DB2 – Task

مدرس: د محمد عبد السلام احمد العشماوي
معيد: د منه الله ممدوح محمود محمد حسنين
الفرقة الرابعة – نظم معلومات - كلية تجارة وإدارة اعمال – جامعة حلوان – القاهرة
الموضوع: قاعدة بيانات نظام إدارة الفندق
رقم المجموعة: 11
العام الجامعي: 2021-2022 / 2021-12-25

الاسم	رقم الجلوس
احمد شعبان احمد عبد الجابر	18719
خالد محمود عبد المنعم الجابري	18742
زكريا شاهين صالح علواني صالح	18752
عبد الرحمن صالح احمد امين	18762
عمر محمد كمال مهدي الشيخ	18771

*يعتمد علي MS SQL

Index

**يمكنك الانتقال الي العنوان عن طريق Click علي العنوان*

Business Rules

المكان: فندق

المشكلة: تواجهه إدارة الفندق مشكلة في عملية حوكمة عمليات حجز خدمات وغرف الفندق، وعدم توفر امكانية تحليل البيانات بسبب عدم تخزين البيانات بشكل رقمي ومنظم لذلك قررت إدارة الفندق رقمنه عمليات حجز الخدمات والغرف

Tech Stack

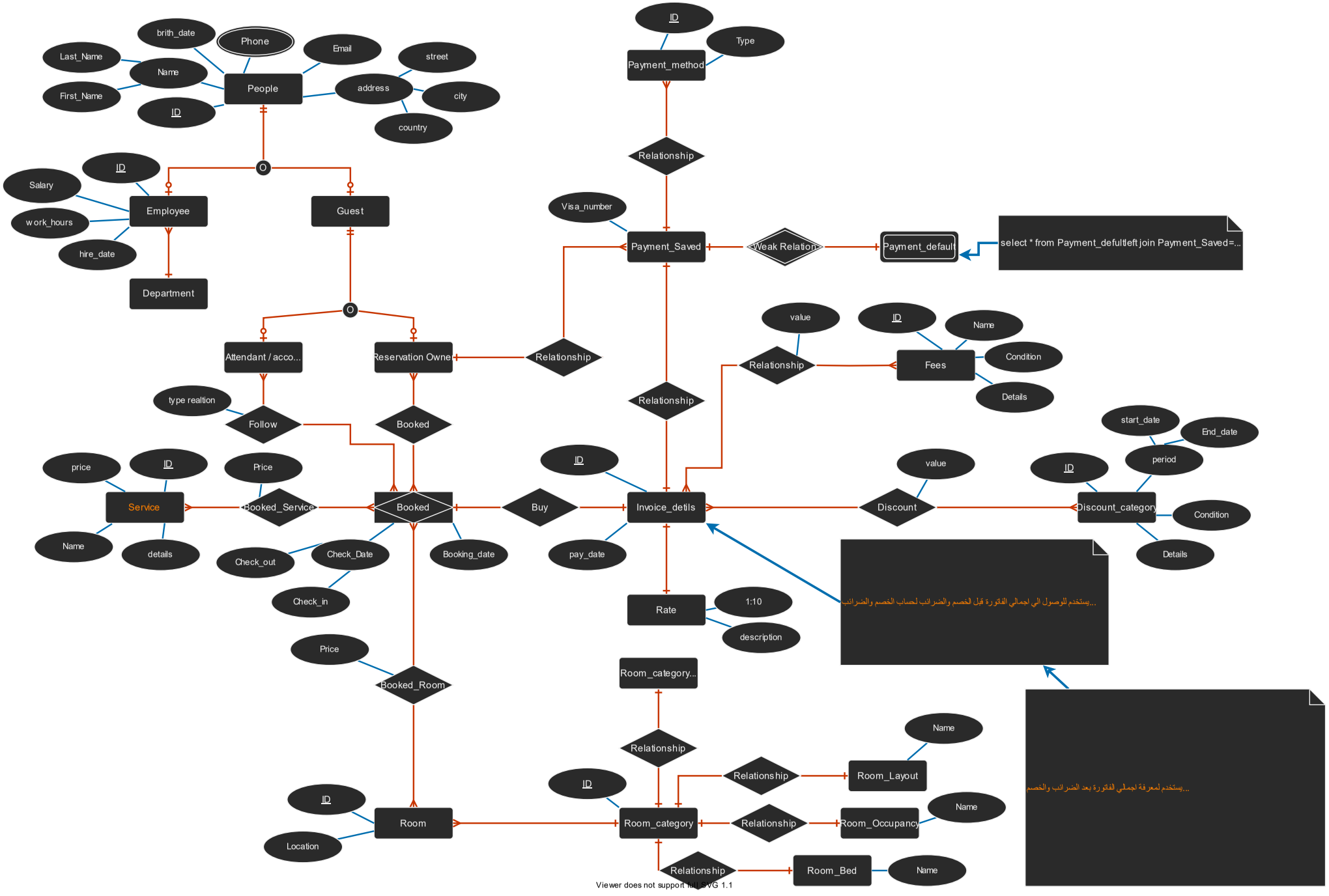
MS-SQL, ReacJS, C#(.Not Core)

