## Technical Report

Yalem Gebremariam

April 1, 2023

**Travel\_Reservation Database**

# Database Design Document

**Version 1.0**

**04/01/2023**

**Table of Contents**

[Database Design Document 1](#_Toc131429809)

[Introduction 1](#_Toc131429810)

[1. Overview 2](#_Toc131429811)

[2. Literature Review 3](#_Toc131429812)

[3. Assumptions/Constraints/Risks 4](#_Toc131429813)

[3.1 Assumptions 4](#_Toc131429814)

[3.2 Constraints 4](#_Toc131429815)

[3.3 Risks 4](#_Toc131429816)

[4. Design Decisions 5](#_Toc131429817)

[4.1 Key Factors Influencing Design 5](#_Toc131429818)

[4.2 Functional Design Decisions*.* 5](#_Toc131429819)

[4.3 Database Management System Decisions 5](#_Toc131429820)

[4.4 Security and Privacy Design Decisions 5](#_Toc131429821)

[4.5 Performance and Maintenance Design Decisions 5](#_Toc131429822)

[Statement of Work 7](#_Toc131429823)

[**1** **Overview:** 7](#_Toc131429824)

[**1.1** **Executive Summary** 7](#_Toc131429825)

[**2** **Purpose and Objectives:** 7](#_Toc131429826)

[**2.2** **Project Scope** 8](#_Toc131429827)

[**2.2.1** **Work within the scope of my project** 8](#_Toc131429828)

[**2.2.2** **Work outside the scope of my project** 8](#_Toc131429829)

[**2.3** **Database Goals, Expectations, and Deliverables** 8](#_Toc131429830)

[**2.4** **Database Benefits** 8](#_Toc131429831)

[**3** **Hardware and Software** 8](#_Toc131429832)

[**3.1** **Hardware** 8](#_Toc131429833)

[**3.2** **Software Tools** 8](#_Toc131429834)

[**3.2.1** **Diagramming Tool Identified** 9](#_Toc131429835)

[**3.2.2** **Database and Access Method Identified** 9](#_Toc131429836)

[**4** **Project Management Methodology** 9](#_Toc131429837)

[**4.1** **SQL Usage and Style Guide** 9](#_Toc131429838)

[Requirements Analysis 10](#_Toc131429839)

[Detailed Database Design 15](#_Toc131429840)

[4.6 Data Software Objects and Resultant Data Structures 15](#_Toc131429841)

[4.7 Database Management System Files 15](#_Toc131429842)

[**DML and Query Source Code** 20](#_Toc131429843)

[5. Database Administration and Monitoring 47](#_Toc131429844)

[5.1 Roles and Responsibilities 47](#_Toc131429845)

[5.2 System Information 47](#_Toc131429846)

[5.2.1 Database Management System Configuration 47](#_Toc131429847)

[5.2.2 Database Support Software 47](#_Toc131429848)

[5.2.3 Security and Privacy 47](#_Toc131429849)

[5.3 Performance Monitoring and Database Efficiency 47](#_Toc131429850)

[5.3.1 Operational Implications 48](#_Toc131429851)

[5.3.2 Data Transfer Requirements 48](#_Toc131429852)

[5.3.3 Data Formats 48](#_Toc131429853)

[5.4 Backup and Recovery 48](#_Toc131429854)

**List of Figures**

Figure 1 – Appendix A 49

**List of Tables**

[Table 2 - Acronyms](#_41mghml) 51

## Introduction

This technical report seeks to explain a travel agency database system. Trip Executive company (TEC inc.) is a trip and travel agency that offers services of making travel arrangements from booking flights, rental cars, making hotel reservations, and sightseeing tours recommendations.

The examples used for TEC in this document for travel arrangements revolve around booking or reserving, hotel, flights and car rentals for customers. TEC currently does not have an online booking system and bookings are done through the phone. This document will be used to implement a trip booking and customer information tracking database in a travel agency, where the audience would be the management team at TEC.

## Overview

This database will interact directly with an application sever that receives information regarding customer information and, travel agents inputs of available facilities, allow the travel agent to receive and approve. Objective for this database system include the ability to protect customer information, store large quantities of data (ex. Customer payment information, reservations, information on available facilities for booking), to reduce costs and improve efficiency by eliminating manual entry of reservation information by the agents. This database system has three main dependencies. The first and most important is the application sever. The application is what will coordinate all the external data and funnel it into the database. This application server is responsible for receiving reservation requests, payments, input from the hotel, car rental and airline companies and returning confirmation to the customers. The next dependency is the agency computers that will be communication with the application sever by sending booking information. The third dependencies is the agents access to computers in order to review and evaluate data.

Since there is no DBMS system in place at TEC Inc., this database will not interfere with any current database systems. However, great lengths will have to be taken in order to ensure proper interfaces exist to communicate with the TEC application server. This will probably be done though using a .NET coding framework. Overall, this DMBS system will be an asset for TEC Inc. as they are booking reservations manually. This DBMS system will help improve the speed of inquireing and making reservations, increase productivity, and improve the data integrity of customers information.

## Literature Review

Introduction

Much research on database management systems in a diverse field of study has been done. Database management system is the backbone of an organization that determines its functionality and productivity. This project explores the process of creating a database system for Trip Executive Company (TEC inc.). Initially a literature review was done to assess the different applications of database management systems in different fields of study. A short review of each is done that states the general objective, methodology used, general findings and conclusions of the articles. Based on this a general conclusion is given highlighting the findings and a connection is made to the specific project of creating database management system for Trip Executive Company(TEC inc.).

Boyle, J., Rovira, H., Cavnor, C. C., Burdick, D. M., Killcoyne, S., & Shmulevich, I. (2009b). Adaptable data management for systems biology investigations. BMC Bioinformatics, 10(1). https://doi.org/10.1186/1471-2105-10-79

The article, Adaptable data management for systems biology investigations is unique as each test done varies from one another. The aim of the research evolves over the research as the data is collected from a continues changing technologies used. The result of the article shows that an adaptable data management system is constructed to help the continuous mining and investigation for biological tests data. The use of changing techniques ranges from in-house use to high-tech use. For this data management systems are implemented. This is important as it needs to be readily developed and adapted to the new use. Based on the research done the article states that data management is essential for any research business. It is the backbone of most applications and must be ready to be applied for newly developed areas.

Metnitz, P. G. H., & Lenz, K. (1995). Patient data management systems in intensive care — the situation in Europe. Intensive Care Medicine, 21(9), 703–715. https://doi.org/10.1007/bf01704737

The article, Patient data management systems in intensive care-the situation in Europe, is an article that aims to investigate the functionality of computerized patient data management systems (PDMA) in Europe. The methodology used for the research was a questionnaire that explores the benefit and functionality. Based on the research done major differences in characteristics and performance were identified between the different systems. The article concludes that systems that use graphic data have more advantages than systems that use numerical data. Data can be communicated with different industry partners together with hospital management for the planning phase, buying phase, implementing phase of computerized patient data management systems (PDMA). The research also recommends that better communication can be applied between nurses, scientists and clinicians and vendors.

Pangalos, G. (1988). Design of hospital database systems in a non-relational environment. Journal of Medical Informatics, 13(1), 27–33. https://doi.org/10.3109/14639238809003572

The article, design for hospital database systems in a non-relational environment focuses on non-relational database management systems. The articles states the lack of design procedure that is suitable for non-relational database management systems. It continues to argue the importance of these design procedure since many of the major database management systems that are used today are non-relational. It proposes a design procedure that supports the design of ideal non-relational hospital database. An example was used to develop the procedure further for future betterment of the design.

Rodríguez-Terol, A., Caraballo, M., Palma, D. A., Santos-Ramos, B., Molina, T. J., Desongles, T., & Aguilar, A. (2009b). Quality of interaction database management systems. Farmacia Hospitalaria, 33(3), 134–146. https://doi.org/10.1016/s2173-5085(09)70079-6

The article, quality of interaction database management systems is about identifying “drug interaction databases (DID) and it examines the structural quality. The methodology used in this research was a literature for DID. The process defined a sequence of qualities of their structures and elimination criteria. Four criteria were used to assess the quality which are classifying based on severity, classifying based on evidence, reference data, classifying based on clinical management. The research finding identified 54 DID, 30 out of these were collected with the elimination criteria and 15 did not pass the criteria. Based on the finding the research concludes that a third of the DID does align with the minimum criteria. Some differences were recognized between foreign and local DID. Some of the reference DID used were structurally defected.

