Практика по базам данных

ОТЧЕТ

Иванов Иван

Группа ХХХ

Предметная область: «Питомцы»

Реализация: MS SQLServer

Содержание

[ОПИСАНИЕ СИСТЕМЫ 2](#_Toc85014357)

[Требования 2](#_Toc85014358)

[Модель данных 2](#_Toc85014359)

[Функциональность 3](#_Toc85014360)

[Серверная часть 3](#_Toc85014361)

[Клиентская часть 4](#_Toc85014362)

[СКРИПТЫ 5](#_Toc85014363)

[Серверная часть 5](#_Toc85014364)

[Хранимые процедуры и функции 5](#_Toc85014365)

[Триггеры 6](#_Toc85014366)

[Представления 6](#_Toc85014367)

[Клиентская часть 7](#_Toc85014368)

[ПРИЛОЖЕНИЕ: Создание и заполнение базы данных 10](#_Toc85014369)

# ОПИСАНИЕ СИСТЕМЫ

## Требования

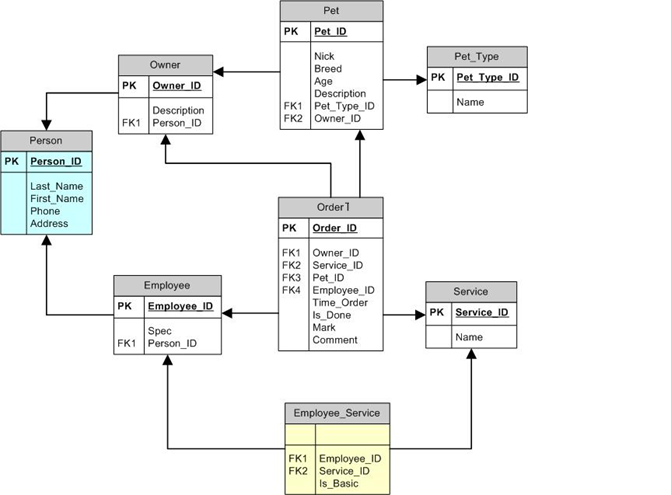
Вы создали компанию, которая обеспечивает для жителей окрестных домов следующие услуги: выгул собак и причесывание кошек.

Разработайте структуру хранения данных так, чтобы можно было работать со следующей информацией:

* Списки владельцев животных (ФИО, адрес, телефон, питомец).
* Списки кошек (кличка, возраст, порода) и собак (кличка, порода, время выгула).
* Списки исполнителей (ФИО, адрес, телефон, специализация).
* Заказы на неделю (владелец, питомец, исполнитель, дата, отметка о выполнении, оценка работы).
* Пакет заказов на день.

Следует предусмотреть также возможность создания для клиентов разного рода отчетов по их заказам (по периодам, по отдельным питомцам, отдельным услугам). И наконец, нужно будет поддерживать систему рейтинга сотрудников - исполнителей заказов.

## Модель данных



## Функциональность

### Серверная часть

:

|  |  |  |
| --- | --- | --- |
| ***Хранимые процедуры\функции*** | *Реализация* | *Комментарии* |
| Регистрация нового клиента | AddNewOwner() | (фамилия, имя, телефон, …) |
| Добавление питомца |  |  |
| Оформление нового заказа |  |  |
| Обработка заказа | UpdDelOrder() | (номер заказа, оценка, комментарий) перевод в состояние «выполнен» или удаление |
| Добавление исполнителя |  |  |
| Добавление услуги |  |  |
| Удаление заказа |  |  |
| … |  |  |
| Количество заказов на выгул | GetCountWalkingDay() | (дата), функция |
| Вычисление рейтинга исполнителя |  |  |
| … |  |  |

