

Курсовая работа по дисциплине:
«Распределенные информационные
системы»

На тему: «Автоматизация работы парковки» (вариант №3).

Выполнил: студент гр. 4510
Бариев А.Р.

Проверил:

Дата _____
Оценка _____

2011 г.

Содержание

Содержание.....	2
Введение.....	3
Системный анализ предметной области.....	4
Объекты предметной области:.....	4
Ограничения и особенности системы:.....	4
Состав групп пользователей системы:.....	4
Инфологическая модель предметной области.....	6
Составление перечня атрибутов предметной области.....	6
Агрегация атрибутов в сущности:.....	6
Определение первичных ключей.....	7
Установление связей между сущностями.....	7
Нормализация сущностей.....	7
Графическое изображение инфологической модели предметной области.....	8
Проектирование концептуальной модели предметной области.....	9
Преобразование сущностей в отношения.....	9
Определение внешних ключей.....	10
Графическое изображение концептуальной модели предметной области.....	11
Руководство пользователя.....	12
Запросы инициализации базы данных.....	15
Создание.....	15
Заполнение (сервер demo_on).....	17
Заполнение (сервер demo_on2).....	18
Заключение.....	20

Введение

В повседневной жизни все большее внимание уделяется автоматизированию в различных сферах деятельности. Это обусловлено тем, что использование автоматизации позволяет в значительной мере упростить решения задач, сократить время их выполнения, а так же упростить работу пользователя.

В свою очередь, скорость выполнения и эффективность автоматизированных систем зависит от качества и правильности их реализации. Для удобства манипулирования данными используются системы управления базами данных (СУБД). Наибольшее распространение в настоящее время получили реляционные базы данных и стандартизированный язык запросов SQL (Structured Query Language).

В данной работе спроектирована система автоматизации работы парковки, необходимая для эффективной работы парковки, учёта затраченных ресурсов и доходов, получаемых в результате её деятельности.

Системный анализ предметной области

Основной задачей оператора, работающего с данной системой является обслуживание клиентов: постановка автомобиля на парковку, выдача автомобиля по квитанции.

Объекты предметной области:

1. Клиент

- информация о клиенте должна содержать его имя, фамилию, отчество и номер паспорта
- необходимо хранить информацию о его машине: марку, цвет и гос. номер

2. Машина клиента

- марка, цвет и гос. номер

3. Парковка(действие)

- документ, подтверждающий получение фирмой машины и позволяющей забрать клиенту машину
- время парковки

4. Парковочное место

- необходимо хранить все парковочные места и их состояние(занят/не занят)

5. Платежи

- для хранения информации о совершенных платежах клиентами
- время совершения платежа

Ограничения и особенности системы:

- при парковке автомобиля необходимо изменять состояние парковочного места
- парковочное место может быть свободно, либо занято одним автомобилем
- при совершении платежа необходимо автоматически заполнять время отдачи автомобиля клиенту
- необходимо создать базу данных марок автомобилей

Состав групп пользователей системы:

В системе имеется одна группа пользователей — операторы ПК, осуществляющие работу с клиентами с помощью проектируемой системы.

Алгоритм работы оператора:

В случае принятия автомобиля:

1. Зарегистрировать клиента, если не имеется записи о его регистрации
2. Зарегистрировать марку машины, если информация о марке машины не

содержится в системе

3. Выбрать парковочное место для клиента
4. Выдать талон о приеме автомобиля клиенту

В случае выдачи автомобиля:

1. Рассчитать и распечатать документ, содержащий информацию о дате приема автомобиля, дату выдачи и стоимость парковки
2. Принять платеж пользователя
3. Выдать автомобиль клиенту

Инфологическая модель предметной области

Составление перечня атрибутов предметной области

Таблица 1. Перечень атрибутов:

Обозначение	Атрибут	Примечание
X1	Фамилия	
X2	Имя	
X3	Отчество	
X4	Номер паспорта	
X5	ID клиента	
X6	Марка машины	
X8	Номер машины	
X9	Регион машины	
X10	ID машины	
X11	Состояние парковочного места	
X12	Описание парковочного места	
X13	Номер парковочного места	
X14	ID парковочного места	
X15	Время поставки на парковку	
X16	Время снятия с парковки	
X17	ID парковки	
X18	Сумма платежа	
X19	ID платежа	
X20	Стоимость часа парковки	
X21	ID тарифа	

Агрегация атрибутов в сущности:

