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:

Горнолыжный курорт»

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы 945 |  |  |
| Карасев Е.Б. |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  дата сдачи на проверку, подпись |
|  |  |  |
| Руководитель работы |  |  |
| ассистент кафедры ЭВМ |  |  |
| Тарасова В.Ю. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  оценка | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  дата защиты, подпись |

Рязань 2020

**Содержание**

[Введение 2](#_Toc63965314)

[1 Семантическое описание предметной области 3](#_Toc63965315)

[2 Разработка базы данных 5](#_Toc63965316)

[2.1 Разработка логической модели данных 5](#_Toc63965317)

[2.2 Определение типов, хранимых данных 9](#_Toc63965318)

[2.3 Задание ограничений целостности данных 9](#_Toc63965319)

[2.4 Построение схемы БД 11](#_Toc63965320)

[2.5 Разработка сценария для создания БД и основных объектов структуры БД 11](#_Toc63965321)

[2.6 Разработка сценария для заполнения таблиц БД 12](#_Toc63965322)

[3 Разработка основных операций реляционной алгебры 15](#_Toc63965323)

[4 Разработка запросов к БД для реализации информационных потребностей пользователя 18](#_Toc63965324)

[Заключение 39](#_Toc63965325)

[Список литературы 40](#_Toc63965326)

# Введение

База данных (БД) — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов.

Для изменения и управления данными существуют системы управления базами данных (СУБД). Это комплекс программно-языковых средств, позволяющих создать БД и управлять данными.

Целью данной курсовой работы является создание реляционной базы данных для хранения информации и работы с ней. Предметная область курсовой работы – «Горнолыжный курорт». В базу должны входить различные данные о подъемниках, тарифах, карточка и информация о соответствии подъемника тарифу. Необходимо предусмотреть то, чтобы пользователь мог удобно пользоваться этой базой данных и получать необходимую ему информацию.

# 1 Семантическое описание предметной области

Горнолыжный курорт – это база отдыха для всех людей любого возраста.

В горнолыжном курорте существуют тарифы для отдыхающих. О тарифах требуется хранить следующую информацию: Стоимость, период действия, количество подъемов, льготная скидка.

На каждом горнолыжном курорте есть подъемники. Бывают подъемники: кресельные, гондольные, бугельные и другие. О подъемниках должна храниться информация: наименование, время работы, тип подъемника, высота начальная, высота конечная.

О карточках требуется хранить следующую информацию: тарифа, даты покупки и периода действия.

Отдыхающий, приобретая тариф, получает карточку, где прописан тариф и соответствующий подъемник. В соответствии подъемников тарифу должен быть прописан режим работы. Режим работы может быть: вечернее катание, дневное катание.

# 2 Разработка базы данных

# 2.1 Разработка логической модели данных

***Описание основных сущностей ПО.***

На основании проведенного анализа предметной области «Горнолыжный курорт», можно выделить следующие сущности:

Тарифы, Подъемники, Соответствие подъемников и тарифов, Карточки.

Представим список сущностей предметной области (таблица 1).

Таблица 1. Список сущностей

|  |  |  |
| --- | --- | --- |
| № | Название | Назначение |
| 1 | Тарифы | Перечень тарифов, которые существуют на горнолыжном курорте |
| 2 | Подъемники | Описание подъемников, которые существуют на горнолыжном курорте |
| 3 | Соответствие подъемников и тарифов | Соответствие тарифов и подъемников |
| 4 | Карточки | Информация о тарифе, дате покупки и периоде действия |

Для каждой сущности приведем описание ее атрибутов в виде таблицы. На данном этапе выделяются ключевые (это первичные, внешние и альтернативные ключи) и неключевые атрибуты. Результаты приведены в таблицах 2-5.

Следует отметить, что для сущностей Тарифы, в отличие от других сущностей, нет атрибута, с помощью которого можно идентифицировать конкретный тариф. Поэтому требуется ввести дополнительный атрибут, например, id тарифа, который и будет являться первичным ключом.

Атрибут id тарифа является дополнительным служебным полем, добавленным к уже имеющимся информационным полям таблицы. Такие атрибуты называются суррогатными, их единственное предназначение - служить первичным ключом.

Также мы добавляем в сущности Подъемники, Карточки и Соответствие подъемников и тарифов дополнительные атрибуты, которые после будут являться первичными ключами.

1. В сущность Подъемники – id подъемника;
2. В сущность Карточки – Id тарифа;
3. В сущность Соответствие подъемников и тарифов – id подъемника и id тарифа.

Таблица 2. Список атрибутов таблицы «Подъемники»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК  (Первичный ключ) | id подъемника | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому тарифу.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Наименование |  |
|  | Тип подъемника |  |
|  | Высота начальная |  |
|  | Высота конечная |  |

Таблица 3. Список атрибутов таблицы «Тарифы»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК  (Первичный ключ) | id тарифа | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому пользователю.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Стоимость |  |
|  | Период действия |  |
|  | Количество подъемов |  |
|  | Льготная скидка |  |

Таблица 4. Список атрибутов таблицы «Соответствие подъемников и тарифов»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК  (Первичный ключ) | id тарифа | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому фильму.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
| ПК  (Первичный ключ) | id подъемника | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому фильму.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Режим работы |  |

Таблица 5. Список атрибутов таблицы «Карточки»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК  (Первичный ключ) | Id тарифа | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому пользователю.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Дата покупки |  |
|  | Период действия |  |

***Построение логической модели***

В таблице 6 представлен список связей в предметной области «Горнолыжный курорт».

Таблица 6. Список связей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущности, участвующие в связи | Тип связи | Обоснование |
| 1 | Подъемники – Соответствие подъемников и тарифов | 1:N | Один подъемник может быть в нескольких тарифах, но в одном тарифе может быть в один подъемник. |
| 2 | Тарифы – Соответствие подъемников и тарифов | 1:N | Один подъемник может быть в нескольких тарифах, но в одном тарифе может быть в один подъемник. |
| 3 | Тарифы – Карточки | 1:1 | У отдыхающего может быть одна карточка, соответствующая тарифу. |

Логическая модель предметной области «Горнолыжный курорт» представлена на рисунке 1.