|  |  |  |
| --- | --- | --- |
| ***Триггеры*** | *Реализация* | *Комментарии* |
| Запрет на удаление физ.лица | Tr\_Del\_ Person |  |
| Деактивация вместо удаления клиента | Tr\_Del\_Client | для сохранения клиентской базы |
| Деактивация вместо удаления сотрудника |  | для отчетности в налоговую |
| Контроль повторного добавления клиента |  | откат при совпадении ФИО и адреса |
| Формирование номера при добавлении заказа |  | с использованием секвенции |
| После 100-го заказа исполнитель переводится в категорию СУПЕР-специалиста | Tr\_Ins\_Order1\_Super\_EMP |  |
| Пересчет позиции исполнителя в рейтинге после выполнения заказа |  | позиция хранится в поле описания исполнителя |
| … |  |  |

|  |  |  |
| --- | --- | --- |
| ***Представления*** | *Реализация* | *Комментарии* |
| Заказы | V\_OrderList | все заказы – выполненные и нет;  (услуга, время, исполнитель, питомец, хозяина,...) |
| Клиенты и их питомцы |  | (фамилия и имя клиента, телефон, адрес, кличка и вид питомца) |
| Виды услуг и исполнители |  | (услуга, фамилия исполнителя, телефон, специализация, основная или нет) |
| Реестр питомцев по видам |  | (вид, кличка, порода, возраст) |
| Собаки | V\_Dog | (кличка, порода, возраст, фамилия и имя хозяина) |
| Коты |  | -“- |
| Коровы |  | -“- |
| Рейтинг сотрудников | V\_Rating | (фамилия, имя, кол-во выполненных заказов, средний балл) |
| … |  |  |

### Клиентская часть

|  |  |  |  |
| --- | --- | --- | --- |
| *Экранные формы основные* | *дополнительные* | *Реализация*  *(запрос)* | *Что здесь можно использовать из серверной части* |
| Реестр клиентов |  | (01) Данные о клиентах, включая количество питомцев |  |
|  | Новый клиент |  | AddNewOwner() |
|  | Изменить данные клиента |  |  |
|  | Удалить клиента |  | Для деактивации:  Tr\_Del\_ Person, Tr\_Del\_Client |
|  | Добавить питомца |  |  |
|  | Фильтры | (02) Клиенты, не сделавшие за прошлый год ни одного заказа |  |
| Список питомцев |  |  |  |
|  | Изменить данные |  |  |
|  | Удалить питомца |  |  |
|  | Фильтр  (по хозяевам, по видам и т.д.) | (03) Хозяева собак максимального возраста | V\_Dog |
| Справочник видов питомцев |  | (04) Виды питомцев на буквы C и F, клички, возраст, порода и хозяева представителей этих видов |  |
|  | Добавить |  |  |
|  | Изменить |  |  |
| Реестр сотрудников |  |  |  |
|  | Добавить |  |  |
|  | Изменить |  |  |
|  | Деактивировать |  |  |
|  | Нагрузка | (05) Фамилии исполнителей, не получивших еще ни одного заказа  (06) Фамилии исполнителей, выполнивших в текущем году более пятидесяти заказов | Tr\_Ins\_Order1\_Super\_EMP |
|  | Рейтинг | (07) Обратный рейтинг исполнителей | V\_Rating |
| Справочник услуг |  | (08) Количество имеющихся исполнителей по каждой специальности для каждой услуги |  |
|  | Добавить |  |  |
|  | Изменить |  |  |
| Реестр заказов |  |  |  |
|  | Фильтр  (по периодам, владельцам, видам услуг и т.д.) | (09) Заказы за период  (10) Количество выполненных заказов по видам услуг | V\_OrderList  GetCountOrderDay() |
|  | Новый заказ |  |  |
|  | Изменить заказ |  | UpdDelOrder(‘UPD’, …) |
|  | Удалить заказ |  | UpdDelOrder(‘DEL’, …) |
| Служебные запросы |  | (11) Общий список всех комментариев  (12) Все оценки и комментарии по заказам, исполнителями которых являлись студенты |  |