Conclusion

Database management system is an integral part of an organization. An organization relies on precise and consistent data for efficient decision making. The relationship of these data is maintained and recorded within the organization which are the database (Gunjal, 2003). The collection of these related database is the database system which also includes the description of the data. The whole purpose of database management system is to record and keep track of information. This is very important for the functionality and success of an organization (Metnitz & Lenz, 1995). As the workflow and relationship of one section to another depends on these data that are recorded. The databases that are recorded should be easy, adaptable, economical, and fast to access. Database is collection of data that can be shared by many users (Boyle et al., 2009b). The application of database management system is not limited and can be applied to many industries. Based on the literature review the different application of database management in different sector has been useful for this project.

This project explores the process of creating a database system for Trip Executive Company (TEC inc.). The database will be used to increase the productivity and accuracy of the company. Similar to the conclusion reached by the articles reviewed, the database system will reduce costs, increate the turn-around time for testing, and decrease redundant errors. The methodology used for the design of the database will be using agile framework. Even though the database system is designed to be used in a travel agency, the project is limited to virtualized environment. The findings from the literature review were useful to guide and compare findings in the project. As the literature review explores the functionality, method, gaps and recommendation for future use.

## Assumptions/Constraints/Risks

### Assumptions

Assumptions made in this document is that TEC agency is currently using a Windows 11 operating system, that TEC Inc. will be purchasing new hardware for the DBMS system, and that TEC Inc. will be upgrading and improving the DBMS system for specific site functionality and tracking. TEC Inc. also understand the need to upgrade old hardware, such a Intel I2 processors and hardware discs to solid state drives, to improve the efficiency of the application severs capabilities.

### Constraints

Current limitations of this data base system is that it can only record reservations of hotel, car rental and flight information. As TEC Inc. grows new tables and reservation tracking capabilities will have to be added to keep up with additional site seeing and booking reservations.

### Risks

Risked associated with this database design is that it is limited by five database fields, customer, reservation account, hotel reservation, flight reservation and, car rental reservations. For this database to be applied at TEC agency it would need more fields and interactions than are listed in this report in order to be a fully automatic booking system.

## Design Decisions

### Key Factors Influencing Design

Key factors that influenced the database design revolved around the application servers availability to input and output data from the DBMS system. The design aspects that were crucial to the database design are the following transactions: 1) Receiving patient information. 2) Obtaining available facilities for booking. 3) Obtaining payment approval. 4) Generating a confirmation of reservations. Database deliverables include the database, the database documents such as ERD diagrams and table structures, and scripts to simulate the above transactions. At TEC Inc. it is the IT departments responsibility to make the application server and database servers are secure and limit user access accordingly to protect patient data.

### Functional Design Decisions*.*

The functional design decisions of this DBMS system revolve around the application server. The application server is where information is received, imputed, updated, edited, and or distributed and the application server will take the import source .csv or .txt file and will parse and input it into subject query language (SQL) so it can be saved in the database. The application server will take the output from SQL and convert it into a .pdf results file to send to the agent. The application server will be accessible by both Windows and Linux operating systems. When an agent of TEC Inc. needs access to a database file, they will interact with the database through a graphical user interface (GUI), that is simplified and easy to use. The GUI will be able to run on Windows 11 or higher operating systems.

### Database Management System Decisions

For the initial implementation of this database system, there is a broad overlook in handling customer information and reservation data. As the system is updated over time, the categories will become more specific and stream lined. The name of the DBMS is Travel\_Reservation version 1.0. Since this database was designed broadly, it has room to grow as TEC Inc. grows. The general format and table designs have been clearly implanted and can be elaborated on in the future.

### Security and Privacy Design Decisions

The security of this database system relies heavily on data compartmentalization. In this version of the booking reservation database, there are two types of users, customers, travel agents. Travel agents have access to customers information. It is up to IT to have proper firewalls in place to protect the application server as well as the DBMS system.

### Performance and Maintenance Design Decisions

* *.*In order to maintain database consistently only one reservation file per a particular hotel room and car rental can be opened at a time. Access will be denied if a record is currently being processed or updated. After the user saves and pushes the changes to the application server and closes the query a different date reservation for that particular hotel room and car rental can be accessed by an agent. The database will be backed up daily at 0:00, and all updates will occur on a Sunday since TEC Inc. is closed on Sundays.
* To reduce concurrence issues, data will be imported into the application server directly, it will come from the hotel, airline and car rental electronically.
* The DBMS system is backed up nightly. If the DBMS system were to require a restoration step the GUI would interact with the DBMS file and only import files that currently do not exist in the database therefore elevating duplicates. Nonstandard technologies will not be imported into the database. The GUI will only except certain file types. System maintenance will be performed on the weekend as to not impact sample processing during the week. While the system is going through a maintenance step, data cannot be exported or imported into the database.
* The database is designed in a top down approach where all the information necessary for a table is in one location. Since this database system will be using solid state drives they will not have to be defragged and will perform at a faster level. This database will continue to grow over time as additional site seeing and vacationing options become available. The solid state drives will be daisy chained in order to have expanding storage over time. The system will not be able to receive or edit information during a maintenance period.
* Customer information and data will be automatically archived by the DBMS system after it has been three years. This will be done by moving the data to a new location and then exporting the data onto a removable memory device and placed in storage. This process will not be affected by maintenance as it occurs in real time at the beginning of a week day.

## Statement of Work

**Statement of Work**

1. **Overview:**

Trip Executive company (TEC inc.) is a trip and travel agency that offers services of making travel arrangements from booking flights, rental cars, making hotel reservations, and sightseeing tours recommendations.

TEC is in the process of acquiring a program to give online booking services and upgrading the current system used.

* 1. **Executive Summary**

A database will be created for TEC in order to increase the efficiency and accuracy of their operations by streamlining the tracking of reservations, and obviating the need for agents to engage in laborious data entry and transcription. This will reduce costs, increase the turn-around time for reservations, and decrease the frequency of errors by reducing manual activities. Each customer will need to be provided with a customer ID, which is associated with an electronic customer record. This ID will be used to track customer’s reservation history and current active reservations in the database. The different reservation numbers that will be generated as reservations are made will be used to track different reservations in the database, which will be accessible by the agency.

1. **Purpose and Objectives:**
   1. **Objectives of your database project**

The database will be used to store customers’ hotel reservation confirmation number, flight number, bank account information used to make payments, car rental confirmation number and the prices for the services offered. This database will interact directly with an application sever that receives information regarding electronic customer information and reservations made, agency staff member’s inputs from taking booking information and the customer’s input on the official TEC website, allow reservations to be reviewed by designated agency staff member , and also electronically return the confirmation number for the reservations and customer’s account number to the customer and the agency.

Objective for this database system include the ability to protect customer data including bank account information, store large quantities of data (ex. customer information, prices, reservation information), to increase profitability and improve efficiency by providing easy to access information.

* 1. **Project Scope**

This database system is designed to be implemented in a travel agency, however for the purposes of this project it will be created in a virtualized environment. Interactions with peripheral IT systems will be simulated because those IT systems will not be created.

**2.2.1** **Work within the scope of my project**

* + - The creation of a database system that will store input for customer information, flight, hotel, and rental car reservations, and price information.

**2.2.2** **Work outside the scope of my project**

* + - Capabilities of other IT systems and software programs to interact with the database, including:
      * The creation and sending of electronic customer information to the service providing agencies.
      * The creation of sending of reservation confirmation numbers.
  1. **Database Goals, Expectations, and Deliverables**

The database goals and expectations include simulating the following transactions: 1) Recording customer information including bank account information. 2) Recording booking preferences and prices for the customer. 3) Obtaining confirmation of the booking from the hotel, car rental and airline. 4) Generating a confirmation page to be sent out to the customer and the service providing companies. Database deliverables include the database, the database documents such as ERD diagrams and table structures, and scripts to simulate the above transactions.

* 1. **Database Benefits**

Benefits of this database system would increasing profitability by giving customers easy access to different booking options online and increasing efficiency of the in person/over the phone bookings as well by providing the agency the means of storing and accessing customer information mor effectively. This will increase productivity and customer satisfaction. The database will also provide security measures to protect sensitive customer information.

1. **Hardware and Software**

This section will discuss the hardware and software to be used in the UMGC project.