Таблица 2. Перечень сущностей:

Y1	Клиент	X1,X2,X3,X4,X5
Y2	Машина	X6,X8,X9,X10
Y3	Парковочное место	X11,X12,X13,X14
Y4	Парковка	X15,X16,X17,X5,X10,X14
Y5	Платеж	X18,X19,X17,X21
Y6	Тариф	X20,X21

Определение первичных ключей

Таблица 3. Перечень первичных ключей сущностей:

Сущность	Первичный ключ
Клиент	ID клиента
Машина	ID машины
Парковочное место	ID парковочного места
Парковка	ID парковки
Платеж	ID платежа
Тариф	ID тарифа

Установление связей между сущностями

Таблица 4. Связи между сущностями:

Сущность 1	Сущность 2	Тип связи	Обязательность
Клиент	Парковка	1 :- M	1 :- 0
Машина	Парковка	1 :- M	1 :- 0
Парковочное место	Парковка	1 :- M	1 :- 0
Парковка	Платеж	1 :- 1	1 :- 0
Платеж	Тариф	M :- 1	1 :- 1

Нормализация сущностей

Сущность находится в первой нормальной форме, если все ее атрибуты можно рассматривать как неделимые элементы.

Пусть – два атрибута некоторой сущности. функционально зависит от , если любому значению соответствует не более одного значения Y. Функциональная зависимость обозначается

Сущность находится во второй нормальной форме, если она находится в первой нормальной форме и каждый неключевой атрибут функционально зависит от ключа.

Пусть – три атрибута некоторой сущности и имеются функциональные зависимости:

причем отсутствуют функциональные зависимости Y от Z: $Z \rightarrow Y$ или тогда говорят, что Z транзитивно зависит от X.

Сущность находится в третьей нормальной форме, если она находится во второй нормальной форме и отсутствует транзитивная зависимость каждого не ключевого атрибута от первичного ключа.

Ни один из атрибутов X1-X20 не является составным, значит все сущности

находятся в первой нормальной форме.

Все сущности находятся во второй нормальной форме, так как они находятся в первой нормальной форме и не ключевые атрибуты функционально зависят от ключей.

Все сущности находятся в третьей нормальной форме, так как они находятся во второй нормальной форме и отсутствует транзитивная зависимость каждого не ключевого атрибута от первичного ключа.

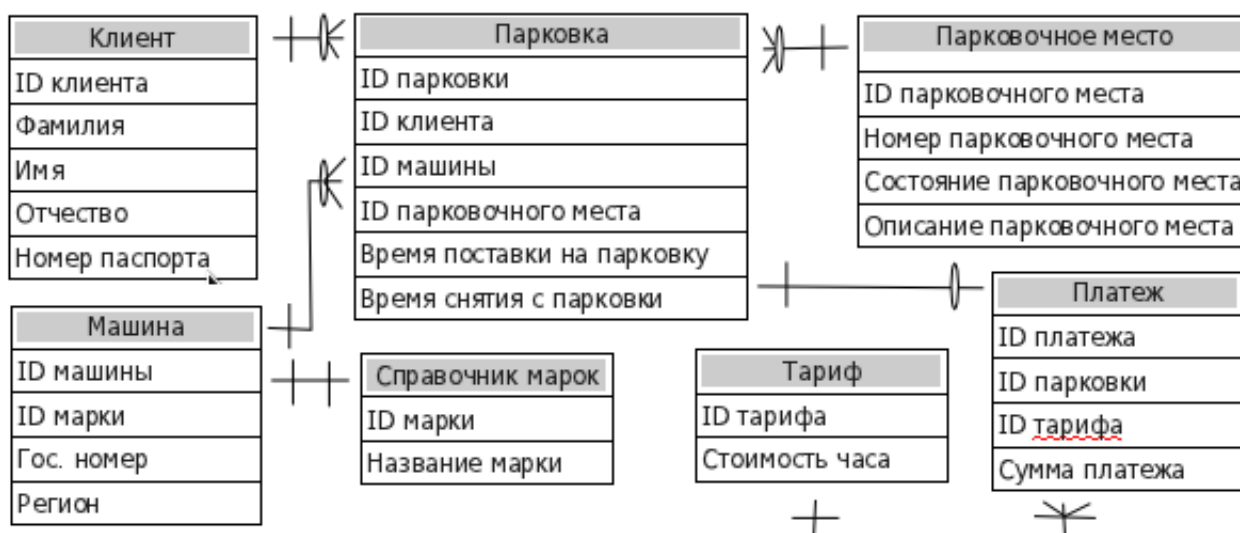
Оформим марку машин в справочник марок, в результате получим следующие две сущности:

Машина
ID машины
ID марки
Цвет
Гос. номер

Справочник марок
ID марки
Название марки

Сущность справочник марок будет иметь первичным ключом поле «ID марки», связь сущностей «Машина-Марка» будет иметь тип «М :- 1, 0 :- 1».

