



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ОТЧЕТ

по лабораторной работе № 1

Название: Введение в SQL

Дисциплина: Базы данных

Студент

ИУ6-33Б
(Группа)

19.11.2023

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

М.А. Скворцова

(И.О. Фамилия)

Москва, 2023

Цель

Первое практическое задание заключается в знакомстве со средой pgAdmin и написании SQL-запросов с использованием оператора SELECT.

Ход работы

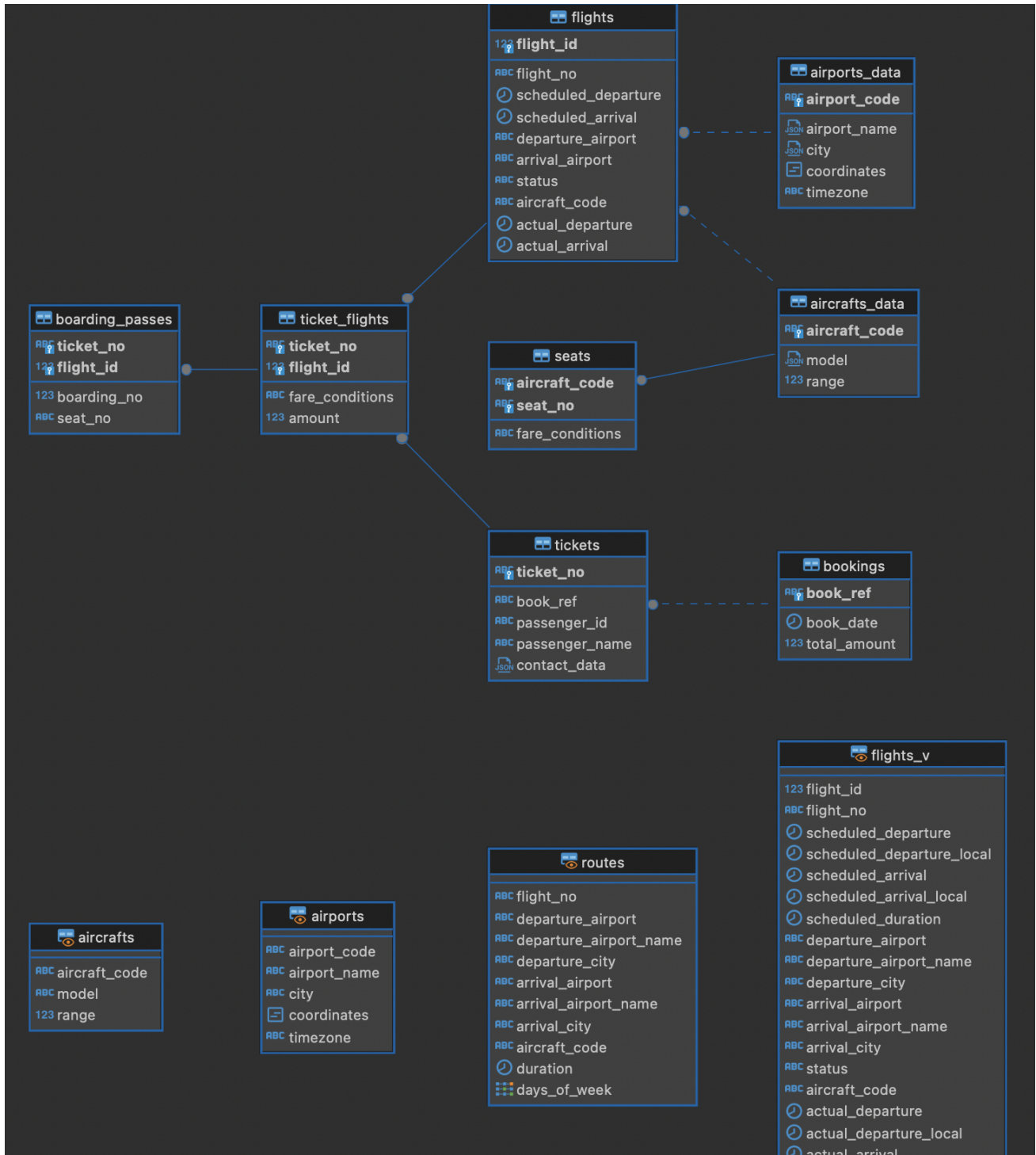


Рисунок 1. Схема импортируемой базы данных

— Вывести все самолеты, дальность полета которых > 5000км
 select * from aircrafts_data ad where range > 5000;

