Deakin University

Database Fundamentals

OnTrack Submission

Miniproject Part-1 - Database Design and Normalisation

Submitted By:
Sathiyanarayanan Senthil Kumar
s223789819
2023/08/17 22:30

Tutor:
Mehul Warade

Outcome	Weight
Fundamental concepts of database	♦♦◊◊◊
Relational Database Modelling	♦♦♦♦♦

This task is to design a database model from scratch for a real life scenario. For this task, I have taken Booking.com and identified Entities, attributes and designed my own database model for it. By completing this task I got clear on database modelling - identifying entities, attributes, Primary keys and foreign keys identification and ERD. So I believe this task is an excellent example of ULO 1 and ULO 2.

August 17, 2023

Database Fundamentals Mini Project – Part 1

1. Organization - Booking.com:

Booking.com is a Dutch based organization grown to one of the world's leading digital travel companies that enables users to search for and book accommodations, flights, and other travel-related services all around the world. Even though the agency offers other services, their main functionality revolves around managing reservations, listing properties (from homes to hotels), displaying prices for the property, handling customer profiles, storing customer reviews and calculating the rating of the property based on customer reviews.

Official Website: https://www.booking.com/

2. Entities and Attributes:

To meet the above business requirements, the following entities and attributes are essential.

(a) PROPERTY:

The PROPERTY entity contains the details of the property like property name, address etc. These are listed as attributes below.

Key	Attribute	Description	Constrains
PK	**property_id**	Used to uniquely identify a property	NOTNULL, UNIQUE
	property_name	Name of the property (Need not be unique)	NOTNULL
	property_type	Tells whether it is a hotel/ house/ apartment	NOTNULL
	property_description	Gives a brief about the property	NOTNULL
	property_phone	Gives the phone number of the property	NOTNULL
	property_street	Tells the property number and street name	NOTNULL
	property_city	Used to store the city	NOTNULL
	property_country	Used to store the country	NOTNULL

Table 2.1: *PROPERTY - Attributes*

(b) ROOM:

The ROOM entity contains the information about each room as entering all the room details in property entity would lead to data inconsistency. The attributes are listed below.

Key	Attribute	Description	Constrains
PK	**room_num**	Room number along with property id will map to a unique room	NOTNULL
PK, FK	**property_id**	This is a foreign key that is used in PROPERTY entity	NOTNULL

	room_specification	Lists all the amenities like air conditioning, wifi, TV etc	NOTNULL
	room_price	Price for that room per night	NOTNULL
	room_capacity	Determines the maximum number of occupants for the room	NOTNULL
	room_status	Used to check whether the room is occupied/ available/ under maintenance	NOTNULL

Table 2.2: *ROOM - Attributes*

(c) CUSTOMER:

The CUSTOMER entity contains the details of all the customers who wish to use the website's services. The attributes are,

Key	Attribute	Description	Constrains
PK	**cust_id**	Uniquely identifies each customer	NOTNULL, UNIQUE
	cust_fname	First name of the customer	NOTNULL
	cust_lname	Last name of the customer	NOTNULL
	cust_phone	Customer's phone number	NOTNULL
	cust_email	Customer's Email	NOTNULL
	cust_credit_card	Credit card number of customer	NOTNULL
	cust_credit_exp	Credit card expiry date of the customer	NOTNULL

Table 2.3: *CUSTOMER - Attributes*

(d) RESERVATION:

The RESERVATION entity contains details of each reservation. The attributes are listed below.

Key	Attribute	Description	Constrains
PK	**reservation_id**	Uniquely stores each reservation	NOTNULL, UNIQUE
FK	**property_id**	Foreign key from PROPERTY entity	NOTNULL
FK	**room_num**	Foreign key from ROOM entity	NOTNULL
FK	**cust_id**	Foreign key from CUSTOMER entity	NOTNULL
	start_date	Start date of stay	NOTNULL
	end_date	End date of stay	NOTNULL
	guests	Number of guests	NOTNULL
	reservation_date	Date of reservation	NOTNULL

Table 2.4: *RESERVATION - Attributes*

(e) REVIEW:

The REVIEW entity stores all the customer reviews. The attributes are listed below.