# СКРИПТЫ

## Серверная часть

### Хранимые процедуры и функции

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- Добавление нового клиента

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE PROCEDURE AddNewOwner

@var\_last\_name AS CHAR(20),

@var\_first\_name AS CHAR(20),

@var\_phone AS CHAR(20),

@var\_address AS CHAR(50),

@var\_description AS CHAR(50)

AS

BEGIN

DECLARE @var\_new\_person\_id INTEGER;

SELECT @var\_new\_person\_id = MAX(Person\_Id) + 1 FROM Person;

INSERT INTO Person (Person\_ID, Last\_Name, First\_Name, Phone, Address)

VALUES(@var\_new\_person\_id, @var\_last\_name, @var\_first\_name, @var\_phone, @var\_address);

INSERT INTO Owner(Owner\_ID, Description, Person\_ID)

SELECT MAX(Owner\_ID) + 1, @var\_description, @var\_new\_person\_id FROM Owner;

END;

-- Пример вызова процедуры:  
-- EXECUTE AddNewOwner @var\_last\_name='Иванов',@var\_first\_name = 'Иван', @var\_phone='892011111111', @var\_address='', @var\_description = 'He does not remember his own address!'

-- DROP PROCEDURE AddNewOwner;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- Обработка заказа (перевод в состояние «выполнен» или удаление)

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE PROCEDURE UpdDelOrder

@var\_upd\_del AS CHAR(3),

@var\_order\_id AS INTEGER,

@var\_mark AS INTEGER,

@var\_comments AS VARCHAR(50)

AS

BEGIN

DECLARE @var\_new\_person\_id integer;

DECLARE @var\_new\_owner\_id integer;

IF @var\_upd\_del = 'UPD'

UPDATE ORDER1 SET Is\_Done = 1, Mark = @var\_mark, Comments = @var\_comments

WHERE Order\_ID = @var\_order\_id

IF @var\_upd\_del = 'DEL'

DELETE FROM ORDER1 WHERE Order\_ID = @var\_order\_id

END;

-- Пример вызова процедуры:

-- EXECUTE UpdDelOrder @var\_upd\_del = 'UPD', @var\_order\_id = 20, @var\_mark = 5, @var\_comments = ':(((';

-- EXECUTE UpdDelOrder @var\_upd\_del = 'DEL', @var\_order\_id = 25, @var\_mark = 0, @var\_comments = '';

-- DROP PROCEDURE UpdDelOrder;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- Количество заказов на выгул собак на день

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE FUNCTION GetCountWalkingDay (@day date)

RETURNS INTEGER

BEGIN

DECLARE @ResultInt INTEGER;

SELECT @ResultInt = COUNT(\*) FROM Order1 JOIN Pet ON Order1.Pet\_ID = Pet.Pet\_ID

JOIN Pet\_Type ON Pet.Pet\_Type\_ID = Pet\_Type.Pet\_Type\_ID

JOIN Service ON Order1.Service\_ID = Order1.Service\_ID

WHERE Service.Name = 'Walking' AND Pet\_Type.Name = 'DOG'

AND CONVERT(DATE, Order1.Time\_Order) = @day;

RETURN @ResultInt;

END;

-- Вызов:

-- SELECT dbo.GetCountWalkingDay (GETDATE())

-- DROP FUNCTION GetCountWalkingDay;

### Триггеры

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- Запрет на удаление

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER TR\_del\_Person ON Person FOR DELETE

AS

ROLLBACK;

-- Проверка: DELETE FROM Person;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- Деактивация вместо удаления клиента

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER TR\_del\_Owner ON Owner INSTEAD OF DELETE

AS

UPDATE Owner SET Description = ' DEACTIVATED ' + Description

WHERE Owner\_ID = (SELECT Owner\_ID FROM DELETED);

-- Проверка: DELETE FROM Owner WHERE Owner\_ID = 6;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- После 100-го заказа исполнитель переводится в категорию СУПЕР-специалиста

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER Tr\_Ins\_Order1\_Super ON Order1 FOR INSERT

