

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

domingo, 17 de março de 2019

Hotel BDAD

Bases de Dados
2018/2019



2MIEIC03_04:

Gonalo Marantes
Jos  Miguel Sim es
V tor Hugo Gonalves

up201706917@fe.up.pt
up201704317@fe.up.pt
up201703917@fe.up.pt

Index

Introduction.....2

Project Specification.....3

Conceptual Model.....5

Reviewed Conceptual Model.....6

Relational Model.....7

Functional Dependencies and Normal Form Analysis.....9

 Functional Dependencies.....9

 Normal Form Analysis.....10

Restrictions.....11

Introduction

For the development of our project of the curricular unit of "*Bases de Dados*", we had the opportunity to choose an appropriate theme to create and analyze a database. This way, we decided to choose a defiant theme, but also useful and present in our day-to-day life. Our choice was about a database to manage a hotel's website, focusing on booking reservations, as their related information.

In the database, we can access and manipulate information relative to clients, reservations, rooms, user's avaliations, and others. We tried to group and organize the data at the most simple and appealing way, as is possible to see in the next page of this report.

Project Specification

Client

Any Hotel client in the hotel's website's database is a **Client**. Each client has a unique ID, which is the class's primary key. We also have access to other information, such as, name, email, password, tax number (for payment informations) and photo. A client can reserve multiple **rooms** through **reservations**.

Reservation

All **reservations** also have an unique id, a startDate and endDate, as well as an associated finalPrice, and a isPaid indicator in case the **Client** decides to pay upfront. This finalPrice attribute can be calculated by multiplying the number of rooms by the number of nights spent (endDate - startDate) and finally adding any extraCost.

ReservationType

This module specifies what type of reservation the client intends, and depending on the type, there is an associated extraCost. The type can vary from all included, full board, half board and only accommodation.

Guest

Each Reservation also includes **guests**, which represent the people that are staying in the hotel. Guests only have name as their attribute. The Client itself, who made the reservation, might not be part of this guest list.

Cancelling

If a client decides to cancel a reservation he had previously booked, he can do it on a certain date.

Room

Regarding the **Rooms** class, each **Room** has a number, a certain capacity and a price (per night). The **Room** class is a generalization of **MeetingRoom** and **Bedroom**, each with its own associated class that specifies its type.

