**Запросы**

Создание запросов к базам данных с помощью языка SQL является необходимым знанием для аналитика данных. В рамках данного занятия будут изучены следующие разделы:

1. Структура запроса SELECT
2. Использование оператора LIKE
3. Использование регулярных выражений с оператором LIKE
4. Предикаты в запросах
5. Вычисляемые столбцы
6. Упорядочивание строк
7. Отбор уникальных значений столбца с использованием ключевого слова DISTINCT
8. Отбор определенного количества значений и пропуски строк
9. Условные выражения

**Структура запроса SELECT**

Для выборки информации из таблиц базы данных служит команда SELECT. Ее синтаксис, упрощенный до предела, таков:

**SELECT имя-атрибута, имя-атрибута, ...**

**FROM имя-таблицы;**

Начнем с различных условий отбора строк в предложении WHERE. Эти условия могут конструироваться с использованием следующих операторов сравнения: =, < >, >, > =, < =. В предыдущих разделах мы уже использовали ряд таких операторов, поэтому сейчас рассмотрим некоторые другие способы осуществления отбора строк.

**Задание**

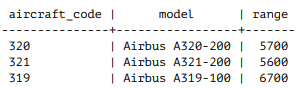
Создайте различные запросы к таблице товаров с использованием оператора SELECT и WHERE

**Использование оператора LIKE**

Для начала поставим перед собой такую задачу: выбрать все самолеты компании Airbus. В этом нам поможет оператор поиска шаблонов LIKE:

**SELECT \* FROM aircrafts WHERE model LIKE 'Airbus%';**

Обратите внимание на символ «%», имеющий специальное значение. Он соответствует любой последовательности символов, т. е. вместо него могут быть подставлены любые символы в любом количестве, а может и не быть подставлено ни одного символа. В результате будут выбраны строки, в которых значения атрибута model начинаются с символов «Airbus»:



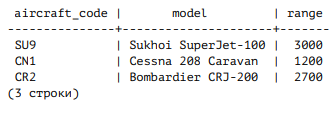
Шаблон в операторе LIKE всегда покрывает всю анализируемую строку. Поэтому если требуется отыскать некоторую последовательность символов где-то внутри строки, то шаблон должен начинаться и завершаться символом «%».

Существует и оператор NOT LIKE. Например, если мы захотим узнать, какими самолетами, кроме машин компаний Airbus и Boeing, располагает наша авиакомпания, то придется усложнить условие:

**SELECT \* FROM aircrafts**

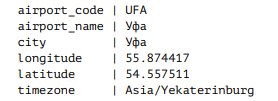
**WHERE model NOT LIKE 'Airbus%'**

**AND model NOT LIKE 'Boeing%';**



Кроме символа «%» в шаблоне может использоваться и символ подчеркивания — «\_», который соответствует в точности одному любому символу. В качестве примера найдем в таблице «Аэропорты» те из них, которые имеют названия длиной три символа (буквы). С этой целью зададим в качестве шаблона строку, состоящую из трех символов «\_».

**SELECT \* FROM airports WHERE airport\_name LIKE '\_\_\_';**



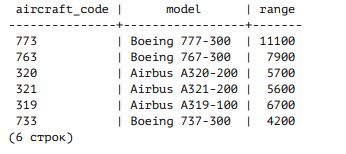
**Задание**

Создайте различные запросы к таблице товаров с использованием оператора LIKE

**Использование регулярных выражений с оператором LIKE**

Существует ряд операторов для работы с регулярными выражениями POSIX. Эти операторы имеют больше возможностей, чем оператор LIKE. Для того чтобы выбрать, например, самолеты компаний A

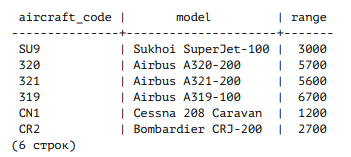
**SELECT \* FROM aircrafts WHERE model ~ '^(A|Boe)';**



Оператор ~ ищет совпадение с шаблоном с учетом регистра символов. Символ «^» в начале регулярного выражения означает, что поиск совпадения будет привязан к началу строки. Если же требуется проверить наличие такого символа в составе строки, то перед ним нужно поставить символ обратной косой черты «\». Выражение в круглых скобках означает альтернативный выбор между значениями, разделяемыми символом «|». Поэтому в выборку попадут значения, начинающиеся либо на «A», либо на «Boe». Для инвертирования смысла оператора ~ нужно перед ним добавить знак «!». В качестве примера отыщем модели самолетов, которые не завершаются числом 300.

**SELECT \* FROM aircrafts WHERE model !~ '300$';**

В этом регулярном выражении символ «$» означает привязку поискового шаблона к концу строки. Если же требуется проверить наличие такого символа в составе строки, то перед ним нужно поставить символ обратной косой черты «\».



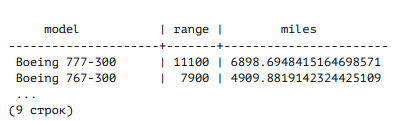
**Задание**

Создайте различные запросы к таблице товаров с использованием оператора LIKE и регулярных выражений.