Графическое изображение инфологической модели предметной области



Проектирование концептуальной модели предметной области

Преобразование сущностей в отношения

Таблица «Клиент»

Клиент
ID клиента
Фамилия
Имя
Отчество
Номер паспорта

Client		
id	integer	Not null
middlename	char(30)	Not null
name	char(30)	Not null
surname	char(30)	Not null
passp_number	char(11)	Not null

Таблица «Машина»

Машина
ID машины
ID марки
Гос. номер
Регион

Car		
id	integer	Not null
mark_id	char(10)	Not null
st_number	char(10)	Not null
Region	char(15)	Not null

Таблица «Парковочное место»

Парковочное место
ID парковочного места
Номер парковочного места
Состояние парковочного места
Описание парковочного места

Parking_place		
id	integer	Not null
number	char(10)	Not null
state	boolean	Not null
description	char(100)	null

Таблица «Парковка»

Парковка
ID парковки
ID клиента
ID машины
ID парковочного места
Время поставки на парковку

Parking		
id	integer	Not null
client_id	integer	Not null
car_id	integer	Not null
parking_place_id	integer	Not null
put_date	datetime	Not null

Время снятия с парковки

pickup_date datetime null

Таблица «Платеж»

Платеж
ID платежа
ID парковки
ID тарифа
Сумма платежа

Payment		
id	integer	Not null
parking_id	integer	Not null
rate_id	integer	Not null
cost	integer	Not null

Таблица «Тариф»

Тариф
ID тарифа
Стоимость часа

Rate		
id	integer	Not null
h_cost	integer	Not null

Таблица «Справочник марок»

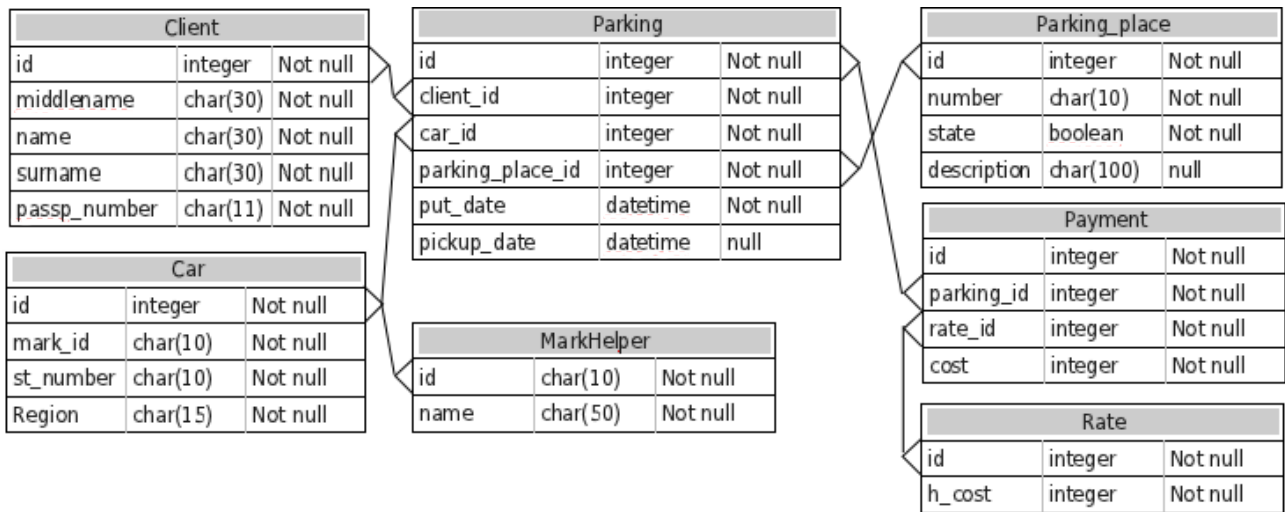
Справочник марок
ID марки
Название марки