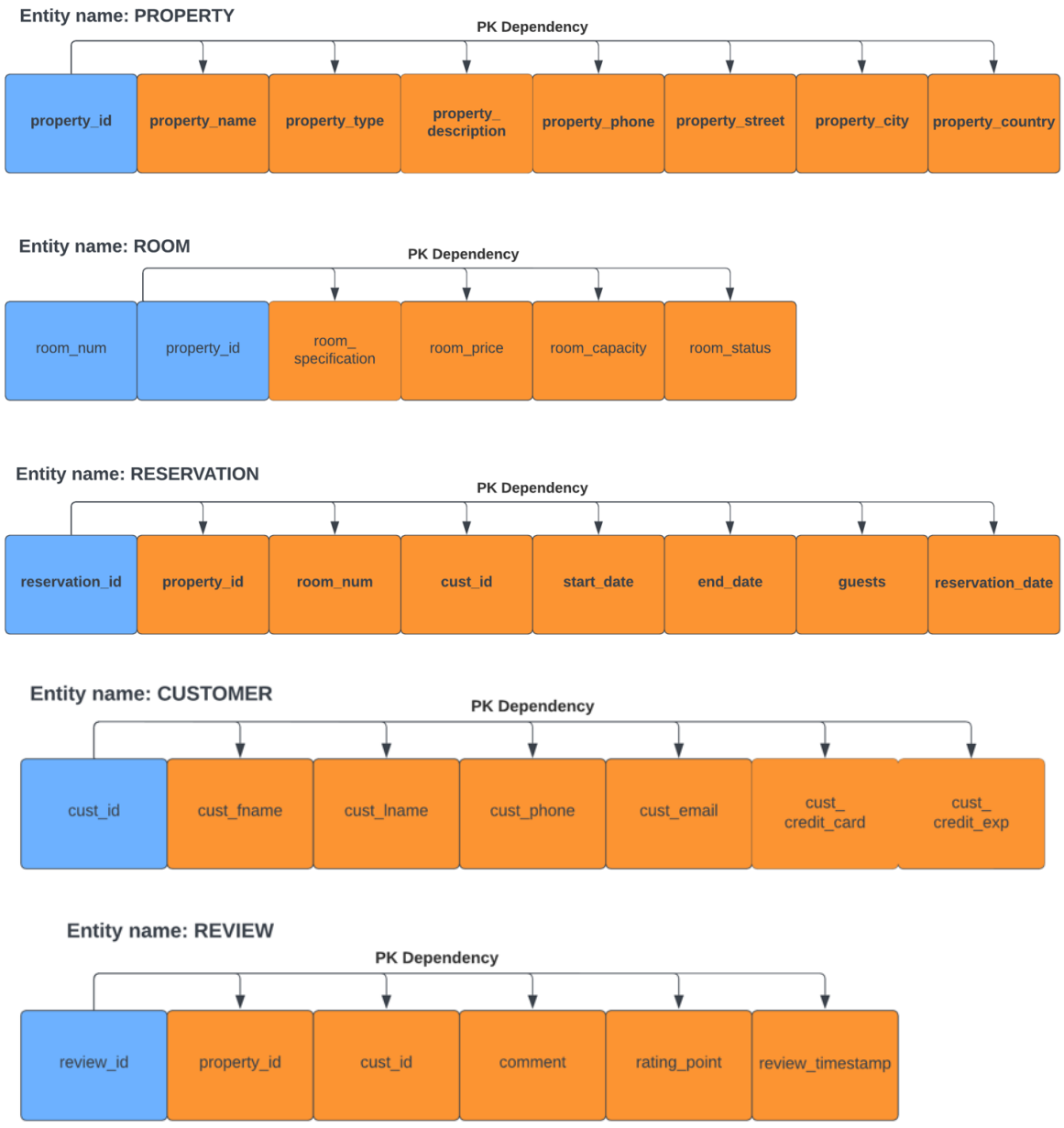
Key	Attribute	Description	Constrains
PK	**review_id**	Uniquely stores each review	NOTNULL, UNIQUE
FK	**property_id**	Foreign key from PROPERTY entity	NOTNULL
FK	**cust_id**	Foreign key from CUSTOMER entity	NOTNULL

	comment	Stores the review provided by the customer	NOTNULL
	rating_point	Gets and stores a value between 0 to 10	NOTNULL
	review_timestamp	Records the timestamp of the comment	NOTNULL

Table 2.5: *REVIEW - Attributes*

3.Normalization:

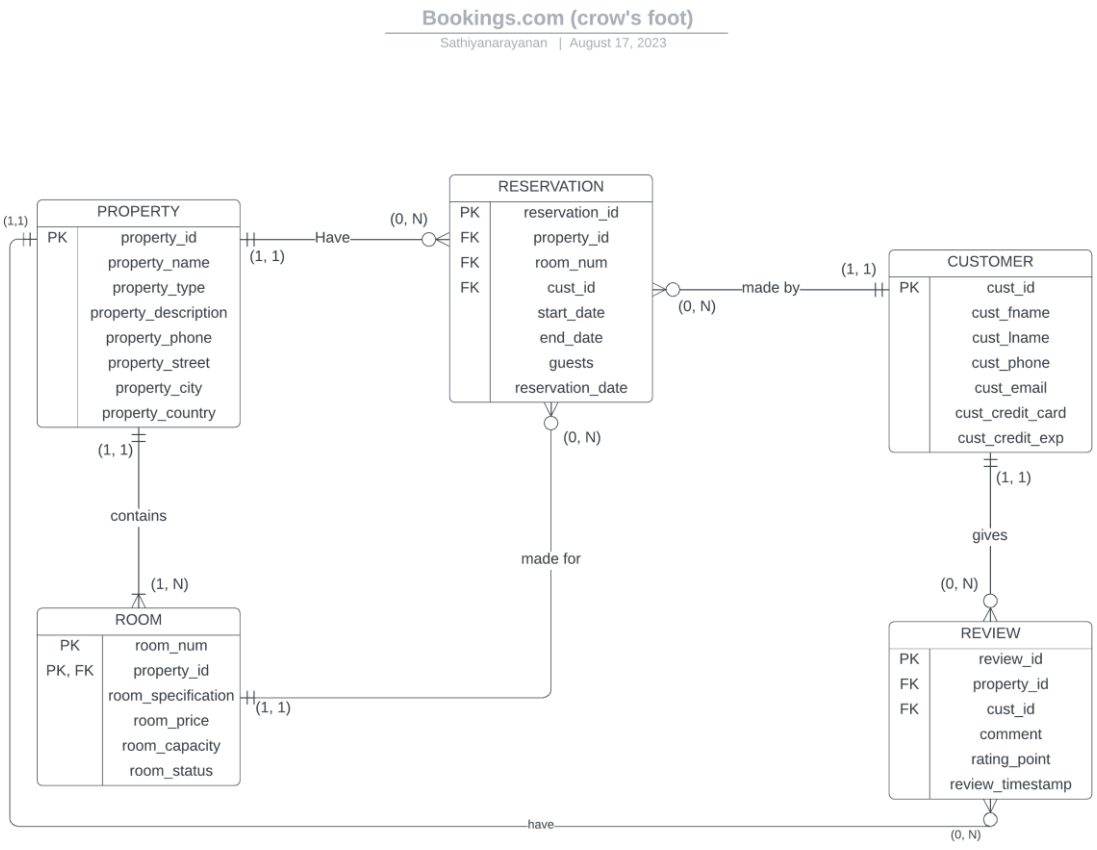
The dependency diagrams for all the above mentioned entities are given below,



Img 3.1: *dependency diagram*

- **For an entity to be 1NF,** it should not have repeating groups and it should have a primary key that uniquely identifies any single row. Both these conditions are satisfied for all 5 (PROPERTY, ROOM, RESERVATION, CUSTOMER and REVIEW) entities. The boxes highlighted in blue are Primary Keys which can uniquely identify each row and there are no repeating groups. Thus, this satisfies both the conditions, and all the above entities are in 1NF.
- **For the entities to be in 2NF,** it should be in 1NF and there should not be any partial dependency. Out of the above 5 entities, 4 (PROPERTY, RESERVATION, CUSTOMER and REVIEW) have only 1 primary key which means they are already in 2NF. For the remaining ROOM entity, as shown in the image, both primary keys (room_num and property_id) are required to get the details. So, there is no partial dependency. With this we can conclude all the entities are in 2NF.
- **For the entities to be in 3 NF,** it should be in 2NF, and it should not have any transitive dependency. For the above entities, as shown in the diagram, there is no transitive dependency i.e all the remaining attributes are dependent on their respective primary keys. So, we can conclude that all the above 5 entities are in 3NF.

4. Entity Relationship Diagram:



Img 4.1: *ERD – Crow's foot model for Booking.com*

The above diagram shows the relationship among all the 5 entities with their cardinalities. A brief of each is explained below,

- **PROPERTY <--- Have ---> RESERVATION:** One reservation would have one and only property, so its cardinality is (1, 1). Whereas 1 property can have 0 to N number of reservations, thus (0, N).
- **RESERVATION <--- made by ---> CUSTOMER:** One customer can make 0 to N number of reservations, so the cardinality (0, N). Whereas, in one reservation there can only be one customer, thus (1, 1).
- **PROPERTY <--- contains ---> ROOM:** One room can only be at one property, so the cardinality here is (1, 1). Whereas 1 property can have multiple rooms (Minimum 1), thus (1, N).
- **RESERVATION <--- made for ---> ROOM:** One room can have 0 to N number of reservations (Not at the same time though) so the cardinality is (0, N). Whereas one reservation can only have 1 room in it, thus (1, 1).
- **CUSTOMER <---gives---> REVIEW:** Each review is given by a customer. So, the cardinality here is (1, 1). Whereas a customer can give 0 to N number of reviews, thus the cardinality is (0, N).
- **PROPERTY <---have ---> REVIEW:** In one review, there can only be one property. Thus (1, 1). Whereas 1 property can have 0 to N number of reviews, thus (0, N).

Reference:

https://www.booking.com/content/about.en-gb.html