Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

Рязанский государственный радиотехнический университет  
имени В.Ф. Уткина

Кафедра ЭВМ

К защите

Руководитель работы:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

по дисциплине

**«Базы данных»**

Тема:

«Разработка базы данных для предметной области:

Птицеферма»

Выполнил студент группы 145

Мелёхин Д.В.

дата сдачи на проверку, подпись

Руководитель работы

ассистент каф. ЭВМ

Баранова С.Н.

оценка дата защиты, подпись

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc122985368)

[1 Семантическое описание предметной области 5](#_Toc122985369)

[2 Разработка базы данных 6](#_Toc122985370)

[2.1 Разработка логической модели данных 6](#_Toc122985371)

[2.2 Определение типов хранимых данных 12](#_Toc122985372)

[2.3 Задание ограничений целостности данных 14](#_Toc122985373)

[2.4 Построение схемы БД 16](#_Toc122985374)

[2.5 Разработка сценария для создания БД и основных объектов структуры БД 17](#_Toc122985375)

[3 Разработка основных операций реляционной алгебры 20](#_Toc122985376)

[4 Разработка запросов к БД для реализации информационных потребностей пользователя 24](#_Toc122985377)

[ЗАКЛЮЧЕНИЕ 40](#_Toc122985378)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 41](#_Toc122985379)

## ВВЕДЕНИЕ

База данных (БД) ‒ это поименованная совокупность данных, организованная по определенным правилам, которые включают общие принципы описания, хранения и манипулирования данными. В БД хранится информация об объектах предметной области (сущностях) и связях между ними.

Основными компонентами программного обеспечения информационной системы являются системы управления базами данных и пользовательские прикладные программы или клиентские приложения.

СУБД осуществляет следующие манипуляции с базами данных:

* создание баз данных и управление ими: удаление, изменение и объединение;
* хранение данных в структурированном виде и необходимом формате;
* защита данных от нежелательных изменений и попыток взлома;
* загрузка и сортировка данных с помощью фильтров;
* создание резервных копий, восстановление баз данных, поддерживание общей целостности.

Цель данной курсовой работы ‒ создание базы данных; предметная область БД ‒ «Птицеферма». База будет содержать информацию о птичниках и птицах, содержащихся в них; информацию о скупщиках и продажах. У пользователя будет возможность получать необходимую информацию в желаемом виде.

**Задание**

База данных «Птицеферма» должна хранить информацию о продаже скупщикам птицы и о содержании птиц. Требуется хранить данные:

1. О видах птиц (Наименование вида, средний вес птиц данного вида, цена мяса данного вида птицы за килограмм).
2. О птичниках (Номер, вид птичника, вместимость, вид птицы, ФИО ответственного). Вид птичника выбрать из списка: брудергауз, батарейный цех, акклиматизатор, репродукторный, взрослый.
3. О скупщиках (Наименование, контактный телефон, адрес, статус). Статус выбрать из списка: физическое лицо, юридическое лицо.
4. О продаже птиц скупщику с указанием даты продажи, количества проданных птиц данного вида и общей стоимости.

# Семантическое описание предметной области

Птицеферма предназначена для выращивания и продажи домашней птицы. Птицы бывают различных пород и видов. Каждая из них отличается своим весом, ценой и особенностями выращивания. Птицеферма располагает в себе множество птичников, разделённых на виды: брудергауз, батарейный цех, акклиматизатор, репродукторный, взрослый. [Брудергауз](https://dic.academic.ru/dic.nsf/dic_fwords/43781/%D0%B1%D1%80%D1%83%D0%B4%D0%B5%D1%80%D0%B3%D0%B0%D1%83%D0%B7) ‒ обогреваемый птичник для выращивания молодняка птицы (кур, индеек, уток и гусей) с суточного до 60 70 суточного возраста. Батарейный цех предназначен для выращивания молодой птицы в клеточных батареях. Акклиматизатор ‒ птичник для выращивания молодняка кур, индеек, уток и гусей в условиях, близких к условиям содержания взрослой птицы. По истечении некоторого срока птица переходит во взрослый птичник. В репродукторном птичнике выращивается ремонтный молодняк. Каждый птичник имеет свой индивидуальный номер и вместимость. В птичнике хранится только один вид птиц; за каждым птичником есть ответственное лицо.

У птицефермы есть тёплая база скупщиков, в которой находится большое количество как физических, так и юридических лиц. Информация о них включает в себя наименование, контактный телефон, адрес и статус. С каждым новым заказчиком база обновляется. Данный инструмент очень важен для производства, так как он позволяет коммуницировать со скупщиками для продолжения сотрудничества.

Отчет о продажах имеет большое значение для компаний с развитой торговой сетью. На основании такого отчета можно сделать выводы об эффективности работы сотрудников, оценить оптимальную товарную матрицу, понять структуру клиентов и, при необходимости, скорректировать работу персонала. У птицефермы есть база, хранящая данные о продаже птиц скупщику с указанием даты продажи, количества проданных птиц данного вида и общей стоимости.

# Разработка базы данных

## Разработка логической модели данных

**Описание основных сущностей ПО.**

На основании проведенного анализа предметной области «Птицеферма» можно выделить следующие сущности: Птицы, Птичники, Скупщики, Продажи. Представим список сущностей предметной области (таблица 1).

Таблица 1 ‒ Список сущностей

|  |  |  |
| --- | --- | --- |
| № | Название | Назначение |
| 1 | Птицы | Номер вида, наименование вида, закреплённый птичник, средний вес птиц данного вида, цена мяса данного вида птицы за килограмм. |
| 2 | Птичники | Номер, вид птичника (брудергауз, батарейный цех, акклиматизатор, репродукторный, взрослый), вместимость, ФИО ответственного. |
| 3 | Скупщики | Идентификационный номер, наименование, контактный телефон, адрес, статус (физическое лицо, юридическое лицо). |
| 4 | Продажи | Скупщик, дата продажи, количество проданных птиц данного вида, общая стоимость. |

Для каждой сущности приведём описание ее атрибутов в виде таблицы. На данном этапе выделяем ключевые (это первичные, внешние и альтернативные ключи) и неключевые атрибуты. Следует отметить, что для сущностей Птицы, Птичники и Скупщики нет атрибутов, с помощью которых можно идентифицировать конкретные данные. Для этого в таблицу «Птицы» был добавлен атрибут «Номер вида», в таблицу «Птичники» атрибут «Номер», а в таблицу «Скупщики» ‒ «Идентификационный номер». Результаты приведены в таблицах 2-5.