MarkHelper		
id	char(10)	Not null
name	char(50)	Not null

Определение внешних ключей

Дочернее отношение	Внешний ключ	Тип	Null	Ссылка
Client	Parking	integer	Not null	id - client_id
Car	Parking	integer	Not null	id - car_id
Parking_place	Parking	integer	Not null	id - parking_place_id
Parking	Payment	integer	Not null	id - parking_id
Rate	Payment	integer	Not null	id - rate_id
MarkHelper	Car	integer	Not null	id - mark_id

Графическое изображение концептуальной модели предметной области



Руководство пользователя

Главное окно программы:

Курсовая работа

Таблицы Отчеты Распределенный запрос

Клиенты

Фильтрация Поиск

Фамилия	Имя	Отчество	Номер паспорта	Сервер
Демидов	Сергей	Владимирович	7245 512351	informix_server1
Киселев	Алексей	Владимирович	1242 612344	informix_server1
Павлова	Татьяна	Витальевна	4512 925612	informix_server1
Хромова	Ирина	Анатольевна	7223 034561	informix_server1

Добавить Изменить Удалить

В программе имеется меню для управления отображением таблиц, меню для генерации отчетов и меню распределенного запроса:

Таблицы Отчеты Распределенный запрос

Клиенты
Машины
Парковочные места
Парковка
Платежи
Справочник машин
Тарифы

Отчет о парковочных местах
Отчет о доходах
Квитанция об оплате
Справка о постановке на парковку

Распределенный запрос
Распределенный запрос 1

При выборе таблицы из списка таблиц происходит переключение отображаемой таблицы на выбранную. В программе так же присутствует возможность фильтрации и поиска значений. Для фильтрации необходимо нажать соответствующую кнопку и выбрать фильтруемое поле:

Затем, следует выбрать параметры фильтрации и нажать кнопку Filter. Записи таблицы будут отфильтрованы, однако при любом изменении в таблице фильтр будет сброшен.

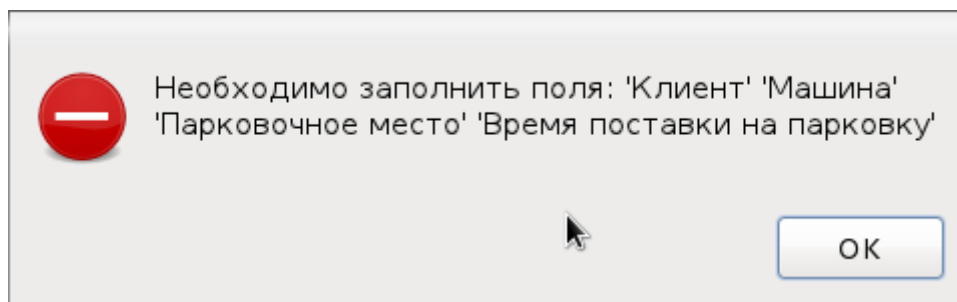
Для поиска так же имеется соответствующая кнопка, расположенная под названием текущей таблицы:

Для поиска следует выбрать поле поиска, выбрать значение и нажать кнопку Find. Для последующего поиска при нахождении нескольких значений используется та же кнопка. При окончании поиска выводится соответствующее уведомление.

Панели фильтрации и поиска можно скрыть нажав на кнопки «Фильтрация» и «Поиск» соответственно.

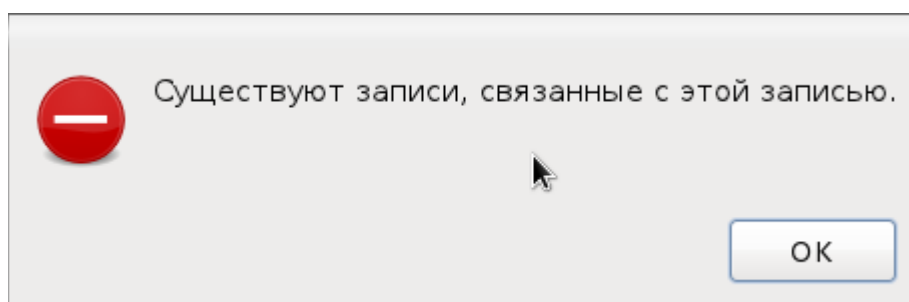
Для добавления записи используется кнопка «Добавить» расположенная внизу программы. При ее нажатии вызывается меню заполнения записи:

Требуется заполнить необходимые поля и нажать кнопку ОК либо отменить действие. При отсутствии заполнения необходимых полей будет выведено предупреждение:

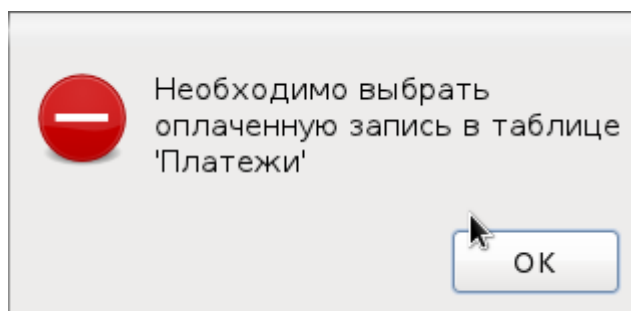


Для изменения записи используется аналогичная кнопка и окно заполнения.

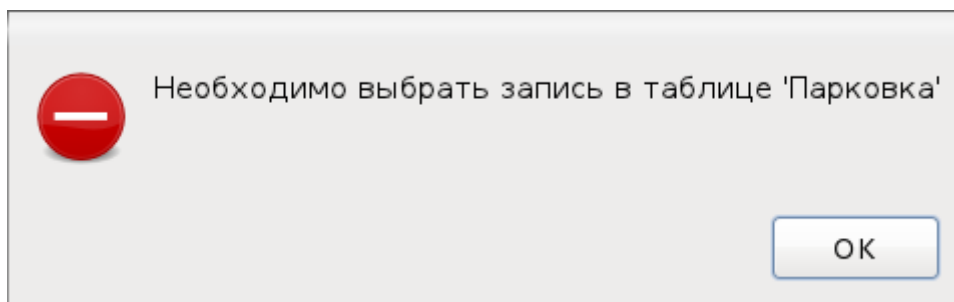
Кнопка «удалить» используется для удаления записей из таблицы. Для этого следует выбрать запись и нажать кнопку. При наличии связанных записей из других таблиц, ссылающихся на эту запись будет выведено сообщение:



Для печати отчетов «Отчет о доходах» и «Отчет о парковочных местах» достаточно нажать соответствующие кнопки. Отчет будет сгенерирован и выведен в html формат и открыт браузером для просмотра. Для печати отчета «Квитанция об оплате» необходимо выбрать запись из таблицы «Платежи», в противном случае будет выведено сообщение об ошибке:



Для печати отчета «Справка о поставке на парковку» необходимо выбрать запись из таблицы «Парковка» и нажать кнопку печати отчета, в противном случае будет выведено сообщение:



Для просмотра распределенного запроса предназначена соответствующая кнопка меню.

Запросы инициализации базы данных

Создание

/* CREATION */

```
CREATE TABLE parking (id INTEGER NOT NULL,client_id INTEGER NOT NULL,car_id  
INTEGER NOT NULL,parking_place_id INTEGER NOT NULL,put_date DATETIME YEAR TO  
SECOND NOT NULL, pickup_date DATETIME YEAR TO SECOND, PRIMARY KEY(id));
```

```
CREATE TABLE parking_place (id INTEGER NOT NULL,number CHAR(10) NOT  
NULL,state BOOLEAN NOT NULL,description CHAR(100),PRIMARY KEY(id));
```

```
CREATE TABLE car (id INTEGER NOT NULL,mark_id CHAR(10) NOT NULL,st_number  
CHAR(10) NOT NULL,region CHAR(15) NOT NULL,PRIMARY KEY(id));
```

```
CREATE TABLE client (id INTEGER NOT NULL,surname CHAR(30) NOT NULL,name  
CHAR(30) NOT NULL,middlename CHAR(30) NOT NULL,passp_number CHAR(11) NOT  
NULL, PRIMARY KEY(id));
```

```
CREATE TABLE payment (id INTEGER NOT NULL,parking_id INTEGER NOT  
NULL,rate_id INTEGER NOT NULL,cost FLOAT NOT NULL,PRIMARY KEY(id));
```

```
CREATE TABLE rate(id INTEGER NOT NULL,h_cost INTEGER NOT NULL,PRIMARY  
KEY(id));
```

```
CREATE TABLE markhelper (id CHAR(10) NOT NULL,name CHAR(50) NOT NULL,  
PRIMARY KEY(id));
```

