Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Институт информатики и вычислительной техники

09.03.01 "Информатика и вычислительная техника"

профиль "Программное обеспечение средств вычислительной техники и автоматизированных систем"

Кафедра прикладной математики и кибернетики

**Расчетно-графическая работа**

**по курсу «Базы данных»**

**Вариант №19 «Аэропорт»**

Выполнил:

студент гр.ИП-213 Дмитриев Атон Александрович

ФИО студента

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Проверил:

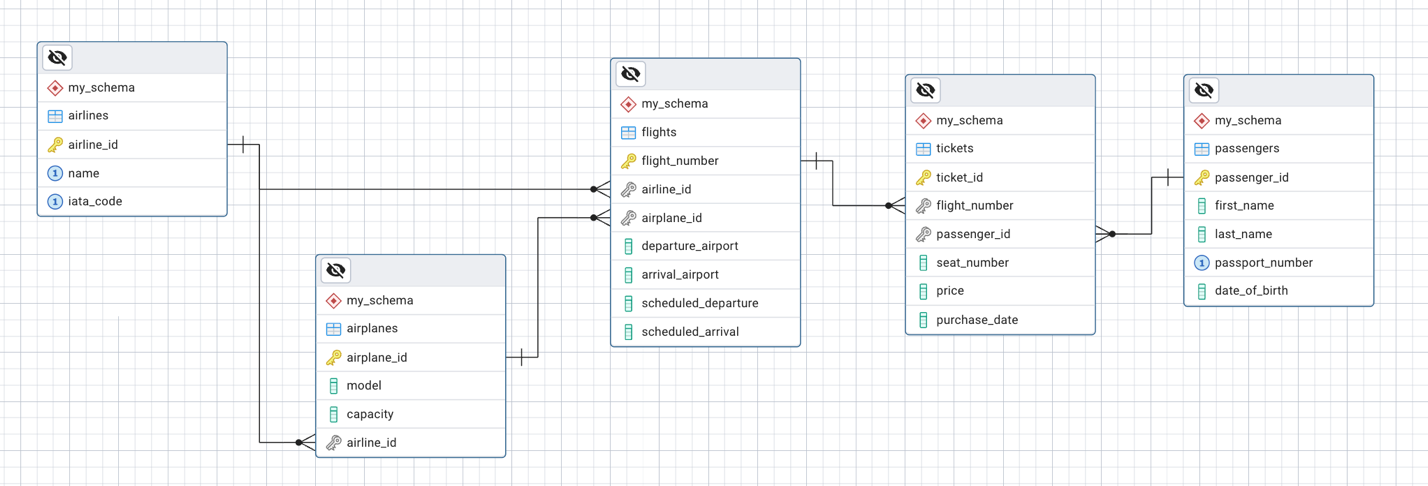
Преподаватель

ФИО преподавателя

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2025 г.

1. Описание базы данных

**

*Таблица 1. Сущности и связи*

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность 1 | Сущность 2 | Тип связи (1:1, 1:M, M:M) | Описание связи |
| Авиакомпании | Самолёты | 1:M | Одна авиакомпания может владеть несколькими самолетами |
| Авиакомпании | Рейсы | 1:M | Одна авиакомпания может выполнять несколько рейсов |
| Самолёты | Рейсы | 1:М | Один самолет может быть назначен на несколько рейсов (в разное время) |
| Билеты | Рейсы | M:1 | Несколько билетов может быть продано на один рейс |
| Пассажиры | Билеты | 1:М | Один пассажир может купить много билетов |

*Таблица 2. Атрибуты сущностей*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Сущность, название таблицы в БД | Название атрибута | Название атрибута в БД (имя поля) | Тип данных | Not Null | Primary key | Foreign key | Дополнительные ограничения целостности |
| Авиакомпании airlines | номер авиакомпании название авиакомпании код IATA авиакомпании | airline\_id name iata\_code | INT VARCHAR(100) CHAR(2) | 1 1 1 | P |  |  |
| Самолеты airplanes | номер самолета модель самолета вместимость самолета номер авиакомпании-владельца | airplane\_id model capacity airline\_id | INT VARCHAR(50) INT INT | 1 1 1 | P | F | > 0 |
| Рейсы flights | номер рейса номер авиакомпании номер самолета аэропорт вылета аэропорт прибытия время вылета по расписанию время прибытия по расписанию | flight\_number airline\_id airplane\_id departure\_airport arrival\_airport scheduled\_departure scheduled\_arrival | INT INT INT VARCHAR(3) VARCHAR(3) TIMESTAMP TIMESTAMP | 1  1 1 1 1 | P | F  F |  |
| Пассажиры passengers | номер пассажира имя пассажира фамилия пассажира номер паспорта дата рождения | passenger\_id first\_name last\_name passport\_number date\_of\_birth | INT VARCHAR(50) VARCHAR(50) VARCHAR(20) DATE | 1 1 1 1 1 | P |  | UNIQUE  Дата в прошлом |
| Билеты tickets | номер билета номер рейса номер пассажира номер места цена билета дата покупки | ticket\_id flight\_number passenger\_id seat\_number price purchase\_date | INT INT INT VARCHAR(5) DECIMAL(10,2) DATE | 1   1 1 1 | P | F  F | > 0  CURRENT\_DATE |

