Hotel Management System Project for CS202 2024-2025 FALL Semester

Members: Berra Eğcin, S040865 | Nezife Havin Yılmaz, S040171

(GROUP 9)

This document implies all the MySQL features of CS202 2024-2025 Fall semester project description and additional elements to personalize the project for group 9 (Berra, Havin). Note that it's the first part of our project. Our project includes DDL, DML codes for database application and ER Diagram for visual representation of how we manage entities, attributes and relationships. ***ER Diagram is on the last page of the document.***

Design Descriptions:

- To begin with the initial application of the project, we have to be able to create different hotels, so we started with adding a "Hotel" entity. For the attributes; the "Hotel" entity has "hotel_ID" as primary key, "hotel_name", "address" as composite attribute (includes "city", "district" and "street" attributes).
- Different "Hotels" should have different database managers, so we added an "Administrator" entity. For the attributes; the "Administrator" entity has "admin_ID" as the primary key. All the hotels have only 1 administrator.
- As the project description says, we needed to add "Receptionist" and "Housekeeping" to run the hotel. Since both of them work for the same hotel, we created an ISA hierarchy between "Employees" and roles, then connected them to the "Administrator". "Employees" entity carries the shared attributes for "Receptionist" and "Housekeeping". For the attributes; "employee_ID" as primary key, "role" for describing the title of the work they do, "e_name" as composite attribute (includes "e_first_name" and "e_last_name").
- Housekeeping needs to have plans of their own timetable, so we created a "Schedule" entity. For the attributes; "schedule_ID" as primary key, "status" for changing the conditions of the housekeeper, "cleaning_date" to provide information about when the rooms were cleaned.
- We need to have customers to run the hotel, so we added a "Guest" entity. For the attributes; "guest_ID" as primary key, "birth_date","b_points" for calculating how much points do a customer have, "age" as derived attribute, "phone_number", "g_name" as composite attribute (includes "g_first_name" and "g_last_name").
- Since customers can come as several people, we created "Invitee" as a weak entity. For the attributes; "guest_ID" and "invitee_ID" as composite primary key, "i_name" as composite attribute (includes "i_first_name" and "i_last_name").
- Customers need to make a reservation for their stay, so we created a "Booking" entity. For the attributes; "booking_ID" as primary key, "b_status" to check if it's approved from Receptionist, "num_of_guest" to write the total number of guests, "check_in", "check_out".

- Customers need to make reservations for their rooms, so we created an “Room” entity. For the attributes; “room_ID” as primary key, “isClean” for checking if a room is clean or not, “isAvailable” for scanning if a room is available or not.
- Customers should choose which type of room they want to stay in, so we created a “RoomType” entity. For the attributes; “type_ID” as primary key, “price” to get the price of the room, “bed_type” to know which bed type can be suitable for customers , “type_name” to get room’s type, “max_occupancy”, “bed_count” to get how many bed’s do this room have, “description” to gather all the information as string.
- Customers can stay in the hotel by way of paying money, so we created a “Payment” entity. For the attributes; “pay_ID” for primary key, “pay_method” to know if a customer paid with cash or credit card, “pay_time” to know when a customer made payment, “amount” to find the total checkout.
- For additional feature, we wanted to make something special called “Loyalty Program” for our customers. Depending on how many times a customer stayed at the hotel, the system will give a discount for which ranking that the customer is in. That is why we created a “Loyalty” entity. For the attributes; “loyalty_id” as primary key, “loyalty_rank” to find which customer is in as loyalty level. Separately, “discount_percentage”, “min_points” and “max_points” as derived attributes from “Guest” entity. Loyalty rankings are (from lowest to highest) “bronze”, “silver” and “gold”. We created their limits to check and see which customer made more reservations. According to their rankings, the discount percentages are (from lowest to highest again) %5, %15 and %25.

Manipulating the Data:

1. **Created Triggers:**
   - Any value that gets price or payment as negative should return an error.
   - Any ID on their own Entity shouldn’t be null or duplicate itself.
   - Check out date should be bigger than the check in date.
   - Since only 18 or older guests can make reservation, we need to calculate guest’s age with the current year – the guest’s birth year.
2. **Created Procedures:**

i. **for Administrator to:**

   i. check the room status
   ii. add room
   iii. modify room
   iv. handle any unpaid booking
   v. view user accounts
   vi. generate hotel’s revenue report
   vii. view all the bookings

viii. view housekeeping records

ix. view popular room types

x. view the hotel's employees

ii. for Receptionist to:

i. view pending bookings from guests

ii. confirm booking for guests

iii. view available rooms

iv. process the payment

v. assign housekeeping tasks to housekeepers

vi. edit schedule after checkout

vii. view housekeepers availability

viii. modify the bookings

iii. for Housekeeping to:

i. view rooms

ii. view pending tasks

iii. view completed tasks

iv. mark housekeeping tasks (as completed)

v. view their schedule

vi. view their today's tasks

iv. for Guests to:

i. view available rooms

ii. add booking

iii. view their bookings

iv. cancel their booking

v. check their payment status

3. <u>Inserted Statements:</u>

- We started with adding base data to our database. According to tables that we created before, we wrote initial values to check if the tables are correctly getting string, date, integer or NULL values.