Таблица 2 ‒ Список атрибутов таблицы «Продажи»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ВК (внешний ключ)  ПК (первичный ключ) | Скупщик | С помощью данного внешнего ключа можно определить данные о скупщике.  Является частью составного ключа. |
| ПК (первичный ключ) | Дата продажи | Ключевое поле. Представляет собой первичный ключ, является частью составного ключа. |
| ВК (внешний ключ) | Вид птицы | С помощью данного внешнего ключа можно определить информацию о данном виде птицы. |
|  | Количество проданных птиц данного вида |  |
|  | Общая стоимость |  |

Таблица 3 ‒ Список атрибутов таблицы «Птицы»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Номер вида | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому виду птицы. Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Наименование вида |  |
| ВК (внешний ключ) | Закреплённый птичник | С помощью данного внешнего ключа можно определить информацию о принадлежности вида к конкретному птичнику. |
|  | Средний вес птиц данного вида |  |
|  | Цена мяса данного вида птицы за килограмм |  |

Таблица 4 ‒ Список атрибутов таблицы «Птичники»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Номер | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому номеру птичника. Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Вид птичника |  |
|  | Вместимость |  |
|  | ФИО ответственного |  |

Таблица 5 ‒ Список атрибутов таблицы «Скупщики»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Идентификационный номер | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому скупщику. Значения автоматически генерируется СУБД. |
|  | Наименование |  |

Продолжение таблицы 5.

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
|  | Контактный телефон |  |
|  | Адрес |  |
|  | Статус |  |

**Построение логической модели**

На основании семантического описания предметной области и списка атрибутов из таблиц 2-5 опишем классы объектов (сущностей) и их свойства, расставим существующие связи между ними и приведем обоснование типов связей (1:1, 1:М и т.д.). Результат представим в виде таблицы Список связей с описанием характеристик связей (таблица 6) и логической модели в графическом виде, где будут изображены все таблицы и связи между ними (рисунок 1).

Таблица 6 ‒ Список связей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущности, участвующие в связи | Тип связи | Обоснование |
| 1 | Продажи ‒ Скупщики | 1:N | Каждый скупщик может совершать покупку любое количество раз. |
| 2 | Птичники ‒ Птицы | 1:N | Несколько видов птиц может содержаться в одном птичнике. |
| 3 | Птицы ‒ Продажи | 1:N | Каждый скупщик может приобретать любой вид птиц. |

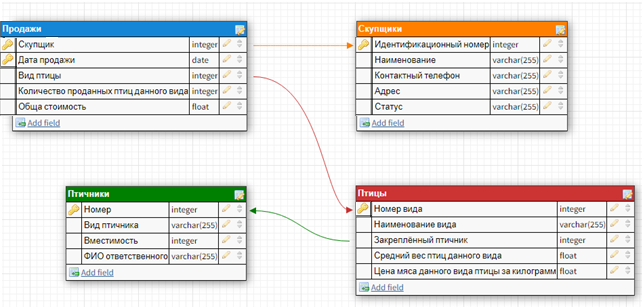


Рисунок 1 ‒ Связи между таблицами

## Определение типов хранимых данных

Для каждой таблицы приведем список всех атрибутов с указанием типа данных. При выборе типов данных следует учитывать, что далее для реализации будет использоваться СУБД Microsoft SQL Server. Результаты приведены в таблицах 7-10.

Таблица 7 ‒ Список атрибутов таблицы «Продажи»

|  |  |
| --- | --- |
| Название | Тип данных |
| Скупщик | int |
| Дата продажи | date |
| Вид птицы | int |
| Количество проданных птиц данного вида | int |
| Общая стоимость | money |

Таблица 8 ‒ Список атрибутов таблицы «Птицы»

|  |  |
| --- | --- |
| Название | Тип данных |
| Номер вида | int |
| Наименование вида | varchar(30) |
| Закреплённый птичник | int |
| Средний вес птиц данного вида | numeric (5, 3) |
| Цена мяса данного вида птицы за килограмм | money |

Таблица 9 ‒ Список атрибутов таблицы «Птичники»

|  |  |
| --- | --- |
| Название | Тип данных |
| Номер | int |
| Вид птичника | varchar(14) |
| Вместимость | smallint |

Продолжение таблицы 9.

|  |  |
| --- | --- |
| Название | Тип данных |
| ФИО ответственного | varchar(30) |

Таблица 10 ‒ Список атрибутов таблицы «Скупщики»

|  |  |
| --- | --- |
| Название | Тип данных |
| Идентификационный номер | int |
| Наименование | varchar(30) |
| Контактный телефон | varchar(18) |
| Адрес | varchar(60) |
| Статус | varchar(20) |

## Задание ограничений целостности данных

Для каждой таблицы необходимо выделить ограничения, накладываемые на значения атрибутов. Это могут быть ограничения на значения, значения по умолчанию, запрет/разрешение использования NULL-значений, поддержка уникальности значений и др. Результаты представлены в таблицах 11-14.

Таблица 11 ‒ Ограничения таблицы «Продажи»

|  |  |
| --- | --- |
| Название | Ограничения |
| Скупщик | Первичный ключ, внешний ключ. |
| Дата продажи | Первичный ключ, ДД-ММ-ГГГГ. |
| Вид птицы | Внешний ключ. |
| Количество проданных птиц данного вида | Значения больше 0; по умолчанию 20. |
| Общая стоимость | Положительные значения. |

Таблица 12 ‒ Ограничения таблицы «Птицы»

|  |  |
| --- | --- |
| Название | Ограничения |
| Номер вида | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |
| Наименование вида | Уникальное значение. |
| Закреплённый птичник | Внешний ключ. |
| Средний вес птиц данного вида | Значения больше 0; по умолчанию 1,500. |
| Цена мяса данного вида птицы за килограмм | Значения больше 0; по умолчанию 450. |

Таблица 13 ‒ Ограничения таблицы «Птичники»

|  |  |
| --- | --- |
| Название | Ограничения |
| Номер | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |
| Вид птичника | Брудергауз, батарейный цех, акклиматизатор, репродукторный, взрослый. |
| Вместимость | Значения больше 0; по умолчанию 5000. |
| ФИО ответственного |  |

Таблица 14 ‒ Ограничения таблицы «Скупщики»

|  |  |
| --- | --- |
| Название | Ограничения |
| Идентификационный номер | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |
| Наименование |  |
| Контактный телефон | +7 (NNN) NNN-NN-NN, где N – цифра от 0 до 9. |
| Адрес |  |
| Статус | Физическое лицо, юридическое лицо. |

## Построение схемы БД

Схема БД представляет собой набор заголовков таблиц. То есть на схеме БД указываются имена таблиц (сущностей), их атрибуты и связи между таблицами. Для уточнения схемы БД для каждой сущности (таблицы) указывают наименование атрибута, идентификатор, тип, длина, формат, ограничения. На рисунке 2 представлена схема базы данных «Птицеферма».