  
Рисунок 1 – Логическая модель предметной области «Горнолыжный курорт»

# 2.2 Определение типов, хранимых данных

Для каждой таблицы приведем список всех атрибутов с указанием типа данных. Результаты приведены в таблицах 7-10.

Таблица 7. Список атрибутов таблицы «Подъемники»

|  |  |
| --- | --- |
| Название | Типы данных |
| Id подъемника | int |
| Название | nchar(20) |
| Тип подъемника | nchar(20) |
| Высота начальная | int |
| Высота конечная | int |

Таблица 8. Список атрибутов таблицы «Тарифы»

|  |  |
| --- | --- |
| Название | Типы данных |
| id тарифа | int |
| Стоимость | nchar(20) |
| Период действия | nchar(20) |
| Количество подъемов | int |
| Льготная скидка | nchar(20) |

Таблица 9. Список атрибутов таблицы «Соответствие подъемников и тарифов»

|  |  |
| --- | --- |
| Название | Типы данных |
| id подъемника | int |
| id тарифа | int |
| Режим работы | nchar(20) |

Таблица 10. Список атрибутов таблицы «Карточки»

|  |  |
| --- | --- |
| Название | Типы данных |
| id тарифа | int |
| Дата покупки | int |
| Период действия | nchar(20) |

# 2.3 Задание ограничений целостности данных

Для каждой таблицы необходимо выделим ограничения, накладываемые на значения атрибутов. Результаты представлены в таблицах 11-14.

Таблица 11. Ограничения таблицы «Подъемники»

|  |  |
| --- | --- |
| Название | Ограничение |
| id подъемника | Первичный ключ. Это уникальное значение, не может принимать NULL-значения.  Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |
| Название | Начинается с букв русского алфавита: А-Я. |
| Тип подъемника | Начинается с букв русского алфавита: А-Я. |
| Высота начальная | Значения записываются в днях, в диапазоне от 1 до 2000. Значение больше 0. |
| Высота конечная | Значение больше 0. По умолчанию 1500. |

Таблица 12. Ограничения таблицы «Тарифы»

|  |  |
| --- | --- |
| Название | Ограничения |
| id тарифа | Первичный ключ. Это уникальное значение, не может принимать NULL-значения.  Может принимать следующие значения: 2\*\*, где \*\*-порядковый номер, начиная с 01. |
| Стоимость | 00,0000; NOT NULL |
| Количество подъемов | От 1 до 10. |
| Период действия | Количество дней. |
| Льготная скидка | Да/Нет. Начинается с букв русского алфавита: А-Я. |

Таблица 13. Ограничения таблицы «Соответствие подъемников и тарифов»

|  |  |
| --- | --- |
| Название | Ограничения |
| id подъемника | Атрибут, входящий в составной первичный ключ. Атрибут, входящий во внешний ключ |

Продолжение таблицы 13.

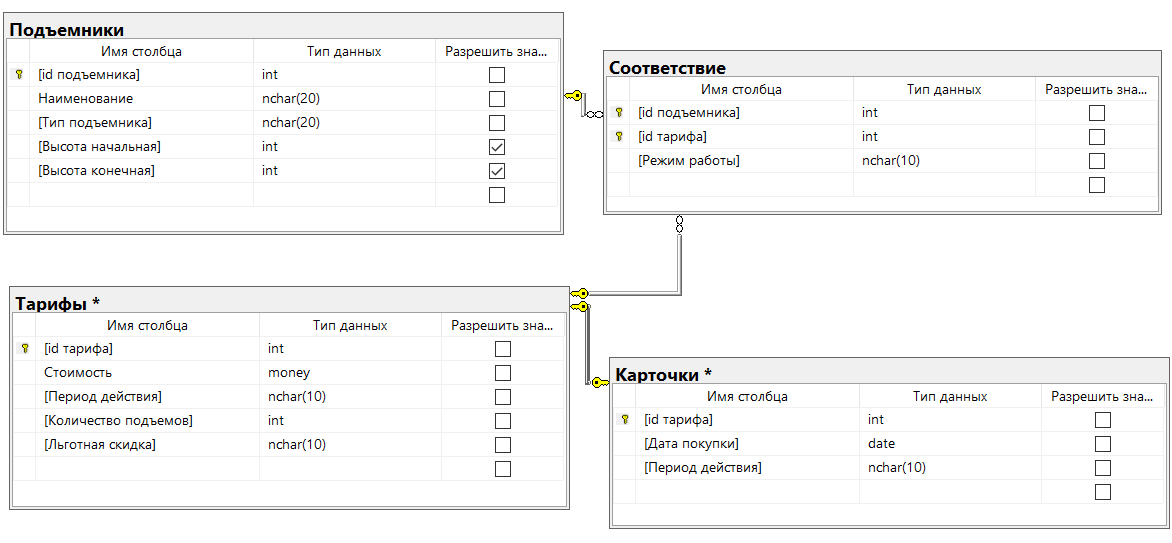
|  |  |
| --- | --- |
| id тарифа | Атрибут, входящий в составной первичный ключ. Атрибут, входящий во внешний ключ |
| Режим работы | Значение по умолчанию: с 10 до 20 часов |

Таблица 14. Ограничения таблицы «Карточки»

|  |  |
| --- | --- |
| Название | Ограничения |
| id тарифа | Внешний ключ. |
| Дата покупки |  |
| Период действия | Показывается количество дней, в которые действует тариф. |

# 2.4 Построение схемы БД

Схема БД представляет собой набор заголовков таблиц. На рисунке 2 представлена схема базы данных «Горнолыжный курорт».