* 1. **Hardware**

ASUS Vivo Book Intel® Core™ i3-8130U Processor 2.2 GHz (4M Cache, up to 3.4 GHz, 2cores)

* 1. **Software Tools**

The following software tools will be used in this project:

* + - Oracle SQL Developer Version 22.2.1 will be used for database creation and manipulation.
    - Oracle Database 19c, using Virtual Desktop Access (VDA)
    - ER Assistant Version 2.10 will be used for diagramming
    - Google Chrome Version 105.0. 5195.127 will be used.
    - Microsoft Office 365, running on Windows
    1. **Diagramming Tool Identified**

ER Assistant will be used running on Windows 11 as the diagramming tool with cows feet notation.

* + 1. **Database and Access Method Identified**

UMGC VDA connection with Intel® Core™ i3-8130U Processor 2.2 GHz (4M Cache, up to 3.4 GHz, 2cores), running on the Windows 11 operating system. This 64-bit operating system has 8.GB of ram. Oracle SQL developer will be used to create the database. a consumer-class computer running Windows, macOS, or Linux

1. **Project Management Methodology**

The design of the database will be carried out using the agile framework. This project management framework will have characteristics of continuous adaptation and frequent iteration. Following this approach will have the benefits of having increased flexibility and easily adapt to change when working with the software development team.

* 1. **SQL Usage and Style Guide**

(Adapted from Simon Holywell’s SQL style guide, available at <http://sqlstyle.guide/.)>

* + - SQL statement structure for readability
      * SQL statements will be broken up over multiple lines to insure readability
      * SQL statements will include white space and indentation to make code easier to read
    - Script format comment usage
      * Comments will begin with “/\*” and end with “\*/” (C-style comments)
      * If a comment begins on a line with a SQL statement “--" will be used as the being of the comment (but the comment is limited to this line of code only)
    - Object naming conventions
      * All names will begin with a letter and can not end in an underscore
      * For names with more than one word, an underscore will be used to separate the words (snake\_case).
      * Names will be descriptive
      * Dates and time will be inputted as YYYY-MM-DD, HH:MM:SS.SSSSS (ISO- 8601 format)
      * A name should be unique and not have a conflict with reserved SQL keywords

## Requirements Analysis

Definition Document

This definition document seeks to plan for and explain the ERD diagram for TEC Inc, and describe the relationships and business rules between the different tables.

**Business Rules, Relationship and Cardinality Description**

1. Relationship: rents between Customer and RentalCar\_reservation

Cardinality: 1:M between Customer and RentalCar\_reservation

Business rule: a customer can have zero to many rental car reservations; a rental car can only be reserved for one and only one customer.

1. Relationship: stays between Customer and Hotel\_reservation

Cardinality: 1:M between Customer and Hotel\_reservation

Business rule: a customer can have zero to many hotel reservations; a hotel can only be reserved for one and only one customer.

1. Relationship: reserves between Flight\_reservation and customer

Cardinality: 1:M between Flight\_reservation and customer

Business rule: a flight can have zero to many customers; a customer can only reserve one and only one flight at a time.

1. Relationship: has between Customer and ReservationAccount

Cardinality: 1:1 between Customer and ReservationAccount

Business rule: a customer can have one and only one reservation account; a reservation account should only belong to one and only one customer.

**Entity and Attribute Description**

* + 1. Entity Name: Flight\_reservation

Entity Description: reservation of flight seat for a customer

Main attributes of Flight\_resrvation:

Attribute Name: seat\_number

Attribute Description: seat number.

Attribute Name: L\_name

Attribute Description: last name.

Attribute Name: Flight\_reserv\_number

Attribute Description: flight reservation number.

Attribute Name: Flight\_type

Attribute Description: flight type (one-way or two-way).

Attribute Name: TIME\_flight

Attribute Description: time of flight.

Attribute Name: DATE\_flight

Attribute Description: date of flight.

* + 1. Entity Name: Hotel\_reservation

Entity Description: reservation for hotel stay for customer

Main attributes of Hotel\_reservation:

Attribute Name: Hotel\_resrv\_no

Attribute Description: hotel reservation number.

Attribute Name: L\_NAME

Attribute Description: last name.

Attribute Name: Room\_number

Attribute Description: room number.

Attribute Name: no\_bedrooms

Attribute Description: number of bedrooms.

Attribute Name: no\_beds

Attribute Description: number of beds.

Attribute Name: no\_nights

Attribute Description: number of nights at hotel

* + 1. Entity Name: RentalCar\_reservation

Entity Description: reservation of rental cars by customer

Main attributes of RentalCar\_reservation:

Attribute Name: Car\_resrv\_no

Attribute Description: car reservation number.

Attribute Name: L\_NAME

Attribute Description: last name.

Attribute Name: License\_no

Attribute Description: license number.

Attribute Name: Make\_model\_car

Attribute Description: the make and model of car.

Attribute Name: PickUp\_DATE

Attribute Description: pick up date.

Attribute Name: DropOff\_DATE

Attribute Description: Drop off date.

* + 1. Entity Name: ReservationAccount

Entity Description: Reservation account information for all reservation types

Main attributes of Reservation Account:

Attribute Name: L\_NAME

Attribute Description: last name.

Attribute Name: Reservation\_type

Attribute Description: reservation type.

Attribute Name: No\_reservation

Attribute Description: total number of reservations

Attribute Name: cost

Attribute Description: cost of the service.

Attribute Name: No\_guests

Attribute Description: number of guests on the reservation

Attribute Name: Card\_no

Attribute Description: payment card no.

Attribute Name: Reservation\_no

Attribute Description: reservation no.

* + 1. Entity Name: Customer

Entity Description: customer making the reservation.

Main attributes of Customer:

Attribute Name: L\_NAME

Attribute Description: last name.

Attribute Name: F\_NAME

Attribute Description: first name.

Attribute Name: Balance

Attribute Description: Outstanding balance.

Attribute Name: Special\_requests

Attribute Description: Special requests.

Attribute Name: Length\_Stay

Attribute Description: Length of stay.

Attribute Name: Customer\_ID

Attribute Description: Customer ID (optional).

**Assumptions and Special Considerations**

It is important to note that a customer can exist without current active reservations. The customer’s past reservations and transactions will still be present in the database.

## Detailed Database Design

### Data Software Objects and Resultant Data Structures

There are no data software objects and resultant data structures in this DBMS system. Non-DBMS files that may be required to keep customer information is a .csv or .txt file from the application. Output files are generated by GUI to send back to the application.

### Database Management System Files

* *.*There are five main database schemas customer, reservationaccount, hotel\_reservation, carrental\_reservation and, flight\_reservation. See the ERD in Appendix A.
* Multiple indexes were made for patient (first and last name), doctor (first and last name), a patient sample index (patient id), and employee index on test record (employee ID), a instrument index on test record (instrument id), a test sample index on test record (sample number), and lastly a patient test index on test record (patient id).
* The SQL file is grouped into the five main schemas with their associated DML and DDL grouped with them, and has the twenty useful queries at the end .

**DDL Source Code**

/\* DROP statements to clean up objects from previous run \*/

--Dropping tables

DROP TABLE Customer CASCADE CONSTRAINTS;

DROP TABLE Hotel\_reservation CASCADE CONSTRAINTS;

DROP TABLE Flight\_reservation CASCADE CONSTRAINTS;

DROP TABLE RentalCar\_reservation CASCADE CONSTRAINTS;

DROP TABLE ReservationAccount CASCADE CONSTRAINTS;

--Dropping sequences

DROP SEQUENCE Customer\_ID\_SEQ;

DROP SEQUENCE Hotel\_reservation\_no\_SEQ;

DROP SEQUENCE Flight\_reservation\_no\_SEQ;

DROP SEQUENCE Car\_reservation\_no\_SEQ;

DROP SEQUENCE Reservation\_no\_SEQ;

--Dropping views

DROP VIEW multiple\_reservation;

DROP VIEW UNPAID\_ACCOUNT;

--Creating tables

CREATE TABLE Customer

(Customer\_ID VARCHAR2(30) NOT NULL PRIMARY KEY

,L\_NAME VARCHAR2(25)

,F\_NAME VARCHAR2(25)

,Balance VARCHAR2(25)

,Special\_requests VARCHAR2(25)

,Length\_Stay NUMBER(30,0)

);

CREATE TABLE ReservationAccount

(Reservation\_no NUMBER(7,0) NOT NULL PRIMARY KEY

,L\_NAME VARCHAR2(25)