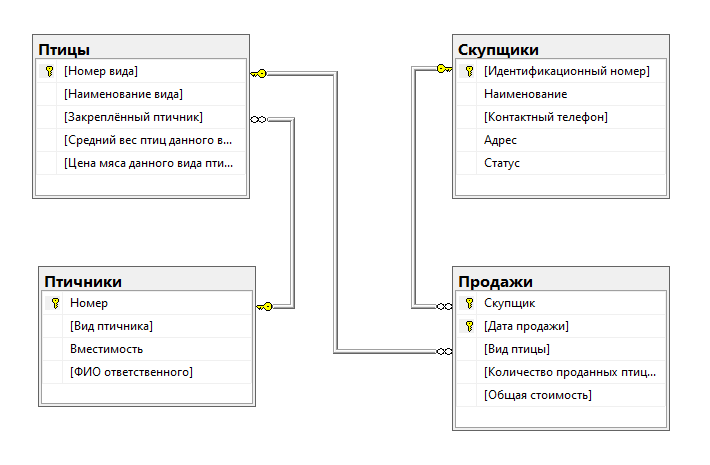


Рисунок 2 ‒ Схема базы данных «Птицеферма»

## Разработка сценария для создания БД и основных объектов структуры БД

В данном разделе требуется описать сценарий создания разработанной БД и основных объектов структуры БД. При создании таблиц будем использовать следующие способы задания ограничений:

1. При описании оператора CREATE TABLE будем задавать следующие ограничения:

* ограничения NOT NULL как ограничение на столбец;
* ограничения DEFAULT как ограничение на столбец;
* неповторяющиеся ограничения CHECK и UNIQUE, то есть, если конкретное ограничение будет использоваться только один раз в БД. Такие ограничения будем задавать как ограничения на столбец или на таблицу (в зависимости от количества столбцов, входящих в ограничение).

1. При описании оператора ALTER TABLE будем задавать следующие ограничения:

* повторяющиеся ограничения CHECK и UNIQUE, то есть, если конкретное ограничение будет использоваться несколько раз в БД;
* первичные ключи (простые и составные);
* внешние ключи. Всем ограничениям, описанным в операторе ALTER TABLE, будем задавать имена.
* Создание файлов БД

USE master

CREATE DATABASE Птицеферма

ON

(NAME = 'Птицеферма',

FILENAME = 'D:\SQL\MSSQL13.MSSQLSERVER\MSSQL\DATA\Птицеферма.mdf',

SIZE = 1,

MAXSIZE = 10,

FILEGROWTH = 1)

LOG ON

(NAME = 'Птицеферма\_log',

FILENAME = 'D:\SQL\MSSQL13.MSSQLSERVER\MSSQL\DATA\Птицеферма\_log.mdf',

SIZE = 1,

MAXSIZE = 5,

FILEGROWTH = 1)

GO

* Создание и заполнение таблиц

USE Птицеферма

GO

CREATE TABLE Продажи

(Скупщик int not null, [Дата продажи] date not null,

[Вид птицы] int not null, [Количество проданных птиц данного вида] int not null check ([Количество проданных птиц данного вида] > 0) default 20,

[Общая стоимость] money not null check ([Общая стоимость] >= 0))

GO

CREATE TABLE Птицы

([Номер вида] int not null identity (1, 1), [Наименование вида] varchar(30) not null unique, [Закреплённый птичник] int not null,

[Средний вес птиц данного вида] numeric (5,3) not null check ([Средний вес птиц данного вида] >0) default 1.500,

[Цена мяса данного вида птицы за килограмм] money not null check ([Цена мяса данного вида птицы за килограмм] >0) default 450.00)

GO

CREATE TABLE Птичники

(Номер int not null identity (1, 1), [Вид птичника] varchar(14) not null check ([Вид птичника] IN ('Брудергауз', 'Батарейный цех', 'Акклиматизатор', 'Репродукторный', 'Взрослый')),

Вместимость smallint not null check (Вместимость > 0) default 5000, [ФИО ответственного] varchar(30) not null)

GO

CREATE TABLE Скупщики

([Идентификационный номер] int not null identity (1, 1), Наименование varchar(30) not null, [Контактный телефон] varchar (18) not null check ([Контактный телефон]

like '+7([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'), Адрес varchar(60) not null, Cтатус varchar(20) not null check

(Cтатус IN ('Физическое лицо', 'Юридическое лицо')))

GO

ALTER TABLE Скупщики

ADD CONSTRAINT PK\_Скупщики primary key ([Идентификационный номер])

GO

ALTER TABLE Птицы

ADD CONSTRAINT PK\_Птицы primary key ([Номер вида])

GO

ALTER TABLE Птичники

ADD CONSTRAINT PK\_Птичники primary key (Номер)

GO

ALTER TABLE Продажи

ADD CONSTRAINT PK\_Продажи primary key (Скупщик, [Дата продажи])

GO

ALTER TABLE Продажи

ADD CONSTRAINT FK\_Продажи\_Скупщики foreign

key (Скупщик) references Скупщики([Идентификационный номер])

GO

ALTER TABLE Продажи

ADD CONSTRAINT FK\_Продажи\_Птицы foreign

key ([Вид птицы]) references Птицы([Номер вида])

GO

ALTER TABLE Птицы

ADD CONSTRAINT FK\_Птицы\_Птичники foreign

key ([Закреплённый птичник]) references Птичники(Номер)

GO

DBCC CHECKIDENT('Скупщики', RESEED, 0) --сброс счетчика

GO

INSERT INTO Скупщики VALUES

('ИП Воробьева А. С.', '+7(930) 344-64-24', 'г. Екатеринбург, ул. Комарова, 31, В. 96', 'Физическое лицо'),

('ИП Кузнецова А. В.', '+7(920) 513-22-99', 'г. Самара, ул. Полевая, 29, В. 42', 'Физическое лицо'),

('АО «Сотрудник»', '+7(900) 888-37-43', 'г. Воронеж, ул. Шоссейная, 40, оф. 10', 'Юридическое лицо'),

('ПАО «Бесконечная таблица»', '+7(920) 694-99-90', 'г. Челябинск, ул. Строителей, 11, оф. 32', 'Юридическое лицо'),

('ООО «Чувство»', '+7(920) 797-10-01', 'г. Уфа, ул. Заводская, 19, оф. 13', 'Юридическое лицо'),

('ИП Морозов Р. Н.', '+7(915) 688-12-21', 'г. Новосибирск, ул. Коммунистическая, 11, В. 68', 'Физическое лицо'),

('ООО «Структура»', '+7(900) 100-16-61', 'г. Тольятти, ул. Октябрьская, 48, оф. 9', 'Юридическое лицо'),

('ИП Суворов К. Т.', '+7(903) 900-01-05', 'г. Казань, ул. Майская, 50, В. 6', 'Физическое лицо'),

('ИП Горшкова Е. Д.', '+7(910) 963-56-34', 'г. Ижевск, ул. Восточная, 16, В. 17', 'Физическое лицо'),

('ООО «Замечательная вода»', '+7(900) 970-55-55', 'г. Казань, ул. Родниковая, 5, оф. 61', 'Юридическое лицо')

GO

DBCC CHECKIDENT('Птичники', RESEED, 0) --сброс счетчика

GO

INSERT INTO Птичники VALUES

('Брудергауз', 5500, 'Никитин Александр Тимурович'),

('Батарейный цех', 22000, 'Серебрякова Ева Давидовна'),

('Акклиматизатор', 16000, 'Григорьев Даниил Кириллович'),

('Репродукторный', 8200, 'Князев Даниил Фёдорович'),

('Взрослый', 3900, 'Кузнецова Елизавета Егоровна')