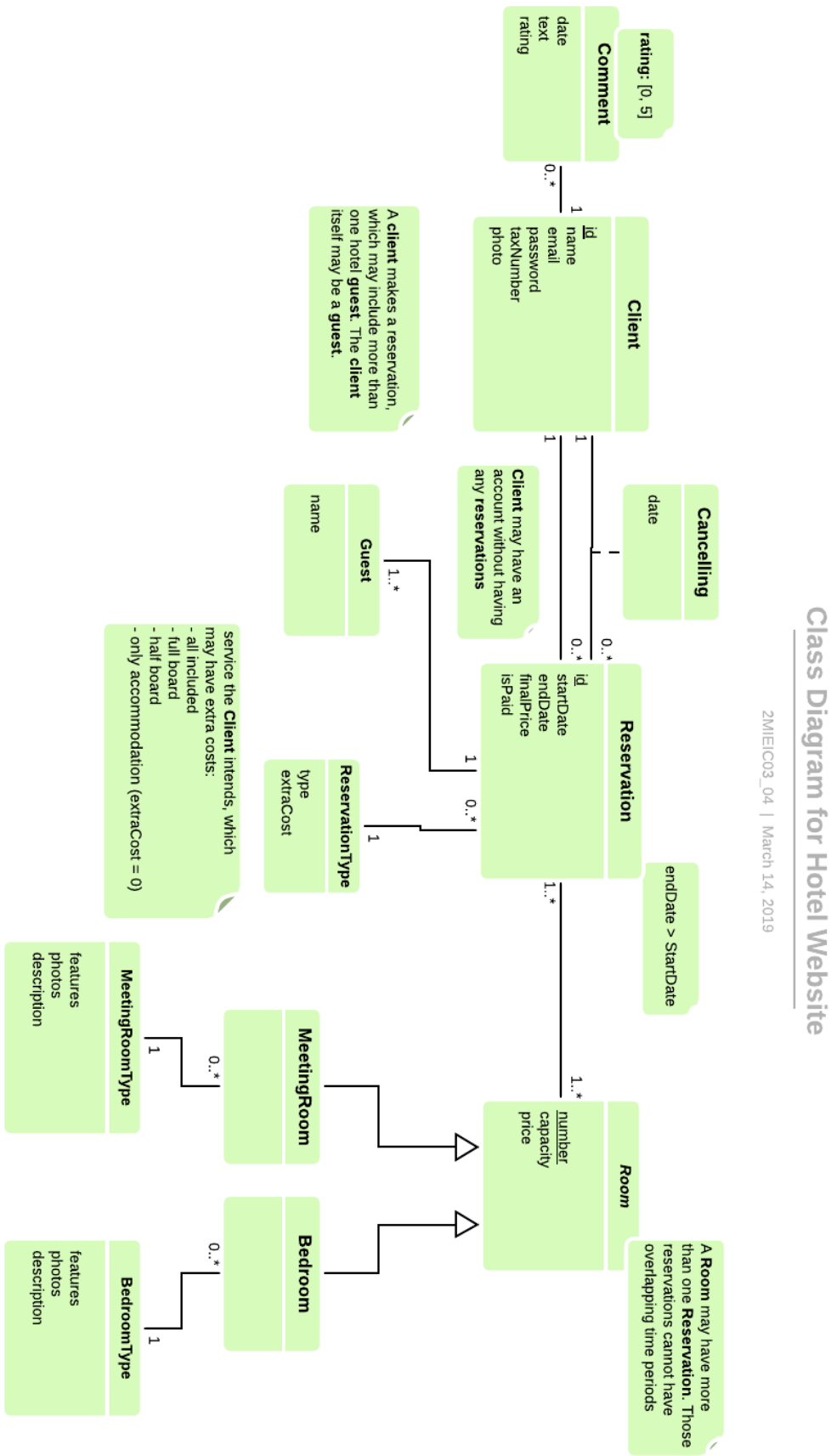
MeetingRoomType and BedroomType

These modules specify the meeting room and bedroom type accordingly. The client has access to the room's features, photos and a small description.

