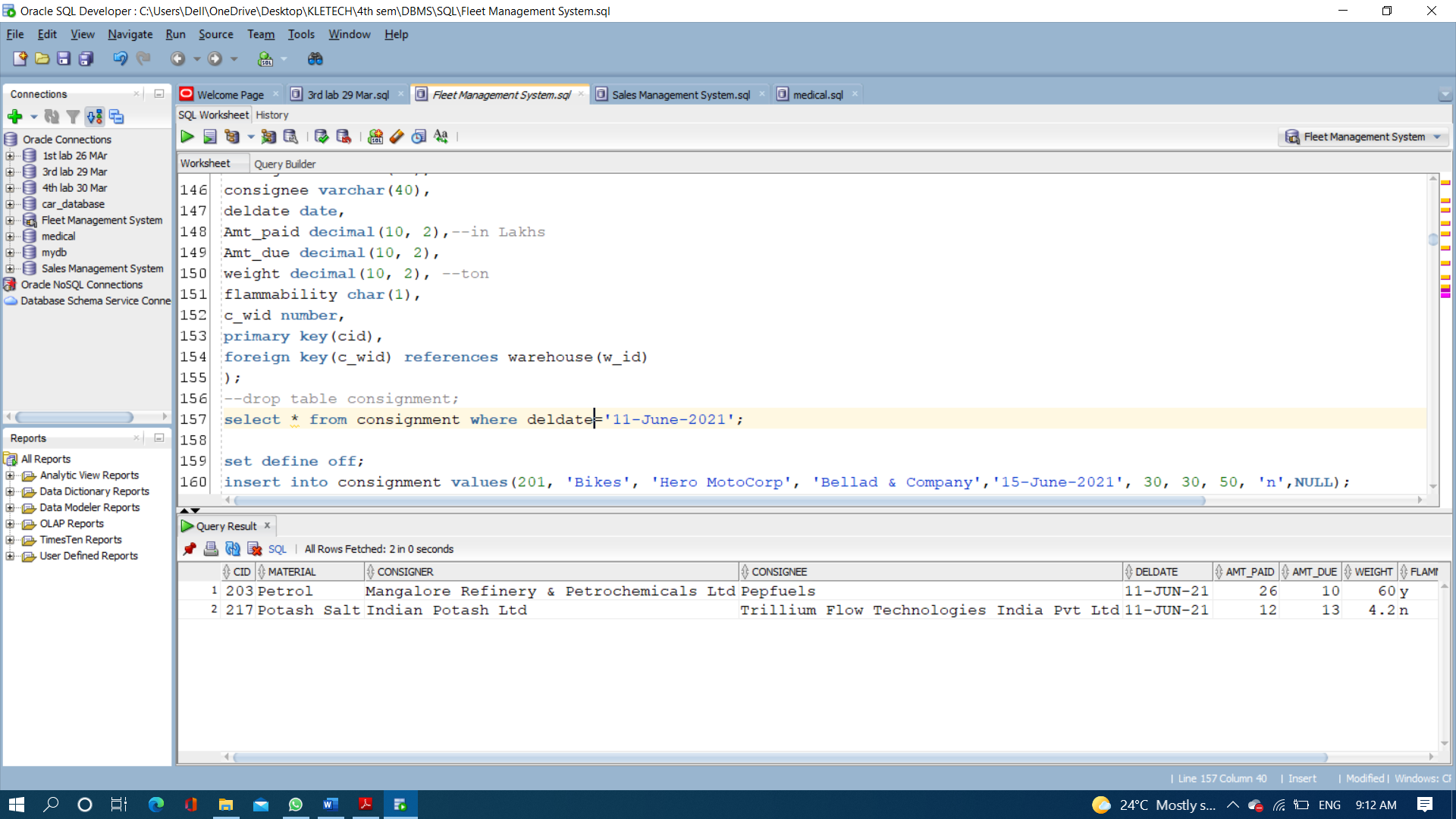


Table 4.3.1 (b)