  
Рисунок 2 – Схема базы данных «Горнолыжный курорт»

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

use master

CREATE DATABASE Карасев\_945 ON

(NAME = 'Карасев\_945',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Карасев\_945.mdf',

SIZE=1,

MAXSIZE=10,

FILEGROWTH=1)

LOG ON

(NAME = 'Карасев\_945\_log',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Карасев\_945\_log.ldf',

SIZE = 1,

MAXSIZE = 5,

FILEGROWTH = 1)

GO

use Карасев\_945

CREATE TABLE Подъемники

([id подъемника] int NOT NULL identity(1,1),

Название nchar(20) NOT NULL check (Название like '[А-Я]%'),

[Тип подъемника] nchar(20) NOT NULL check ([Тип подъемника] in ('Кресельный', 'Гондольный', 'Бугельный', 'Ленточный', 'Кабинный')),

[Высота начальная] int check ([Высота начальная]>=0) DEFAULT 0,

[Высота конечная] int check ([Высота конечная] >0) DEFAULT 1500,

CONSTRAINT PK\_POD

PRIMARY KEY ([id подъемника]))

GO

CREATE TABLE Тарифы

([id тарифа] int PRIMARY KEY,

Стоимость money check (Стоимость >= 1700) DEFAULT 1700,

[Количество подъемов] int check ([Количество подъемов] between 1 and 10),

[Период действия] nchar(10) NOT NULL,

[Льготная скидка] nchar(10) NOT NULL check ([Льготная скидка] like '[А-Я]%'))

GO

CREATE TABLE [Соответствие подъемников и тарифов]

([id подъемника] int,

[id тарифа] int,

[Режим работы] nchar(20) not null default 'с 10 до 20 часов',

CONSTRAINT PK\_sootv

PRIMARY KEY ([id подъемника], [id тарифа]))

GO

CREATE TABLE Карточки

([id тарифа ] int,

[Дата покупки] date not null unique,

[Период действия] nchar(20) NOT NULL)

GO

ALTER TABLE Карточки

ADD CONSTRAINT FK\_тариф\_карточки FOREIGN KEY ([id тарифа]) REFERENCES [Тарифы]([id тарифа]) ON DELETE CASCADE

GO

# 2.6 Разработка сценария для заполнения таблиц БД

use Карасев\_945

go

insert into Тарифы ([id тарифа],Стоимость,[Количество подъемов],[Период действия],[Льготная скидка])

values

(201, 1800, 3,'2 дня','нет'),

(202, default, 5,'3 дня','нет'),

(203, 3000, 9,'4 дня','нет'),

(204, 2500, 8,'3 дня','есть'),

(205, default, 4,'2 дня','нет'),

(206, 1999, 5,'3 дня','нет'),

(207, 3100, 10,'5 дней','есть'),

(208, 3000, 9,'3 дня','нет'),

(209, default, 2,'1 день','нет'),

(210,2700, 9,'4 дня','нет')

use Карасев\_945

go

insert into Подъемники (Название,[Тип подъемника],[Высота начальная],[Высота конечная])

values

('Красивый', 'Ленточный',1, default),

('Опасный', 'Кабинный', 0, 1600),

('Длинный', 'Бугельный', 3, 1300),

('Короткий','Гондольный',2, 1900),

('Спуск', 'Кресельный', 0,2500),

('Ласковый', 'Ленточный',0,1700),

('Грозный', 'Гондольный',null,2100),

('Поляна','Бугельный',default, 2000),

('Сочи','Кресельный',0, 1400),

('Роза Хутор', 'Кабинный',1, 1200)

use Карасев\_945

go

insert into [Соответствие подъемников и тарифов]

values

(1,201,'с 10 до 20 часов'),

(2,202,'с 10 до 17 часов'),

(3, 203, 'с 10 до 20 часов'),

(4, 204, 'с 10 до 20 часов'),

(5,205 , 'с 10 до 20 часов'),

(6, 206, 'с 10 до 20 часов'),

(7,207 , 'с 10 до 20 часов'),

(8,208 , 'с 9 до 19 часов'),

(9,209 , 'с 10 до 20 часов'),

(10, 201, 'с 10 до 20 часов'),

(6,203, 'с 10 до 20 часов'),

(7,206 , 'с 10 до 20 часов'),

(8, 209, 'с 10 до 20 часов'),

(2, 203, 'с 10 до 15 часов'),

(3,201 , 'с 10 до 18 часов'),

(4, 207, 'с 10 до 20 часов')

use Карасев\_945

go

insert into Карточки

values

(201,'2020-12-06','2 дня'),

(202,'2020-12-01','3 дня'),

(203,'2020-11-03','4 дня'),

(204,'2020-11-26','3 дня'),

(205,'2020-11-23','2 дня'),

(206,'2020-12-05','3 дня'),

(207,'2020-12-02','5 дней'),

(208,'2020-12-07','3 дня'),

(209,'2020-11-15','1 день'),

(210,'2020-12-10','4 дня'),

(203,'2020-10-15','4 дня'),

(201,'2020-01-10','2 дня'),

(204,'2020-04-26','3 дня'),

(205,'2020-10-23','2 дня'),

(209,'2020-09-05','1 день'),

(201,'2020-08-05','2 дня'),

(208,'2020-04-07','3 дня'),

(209,'2020-05-15','1 день'),

(202,'2020-11-10','3 дня'),

(205,'2020-01-09','2 дня')

# 3 Разработка основных операций реляционной алгебры

1. Вывести полную информацию о подъемниках, тип которых «Кресельный».

R = Подъемники WHERE [Тип подъемника] = ‘Кресельный’

1. Вывести Стоимость, период действия и количество подъемов тарифов, у которых есть льготные скидки.

R= (Тарифы WHERE [Льготная скидка] =’есть’) [Стоимость, Период действия, Количество подъемов]

1. Вывести список подъемников, у которых не известна начальная высота.

R= (Подъемники WHERE [Начальная высота] = NULL) [Название]

1. Вывести информацию о дате покупки и стоимость тарифа.  
   R1 = Тариф RENAME [id тарифа] AS Стоимость  
   R2 = (R1 JOIN Карточки) [Дата покупки, Стоимость]
2. Вывести стоимость тарифа, который никто из отдыхающих не купил.

R1 = Тарифы [id тарифа]

R2 = Карточка [id тарифа]

R3 = R1 EXCEPT R2  
R4 = (R3 JOIN Тарифы) [Стоимость]

1. Вывести информацию о подъемниках, у которых не указана высота начальная и не указана высота конечная.

R1=(Пользователь WHERE [Высота начальная]=NULL)  
R2=(Пользователь WHERE [Высота конечная]=NULL)

R3=R1 UNION R2

1. Вывести информацию о стоимости тарифа 1700 и 2000.

R1= (Тариф WHERE Стоимость = 1700)

R2= (Тариф WHERE Стоимость = 2000)

R3= R1 UNION R2

1. Вывести стоимость тарифа, у которого ID тарифа равно 202 и в то же время количество подъемов было равно 10.  
   R1= (Тариф WHERE [ID тарифа]=202) [Стоимость]  
   R2= (Тариф WHERE [Количество подъемов]=10) [Стоимость]

R3= R1 INTERSECT R2

1. Вывести информацию о количестве подъемов тарифа, у которых есть льготная скидка и вычесть из них информацию о подъемнике типа «Ленточные».