,Reservaton\_typ VARCHAR2(25)

,No\_reservation VARCHAR2(25)

,R\_Cost\_DollarAmount VARCHAR2(50)

,No\_guests VARCHAR2(5)

,Card\_no VARCHAR2(15)

,Customer\_ID VARCHAR2(30),

FOREIGN KEY (Customer\_ID) REFERENCES Customer (Customer\_ID)

);

CREATE TABLE RentalCar\_reservation

(Car\_reservation\_no NUMBER(7,0) NOT NULL PRIMARY KEY

,L\_NAME VARCHAR2(25)

,License\_no VARCHAR2(25)

,Make\_model\_car VARCHAR2(25)

,PickUp\_DATE DATE

,Customer\_ID VARCHAR2(30),

FOREIGN KEY (Customer\_ID) REFERENCES Customer (Customer\_ID)

,DropOff\_DATE DATE

,Card\_no VARCHAR2(15)

);

CREATE TABLE Flight\_reservation

(Flight\_reservation\_no NUMBER(7,0) NOT NULL PRIMARY KEY

,seat\_no VARCHAR2(25)

,L\_NAME VARCHAR2(25)

,Flight\_type VARCHAR2(25)

,Date\_Flight\_departed DATE

,Customer\_ID VARCHAR2(30),

FOREIGN KEY (Customer\_ID) REFERENCES Customer (Customer\_ID)

,DATE\_flight\_arrived DATE

);

CREATE TABLE Hotel\_reservation

(Hotel\_reservation\_no NUMBER(7,0) NOT NULL PRIMARY KEY

,L\_NAME VARCHAR2(25)

,Room\_no VARCHAR2(25)

,no\_bedrooms NUMBER (3,0)

,no\_beds NUMBER (6,0)

,no\_nights NUMBER (30,0)

,Customer\_ID VARCHAR2(30),

FOREIGN KEY (Customer\_ID) REFERENCES Customer (Customer\_ID)

);

--Creating indexes

CREATE INDEX RES\_Customer\_ID\_FK ON ReservationAccount(Customer\_ID);

CREATE INDEX REN\_Customer\_ID\_FK ON RentalCar\_reservation(Customer\_ID);

CREATE INDEX Fli\_Customer\_ID\_FK ON Flight\_reservation(Customer\_ID);

CREATE INDEX HOT\_Customer\_ID\_FK ON Hotel\_reservation(Customer\_ID);

/\*This view shows balance, customer id of accounts that have outstanding balance\*/

CREATE VIEW Unpaid\_Account AS

SELECT Balance, Customer\_ID

FROM Customer

WHERE Balance != 0;

/\*This view shows number of reservations, reservation number of customers that have more than one reservations\*/

CREATE VIEW multiple\_reservation AS

SELECT No\_reservation, Reservation\_no

FROM ReservationAccount

WHERE No\_reservation > 1;

--creating sequences

CREATE SEQUENCE Customer\_ID\_SEQ

INCREMENT BY 1

START WITH 100

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

CREATE SEQUENCE Reservation\_no\_SEQ

INCREMENT BY 1

START WITH 500

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

CREATE SEQUENCE Car\_reservation\_no\_SEQ

INCREMENT BY 1

START WITH 1000

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

CREATE SEQUENCE Flight\_reservation\_no\_SEQ

INCREMENT BY 1

START WITH 2000

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

CREATE SEQUENCE Hotel\_reservation\_no\_SEQ

INCREMENT BY 1

START WITH 3000

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

/\*This trigger populates the primary key and audit columns with appropriate values\*/

CREATE OR REPLACE TRIGGER Customer\_ID\_TRG

BEFORE INSERT ON Customer

FOR EACH ROW

BEGIN

IF :NEW.Customer\_ID IS NULL THEN

:NEW.Customer\_ID := Customer\_ID\_SEQ.NEXTVAL;

END IF;

END;

/

/\*This trigger populates the primary key and audit columns with appropriate values\*/

CREATE OR REPLACE TRIGGER Reservation\_no\_TRG

BEFORE INSERT ON ReservationAccount

FOR EACH ROW

BEGIN

IF :NEW.Reservation\_no IS NULL THEN

:NEW.Reservation\_no := Reservation\_no\_SEQ.NEXTVAL;

END IF;

END;

/

/\*This trigger populates the primary key and audit columns with appropriate values\*/

CREATE OR REPLACE TRIGGER Car\_reservation\_no\_TRG

BEFORE INSERT ON RentalCar\_reservation

FOR EACH ROW

BEGIN

IF :NEW.Car\_reservation\_no IS NULL THEN

:NEW.Car\_reservation\_no := Car\_reservation\_no\_SEQ.NEXTVAL;

END IF;

END;

/

/\*This trigger populates the primary key and audit columns with appropriate values\*/

CREATE OR REPLACE TRIGGER Flight\_reservation\_no\_TRG

BEFORE INSERT ON Flight\_reservation

FOR EACH ROW

BEGIN

IF :NEW.Flight\_reservation\_no IS NULL THEN

:NEW.Flight\_reservation\_no := Flight\_reservation\_no\_SEQ.NEXTVAL;

END IF;

END;

/

/\*This trigger populates the primary key and audit columns with appropriate values\*/

CREATE OR REPLACE TRIGGER Hotel\_reservation\_no\_TRG

BEFORE INSERT ON Hotel\_reservation

FOR EACH ROW

BEGIN

IF :NEW.Hotel\_reservation\_no IS NULL THEN

:NEW.Hotel\_reservation\_no := Hotel\_reservation\_no\_SEQ.NEXTVAL;

END IF;

END;

/

-- Check the DBMS data dictionary to make sure that all objects have been created successfully

SELECT TABLE\_NAME FROM USER\_TABLES;

SELECT OBJECT\_NAME, STATUS, CREATED, LAST\_DDL\_TIME FROM USER\_OBJECTS;

**DML and Query Source Code**

/\*insert customer data\*/

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Gebremariam', 'Yalem', 100, null, 3);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Gebremariam', 'Yewoin', null, null, 2);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Habte', 'Kiflom', 25, null, 1);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Bekila', 'Haile', null, 'ramp accessible', 10);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Lemma', 'Rahel', null, null, 5);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Smith', 'Mike', 300, null, 2);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Tshige', 'Gebeyew', null, 'water view', 12);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Black', 'John', null, null, 1);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Kebede', 'Betty', 5, null, 1);

insert into customer values (Customer\_ID\_SEQ.NEXTVAL, 'Dante', 'Kevin', 50, 'balcony', 5);

commit;

/\*insert hotel reservation data\*/

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Gebremariam', 144, 2, 3, 3, (select Customer\_ID from customer where L\_name = 'Gebremariam' and F\_name = 'Yalem'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Gebremariam', 200, 1, 1, 2, (select Customer\_ID from customer where L\_name = 'Gebremariam' and F\_name = 'Yewoin'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Habte', 100, 3, 5, 1, (select Customer\_ID from customer where L\_name = 'Habte' and F\_name = 'Kiflom'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Bekila', 12, 1, 2, 10, (select Customer\_ID from customer where L\_name = 'Bekila' and F\_name = 'Haile'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Lemma', 145, 2, 2, 5, (select Customer\_ID from customer where L\_name = 'Lemma' and F\_name = 'Rahel'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Smith', 36, 1, 1, 2, (select Customer\_ID from customer where L\_name = 'Smith' and F\_name = 'Mike'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Tshige', 101, 3, 3, 12, (select Customer\_ID from customer where L\_name = 'Tshige' and F\_name = 'Gebeyew'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Black', 29, 1, 2, 1, (select Customer\_ID from customer where L\_name = 'Black' and F\_name = 'John'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Kebede', 205, 2, 2 , 1, (select Customer\_ID from customer where L\_name = 'Kebede' and F\_name = 'Betty'));

insert into Hotel\_Reservation values (Hotel\_reservation\_no\_SEQ.NEXTVAL, 'Dante', 206, 1, 1, 5, (select Customer\_ID from customer where L\_name = 'Dante' and F\_name = 'Kevin'));

commit;

