## **Group Members:**

* Raghd Al Juma D19125768
* YuJiao kan D20125727

**Domain Types**

|  |  |  |  |
| --- | --- | --- | --- |
| Domain Type Name | Type Definition | Reason Introduced | Example |
| Name | VARCHAR(30) | Attributes that are character based and short Like names. | firstname in entity customer.  “Sarach”,  “Eva” |
| Email | VARCHAR(254) | For Emails | Email in entity staff  “[Jom@gmail.com](mailto:Jom@gmail.com)” |
| Age | NUMERIC precision 3 | Attributes that are of a numeric value | Age in entity customer  “18,23,35” |
| Amount | FLOAT | Attributes that may hold decimals | Charge in entity bill  “79.8,305.6” |
| Phone | VARCHAR(10) | Attribute that are Phone numbers of length 10 (Irish Numbers) | phoneNumber in entity manager  “0876654322”  “0896654322” |
| Description | VARCHAR(30) | For attributes which are character based descriptions | address in entity staff |
| ID | NUMERIC Precision 10 | Attributes used as identifiers with Numeric type | bookingID in entity booking  “01,02,03” |
| Time | TIMESTAMP WITH LOCAL TIME ZONE | Date and time attributes which require a very clearly defined time element | bookingTime in entity booking  “18:30:22”  “10:23:12” |
| Date | DATE | Attributes which hold only the date and does not require a defined time | DOB in entity staff  “2021/12/5” |

## **Major Decisions**

Entities:

It was decided to have eight tables, these are:

* Customer
* Branch
* Table
* Bill
* Booking
* Staff
* Manager
* Guest

Attributes in Entities:

Customer:

This table will contain data (i.e. first name, last name, address and age) about the customer along with a customer ID as a primary key.

Branch:

This table will contain information about the branch (i.e. branch address and phone number) and each branch will have a unique ID.

Table:

This table will contain the table number as its primary key and the table location near the window or interior, and the table type (number of seats). It will also have a foreign key to the branch ID.

Guest:

This table will have all the data of the guests attending the booking. That is their first and last name, email, phone number and a unique guest ID. It will be connected to the booking ID by a foreign key.

Staff:

This table contains the waiters first name, last name, address, phone number, email, date of birth, date they started working and the Branch they work in connected by a foreign key with the Branch ID. Also each staff member will have a unique ID.

Manager:

This table contains the waiters first name, last name, address, phone number, email, date of birth, date they started working and the Branch they work in connected by a foreign key with the Branch ID. Also each manager will have a unique ID.

Booking:

This will contain the information about the booking and the contact tracing with the staff, like the number of people attending and the booking time, Also it will contain foreign keys of the Customer ID from the customer table, the table number from the Table table, the manager and the staff.

Also it will contain a primary key which is the Booking ID.

Bill:

We decided to have a Bill table which will have a unique bill ID, the duration of the booking, the charge which will have the total amount to be paid by the customer including their food bill and their fine for overstaying. Also it will have a Fine attribute which will have the amount the customer was fined for overstaying. Also the bill table has a foreign key of the booking ID and the customer ID.

Calculate fine:

We decided to calculate the fine by using an update statement, where we take the duration of the customer stay and minus 2 hours from it and multiply the rest with 5 euros to get the total fine. Then we update all of the fine rows where the length of stay is over 2 hours.