- We determined that administrator's are created directly and stated that it is not duplicating itself in ER Diagram (with cardinality constraint). The administrator doesn't have any hierarchy on top of itself so it cannot create any other administrator.

Relationships as sentences:

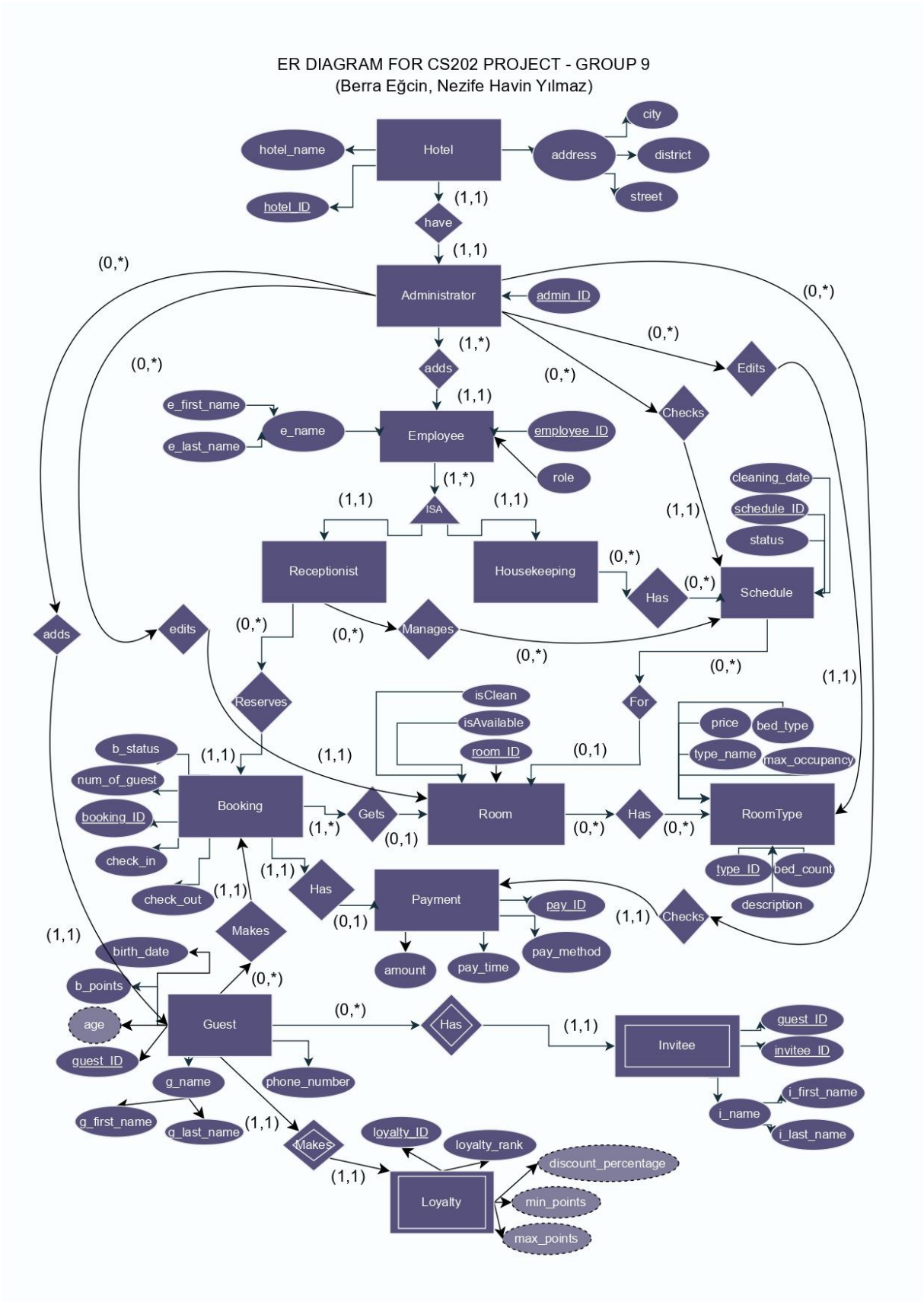
- Hotel **have** Administrator.
- Administrator **adds** Employee.
- Administrator **checks** Schedule.
- Administrator **edits** Room.
- Administrator **edits** RoomType.
- Administrator **checks** Payment.
- Administrator **adds** Guest.
- Employee **(has ISA connection to)** Receptionist and Housekeeping.
- Housekeeping **has** Schedule.
- Schedule **for** Room.
- Receptionist **reserves** Booking.
- Receptionist **manages** Schedule.
- Booking **gets** Room.
- Booking **has** Payment.
- Room **has** RoomType.
- Guest **makes** Booking.
- Guest **has** Invitee.
- Guest **makes** Loyalty.

