**ТЕМА № 2. ОСНОВЫ ЯЗЫКА SQL**

**ОПЕРАТОР ВЫБОРКИ ДАННЫХ**

**Цели занятия:**

1. Развивать практические умения в написании SQL-команд для выборки данных из таблицы.
2. Развивать практические умения в написании SQL-команд для выборки данных с использованием обобщающих функций, сортировки и группировки результатов.

**УЧЕБНЫЕ ВОПРОСЫ**

**Задание В.1:** Изучить БД с именем «*database0*», которая уже загружена в ваш профиль.

***Решение:***

В представленной БД содержится одна таблица «*table*\_1» с данными, структура и содержимое которой (частично) представлено на рисунке 6.

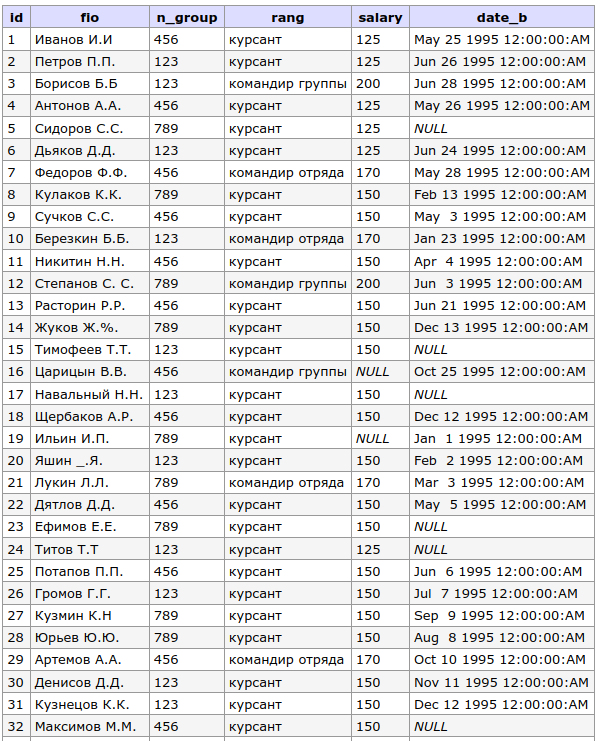


Рисунок 6

Таблица «*table*\_1» содержит информацию о студентах и сотрудниках (далее будут называться только «Сотрудники»), при этом столбцы имеют следующее значение: «id» – идентификатор (первичный ключ), «fio» – фамилия и инициалы, «n\_group» – номер группы, «rang» – должность, «date\_b» – дата рождения, «salary» – должностной оклад.

**1. ПРОСТЫЕ ЗАПРОСЫ. СОРТИРОВКА РЕЗУЛЬТАТОВ**

**Задание 1.1:** Выбор столбцов с помощью предложений SELECT, FROM, DISTINCT, INTO, AS:

**а)** выбрать всю информацию из таблицы *table*\_1;

**б)** выбрать информацию о должностях из таблицы *table*\_1;

**в)** выбрать неповторяющиеся должности из таблицы *table*\_1;

**г)** выбрать несколько столбцов: должность, фамилию, номер группы и дату рождения сотрудников из таблицы *table*\_1;

**д)** по заданию 1.1 (г) задать псевдонимы столбцам *rang*, *fio*, *n\_group*, *date\_r* – «*должность*», «*фамилия*», «*номер группы*», «*дата рождения*».

**Задание 1.2:** Создание производных столбцов:

**а)** конкатенировать через пробел должность и фамилию сотрудников из таблицы *table*\_1 в одну строку и задать псевдоним результирующему столбцу «*список*»;

**SELECT rang + ' ' + fio AS список FROM table\_1**

**б)** выбрать должность, фамилию, текущий оклад сотрудников из таблицы *table*\_1 и увеличенный на 20 у.е. (уменьшенный на 20 у.е., увеличенный в 1,5 раза, уменьшенный в 1.5 раза) оклад и задать псевдоним результирующему столбцу «*new salary*»;

*Для округления результата можно воспользоваться функцией:*

**SELECT rang, fio, salary, round (salary/1.5, 1)**

**AS "new salary" FROM table\_1**

**в)** выбрать идентификатор, идентификатор уменьшенный вдвое и фамилию сотрудников из таблицы *table*\_1 и задать псевдоним результирующему столбцу «*new id*»;

***Следует обратить внимание*** на то, что результирующий столбец будет того же типа, что и исходный (*id* – целочисленный тип). В связи с этим результирующий столбец принадлежит к целочисленному типу, при этом дробная часть результата игнорируется.

