

Московский государственный технический университет им.Н.Э.Баумана
кафедра "Системы обработки информации и управления"



Постреляционные базы данных

к Домашнее задание 1

Домашнее задание «Построение постреляционных моделей баз данных»
по дисциплине «Постреляционные базы данных»

Инструктор : Мария Валерьевна

Email:2623859464@qq.com

Студент: Ван Чаочао

группа ИУ5И-22М

2022/03/21

Цель работы:

Освоить технологии построения моделей базы данных: ER, объектной, объектно-реляционной и полуструктурированных данных. Сравнить возможности этих моделей с возможностями реляционной модели.

Исходные данные для выполнения ДЗ:

Описание предметной области по варианту студента.

Пункты задания для выполнения:

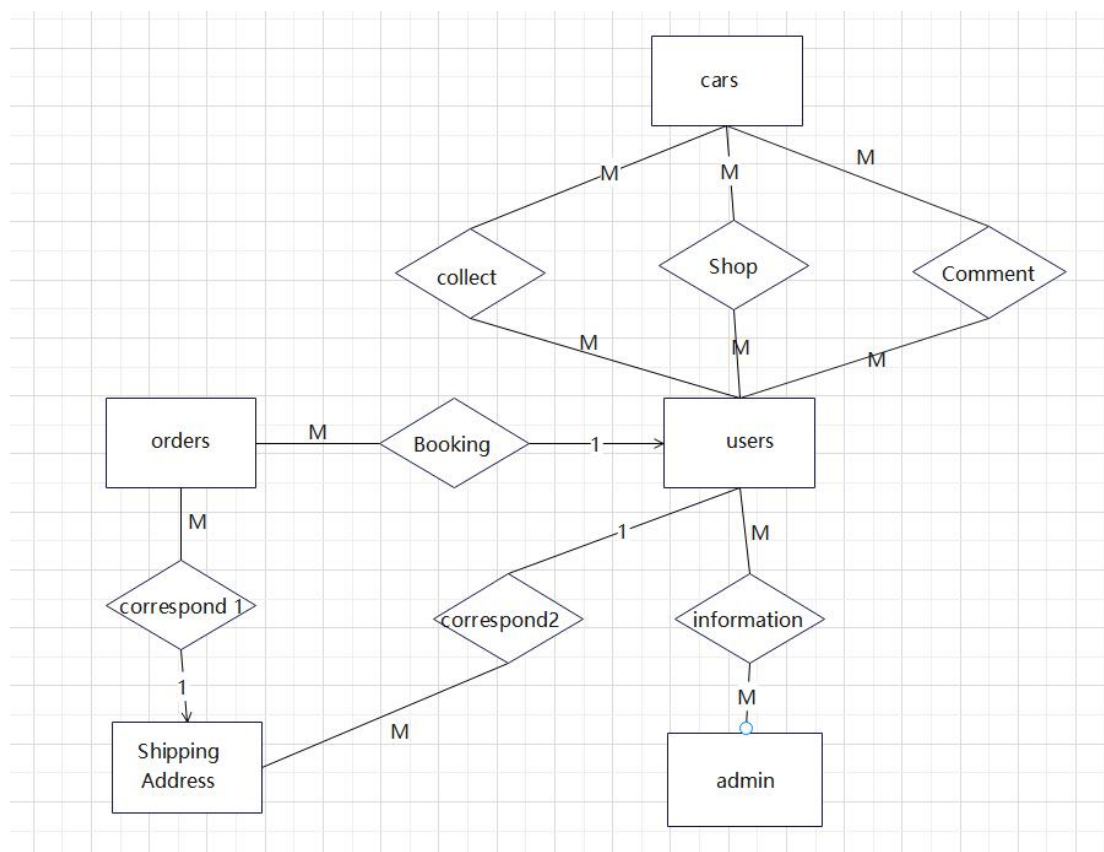
1. На основе описания предметной области выделить сущности (от 4-х до 6-ти), их атрибуты и связи. Указать ключевые атрибуты и типы связей. Построить ER-модель. Продемонстрировать в модели связи один-ко-многим (1-M) и многие-ко-многим (M-M), связь типа ISA.