Database Specifications

- يمكن ان يكون العميل هو صاحب الحجز او يكون العميل هو مرافق لصاحب الحجز من الأسرة او صديق
- يجب تخزين بيانات المرافقين كعملاء حيث يمكن ان يكون المرافق صاحب حجز في المستقبل... وايضا لتوفير فرصة لتحليل البيانات في المستقبل لتحسين خدمات الفندق
    - إمكانية معرفة أي عميل كان مرافق لمن في أي حجز معين
- يتوفر طرق للدفع، ويمكن للعملاء حفظ وسيلة الدفع المفضلة لهم
    - وأيضا تخزين بيانات جميع وسائل الدفع للعميل (رقم الفيزا)
- يتم الدفع عند الخروج من الفندق
- يمكن ان يحصل العميل علي خصم بموجب كوبون خصم ويمكن ان يتعدد الخصومات
    - حيث يقوم البرنامج بالتحقق من شروط تحقق الخصم ويطبق عليه الخصم
- يتم تطبيق ضرائب ورسوم علي قيمة الفاتورة بعد الخصم
    - مع توفير مرونة للتعامل مع التغيرات الضريبية المستمرة
    - من الممكن تحقق اكثر من قاعدة ضريبية علي نفس الفاتورة
- قد يتم تقديم خدمات اضافيه للعميل (حسب الطلب) وله سعر معين
- منتجات الفندق من غرف والخ.. تنقسم الي 3 متغيرات(مواصفات) رئيسية مع وجود إمكانية للإضافة متغيرات اخري
    - يمكن ان يتكرر نفس المواصفات لأكثر من 100 غرف مع اختلاف مكان الغرفة فقط
- يجب تخزين أسعار الغرف والخدمات بالأسعار القديمة إذا تم بيع غرفة او خدمة بهذا السعر القديم.... لتوفير فرصة لتحليل البيانات في المستقبل
- لكل موظف قسم محدد، وإتاحة إمكانية التوسع ليشمل النظام إدارة الكاملة للموظفين
- توفير إمكانية تقييم الفندق عند الدفع...حيث يكون سؤالين.. سؤال اختر من 1 الي 10.. وسؤال مفتوح لكتابة الاقتراحات