/\*insert flight reservation data\*/

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 1, 'Gebremariam', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Gebremariam' and F\_name = 'Yalem'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 2, 'Gebremariam', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Gebremariam' and F\_name = 'Yewoin'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 3, 'Habte', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Habte' and F\_name = 'Kiflom'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 4, 'Bekila', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Bekila' and F\_name = 'Haile'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 5, 'Lemma', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Lemma' and F\_name = 'Rahel'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 6, 'Smith', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Smith' and F\_name = 'Mike'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 2, 'Tshige', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Tshige' and F\_name = 'Gebeyew'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 3, 'Black', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Black' and F\_name = 'John'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 5, 'Kebede', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Kebede' and F\_name = 'Betty'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

insert into Flight\_Reservation values (Flight\_reservation\_no\_SEQ.NEXTVAL, 1, 'Dante', 'two way', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Dante' and F\_name = 'Kevin'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'));

commit;

/\*insert rental car reservation\*/

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Gebremariam', 'G12', 'Toyota Corolla', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Gebremariam' and F\_name = 'Yalem'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 1);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Gebremariam', 'G13', 'Toyota Camry', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Gebremariam' and F\_name = 'Yewoin'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 2);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Habte', 'H12', 'Toyota Rav4', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Habte' and F\_name = 'Kiflom'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 3);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Bekila', 'B12', 'Volvo XC40', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Bekila' and F\_name = 'Haile'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 4);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Lemma', 'L12', 'Ford Mustang', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Lemma' and F\_name = 'Rahel'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 5);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Smith', 'S12', 'Ford Edge', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Smith' and F\_name = 'Mike'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 6);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Tshige', 'T12', 'Ford Escape', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Tshige' and F\_name = 'Gebeyew'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 7);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Black', 'B12', 'Ford Expolrer', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Black' and F\_name = 'John'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 8);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Kebede', 'K12', 'Dodge Charger', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Kebede' and F\_name = 'Betty'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 9);

insert into RentalCar\_Reservation values (Car\_reservation\_no\_SEQ.NEXTVAL, 'Dante', 'D12', 'Dodge Journey', TO\_DATE('30-JAN-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), (select Customer\_ID from customer where L\_name = 'Dante' and F\_name = 'Kevin'), TO\_DATE('02-Feb-2023 10:18:00','DD-MON-YYYY HH24:MI:SS'), 10);

commit;

/\*insert reservation account data\*/

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Gebremariam', 'All', 2, 300, 3, 1, (select Customer\_ID from customer where L\_name = 'Gebremariam' and F\_name = 'Yalem'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Gebremariam', 'All', 1, 100, 2, 2, (select Customer\_ID from customer where L\_name = 'Gebremariam' and F\_name = 'Yewoin'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Habte', 'All', 3, 500, 1, 3, (select Customer\_ID from customer where L\_name = 'Habte' and F\_name = 'Kiflom'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Bekila', 'All', 1, 200, 1, 4, (select Customer\_ID from customer where L\_name = 'Bekila' and F\_name = 'Haile'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Lemma', 'All', 2, 200, 2, 5, (select Customer\_ID from customer where L\_name = 'Lemma' and F\_name = 'Rahel'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Smith', 'All', 1, 100, 2, 6, (select Customer\_ID from customer where L\_name = 'Smith' and F\_name = 'Mike'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Tshige', 'All', 3, 300, 2, 7, (select Customer\_ID from customer where L\_name = 'Tshige' and F\_name = 'Gebeyew'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Black', 'All', 1, 200, 1, 8, (select Customer\_ID from customer where L\_name = 'Black' and F\_name = 'John'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Kebede', 'All', 2, 200 , 1, 9, (select Customer\_ID from customer where L\_name = 'Kebede' and F\_name = 'Betty'));

insert into ReservationAccount values (Reservation\_no\_SEQ.NEXTVAL, 'Dante', 'All', 1, 100, 1, 10, (select Customer\_ID from customer where L\_name = 'Dante' and F\_name = 'Kevin'));

commit;

/\* 20 SQL Queries \*/

-- Q1. Select all columns and all rows from one table

SELECT \* FROM Flight\_Reservation;

-- Q2. Select five columns and all rows from one table

SELECT Flight\_Reservation\_NO, Seat\_NO, L\_Name, Flight\_Type, date\_flight\_departed

FROM Flight\_Reservation;

-- Q3. Select all columns from all rows from one view

SELECT \* FROM Multiple\_Reservation;

-- Q4. Using a join on 2 tables, select all columns and all rows

SELECT \* FROM Customer LEFT OUTER JOIN Flight\_Reservation ON Customer.Customer\_id = Flight\_Reservation.Customer\_id;

-- Q5. Select and order data retrieved from one table

SELECT \* FROM Flight\_Reservation

ORDER BY L\_Name;

-- Q6. Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 10 rows

SELECT Flight\_Reservation.Seat\_NO, Customer.L\_Name, Customer.F\_Name, Customer.Length\_Stay, Hotel\_Reservation.no\_bedrooms

FROM Flight\_Reservation INNER JOIN Customer ON Flight\_Reservation.Customer\_id = Customer.Customer\_id

INNER JOIN Hotel\_Reservation ON Hotel\_Reservation.L\_Name = Customer.L\_name

FETCH FIRST 10 ROWS ONLY;

-- Q7. Select distinct rows using joins on 3 tables

SELECT DISTINCT \*

FROM Flight\_Reservation INNER JOIN Customer ON Flight\_Reservation.Customer\_id = Customer.Customer\_id

INNER JOIN Hotel\_Reservation ON Hotel\_Reservation.Customer\_ID = Customer.Customer\_ID;

-- Q8. Use GROUP BY and HAVING in a select statement using one or more tables

SELECT Flight\_Reservation.Seat\_no, AVG(ReservationAccount.R\_Cost\_DollarAmount)

FROM Flight\_Reservation INNER JOIN ReservationAccount ON ReservationAccount.Customer\_ID = Flight\_Reservation.Customer\_id

GROUP BY Flight\_Reservation.Seat\_NO, Flight\_Reservation.L\_Name

HAVING Flight\_Reservation.L\_Name = 'Gebremariam';

-- Q9. Use IN clause to select data from one or more tables

SELECT \* FROM Customer

WHERE Customer\_id IN (121, 122, 123);

-- Q10. Select length of one column from one table (use LENGTH function)

SELECT LENGTH(Customer\_ID) FROM Customer;

-- Q11. Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed

SELECT \* FROM ReservationAccount;

DELETE FROM ReservationAccount

WHERE Card\_no = 8;

SELECT \* FROM ReservationAccount;

ROLLBACK;

-- Q12. Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed

SELECT \* FROM Customer;

UPDATE Customer

SET special\_requests = 'Perks'

WHERE F\_name = 'Gebeyew';

SELECT \* FROM Customer;

ROLLBACK;

/\*13. Advanced queries\*/

--Row subquery, max number of bedrooms available for hotel reservation

SELECT \*FROM Hotel\_Reservation

WHERE no\_bedrooms = (SELECT MAX(no\_bedrooms) FROM Hotel\_Reservation);

--selecting/displaying the amount of the second highest no of bedrooms

SELECT MAX(no\_bedrooms) FROM hotel\_reservation WHERE no\_bedrooms < (SELECT MAX(no\_bedrooms) FROM hotel\_reservation);

--return all records from the left table and also from the right table

SELECT \* FROM Hotel\_Reservation hotl

FULL OUTER JOIN flight\_reservation flight

ON hotl.customer\_id = flight.customer\_id;

--return different reservation numbers from the three reservation tables for customers

SELECT

hotel\_reservation.l\_name,

hotel\_reservation.hotel\_reservation\_no,

rentalcar\_reservation.car\_reservation\_no,

flight\_reservation.flight\_reservation\_no

FROM hotel\_reservation

JOIN flight\_reservation

ON hotel\_reservation.customer\_id = flight\_reservation.customer\_id

JOIN rentalcar\_reservation

ON rentalcar\_reservation.customer\_id = flight\_reservation.customer\_id;

--return number of bedrooms tht have atleast 2 beeds and 2 bedrooms

select no\_bedrooms, count(distinct no\_beds)

from hotel\_reservation hot

where exists (

select \* from hotel\_reservation

where no\_beds = hot.no\_beds

and no\_bedrooms =2)

group by no\_bedrooms

--returns the number of bedrooms

SELECT COUNT(reservation\_no), no\_guests

FROM ReservationAccount

GROUP BY no\_guests

HAVING COUNT (reservation\_no) > 2

ORDER BY COUNT(reservation\_no) DESC;

--return customers with similar last name

SELECT A.F\_name AS First\_name1, B.F\_name AS F\_name2, A.L\_name

FROM Customer A, Customer B

WHERE A.Customer\_ID <> B.Customer\_ID

AND A.L\_Name = B.L\_name

ORDER BY A.L\_Name;

--return the minimum number of bedrooms available

SELECT \*FROM Hotel\_Reservation

WHERE no\_bedrooms = (SELECT min(no\_bedrooms) FROM Hotel\_Reservation);

**DDL, DML, and Query Output**

**DDL Output**

Table CUSTOMER dropped.