1.2 Данные БД (содержимое таблиц)

Таблица airlines

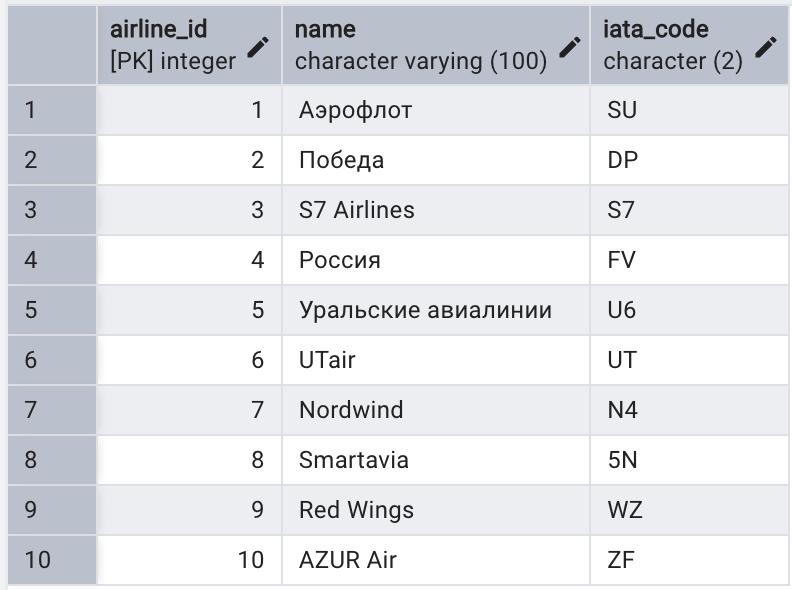


Таблица airplanes

Изображение выглядит как текст, снимок экрана, меню, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Таблица flightsИзображение выглядит как текст, снимок экрана, чек, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Таблица passengersИзображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Таблица ticketsИзображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

1. Запросы к БД

2.1. Найти пассажиров, у которых суммарная стоимость всех купленных билетов превышает 20 000 рублей.

SELECT

p.first\_name,

p.last\_name,

COUNT(t.ticket\_id) AS tickets\_count,

SUM(t.price) AS total\_spent

FROM passengers p

JOIN tickets t ON p.passenger\_id = t.passenger\_id

GROUP BY p.passenger\_id, p.first\_name, p.last\_name

HAVING SUM(t.price) > 20000

ORDER BY total\_spent DESC;

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

2.2. Вывести авиакомпании и количество различных моделей самолетов в их парке, у которых в парке более 2 моделей.

SELECT

a.name AS airline\_name,

COUNT(DISTINCT ap.model) AS different\_models\_count

FROM airlines a

JOIN airplanes ap ON a.airline\_id = ap.airline\_id

GROUP BY a.airline\_id, a.name

HAVING COUNT(DISTINCT ap.model) > 2

ORDER BY different\_models\_count DESC;

Изображение выглядит как текст, Шрифт, снимок экрана, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

2.3. Вывести полную информацию о всех билетах: ФИО пассажира, номер рейса, направление, время вылета, номер места и стоимость.

SELECT

p.first\_name,

p.last\_name,

f.flight\_number,

CONCAT(f.departure\_airport, ' → ', f.arrival\_airport) AS route,

TO\_CHAR(f.scheduled\_departure, 'DD.MM.YYYY HH24:MI') AS departure\_time,

t.seat\_number,

t.price

FROM tickets t

JOIN passengers p ON t.passenger\_id = p.passenger\_id

JOIN flights f ON t.flight\_number = f.flight\_number

JOIN airlines a ON f.airline\_id = a.airline\_id

ORDER BY f.scheduled\_departure, p.last\_name;

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

2.4. Найти все рейсы S7 Airlines с указанием модели самолета и его вместимости.