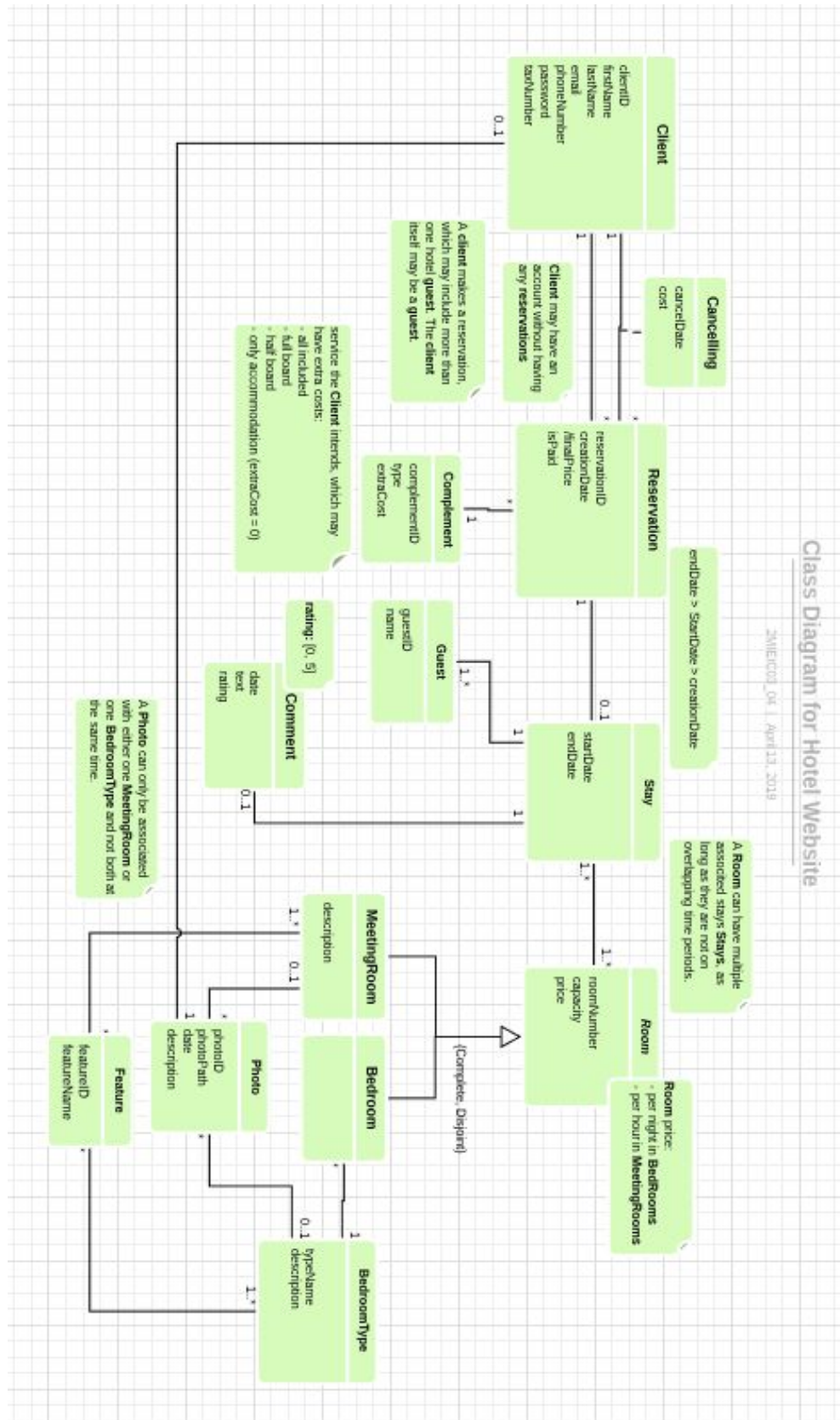
Comment

After the client leaves the hotel, he can evaluate those days he passed there. In a rating between 0 and 5, and a small text (with a date), the client could explain to others how good was his stay in the hotel, to help people decide if they will make a reservation there in the future, and help the managers to improve the hotel's quality.

Conceptual Model



Reviewed Conceptual Model



Relational Model

Client (clientID, firstName, lastName, email, phoneNumber, password, taxNumber)

clientID → firstName, lastName, email, phoneNumber, password, taxNumber

clientID is the *primary key*

Reservation (reservationID, creationDate, finalPrice, isPaid, client→Client, complement→Complement)

reservationID → creationDate, finalPrice, isPaid, client, complement

reservationID is the *primary key*

client and complement are *foreign keys*

isPaid is a *derived attribute*

Cancelling (reservationID→Reservation, client→Client, cancelDate, cost)

reservationID → client, cancelDate, cost

reservationID is the *primary key*

reservation and client are *foreign keys*

Complement (complementID, type, extraCost)

complementID → type, extraCost

complementID is both a *primary key* and a *foreign key*

Stay (reservationID→Reservation, startDate, endDate)

reservationID → startDate, endDate

reservationID is both a *primary key* and a *foreign key*

Guest (guestID, reservationID→Reservation, firstName, lastName)

guestID, reservationID → firstName, lastName

guestID and reservationID are the composite *primary key*

reservationID is a *foreign key*

Comment (reservationID→Reservation, date, text, rating)