1.1 сущности

User	Cars	Order	
<u>UserID</u> Username Password Real Name Gender Date of Birth Email Phone City Address	<u>Cars Number</u> Car Name Car Description Price Stock Sold	<u>Order Number</u> User ID (FK) Consignee Conspnum shipping method shipping address total order amount order status	

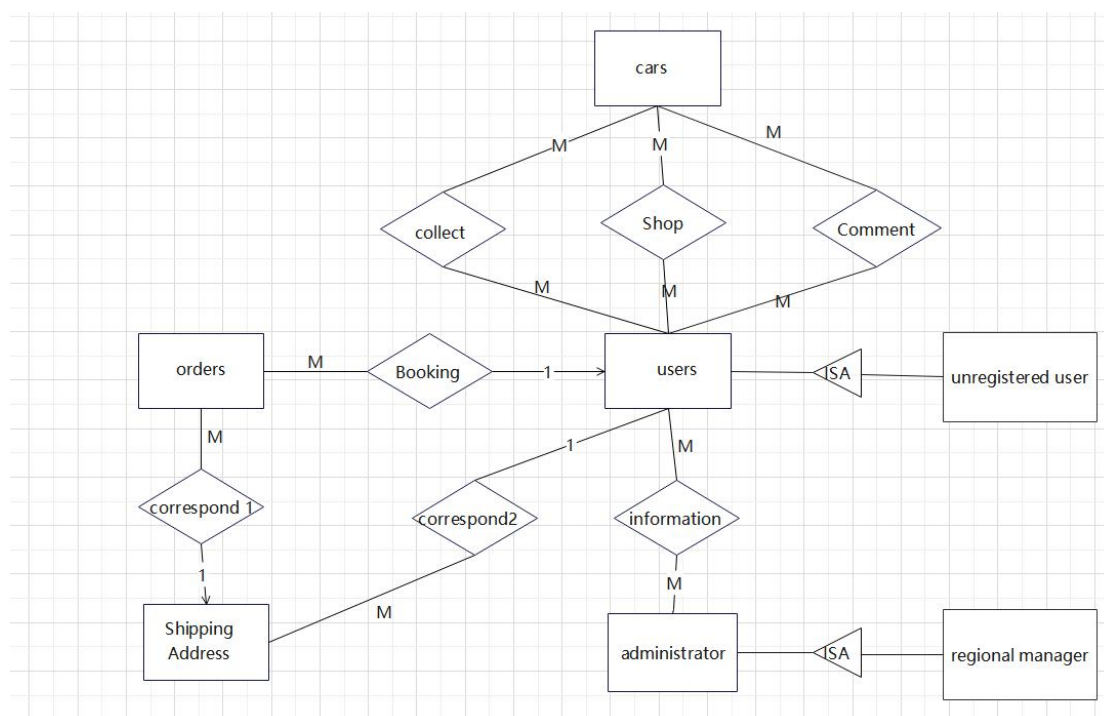
ShipAddr	Admin	
<u>User ID</u> Address Postal Code phonenumber	<u>Admin ID</u> Password Permissions Contact Properties	

1.2 ER-модель



фигура 1: Веб-сервис по продажам автомобилей ER

1.3 связь типа ISA.



2. На основе описания предметной области построить объектную модель БД на языке ODL. Привести описания классов, их свойств, методов и связей. Указать ключи и экстененты. Методы классов следует выделять так, чтобы их можно было использовать в типовых запросах вместо вложенных подзапросов. Продемонстрировать наследование, интерфейсы, сложные типы данных (коллекции и структуры).

```
class Users()
{
attribute integer UserID;
attribute string Username;
attribute varchar(20) Password;
attribute string RealName;
attribute string Gender;
attribute Date DateOfBirth;
attribute enum Info {Email,Phone,City,Address} type;
relationship set<Admin> information_by
    inverse Admin::information;
relationship set<Cars> collect
    inverse Cars::collect_by;
relationship set<Cars> shop
    inverse Cars::shop_by;
relationship set<Cars> comment
    inverse Cars::comment_by;
relationship set<Order> booking
    inverse Order::booking_by;
relationship set<ShipAddr> correspond2
    inverse ShipAddr::correspond2_by;
key(UserID)
};
```

```
class Cars()
{
attribute integer CarsID;
attribute string CarName;
attribute string CarDescription;
attribute integer Price;
attribute integer Stock;
attribute integer Sold;
relationship set<Users> collect_by
    inverse Users::collect;
relationship set<Users> shop_by
    inverse Users::shop;
relationship set<Users> comment_by
    inverse Users::comment;
key(CarsID)
```

```
};
```

```
class Order()  
{  
attribute integer OrderNumber;  
attribute integer UserID;  
attribute string Consignee;  
attribute varchar(11) conspnum;  
attribute varchar(50) shipmethod;  
attribute varchar(50) shipaddress;  
attribute varchar(100) orderstatus;  
relationship set<Users> booking_by  
    inverse Users::booking;  
relationship set<ShipAddr> correspond2_by  
    inverse ShipAddr::correspond2;  
key(OrderNumber)  
};
```

```
class ShipAddr()  
{  
attribute integer UserID;  
attribute varchar(50) Address;  
attribute varchar(11) phonenumber;  
relationship set<Users> correspond2_by  
    inverse Users::correspond2;  
relationship set<Order> correspond2  
    inverse Order::correspond2_by;  
key(UserID)  
};
```

```
class Admin()  
{  
attribute integer AdminID;  
attribute varchar(20) Password;  
attribute enum Permissions {AuditInfo,ManageOrders} type;  
relationship set<Users> information  
    inverse Users::information_by;  
key(AdminID)  
};
```

