**ASSIGNMENT**



Môn:**DBI202- Database System**

Lớp: **SI1401**

Nhóm Như Quỳnh

* *Nguyễn Lê Như Quỳnh*
* *Phạm Văn Trọng Nhân*
* *Phạm Hồng Sáng*
* *Trương Khánh Tường*

Assignment of duties:

* Description: Nhân
* Management Objectives: Quỳnh
* Entity-attribute identification Quỳnh
* Entity-contact modeling: Nhân
* Convert the ER model to a relational schema: Sáng
* Standardize the schema: Tường
* Relational model :Sáng
* Format the tables: Quỳnh

Table Content

**I. Introduce problem** (Giới thiệu bài toán)

1. Description (Mô tả bài toán)

2. Management Objectives (Mục tiêu quản lý)

**II. Describe Enity** (Mô tả thực thể)

1. Entity-attribute identification (Xác định thực thể - thuộc tính)

2. Entity-contact modeling (Lập mô hình thực thể- liên hệ)

**III. Related Schema - Related Model** (Lược đồ quan hệ - Mô hình quan hệ)

1. Convert the ER model to a relational schema (Chuyển mô hình ER sang lược đồ quan hệ)

2. Standardize the schema (Chuẩn hóa lược dồ)

3. Relational model (Mô hình quan hệ)

**IV. Data Dictionaries** (Từ điển dữ liệu)

1. Format the tables (Định dạng các bảng)

**I. Introduce problem (Giới thiệu bài toán)**

**1. Description**

With the development of current science and technology, old methods are gradually being replaced by new methods which are faster, more accurate, more efficient and more economical. The same holds true!

Instead of using chalk to write cars and handing out vouchers like before, nowadays most parking lots use cards and cameras to manage the car more efficiently, accurately and quickly.

ABC Company needs to use the database system to manage the garage for its employees.

* Operating room information including employee arrival time, staff exit time and employee identification number. An executive room manages many security guards and vehicle access history. Because there are a lot of employees who are managed by the operating room.
* Employee information includes employee IDs that are identical to ID cards, employee names, phone numbers, positions, gender, qualifications, salary, and address. Many employees check multiple cards. Because one employee must check multiple cards at work, one card can be checked by different employees when entering the garage at different times.
* Card information including vehicle code, card code, customer code, registration time, expiry time. Multiple tags can be kept by a single customer, multiple cards can store multiple histories. Because A customer can have many cars, each car has 1 card. When a card is used for more than one day, it will be saved to many history tables, whereas one day a history board must store the history of many different vehicles.
* Customer information includes customer code that matches with ID, customer name, address, date of birth, gender. A customer can own multiple cars. As allowed by Vietnamese law.
* Vehicle information including vehicle code coincides with the license plate, vehicle color, vehicle owner code, model. Many vehicles can be owned by 1 vehicle owner.
* Historical information including vehicle arrival time, vehicle departure time, card code. Many vehicles are saved by a history, an operating room manages a history board. Because every day a history table stores the history of many cars.

**2. Management Objectives**

* Managing employees' working days and hours.
* Manage the registration time, expiry date, code of vehicle card, license plate and customer code
* Manage personal information, codes, positions and salaries of employees.
* Manage the access time history of each card
* Manage ID, name, date of birth and gender of customers in the database
* Manage the license plate, color,vehicle model and its owner in the database.

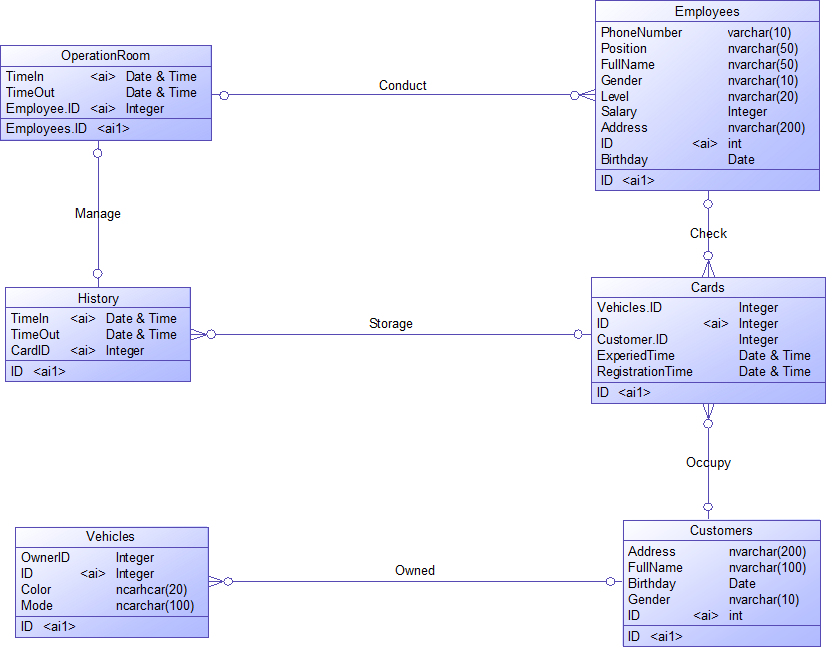
**II. Describe Enity (Mô tả thực thể)**

**1. Entity-attribute identification (Xác định thực thể - thuộc tính)**

Based on the problem description and management goal, we can give some entities and attributes of that entity as follows:

* **OperationRoom:** TimesIn, TimesOut,ID.
* **Card:** CardID, TimesRegisttration, TimesExpired, license plates, CusID.
* **Employees:** ID, Gender, PhoneNumber,Position , Salary, Fullname, Address,Level,birthday.
* **Customers:** CusID, Gender, Fullname, Birthday
* **Vehicle:** model, Color, OwnerID, ID
* **History:** TimesIn, TimesOut, CardID.

**2.** **Entity-contact modeling (Lập mô hình thực thể- liên hệ)**

****

**III. Related Schema - Related Model (Lược đồ quan hệ - Mô hình quan hệ)**

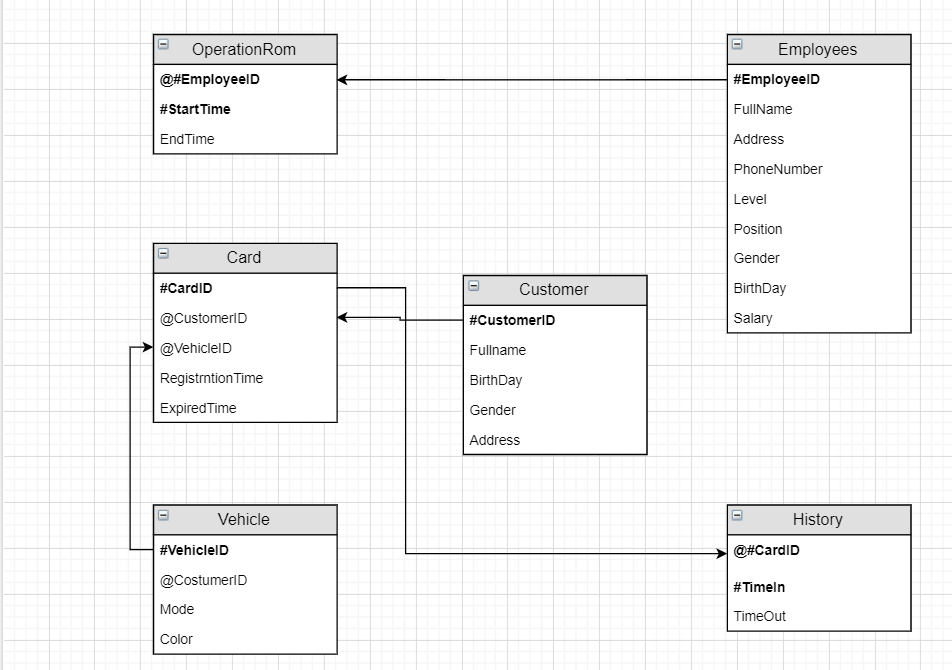
**1. Convert the ER model to a relational schema (Chuyển mô hình ER sang lược đồ quan hệ)**

|  |  |  |  |
| --- | --- | --- | --- |
| Relational schema | Key | Foreign Key | Function dependencies |
| OperationRoom | Startime, EmployeeID | EmployeeID | Startime, EmployeeID->EndTime  EndTime, EmployeeID->Startime |
| Employees | EmployeeID | X | EmployeeID-> FullName, Address, Level, Position, Gender, Brthday, Salary |
| Card | CardID | CustomerID, VehicleID | CardID-> CustomerID, VehicleID, RegistrantionTime, ExpiredTime, LicensePlates, PhoneNumber  VehicleID->CardID |
| Customer | CustomerID | X | CustomerID-> Fullname, Birthday, Gender |
| Vehicle | VehicleID | CustomerID | VehicleID-> CustomerID, Mode, Color |
| History | CardID, TimeIn | CardID | CardID, TimeIn-> TimeOut  CardID, TimeOut-> TimeIn |