R1 = ([Соответствие подъемника тарифу] WHERE [Режим работы] = ‘с 10 до 20’) [id подъемника]  
R2 = (Подъемник WHERE [Тип подъемника] = ‘Ленточные) [id подъемника]  
R3 = R1 EXCEPT R2

1. Вывести дату покупки тарифов, у которых стоимость больше 2000.  
   R1 = (Тарифы WHERE Стоимость > 2000)[ID тарифа]  
   R2 = (R1 JOIN Карточки)[Дата покупки]
2. Получить информацию о подъемниках в следующем виде:

|  |  |  |  |
| --- | --- | --- | --- |
| Стоимость | Количество подъемов | Id тарифа | Дата подписки |

R = (Тарифы JOIN Карточки) [Стоимость, [Количество подъемов], [id тарифа], [Дата подписки]]

1. Вывести дату покупки отдыхающими тех тарифов, по которым имеется льготная скидка.  
   R1= (Тарифы WHERE [Льготная скидка] = ‘есть’)[ID тарифа]  
   R2 = (R1 JOIN Карточки) [Дата покупки]]
2. Вывести Id тарифов, у которых количество подъемов больше 8.  
   R=(Фильмы WHERE [Количество подъемов] >8) [id тарифов]
3. Вывести id тарифа и id подъемника, на которые отдыхающий приобрел карточку, где тип подъемника «Ленточный», а стоимость выше 1999.

R1= (Подъемники WHERE [Тип подъемника] = ‘Ленточный’) [ID подъемника]

R2 = (Тарифы WHERE [Стоимость] <1999) [ID тарифа]

R3 = R1 JOIN [Соответствие подъемника тарифу]

R4 = R2 JOIN [Соответствие подъемника тарифу]

R5 = R3 UNION R4

1. Вывести полную информацию о подъемниках, на которые отдыхающие купили карточки и тип подъемника которых «Гондольный».

R1= ([Соответствие подъемника тарифу])[ ID подъемника]

R2=(Подъемники WHERE [Тип подъемника]=’Гондольный’)[ID подъемника]

R3=R1 EXCEPT R2

R4 = (R3 JOIN Подъемники) [[ ID подъемника], Название,[Высота начальная], [Высота конечная], [Тип подъемника]

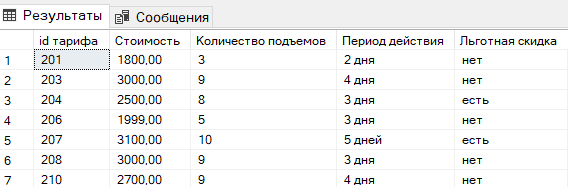
# 4 Разработка запросов к БД для реализации информационных потребностей пользователя

1. Вывести полную информацию обо всех тарифах, цена которых больше 1700.

SELECT \*

FROM Тарифы

WHERE Стоимость > 1700

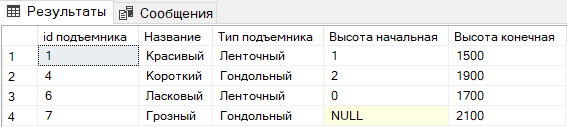
  
Рисунок 3 – Запрос с использованием операторов сравнения

1. Вывести информацию о подъемниках типа Гондольный и Ленточный.

SELECT \*

FROM Подъемники

WHERE [Тип подъемника] = 'Гондольный' or [Тип подъемника] ='Ленточный'

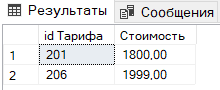
  
Рисунок 4 – Запрос с использованием OR

1. Вывести id тарифа, у стоимость которых больше 1700 и меньше 2500.

SELECT [id Тарифа],Стоимость

FROM Тариф

WHERE Стоимость <1700 and Стоимость <2500

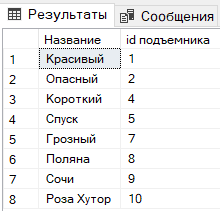
  
Рисунок 5 – Запрос с использованием AND

1. Вывести название подъемника, id которых не равны 6 и 3.

SELECT Название, [id подъемника]

FROM Подъемники

WHERE [id подъемника] not in (3,6)

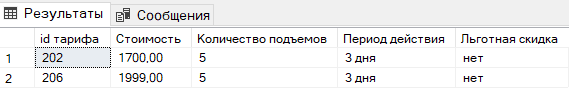
  
Рисунок 6 – Запрос с использованием NOT

1. Вывести информацию о тарифах, стоимость покупки которых равна 1700 и 1999, а период действия равен 3 дням.

SELECT \*

FROM Тарифы

WHERE (Стоимость=1700 or Стоимость=1999) and [Период действия]= '3 дня'

  
Рисунок 7 – Запрос на использование комбинации логических операторов

1. Вывести id подъемников с указанием их конечной высотой в километрах и метрах.

SELECT [id подъемника], [Высота конечная]/1000 as километры, [Высота конечная]%1000 as метры

FROM Подъемники

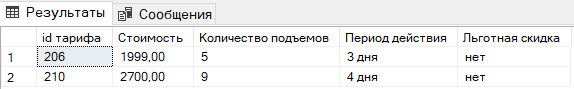
  
Рисунок 8 – Запрос на использование выражений над столбцами

1. Получить информацию о тарифах с id равным 210 и 206.

SELECT \*

FROM Тарифы

WHERE [id тарифа] in (206,210)

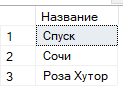
  
Рисунок 9 – Запрос с проверкой на принадлежность множеству

1. Вывести названия подъемников, начинающихся с букв от 'Р' по 'Т'.

SELECT Название

FROM Подъемники

WHERE Название BETWEEN 'Р' AND 'Т'

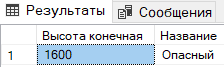
  
Рисунок 10 – Запрос с проверкой на принадлежность диапазону значений

1. Вывести название и конечную высоту тех подъемников, название которых начинается на букву ‘О’.