Relationships:

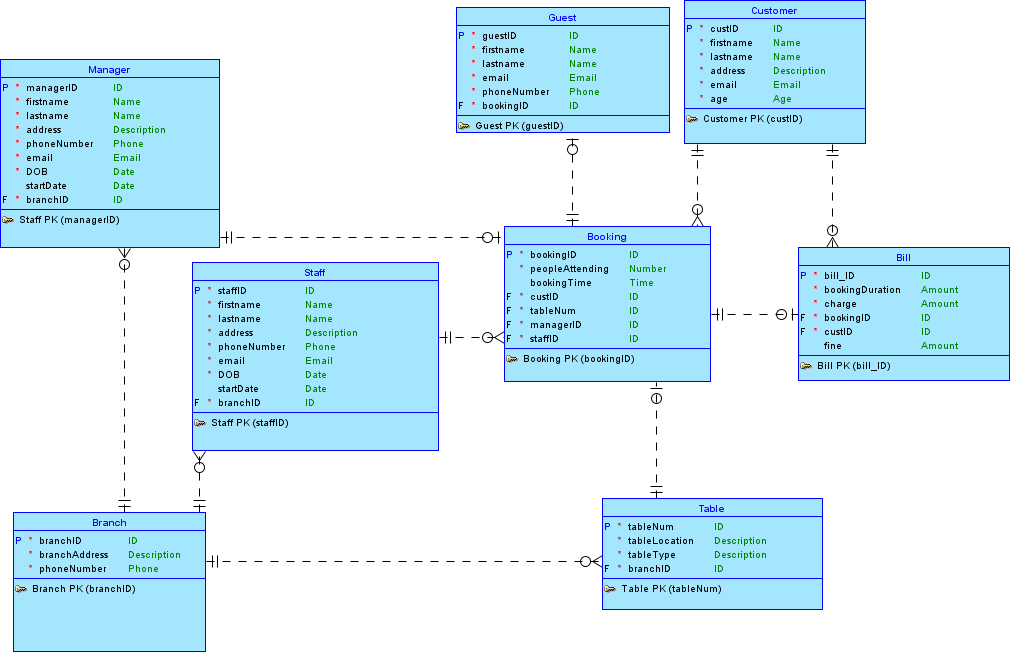
These are the relationships that we decided to add:

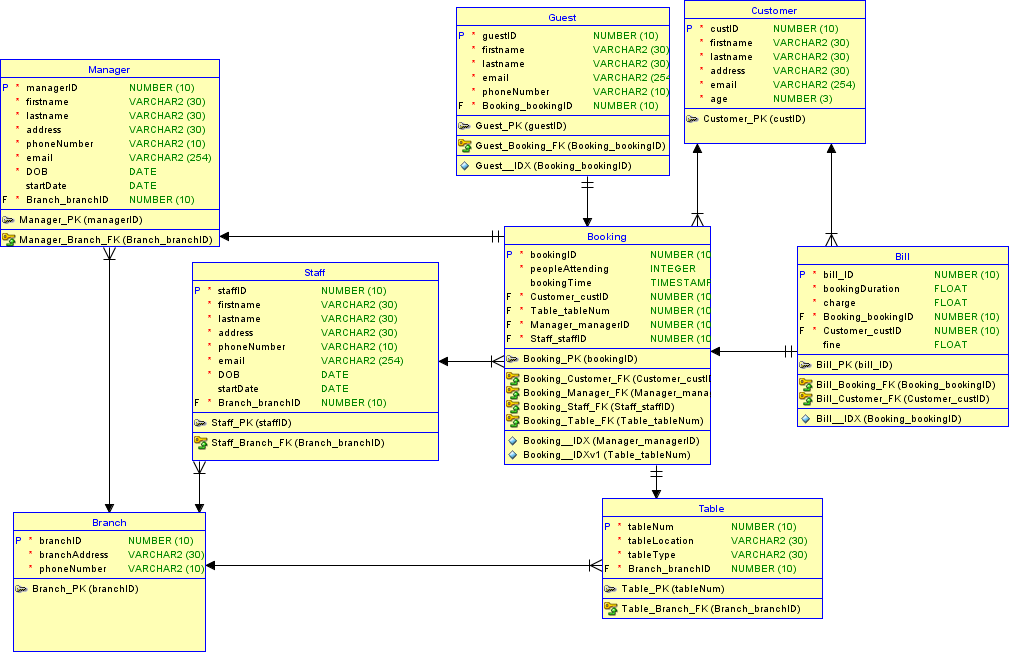
1. Manager- Branch: Many to One
2. Staff- Branch: Many to One
3. Booking- Manager: One to One
4. Booking- Staff: Many to One
5. Guest- Booking: One to One
6. Booking- Table: One to One
7. Table- Branch: Many to One
8. Booking- Customer: Many to One
9. Bill- Customer: Many to One
10. Bill- Booking: One to One