AS

BEGIN

IF (SELECT COUNT(\*) FROM Order1

WHERE Order1.Employee\_ID = (SELECT Employee\_ID FROM INSERTED) AND Is\_Done=1) > 100

UPDATE Employee SET Spec = 'SUPER-' + Spec WHERE Employee.Employee\_ID =

(SELECT Employee\_ID FROM INSERTED);

END;

-- Проверка: INSERT INTO Order1 VALUES (33, 5, 2, 8, 1, '2018-09-10 16:00', 0, 0);

### Представления

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- Реестр заказов (услуга, время, исполнитель, питомец, хозяина,...)

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW V\_OrderList(Name, Time\_Order, emp\_Last\_Name, Nick, own\_Last\_Name, Is\_Done)

AS

SELECT Service.Name, Order1.Time\_Order, emp.Last\_Name, Pet.Nick, own.Last\_Name, Order1.Is\_Done

FROM Order1, Employee, Person AS emp, Pet, Owner, Person AS own, Service

WHERE Order1.Employee\_ID = Employee.Employee\_ID AND Employee.Person\_ID = emp.Person\_ID

AND Order1.Pet\_ID = Pet.Pet\_ID AND Pet.Owner\_ID = Owner.Owner\_ID

AND Owner.Person\_ID = own.Person\_ID AND Order1.Service\_ID = Service.Service\_ID;

-- DROP VIEW V\_OrderList;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- Собаки (кличка, порода, возраст, фамилия и имя хозяина)

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW V\_Dog

AS

SELECT Nick, Age,Breed, Last\_Name, First\_Name

FROM Pet\_Type JOIN Pet ON Pet\_Type.Pet\_Type\_ID = Pet.Pet\_Type\_ID

JOIN Owner ON Pet.Owner\_ID = Owner.Owner\_ID

JOIN Person ON Person.Person\_ID = Owner.Person\_ID

WHERE Name = 'dog';

-- DROP VIEW V\_Dog;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- Рейтинг сотрудников (фамилия, имя, количество выполненных заказов, средний балл)

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW V\_Rating (Last\_Name, First\_Name, CNT, AVG)

AS

SELECT Last\_Name, First\_Name, COUNT(\*), AVG(CONVERT(FLOAT,Mark))

FROM Person JOIN Employee ON Person.Person\_ID = Employee.Person\_ID

JOIN Order1 ON Employee.Employee\_ID = Order1.Employee\_ID

WHERE Is\_Done = 1 GROUP BY Last\_Name, First\_Name;

-- DROP VIEW V\_Rating;

## Клиентская часть

(запросы для экранных форм и отчетов)

-- (01) Данные о клиентах, включая количество питомцев

select Person.First\_Name, Person.Last\_Name, Person.Phone, cnt

from Person join (select Owner.Person\_ID, COUNT(\*) as cnt

from Owner join Pet on Pet.Owner\_ID = Owner.Owner\_ID

GROUP BY Owner.Person\_ID) ttt

ON Person.Person\_ID = ttt.Person\_ID

-- (02) Клиенты, не сделавшие за прошлый год ни одного заказа

SELECT Person.Last\_Name, Person.First\_Name from Owner JOIN Person ON Owner.Person\_ID = Person.Person\_ID

WHERE NOT EXISTS (SELECT \* FROM order1

WHERE Order1.Owner\_ID = Owner.Owner\_ID

AND DATEPART(YEAR, Order1.Time\_Order) = DATEPART(YEAR, GETDATE()) - 1);

-- (03) Хозяева собак максимального возраста

SELECT Last\_Name, First\_Name FROM V\_Dog

WHERE Age = (SELECT MAX(Age) FROM V\_Dog);

-- (04) Виды питомцев на буквы C и F, клички, возраст, порода и хозяева представителей этих видов

SELECT Pet\_Type.Name, Pet.Nick, Pet.Age, Pet.Breed, Person.Last\_Name, Person.First\_Name