SELECT [Высота конечная], Название

FROM Подъемники

WHERE Название LIKE 'О%'

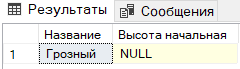
  
Рисунок 11 – Запрос с проверкой на соответствие шаблону

1. Вывести название подъемников, у которых не указана начальная высота.

SELECT Название, [Высота начальная]

FROM Подъемники

WHERE [Высота начальная] is null

  
Рисунок 12 – запрос с проверкой на неопределенное значение

1. Вывести название, начальную высоту и тип всех подъемников. Если начальная высота не указана, то в соответствующем столбце вывести строку: «1111».

SELECT Название, [Высота начальная], [Тип подъемника]

FROM Подъемники

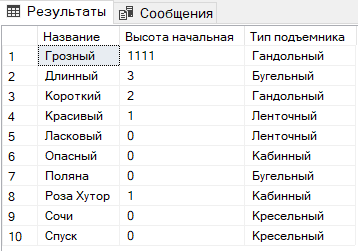
WHERE [Высота начальная] is not null

UNION

SELECT Название, '1111', [Тип подъемника]

FROM Подъемники

WHERE [Высота начальная] is null

  
Рисунок 13 – Запрос на объединение таблиц

1. Вывести коды тарифов, у которых количество подъемов больше 5, и меньше 10.

SELECT [id тарифа]

FROM Тарифы

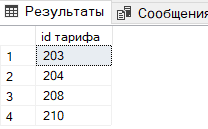
WHERE [Количество подъемов]>5

INTERSECT

SELECT [id тарифа]

FROM Тарифы

WHERE [Количество подъемов] <10

  
Рисунок 14 – Запрос на пересечение таблиц

1. Вывести названия тарифов, которые никто из отдыхающих не приобрел.

SELECT [id тарифа]

FROM Тарифы

EXCEPT

SELECT [id тарифа]

FROM [Соответствие подъемников и тарифов]



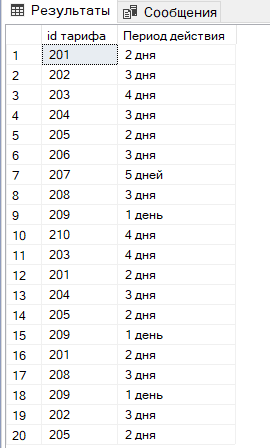
Рисунок 15 – Запрос на вычитание таблиц

1. Вывести id тарифов и период действия для карточек.

SELECT Тарифы.[id тарифа], Карточки.[Период действия]

FROM Тарифы, Карточки

where [Тарифы].[id тарифа]=Карточки.[id тарифа]

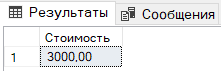
  
Рисунок 16 – Декартово произведение таблиц

1. Вывести стоимость тарифа, дата покупки на который была осуществлена 15 октября 2020 года.

SELECT DISTINCT Стоимость

FROM Тарифы INNER JOIN Карточки ON Тарифы.[id тарифа]=Карточки.[id тарифа]

WHERE [Дата покупки]='2020-10-15'

  
Рисунок 17 – Запрос с использованием соединения двух таблиц по равенству и условием отбора

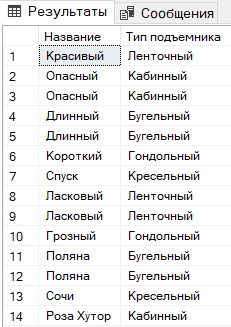
1. Вывести название и типы подъемников, исключив информацию о тарифе, где количество подъемов равно 10.

SELECT Название, [Тип подъемника]

FROM Подъемники INNER JOIN [Соответствие подъемников и тарифов] ON Подъемники.[id подъемника]=[Соответствие подъемников и тарифов].[id подъемника]

INNER JOIN Тарифы ON [Соответствие подъемников и тарифов].[id тарифа]=Тарифы.[id тарифа]

WHERE [Количество подъемов] <> 10

  
Рисунок 18 – Запрос с использованием соединения трех таблиц по равенству и условием отбора

1. Вывести название, дату покупки, типы подъемников карточки на которые приобрели 2020-04-07, 2020-11-10, 2020-12-01.

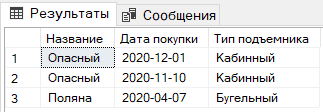
SELECT Название, [Дата покупки], [Тип подъемника]

FROM Тарифы INNER JOIN [Соответствие подъемников и тарифов] ON Тарифы.[id тарифа]=[Соответствие подъемников и тарифов].[id тарифа]

INNER JOIN Подъемники ON [Соответствие подъемников и тарифов].[id подъемника]=Подъемники.[id подъемника]

INNER JOIN Карточки ON Тарифы.[id тарифа]=Карточки.[id тарифа ]

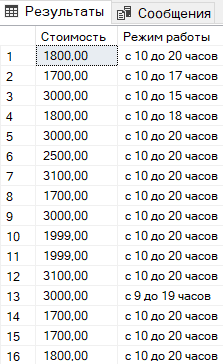
WHERE [Дата покупки] = '2020-04-07' OR [Дата покупки] = '2020-11-10' OR [Дата покупки] = '2020-12-01'

  
Рисунок 19 – Запрос с использованием соединения по четырем таблицам

1. Вывести информацию обо всех существующих тарифах, указав их стоимость и режим работы.

SELECT Стоимость, [Режим работы]

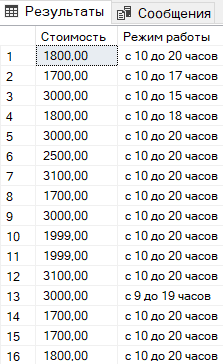
FROM [Соответствие подъемников и тарифов] left join Тарифы on [Соответствие подъемников и тарифов].[id тарифа]=Тарифы.[id тарифа]

  
Рисунок 20 – Запрос с использованием левого внешнего соединения

1. Вывести информацию обо всех существующих тарифах, указав их стоимость и режим работы.

SELECT Стоимость, [Режим работы]

FROM Тарифы right join [Соответствие подъемников и тарифов] on Тарифы.[id тарифа]=[Соответствие подъемников и тарифов].[id тарифа]

  
Рисунок 21 – Запрос с использованием правого внешнего соединения

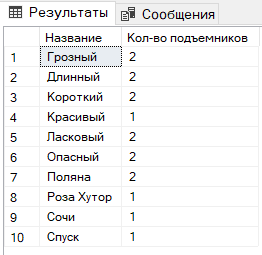
1. Вывести название подъемников и их количество.