GO

DBCC CHECKIDENT('Птицы', RESEED, 0) --сброс счетчика

GO

INSERT INTO Птицы VALUES

('Кохинхин', 3, 3.800, 550.00),

('Плимутрок',3, 3.000, 400.00),

('Араукана', 2, 2.000, 410.00),

('Глазчатая индейка', 2, 4.500, 690.00),

('Леггорн', 1, 2.200, 480.00),

('Юрловская голосистая', 4, 3.200, 430.00),

('Брама светлая', 0, 2.900, 380.00),

('Сибрайт', 4, 2.650, 510.00),

('Иокогама', 1, 2.200, 500.00),

('Брекель', 0, 3.600, 550.00)

GO

INSERT INTO Продажи VALUES

(5, '2022-05-01', 3, 30, 93150.00),

(2, '2022-04-23', 5, 1200, 1651200.00),

(1, '2022-07-11', 4, 28, 29568.00),

(6, '2022-06-06', 9, 950, 1881000.00),

(9, '2022-05-29', 0, 660, 1379400.00),

(7, '2022-09-01', 2, 15, 12300.00),

(0, '2022-06-12', 8, 42, 46200.00),

(5, '2022-05-11', 3, 10, 31050.00),

(3, '2022-08-26', 6, 900, 991800.00),

(4, '2022-07-03', 1, 1320, 1584000.00)

GO

# Разработка основных операций реляционной алгебры

1. Вывести полную информацию о скупщиках, которые имеют статус «Физическое лицо».

R = Скупщики WHERE Статус = 'Физическое лицо'

1. Вывести наименование вида и цену мяса птиц.

R = Птицы [Наименование\_вида, Цена\_мяса\_данного\_вида\_за\_ килограмм]

1. Вывести вид птичников, вместимость которых больше 10000.

R = Птичники [Вид\_птичника] WHERE Вместимость > 10000

1. Вывести наименование скупщика, его статус и дату продажи.

R1 = Продажи RENAME Скупщик AS Идентификационный\_номер

R2 = (R1 JOIN Скупщики) [Наименование, Статус, Дата\_продажи]

1. Вывести полную информацию о птицах с названием закрепленного птичника.

R1 = Птицы RENAME Закреплённый\_птичник AS Номер

R2 = (R1 JOIN Птичники) [Номер\_вида, Наименование\_вида, Вид\_птичника, Средний\_вес\_птиц\_данного\_вида, Цена\_мяса\_данного\_вида\_ за\_килограмм]

1. Вывести информацию о птичниках «Взрослый» и «Брудергауз».

R1 = Птичники WHERE Вид\_птичника = 'Взрослый'

R2 = Птичники WHERE Вид\_птичника = 'Брудергауз'

R3 = R1 UNION R2

1. Вывести наименование скупщиков, которые не совершали ни одной покупки.

R1 = Скупщики (Идентификационный\_номер)

R2 = Продажи [Скупщик] RENAME Скупщик AS Идентификационный\_ номер

R3 = R1 EXCEPT R2

R4 = (R3 JOIN Скупщики) [Наименование]

1. Вывести информацию о скупщиках, которые не приобретали Араукану.

R1 = Скупщики [Идентификационный\_номер]

R2 = Продажи [Вид\_птицы] RENAME Вид\_птицы AS Номер\_вида

R3 = R2 JOIN Птицы

R4 = (R3 WHERE Наименование\_вида = 'Араукана') [Скупщик]

R5 = R4 RENAME Скупщик AS Идентификационный\_номер

R6 = R1 EXCEPT R5

R7 = R6 JOIN Скупщики

1. Вывести наименование скупщиков, которые приобретали Иокогану, но не приобретали Кохинхина.

R1 = Продажи RENAME Вид\_птицы AS Номер\_вида

R2 = R1 JOIN Птицы

R3 = R2 WHERE Наименование\_вида = 'Иокогана'

R4 = R2 WHERE Наименование\_вида = 'Кохинхин'

R5 = R3 EXCEPT R4

R6 = R5 RENAME Скупщик AS Идентификационный\_номер

R7 = (R6 JOIN Скупщики) [Наименование]

1. Вывести наименование и контактный телефон скупщиков, которые не приобретали ни Сибрайта, ни Брекель.

R1 = Скупщики [Идентификационный\_номер]

R2 = Продажи RENAME Вид\_птицы AS Номер\_вида

R3 = R2 JOIN Птицы

R4 = (R3 WHERE Наименование\_вида = 'Сибрайт') [Скупщик]

R5 = (R3 WHERE Наименование\_вида = 'Брекель') [Скупщик]

R6 = R4 RENAME Скупщик AS Идентификационный\_номер

R7 = R5 RENAME Скупщик AS Идентификационный\_номер

R8 = (R1 EXCEPT R6) EXCEPT R7

R9 = (R8 JOIN Скупщики) [Наименование, Контактный\_телефон]

1. Вывести информацию о скупщиках, которые приобрели все виды птиц.

R1 = Птицы [Номер\_вида]

R2 = Продажи RENAME Вид\_птицы AS Номер\_вида

R3 = R2 [Скупщик, Номер\_вида]

R4 = R3 DIVIDEBY R1

R5 = R4 RENAME Скупщик AS Идентификационный\_номер

R6 = R5 JOIN Скупщики

1. Вывести информацию о продажах в следующем виде: Наименование скупщика, дата продажи, наименование вида птицы, количество проданных птиц данного вида, общая стоимость.

R1 = Продажи RENAME Скупщик AS Идентификационный\_номер

R2 = R1 JOIN Скупщики

R3 = Скупщики RENAME Наименование AS Наименование\_скупщика

R4 = Продажи RENAME Вид\_птицы AS Номер\_вида

R5 = R4 JOIN Птицы

R6 = Птицы RENAME Наименование\_вида AS Наименование\_вида\_ птицы

R7 = Продажи [Наименование\_скупщика, Дата\_продажи, Наименование\_ вида\_птицы, Количество\_проданых\_птиц\_данного\_вида, Общая\_стоимость]

1. Вывести наименование юридических лиц и вид приобретённых птиц, если стоимость покупки составила более 1000000 рублей.

R1 = Продажи WHERE Общая\_стоимость > 1000000

R2 = Скупщики WHERE Статус = 'Юридическое лицо'

R3 = Продажи RENAME Скупщик AS Идентификационный\_номер

R4 = R3 JOIN R2 RENAME Наименование AS Наименование\_ юридического\_лица

R5 = Продажи RENAME Вид\_птицы AS Номер\_вида

R6 = (R5 JOIN Птицы) RENAME Наименование\_вида AS Вид\_птицы

R7 = Продажи [Наименование\_юридического\_лица, Вид\_птицы]

1. Вывести информацию о скупщиках, приобретавших птиц из 2 и 4 птичников.