FROM Pet\_Type LEFT JOIN Pet ON Pet\_Type.Pet\_Type\_ID = Pet.Pet\_Type\_ID

LEFT Join Owner ON Owner.Owner\_ID = Pet.Owner\_ID

LEFT Join Person ON Person.Person\_ID = Owner.Person\_ID

WHERE Pet\_Type.Name LIKE 'C%' OR Pet\_Type.Name LIKE 'F%'

ORDER BY Pet\_Type.Name;

-- (05) Фамилии исполнителей, не получивших еще ни одного заказа

SELECT Person.Last\_Name FROM Employee, Person

WHERE Employee.Person\_ID = Person.Person\_ID

AND Employee\_ID NOT IN (SELECT Employee\_ID FROM Order1);

-- (06) Фамилии исполнителей, выполнивших в текущем году более пятидесяти заказов

SELECT Person.Last\_Name, COUNT(\*)

FROM Person JOIN Employee ON Person.Person\_ID = Employee.Person\_ID

JOIN Order1 ON Employee.Employee\_ID = Order1.Employee\_ID

WHERE Order1.Is\_Done = 1

AND DATEPART(YEAR, Order1.Time\_Order) = DATEPART(YEAR, GETDATE())

GROUP BY Person.Last\_Name

HAVING COUNT(\*) > 50;

-- (07) Обратный рейтинг исполнителей

SELECT \* FROM V\_Rating ORDER BY 4 DESC;

--(08) Количество имеющихся исполнителей по каждой специальности для каждой услуги

SELECT Service.Name, Employee.Spec, COUNT(\*)

FROM Service

JOIN Employee\_Service ON Service.Service\_ID = Employee\_Service.Service\_ID

JOIN Employee ON Employee\_Service.Employee\_ID = Employee.Employee\_ID

GROUP BY Service.Name, Employee.Spec

ORDER BY 1

-- (09) Заказы за период

SELECT \* FROM V\_OrderList

WHERE Time\_Order BETWEEN '2018-09-04' AND '2018-09-11'

ORDER BY 2 DESC;

-- (10) Количество выполненных заказов по видам услуг

SELECT Name, COUNT(\*) FROM V\_OrderList WHERE Is\_Done = 1

GROUP BY Name ORDER BY Name;

-- (11) Общий список всех комментариев

SELECT Comments, 'о заказе' FROM Order1 WHERE Comments IS NOT NULL AND Comments <> ''

UNION

SELECT Description, 'о клиенте' FROM Owner WHERE Description IS NOT NULL AND Description <> ''

UNION

SELECT Description, 'о питомце' FROM Pet WHERE Description IS NOT NULL AND Description <> ''

ORDER BY 2;

-- (12) Все оценки и комментарии по заказам, исполнителями которых являлись студенты

SELECT Mark, Comments FROM Order1 WHERE Employee\_ID IN

(SELECT Employee\_ID FROM Employee WHERE Spec LIKE '%student%') AND Is\_Done = 1;

# ПРИЛОЖЕНИЕ: Создание и заполнение базы данных

-- CREATE DATABASE pets\_db;

-- GO

-- USE pets\_db;

-- CREATE DATABASE pets\_db\_2019;

-- GO

-- USE pets\_db\_2019;

---------------------------------------------------------------

-- Создание таблиц и PK

---------------------------------------------------------------

CREATE TABLE Person(

Person\_ID INTEGER NOT NULL,

Last\_Name VARCHAR(20) NOT NULL,

First\_Name VARCHAR(20),

Phone VARCHAR(15) NOT NULL,

Address VARCHAR(50) NOT NULL,

CONSTRAINT Person\_PK PRIMARY KEY (Person\_ID)

)

;

CREATE TABLE Owner(

Owner\_ID INTEGER NOT NULL,

Description VARCHAR(50),

Person\_ID INTEGER NOT NULL,

CONSTRAINT Owner\_PK PRIMARY KEY (Owner\_ID)