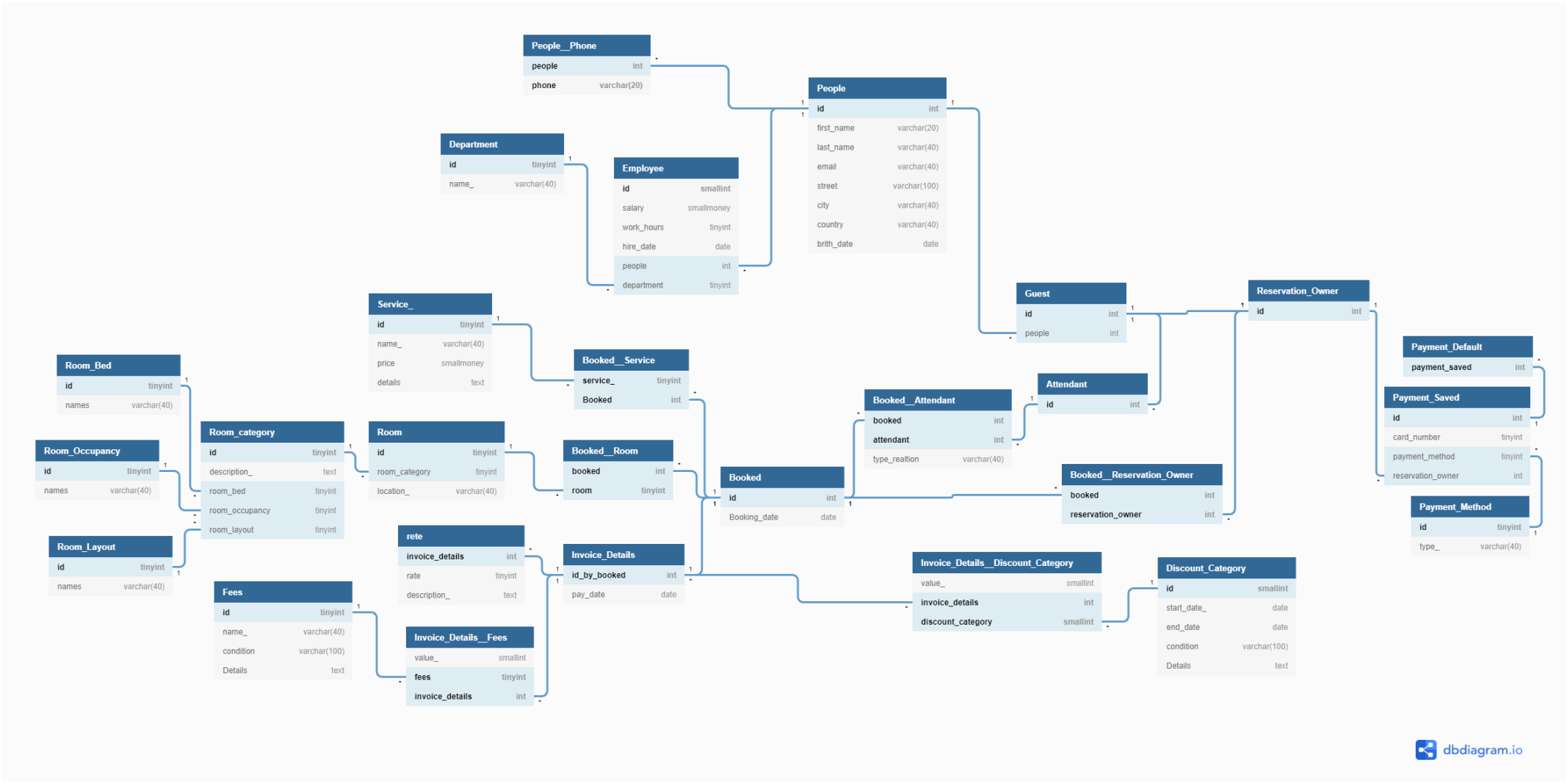
ونعرض هنا تحليل وتصميم وتنفيذ جزء Database

EERD



Viewer does not support full SVG 1.1

Schem


People__Phone
people int
phone varchar(20)
People
id int
first_name varchar(20)
last_name varchar(40)
email varchar(40)
street varchar(100)
city varchar(40)
country varchar(40)
brith_date date
Department
id tinyint
name_ varchar(40)
Employee
id smallint
salary smallmoney
work_hours tinyint
hire_date date
people int
department tinyint
Guest
id int
people int
Reservation_Owner
id int
Payment_Default
payment_saved int
Payment_Saved
id int
card_number tinyint
payment_method tinyint
reservation_owner int
Payment_Method
id tinyint
type_ varchar(40)
Service_
id tinyint
name_ varchar(40)
price smallmoney
details text
Booked__Service
service_ tinyint
Booked int
Room_Bed
id tinyint
names varchar(40)
Room_category
id tinyint
description_ text
room_bed tinyint
room_occupancy tinyint
room_layout tinyint
Room
id tinyint
room_category tinyint
location_ varchar(40)
Booked__Room
booked int
room tinyint
Booked__Attendant
booked int
attendant int
type_realtion varchar(40)
Attendant
id int
Booked
id int
Booking_date date
Booked__Reservation_Owner
booked int
reservation_owner int
Room_Occupancy
id tinyint
names varchar(40)
Room_Layout
id tinyint
names varchar(40)
rete
invoice_details int
rate tinyint
description_ text
Invoice_Details
id_by_booked int
pay_date date
Fees
id tinyint
name_ varchar(40)
condition varchar(100)
Details text
Invoice_Details__Fees
value_ smallint
fees tinyint
invoice_details int
Invoice_Details__Discount_Category
value_ smallint
invoice_details int
discount_category smallint
Discount_Category
id smallint
start_date_ date
end_date date
condition varchar(100)
Details text
dbdiagram.io