R1 = Птицы WHERE Закреплённый птичник = 2

R2 = Продажи RENAME Вид\_птицы AS Номер\_вида

R3 = R1 JOIN Птицы

R4 = Продажи [Скупщик]

R5 = Птицы WHERE Закреплённый птичник = 4

R6 = Продажи RENAME Вид\_птицы AS Номер\_вида

R7 = R6 JOIN Птицы

R8 = Продажи [Скупщик]

R9 = R4 INTERSECT R8

R10 = R9 RENAME Скупщик AS Идентификационный\_номер

R11 = R10 JOIN Скупщики

1. Вывести наименование скупщиков, приобретавших только Плимутрока.

R1 = Продажи RENAME Вид\_птицы AS Номер\_вида

R2 = R1 JOIN Птицы

R3 = Продажи [Скупщик, Наименование\_вида]

R4 = R3 WHERE Наименование\_вида = 'Плимутрок'

R5 = R3 WHERE Наименование\_вида ≠ 'Плимутрок'

R6 = R4 EXCEPT R5

R7 = R6 RENAME Скупщик AS Идентификационный\_номер

R8 = (R7 JOIN Скупщики) [Наименование]

# Разработка запросов к БД для реализации информационных потребностей пользователя

Для созданной базы данных разработать на языке Transact-SQL следующие типы запросов:

1. Запросы с выбором строк по условию:

* запрос с использованием операторов сравнения;

Вывести виды птиц, цена мяса за килограмм которых больше 500 рублей (рисунок 1).

SELECT [Наименование вида]

FROM Птицы

WHERE ([Цена мяса данного вида птицы за килограмм] > 500)

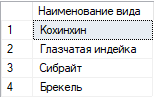


Рисунок 1 ‒ Запрос 1

* запрос с использованием логических операторов AND, OR и NOT;

Вывести информацию о птичниках взрослый и акклиматизатор (рисунок 2).

SELECT \*

FROM Птичники

WHERE ([Вид птичника] = 'Акклиматизатор') OR ([Вид птичника] = 'Взрослый')



Рисунок 2 ‒ Запрос 2

Вывести информацию о птичнике взрослый, если его вместимость больше 12000 птиц (рисунок 3).

SELECT \*

FROM Птичники

WHERE ([Вид птичника] = 'Акклиматизатор') AND ([Вместимость] > 12000)



Рисунок 3 ‒ Запрос 3

Вывести информацию обо всех птичниках, кроме батарейного цеха (рисунок 4).

SELECT \*

FROM Птичники

WHERE NOT [Вид птичника] = 'Батарейный цех'

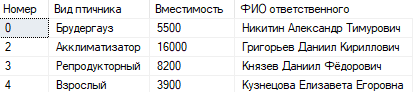


Рисунок 4 ‒ Запрос 4

* запрос на использование комбинации логических операторов;

Вывести вид птиц, средний вес которых больше 2 килограммов и меньше 3 килограммов или цена мяса за килограмм которых меньше 500 рублей (рисунок 5).

SELECT [Наименование вида]

FROM Птицы

WHERE ([Средний вес птиц данного вида] > 2) AND ([Средний вес птиц данного вида] < 3)

OR ([Цена мяса данного вида птицы за килограмм] < 500)

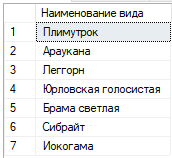


Рисунок 5 ‒ Запрос 5

* запрос на использование выражений над столбцами;

Вывести вид птиц, цена мяса за 5 килограммов которых не превышает 2000 рублей (рисунок 6).

SELECT [Наименование вида]

FROM Птицы

WHERE ([Цена мяса данного вида птицы за килограмм]\*5 < 2000)



Рисунок 6 ‒ Запрос 6

* запрос с проверкой на принадлежность множеству;

Вывести наименование скупщиков, идентификационный номер которых 3 и 9 (рисунок 7).

SELECT [Наименование]

FROM Скупщики

WHERE ([Идентификационный номер] IN (3, 9))



Рисунок 7 ‒ Запрос 7

* запрос с проверкой на принадлежность диапазону значений;

Вывести виды птиц, цена мяса за килограмм которых больше 400 и меньше 500 рублей (рисунок 8).

SELECT [Наименование вида]

FROM Птицы

WHERE ([Цена мяса данного вида птицы за килограмм] BETWEEN 401 AND 499)

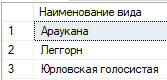


Рисунок 8 ‒ Запрос 8

* запрос с проверкой на соответствие шаблону;

Вывести информацию о продажах, совершённых в 2022 году (рисунок 9).

SELECT \*

FROM Продажи

WHERE ([Дата продажи] LIKE '2022%')



Рисунок 9 ‒ Запрос 9

* запрос с проверкой на неопределенное значение.

Вывести информацию о птичниках, у которых отсутствует ответственное лицо (рисунок 10).

SELECT \*

FROM Птичники

WHERE ([ФИО ответственного] IS NULL)



Рисунок 10 ‒ Запрос 10

1. Запросы на теоретико-множественные реляционные операции:

* запрос на объединение таблиц;

Вывести номер, вид птичника и ФИО ответственного; если ФИО не указано, вывести в соответствующую запись «Не назначен» (рисунок 11).

SELECT Номер, [Вид птичника], [ФИО ответственного]

FROM Птичники

WHERE ([ФИО ответственного] IS NOT NULL)

UNION

SELECT Номер, [Вид птичника], 'Не назначен'

FROM Птичники

WHERE ([ФИО ответственного] IS NULL)

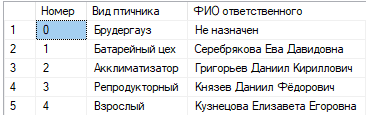


Рисунок 11 ‒ Запрос 11

* запрос на пересечение таблиц;

Вывести вид птиц, цена мяса за килограмм которых меньше 400 рублей и средний вес которых меньше 3 килограммов (рисунок 12).

SELECT [Наименование вида]

FROM Птицы

WHERE ([Цена мяса данного вида птицы за килограмм] < 400)

INTERSECT

SELECT [Наименование вида]

FROM Птицы

WHERE ([Средний вес птиц данного вида] < 3)



Рисунок 12 ‒ Запрос 12

* запрос на вычитание таблиц;

Вывести номер вида птиц, которых еще не приобретали (рисунок 13).

SELECT [Номер вида]

FROM Птицы

EXCEPT

SELECT [Вид птицы]

FROM Продажи



Рисунок 13 ‒ Запрос 13

* запрос с использованием декартового произведения таблиц;

Вывести виды птиц и виды птичников, в которых они содержатся (рисунок 14).

SELECT [Наименование вида], [Вид птичника]

FROM Птичники, Птицы

WHERE Птицы.[Закреплённый птичник] = Птичники.Номер

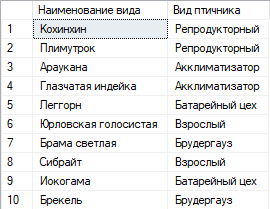


Рисунок 14 ‒ Запрос 14

1. Многотабличные запросы:

* запрос с использованием соединения двух таблиц по равенству и условием отбора;

Вывести виды птиц, которые содержатся в репродукторном птичнике (рисунок 15).