)

;

CREATE TABLE Employee(

Employee\_ID INTEGER NOT NULL,

Spec VARCHAR(15),

Person\_ID INTEGER NOT NULL,

CONSTRAINT Employee\_PK PRIMARY KEY (Employee\_ID)

)

;

CREATE TABLE Pet\_Type(

Pet\_Type\_ID INTEGER NOT NULL,

Name VARCHAR(15) NOT NULL,

CONSTRAINT Pet\_Type\_PK PRIMARY KEY (Pet\_Type\_ID)

)

;

CREATE TABLE Pet(

Pet\_ID INTEGER NOT NULL,

Nick VARCHAR(15) NOT NULL,

Breed VARCHAR(20),

Age INTEGER,

Description VARCHAR(50),

Pet\_Type\_ID INTEGER NOT NULL,

Owner\_ID INTEGER NOT NULL,

CONSTRAINT Pet\_PK PRIMARY KEY (Pet\_ID)

)

;

CREATE TABLE Service(

Service\_ID INTEGER NOT NULL,

Name VARCHAR(15) NOT NULL,

CONSTRAINT Service\_PK PRIMARY KEY (Service\_ID)

)

;

CREATE TABLE Employee\_Service(

Employee\_ID INTEGER NOT NULL,

Service\_ID INTEGER NOT NULL,

Is\_Basic INTEGER

)

;

CREATE TABLE order1(

Order\_ID INTEGER NOT NULL,

Owner\_ID INTEGER NOT NULL,

Service\_ID INTEGER NOT NULL,

Pet\_ID INTEGER NOT NULL,

Employee\_ID INTEGER NOT NULL,

Time\_Order DATETIME DEFAULT GETDATE() NOT NULL,

Is\_Done INTEGER DEFAULT 0 NOT NULL,

Mark INTEGER,

Comments VARCHAR(50),

CONSTRAINT Order\_Is\_Done CHECK (Is\_Done in (0,1)),

CONSTRAINT Order\_PK PRIMARY KEY (Order\_ID)

)

;

---------------------------------------------------------------

-- Создание FK

---------------------------------------------------------------

ALTER TABLE Owner ADD CONSTRAINT FK\_Owner\_Person

FOREIGN KEY (Person\_ID)

REFERENCES Person(Person\_ID)

;

ALTER TABLE Employee ADD CONSTRAINT FK\_Employee\_Person

FOREIGN KEY (Person\_ID)

REFERENCES Person(Person\_ID)

;

ALTER TABLE Pet ADD CONSTRAINT FK\_Pet\_0wner

FOREIGN KEY (Owner\_ID)

REFERENCES Owner(Owner\_ID)

;

ALTER TABLE Pet ADD CONSTRAINT FK\_Pet\_Pet\_Type

FOREIGN KEY (Pet\_Type\_ID)

REFERENCES Pet\_Type(Pet\_Type\_ID)

;

ALTER TABLE Employee\_Service ADD CONSTRAINT FK\_Empl\_Serv\_Employee

FOREIGN KEY (Employee\_ID)

REFERENCES Employee(Employee\_ID)

;

ALTER TABLE Employee\_Service ADD CONSTRAINT FK\_Empl\_Serv\_Service

FOREIGN KEY (Service\_ID)

REFERENCES Service(Service\_ID)

;

ALTER TABLE Order1 ADD CONSTRAINT FK\_Order\_Employee

FOREIGN KEY (Employee\_ID)

REFERENCES Employee(Employee\_ID)

;

ALTER TABLE order1 ADD CONSTRAINT FK\_Order\_0wner

FOREIGN KEY (Owner\_ID)

REFERENCES Owner(Owner\_ID)

;

ALTER TABLE order1 ADD CONSTRAINT FK\_Order\_Pet

FOREIGN KEY (Pet\_ID)

REFERENCES Pet(Pet\_ID)

;