SELECT

f.flight\_number,

CONCAT(f.departure\_airport, ' → ', f.arrival\_airport) AS route,

TO\_CHAR(f.scheduled\_departure, 'DD.MM.YYYY HH24:MI') AS departure\_time,

ap.model AS airplane\_model,

ap.capacity

FROM flights f

JOIN airlines a ON f.airline\_id = a.airline\_id

JOIN airplanes ap ON f.airplane\_id = ap.airplane\_id

WHERE a.name = 'S7 Airlines'

ORDER BY f.scheduled\_departure;



2.5. Создать представление frequent\_flyers, которое показывает пассажиров, которые купили билеты на рейсы в Новосибирск (OVB).

CREATE VIEW frequent\_flyers AS

SELECT DISTINCT

p.passenger\_id,

p.first\_name,

p.last\_name,

p.passport\_number

FROM passengers p

WHERE p.passenger\_id IN (

SELECT t.passenger\_id

FROM tickets t

JOIN flights f ON t.flight\_number = f.flight\_number

WHERE f.arrival\_airport = 'OVB'

);

-- Посмотреть пассажиров, летающих в Новосибирск

SELECT \* FROM frequent\_flyers;



2.6. Создать представление flight\_load для анализа загрузки рейсов.

CREATE VIEW flight\_load AS

SELECT

f.flight\_number,

CONCAT(f.departure\_airport, ' → ', f.arrival\_airport) AS route,

TO\_CHAR(f.scheduled\_departure, 'DD.MM.YYYY HH24:MI') AS departure\_time,

a.name AS airline,

ap.model AS airplane\_model,

ap.capacity,

COUNT(t.ticket\_id) AS tickets\_sold,

ROUND((COUNT(t.ticket\_id) \* 100.0 / ap.capacity), 1) AS load\_percentage

FROM flights f

JOIN airlines a ON f.airline\_id = a.airline\_id

JOIN airplanes ap ON f.airplane\_id = ap.airplane\_id

LEFT JOIN tickets t ON f.flight\_number = t.flight\_number

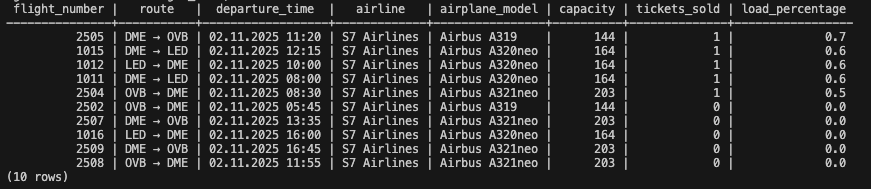
GROUP BY f.flight\_number, f.departure\_airport, f.arrival\_airport,

f.scheduled\_departure, a.name, ap.model, ap.capacity

ORDER BY load\_percentage DESC;

-- Посмотреть загрузку рейсов

SELECT \* FROM flight\_load;



1. Листинги SQL-скриптов

-------------- 1 База данных --------------

CREATE TABLE airlines (

airline\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL UNIQUE,

iata\_code CHAR(2) NOT NULL UNIQUE

);

CREATE TABLE airplanes (

airplane\_id INT PRIMARY KEY,

model VARCHAR(50) NOT NULL,

capacity INT NOT NULL,

airline\_id INT,

FOREIGN KEY (airline\_id) REFERENCES airlines(airline\_id),

CONSTRAINT chk\_capacity CHECK (capacity > 0)

);

CREATE TABLE flights (

flight\_number INT PRIMARY KEY,

airline\_id INT,

airplane\_id INT,

departure\_airport VARCHAR(3) NOT NULL,

arrival\_airport VARCHAR(3) NOT NULL,

scheduled\_departure TIMESTAMP NOT NULL,

scheduled\_arrival TIMESTAMP NOT NULL,

FOREIGN KEY (airline\_id) REFERENCES airlines(airline\_id),

FOREIGN KEY (airplane\_id) REFERENCES airplanes(airplane\_id)

);

CREATE TABLE passengers (

passenger\_id INT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

passport\_number VARCHAR(20) NOT NULL UNIQUE,

date\_of\_birth DATE NOT NULL,

CONSTRAINT chk\_dob\_past CHECK (date\_of\_birth < CURRENT\_DATE)

);

CREATE TABLE tickets (

ticket\_id INT PRIMARY KEY,

flight\_number INT,

passenger\_id INT,

seat\_number VARCHAR(5),

price DECIMAL(10, 2) NOT NULL,

purchase\_date DATE NOT NULL DEFAULT CURRENT\_DATE,

FOREIGN KEY (flight\_number) REFERENCES flights(flight\_number),

FOREIGN KEY (passenger\_id) REFERENCES passengers(passenger\_id),

CONSTRAINT chk\_price CHECK (price > 0)

);