**Задание 1.3:** Фильтрация строк с помощью предложения WHERE:

**а)** использование операций сравнения:

– выбрать всю информацию из таблицы *table*\_1 только о сотрудниках в должности «командир группы»;

***Следует обратить внимание*** на то, что регистр в одинарных кавычках не имеет значения.

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках 123 группы;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках со всех групп, кроме 123 группы;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, родившихся после 30 июня 1995 года;

***Следует обратить внимание*** на то, что даты сравниваются хронологически, при этом символ «>» означает позже, а символ «<» – раньше.

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, у которых оклад не менее 170 у.е.

**б)** использование оператора BETWEEN … AND, который проверяет значения на принадлежность диапазону:

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, родившихся во втором полугодии 1995 года;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, родившихся во первом и четвертом кварталах 1995 года;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, у которых оклад от 150 до 200 у.е.

**в)** использование оператора IN, который проверяет значения на членство в множестве:

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках 123 и 456 групп;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках во всех должностях, кроме «командир отряда» и «командир группы».

**г)** использование оператора IS NULL, который проверяет значения на равенство NULL:

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках с неизвестной датой рождения;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках с известным должностным окладом.

**д)** использование оператора LIKE, который проверяет значения на соответствие шаблону:

***Знак «\_» в предложении LIKE означает любой одиночный символ, а знак «%» – любую последовательность из нуля и более символов.***

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, у которых фамилия начинается с буквы «А»;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, у которых вторая буква в фамилии «о»;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, у которых в фамилии встречается сочетание букв «ин»;

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, у которых в фамилии или инициалах встречается символ «\_»;

**SELECT \* FROM table\_1 WHERE fio LIKE '%#\_%' escape '#'**

***Следует отметить,*** что вместо символа «#» в принципе может быть любой символ (за исключением символов «\_» и «%»).

– выбрать всю информацию из таблицы *table*\_1 о сотрудниках, у которых в фамилии или инициалах встречается символ «%».

**е)** комбинирование условий с помощью операторов AND и OR:

– выбрать всю информацию из таблицы *table*\_1 только о командирах отрядов 123 группы;

– выбрать всю информацию из таблицы *table*\_1 о всех сотрудниках 123 группы, родившихся в июне;

– выбрать всю информацию из таблицы *table*\_1 о командирах групп и о командирах отрядов 123 группы;

– выбрать всю информацию из таблицы *table*\_1 о командирах отрядов 123 и 456 групп, а также о сотрудниках 123 группы, родившихся в июне и сентябре.

**Задание 1.4:** Сортировка строк с помощью предложения ORDER BY:

**а)** выбрать всю информацию из таблицы *table*\_1 и отсортировать ее по фамилии в алфавитном порядке;

где «2» – порядковый номер столбца «*fio*» в таблице *table*\_1.

**б)** выбрать всю информацию из таблицы *table*\_1 и отсортировать ее по фамилии в порядке обратном алфавитному;

**в)** выбрать номера групп и фамилии сотрудников из таблицы *table*\_1 и провести сортировку по номеру группы в порядке возрастания, а в группах по фамилии в алфавитном порядке;

**г)** выбрать номера групп, фамилии и оклады сотрудников из таблицы *table*\_1 и провести сортировку по номеру группы в порядке возрастания, а в группах по должностному окладу в порядке убывания;

***Следует обратить внимание*** на сортировку пустых значений NULL, которые располагаются в конце результирующего набора.

