**CSI2132 Database I**

**eHotel-Course Project:**

**Entity Relationship Model Related Design**

**Deliverable 1 Report**

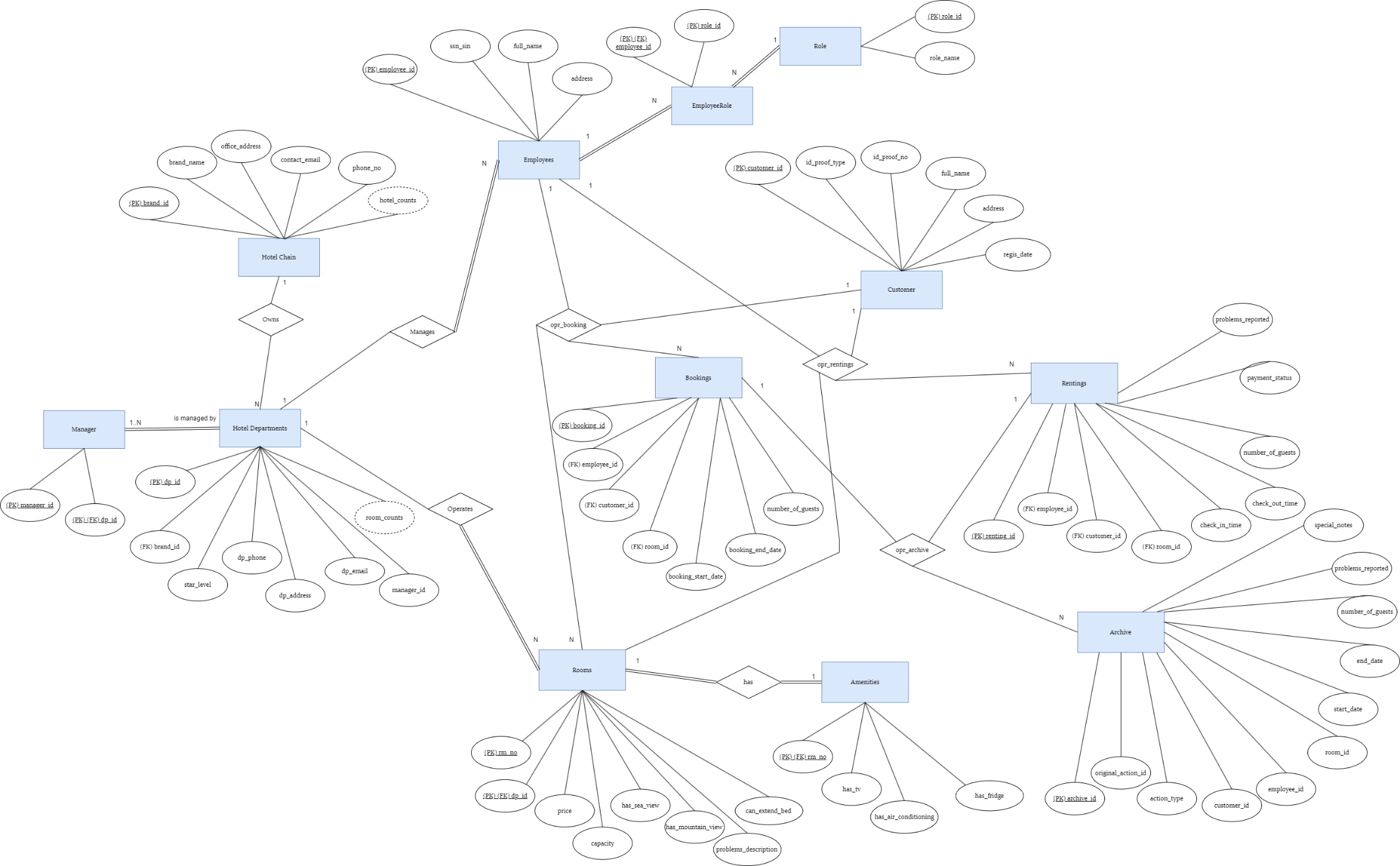
Roy, Rui 300176548

Yucheng, Chen 300194614

University of Ottawa

Faculty of Engineering

February 16, 2024

1. **ER Diagram**

Justification for the ER Diagram:

The ER diagram is designed to comprehensively represent the data structure of a hotel management system. Key entities such as *Hotel Chain, Hotel, Room, Booking, Renting, Customer, Employee,* and *Amenities* are included to reflect all aspects of hotel operations.

**Entities**: Each entity represents a distinct aspect of the hotel's operations, ensuring that data is organized logically and efficiently. Attributes are selected to capture essential details required for the system's functionality.

For some special condition, like “Every hotel needs to have a manager,” Managers entity is specifically designed to satisfy the business requirement that every hotel must have a designated manager. The Managers entity enforces data integrity by using a foreign key that references the primary key of the Employees entity. This ensures that every manager is an existing employee.

**Relationships:** Defined relationships and cardinalities between entities reflect real-world interactions and business rules, such as customers making bookings and employees managing rentals.

**Archiving:** The inclusion of an Archive entity allows for historical data preservation, aligning with the business need to maintain records even after associated entities are deleted.