reservationID → date, text, rating

reservation is both a *primary key* and a *foreign key*

Room (roomNumber, capacity, price)

roomNumber → capacity, price

roomNumber is the *primary key*

RoomStay (stay→Stay, room→Room)

stay and room are both *foreign keys* and the *composite primary key*

MeetingRoom (roomNumber→Room, description)

roomNumber → description

roomNumber is the *primary key* and a *foreign key*

Bedroom (roomNumber→Room, bedroomType→BedroomType)

roomNumber → bedroomType

roomNumber is the *primary key*

roomNumber and bedroomType are *foreign keys*

BedroomType (typeName, description)

typeName →description

type is the *primary key*

Photo (photoID, meetingRoom→MeetingRoom, bedroom→Bedroom, clientID→Client, photoPath, date, description)

photoID → meetingRoom, bedroom, clientID, photoPath, date, description

photoID is the *primary key*

meetingRoom, bedroom and clientID are *foreign keys*

Feature (featureID, featureName)

featureID →featureName

featureID is the *primary key*

MeetingRoomFeature (roomNumber→MeetingRoom, featureID→Feature)

roomNumber and featureID are the *composite primary key* and *foreign keys*

BedroomTypeFeature (typeName→BedroomType, featureID→Feature)

typeName and featureID are the *composite primary key* and *foreign keys*

Functional Dependencies and Normal Form Analysis

An attribute or set of attributes are a key for a relation if their closure is all the attributes in the relation.

Functional Dependencies

Client:
$$\{\text{clientID}\}^+ = \{\text{clientID}, \text{firstName}, \text{lastName}, \text{email}, \text{phoneNumber}, \text{password}, \text{taxNumber}\}$$
Reservation:
$$\{\text{reservationID}\}^+ = \{\text{reservationID}, \text{reservationDate}, \text{creationDate}, \text{finalPrice}, \text{isPaid}, \text{client}, \text{complement}\}$$
Cancelling:
$$\{\text{reservationID}\}^+ = \{\text{reservationID}, \text{client}, \text{cancelDate}, \text{cost}\}$$
Complement:
$$\{\text{complementID}\}^+ = \{\text{complementID}, \text{type}, \text{extraCost}\}$$
Stay:
$$\{\text{reservationID}\}^+ = \{\text{reservationID}, \text{startDate}, \text{endDate}\}$$
Guest:
$$\{\text{reservationID}\}^+ = \{\text{guestID}, \text{reservationID}, \text{name}\}$$
Comment:
$$\{\text{reservationID}\}^+ = \{\text{reservationID}, \text{date}, \text{text}, \text{rating}\}$$
Room:
$$\{\text{roomNumber}\}^+ = \{\text{reservationID}, \text{capacity}, \text{price}\}$$
RoomStay:
$$\{\text{stay}, \text{room}\}^+ = \{\text{stay}, \text{room}\}$$
MeetingRoom:
$$\{\text{roomNumber}\}^+ = \{\text{roomNumber}, \text{description}\}$$
Bedroom:
$$\{\text{roomNumber}\}^+ = \{\text{roomNumber}, \text{bedroomType}\}$$
BedroomType:
$$\{\text{typeName}\}^+ = \{\text{typeName}, \text{description}\}$$

Photo:

$\{\text{photoID}\}^+ = \{\text{photoID}, \text{meetingRoom}, \text{bedroom}, \text{client}, \text{photoPath}, \text{date}, \text{description}\}$

Feature:

$\{\text{featureID}\}^+ = \{\text{featureID}, \text{featureName}\}$

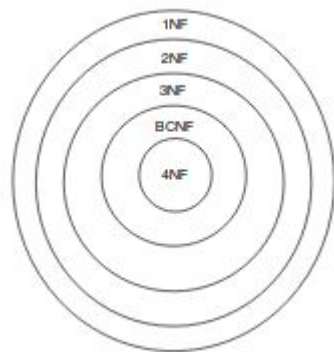
MeetingRoomFeature:

$\{\text{roomNumber}, \text{featureID}\}^+ = \{\text{roomNumber}, \text{featureID}\}$

BedroomTypeFeature:

$\{\text{typeName}, \text{featureID}\}^+ = \{\text{typeName}, \text{featureID}\}$

Normal Form Analysis



As we can see by the image above, the Boyce-Codd Normal Form (BCNF) is a subset of the third Normal Form (3NF). If we can prove each relation is in BCNF we also conclude the relation is in the 3NF.

A relation is in BCNF if the left side of the functional dependencies is a key for the relation. According to the last chapter (Functional dependencies), we proved exactly that.

Therefore, the relational model is in BCNF and consequently in 3NF.

Restrictions

Restrictions are used in databases to provide extra security to the user and also a better maintenance of the data. In the following chapter will be created different constraint in every class to obtain a reliable database.

Client

- ClientID must be positive and two different clients must not have the same ID;
- FirstName must not be null;
- LastName must not be null;
- Email must be unique for each client;
- Password must not be null;
- TaxNumber must be a 9-digit number, unique for each client;
- Photo must be null if no one is loaded or if someone is removed. Must be capable of be updated.

Reservation

- ReservationID must be positive and two different reservations must not have the same ID;
- ReservationDate must not be null and must be a day equal or some days after creationDate;
- CreationDate must not be null;
- FinalPrice should not be null. $\text{finalPrice} = \text{numOfNights} * (\text{price from Room} + \text{extraCost from Complement})$.
- Client must not be null. If deleted the reservation should also be deleted.

Cancelling

- Reservation must not be null;
- Client must not be null;
- Date must not be null and must be a day equal or some days before creationDate;
- Cost must not be null. Should be a tabled value according to the difference of days between date and reservationDate from reservation.

Complement

- ComplementID must be positive and two different complements must not have the same ID;
- Type must no be null, and specify a complement with the referred ID;

-
- ExtraCost must not be null and should be the price for that complement for one night.

Stay

- Reservation must not be null;
- StartDate must not be null;
- EndDate must not be null and must be some days after startDate.

Guest

- GuestId must not be null;
- Reservation must not be null;
- firstName should not be null;
- lastName should not be null.

Comment

- Stay must not be null;
- Date must not be null;
- Text must not be null;
- Rating must not be null and must be an integer between 0 and 5.

Room

- RoomNumber must not be null and must be unique for each room;
- Capacity must be positive;
- Price must be the value of that room for one night or hour (case of meetingRoom).

RoomStay

- Stay must not be null;
- Room must not be null. If a stay is scheduled for this room in these days, another one could not be.

MeetingRoom

- RoomNumber must not be null and must be unique for each room;
- Description must not be null and shortly describe this meetingRoom.

Bedroom

- RoomNumber must not be null and must be unique for each room;
- BedroomType must not be null. Should be the reference to the value of price in Room.

BedroomType

-
- BedroomTypeID must be positive and two different bedroom types must not have the same ID
 - Typename must not be null and should be unique;
 - Description must shortly describe this bedroomType.

Photo

- PhotoID must be positive and two different photos must not have the same ID;
- Date must not be null;
- Description must not be null and shortly describe this photo;
- PhotoPath must not be null;
- MeetingRoom must not be null if the photo describes a MeetingRoom;
- Bedroom must not be null if the photo describes a Bedroom;
- Client must not be null if the photo describes a Client.

Feature

- FeatureID must be positive and two different features must not have the same ID;
- FeatureName must be unique for each feature.

MeetingRoomFeature

- RoomNumber must not be null;
- FeatureID must not be null.

BedroomTypeFeature

- bedroomTypeID must not be null;
- FeatureID must not be null.