aircrafts_data 1 ×

select * from aircrafts_data ad where range > 5000 | Enter a SQL expression to filter results (use Ctrl+Space)

	aircraft_code	model	range
1	773	{"en": "Boeing 777-300", "ru": "Боинг 777-300"}	11,100
2	763	{"en": "Boeing 767-300", "ru": "Боинг 767-300"}	7,900
3	320	{"en": "Airbus A320-200", "ru": "Аэробус A320-200"}	5,700
4	321	{"en": "Airbus A321-200", "ru": "Аэробус A321-200"}	5,600
5	319	{"en": "Airbus A319-100", "ru": "Аэробус A319-100"}	6,700

Рисунок 2. Запрос 1

— Билеты с стоимостью с учетом скидки
 SELECT amount, ROUND ((amount / 100), 2) * 30 as sale, amount - ROUND ((amount / 100), 2) * 30 as price
 FROM ticket_flights
 LIMIT 100;

ticket_flights 1 ×

SELECT amount, ROUND ((amount / 100), 2) * 30 as | Enter a SQL expression to filter results (use Ctrl+Space)

	amount	sale	price
1	42,100	12,630	29,470
2	42,100	12,630	29,470
3	42,100	12,630	29,470
4	42,100	12,630	29,470
5	42,100	12,630	29,470
6	23,900	7,170	16,730
7	23,900	7,170	16,730
8	23,900	7,170	16,730
9	23,900	7,170	16,730
10	23,900	7,170	16,730
11	23,900	7,170	16,730
12	23,900	7,170	16,730
13	23,900	7,170	16,730
14	23,900	7,170	16,730
15	23,900	7,170	16,730
16	23,900	7,170	16,730
17	23,900	7,170	16,730
18	23,900	7,170	16,730
19	14,000	4,200	9,800
20	14,000	4,200	9,800
21	14,000	4,200	9,800
22	14,000	4,200	9,800
23	14,000	4,200	9,800
24	14,000	4,200	9,800
25	14,000	4,200	9,800
26	14,000	4,200	9,800
27	14,000	4,200	9,800
28	14,000	4,200	9,800
29	15,400	4,620	10,780
30	14,000	4,200	9,800
31	14,000	4,200	9,800
32	14,000	4,200	9,800
33	14,000	4,200	9,800
34	14,000	4,200	9,800
35	14,000	4,200	9,800

Рисунок 3. Запрос 2

Аэропорты с их количеством в каждой временной зоне отсортированные

```
select *, count( timezone ) over ( partition by timezone order by airport_code )
from airports_data
order by count desc;
```

airports_data 1 ×

select *, count(timezone) over (partition by timezone) Enter a SQL expression to filter results (use Ctrl+Space)