SELECT [Наименование вида], [Вид птичника]

FROM Птицы inner join Птичники on Птицы.[Закреплённый птичник] = Птичники.Номер

WHERE [Вид птичника] = 'Репродукторный'



Рисунок 15 ‒ Запрос 15

* запрос с использованием соединения трех таблиц по равенству и условием отбора;

Вывести информацию о продажах птиц, закреплённых за акклиматизатором, с указанием вида птиц и вида птичника (рисунок 16).

SELECT Скупщик, [Дата продажи], [Наименование вида], [Количество проданных птиц данного вида], [Вид птичника], [Общая стоимость]

FROM Продажи inner join Птицы on Продажи.[Вид птицы] = Птицы.[Номер вида]

inner join Птичники on Птицы. [Закреплённый птичник] = Птичники.Номер

WHERE [Вид птичника] = 'Акклиматизатор'

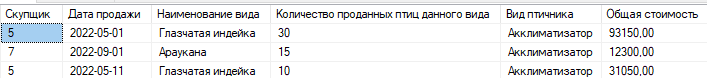


Рисунок 16 ‒ Запрос 16

* запрос с использованием соединения по четырем таблицам;

Вывести информацию о продажах с указанием наименования скупщика, вида приобретённых птиц и закреплённых за ними птичников, если сумма покупки составила более 500000 рублей (рисунок 17).

SELECT Наименование, [Дата продажи], [Наименование вида], [Вид птичника], [Количество проданных птиц данного вида], [Цена мяса данного вида птицы за килограмм], [Общая стоимость]

FROM Скупщики inner join Продажи on Скупщики.[Идентификационный номер] = Продажи.Скупщик

inner join Птицы on Птицы.[Номер вида] = Продажи.[Вид птицы] inner join Птичники

on Птичники.Номер = Птицы.[Закреплённый птичник]

WHERE [Общая стоимость]>500000

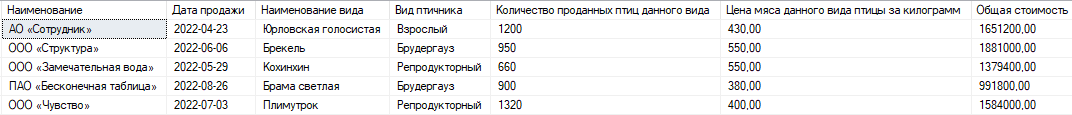


Рисунок 17 ‒ Запрос 17

* запрос с использованием левого внешнего соединения;

Вывести виды птиц, указав закреплённые птичники (рисунок 18).

SELECT [Наименование вида], [Вид птичника]

FROM Птицы left join Птичники on Птицы.[Закреплённый птичник] = Птичники.Номер

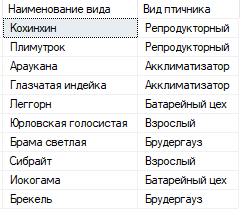


Рисунок 18 ‒ Запрос 18

* запрос на использование правого внешнего соединения;

Вывести наименование скупщика и стоимость его покупок (рисунок 19).

SELECT [Наименование], [Общая стоимость]

FROM Продажи right join Скупщики on Продажи.[Скупщик] = Скупщики.[Идентификационный номер]

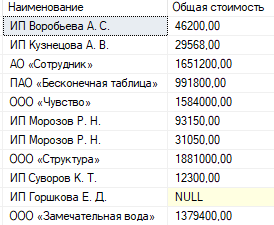


Рисунок 19 ‒ Запрос 19

* запрос с использованием функции COUNT;

Вывести количество продаж юридическим лицам (рисунок 20).

SELECT count (Продажи.[Общая стоимость]) AS 'Кол-во продаж юр. лицам'

FROM Продажи inner join Скупщики on Продажи.Скупщик=Скупщики.[Идентификационный номер] WHERE Cтатус = 'Юридическое лицо'



Рисунок 20 ‒ Запрос 20

* запрос с использованием функции SUM;

Вывести общую стоимость всех продаж (рисунок 21).

SELECT SUM (Продажи.[Общая стоимость]) AS 'Общая стоимость всех продаж'

FROM Продажи



Рисунок 21 ‒ Запрос 21

* 3 запроса с использованием временных функций;

Вывести дни, в которые были совершены продажи Кузнецовой А. В. (рисунок 22).

SELECT DAY (Продажи.[Дата продажи]) AS 'День продажи'

FROM Продажи inner join Скупщики on Продажи.Скупщик=Скупщики.[Идентификационный номер]

WHERE Наименование = 'ИП Кузнецова А. В.'



Рисунок 22 ‒ Запрос 22

Вывести месяцы, в которые были совершены продажи физическим лицам (рисунок 23).

SELECT MONTH (Продажи.[Дата продажи]) AS 'Месяц продажи'

FROM Продажи inner join Скупщики on Продажи.Скупщик=Скупщики.[Идентификационный номер] WHERE Cтатус = 'Юридическое лицо'

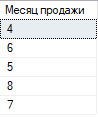


Рисунок 23 ‒ Запрос 23

Вывести годы, в которые были совершены все продажи (рисунок 24).

SELECT DISTINCT YEAR (Продажи.[Дата продажи]) AS 'Год продаж'

FROM Продажи inner join Скупщики on Продажи.Скупщик=Скупщики.[Идентификационный номер]



Рисунок 24 ‒ Запрос 24

* запрос с использованием группировки по одному столбцу;

Вывести статус и минимальную сумму продажи физическим и юридическим лицам (рисунок 25).

SELECT Cтатус, MIN(Продажи.[Общая стоимость]) AS 'Минимальная сумма продажи'

FROM Продажи inner join Скупщики on Продажи.Скупщик=Скупщики.[Идентификационный номер] group by Cтатус



Рисунок 25 ‒ Запрос 25

* запрос на использование группировки по нескольким столбцам;

Вывести статус и минимальную сумму продажи физическим и юридическим лицам по каждому птичнику (рисунок 26).

SELECT Cтатус, [Вид птичника], MIN(Продажи.[Общая стоимость]) AS 'Минимальная сумма продажи'

FROM Продажи inner join Скупщики on Продажи.Скупщик = Скупщики.[Идентификационный номер]

inner join Птицы on Птицы.[Номер вида] = Продажи.[Вид птицы] inner join Птичники on Птичники.Номер = Птицы.[Закреплённый птичник]

group by Cтатус, [Вид птичника]

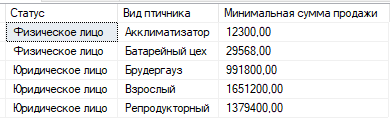


Рисунок 26 ‒ Запрос 26

* запрос с использованием условия отбора групп HAVING;

Вывести статус и минимальную сумму продажи физическим лицам по каждому птичнику (рисунок 27).