ALTER TABLE order1 ADD CONSTRAINT FK\_Order\_Service

FOREIGN KEY (Service\_ID)

REFERENCES Service(Service\_ID)

;

---------------------------------------------------------------

-- Заполнение таблиц тестовыми данными

---------------------------------------------------------------

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (1, 'Ivanov', 'Vania', '+79123456789', 'Srednii pr VO, 34-2');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (2, 'Petrov', 'Petia', '+79234567890', 'Sadovaia ul, 17\2-23');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (3, 'Vasiliev', 'Vasia', '+7345678901', 'Nevskii pr, 9-11');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (4, 'Orlov', 'Oleg', '+7456789012', '5 linia VO, 45-8');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (5, 'Galkina', 'Galia', '+7567890123', '10 linia VO, 35-26');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (6, 'Sokolov', 'S.', '+7678901234', 'Srednii pr VO, 27-1');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (7, 'Vorobiev', 'Vova', '123-45-67', 'Universitetskaia nab, 17');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (8, 'Ivanov', 'Vano', '+7789012345', 'Malyi pr VO, 33-2');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (9, 'Sokolova', 'Sveta', '234-56-78', '');

INSERT INTO Person(Person\_ID, Last\_Name, First\_Name, Phone, Address) VALUES (10, 'Zotov', 'Misha', '111-56-22', '');

INSERT INTO Owner(Owner\_ID, Description, Person\_ID) VALUES (1, 'good boy', 4);

INSERT INTO Owner(Owner\_ID, Description, Person\_ID) VALUES (2, '', 5);

INSERT INTO Owner(Owner\_ID, Description, Person\_ID) VALUES (3, '', 6);

INSERT INTO Owner(Owner\_ID, Description, Person\_ID) VALUES (4, 'from the ArtsAcademy', 7);

INSERT INTO Owner(Owner\_ID, Description, Person\_ID) VALUES (5, '', 8);

INSERT INTO Owner(Owner\_ID, Description, Person\_ID) VALUES (6, 'mean', 9);

INSERT INTO Employee(Employee\_ID, Spec, Person\_ID) VALUES (1, 'boss', 1);

INSERT INTO Employee(Employee\_ID, Spec, Person\_ID) VALUES (2, 'hairdresser', 2);

INSERT INTO Employee(Employee\_ID, Spec, Person\_ID) VALUES (3, 'student',3);

INSERT INTO Employee(Employee\_ID, Spec, Person\_ID) VALUES (4, 'student', 10);

INSERT INTO Pet\_Type(Pet\_Type\_ID, NAME) VALUES (1, 'DOG');

INSERT INTO Pet\_Type(Pet\_Type\_ID, NAME) VALUES (2, 'CAT');

INSERT INTO Pet\_Type(Pet\_Type\_ID, NAME) VALUES (3, 'COW');

INSERT INTO Pet\_Type(Pet\_Type\_ID, NAME) VALUES (4, 'FISH');

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (1, 'Bobik', 'unknown', 3, NULL, 1, 1);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (2, 'Musia', NULL, 12, NULL, 2, 1);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (3, 'Katok', NULL, 2, 'crazy guy', 2, 1);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (4, 'Apelsin', 'poodle', 5, NULL,1, 2);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (5, 'Partizan', 'Siamese', 5, 'very big', 2, 2);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (6, 'Daniel', 'spaniel', 14, NULL, 1, 3);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (7, 'Model', NULL, 5, NULL, 3, 4);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (8, 'Markiz', 'poodle', 1, NULL, 1, 5);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (9, 'Zombi', 'unknown', 7, 'wild', 2, 6);

INSERT INTO Pet(Pet\_ID, Nick, Breed, Age, Description, Pet\_Type\_ID, Owner\_ID) VALUES (10, 'Las', 'Siamese', 7, '', 2, 6);

INSERT INTO Service(Service\_ID, NAME) VALUES (1, 'Walking');

INSERT INTO Service(Service\_ID, NAME) VALUES (2, 'Combing');