Table HOTEL\_RESERVATION dropped.

Table FLIGHT\_RESERVATION dropped.

Table RENTALCAR\_RESERVATION dropped.

Table RESERVATIONACCOUNT dropped.

Sequence CUSTOMER\_ID\_SEQ dropped.

Sequence HOTEL\_RESERVATION\_NO\_SEQ dropped.

Sequence FLIGHT\_RESERVATION\_NO\_SEQ dropped.

Sequence CAR\_RESERVATION\_NO\_SEQ dropped.

Sequence RESERVATION\_NO\_SEQ dropped.

View MULTIPLE\_RESERVATION dropped.

View UNPAID\_ACCOUNT dropped.

Table CUSTOMER created.

Table RESERVATIONACCOUNT created.

Table RENTALCAR\_RESERVATION created.

Table FLIGHT\_RESERVATION created.

Table HOTEL\_RESERVATION created.

Index RES\_CUSTOMER\_ID\_FK created.

Index REN\_CUSTOMER\_ID\_FK created.

Index FLI\_CUSTOMER\_ID\_FK created.

Index HOT\_CUSTOMER\_ID\_FK created.

View UNPAID\_ACCOUNT created.

View MULTIPLE\_RESERVATION created.

Sequence CUSTOMER\_ID\_SEQ created.

Sequence RESERVATION\_NO\_SEQ created.

Sequence CAR\_RESERVATION\_NO\_SEQ created.

Sequence FLIGHT\_RESERVATION\_NO\_SEQ created.

Sequence HOTEL\_RESERVATION\_NO\_SEQ created.

Trigger CUSTOMER\_ID\_TRG compiled

Trigger RESERVATION\_NO\_TRG compiled

Trigger CAR\_RESERVATION\_NO\_TRG compiled

Trigger FLIGHT\_RESERVATION\_NO\_TRG compiled

Trigger HOTEL\_RESERVATION\_NO\_TRG compiled

TABLE\_NAME

--------------------------------------------------------------------------------------------------------------------------------

INSTRUCTOR

GRADE

GRADE\_TYPE

GRADE\_CONVERSION

GRADE\_TYPE\_WEIGHT

SECTION

COURSE

ENROLLMENT

RESERVATIONS

RESERVATION

STUDENT

TABLE\_NAME

--------------------------------------------------------------------------------------------------------------------------------

ZIPCODE

CUSTOMER

RESERVATIONACCOUNT

RENTALCAR\_RESERVATION

FLIGHT\_RESERVATION

HOTEL\_RESERVATION

17 rows selected.

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

INSTRUCTOR VALID 28-FEB-23 01-MAR-23

GRADE VALID 28-FEB-23 28-FEB-23

GRADE\_TYPE VALID 28-FEB-23 28-FEB-23

GRADE\_CONVERSION VALID 28-FEB-23 28-FEB-23

GRADE\_TYPE\_WEIGHT VALID 28-FEB-23 28-FEB-23

SECTION VALID 28-FEB-23 28-FEB-23

COURSE VALID 28-FEB-23 28-FEB-23

ENROLLMENT VALID 28-FEB-23 28-FEB-23

RESERVATIONS VALID 20-FEB-23 20-FEB-23

RESERVATION VALID 19-FEB-23 19-FEB-23

SYS\_C007513 VALID 19-FEB-23 19-FEB-23

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

SYS\_C007514 VALID 20-FEB-23 20-FEB-23

STUDENT VALID 28-FEB-23 28-FEB-23

ZIPCODE VALID 28-FEB-23 28-FEB-23

INST\_ZIP\_FK\_I VALID 28-FEB-23 28-FEB-23

GR\_GRTW\_FK\_I VALID 28-FEB-23 28-FEB-23

GRTW\_GRTYP\_FK\_I VALID 28-FEB-23 28-FEB-23

SECT\_CRSE\_FK\_I VALID 28-FEB-23 28-FEB-23

SECT\_INST\_FK\_I VALID 28-FEB-23 28-FEB-23

CRSE\_CRSE\_FK\_I VALID 28-FEB-23 28-FEB-23

ENR\_SECT\_FK\_I VALID 28-FEB-23 28-FEB-23

STU\_ZIP\_FK\_I VALID 28-FEB-23 28-FEB-23

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

STUDENT\_ID\_SEQ VALID 01-MAR-23 01-MAR-23

CAR\_RESERVATION\_NO\_TRG VALID 03-APR-23 03-APR-23

FLIGHT\_RESERVATION\_NO\_TRG VALID 03-APR-23 03-APR-23

HOTEL\_RESERVATION\_NO\_TRG VALID 03-APR-23 03-APR-23

CUSTOMER VALID 03-APR-23 03-APR-23

SYS\_C008026 VALID 03-APR-23 03-APR-23

RESERVATIONACCOUNT VALID 03-APR-23 03-APR-23

SYS\_C008028 VALID 03-APR-23 03-APR-23

RENTALCAR\_RESERVATION VALID 03-APR-23 03-APR-23

CUSTOMER\_ID\_TRG VALID 03-APR-23 03-APR-23

RESERVATION\_NO\_TRG VALID 03-APR-23 03-APR-23

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

SYS\_C008031 VALID 03-APR-23 03-APR-23

FLIGHT\_RESERVATION VALID 03-APR-23 03-APR-23

SYS\_C008034 VALID 03-APR-23 03-APR-23

HOTEL\_RESERVATION VALID 03-APR-23 03-APR-23

SYS\_C008037 VALID 03-APR-23 03-APR-23

RES\_CUSTOMER\_ID\_FK VALID 03-APR-23 03-APR-23

REN\_CUSTOMER\_ID\_FK VALID 03-APR-23 03-APR-23

FLI\_CUSTOMER\_ID\_FK VALID 03-APR-23 03-APR-23

HOT\_CUSTOMER\_ID\_FK VALID 03-APR-23 03-APR-23

CUSTOMER\_ID\_SEQ VALID 03-APR-23 03-APR-23

RESERVATION\_NO\_SEQ VALID 03-APR-23 03-APR-23

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

CAR\_RESERVATION\_NO\_SEQ VALID 03-APR-23 03-APR-23

FLIGHT\_RESERVATION\_NO\_SEQ VALID 03-APR-23 03-APR-23

HOTEL\_RESERVATION\_NO\_SEQ VALID 03-APR-23 03-APR-23

UNPAID\_ACCOUNT VALID 03-APR-23 03-APR-23

MULTIPLE\_RESERVATION VALID 03-APR-23 03-APR-23

49 rows selected.

**DML OUTPUT**

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.

**Query Output**

FLIGHT\_RESERVATION\_NO SEAT\_NO L\_NAME FLIGHT\_TYPE DATE\_FLIG CUSTOMER\_ID DATE\_FLIG

--------------------- ------------------------- ------------------------- ------------------------- --------- ------------------------------ ---------

2000 1 Gebremariam two way 30-JAN-23 100 02-FEB-23

2001 2 Gebremariam two way 30-JAN-23 101 02-FEB-23

2002 3 Habte two way 30-JAN-23 102 02-FEB-23

2003 4 Bekila two way 30-JAN-23 103 02-FEB-23

2004 5 Lemma two way 30-JAN-23 104 02-FEB-23

2005 6 Smith two way 30-JAN-23 105 02-FEB-23

2006 2 Tshige two way 30-JAN-23 106 02-FEB-23

2007 3 Black two way 30-JAN-23 107 02-FEB-23

2008 5 Kebede two way 30-JAN-23 108 02-FEB-23

2009 1 Dante two way 30-JAN-23 109 02-FEB-23

10 rows selected.

FLIGHT\_RESERVATION\_NO SEAT\_NO L\_NAME FLIGHT\_TYPE DATE\_FLIG

--------------------- ------------------------- ------------------------- ------------------------- ---------

2000 1 Gebremariam two way 30-JAN-23

2001 2 Gebremariam two way 30-JAN-23

2002 3 Habte two way 30-JAN-23

2003 4 Bekila two way 30-JAN-23

2004 5 Lemma two way 30-JAN-23

2005 6 Smith two way 30-JAN-23

2006 2 Tshige two way 30-JAN-23

2007 3 Black two way 30-JAN-23

2008 5 Kebede two way 30-JAN-23

2009 1 Dante two way 30-JAN-23

10 rows selected.