SELECT Cтатус, [Вид птичника], MIN(Продажи.[Общая стоимость]) AS 'Минимальная сумма продажи'

FROM Продажи inner join Скупщики on Продажи.Скупщик = Скупщики.[Идентификационный номер]

inner join Птицы on Птицы.[Номер вида] = Продажи.[Вид птицы] inner join Птичники on Птичники.Номер = Птицы.[Закреплённый птичник]

group by Cтатус, [Вид птичника]

HAVING Cтатус = 'Физическое лицо'



Рисунок 27 ‒ Запрос 27

* запрос с использованием сортировки по столбцу;

Вывести статус и минимальную сумму продажи физическим и юридическим лицам по каждому птичнику в порядке убывания суммы продаж (рисунок 28).

SELECT Cтатус, [Вид птичника], MIN(Продажи.[Общая стоимость]) AS 'Минимальная сумма продажи'

FROM Продажи inner join Скупщики on Продажи.Скупщик = Скупщики.[Идентификационный номер]

inner join Птицы on Птицы.[Номер вида] = Продажи.[Вид птицы] inner join Птичники on Птичники.Номер = Птицы.[Закреплённый птичник]

group by Cтатус, [Вид птичника]

ORDER BY MIN(Продажи.[Общая стоимость]) DESC

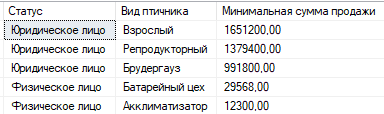


Рисунок 28 ‒ Запрос 28

1. Запросы на модификацию таблиц:

* запрос на добавление новых данных в таблицу;

Добавить 3 новые строки в таблицу Птицы (рисунок 29).

INSERT INTO Птицы

VALUES

('Серая кряква', 1, 2.100, 650.00),

('Чёрная кряква', 1, 2.200, 640.00),

('Шилохвость', 1, 2.500, 690.00)

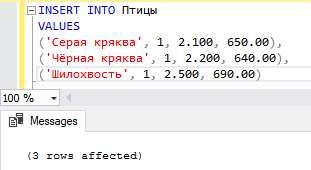


Рисунок 29 ‒ Запрос 29

* запрос на добавление новых данных по результатам выполненного запроса;

USE Птицеферма

CREATE TABLE [Птица "Кохинхин"]

([Номер вида] int not null, [Наименование вида] varchar(30) not null, [Закреплённый птичник] int not null,

[Средний вес птиц данного вида] numeric (5,3) not null check ([Средний вес птиц данного вида] >0) default 1.500,

[Цена мяса данного вида птицы за килограмм] money not null check ([Цена мяса данного вида птицы за килограмм] >0) default 450.00)

GO INSERT INTO [Птица "Кохинхин"]

SELECT \*

FROM Птицы

WHERE [Наименование вида] = 'Кохинхин'

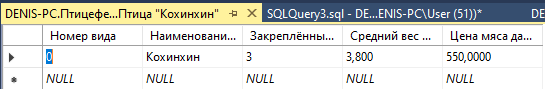


Рисунок 30 ‒ Запрос 30

* запрос на обновление существующих данных в таблице;

Изменить цену мяса птицы Кохинхин на 520 рублей за килограмм (рисунок 31).

UPDATE [Птица "Кохинхин"]

SET [Цена мяса данного вида птицы за килограмм] = 520

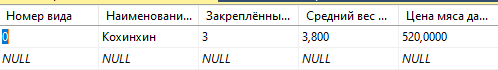


Рисунок 31 ‒ Запрос 31

* запрос на обновление существующих данных по результатам подзапроса во фразе WHERE;

Изменить средний вес птицы Глазчатая индейка на 4,6 килограмма (рисунок 32).

UPDATE Птицы

SET [Средний вес птиц данного вида] = 4.600

WHERE [Наименование вида] IN (select [Наименование вида] from Птицы where [Цена мяса данного вида птицы за килограмм] = 690)

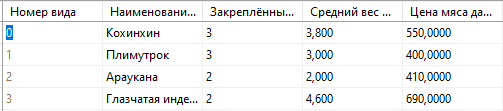


Рисунок 32 ‒ Запрос 32

* запрос на удаление существующих данных;

Удалить данные из таблицы Птица «Кохинхин» (рисунок 33).

DELETE

FROM [Птица "Кохинхин"]

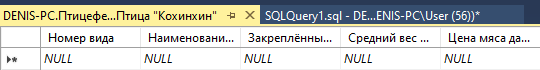


Рисунок 33 ‒ Запрос 33

* запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц.

Удалить информацию о продажах скупщику, контактный номер которого +7(900) 888-37-43 (рисунок 34).

DELETE Продажи

FROM Продажи inner join Скупщики on Продажи.Скупщик = Скупщики.[Идентификационный номер]

WHERE [Контактный телефон] = '+7(900) 888-37-43'



Рисунок 34 ‒ Запрос 34

* запрос на удаление существующих данных по результатам подзапроса во фразе WHERE;

Удалить информацию о птице Шилохвость (рисунок 35).

DELETE

FROM Птицы WHERE [Наименование вида] IN (select [Наименование вида] from Птицы

WHERE [Цена мяса данного вида птицы за килограмм]=690 and [Средний вес птиц данного вида]=2.500)

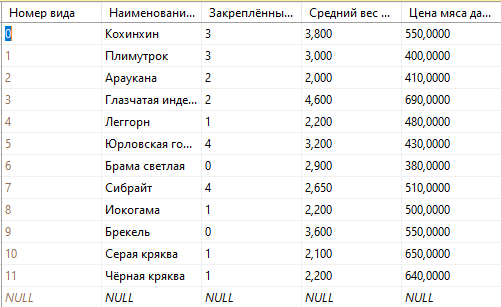


Рисунок 35 ‒ Запрос 35

1. Многотабличные запросы с использованием подзапросов:

* 3 запроса с использованием операций сравнения;

Вывести вид птичника, за которым закреплён Леггорн (рисунок 36).

SELECT [Вид птичника]

FROM Птичники WHERE Номер = (SELECT [Закреплённый птичник] FROM Птицы WHERE [Наименование вида] = 'Леггорн')



Рисунок 36 ‒ Запрос 36

Вывести вид птицы, продажа которой включала 1320 особей (рисунок 37).

SELECT [Наименование вида]

FROM Птицы WHERE [Номер вида] = (SELECT [Вид птицы] FROM Продажи WHERE [Количество проданных птиц данного вида] = 1320)



Рисунок 37 ‒ Запрос 37

Вывести вид птицы, дата продажи которой была 2022-08-26 (рисунок 38).

SELECT [Наименование вида]

FROM Птицы WHERE [Номер вида] = (SELECT [Вид птицы] FROM Продажи WHERE [Дата продажи] = '2022-08-26')



Рисунок 38 ‒ Запрос 38

* запрос с использованием операции ANY;

Вывести наименования скупщиков, совершавших покупку (рисунок 39).