	airport_code	airport_name	city	coordinates	timezone	count
1	VOZ	{ "en": "Voronezh International Airport", "ru": "Воронежский международный аэропорт"	{ "en": "Voronezh", "ru": "Воронеж" }	(39.22959899902344,51.81420135498047)	Europe/Moscow	44
2	VKT	{ "en": "Vorkuta Airport", "ru": "Воркута" }	{ "en": "Vorkuta", "ru": "Воркута" }	(63.993099212646484,67.4886016845703)	Europe/Moscow	43
3	VKO	{ "en": "Vnukovo International Airport", "ru": "Внуков" }	{ "en": "Moscow", "ru": "Москва" }	(37.2615013123,55.5914993286)	Europe/Moscow	42
4	UUA	{ "en": "Bugulma Airport", "ru": "Бугульма" }	{ "en": "Bugulma", "ru": "Бугульма" }	(52.801700592041016,54.6399993896484)	Europe/Moscow	41
5	USK	{ "en": "Usinsk Airport", "ru": "Усинск" }	{ "en": "Usinsk", "ru": "Усинск" }	(57.3671989440918,66.00469970703125)	Europe/Moscow	40
6	URS	{ "en": "Kursk East Airport", "ru": "Курск-Восток" }	{ "en": "Kursk", "ru": "Курск" }	(36.29560089111328,51.7505989074707)	Europe/Moscow	39
7	UCT	{ "en": "Ukhta Airport", "ru": "Ухта" }	{ "en": "Ukhta", "ru": "Ухта" }	(53.8046989440918,63.566898345947266)	Europe/Moscow	38
8	TBW	{ "en": "Donskoye Airport", "ru": "Донское" }	{ "en": "Tambow", "ru": "Тамбов" }	(41.482799530029,52.806098937988)	Europe/Moscow	37
9	SVO	{ "en": "Sheremetyevo International Airport", "ru": "Шереметьево" }	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Europe/Moscow	36
10	STW	{ "en": "Stavropol Shpakovskoye Airport", "ru": "Ставрополь" }	{ "en": "Stavropol", "ru": "Ставрополь" }	(42.11280059814453,45.10919952392578)	Europe/Moscow	35
11	SKX	{ "en": "Saransk Airport", "ru": "Саранск" }	{ "en": "Saransk", "ru": "Саранск" }	(45.212257385253906,54.12512969970703)	Europe/Moscow	34
12	SCW	{ "en": "Syktyvkar Airport", "ru": "Сыктывкар" }	{ "en": "Syktyvkar", "ru": "Сыктывкар" }	(50.84510040283203,61.64699935913086)	Europe/Moscow	33
13	ROV	{ "en": "Rostov-on-Don Airport", "ru": "Ростов" }	{ "en": "Rostov", "ru": "Ростов-на-Дону" }	(39.8180999756,47.2582015991)	Europe/Moscow	32
14	PKV	{ "en": "Pskov Airport", "ru": "Псков" }	{ "en": "Pskov", "ru": "Псков" }	(28.395599365234375,57.78390121459961)	Europe/Moscow	31
15	PEZ	{ "en": "Penza Airport", "ru": "Пенза" }	{ "en": "Penza", "ru": "Пенза" }	(45.02109909057617,53.110599517822266)	Europe/Moscow	30
16	PES	{ "en": "Petrozavodsk Airport", "ru": "Петрозаводск" }	{ "en": "Petrozavodsk", "ru": "Петрозаводск" }	(34.154701232910156,61.88520050048828)	Europe/Moscow	29
17	OGZ	{ "en": "Beslan Airport", "ru": "Беслан" }	{ "en": "Beslan", "ru": "Владикавказ" }	(44.6066017151,43.2051010132)	Europe/Moscow	28
18	NNM	{ "en": "Naryan Mar Airport", "ru": "Нарьян-Мар" }	{ "en": "Naryan-Mar", "ru": "Нарьян-Мар" }	(53.12189865112305,67.63999938964844)	Europe/Moscow	27
19	NBC	{ "en": "Begishevo Airport", "ru": "Бегеишево" }	{ "en": "Nizhnekamsk", "ru": "Нижнекамск" }	(52.092498779296875,55.566470108032226)	Europe/Moscow	26
20	NAL	{ "en": "Nalchik Airport", "ru": "Нальчик" }	{ "en": "Nalchik", "ru": "Нальчик" }	(43.636600494384766,43.51290130615234)	Europe/Moscow	25
21	MRV	{ "en": "Mineralnyye Vody Airport", "ru": "Минеральные воды" }	{ "en": "Mineralnyye Vody", "ru": "Минеральные воды" }	(43.08190155029297,44.225101470947266)	Europe/Moscow	24
22	MMK	{ "en": "Murmansk Airport", "ru": "Мурманск" }	{ "en": "Murmansk", "ru": "Мурманск" }	(32.75080108642578,68.78170013427734)	Europe/Moscow	23
23	MCX	{ "en": "Uytash Airport", "ru": "Уйташ" }	{ "en": "Makhachkala", "ru": "Махачкала" }	(47.65230178833008,42.81679916381836)	Europe/Moscow	22
24	URJ	{ "en": "Uray Airport", "ru": "Петрозаводск" }	{ "en": "Ural", "ru": "Урай" }	(64.82669830322266,60.10329818725586)	Asia/Yekaterinburg	22
25	LPK	{ "en": "Lipetsk Airport", "ru": "Липецк" }	{ "en": "Lipetsk", "ru": "Липецк" }	(39.53779983520508,52.70280075073242)	Europe/Moscow	21
26	UFA	{ "en": "Ufa International Airport", "ru": "Уфа" }	{ "en": "Ufa", "ru": "Уфа" }	(55.874401092529,54.557498931885)	Asia/Yekaterinburg	21
27	TJM	{ "en": "Roshchino International Airport", "ru": "Тюмень" }	{ "en": "Tyumen", "ru": "Тюмень" }	(65.3243026733,57.18960189819996)	Asia/Yekaterinburg	20
28	LED	{ "en": "Pulkovo Airport", "ru": "Пулково" }	{ "en": "St. Petersburg", "ru": "Санкт-Петербург" }	(30.262500762939453,59.8003005981445)	Europe/Moscow	20
29	SVX	{ "en": "Koltsovo Airport", "ru": "Кольцово" }	{ "en": "Yekaterinburg", "ru": "Екатеринбург" }	(60.802700042725,56.743099212646)	Asia/Yekaterinburg	19
30	KZN	{ "en": "Kazan International Airport", "ru": "Казань" }	{ "en": "Kazan", "ru": "Казань" }	(49.278701782227,55.606201171875)	Europe/Moscow	19
31	SLY	{ "en": "Salekhard Airport", "ru": "Салехард" }	{ "en": "Salekhard", "ru": "Салехард" }	(66.61100006103516,66.5907974243164)	Asia/Yekaterinburg	18
32	KVX	{ "en": "Pobedilovo Airport", "ru": "Победилово" }	{ "en": "Kirov", "ru": "Киров" }	(49.348300933838,58.503299713135)	Europe/Moscow	18
33	SGC	{ "en": "Surgut Airport", "ru": "Сургут" }	{ "en": "Surgut", "ru": "Сургут" }	(73.40180206298828,61.34370040893555)	Asia/Yekaterinburg	17
34	KRR	{ "en": "Krasnodar Pashkovsky International Airport", "ru": "Краснодар" }	{ "en": "Krasnodar", "ru": "Краснодар" }	(39.170501708984,45.034698486328)	Europe/Moscow	17
35	REN	{ "en": "Orenburg Central Airport", "ru": "Оренбург" }	{ "en": "Orenburg", "ru": "Оренбург" }	(55.45669937133789,51.79579925371094)	Asia/Yekaterinburg	16
36	KLF	{ "en": "Grabtsevo Airport", "ru": "Калуга" }	{ "en": "Kaluga", "ru": "Калуга" }	(36.366687012,54.5499992371)	Europe/Moscow	16
37	PEE	{ "en": "Bolshoye Savino Airport", "ru": "Пермь" }	{ "en": "Perm", "ru": "Пермь" }	(56.021198272705,57.914501190186)	Asia/Yekaterinburg	15
38	JOK	{ "en": "Yoshkar-Ola Airport", "ru": "Йошкар-Ола" }	{ "en": "Yoshkar-Ola", "ru": "Йошкар-Ола" }	(47.904701232910156,56.700599670410156)	Europe/Moscow	15
39	IWA	{ "en": "Ivanovo South Airport", "ru": "Иваново" }	{ "en": "Ivanovo", "ru": "Иваново" }	(40.940799713134766,56.93939971923828)	Europe/Moscow	14
40	OVS	{ "en": "Sovetskiy Airport", "ru": "Советский" }	{ "en": "Sovetskiy", "ru": "Советский" }	(63.60191345214844,61.32662200927734)	Asia/Yekaterinburg	14
41	OSW	{ "en": "Orsk Airport", "ru": "Орск" }	{ "en": "Orsk", "ru": "Орск" }	(58.59560012817383,51.0724983215332)	Asia/Yekaterinburg	13
42	IAR	{ "en": "Tunoshna Airport", "ru": "Туношна" }	{ "en": "Yaroslavl", "ru": "Ярославль" }	(40.15739822387695,57.56069946289062)	Europe/Moscow	13