Functional Dependencies:

- Hotel:
    - Attributes: hotel_ID(PK), admin_ID(FK) hotel_name, address = (city, district, street)
        - hotel table: hotel_ID -> hotel_name
        - address table: hotel_ID -> street, city, district
- Guest:
    - Attributes: guest_ID(PK), birth_date, b_points, loyalty_id(FK),age(derived),phone_number, g_name = (g_first_name, g_last_name)
        - guest table:
            - guest_ID -> birth_date, b_points, loyalty_ID, age, phone_number

        - birth_date -> age (calculated using the trigger)
    - guest_name table: guest_ID -> g_first_name, g_last_name
- Invitee:
    - Attributes: (guest_ID, invitee_ID)(PK), i_name = (i_first_name, i_last_name)
        - invitee table: invitee_ID, guest_ID -> invitee_ID, guest_ID (due to composite primary key)
        - inviteeName table: invitee_ID -> i_first_name, i_last_name
- Loyalty:
    - Attributes: loyalty_ID(PK), loyalty_rank, min_points, max_points, discount_percentage
        - loyalty table:
            - loyalty_ID -> loyalty_rank, min_points, max_points, discount_persentage
            - loyalty_rank -> min_points, max_points, discount_persentage
- Booking:
    - Attributes: booking_ID(PK), guest_ID(FK),room_ID(FK), num_of_guest, check_in, check_out, b_status
        - booking table:
            - booking_ID -> guest_ID, room_ID, num_Of_Guests, check_in, check_out, b_status
            - guest_ID -> guest_name
- Payment:
    - Attributes: pay_ID(PK), booking_ID(FK), pay_method, pay_time
        - payment table: pay_ID -> booking_ID, pay_method, pay_time
- Employee:
    - Attributes: employee_ID(PK), role, e_name = (e_first_name, e_last_name)
        - employee table: employee_ID -> role
        - employee_name table: employee_ID -> e_first_name, e_last_name
- Housekeeping:
    - Attributes: employee_ID(PK)(FK)
        - housekeeping table: employee_ID -> employee_ID
- Schedule:

    - Attributes: schedule_ID(FK), employee_ID(PK)(FK), room_ID(FK), cleaning_date, status
        - schedule table: schedule_ID -> employee_ID, room_ID, cleaning_date, status
- Administrator:
    - Attributes: admin_ID(PK)
- Room:
    - Attributes: room_ID(PK), type_ID(FK), isClean, isAvailable
        - room table: room_ID -> type_ID, isClean, isAvailable
- RoomType:
    - Attributes: type_ID(PK), room_ID(FK), type_name,max_occupancy, price, bed_count,bed_type,description
        - room_type table: type_ID -> room_ID, type_name, max_occupancy, price, bed_count, bed_type, description


ER DIAGRAM FOR CS202 PROJECT - GROUP 9
(Berra Eğcin, Nezife Havin Yılmaz)
Hotel
hotel_name
hotel_ID
address
city
district
street
have
(1,1)
Administrator
admin_ID
adds
Edits
Checks
Employee
e_first_name
e_last_name
e_name
employee_ID
role
ISA
Receptionist
Housekeeping
Has
Schedule
cleaning_date
schedule_ID
status
Manages
edits
Reserves
For
isClean
isAvailable
room_ID
Booking
b_status
num_of_guest
booking_ID
check_in
check_out
Gets
Room
RoomType
price
bed_type
type_name
max_occupancy
type_ID
bed_count
description
Payment
pay_ID
amount
pay_time
pay_method
Makes
Guest
birth_date
b_points
age
guest_ID
g_name
g_first_name
g_last_name
phone_number
Invitee
invitee_ID
i_name
i_first_name
i_last_name
Loyalty
loyalty_ID
loyalty_rank
discount_percentage
min_points
max_points