**2.Standardize the schema**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Dependency** | **Main courses** | **Relational schema** | **3NF** |
| Startime, EmployeeID->EndTime  EndTime, EmployeeID->Startime | StartTime,  EmployeesID | **OperationRoom:** StartTime, EndTime, EmployeesID. |  |
| CardID-> CustomerID, VehicleID, RegistrantionTime, ExpiredTime, PhoneNumber.  VehicleID->CardID | CardID | **Card:** CardID, CustomerID, VehicleID, RegisttrationTimes, ExpiredTimes. |  |
| EmployeeID-> FullName, Address, Level, Position, Gender, Brthday, Salary | EmployeesID | **Employees:** EmployeesID, FullName, Address, PhoneNumber, Level, Position, Gender, Birthday, Salary |  |
| CustomerID-> Fullname, Birthday, Gender | CustomersID | **Customers:** CustomersID, Gender, Address, Fullname, Birthday |  |
| VehicleID-> CustomerID, Mode, Color | VehicleID | **Vehicle:** VehicleID, CustomerID, Mode, Color. |  |
| CardID, TimeIn-> TimeOut  CardID, TimeOut-> TimeIn | TimesIn, CardID | **History:** TimesIn, TimesOut, CardID. |  |

**3. Relational model (Mô hình quan hệ)**

****

**IV. Data Dictionaries (Từ điển dữ liệu)**

**1. Format the tables (Định dạng các bảng)**

**Table Card**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Types** | **Check** | **Key/Constraint** |
| CardID | Int |  | Primary key, not null |
| TimesRegisttration | Date |  | Not null |
| TimesExpired | Date |  | Not null |
| license plates | Nvarchar(10) |  | Not null |
| CusID | int |  | Foreign key of customer table , not null |

Câu lệnh tạo bảng

create table Card (

CardID int not null,

TimesRegisttration time not null,

TimesExpired time not null,

licenseplates nvarchar(10)not null,

CusID int FOREIGN KEY REFERENCES Customers(CusID) not null,

Primary key(CardID))

Bảng Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Types** | **Check** | **Key/Constraint** |
| CusID | Int |  | Primary key,not null |
| Fullname | Nvarchar(50) |  | Not null |
| Gender | Nvarchar (10) |  | =’Nữ’ and ‘Nam’ |
| Birthday | Date |  | Date of birth minus the current date greater than or equal to 16, not null |

Câu lệnh tạo bảng:

create table Customers (

CusID int primary key not null,

Gender nvarchar(10) CHECK ((Gender='Nữ'and Gender = 'nữ')and (Gender='Nam' and Gender='nam')) not null,

Fullname nvarchar(50) not null,

Birthday Date not null)

Alter table Customers add constraint check\_bir check((Birthday -CurDate()>='16 year'))

Bảng Employes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Types** | **Check** | | **Key/Constraint** |
| ID | Int |  | | Not null, primary key |
| Gender | Nvarchar (10) |  | | =’Nữ’ and ‘Nam’ |
| PhoneNumber | Nvarchar(20) |  | | Not null |
| Position | Nvarchar(50) |  | | Not null |
| Salary | Nvarchar(8) |  | | Not null |
| Fullname | Nvarchar(50) |  | | Not null |
| Address | Nvarchar(200) |  | | Not null |
| Level | Nvarchar(100) |  | Not null | |
| Birthday | Date |  | Date of birth minus the current date greater than or equal to 16, not null | |

Câu lệnh tạo bảng employees

create table Employees (

ID int primary key not null,

Gender nvarchar(10) check ((Gender='Nữ'and Gender = 'nữ')and (Gender='Nam' and Gender='nam')),

PhoneNumber nvarchar(20) not null,

Salary nvarchar (8) not null,

Fullname nvarchar (50) not null,

Position nvarchar(50) not null,

Address nvarchar (200) not null,

Level nvarchar (10) not null,

Birthday Date not null)

Alter table Employees add constraint check\_bir check((Birthday -CurDate()>='16 year'))

Bảng OperationRoom

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Types** | **Check** | **Key/Constraint** |
| ID | int |  | Primary key, foreign key of Employees table, not null |
| TimesIn | Time |  | Not null |
| TimesOut | Time |  | Not null |

Câu lệnh tạo bảng

create table OperationRoom(

TimeIn time not null,

TimeOut time not null,

ID int FOREIGN KEY REFERENCES Employees(ID) not null,

Primary key(ID))

Bảng Vehicle

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Types** | **Check** | **Key/Constraint** |
| ID | Nvarchar(10) |  | Primary key, not null |
| OwnerID | Int |  | Forgein key CusID của Customer table, not null |
| Color | Nvarchar(20) |  | Not null |
| model | Nvarchar(100) |  | Not null |

Câu lệnh tạo bảng:

create table Vehicle (

model nvarchar(100),

color nvarchar(20),

OwnerID int references Customers (CusID) not null,

ID nvarchar(10) not null,

Primary key (ID,OwnerID))

Bảng History

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Types** | **Check** | **Key/Constraint** |
| CardID | int |  | Foreign key of Card table, primary key, not null |
| TimesIn | Time |  | Not null |
| TimesOut | Time |  | Not null |

Câu lệnh tạo bảng

create table History(

TimesIn time not null,

TimesOut time not null,

CardID int references Card (CardID) not null,

primary key(CardID))