/* FK */

```
ALTER TABLE parking ADD CONSTRAINT FOREIGN KEY (client_id) REFERENCES  
client(id);
```

```
ALTER TABLE parking ADD CONSTRAINT FOREIGN KEY (car_id) REFERENCES car(id);
```

```
ALTER TABLE parking ADD CONSTRAINT FOREIGN KEY (parking_place_id)  
REFERENCES parking_place(id);
```

```
ALTER TABLE payment ADD CONSTRAINT FOREIGN KEY (parking_id) REFERENCES  
Parking(id);
```

```
ALTER TABLE payment ADD CONSTRAINT FOREIGN KEY (rate_id) REFERENCES  
Rate(id);
```

```
ALTER TABLE car ADD CONSTRAINT FOREIGN KEY (mark_id) REFERENCES  
markhelper(id);
```

/* PROCEDURES */

```
CREATE PROCEDURE payment_inserted(parking_id INTEGER)
```

```
UPDATE parking
```

```
SET pickup_date=CURRENT YEAR TO SECOND
```



```
WHERE id = parking_id;
END PROCEDURE;
```

```
CREATE PROCEDURE payment_updated(parking_id_old INTEGER,parking_id_new
INTEGER)
```

```
DEFINE old_date DATETIME YEAR TO SECOND;
LET old_date = (SELECT parking.pickup_date
FROM parking
WHERE parking.id = parking_id_old);
```

```
UPDATE parking
SET pickup_date=""
WHERE id = parking_id_old;
UPDATE parking
SET pickup_date=old_date
WHERE id = parking_id_new;
END PROCEDURE;
```

```
CREATE PROCEDURE payment_deleted(parking_id INTEGER)
UPDATE parking
SET pickup_date=""
WHERE id = parking_id;
END PROCEDURE;
```

```
CREATE PROCEDURE updatestate()
DEFINE place_id INTEGER;
UPDATE parking_place SET state="f";
FOREACH SELECT parking_place_id INTO place_id FROM parking WHERE pickup_date
is NULL ORDER BY parking_place_id
UPDATE parking_place SET state = "t" WHERE id = place_id;
END FOREACH
END PROCEDURE;
```

```
/* TRIGGERS */
CREATE TRIGGER payment_insert
INSERT ON payment
```

```
REFERENCING NEW AS newrow
FOR EACH ROW (EXECUTE PROCEDURE payment_inserted(newrow.parking_id))
ENABLED;
```

```
CREATE TRIGGER payment_delete
DELETE ON payment
REFERENCING OLD AS newrow
FOR EACH ROW (EXECUTE PROCEDURE payment_deleted(newrow.parking_id))
ENABLED;
```

```
CREATE TRIGGER payment_update
UPDATE ON payment
REFERENCING OLD AS oldrow
          NEW AS newrow
FOR EACH ROW (EXECUTE PROCEDURE
payment_updated(oldrow.parking_id,newrow.parking_id))
ENABLED;
```

```
CREATE TRIGGER parking_insert
INSERT ON parking
FOR EACH ROW (EXECUTE PROCEDURE updatestate())
ENABLED;
```

```
CREATE TRIGGER parking_delete
DELETE ON parking
FOR EACH ROW (EXECUTE PROCEDURE updatestate())
ENABLED;
```

```
CREATE TRIGGER parking_update
UPDATE ON parking
FOR EACH ROW (EXECUTE PROCEDURE updatestate())
ENABLED;
```

Заполнение (сервер demo_on)

```

INSERT INTO markhelper VALUES ("BMW-X5", "BMW X5");
INSERT INTO markhelper VALUES ("BMW-760", "BMW 760");
INSERT INTO markhelper VALUES ("LEXUS-ES", "Lexus ES");
INSERT INTO markhelper VALUES ("LEXUS-GS", "Lexus GS");
INSERT INTO markhelper VALUES ("VOLVO-S40", "Volvo S40");
INSERT INTO markhelper VALUES ("VOLVO-C70", "Volvo C70");
INSERT INTO markhelper VALUES ("AUDI-Q7", "Audi Q7");
INSERT INTO markhelper VALUES ("AUDI-A8", "Audi A8");
INSERT INTO markhelper VALUES ("VAZ-2109", "Ваз 2109");
INSERT INTO markhelper VALUES ("VAZ-2110", "Ваз 2110");
INSERT INTO markhelper VALUES ("VAZ-2112", "Ваз 2112");
INSERT INTO markhelper VALUES ("VAZ-2114", "Ваз 2114");