SELECT Название, COUNT ([Соответствие подъемников и тарифов].[id подъемника]) as [Кол-во подъемников]

FROM Подъемники, [Соответствие подъемников и тарифов]

WHERE Подъемники.[id подъемника]=[Соответствие подъемников и тарифов].[id подъемника]

GROUP BY Подъемники.Название

  
Рисунок 22 – Запрос с использованием функции COUNT

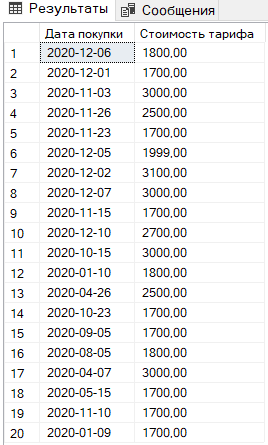
1. Найти сколько заплатит отдыхающий, если купит все тарифы.

SELECT [Дата покупки], SUM (Тарифы.Стоимость) as [Стоимость тарифа]

FROM Карточки, Тарифы

WHERE Карточки.[id тарифа]=Тарифы.[id тарифа]

GROUP BY Карточки.[Дата покупки]

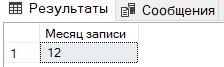
  
Рисунок 23 – Запрос с использованием функции SUM

1. Вывести месяц, когда приобрели тариф, стоимость которого равна 1999.

SELECT DISTINCT MONTH([Дата покупки]) AS [Месяц записи]

FROM Тарифы INNER JOIN Карточки ON Тарифы.[id тарифа]=Карточки.[id тарифа ]

WHERE Стоимость=1999

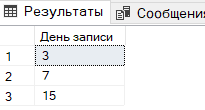
  
Рисунок 24 – Запрос с использованием временных функций

1. Вывести день, когда приобрели тариф, стоимость которого равна 3000.

SELECT DISTINCT DAY([Дата покупки]) AS [День записи]

FROM Тарифы INNER JOIN Карточки ON Тарифы.[id тарифа]=Карточки.[id тарифа ]

WHERE Стоимость=3000

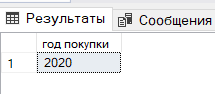
  
Рисунок 25 – Запрос с использованием временных функций

1. Вывести год, когда приобрели тариф, стоимость которого равна 1800.

SELECT DISTINCT YEAR([Дата покупки]) AS [год покупки]

FROM Тарифы INNER JOIN Карточки ON Тарифы.[id тарифа]=Карточки.[id тарифа ]

WHERE Стоимость=1800

  
Рисунок 26 – Запрос с использованием временных функций

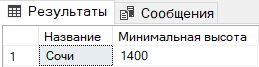
1. Вывести название и минимальную конечную высоту подъемников, названия которых начинаются с буквы О.

SELECT Название, MIN([Высота конечная]) AS[Минимальная высота]

FROM Подъемники

WHERE Название LIKE 'С%'

GROUP BY Название

  
Рисунок 27 – Запрос с использованием группировки по одному столбцу

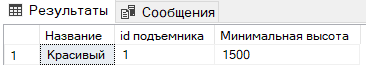
1. Вывести название и минимальную конечную высоту подъемников, названия которых начинаются с буквы К.

SELECT Название,[id подъемника], MIN([Высота конечная]) AS[Минимальная высота]

FROM Подъемники

WHERE Название LIKE 'К%'

GROUP BY Название, [id подъемника]

  
Рисунок 28 – Запрос на использование группировки по нескольким столбцам

1. Вывести количество покупок тех отдыхающих, которые покупали тариф не менее двух раз.

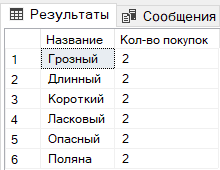
select Подъемники.Название, COUNT ([Соответствие подъемников и тарифов].[id подъемника]) As [Кол-во покупок]

from Подъемники, [Соответствие подъемников и тарифов]

where Подъемники.[id подъемника]= [Соответствие подъемников и тарифов].[id подъемника]

group by Подъемники.Название

having COUNT([Соответствие подъемников и тарифов].[id подъемника])>=2

  
Рисунок 29 – Запрос с использованием условия отбора групп HAVING

1. Вывести название подъемников в порядке возрастания в зависимости от количества приобретенных тарифов.

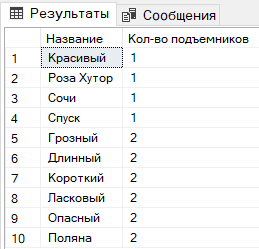
SELECT Подъемники.Название, COUNT ([Соответствие подъемников и тарифов].[id подъемника]) AS [Кол-во подъемников]

FROM Подъемники, [Соответствие подъемников и тарифов]

WHERE Подъемники.[id подъемника]=[Соответствие подъемников и тарифов].[id подъемника]

GROUP BY Подъемники.Название

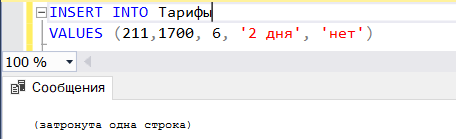
ORDER BY COUNT ([Соответствие подъемников и тарифов].[id подъемника]) ASC

  
Рисунок 30 – Запрос с использованием сортировки по столбцу

1. Добавить новую строку в таблицу «Тарифы».

INSERT INTO Тарифы

VALUES (211,1700, 6, '2 дня', 'нет')

  
Рисунок 31 – Запрос на добавление новых данных в таблицу

1. Создать таблицу «Подъемник «Длинный»», поместив в нее информацию о подъемнике «Длинный».

use Карасев\_945

CREATE TABLE [Подъемник "Длинный"]

([id подъемника] int,

Название nchar(20) NOT NULL check (Название like '[А-Я]%'),

[Тип подъемника] nchar(20) NOT NULL check ([Тип подъемника] in ('Кресельный', 'Гондольный', 'Бугельный', 'Ленточный', 'Кабинный')),

[Высота начальная] int check ([Высота начальная]>=0),

[Высота конечная] int check ([Высота конечная] >0))

go

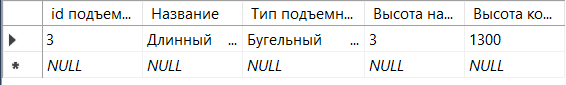
INSERT INTO [Подъемник "Длинный"]

SELECT \*

FROM Подъемники

WHERE Название='Длинный'

Go

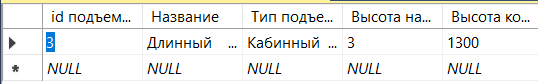
  
Рисунок 32 – Запрос на добавление новых данных по результатам выполненного запроса

1. Изменить тип подъемника «Бугельный» на «Кабинный» в таблице «Подъемник «Длинный»».