Рисунок 4. Запрос 3

Все самолеты с зарезервированной дальностью полета

```
with aircraft as (
select aircraft_code, model, range, (range / 1000) * 1000 * 0.10 as reserve
from aircrafts_data
)
select *, range-reserve as real_range
from aircraft;
```

aircrafts_data 1 ×

with aircraft as (select aircraft_code, model, range) Enter a SQL expression to filter results (use Ctrl+Space)

	aircraft_code	model	range	reserve	real_range
1	773	{ "en": "Boeing 777-300", "ru": "Боинг 777-300" }	11,100	1,100	10,000
2	763	{ "en": "Boeing 767-300", "ru": "Боинг 767-300" }	7,900	700	7,200
3	SU9	{ "en": "Sukhoi Superjet-100", "ru": "Сухой Суперджет-100" }	3,000	300	2,700
4	320	{ "en": "Airbus A320-200", "ru": "Аэробус A320-200" }	5,700	500	5,200
5	321	{ "en": "Airbus A321-200", "ru": "Аэробус A321-200" }	5,600	500	5,100
6	319	{ "en": "Airbus A319-100", "ru": "Аэробус A319-100" }	6,700	600	6,100
7	733	{ "en": "Boeing 737-300", "ru": "Боинг 737-300" }	4,200	400	3,800
8	CN1	{ "en": "Cessna 208 Caravan", "ru": "Сессна 208 Караван" }	1,200	100	1,100
9	CR2	{ "en": "Bombardier CRJ-200", "ru": "Бомбардье CRJ-200" }	2,700	200	2,500