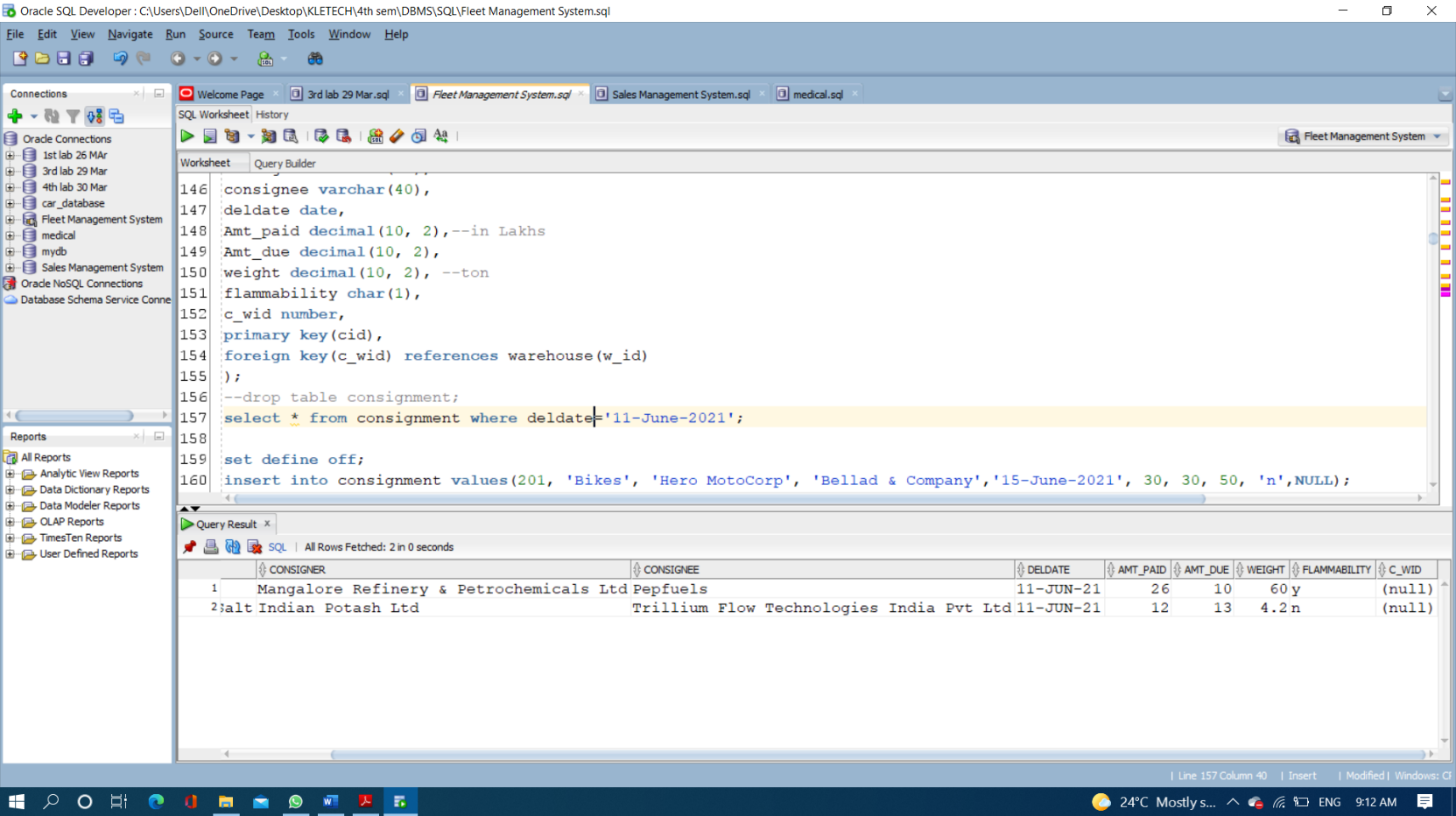


Table 4.3.2