-------------- 2 Заполнение таблиц --------------

INSERT INTO airlines (airline\_id, name, iata\_code) VALUES

(1, 'Аэрофлот', 'SU'),

(2, 'Победа', 'DP'),

(3, 'S7 Airlines', 'S7'),

(4, 'Россия', 'FV'),

(5, 'Уральские авиалинии', 'U6'),

(6, 'UTair', 'UT'),

(7, 'Nordwind', 'N4'),

(8, 'Smartavia', '5N'),

(9, 'Red Wings', 'WZ'),

(10, 'AZUR Air', 'ZF');

INSERT INTO airplanes (airplane\_id, model, capacity, airline\_id) VALUES

-- Аэрофлот

(1, 'Boeing 777-300ER', 375, 1),

(2, 'Airbus А321', 230, 1),

(3, 'Airbus A321NEO', 240, 1),

(4, 'Boeing 737-800', 158, 1),

-- Победа

(5, 'Boeing 737-800', 189, 2),

-- S7 Airlines

(6, 'Airbus A319', 144, 3),

(7, 'Airbus A320neo', 164, 3),

(8, 'Airbus A321neo', 203, 3),

(9, 'Airbus A321nx', 211, 3),

(10, 'Boeing 737-800', 168, 3),

(11, 'Embraer 170', 78, 3),

-- Россия

(12, 'Airbus A319-100', 138, 4),

-- Уральские авиалинии

(13, 'Airbus А319-100', 140, 5),

-- UTAir

(14, 'Airbus A319-100', 138, 6),

-- Nordwind

(15, 'Boeing 737-800', 189, 7),

-- Smartavia

(16, 'Airbus A320neo', 164, 8),

-- Red Wings

(17, 'Airbus A321neo', 203, 9),

-- AZUR Air

(18, 'Airbus A321nx', 211, 10);

INSERT INTO flights (flight\_number, airline\_id, airplane\_id, departure\_airport, arrival\_airport, scheduled\_departure, scheduled\_arrival) VALUES

-- Новосибирск (OVB) → Москва (DME)

(2502, 3, 6, 'OVB', 'DME', '2025-11-02 05:45:00', '2025-11-02 06:20:00'),

(2504, 3, 8, 'OVB', 'DME', '2025-11-02 08:30:00', '2025-11-02 09:05:00'),

(2508, 3, 8, 'OVB', 'DME', '2025-11-02 11:55:00', '2025-11-02 12:35:00'),

-- Москва (DME) → Новосибирск (OVB)

(2505, 3, 6, 'DME', 'OVB', '2025-11-02 11:20:00', '2025-11-02 19:30:00'),

(2507, 3, 8, 'DME', 'OVB', '2025-11-02 13:35:00', '2025-11-02 21:45:00'),

(2509, 3, 8, 'DME', 'OVB', '2025-11-02 16:45:00', '2025-11-03 00:55:00'),

-- Москва (DME) → Санкт-Петербург (LED)

(1011, 3, 7, 'DME', 'LED', '2025-11-02 08:00:00', '2025-11-02 09:30:00'),

(1015, 3, 7, 'DME', 'LED', '2025-11-02 12:15:00', '2025-11-02 13:45:00'),

-- Санкт-Петербург (LED) → Москва (DME)

(1012, 3, 7, 'LED', 'DME', '2025-11-02 10:00:00', '2025-11-02 11:30:00'),

(1016, 3, 7, 'LED', 'DME', '2025-11-02 16:00:00', '2025-11-02 17:30:00');

INSERT INTO passengers (passenger\_id, first\_name, last\_name, passport\_number, date\_of\_birth) VALUES

(1, 'Иван', 'Петров', '4512 123456', '1985-03-15'),

(2, 'Мария', 'Сидорова', '5214 654321', '1992-07-22'),

(3, 'Алексей', 'Козлов', '4011 789012', '1978-11-03'),

(4, 'Елена', 'Васильева', '4809 345678', '1989-05-18'),

(5, 'Дмитрий', 'Николаев', '4615 901234', '1995-12-30');

INSERT INTO tickets (ticket\_id, flight\_number, passenger\_id, seat\_number, price) VALUES

-- Мария Сидорова (перелет туда-обратно Москва-СПб)

(1, 1011, 2, '15C', 8900.00), -- S7 1011 DME → LED

(2, 1012, 2, '15D', 8700.00), -- S7 1012 LED → DME

-- Алексей Козлов (рейс в Новосибирск)

(3, 2505, 3, '20F', 21500.00), -- S7 2505 DME → OVB

-- Елена Васильева (несколько рейсов)

(4, 2504, 4, '05A', 19200.00), -- S7 2504 OVB → DME

(5, 1015, 4, '10B', 9500.00); -- S7 1015 DME → LED