```

```

INSERT INTO car VALUES (200100, "BMW-X5", "B145HP", "116rus");
INSERT INTO car VALUES (200101, "BMW-760", "C345HP", "116rus");
INSERT INTO car VALUES (200102, "LEXUS-ES", "K748HP", "116rus");

```

```

INSERT INTO client VALUES(300104, "Демидов", "Сергей", "Владимирович", "7245 512341");
INSERT INTO client VALUES(300105, "Киселев", "Алексей", "Владимирович", "1242 612344");
INSERT INTO client VALUES(300106, "Павлова", "Татьяна", "Витальевна", "4512 925612");
INSERT INTO client VALUES(300107, "Хромова", "Ирина", "Анатольевна", "7223 034561");

```

```

INSERT INTO parking_place VALUES(400103, "4J", "f", "Место 4J");
INSERT INTO parking_place VALUES(400104, "4H", "f", "Место 4H");
INSERT INTO parking_place VALUES(400105, "BN", "f", "Место BN");
INSERT INTO parking_place VALUES(400106, "4T", "f", "Место 4T");
INSERT INTO parking_place VALUES(400107, "8H", "f", "Место 8H");

```

```

INSERT INTO rate VALUES (500100, 50);

```

```

INSERT INTO parking VALUES (600101, 300104, 200101, 400106, "2010-10-17 08:23:54", "");

```

Заполнение (сервер demo_on2)

```
INSERT INTO markhelper VALUES ("BMW-X5", "BMW X5");
INSERT INTO markhelper VALUES ("BMW-760", "BMW 760");
INSERT INTO markhelper VALUES ("LEXUS-ES", "Lexus ES");
INSERT INTO markhelper VALUES ("LEXUS-GS", "Lexus GS");
INSERT INTO markhelper VALUES ("VOLVO-S40", "Volvo S40");
INSERT INTO markhelper VALUES ("VOLVO-C70", "Volvo C70");
INSERT INTO markhelper VALUES ("AUDI-Q7", "Audi Q7");
INSERT INTO markhelper VALUES ("AUDI-A8", "Audi A8");
INSERT INTO markhelper VALUES ("VAZ-2109", "Ваз 2109");
INSERT INTO markhelper VALUES ("VAZ-2110", "Ваз 2110");
INSERT INTO markhelper VALUES ("VAZ-2112", "Ваз 2112");
INSERT INTO markhelper VALUES ("VAZ-2114", "Ваз 2114");
```

```
INSERT INTO car VALUES (200103, "VOLVO-C70", "O194HP", "116rus");
INSERT INTO car VALUES (200104, "VAZ-2109", "A607HP", "116rus");
INSERT INTO car VALUES (200105, "VAZ-2109", "T241HP", "116rus");
INSERT INTO car VALUES (200106, "VAZ-2109", "P523HP", "116rus");
INSERT INTO car VALUES (200107, "AUDI-A8", "C734HP", "116rus");
```

```
INSERT INTO client VALUES(300100, "Иванов", "Иван", "Иванович", "5438 724353");
INSERT INTO client VALUES(300101, "Алексеев", "Павел", "Сергеевич", "2548 622343");
INSERT INTO client VALUES(300102, "Бодров", "Сергей", "Борисович", "7234 213415");
INSERT INTO client VALUES(300103, "Глазов", "Михаил", "Михайлович", "0128 324513");
```

```
INSERT INTO parking_place VALUES(400100, "1A", "f", "Место 1A");
INSERT INTO parking_place VALUES(400101, "1B", "f", "Место 1B");
INSERT INTO parking_place VALUES(400102, "3C", "f", "Место 3C");
```

```
INSERT INTO rate VALUES (500100, 50);
```

```
INSERT INTO parking VALUES (600100, 300100, 200103, 400100, "2011-05-25  
16:34:54", "");
```

Заключение

В данной курсовой работе была реализована система автоматизации работы парковки. Система позволяет автоматизировать запись парковок автомобилей, печать справок о поставке на парковку, печать квитанции оплаты, автоматизированное управление свободными местами, тарифицируемый расчет стоимости парковки. Для обеспечения работы системы были использованы:

СУБД INFORMIX ONLINE v 11.7, платформа разработки Eclipse, средство объекто-реляционного связывания Hibernate, так же отчетная система JasperReports.