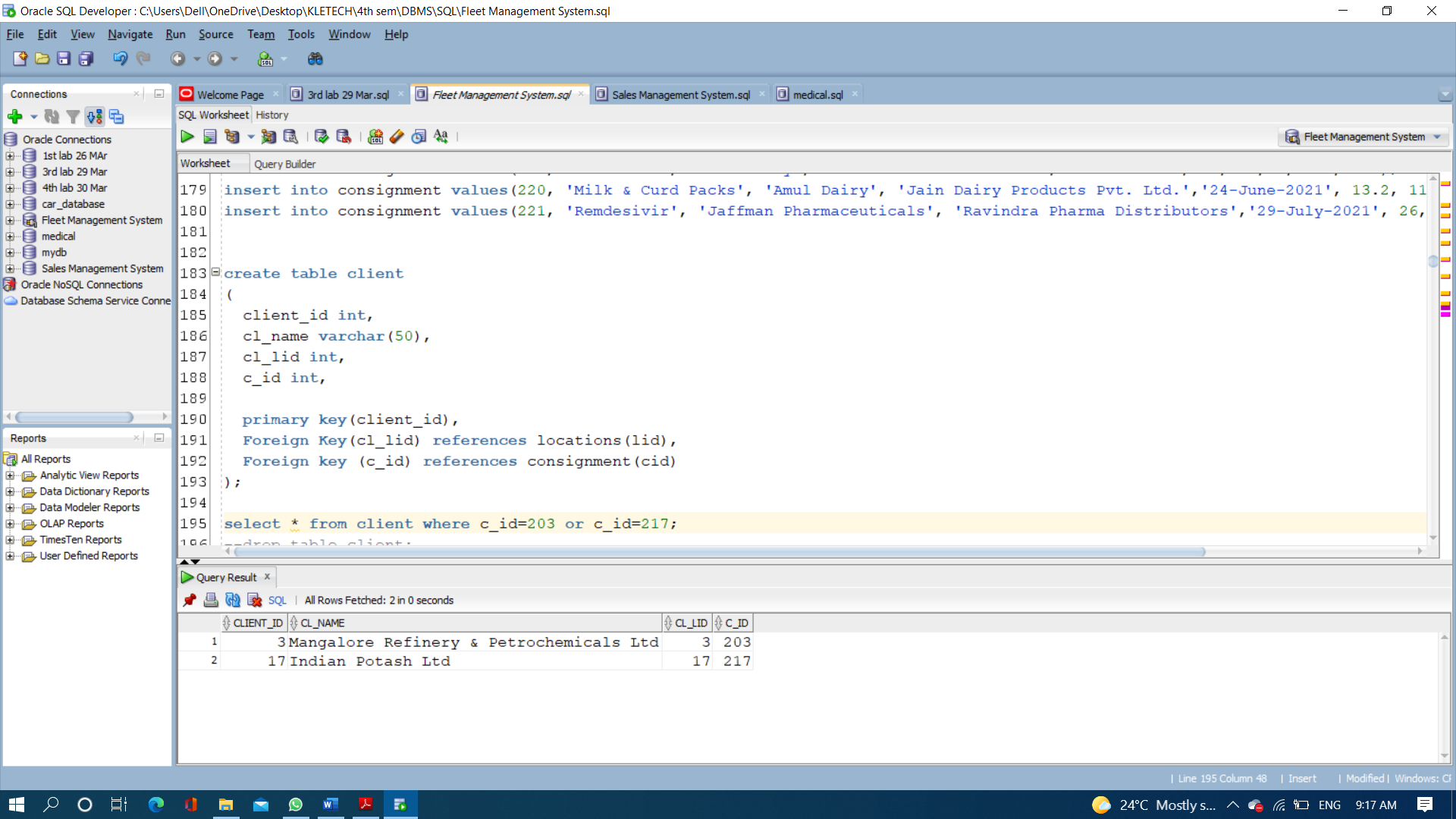


Table 4.3.3

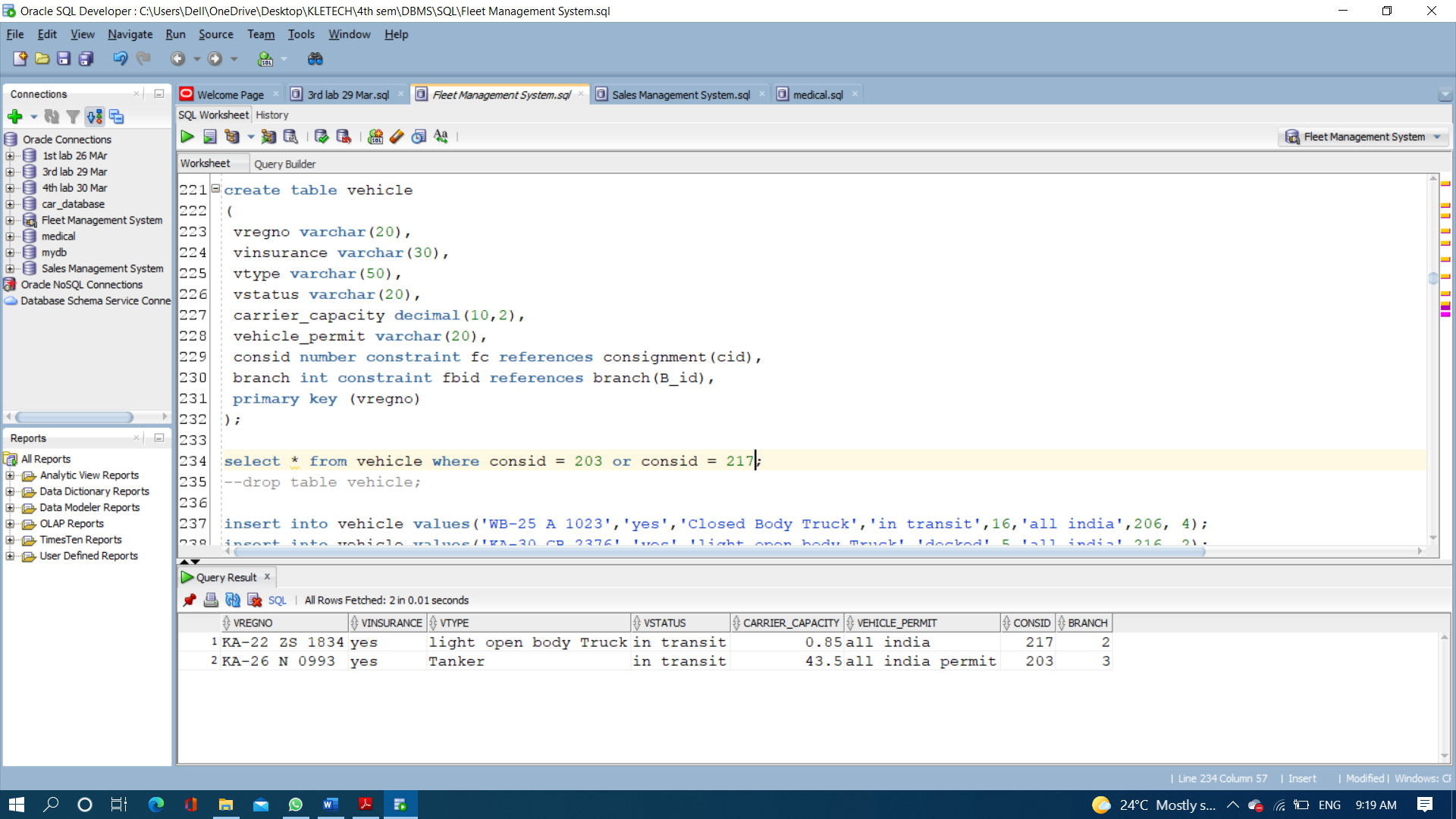