# 

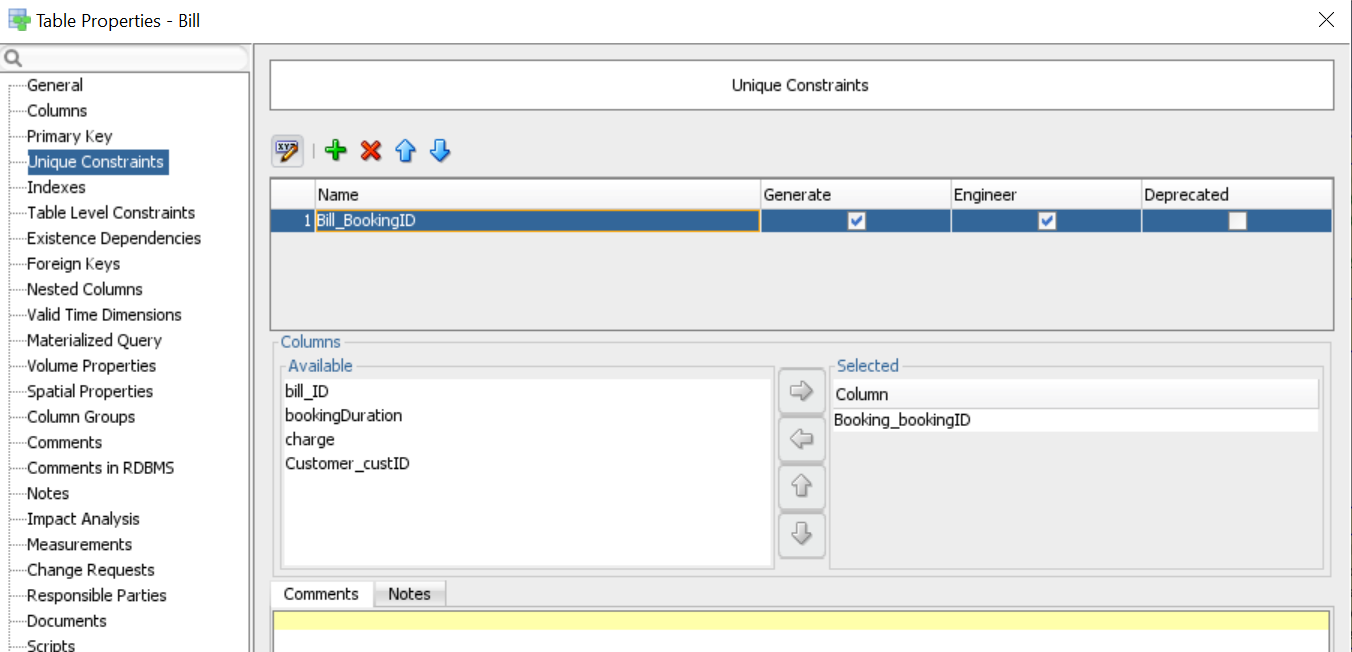
# 

# **ERD**

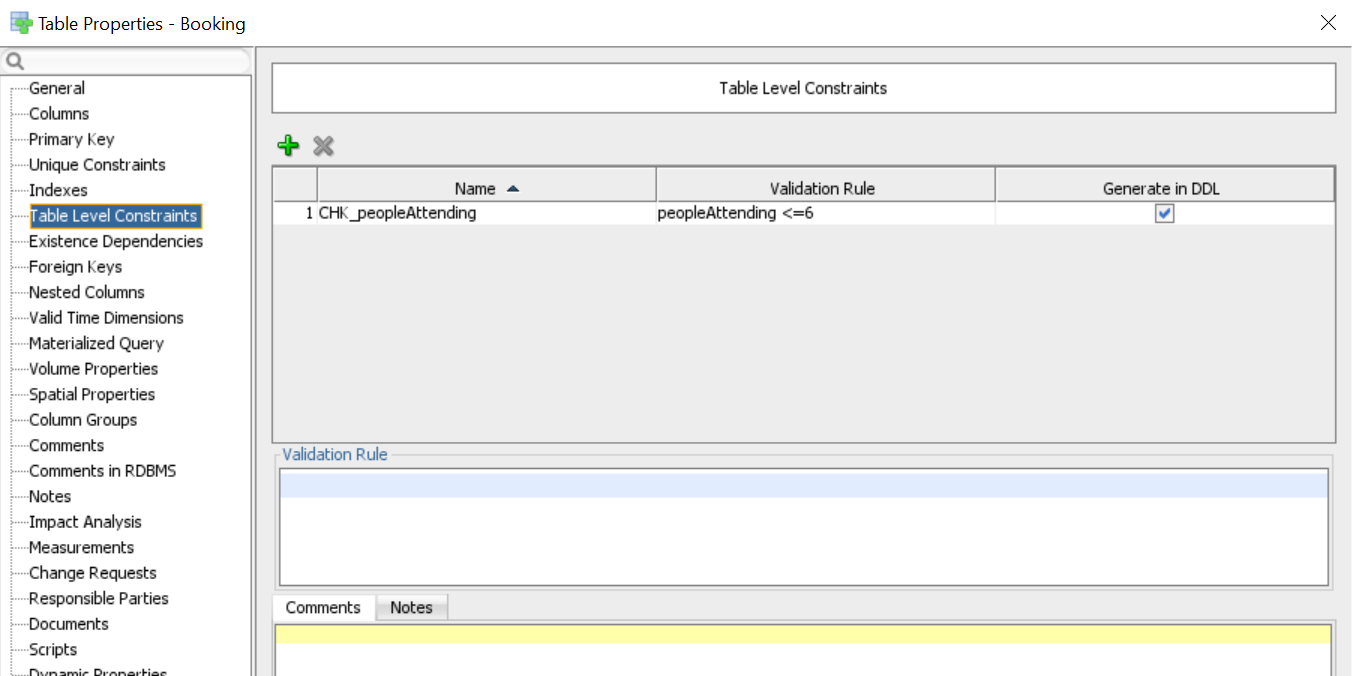