NO\_RESERVATION RESERVATION\_NO

------------------------- --------------

2 500

3 502

2 504

3 506

2 508

CUSTOMER\_ID L\_NAME F\_NAME BALANCE SPECIAL\_REQUESTS LENGTH\_STAY FLIGHT\_RESERVATION\_NO SEAT\_NO L\_NAME FLIGHT\_TYPE DATE\_FLIG CUSTOMER\_ID DATE\_FLIG

------------------------------ ------------------------- ------------------------- ------------------------- ------------------------- ----------- --------------------- ------------------------- ------------------------- ------------------------- --------- ------------------------------ ---------

100 Gebremariam Yalem 100 3 2000 1 Gebremariam two way 30-JAN-23 100 02-FEB-23

101 Gebremariam Yewoin 2 2001 2 Gebremariam two way 30-JAN-23 101 02-FEB-23

102 Habte Kiflom 25 1 2002 3 Habte two way 30-JAN-23 102 02-FEB-23

103 Bekila Haile ramp accessible 10 2003 4 Bekila two way 30-JAN-23 103 02-FEB-23

104 Lemma Rahel 5 2004 5 Lemma two way 30-JAN-23 104 02-FEB-23

105 Smith Mike 300 2 2005 6 Smith two way 30-JAN-23 105 02-FEB-23

106 Tshige Gebeyew water view 12 2006 2 Tshige two way 30-JAN-23 106 02-FEB-23

107 Black John 1 2007 3 Black two way 30-JAN-23 107 02-FEB-23

108 Kebede Betty 5 1 2008 5 Kebede two way 30-JAN-23 108 02-FEB-23

109 Dante Kevin 50 balcony 5 2009 1 Dante two way 30-JAN-23 109 02-FEB-23

10 rows selected.

FLIGHT\_RESERVATION\_NO SEAT\_NO L\_NAME FLIGHT\_TYPE DATE\_FLIG CUSTOMER\_ID DATE\_FLIG

--------------------- ------------------------- ------------------------- ------------------------- --------- ------------------------------ ---------

2003 4 Bekila two way 30-JAN-23 103 02-FEB-23

2007 3 Black two way 30-JAN-23 107 02-FEB-23

2009 1 Dante two way 30-JAN-23 109 02-FEB-23

2000 1 Gebremariam two way 30-JAN-23 100 02-FEB-23

2001 2 Gebremariam two way 30-JAN-23 101 02-FEB-23

2002 3 Habte two way 30-JAN-23 102 02-FEB-23

2008 5 Kebede two way 30-JAN-23 108 02-FEB-23

2004 5 Lemma two way 30-JAN-23 104 02-FEB-23

2005 6 Smith two way 30-JAN-23 105 02-FEB-23

2006 2 Tshige two way 30-JAN-23 106 02-FEB-23

10 rows selected.

SEAT\_NO L\_NAME F\_NAME LENGTH\_STAY NO\_BEDROOMS

------------------------- ------------------------- ------------------------- ----------- -----------

1 Gebremariam Yalem 3 2

2 Gebremariam Yewoin 2 2

1 Gebremariam Yalem 3 1

2 Gebremariam Yewoin 2 1

3 Habte Kiflom 1 3

4 Bekila Haile 10 1

5 Lemma Rahel 5 2

6 Smith Mike 2 1

2 Tshige Gebeyew 12 3

3 Black John 1 1

10 rows selected.

FLIGHT\_RESERVATION\_NO SEAT\_NO L\_NAME FLIGHT\_TYPE DATE\_FLIG CUSTOMER\_ID DATE\_FLIG CUSTOMER\_ID L\_NAME F\_NAME BALANCE SPECIAL\_REQUESTS LENGTH\_STAY HOTEL\_RESERVATION\_NO L\_NAME ROOM\_NO NO\_BEDROOMS NO\_BEDS NO\_NIGHTS CUSTOMER\_ID

--------------------- ------------------------- ------------------------- ------------------------- --------- ------------------------------ --------- ------------------------------ ------------------------- ------------------------- ------------------------- ------------------------- ----------- -------------------- ------------------------- ------------------------- ----------- ---------- ---------- ------------------------------

2000 1 Gebremariam two way 30-JAN-23 100 02-FEB-23 100 Gebremariam Yalem 100 3 3000 Gebremariam 144 2 3 3 100

2001 2 Gebremariam two way 30-JAN-23 101 02-FEB-23 101 Gebremariam Yewoin 2 3001 Gebremariam 200 1 1 2 101

2002 3 Habte two way 30-JAN-23 102 02-FEB-23 102 Habte Kiflom 25 1 3002 Habte 100 3 5 1 102

2003 4 Bekila two way 30-JAN-23 103 02-FEB-23 103 Bekila Haile ramp accessible 10 3003 Bekila 12 1 2 10 103

2004 5 Lemma two way 30-JAN-23 104 02-FEB-23 104 Lemma Rahel 5 3004 Lemma 145 2 2 5 104

2005 6 Smith two way 30-JAN-23 105 02-FEB-23 105 Smith Mike 300 2 3005 Smith 36 1 1 2 105

2006 2 Tshige two way 30-JAN-23 106 02-FEB-23 106 Tshige Gebeyew water view 12 3006 Tshige 101 3 3 12 106

2007 3 Black two way 30-JAN-23 107 02-FEB-23 107 Black John 1 3007 Black 29 1 2 1 107

2008 5 Kebede two way 30-JAN-23 108 02-FEB-23 108 Kebede Betty 5 1 3008 Kebede 205 2 2 1 108

2009 1 Dante two way 30-JAN-23 109 02-FEB-23 109 Dante Kevin 50 balcony 5 3009 Dante 206 1 1 5 109

10 rows selected.

SEAT\_NO AVG(RESERVATIONACCOUNT.R\_COST\_DOLLARAMOUNT)

------------------------- -------------------------------------------

1 300

2 100

no rows selected

LENGTH(CUSTOMER\_ID)

-------------------

3

3

3

3

3

3

3

3

3

3

10 rows selected.

RESERVATION\_NO L\_NAME RESERVATON\_TYP NO\_RESERVATION R\_COST\_DOLLARAMOUNT NO\_GU CARD\_NO CUSTOMER\_ID

-------------- ------------------------- ------------------------- ------------------------- -------------------------------------------------- ----- --------------- ------------------------------

500 Gebremariam All 2 300 3 1 100

501 Gebremariam All 1 100 2 2 101

502 Habte All 3 500 1 3 102

503 Bekila All 1 200 1 4 103

504 Lemma All 2 200 2 5 104

505 Smith All 1 100 2 6 105

506 Tshige All 3 300 2 7 106

507 Black All 1 200 1 8 107

508 Kebede All 2 200 1 9 108

509 Dante All 1 100 1 10 109

10 rows selected.

1 row deleted.

RESERVATION\_NO L\_NAME RESERVATON\_TYP NO\_RESERVATION R\_COST\_DOLLARAMOUNT NO\_GU CARD\_NO CUSTOMER\_ID

-------------- ------------------------- ------------------------- ------------------------- -------------------------------------------------- ----- --------------- ------------------------------

500 Gebremariam All 2 300 3 1 100

501 Gebremariam All 1 100 2 2 101

502 Habte All 3 500 1 3 102

503 Bekila All 1 200 1 4 103

504 Lemma All 2 200 2 5 104

505 Smith All 1 100 2 6 105

506 Tshige All 3 300 2 7 106

508 Kebede All 2 200 1 9 108

509 Dante All 1 100 1 10 109

9 rows selected.

Rollback complete.

CUSTOMER\_ID L\_NAME F\_NAME BALANCE SPECIAL\_REQUESTS LENGTH\_STAY

------------------------------ ------------------------- ------------------------- ------------------------- ------------------------- -----------

100 Gebremariam Yalem 100 3

101 Gebremariam Yewoin 2

102 Habte Kiflom 25 1

103 Bekila Haile ramp accessible 10

104 Lemma Rahel 5

105 Smith Mike 300 2

106 Tshige Gebeyew water view 12

107 Black John 1

108 Kebede Betty 5 1

109 Dante Kevin 50 balcony 5

10 rows selected.

