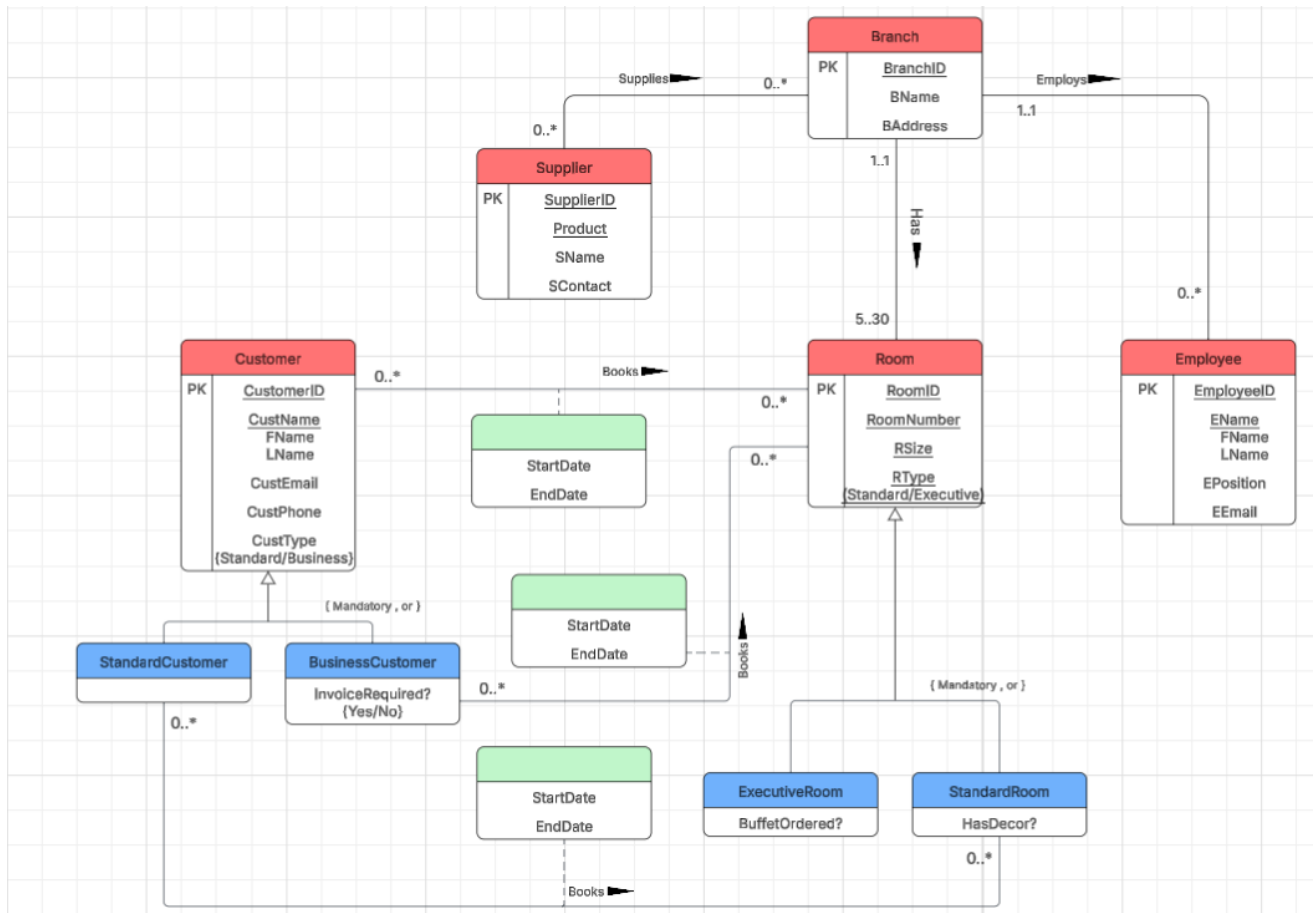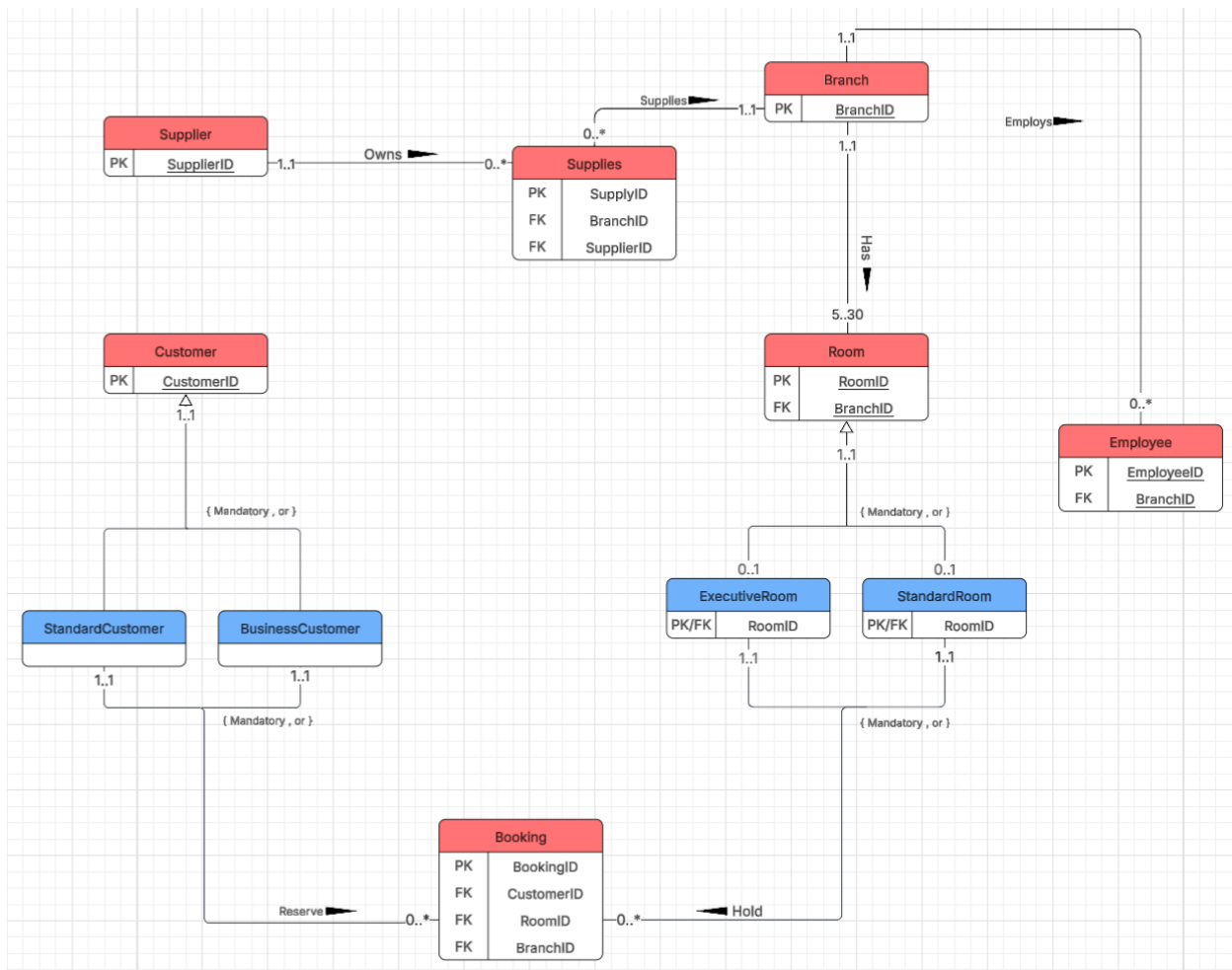# Conceptual Entity Relationship Diagram