Query (Select & View)

يمكنك عرض الكود بشكل أفضل علي GitHub
https://github.com/zakaria-shahen/DB2_EERD

- لمعرفة اجمالي الفاتورة قبل الضرائب والخصم

```
-- Create view
create view Invoice_Before as
select
    iif(sum(Booked__Service.price) is null, sum(Booked__Room
.price),
        sum(Booked__Room.price) + sum(Booked__Service.price)
) as "Total berfore Discount"
from Invoice_Details
Left join Booked__Room on Invoice_Details.id_by_booked = Boo
ked__Room.booked
left join Booked__Service on Invoice_Details.id_by_booked =
Booked__Service.booked;

-- usgin View
select * from Invoice_Before;
```

- لمعرفة قيمة الفاتورة مع الضرائب والخصم

```
-- create view
create view Invoice as
select
    iif(sum(Booked__Service.price) is null, sum(Booked__Room
.price),
        sum(Booked__Room.price) + sum(Booked__Service.price)
) as "Price",
    sum(Invoice_Details__Discount.value_) as "Discount",
    sum(Invoice_Details__Fees.value_) as "Fees"
from Invoice_Details
Left join Booked__Room on Invoice_Details.id_by_booked = Boo
ked__Room.booked
left join Booked__Service on Invoice_Details.id_by_booked =
Booked__Service.booked
```

```
left join Invoice_Details__Discount on Invoice_Details.id_by
_booked = Invoice_Details__Discount.invoice_details
left join Invoice_Details__Fees on invoice_details.id_by_boo
ked = Invoice_Details__Fees.invoice_details;

-- using view
select * from Invoice;
```