UPDATE [Подъемник "Длинный"]

set [Тип подъемника] ='Кабинный'

WHERE [Тип подъемника] ='Бугельный'

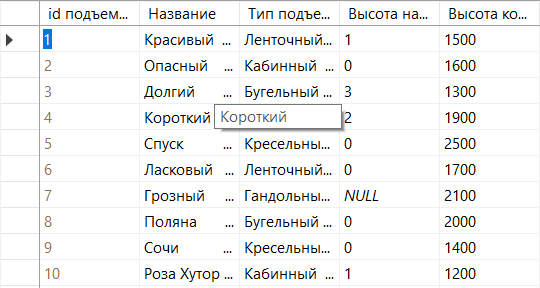
  
Рисунок 33 – Запрос на обновление существующих данных в таблице

1. Изменить название подъемника «Длинный» на «Долгий»

UPDATE Подъемники

set название='Долгий'

WHERE [id подъемника]=3

  
Рисунок 34 – Запрос на обновление существующих данных по результатам подзапроса во фразе WHERE

1. Удалить все записи из таблицы «Подъемник «Длинный»».

DELETE

FROM [Подъемник "Длинный"]

  
Рисунок 35 – Запрос на удаление существующих данных

1. Добавить скидку 20% на стоимость тарифа, который был приобретен 15.10.2020.

UPDATE Тарифы

SET Стоимость=Стоимость\*0.8

WHERE [id тарифа] in (SELECT [id тарифа]

FROM Карточки

WHERE [Дата покупки]='2020-10-15')

  
Рисунок 36 – Запрос на удаление существующих данных по результатам подзапроса во фразе WHERE

1. Удалить тарифы, которые были приобретены отдыхающими позже 05.12.2020.

DELETE [Тарифы]

FROM Карточки INNER JOIN Тарифы ON Карточки.[id тарифа ]=Тарифы.[ID Тарифа]

WHERE [Дата покупки]> '2020-12-05'

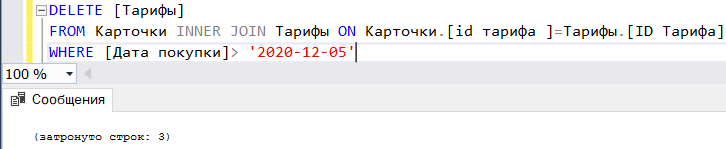


Рисунок 37 – запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц

1. Вывести дату покупки тарифа, стоимость которого равна 2500.

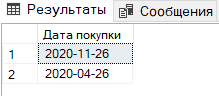
SELECT [Дата покупки]

FROM Карточки

WHERE [id тарифа ] = (SELECT [id тарифа ]

FROM Тарифы

WHERE Стоимость=2500)

  
Рисунок 38 – Запрос с использованием операций сравнения

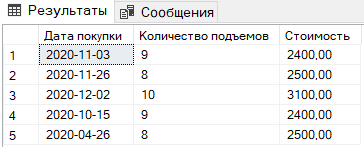
1. Вывести информацию о тарифах, а также о дате покупки, количество подъемов которых не ниже, чем среднее количество подъемов.

SELECT Карточки.[Дата покупки], Тарифы.[Количество подъемов], Тарифы.Стоимость

FROM Карточки INNER JOIN Тарифы ON Карточки.[id тарифа ]=Тарифы.[id тарифа]

WHERE [Количество подъемов] >=(SELECT AVG ([Количество подъемов])

FROM Тарифы)

  
Рисунок 39 – Запрос с использованием операций сравнения

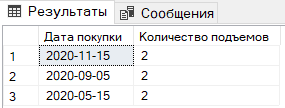
1. Вывести информацию о тарифах, а также о дате покупки, количество подъемов которых минимальный.

SELECT Карточки.[Дата покупки], Тарифы.[Количество подъемов]

FROM Карточки INNER JOIN Тарифы ON Карточки.[id тарифа ]=Тарифы.[id тарифа]

WHERE [Количество подъемов] =(SELECT min ([Количество подъемов])

FROM Тарифы)

  
Рисунок 40 – Запрос с использованием операций сравнения

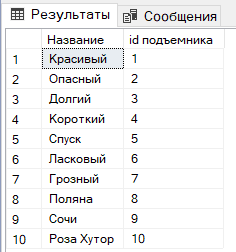
1. Вывести названия подъемников, на которые куплены тарифы.

SELECT Название, [id подъемника]

FROM Подъемники

WHERE [id подъемника] = any(SELECT [id подъемника]

FROM [Соответствие подъемников и тарифов])

  
Рисунок 41 – Запрос с использованием операции ANY

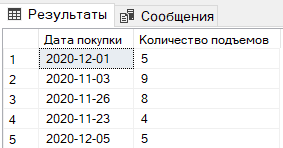
1. Вывести информацию о тарифах, количество подъемов которых не меньше, чем количество подъемов каждого из тарифов.

SELECT Карточки.[Дата покупки], Тарифы.[Количество подъемов]

FROM Карточки INNER JOIN Тарифы ON Карточки.[id тарифа ]=Тарифы.[id тарифа]

WHERE [Количество подъемов] >= ALL(SELECT min ([Количество подъемов])

FROM Тарифы)

  
Рисунок 42 – Запрос с использованием операции ALL

1. Выведем список тарифов, которые ни один покупатель не приобрел.

SELECT [id тарифа]

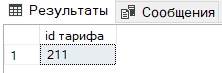
FROM Тарифы

WHERE [id тарифа] IN

(SELECT Тарифы.[id тарифа]

FROM Тарифы full JOIN Карточки ON Тарифы.[id тарифа] = Карточки.[id тарифа]

WHERE [Дата покупки] IS NULL)

  
Рисунок 43 – Запрос с использованием операции IN

1. Выведем информацию о тех тарифах, которые в настоящий момент хотя бы один отдыхающий приобрел.