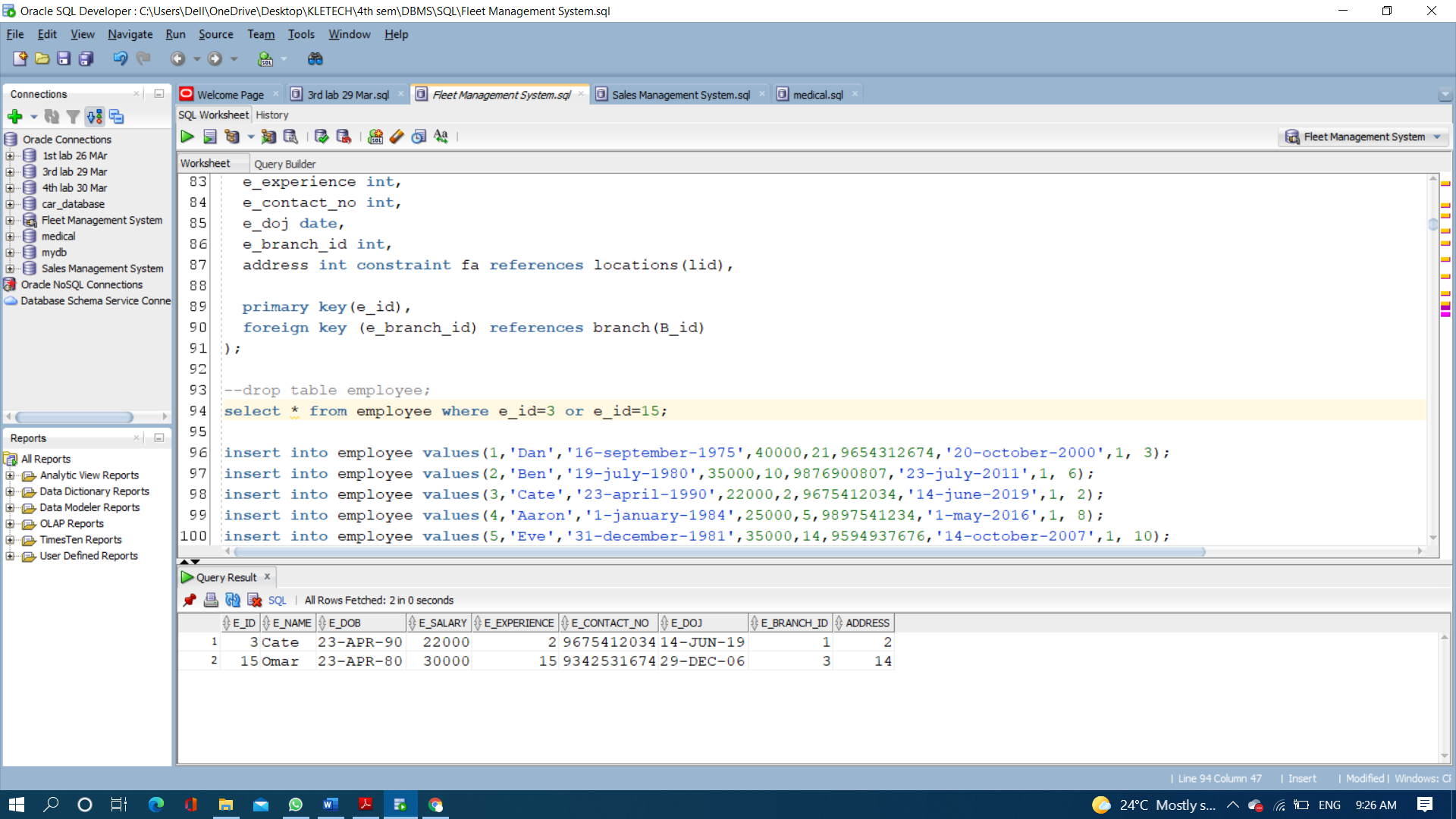


Table 4.3.4



# **REFERENCES**

# https://en.wikipedia.org/wiki/Fleet\_management