ASSIGNMENT

DBI202 - DATABASE SYSTEM OF HOME RENTAL

STUDENT NAME: NGUYEN GIANG NAM | SE161942
STUDENT NAME: LE TRUNG SON | SE161501

Teacher: Nguyen Quoc Trung

FPT UNIVERSITY

Contents

A. INTRODUCE THE PROBLEM

In recent years, the rental service is one of the industries with the highest growth in the country. Many boarding house race to develop continuously and rapidly according to the development of society in terms of size and quality.

Currently, motels have to directly receive and manage a large and frequent volume of guests, along with a series of services arising according to the needs of customers. As a result, the management of the boarding house's business is becoming more and more complicated.

Moreover, the management work is not only the management of the flow of guesses to the boarding house, the use of various types of services, etc., but the management must also meet the demand for reporting on different types of businesses, income, business situation of the boarding house... so that we can give direction and develop a development plan for that business. But with today's manual storage and handling, it will take a lot of time and manpower without high efficiency. Therefore, it is necessary to computerize the management form, specifically to build a software to meet the needs of comprehensive, unified management and achieve the highest efficiency for business activities of the inn.

Due to the above needs, our group decided to choose the topic "Boarding House Management" as a key to the need to apply information technology to business.

1. **Introduction:**

Building a boarding house management software that provides full information about tenants, types of room, rental status and services associated with that rental room.

2. **Development goals:**

Making the management of the boarding house easier, minimizing manual labor. Bring the main and high efficiency in revenue and expenditure every month.

3. **Description:**

- Customers must provide information to the housekeeper when registering to rent a room such as: Full name, address, ID card number... and the innkeeper will enter and store the above information.
- The house owner will give the customer information about the room and the boarding house's services.
- Store information about customers who rent rooms and customers who use services.
- Look up and update room and customer status.

I) DESCRIBE THE PROBLEM

1. Room operation management:

Management of boarding house rental registration: When guests come to rent a room, the housekeeper must record the information related to the guest according to the general regulations of the hotel to facilitate the management of guests during their registration. All registration information to rent a room is based on the name of the guest registered, the number of rooms, the date of arrival, the date of departure and the contact phone number.

Management of check-in information: After completing the registration procedure, the tenant will be given a room by the housekeeper and officially record the time that they check-in. The hostel will manage information such as check-in room number, check-in name, check-in date, check-in time. If you have booked a room in advance, the check-in information must match the room registration information previously declared.

Check-out management: When the tenant wants to check out, the housekeeper is responsible for checking the room they pay based on the check-in information and the services they use. Also have to bill for everything they have to pay. The check-out is managed by the innkeeper with the following information: Check-out number, full name of payer, date of check-out, time of check-out (check-out policy).

2. Customer management:

When guests come to register and check in at the house, each person must provide information such as: Full name, gender, address, phone, ID number of their nationality (PassPort, Visa) ..., for the house to have can manage and extract information as required (legally).

3. Management of the boarding house's services:

In addition to renting rooms, customers also need to use other services. To meet that, the house has provided all kinds of services and these types of services will be managed according to: service name, unit of calculation, unit price.

4. Room manager:

Each boarding house will have many types of rooms, each type has many rooms, each room has many different equipment. Therefore, the rooms of the inn must be managed based on the number of rooms, the type of room, the basic price; Facilities are managed by: device name, quantity.

II) FUNCTIONAL REQUIREMENTS:

1. **Storage requirements:**
    - Store user information.
    - Store customer information.
    - Store customer information for check-in and check-out.
    - Store room information.
    - Store room status information.
    - Store room type information.
    - Store device information.
    - Store service information.
    - Store customer service usage information.
    - Store service type information.
    - Store unit information.
    - Store the check-out policy.
    - Store room rental registration slip information.
    - Store check-in slip information.
    - Store invoice information.
2. **Important output:**
    - Print customer list.
    - Print invoices.

B. ENTITY – RELATIONSHIP – ER

I) DIFINITION ENTITY – ATTRIBUTE

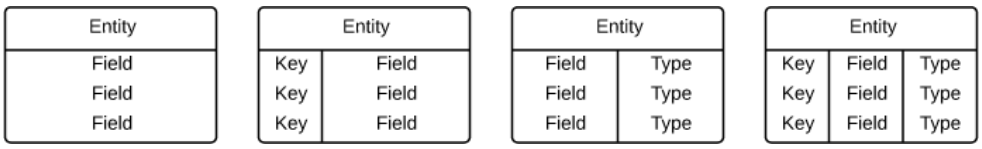
Base on the problem description and management objectives, we can present several entities and attributes of the entity as follow:

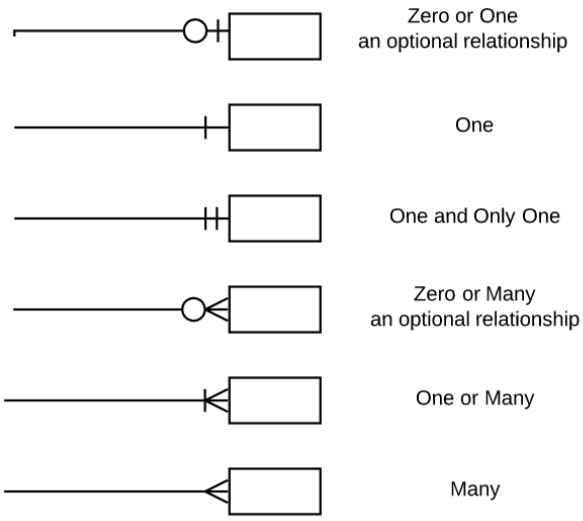
- Tenant: **TenantID**, TenantName, IdentifyNumber, Sex, Job, Address, PhoneNumber, TenantStatus
- Room: **RoomID**, Floor, MaxPeople, Price, RoomStatus
- Service: **ServiceID**, ServiceName, ServicePrice
- RentingManagement: **RentingID**, **TenantID**, **RoomID**, RentingDate, RentingStatus, Deposit
- Payment: **PaymentID**, **RoomID**, PaymentDate, PaymentPrice, Reason
- MoneyManagement: **CollectionID**, **ServiceID**, **TenantID**, **RoomID**, CollectionDate
- Equipment: **EquipmentID**, EquipmentName, Brand, Quantity, EquipmentStatus, EquipmentPrice
- Room_Equipment: **RoomID**, **EquipmentID**
- CheckOut: **CheckOutID**, **TenantID**, CheckOutDate, Note