1 row updated.

CUSTOMER\_ID L\_NAME F\_NAME BALANCE SPECIAL\_REQUESTS LENGTH\_STAY

------------------------------ ------------------------- ------------------------- ------------------------- ------------------------- -----------

100 Gebremariam Yalem 100 3

101 Gebremariam Yewoin 2

102 Habte Kiflom 25 1

103 Bekila Haile ramp accessible 10

104 Lemma Rahel 5

105 Smith Mike 300 2

106 Tshige Gebeyew Perks 12

107 Black John 1

108 Kebede Betty 5 1

109 Dante Kevin 50 balcony 5

10 rows selected.

Rollback complete.

HOTEL\_RESERVATION\_NO L\_NAME ROOM\_NO NO\_BEDROOMS NO\_BEDS NO\_NIGHTS CUSTOMER\_ID

-------------------- ------------------------- ------------------------- ----------- ---------- ---------- ------------------------------

3002 Habte 100 3 5 1 102

3006 Tshige 101 3 3 12 106

MAX(NO\_BEDROOMS)

----------------

2

HOTEL\_RESERVATION\_NO L\_NAME ROOM\_NO NO\_BEDROOMS NO\_BEDS NO\_NIGHTS CUSTOMER\_ID FLIGHT\_RESERVATION\_NO SEAT\_NO L\_NAME FLIGHT\_TYPE DATE\_FLIG CUSTOMER\_ID DATE\_FLIG

-------------------- ------------------------- ------------------------- ----------- ---------- ---------- ------------------------------ --------------------- ------------------------- ------------------------- ------------------------- --------- ------------------------------ ---------

3000 Gebremariam 144 2 3 3 100 2000 1 Gebremariam two way 30-JAN-23 100 02-FEB-23

3001 Gebremariam 200 1 1 2 101 2001 2 Gebremariam two way 30-JAN-23 101 02-FEB-23

3002 Habte 100 3 5 1 102 2002 3 Habte two way 30-JAN-23 102 02-FEB-23

3003 Bekila 12 1 2 10 103 2003 4 Bekila two way 30-JAN-23 103 02-FEB-23

3004 Lemma 145 2 2 5 104 2004 5 Lemma two way 30-JAN-23 104 02-FEB-23

3005 Smith 36 1 1 2 105 2005 6 Smith two way 30-JAN-23 105 02-FEB-23

3006 Tshige 101 3 3 12 106 2006 2 Tshige two way 30-JAN-23 106 02-FEB-23

3007 Black 29 1 2 1 107 2007 3 Black two way 30-JAN-23 107 02-FEB-23

3008 Kebede 205 2 2 1 108 2008 5 Kebede two way 30-JAN-23 108 02-FEB-23

3009 Dante 206 1 1 5 109 2009 1 Dante two way 30-JAN-23 109 02-FEB-23

10 rows selected.

L\_NAME HOTEL\_RESERVATION\_NO CAR\_RESERVATION\_NO FLIGHT\_RESERVATION\_NO

------------------------- -------------------- ------------------ ---------------------

Gebremariam 3000 1000 2000

Gebremariam 3001 1001 2001

Habte 3002 1002 2002

Bekila 3003 1003 2003

Lemma 3004 1004 2004

Smith 3005 1005 2005

Tshige 3006 1006 2006

Black 3007 1007 2007

Kebede 3008 1008 2008

Dante 3009 1009 2009

10 rows selected.

NO\_BEDROOMS COUNT(DISTINCTNO\_BEDS)

----------- ----------------------

1 1

2 2

3 1

COUNT(RESERVATION\_NO) NO\_GU

--------------------- -----

5 1

4 2

FIRST\_NAME1 F\_NAME2 L\_NAME

------------------------- ------------------------- -------------------------

Yalem Yewoin Gebremariam

Yewoin Yalem Gebremariam

HOTEL\_RESERVATION\_NO L\_NAME ROOM\_NO NO\_BEDROOMS NO\_BEDS NO\_NIGHTS CUSTOMER\_ID

-------------------- ------------------------- ------------------------- ----------- ---------- ---------- ------------------------------

3001 Gebremariam 200 1 1 2 101

3003 Bekila 12 1 2 10 103

3005 Smith 36 1 1 2 105

3007 Black 29 1 2 1 107

3009 Dante 206 1 1 5 109

## Database Administration and Monitoring

### Roles and Responsibilities

The security administrator is an employee in the IT department. Requirements for the security administrator is an IT professional with experience in cyber security. The database administrator is the manager in the travel agency. The requirements for the database administrator is an individual who has experience with making multiple reservation and handling customer information. The system administrator would be the CEO of TEC Inc, and the requirements would be a comprehensive understanding of their company and future directions.

### System Information

UMGC VDA connection with Intel® and Xeon® CPU E5-2676 v3 © 2.40Ghz 2.40Ghz processor, running on the Windows 7 operating system. This 64-bit operating system has 8.GB of ram. Oracle SQL developer will be used to create the database. Oracle SQL Developer Version 4.1.5.21 (Build MAIN-21.78) will be used for database creation and manipulation. The server in this project is assumed to be a Virtual Machine running with a simulated Intel and Xeon CPU E5-2676 v3 @ 2.40Ghz with 8GB ram running Windows 7 on Amazon Web Services (AWS); the actual hardware is unknown.

#### Database Management System Configuration

This section will discuss the hardware management system configuration to be used in the UMGC project, which simulates the enterprise system that is being discussed; discussion of hardware and software that could be used in the enterprise system would be beyond the scope of this document.

#### Database Support Software

Oracle SQL Developer Version 22.2.1 will be used for database creation and manipulation on a simulated Windows 11 operating system.

#### Security and Privacy

Data partitioning is used to keep the integrity of the database system, as well as patient privacy. Patient samples have unique numbers that can reference the patient data with, but the laboratory employees do not have access to find the patient information associated with the sample number. Furthermore, certain schema are not accessible by all users. For example on HR employees can access and modify data in the employee table.

### Performance Monitoring and Database Efficiency

Travel agency and IT employees will share in the responsibility for monitoring performance of the database. IT is in charge of space utilization, system resource consumption, whereas the travel agent is responsible for the query performance metrics. This data can be pulled by the associated GUI.

#### Operational Implications

It is the responsibility of the IT department to verify that the new Windows update will not compromise the booking reservation database. Additionally, they are also responsible for making sure the firewall does not block incoming customer information.

#### Data Transfer Requirements

The database must run on a Windows 11 or higher operating system.

#### Data Formats

Data that will be received by the DBMS will be in .csv or .txt format so the application sever can parse and input the data. Data that is sent of the DBMS system will be in a .pdf format or equivalent.

### Backup and Recovery

The DBMS system will be backed up nightly, due to the high sensitivity of customer information. This will be prompted automatically through the application server.

**References**

Boyle, J., Rovira, H., Cavnor, C. C., Burdick, D. M., Killcoyne, S., & Shmulevich, I. (2009a). Adaptable data management for systems biology investigations. BMC Bioinformatics, 10(1). https://doi.org/10.1186/1471-2105-10-79

Gunjal, B. (2003, December 1). Database System: Concepts and Design. ResearchGate. https://www.researchgate.net/publication/257298522\_Database\_System\_Concepts\_and\_Design

Metnitz, P. G. H., & Lenz, K. (1995). Patient data management systems in intensive care — the situation in Europe. Intensive Care Medicine, 21(9), 703–715. https://doi.org/10.1007/bf01704737

Pangalos, G. (1988). Design of hospital database systems in a non-relational environment. Journal of Medical Informatics, 13(1), 27–33. https://doi.org/10.3109/14639238809003572

Rodríguez-Terol, A., Caraballo, M., Palma, D. A., Santos-Ramos, B., Molina, T. J., Desongles, T., & Aguilar, A. (2009a). Quality of interaction database management systems. Farmacia Hospitalaria, 33(3), 134–146. https://doi.org/10.1016/s2173-5085(09)70079-6

**Appendix A: ERD**



Diagram

Description automatically generated

Figure 1: ERD for the Sample Tracking Database

**Appendix B: PDM**

**Diagram

Description automatically generated**

**Appendix C: Acronyms**

**Table 2 - Acronyms**

|  |  |
| --- | --- |
| **Acronym** | **Literal Translation** |
| TEC | Trip Executive company |
| DBMS | Database management system |
| GUI | Graphical User Interface |