ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Курсовая работа

По дисциплине «Информационное обеспечение систем управления»

Выполнил: ст. гр.ВУИ-511

Милаш П.С

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[1. Цель курсового проекта 3](#_Toc90285038)

[2. Формулировка задания 3](#_Toc90285039)

[3. Исходные данные 3](#_Toc90285040)

[4. Практическая часть 4](#_Toc90285041)

[4.1. Анализ предметной области 4](#_Toc90285042)

1. Цель курсового проекта

Целью курсового проекта является изучение методов и закрепление знаний в проектировании реляционных баз данных (РБД) с примерами и заданиями для исполнения в системе управления базами данных (СУБД) Microsoft SQL Server.

1. Формулировка задания

В данном курсовом проекте ставится задача разработать РБД в СУБД Microsoft SQL Server по заданной теме . Проектирование РБД проводится с помощью метода «Сущность-связь». Проверка построенной модели РБД осуществляется с помощью метода нормализации отношений. Пояснительная записка должна содержать пункты по проектированию РБД и пункты по разработке РБД в СУБД Microsoft SQL Server: разработка скриптов на создание и заполнение РБД, разработка необходимых функций, процедур, триггеров и представлений (views).

1. Исходные данные

БД «ГАИ»

Описание предметной области: БД создается для информационного обслуживания работников ГАИ. В БД находятся автомобили, зарегистрированные по данному адресу. Некоторые из них угнаны.

Готовые запросы:

1. Выдавать информацию об автомобиле по его регистрационному знаку (марка, цвет, модель и т.д.).
2. Выдавать информацию об автовладельце по регистрационному знаку данного автомобиля.
3. Выдавать список угнанных автомобилей.
4. Выдавать список автомобилей, попавших в аварию в данный период времени.
5. Выдавать список наиболее угоняемых автомобилей по марке.
6. Практическая часть
   1. Анализ предметной области

В соответствии с предметной областью система строится с учётом следующих особенностей:

* Один автомобиль соответствует одному владельцу
* Стоимость номера зависит от количества удобств.

Выделим базовые сущности ПрО:

* Автомобиль – Регистрациооный номер, модель, цвет, дата выпуска.
* Производитель – Название.
* Модель – Производитель, название.
* Цвет – Название.
* Владелец – Автомобиль, водитель, начало владения авто, конец владения авто.
* Водитель – Имя, отчество, фамилия, пасспорт.
* Угон – Автомобиль, дата угона, угнан.
* Участник ДТП – Автомобиль, ДТП.
* ДТП – Описание, дата.

ER-диаграмма БД приведена на (см. Рисунок 1). На данной ER-диаграмме присутствуют связи один - ко - многим. Такие связи легко реализуются средствами СУБД:

|  |
| --- |
| Рисунок ER-диаграмма БД ГАИ |
|  |

* + 1. Анализ информационных задач и круга пользователей системы

Система создаётся для обслуживания следующих групп пользователей:

- Сотрудники ГАИ

Определим границы информационной поддержки пользователей:

1) Функциональные возможности:

* ведение БД (запись, чтение, модификация, удаление в архив);
* обеспечение логической непротиворечивости БД;
* обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
* реализация наиболее часто встречающихся запросов в готовом виде.

2) Готовые запросы:

* Выдавать информацию об автомобиле по его регистрационному знаку (марка, цвет, модель и т.д.).
* Выдавать информацию об автовладельце по регистрационному знаку данного автомобиля.
* Выдавать список угнанных автомобилей.
* Выдавать список автомобилей, попавших в аварию в данный период времени.
* Выдавать список наиболее угоняемых автомобилей по марке.
  1. Логическое проектирование реляционной БД

## Составление реляционных отношений

## Составление реляционных отношений

Таблица 1 – Схема отношения производитель (Manufacter)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ManufacterID | Целое число | первичный ключ |
| Название | Name | Строка (255) | обязательное поле |

Таблица 2 – Схема отношения модель (Model)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ModelID | Целое число | первичный ключ |
| Производитель | ManufacterID | Целое число | внешний ключ (Manufacter), обязательное поле |
| Название | Name | Строка (255) | Обязательное поле |

Таблица 3 – Схема отношения цвет (Color)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ColorID | Целое число | первичный ключ |
| Название | Name | Строка (255) | обязательное поле |

Таблица 4 – Схема отношения водитель (Driver)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | DriverID | Целое число | первичный ключ |
| Имя | FirstName | Строка (255) | обязательное поле |
| Отчество | SurName | Строка (255) | обязательное поле |
| Фамилия | LastName | Строка (255) | обязательное поле |
| Паспорт | PassportID | Строка (255) | обязательное поле |