Screenshot of Logical Model

Screenshot of Physical Model

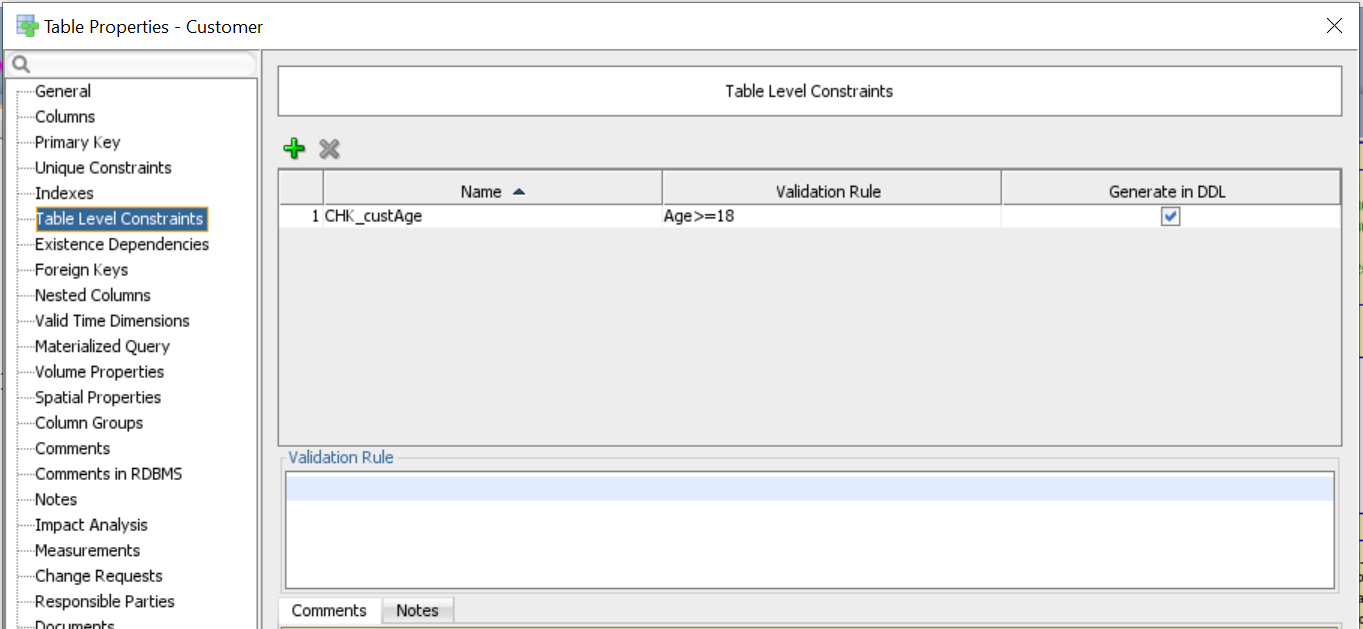