Рисунок 5. Запрос 4

Вывод

Проведена настройка сервера базы данных посредством запуска docker-контейнера с сервером базы данных, импорт данных, установлен клиент DBeaver и настроено подключение до СУБД. Написаны и выполнены SQL-запросы, демонстрирующие различные возможности языка SQL.

Вопросы

2. Операция JOIN используется для объединения данных из нескольких таблиц в базе данных. Различные разновидности JOIN включают:

- INNER JOIN: Возвращает только те строки, которые имеют соответствия в обеих таблицах.
- LEFT JOIN (или LEFT OUTER JOIN): Возвращает все строки из левой таблицы и соответствующие строки из правой таблицы. Если соответствия нет, возвращается NULL.
- RIGHT JOIN (или RIGHT OUTER JOIN): Аналогично LEFT JOIN, но возвращает все строки из правой таблицы и соответствующие из левой.
- FULL JOIN (или FULL OUTER JOIN): Возвращает все строки из обеих таблиц, включая соответствия и NULL, если нет соответствия.

3. Агрегатные функции позволяют выполнять вычисления над данными в столбцах, например, COUNT, SUM, AVG, MAX, и MIN. GROUP BY используется

для группировки строк по значениям одного или нескольких столбцов. HAVING используется для фильтрации групп после применения агрегатных функций.

4. Для выбора только уникальных значений поля можно использовать оператор DISTINCT в запросе. Например, "SELECT DISTINCT column_name FROM table_name;" вернет только уникальные значения из столбца "column_name".

5. Для сортировки по возрастанию используется оператор "ORDER BY column_name ASC", а для сортировки по убыванию - "ORDER BY column_name DESC".

6. Агрегатные функции игнорируют NULL значения, за исключением COUNT(*), который учитывает все строки, включая NULL.
7. UNION объединяет результаты двух запросов и удаляет дубликаты, в то время как UNION ALL также объединяет результаты, но сохраняет все строки, включая дубликаты.
8. COUNT(*) подсчитывает все строки, включая NULL значения, в то время как COUNT(field) подсчитывает только непустые (не NULL) значения в столбце "field".
9. Для подсчета количества уникальных значений поля можно использовать COUNT(DISTINCT field).
10. Для проверки на неопределенное значение используется условие IS NULL. Например, "WHERE column_name IS NULL" выберет строки, в которых столбец "column_name" содержит NULL.
11. Предикат LIKE используется для выполнения поиска по шаблону в текстовых данных. Например, "WHERE column_name LIKE 'abc%'" выберет строки, в которых "column_name" начинается с "abc".
12. Чтобы выбрать только определенное количество строк, можно использовать операторы LIMIT (в большинстве СУБД, такие как MySQL и PostgreSQL) или FETCH FIRST (в Oracle). Например, "SELECT * FROM table_name LIMIT 10" вернет только первые 10 строк.
13. Чтобы исправить неверно работающий запрос, необходимо рассмотреть конкретную проблему и внести соответствующие изменения в запрос, например, изменить условия WHERE, исправить имена таблиц или столбцов и т. д.
14. Для упрощения запросов, обычно, стоит проверить, можно ли убрать лишние условия, оптимизировать использование индексов и объединить несколько запросов в один, если это возможно.
15. Для округления результата до 3 знаков после точки, вы можете использовать функцию ROUND(column_name, 3).

16. Для округления вещественного числа до целого, используйте функцию `CAST(column_name AS INT)` (или другой тип данных для целых чисел в вашей СУБД).

17. Чтобы переписать запрос без использования функции `MAX (MIN)`, вы можете использовать подзапросы или `JOIN` с другой таблицей, чтобы выбрать соответствующие значения.

18. Для изменения формата вывода данных, вы можете использовать функции преобразования типов данных, такие как `DATE_FORMAT()` для дат и времени в MySQL.

19. Для написания или модификации запроса по заданию, необходимо знать конкретные требования и структуру данных.