INSERT INTO Service(Service\_ID, NAME) VALUES (3, 'Milking');

INSERT INTO Employee\_Service(Employee\_ID, Service\_ID, Is\_Basic) VALUES (1, 1, 0);

INSERT INTO Employee\_Service(Employee\_ID, Service\_ID, Is\_Basic) VALUES (1, 2, 0);

INSERT INTO Employee\_Service(Employee\_ID, Service\_ID, Is\_Basic) VALUES (1, 3, 1);

INSERT INTO Employee\_Service(Employee\_ID, Service\_ID, Is\_Basic) VALUES (2, 1, 0);

INSERT INTO Employee\_Service(Employee\_ID, Service\_ID, Is\_Basic) VALUES (2, 2, 1);

INSERT INTO Employee\_Service(Employee\_ID, Service\_ID, Is\_Basic) VALUES (3, 1, 1);

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (1, 1, 1, 1, 3, '2018-09-04 08:00', 1, 5, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (2, 1, 2, 2, 2, '2018-09-04 09:00', 1, 4, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (3, 1, 2, 3, 2, '2018-09-04 09:00', 0, 0, 'That cat is crazy!');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (4, 1, 1, 1, 3, '2018-09-05 00:00', 1, 5, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (5, 1, 1, 1, 3, '2018-09-06 11:00', 1, 3, 'Comming late');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (6, 1, 1, 1, 3, '2018-09-07 17:00', 1, 5, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (7, 1, 2, 2, 2, '2018-09-07 18:00', 1, 5, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (8, 2, 1, 5, 3, '2018-09-07 18:00', 1, 4, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (9, 2, 1, 4, 3, '2018-09-07 10:00', 1, 4, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (10, 2, 1, 5, 3, '2018-09-08 17:00', 1, 4, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (11, 2, 1, 4, 3, '2018-09-08 18:00', 1, 4, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (12, 2, 1, 5, 3, '2018-09-08 12:00', 1, 4, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (13, 2, 1, 4, 3, '2018-09-08 14:00', 1, 4, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (14, 3, 1, 6, 3, '2018-09-09 10:00', 1, 5, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (15, 3, 2, 6, 2, '2018-09-09 18:00', 0, 0, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (16, 3, 1, 6, 3, '2018-09-10 10:00', 0, 0, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (17, 3, 1, 6, 3, '2018-09-10 11:00', 0, 0, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (18, 3, 1, 6, 3, '2018-09-12 10:00', 0, 0, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (19, 3, 1, 6, 3, '2018-09-13 10:00', 0, 0, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark, Comments)

VALUES (20, 4, 3, 7, 1, '2018-09-10 11:00', 1, 5, '');

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark)

VALUES (21, 4, 3, 7, 1, '2018-09-11 11:00', 0, 0);

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark)

VALUES (22, 4, 3, 7, 1, '2018-09-12 11:00', 0, 0);

INSERT INTO order1(Order\_ID, Owner\_ID, Service\_ID, Pet\_ID, Employee\_ID, Time\_Order, Is\_Done, Mark)

VALUES (23, 5, 2, 8, 2, '2018-09-10 16:00', 0, 0);

-------------------------------------------------------------------------------------------------------------------------------------------------------

-- Создание индексов

-------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE INDEX idx\_pet\_nick ON Pet(Nick);

CREATE INDEX idx\_order\_time ON Order1(Time\_Order);

CREATE UNIQUE INDEX idx\_person\_full\_name

ON Person (First\_name, Last\_name);

-------------------------------------------------------------------------------------------------------------------------------------------------------

-- Удаление таблиц

-------------------------------------------------------------------------------------------------------------------------------------------------------

/\*

DROP TABLE Ord;

DROP TABLE Employee\_Service;

DROP TABLE Service;

DROP TABLE Pet;

DROP TABLE Pet\_Type;

DROP TABLE Employee;

DROP TABLE Owner;

DROP TABLE Person;

\*/