-

Script: Create Database and tables

يمكنك عرض الكود بشكل أفضل علي GitHub

https://github.com/zakaria-shahen/DB2_EERD

Query

```sql
-- Create a new database called 'Hotel'
use master;
drop database if exists Hotel;
create database Hotel;
use Hotel;

-- Cerate tables

create table People(
    id int identity(1, 1),
    first_name varchar(20) not null,
    last_name varchar(40),
    email varchar(40),
    street varchar(100),
    city varchar(40),
    country varchar(40),
    brith_date date,
    primary key(id)
);

create table People__Phone(
    people int,
    phone varchar(20),
    primary key(people, phone),
    foreign key(people) references People(id)
);

create table Department(
    id tinyint identity(1, 1),
    name_ varchar(40),
```

```
    primary key(id)
);

create table Employee(
    id smallint identity(1, 1),
    salary smallmoney,
    work_hours tinyint,
    hire_date date,
    people int not null unique,
    department tinyint,
    primary key(id),
    foreign key(people) references People(id),
    foreign key(department) references Department(id)
);
create index people on Employee(people);
create index department on Employee(department);

create table Guest(
    id int identity(1, 1),
    people int not null unique,
    primary key(id),
    foreign key(people) references People(id)
);
create index people on Guest(people);


create table Reservation_Owner(
    id int,
    primary key(id),
    foreign key(id) references Guest(id)
);

--  other name table=> accompanying
create table Attendant(
    id int,
    primary key(id),
    foreign key(id) references Guest(id)
);
```

```
create table Booked(
    id int identity(1, 1),
    Booking_date date,
    Check_in date,
    Check_out date,
    primary key(id)
);


create table Booked__Reservation_Owner(
    booked int,
    reservation_owner int,
    primary key(booked, reservation_owner),
    foreign key(booked) references Booked(id),
    foreign key(reservation_owner) references Reservation_Ow
ner(id)
);

create table Booked__Attendant(
    booked int,
    attendant int,
    type_realtion varchar(40),
    primary key(booked, attendant),
    foreign key(booked) references Booked(id),
    foreign key(attendant) references Attendant(id)
);


create table Room_Bed (
    id tinyint identity(1, 1),
    names varchar(40),
    primary key(id)
);

create table Room_Occupancy(
    id tinyint identity(1, 1),
    names varchar(40),
```

```
    primary key(id)
);

create table Room_Layout(
    id tinyint identity(1, 1),
    names varchar(40),
    primary key(id)
);

create table Room_category(
    id tinyint identity(1, 1),
    description_ text,
    room_bed tinyint,
    room_occupancy tinyint,
    room_layout tinyint,
    primary key(id),
    foreign key(room_bed) references Room_Bed(id),
    foreign key(room_occupancy) references Room_Occupancy(id
),
    foreign key(room_layout) references Room_Layout(id)
);
create index room_bed on Room_category(room_bed);
create index room_occupancy on Room_category(room_occupancy)
;
create index room_layout on Room_category(room_layout);

create table Room(
    id tinyint identity(1, 1),
    room_category tinyint,
    location_ varchar(40),
    primary key(id),
    foreign key(room_category) references Room_category(id)
);
create index room_category on Room(room_category)

create table Booked__Room(
    booked int,
    room tinyint,
```

```
    primary key(booked, room),
    foreign key(booked) references Booked(id),
    foreign key(room) references Room(id)
);
create index booked on Booked__Room(booked)
create index room on Booked__Room(room)

create table Payment_Method(
    id tinyint,
    type_ varchar(40),
    primary key(id)
);

create table Payment_Saved(
    id int,
    card_number tinyint,
    payment_method tinyint,
    reservation_owner int,
    primary key(id),
    foreign key(payment_method) references Payment_Method(id
),
    foreign key(reservation_owner) references Reservation_Ow
ner(id)
);
create index payment_method on Payment_Saved(payment_method)
create index reservation_owner on Payment_Saved(reservation_
owner)

create table Payment_Default(
    payment_saved int,
    primary key(payment_saved),
    foreign key(payment_saved) references Payment_Saved(id)
);

create table Invoice_Details(
    id_by_booked int,
    pay_date date,
    primary key(id_by_booked),
```

```
    foreign key(id_by_booked) references Booked(id)
);


create table Discount_Category(
    id smallint,
    start_date_ date,
    end_date date,
    condition varchar(100), -- or JSON
    Details text,
    primary key(id)
);

create table Invoice_Details__Discount_Category(
    value_ smallint,
    invoice_details int,
    discount_category smallint,
    primary key(invoice_details, discount_category),
    foreign key(invoice_details) references Invoice_Details(
id_by_booked),
    foreign key(discount_category) references Discount_Categ
ory(id)
);

create table Fees(
    id tinyint,
    name_ varchar(40),
    condition varchar(100),
    Details text,
    primary key(id)
);

create table Invoice_Details__Fees(
    value_ smallint,
    fees tinyint,
    invoice_details int,
    primary key(fees, invoice_details),
    foreign key(fees) references Fees(id),
```

```
    foreign key(invoice_details) references Invoice_Details(
id_by_booked)
);

create table rete(
    invoice_details int,
    rate tinyint check(rate >= 1 and  rate <= 10),
    description_ text,
    primary key(invoice_details),
    foreign key(invoice_details) references Invoice_Details(
id_by_booked)
);

create table Service_(
    id tinyint identity(1, 1),
    name_ varchar(40),
    price smallmoney,
    details text,
    primary key(id)
);

create table Booked__Service(
    service_ tinyint,
    Booked int,
    primary key(service_, Booked),
    foreign key(service_) references Service_(id),
    foreign key(booked) references Booked(id)
);
```

Tools

- VSCode
- Azure Data Studio
- draw.io
- https://dbdiagram.io