3. На основе объектной модели построить реляционную модель. Привести перечень отношений с указанием их атрибутов и ключей. Составить на языке SQL описание схемы базы данных (создание таблиц, задание первичных и внешних ключей, ограничений и т.д.). Продемонстрировать реализацию отношения наследования тремя различными методами.
create type Information as enum('Email','Phone','City','Address');

3.1 : Создать таблицу

```
create type Information as enum('Email','Phone','City','Address');
```

```
CREATE TABLE Users (  
    UserID integer primary key,  
    Username varchar(20) not null,  
    Passwordd varchar(20) not null,  
    RealName varchar(20) not null,  
    Gender varchar(10) not null,  
    DateOfBirth Date,  
    Info Information  
);
```

```
CREATE TABLE Cars (  
    CarsID integer primary key,  
    CarName varchar(20) not null,  
    CarDescription varchar(20) not null,  
    Price integer not null,  
    Stock integer not null,  
    Sold integer not null  
);
```

```
CREATE TABLE Orderr (  
    OrderNumber integer primary key,  
    UserID integer not null,  
    Consignee varchar(20) not null,  
    conspnum varchar(11) not null,  
    shipmethod varchar(50) not null,  
    shipaddress varchar(50) not null,  
    orderstatus varchar(100) not null  
);
```

```
CREATE TABLE ShipAddr (  
    UserID integer primary key,  
    Address varchar(50) not null,  
    phonenumber varchar(11) not null  
);
```

```
create type Per as enum('AuditInfo','ManageOrders');
```

```
CREATE TABLE Adminn (  
    AdminID integer primary key,  
    Passwordd varchar(50) not null,  
    Permissions Per  
);
```

Входные данные

	adminid [PK] integer	passwordd character varying (50)	permissions per
1	1	123456789	admin
2	2	987654321	admin
3	3	456123789	admin

	carsid [PK] integer	carname character varying (20)	cardescription character varying (20)	price integer	stock integer	sold integer
1	1	BMW	red	3000000	25	5
2	2	Geely	black	1800000	30	6
3	3	Benz	red	5000000	15	3

	ordernumber [PK] integer	userid integer	consignee character varying (20)	consprum character varying (11)	shipmethod character varying (50)	shipaddress character varying (50)	orderstatus character varying (100)
1	1	2	Чаочао	+79269525693	Доставка	bamstu	ожидающая доставки
2	2	3	wcc	+79568268569	Доставка	bamstu	Заканчивать
3	3	1	wei	+79859632635	Доставка	Красный квадрат	Заканчивать

	userid [PK] integer	address character varying (50)	phonenumber character varying (11)
1	1	bamstu	+79269525693
2	2	bamstu	+79568268569
3	3	красиный квадрат	+79985632635

	userid [PK] integer	username character varying (20)	passwordd character varying (20)	realname character varying (20)	gender character varying (10)	dateofbirth date	info information
1	1	cc	123456789	Ван чаочао	man	1999-08-28	[null]
2	2	han	987654321	wang wei	man	1980-09-06	[null]
3	3	guan	456987321	zhou guanyu	woman	1984-08-12	[null]

Отношение booking:Users - Orders 1-M

userid	ordernumber
2	1
3	2
1	3

Отношение **collect, shop, comment** : users-cars M-M

userid	CarsID
1	2
2	3
3	1
2	3

Отношение **correspond1** : orders-shipping address M-1

ordernumber	shipaddress
1	bamstu
2	bamstu
3	квадрат

Для преобразования связи **ISA** на модели сущность–связь к реляционной модели может быть использован один из трех подходов: сущностный, объектный, пустых значений.

1- При сущностном подходе каждую сущность преобразуют в отношение, содержащее собственные атрибуты и ключи базовой сущности.