**Вычисляемые столбцы**

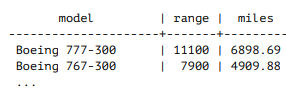
При выборке данных можно проводить вычисления и получать в результирующей таблице вычисляемые столбцы. Если мы захотим представить дальность полета не только в километрах, но и в милях, то нужно вычислить это выражение и для удобства присвоить новому столбцу псевдоним с помощью ключевого слова AS.

**SELECT model, range, range / 1.609 AS miles FROM aircrafts;**

****

По всей вероятности, такая высокая точность представления значений в милях не требуется, поэтому мы можем уменьшить ее до разумного предела в два десятичных знака:

**SELECT model, range, round( range / 1.609, 2 ) AS miles FROM aircrafts;**

****

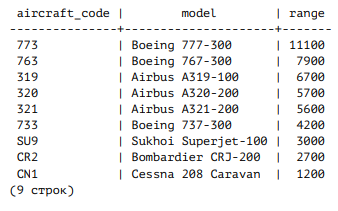
**Задание**

Создайте различные запросы к таблице товаров с вычисляемых столбцов.

**Упорядочивание строк**

Теперь обратимся к такому вопросу, как упорядочение строк при выводе. Если не принять специальных мер, то СУБД не гарантирует никакого конкретного порядка строк в результирующей выборке. Для упорядочения строк служит предложение ORDER BY, которое мы уже использовали ранее. Однако мы не говорили, что можно задать не только возрастающий, но также и убывающий порядок сортировки. Например, если мы захотим разместить самолеты в порядке убывания дальности их полета, то нужно сделать так:

**SELECT \* FROM aircrafts ORDER BY range DESC;**

****

**Задание**

Создайте различные запросы к таблице товаров с использованием упорядочивания строк.

Отбор уникальных значений столбца с использованием ключевого слова DISTINCT

Мы детально разобрались с таблицей «Самолеты» и теперь обратим наше внимание на таблицу «Аэропорты»). В ней есть столбец «Часовой пояс» (timezone). Давайте посмотрим, в каких различных часовых поясах располагаются аэропорты. Если сделать традиционную выборку

**SELECT timezone FROM airports;**

то мы получим список значений, среди которых будет много повторяющихся. Конечно, это неудобно. Для того чтобы оставить в выборке только неповторяющиеся значения, служит ключевое слово DISTINCT:

**SELECT DISTINCT timezone FROM airports ORDER BY 1;**

Обратите внимание, что столбец, по значениям которого будут упорядочены строки, указан не с помощью его имени, а с помощью его порядкового номера в предложении SELECT.

Получим такой результат:



**Задание**

Создайте различные запросы к таблице товаров с использованием DISTINCT.

**Отбор определенного количества значений и пропуски строк**

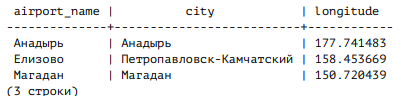
В таблице «Аэропорты» более ста записей. Если мы поставим задачу найти три самых восточных аэропорта, то для ее решения подошел бы такой алгоритм: отсортировать строки в таблице по убыванию значений столбца «Долгота» (longitude) и включить в выборку только первые три строки. Как отсортировать строки по убыванию значений какого-либо столбца, вы уже знаете, а для того чтобы ограничить число строк, включаемых в результирующую выборку, служит предложение LIMIT.

**SELECT airport\_name, city, longitude**

**FROM airports**

**ORDER BY longitude DESC**

**LIMIT 3;**

****

А как найти еще три аэропорта, которые находятся немного западнее первой тройки, т. е. занимают места с четвертого по шестое? Алгоритм будет почти таким же, как в первой задаче, но он будет дополнен еще одним шагом: нужно пропустить три первые строки, прежде чем начать вывод. Для пропуска строк служит предложение OFFSET.

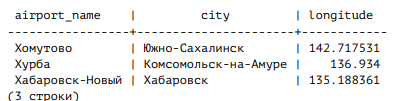
**SELECT airport\_name, city, longitude**

**FROM airports**

**ORDER BY longitude DESC**

**LIMIT 3**

**OFFSET 3;**



**Задание**

Создайте различные запросы к таблице товаров с использованием **LIMIT**

**OFFSET**.

**Условные выражения**

В таблице «Самолеты» есть столбец «Максимальная дальность полета» (range). Мы можем дополнить вывод данных из этой таблицы столбцом «Класс самолета», имея в виду принадлежность каждого самолета к классу дальнемагистральных, среднемагистральных или ближнемагистральных судов. Для этого подойдет конструкция

**CASE WHEN условие THEN выражение**

**[WHEN ... ]**

**[ELSE выражение ]**

**END**

Воспользовавшись этой конструкцией в предложении SELECT и назначив новому столбцу имя с помощью ключевого слова AS, получим следующий запрос:

**SELECT model, range,**

**CASE WHEN range < 2000 THEN 'Ближнемагистральный'**

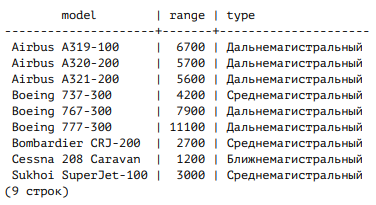
**WHEN range < 5000 THEN 'Среднемагистральный'**

**ELSE 'Дальнемагистральный'**

**END AS type**

**FROM aircrafts**

**ORDER BY model;**

****

**Задание**

Создайте новый столбец с использованием условных выражений.