II) SET-UP ENTITY – RELATIONSHIP

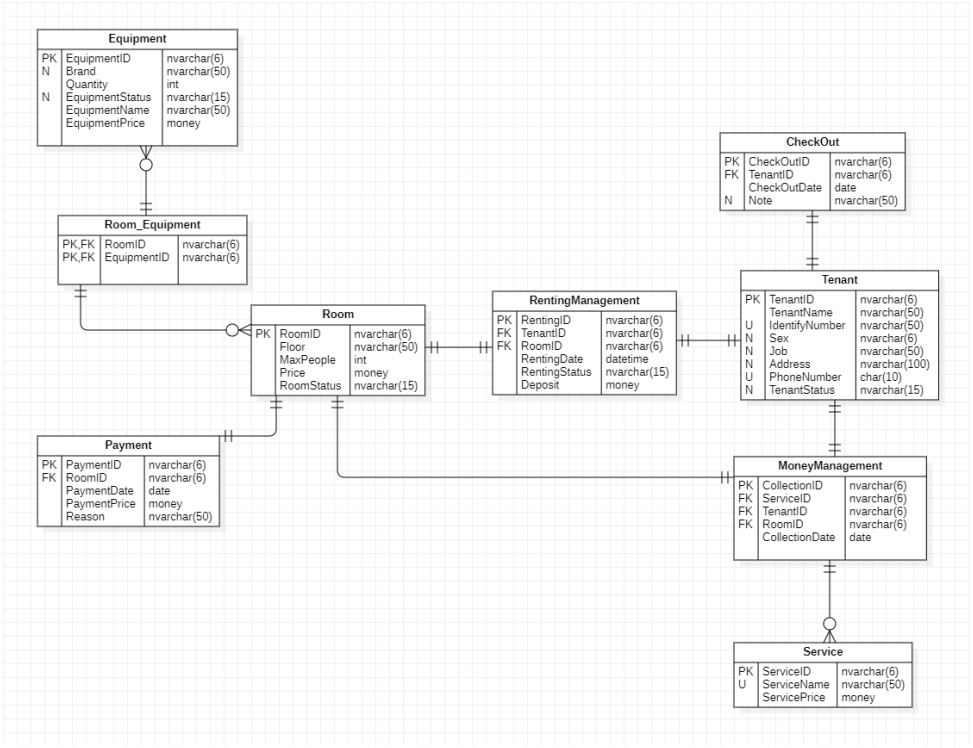
* Some symbols used in the model

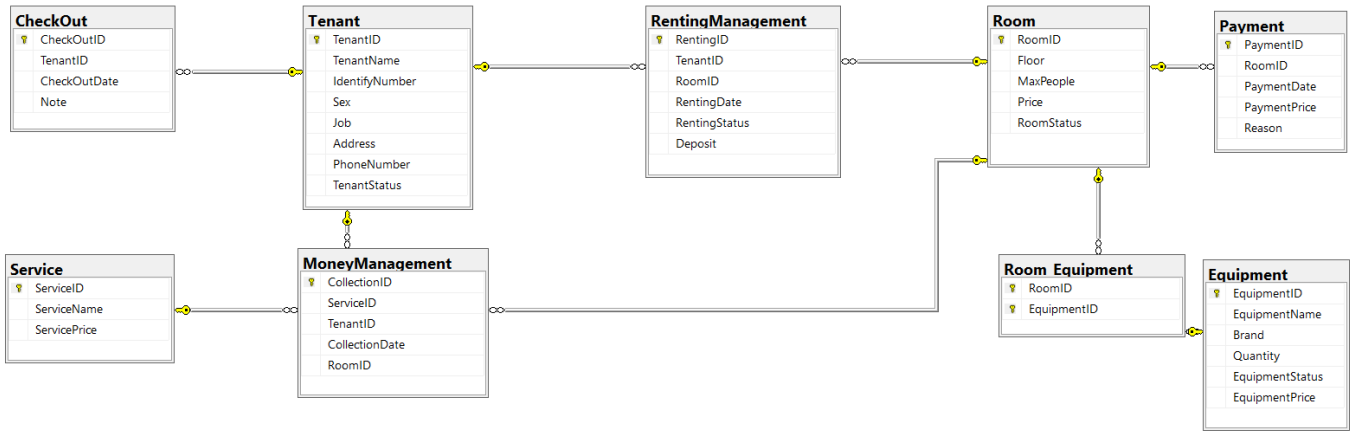
- FK: Foreign Keys
- PK: Primary Keys
- U: Unique
- N: Nullable





Crow's Foot Notation


Equipment
PK EquipmentID nvarchar(6)
N Brand nvarchar(50)
Quantity int
N EquipmentStatus nvarchar(15)
EquipmentName nvarchar(50)
EquipmentPrice money
CheckOut
PK CheckOutID nvarchar(6)
FK TenantID nvarchar(6)
CheckOutDate date
N Note nvarchar(50)
Room_Equipment
PK,FK RoomID nvarchar(6)
PK,FK EquipmentID nvarchar(6)
Tenant
PK TenantID nvarchar(6)
TenantName nvarchar(50)
U IdentifyNumber nvarchar(50)
N Sex nvarchar(6)
N Job nvarchar(50)
N Address nvarchar(100)
U PhoneNumber char(10)
N TenantStatus nvarchar(15)
Room
PK RoomID nvarchar(6)
Floor nvarchar(50)
MaxPeople int
Price money
RoomStatus nvarchar(15)
RentingManagement
PK RentingID nvarchar(6)
FK TenantID nvarchar(6)
FK RoomID nvarchar(6)
RentingDate datetime
RentingStatus nvarchar(15)
Deposit money
Payment
PK PaymentID nvarchar(6)
FK RoomID nvarchar(6)
PaymentDate date
PaymentPrice money
Reason nvarchar(50)
MoneyManagement
PK CollectionID nvarchar(6)
FK ServiceID nvarchar(6)
FK TenantID nvarchar(6)
FK RoomID nvarchar(6)
CollectionDate date
Service
PK ServiceID nvarchar(6)
U ServiceName nvarchar(50)
ServicePrice money


CheckOut
CheckOutID
TenantID
CheckOutDate
Note
Tenant
TenantID
TenantName
IdentifyNumber
Sex
Job
Address
PhoneNumber
TenantStatus
RentingManagement
RentingID
TenantID
RoomID
RentingDate
RentingStatus
Deposit
Room
RoomID
Floor
MaxPeople
Price
RoomStatus
Payment
PaymentID
RoomID
PaymentDate
PaymentPrice
Reason
Service
ServiceID
ServiceName
ServicePrice
MoneyManagement
CollectionID
ServiceID
TenantID
CollectionDate
RoomID
Room Equipment
RoomID
EquipmentID
Equipment
EquipmentID
EquipmentName
Brand
Quantity
EquipmentStatus
EquipmentPrice

C. DATA DICTIONARY

Just for example on some tables (other table are similar, you have to define all the tables in your database).

I) DEFINITION OF TABLES

1. Table Tenant

Column Name	Data Type	Default	Check	Key/ Index/ Constraint
TenantID	nvarchar(6)		TExxxx	PK, not null
TenantName	nvarchar(50)			Not null
IdentifyNumber	nvarchar(50)			Unique, not null
Sex	nvarchar(6)		'Female' or 'Male'	Not null
Job	nvarchar(50)			
Address	nvarchar(100)			
PhoneNumber	char(10)		0 to 9	Unique, not null
TenantStatus	nvarchar(15)	Valid	'Valid' or 'Expired'	

Example:

TenantID	TenantName	IdentifyNumber	Sex	Job	Address	PhoneNumber	TenantStatus
TE0006	Nguyen Van An	1237894560	Male	Teacher	731 Tran Hung Dao, Q5, TpHCM	8823451456	Valid
TE0069	Nguyen Giang Nam	2726856069	Male	Student	32/3 Tran Binh Trong, Q5, TpHCM	9787900564	Valid
TE0050	Le Trung Son	0908070605	Male	Student	34/34B Nguyen Trai, Q1, TpHCM	3745985471	Expired
TE0123	Le Thi Dung	4578342878	Female	Worker	45/2 An Duong Vuong, Q5, TpHCM	5754673009	Valid

Code:

```
CREATE TABLE Tenant (
        TenantID nvarchar(6) CONSTRAINT check_tenantid CHECK(TenantID LIKE 'TE[0-9][0-9][0-9][0-9]' ) not null,
        TenantName nvarchar(50) not null,
        IdentifyNumber nvarchar(50) unique not null,
        Sex nvarchar(6) CONSTRAINT check_sex CHECK(Sex IN('Female','Male')) not null,
        Job nvarchar(50),
        [Address] nvarchar(100),
```

PhoneNumber CHAR(10) unique CONSTRAINT check_phone CHECK(PhoneNumber LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]') not null,

TenantStatus nvarchar(15) DEFAULT 'Valid' CONSTRAINT check_tenantstatus CHECK(TenantStatus IN('Valid','Expired')),

CONSTRAINT pk_tenant PRIMARY KEY (TenantID),

);

Result:

	TenantID	TenantName	IdentifyNumber	Sex	Job	Address	PhoneNumber	TenantStatus
1	TE0001	Nguyen Van An	1237894560	Male	Teacher	731 Tran Hung Dao, Q5, TpHCM	8823451456	Valid
2	TE0050	Le Trung Son	0908070605	Male	Student	34/34B Nguyen Trai, Q1, TpHCM	3745985471	Expired
3	TE0069	Nguyen Gian...	2726856069	Male	Student	32/3 Tran Binh Trong, Q5, TpH...	9787900564	Valid
4	TE0123	Le Thi Dung	4578342878	Fe...	Worker	45/2 An Duong Vuong, Q5, TpH...	5754673009	Valid

2. Table Room

Column Name	Data Type	Default	Check	Key/ Index/ Constraint
RoomID	nvarchar(6)		RMxxxx	PK, not null
Floor	int		between 1 to 6	Not null
MaxPeople	int	1	between 1 to 4	Not null
Price	money			Not null
RoomStatus	nvarchar(15)	Empty	'Empty' or 'Rented'	Not null

Example:

RoomID	Floor	MaxPeople	Price	RoomStatus
RM0001	1	4	4300000	Rented
RM0098	6	2	2000000	Rented
RM0070	4	1	1600000	Empty
RM0032	2	3	3100000	Empty

Code:

CREATE TABLE Room (

RoomID nvarchar(6) CONSTRAINT check_roomid CHECK(RoomID LIKE 'RM[0-9][0-9][0-9][0-9]') not null,

[Floor] int CONSTRAINT check_floor CHECK([Floor] BETWEEN 1 AND 6) not null,

MaxPeople int DEFAULT 1 CHECK(MaxPeople BETWEEN 1 AND 4) not null,

Price money not null,

RoomStatus nvarchar(15) DEFAULT 'Empty' CONSTRAINT check_roomstatus CHECK(RoomStatus IN('Empty','Rented')) not null,

CONSTRAINT pk_room PRIMARY KEY (RoomID)

);

Result:

	RoomID	Floor	MaxPeople	Price	RoomStatus
1	RM0001	1	4	4300000.00	Rented
2	RM0032	2	3	3100000.00	Empty
3	RM0070	4	1	1600000.00	Empty
4	RM0098	6	2	2000000.00	Rented

3. Table Service

Column Name	Data Type	Default	Check	Key/ Index/ Constraint
ServiceID	nvarchar(6)		SExxxx	PK, not null
ServiceName	nvarchar(50)			Unique, not null
ServicePrice	money			not null

Example:

ServiceID	ServiceName	ServicePrice
SE0050	Internet	200000
SE0029	Room Cleaning	250000
SE0022	Insecticidal	440000
SE0016	Clothes Washing	200000

Code:

```
CREATE TABLE [Service] (
        ServiceID nvarchar(6) CONSTRAINT check_serviceid  CHECK(ServiceID LIKE 'SE[0-9][0-9][0-9][0-9]' ) not null,
        ServiceName nvarchar(50) unique not null,
        ServicePrice money not null,
        CONSTRAINT pk_service PRIMARY KEY (ServiceID)
);
```

Result:

	ServiceID	ServiceName	ServicePrice
1	SE0016	Clothes Washing	200000.00
2	SE0022	Insecticidal	440000.00
3	SE0029	Room Cleaning	250000.00
4	SE0050	Internet	200000.00