Bill Table Unique constraints:



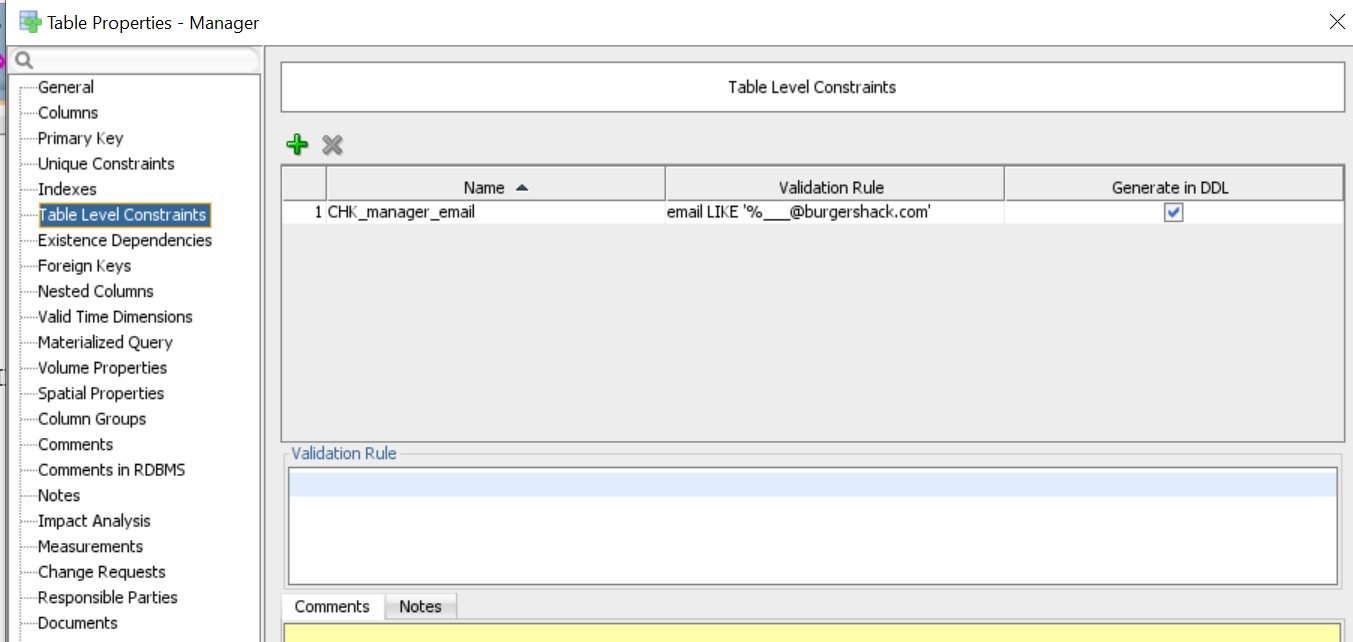
Booking table Check constraints:



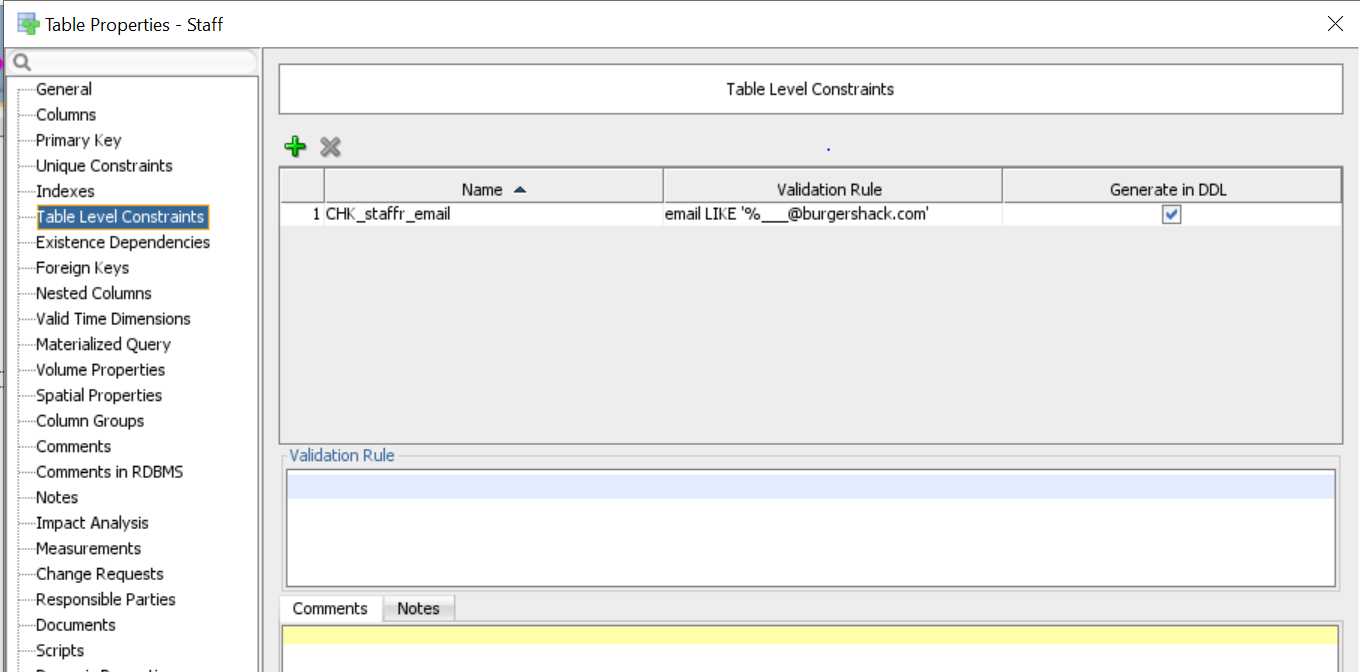
Customer table CHECK constraint:



Manager table CHECK constraint:



Staff table CHECK constraint:



**Query**

You can use a restaurant database to search a particular query. For example, Use a booking ID, find a customer booking table number, check if a customer who is booked is over 18 years and which one table stays longer than 2 hours, so on.