Таблица 5 – Схема отношения автомобиль (Car)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | CarID | Целое число | первичный ключ |
| Регистрационный номер | RegistrationNumber | Строка (255) | обязательное поле |
| Модель | ModelID | Целое число | внешний ключ (Model), обязательное поле |
| Цвет | ColorID | Целое число | внешний ключ (Color), обязательное поле |
| Дата выпуска | ReleaseDate | Дата | обязательное поле |

Таблица 6 – Схема отношения владелец (Owner)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | OwnerID | Целое число | первичный ключ |
| Автомобиль | CarID | Целое число | внешний ключ (Car), обязательное поле |
| Водитель | DriverID | Целое число | внешний ключ (Driver), обязательное поле |
| Дата начала владения | BeginDate | Дата | обязательное поле |
| Дата конца владения | EndDate | Дата |  |

Таблица 7 – Схема отношения угон (Hijacking)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | HijackingID | Целое число | первичный ключ |
| Фильм | CarID | Целое число | внешний ключ (Film), обязательное поле |
| Цена | HijackingDate | Целое число | обязательное поле |
| Статус | Hijact | Бит | обязательное поле |

Таблица 8 – Схема отношения угон (Hijacking)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | HijackingID | Целое число | первичный ключ |
| Автомобиль | CarID | Целое число | внешний ключ (Car), обязательное поле |
| Дата угона | HijackingDate | Дата | обязательное поле |
| Угнан | Hijacking | Бит | обязательное поле |

Таблица 9 – Схема отношения ДТП (Accident)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | AccidentID | Целое число | первичный ключ |
| Описание | AccidentDescription | Строка (255) | обязательное поле |
| Дата | AccidentDate | Строка (255) | Обязательное поле |

Таблица 10 – Схема отношения участник ДТП (AccidentMember)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | AccidentMember | Целое число | первичный ключ |
| Автомобиль | CarID | Строка (255) | внешний ключ (CarID), обязательное поле |
| ДТП | AccidentID | Строка (255) | внешний ключ (Accident), обязательное поле |

## Нормализация полученных отношений

**1НФ.** Для приведения таблиц к 1НФ требуется составить таблицы (один атрибут – один столбец) и разбить сложные атрибуты на простые, а многозначные атрибуты вынести в отдельные отношения.

Все отношения находятся в 1 НФ.

**2НФ.** Все отношения находятся во 2НФ, так как для каждого из них введен уникальный ключ.

## Физическое проектирование БД

Разработка скриптов на создание базы данных и таблиц:

InitDB – создаёт БД TrafficPolice

CREATE DATABASE TrafficPolice;

InitTable – создаёт таблицы.

USE TrafficPolice;

CREATE TABLE Manufacter (

ManufacterID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

);

CREATE TABLE Model (

ModelID INT PRIMARY KEY IDENTITY(1,1),

ManufacterID INT NOT NULL,

FOREIGN KEY (ManufacterID) REFERENCES Manufacter (ManufacterID),

[Name] VARCHAR(255) NOT NULL,

);

CREATE TABLE Color (

ColorID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

);

CREATE TABLE Driver (

DriverID INT PRIMARY KEY IDENTITY(1,1),

FirstName VARCHAR(255) NOT NULL,

SurName VARCHAR(255) NOT NULL,

LastName VARCHAR(255) NOT NULL,

PassportID VARCHAR(255) NOT NULL,

);

CREATE TABLE Car (

CarID INT PRIMARY KEY IDENTITY(1,1),

RegistrationNumber VARCHAR(255) NOT NULL,

ModelID INT NOT NULL,

FOREIGN KEY (ModelID) REFERENCES Model (ModelID),

ColorID INT NOT NULL,

FOREIGN KEY (ColorID) REFERENCES Color (ColorID),

ReleaseDate DATE NOT NULL,

);

CREATE TABLE [Owner] (

[OwnerID] INT PRIMARY KEY IDENTITY(1,1),

CarID INT NOT NULL,

FOREIGN KEY (CarID) REFERENCES Car (CarID),

DriverID INT NOT NULL,

FOREIGN KEY (DriverID) REFERENCES Driver (DriverID),

BeginDate DATE NOT NULL,

EndDate DATE,

);

CREATE TABLE Hijacking (

HijackingID INT PRIMARY KEY IDENTITY(1,1),

CarID INT NOT NULL,

FOREIGN KEY (CarID) REFERENCES Car (CarID),

HijackingDate DATE NOT NULL,

Hijack BIT NOT NULL,

);