User(User ID Username Password Real Name Gender DateOfBirth info)

Order(OrderID User ID (FK) Consignee Conspnum shipmethod shipaddress order status)

2. При объектном подходе для каждой производной сущности введем отношение, включающее все атрибуты как базовой сущности, так и ее

собственные. Тогда получим схему отношений:

User(User ID Username Password Real Name Gender DateOfBirth info)

Order(OrderID User ID (FK) Consignee Conspnum shipmethod shipaddress
order status)

3. При подходе пустых значений создадим одну сущность, содержащую атрибуты базовой и всех производных сущностей. Если при заполнении таблицы данными атрибут сущности отсутствует, запишем вместо его значения пустое значение NULL

Admin(adminID password minissons)

4. На основе описания предметной области построить объектно-реляционную модель. Привести описания типов данных UDT с указанием их свойств и методов. На основе пользовательских типов составить схемы отношения. Для каждого отношения указать ключи, ссылочный атрибут, ограничения и т.д. Задать связи между отношениями. Для любых двух пользовательских типов задать правила сравнения: 1) на равенство и полное, 2) по-элементное и через функцию. Привести описание функций сравнения. Продемонстрировать наследование, ссылки и сложные типы данных. Принцип определения методов UDT соответствует принципам определения методов класса.

создавать пользовательские типы данных

```
CREATE TYPE Carssales AS
(
stock Integer,
sold Integer
)
method getCarssales() RETURNS VARCHAR(100);

CREATE method getCarssales () RETURNS VARCHAR(100)
FOR Carssales
BEGIN
RETURN "sto:" || SELF. stock || "sol:" || SELF.sold
END;
```

```
-----
CREATE TYPE Info AS
(
Email VARCHAR(100);
Address VARCHAR(100)
)
method getInfo() RETURNS VARCHAR(250);
CREATE method getInfo () RETURNS VARCHAR(100)
FOR Info
BEGIN
RETURN "Ema:" || SELF. Email || "Adr:" || SELF. Address
END;
```

Для любых двух пользовательских типов задать правила сравнения:

- 1) на равенство и полное,**
- 2) по-элементное и через функцию. Привести описание функций сравнения.**

Сначала создадим поэлементное сравнение на равенство для объектов типа booking:

```
CREATE ORDERING FOR Carssales EQUALS ONLY BY STATE
```

booking:

```
CREATE OREDERING Carssales ORDER FULL BY RELATIVE WITH fun1
```

```
CREATE FUNCTION Fun1 (IN c1 Carssales, IN c2 Carssales)
RETURNS integer
IF C1.length()<C2. length() THEN RETURN (-1)
ELSEIF C1.length()>C2.length() THEN RETURN (1)
ELSEIF C1.stock()<C2. stock () THEN RETURN (-1)
ELSEIF C1.stock()>C2. stock () THEN RETURN (1)
ELSEIF C1.sold()<C2. sold () THEN RETURN (-1)
```

```
ELSEIF C1. result ()>C2. result () THEN RETURN (1)
ELSE RETURN(0)
ENDIF;
```

Info:

```
CREATE ORDERING FOR INFO EQUALS ONLY BY STATE
```

Info:

```
CREATE OREDERING Info ORDER FULL BY RELATIVE WITH fun2
```

```
CREATE FUNCTION Fun2 (IN F1 Info, IN F2 Info)
```

```
RETURNS integer
```

```
IF F1.length()<F2. length() THEN RETURN (-1)
```

```
ELSEIF F1.length()>F2.length() THEN RETURN (1)
```

```
ELSEIF F1.Email.length()<F2. Email.length() THEN RETURN (-1)
```

```
ELSEIF F1. Email.length()>F2. Email.length() THEN RETURN (1)
```

```
ELSEIF F1. Address.length()<F2. Address.length() THEN RETURN (-1)
```

```
ELSEIF F1. Address.length ()>F2. Address.length() THEN RETURN (1)
```

```
ELSE RETURN(0)
```

```
ENDIF;
```

Продемонстрировать наследование, ссылки и сложные типы данных. Принцип определения методов UDT соответствует принципам определения методов класса.

```
CREATE TYPE user AS
```

```
(
```

```
userID Integer,
```

```
userName varchar(100),
```

```
Phone Integer,
```

```
info varchar(100)
```

```
);
```

```
CREATE TYPE order AS
```

```
(
```

```
orderID Integer,
```

```
userID Integer,
```

```
Name varchar(100),
```

```
Address varchar(100)
```

```
Phone Integer,
```

```
inf REF(Info) SCOPE inf
```