Logical Entity Relationship Diagram

EASE Ltd Database System Report

1. Introduction

The purpose of this report is to outline the planning, design and integration of a database driven website developed for a fictitious hospitality management company called EASE Ltd. The main purpose of this project was to make an efficient, well put together relational database system through the utilization of the Oracle Database Management System, capable of supporting the main operations of the business. Aside from the backend database, I also designed a fully functional, user-directed web page developed using PHP. Using PHP and the Oracle Call Interface (OCI) I was able to create a seamless interaction between the stored data and the front end webpage.

My web application was made to support key operations at the company by managing and collecting vital data such as branches, employees, customers, rooms, bookings, supplies, suppliers and their respective attributes. I aimed to provide dynamic functionality, enabling users to view booking records, explore room types and features (such as Decor and buffet options), view employees and their roles and track supply sources. Overall, the aim is to simulate a realistic service providers database system while demonstrating a practical use of relational database design, SQL and web development techniques.

2. Database Design

My database schema was made to reflect the operational structure and data requirements of EASE Ltd as accurately as possible. My schema contains several related tables, all serving a specific function for the business.

- **Branch**: The Branch table holds information about each branch operated by the company, including a unique primary key BranchID, the name of the branch (BName), and the address of the branch (BAddress). Every branch therefore serves as a central reference point for each room, employee, booking and supply record associated with it.
- **Room, StandardRoom, ExecutiveRoom**: The data for each room is organized well. The Room tables hold general information for the rooms such as room number, room size (RSize), and the type of room (RType). Room has two subtype tables StandardRoom and ExecutiveRoom, which handle attributes specific to those categories. Standard room has a HasDecor attribute, and the Executive room table holds the BuffetOrdered attribute. This separation between room, ensures the safe addition of future rooms while at the same time avoiding overcomplication on the main room table.
- **Employee**: This table holds information on employees such as their names, emails and job positions. All employees are linked to a specific branch through the foreign key BranchID, making branch specific staff management possible.
- **Customer**: The customer table accounts for both types of clients, basic and business, through the CustType attribute. The other fields CustFName, CustLName and CustEmail hold vital information for communicating with customers. I decided to handle the different customer types within one table to simplify booking logic while at the same time preserving flexibility
- **Booking**: The Booking table holds the most foreign keys as it is the entity tying it all together. These include customers, rooms and branch. It also includes StartDate, EndDate and InvoiceRequired, providing vital operational and financial information. This table allows complex queries such as calculating stay durations of customers and providing invoices for business customers.

- **Supplier and Supplies**: Supplier data is stored within the Supplier table, these being contact details and product types. The Supplies table works as a junction between the supplier and branch by recording supply deliveries (SupplyDetails, SupplyDate) to specific branches. The separation between these two tables allows multiple branches to obtain supplies from the same supplier while upholding a simple detailed transaction history.

  All my table definitions and constraints, including primary keys, foreign keys, and data types, are within the submitted DBschema.sql file.

3. Data Insertion

I inserted my test data using Insertion.sql. My script populates the tables with a number of records to allow proper testing. For example, different room types were added, bookings with and without an invoice were included and a number of suppliers were linked to branches.

4. Web Interface Implementation

I created several PHP pages that interact with the database:

- **Index.php** : Home page with a navigation bar and a company welcome.
- **URL:** *https://teach.scam.keele.ac.uk/prin/y1o35/DBcoursework23045944/index.php*
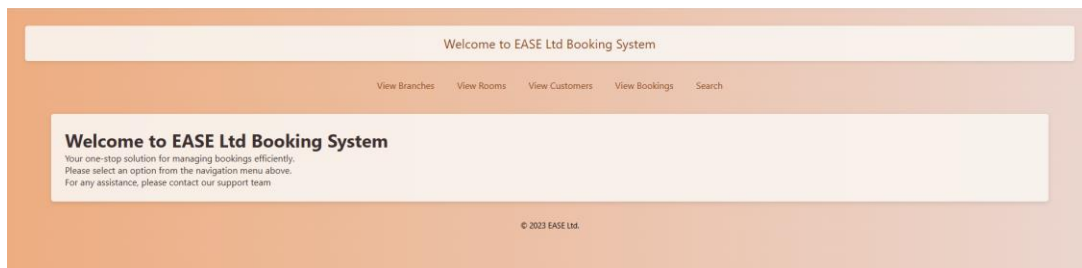
- **View_branches.php, view_rooms.php, view_customers.php, view_bookings.php** :Pages displaying relevant information in a HTML table.

- **Search,php**: Lets the user select any branch they like and be able to view all the information pertaining to said branch. I made sure it used dynamic SQL queries and data formatting for better clarity.

All pages made used oci_connect to communicate with Oracle and included error handling and the cleaning of data using functions such as htmlspecialchars().

5. Features and Functionality

- Throughout my pages I included joins on some tables to display useful data that was not originally in the table selected.
- I made use of SQL CASE statements in booking and room queries in order to provide user friendly labels such as Has Decor/Does not have decor.
- I did my best to format data for improved readability, such as date conversions and stay duration calculations.
- Functions like display_table were included to improve the reusability of code and consistency throughout.

6. Challenges and Resolutions

- **Room Features**: Differentiating room types required some conditional subqueries as figuring out how to implement by ERD logic was hard.
- **Supplies Relationship**: I previously had a BranchSupplier table as a junction between supplier and branch, however I figured a Supplies table would be more fitting. Renaming from BranchSupplier to Supplies required query adjustments and alias renaming which took my time.
- **User Interface**: Making sure all data I put in was readable and formatted involved several iterations.

7. Conclusion

I believe my project perfectly shows the design and integration of a relational database for the company EASE Ltd, along with a dynamic PHP based website. My work closely meets the objectives of the assignment and supports key business functions such as bookings, customer management and supply tracking through the proper implementation of the database.