CREATE TABLE Accident (

AccidentID INT PRIMARY KEY IDENTITY(1,1),

AccidentDescription VARCHAR(255) NOT NULL,

AccidentDate VARCHAR(255) NOT NULL,

);

CREATE TABLE AccidentMember (

AccidentMemberID INT PRIMARY KEY IDENTITY(1,1),

CarID INT NOT NULL,

FOREIGN KEY (CarID) REFERENCES Car (CarID),

AccidentID INT NOT NULL,

FOREIGN KEY (AccidentID) REFERENCES Accident (AccidentID),

);

Разработка скриптов на добавление данных в таблицы

-- Insert Manufacters

INSERT INTO Manufacter VALUES ('Porche');

INSERT INTO Manufacter VALUES ('BMW');

-- Insert Models

INSERT INTO Model VALUES (1, 'Cayenne');

INSERT INTO Model VALUES (1, 'Macan');

INSERT INTO Model VALUES (2, 'x5');

-- Insert Colors

INSERT INTO Color VALUES ('Yellow');

INSERT INTO Color VALUES ('Gray');

INSERT INTO Color VALUES ('Black');

-- Insert Drivers

INSERT INTO Driver VALUES ('Petr', 'Vasilevich', 'Lovin', '1712 252037');

INSERT INTO Driver VALUES ('Andrey', 'Sergeevich', 'Petrov', '3213 213321');

--Insert Cars

INSERT INTO Car VALUES ('ЕК323Х', 1, 1, '1999-04-12');

INSERT INTO Car VALUES ('ОК374Т', 3, 2, '2000-10-15');

-- Insert Owners

INSERT INTO [Owner] VALUES (1,2,'1999-04-19',NULL);

INSERT INTO [Owner] VALUES (2,1,'2003-10-11', '2009-10-19');

INSERT INTO [Owner] VALUES (2,2,'2009-10-20', NULL);

--Insert Hijacking

INSERT INTO Hijacking VALUES (1,'2003-10-24', 1);

--Insert Hijacking

INSERT INTO Accident VALUES ('Два автомобиля столкнулись', '2005-10-11');

INSERT INTO Accident VALUES ('Два автомобиля столкнулись', '2010-10-11');

--Insert AccidentMembers

INSERT INTO AccidentMember VALUES (1, 1);

INSERT INTO AccidentMember VALUES (2, 1);

INSERT INTO AccidentMember VALUES (1, 2);

INSERT INTO AccidentMember VALUES (2, 2);

--1. Выдавать информацию об автомобиле по его регистрационному знаку (марка, цвет, модель и т.д.).  
SELECT a.RegistrationNumber, b.[Name] as ModelName, d.[Name] as Manufacter, c.[Name] as Color FROM Car as a  
JOIN Model as b ON a.ModelID = b.ModelID  
JOIN Color as c ON a.ColorID = c.ColorID  
JOIN Manufacter as d ON b.ManufacterID = d.ManufacterID  
WHERE RegistrationNumber='ЕК323Х'



Рисунок 3 ‑ Результат выполнения запроса

--2. Выдавать информацию об автовладельце по регистрационному знаку данного автомобиля.  
SELECT c.\* FROM [Owner] as a  
JOIN Car as b ON a.CarID = b.CarID  
JOIN Driver as c ON a.DriverID = c.DriverID  
WHERE RegistrationNumber='ЕК323Х' AND a.EndDate IS NULL

  
Рисунок 4 ‑ Результат выполнения запроса

--3. Выдавать список угнанных автомобилей.  
SELECT b.\*, a.HijackingDate, a.Hijack FROM Hijacking as a  
JOIN Car as b ON a.CarID = b.CarID

  
Рисунок 5 ‑ Результат выполнения запроса

--4. Выдавать список автомобилей, попавших в аварию в данный период времени.  
SELECT a.AccidentID, a.AccidentDescription, a.AccidentDate, c.RegistrationNumber FROM Accident as a  
JOIN AccidentMember as b ON a.AccidentID = b.AccidentID  
JOIN Car as c ON b.CarID = c.CarID  
WHERE YEAR(a.AccidentDate) < 2007

  
Рисунок 6 ‑ Результат выполнения запроса

--5. Выдавать список наиболее угоняемых автомобилей по марке.  
SELECT b.ModelID, COUNT(\*) as HijackingCount FROM Hijacking as a  
JOIN Car as b ON a.CarID = b.CarID  
GROUP BY b.ModelID

  
Рисунок 7 ‑ Результат выполнения запроса

## Вывод

Изучил методы и закрепил знания в проектировании реляционных баз данных (РБД) с примерами и заданиями для исполнения в системе управления базами данных (СУБД) Microsoft SQL Server.