SELECT [Наименование]

FROM Скупщики

WHERE [Идентификационный номер] = ANY(SELECT Скупщик FROM Продажи)

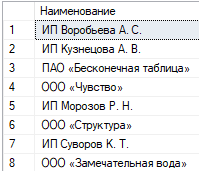


Рисунок 39 ‒ Запрос 39

* запрос с использованием операции ALL;

Вывести виды птиц, закреплённых за акклиматизатором (рисунок 40).

SELECT [Наименование вида]

FROM Птицы WHERE [Закреплённый птичник]= ALL(SELECT Номер FROM Птичники WHERE [Вид птичника] = 'Акклиматизатор')



Рисунок 40 ‒ Запрос 40

* запрос с использованием операции IN;

Вывести вид птичника, у которого нет ответственного лица (рисунок 41).

SELECT [Вид птичника] FROM Птичники

WHERE Номер IN (SELECT Номер FROM Птичники WHERE [ФИО ответственного] IS NULL)



Рисунок 41 ‒ Запрос 41

* запрос с использованием операции EXISTS;

Вывести номера скупщиков, совершавших покупку (рисунок 42).

select distinct Скупщик from Продажи where exists(select \* from Скупщики where Скупщики.[Идентификационный номер]=Продажи.Скупщик)

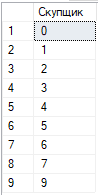


Рисунок 42 ‒ Запрос 42

* запрос с использованием двух вложенных друг в друга подзапросов;

Вывести вид птичника, за которым закреплён вид птицы, дата продажа которого была 2022-05-01 (рисунок 43).

SELECT [Вид птичника]

FROM Птичники

WHERE Номер = (SELECT [Закреплённый птичник] FROM Птицы WHERE [Номер вида] = (SELECT [Вид птицы] FROM Продажи WHERE [Дата продажи] = '2022-05-01'))



Рисунок 43 ‒ Запрос 43

* запрос с использованием трех вложенных друг в друга подзапросов;

Вывести информацию о птичниках, которые закреплены за птицами, приобретёнными ИП Воробьевой А. С. (рисунок 44).

SELECT \*

FROM Птичники

WHERE [Номер] = (SELECT [Закреплённый птичник] FROM Птицы WHERE [Номер вида] = (SELECT [Вид птицы] FROM Продажи WHERE [Скупщик] = (SELECT [Идентификационный номер] FROM Скупщики WHERE Наименование = 'ИП Воробьева А. С.')))



Рисунок 44 ‒ Запрос 44

1. Запросы на изменение структуры существующей таблицы:

* запрос на добавление нового столбца к таблице;

Добавить в таблицу Птицы столбец «Цена оптом» (рисунок 45).

ALTER TABLE Птицы

ADD [Цена оптом] money

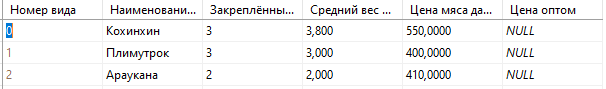


Рисунок 45 ‒ Запрос 45

* запрос на добавление нового ограничения к таблице;

Добавить ограничение по умолчанию в таблицу Птицы, устанавливающее значение цены оптом в 450 рублей (рисунок 46).

ALTER TABLE Птицы

ADD CONSTRAINT OPT\_DEFAULT DEFAULT 450 FOR [Цена оптом]

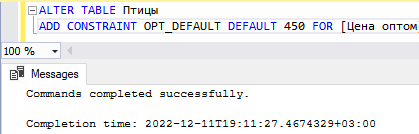


Рисунок 46 ‒ Запрос 46

* запрос на удаление нового столбца к таблице;

Удалить столбец Цена оптом из таблицы Птицы (рисунок 47).

ALTER TABLE Птицы

DROP COLUMN [Цена оптом]

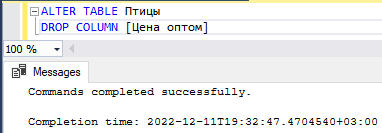


Рисунок 47 ‒ Запрос 47

* запрос на удаление нового ограничения к таблице;

Удалить ограничение столбца Цена оптом (рисунок 48).

ALTER TABLE Птицы

DROP CONSTRAINT OPT\_DEFAULT

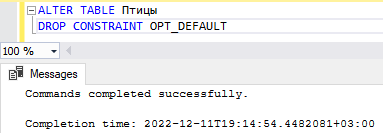


Рисунок 48 ‒ Запрос 48

* запрос на изменение типа данных в таблице;

Изменение типа данных столбца Наименование вида в таблице Птицы (рисунок 49).

ALTER TABLE Птицы

ALTER COLUMN [Наименование вида] varchar (40)

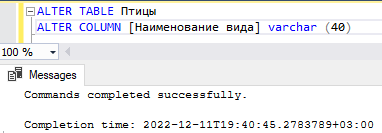


Рисунок 49 ‒ Запрос 49

* запрос на добавление нового первичного ключа к таблице с сохранением целостности таблицы;

Добавить первичный ключ Дата продажи в таблицу Продажи (рисунок 50).

ALTER TABLE [Птица "Кохинхин"]

ADD CONSTRAINT PK\_Кохинхин

PRIMARY KEY ([Номер вида])

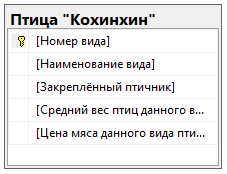


Рисунок 50 ‒ Запрос 50

## ЗАКЛЮЧЕНИЕ

В результате с помощью СУБД Microsoft SQL Server 2018 была создана база данных, в которой хранится информация о птичниках и птицах, содержащихся в них; информация о скупщиках и продажах. На её основе разработаны базовые операции реляционной алгебры. Также были осуществлены манипуляции с БД, такие как: добавление и удаление данных, атрибутов, ограничений и таблиц. Актуальная база данных позволяет пользователю выводить необходимую информацию в требуемом виде с помощью запросов, примеры которых также были обозначены в ходе работы.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

* Кара-Ушанов, В. Ю. SQL — язык реляционных баз данных: учебное пособие / В.Ю. Кара-Ушанов.— Екатеринбург: Изд-во Урал. ун-та, 2016.— 156 с. ISBN 978-5-7996-1622-9.
* Власова, О. В. SQL: учебное пособие / О. В. Власова; Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль: ЯрГУ, 2011. – 136 с. ISBN 978-5-8397-0806-8.
* Моргунов, Е. П. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с. ISBN 978-5-9775-4022-3.
* Осетрова, И.С. Разработка баз данных в MS SQL Server 2014. - СПб: Университет ИТМО, 2016. – 114 с.
* Чалышев, М. М. Oracle SQL. 100 шагов от новичка до профессионала. 20 дней новых знаний и практики  /  М. М. Чалышев —  «Издательские решения»,  ISBN 978-5-00-500618-9.
* Маркин, А. В. Программирование на SQL: учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва: Издательство Юрайт, 2019. — 435 с. ISBN 978-5-534-11093-7.