SELECT \*

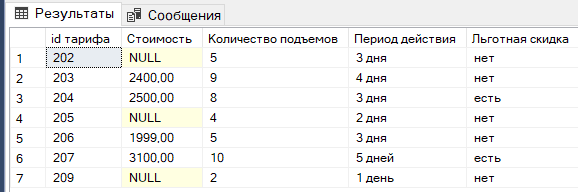
FROM Тарифы

WHERE EXISTS

(SELECT \*

FROM Карточки

WHERE Тарифы.[id тарифа] = Карточки.[id тарифа ])

  
Рисунок 44 – Запрос с использованием операции EXISTS

1. Вывести подъемники, на которых по тарифу можно подняться 10 раз.

SELECT Название

FROM Подъемники

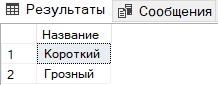
WHERE [id подъемника] IN (SELECT [id подъемника]

FROM [Соответствие подъемников и тарифов]

WHERE [id тарифа] IN (SELECT [id тарифа]

FROM Тарифы

WHERE [Количество подъемов] = 10))

  
Рисунок 45 – Запрос с использованием двух вложенных друг в друга подзапросов

1. Вывести подъемники, на которые карточки были куплены после 15.10.2020.

SELECT DISTINCT Название

FROM Подъемники

WHERE [id подъемника] IN(SELECT [id подъемника]

FROM [Соответствие подъемников и тарифов]

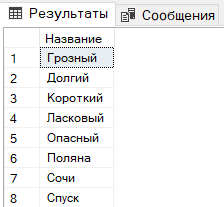
WHERE [id тарифа] IN(SELECT [id тарифа]

FROM Тарифы

WHERE [id тарифа] IN (SELECT [id тарифа]

FROM Карточки

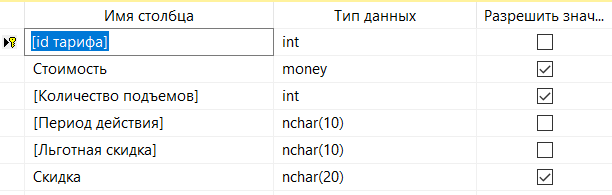
WHERE [Дата покупки] >'2020-10-15')))

  
Рисунок 46 – Запрос с использованием трех вложенных друг в друга подзапросов

1. Добавить к таблице «Тарифы» столбец «Скидка».

ALTER TABLE Тарифы

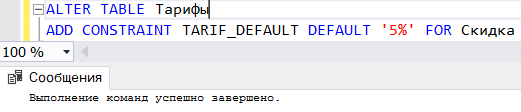
ADD Скидка nchar(20)

  
Рисунок 47 – Запрос на добавление нового столбца к таблице

1. Добавить ограничение по умолчанию в таблицу «Тарифы», где скидка тарифа будет равна 5%.

ALTER TABLE Тарифы

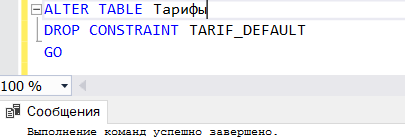
ADD CONSTRAINT TARIF\_DEFAULT DEFAULT '5%' FOR Скидка

  
Рисунок 48 – Запрос на добавление нового ограничения к таблице

1. Удалить ограничение, которое было установлено в прошлом запросе.

ALTER TABLE Тарифы

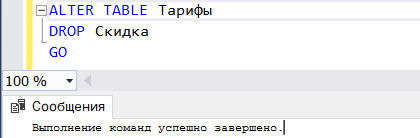
DROP CONSTRAINT TARIF\_DEFAULT  
GO

  
Рисунок 49 – Запрос на удаление нового ограничения к таблице

1. Удалить столбец «Скидка» из таблицы «Тарифы».

ALTER TABLE Тарифы

DROP Скидка  
GO

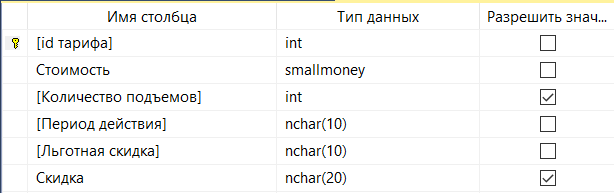
  
Рисунок 50 – Запрос на удаление нового столбца к таблице

1. Изменить тип данных у Стоимости в таблице «Тарифы» с money на smallmoney.

ALTER TABLE Тарифы

ALTER COLUMN Стоимость smallmoney

GO

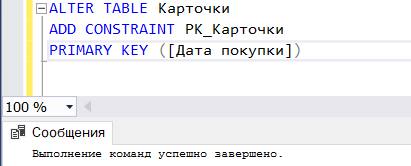
  
Рисунок 51 – Запрос на изменение типа данных в таблице

1. Добавить первичный ключ в таблицу «Карточки».

ALTER TABLE Карточки

ADD CONSTRAINT PK\_Карточки

PRIMARY KEY ([Дата покупки])

  
Рисунок 52 – Запрос на добавление нового первичного ключа к таблице с сохранением целостности таблицы

# Заключение

В результате данной курсовой работы была создана база данных «Горнолыжный курорт».

Также были разработаны основных операций реляционной алгебры и запросы для реализации информационных потребностей пользователя, а также для обеспечения целостности данных. В данной работе для выполнения поставленной задачи были использованы возможности СУБД Microsoft SQL Server 2018.

# Список литературы

1. Р. Виейра. Программирование баз данных Microsoft SQL Server 2005. Базовый курс — М.: «Диалектика», 2007. —832 c.
2. Клайн К. SQL. Справочник. 2-е издание / Пер. с англ. – М.: КУ-ДИЦ-ОБРАЗ, 2006 – 832 с.
3. Ицик Бен-Ган Microsoft SQL Server 2008 Основы T-SQL:Пер. с англ. – СПб:БВХ-Петербург, 2009. – 432с. :ил.
4. Глушаков С.В., Ломотько Д.В. Базы данных: Учебный курс / Харьков, Ростов-на-Дону, Киев,2002
5. Токмаков Г.П. Базы данных. Концепция баз данных, реляционная модель данных, языки SQL и XML. – УлГТУ, 2010. - 193 с.