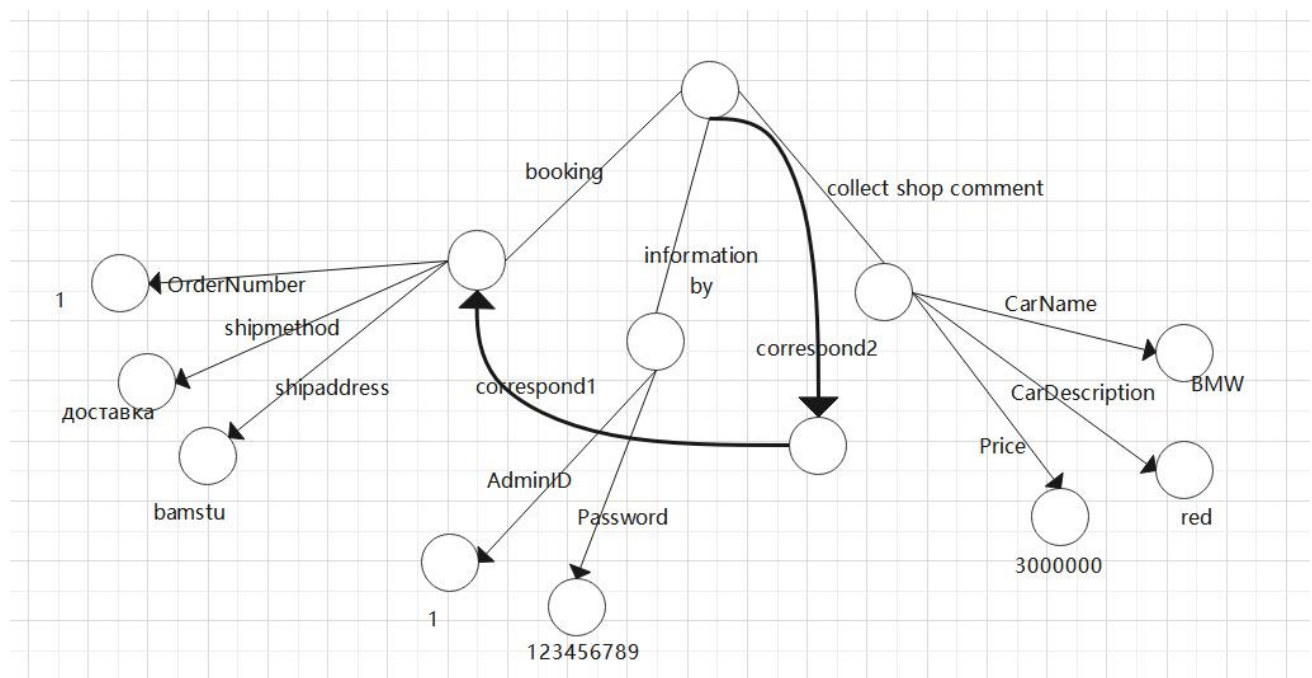
```
);
```

```
CREATE TYPE Admin AS
(
Id Integer,
W_ID Integer,
Name varchar(100),
Phone Integer,
inf REF(Info) SCOPE inf
);
```

создавать типы, наследуются от них

```
CREATE TYPE unregisteruser under user AS
(
Id Integer,
Name varchar(100),
Phone Integer,
Date varchar(100)
)
```

5. На основе описания предметной области построить модель полуструктурированных данных. Привести пример графа полуструктурированных данных, соответствующий ему XML-документ и DTD-определение.



Граф полуструктурированных данных

XML-документ

```
<?xml version="1.0"?>
- <r>
  - <user>
    <UserID>1</UserID>
    <Username>chaochao </Username>
    <Phone>18801035010</Phone>
  - <cars>
    <CarsID>3</CarsID>
    <CarName>Bentley </CarName>
  </cars>
</user>
- <user>
  <UserID>2</UserID>
  <Username>wei </Username>
  <Phone>79269565693</Phone>
  - <cars>
    <CarsID>1</CarsID>
    <CarName>BMW </CarName>
  </cars>
</user>
- <user>
  <UserID>3</UserID>
  <Username>cat </Username>
  <Phone city="london">79269815682</Phone>
  - <cars>
    <CarsID>2</CarsID>
    <CarName>Geely </CarName>
  </cars>
</user>
- <user>
  <UserID>4</UserID>
  <Username>flower </Username>
  <Phone city="moc">79279853569</Phone>
  - <cars>
    <CarsID>4</CarsID>
    <CarName>JEEP </CarName>
  </cars>
</user>
</r>
```

Методическая литература:

1. Постреляционные модели данных и языки запросов / Виноградов В.И., Виноградова М.В. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2017. - [100] с. - Режим доступа: <http://ebooks.bmstu.ru/catalog/254/book1615.html> (дата обращения: 08.12.2017). - ISBN 978-5-7038-4283-6.
2. Гарсиа-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс: Пер. с англ. – М.: Издательский дом «Вильямс», 2004 г. – 1088 с.