1. **Relational Database Schema**

-- Hotel Chain

CREATE TABLE **HotelChain** (

brand\_id INT PRIMARY KEY,

brand\_name VARCHAR(255),

office\_address VARCHAR(255),

contact\_email VARCHAR(255),

phone\_no VARCHAR(20)

hotel\_counts INT

);

-- Hotel

CREATE TABLE **HotelDepartments** (

dp\_id INT PRIMARY KEY,

brand\_id INT,

star\_level INT,

dp\_phone VARCHAR(20),

dp\_address VARCHAR(255),

dp\_email VARCHAR(255),

manager\_id INT,

room\_counts INT,

FOREIGN KEY (brand\_id) REFERENCES HotelChain(brand\_id),

FOREIGN KEY (manager\_id) REFERENCES Manager(employee\_id)

);

-- Employees

CREATE TABLE **Employees** (

employee\_id INT PRIMARY KEY,

ssn\_sin VARCHAR(20),

full\_name VARCHAR(255),

address VARCHAR(255),

FOREIGN KEY (role\_id) REFERENCES Role(role\_id)

);

-- Role

CREATE TABLE **Role** (

role\_id INT PRIMARY KEY,

role\_name VARCHAR(255)

);

-- EmployeeRole (Junction Table for many-to-many relationship between Employees and Roles)

CREATE TABLE **EmployeeRole** (

employee\_id INT,

role\_id INT,

PRIMARY KEY (employee\_id),

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id),

FOREIGN KEY (role\_id) REFERENCES Role(role\_id)

);

-- Customer

CREATE TABLE **Customer** (

customer\_id INT PRIMARY KEY,

full\_name VARCHAR(255),

address VARCHAR(255),

id\_proof\_type VARCHAR(50),

id\_proof\_no VARCHAR(50),

regis\_date DATE

);

-- Rooms

CREATE TABLE **Rooms** (

room\_no INT PRIMARY KEY,

dp\_id INT,

price DECIMAL(10, 2),

capacity INT,

has\_sea\_view BOOLEAN,

has\_mountain\_view BOOLEAN,

can\_extend\_bed BOOLEAN,

problems\_description TEXT,

FOREIGN KEY (dp\_id) REFERENCES Hotel(dp\_id)

);

-- Rooms

CREATE TABLE **Amenities** (

rm\_no INT PRIMARY KEY,

has\_tv BOOLEAN,

has\_air\_conditioning BOOLEAN,

has\_fridge BOOLEAN,

FOREIGN KEY (rm\_no) REFERENCES Rooms(rm\_no)

);

-- Bookings

CREATE TABLE **Booking** (

booking\_id INT PRIMARY KEY,

customer\_id INT,

room\_id INT,

employee\_id INT,

number\_of\_guests INT,

booking\_start\_date DATE,

booking\_end\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (room\_id) REFERENCES Rooms(rm\_id),

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id)

);

-- Rentings

CREATE TABLE **Renting** (

renting\_id INT PRIMARY KEY,

customer\_id INT,

room\_id INT,

employee\_id INT,

check\_in\_date DATE,

check\_out\_date DATE,

number\_of\_guests INT,

payment\_status VARCHAR(100),

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (room\_id) REFERENCES Rooms(room\_id),

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id)

);

-- Archive (Could be for either Bookings or Rentings)

CREATE TABLE **Archive** (

archive\_id INT PRIMARY KEY,

original\_action\_id INT,

action\_type VARCHAR(50), -- 'Booking' or 'Renting'

customer\_id INT,

room\_id INT,

employee\_id INT,

start\_date DATE,

end\_date DATE,

number\_of\_guests INT,

problem\_reported VARCHAR(255),

special\_notes VARCHAR(255),

archive\_date DATE

-- No foreign keys to allow for historical preservation even if original records are deleted

);

1. **Integrity Constraints**
2. **Hotel Chain Constraints:**

*hotel\_chain\_id* (PK): Ensures uniqueness for each hotel chain.

*office\_address*, *contact\_email*, *phone\_no*: Not Null constraints to ensure every hotel chain has contact information.

*hotel\_counts*: A Check constraint to ensure the value is a non-negative integer.

1. **Hotel Departments Constraints:**

*dp\_id* (PK): Uniquely identifies each hotel.

*brand\_id* (FK): Ensures each hotel is associated with a hotel chain.

*star\_level*: A Check constraint to ensure it falls within the allowed range (e.g., 1 to 5).

*room\_counts*, *address*, *hotel\_email*, *hotel\_phone*: Not Null constraints for essential hotel details.

1. **Room Constraints:**

*rm\_id* (PK): Uniquely identifies each room within a hotel.

*hotel\_id* (PK) (FK): Ensures each room is associated with a hotel.

*price*: A Check constraint to ensure the price is a positive value.

*capacity*: Not Null to guarantee every room's capacity is recorded.

*has\_sea\_view*, *has\_mountain\_view*, *can\_extend\_bed*: Boolean fields, defaulting to false.

*problems\_description*: Allows Null to handle rooms without problems.

1. **Customer Constraints:**

*customer\_id* (PK): Uniquely identifies each customer.

*full\_name*, *address*, *id\_type*, id\_number, regis\_date: Not Null constraints to ensure all required customer information is captured.

1. **Employee Constraints:**

*employee\_id* (PK): Uniquely identifies each employee.

*full\_name*, *address*, *ssn\_sin*: Not Null constraints to ensure all required employee information is captured.

*role*: Not Null to guarantee that each employee's role is defined.

1. **Manager Constraints:**

Each hotel must have one and only one manager. This can be enforced by a Unique constraint on a manager\_id field in the Hotel table.

*manager\_id* (FK): References an employee\_id from the Employees table.

1. **Booking and Renting Constraints:**

*booking\_id*, *renting\_id* (PK): Ensure uniqueness for bookings and rentings.

*customer\_id*, *room\_id*, *employee\_id* (FK): Link bookings and rentings to customers, rooms, and employees.

*start\_date*, *end\_date*: Not Null to capture the duration of bookings and rentings.

Archiving process should ensure foreign key references are not violated if related records are deleted.