**2. ИСПОЛЬЗОВАНИЕ ОБОБЩАЮЩИХ ФУНКЦИЙ. ГРУППИРОВАНИЕ РЕЗУЛЬТАТОВ**

**2.1. ИСПОЛЬЗОВАНИЕ ОБОБЩАЮЩИХ ФУНКЦИЙ**

**Задание 2.1:** Поиск минимума посредством функции MIN:

**а)** найти самый большой размер оклада в таблице *table*\_1;

**б)** найти дату рождения самого старшего по возрасту сотрудников из 789 группы;

**в)** найти в списке сотрудников 456 группы, составленном в алфавитном порядке, первую фамилию.

**Задание 2.2:** Вычисление суммы с помощью функции SUM.

**а)** вычислить суммарный оклад всех сотрудников в таблице *table*\_1;

**б)** вычислить суммарный оклад командиров групп;

**в)** вычислить сумму различных значений в столбце «s*alary*» таблицы *table*\_1.

**Задание 2.3:** Вычисление среднего значения с помощью функции AVG.

**а)** вычислить средний оклад всех командиров отрядов;

*Следует обратить внимание* на то, что пустые значения *NULL* участия в расчете среднего значения не принимают. Если имеется необходимость заменить пустые значения *NULL* на «0», то необходимо использовать функцию COALESCE.

**б)** выбрать оклад сотрудников из таблицы *table*\_1, заменив пустые значения NULL на «0»;

**SELECT COALESCE (salary, 0) FROM table\_1**

**в)** вычислить средний оклад всех командиров отрядов с учетом значений NULL, заменив их на «0»;

**SELECT AVG(COALESCE (salary, 0)) FROM table\_1**

**WHERE rang = 'командир отряда'**

**г)** вычислить среднее значение множества различных значений в столбце «s*alary*» таблицы *table*\_1.

**SELECT AVG (DISTINCT salary) FROM table\_1**

**Задание 2.4:** Вычисление количества строк с помощью функции COUNT:

**а)** вычислить количество всех строк в таблице *table*\_1;

**б)** вычислить количество записей в столбце «*salary*»;

**в)** вычислить количество сотрудников 123 группы;

**г)** вычислить количество различных должностей в таблице *table*\_1.

**2.2. ГРУППИРОВАНИЕ РЕЗУЛЬТАТОВ**

**Задание 2.6:** Группирование строк с использованием предложения GROUP BY:

**а)** выбрать номера групп и определить количество сотрудников по группам, задать результирующему столбцу задать псевдоним «*количество*»;

**б)** выбрать номера групп и определить суммарный и средний оклад по группам, задать результирующим столбцам задать псевдонимы «*сумма*» и «*среднее*»;

Д78

**в)** выбрать должности и определить количество человек, суммарный и средний оклад по должностям, задать результирующим столбцам псевдонимы «*количество*», «*сумма*» и «*среднее*»;

**г)** выбрать номера групп, должности и определить количество человек и суммарный оклад по группам и должностям, задать результирующим столбцам псевдонимы «*количество*» и «*сумма*».

**Задание 2.7:** Фильтрация групп с помощью предложения HAVING:

**а)** по заданию 2.6 (а) результат представить только о тех группах, в которых количество сотрудников не меньше 13;

**б)** по заданию 2.6 (б) результат представить только о тех группах, в которых суммарный оклад находится в диапазоне от 1500 до 2000 у.е.;

**в)** по заданию 2.6 (в) результат представить только о тех должностях, количество которых больше 5 и средний оклад больше 150 у.е.:

**г)** по заданию 2.6 (в) результат представить только о тех должностях, количество которых в группе меньше 5.

**ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ**

***Задание 1.*** Создание производных столбцов:

**а)** выбрать идентификатор, остаток от деления идентификатора по модулю 2 и фамилию сотрудников из таблицы *table*\_1 и задать псевдоним результирующему столбцу «*new id*»;

***Следует обратить внимание*** на то, что операция остаток от деления применяется только к целочисленному типу данных, поэтому применение этой операции к столбцу «*salary*» приведет к ошибке несовместимости типов.

**б)** выбрать должность, фамилию, дату рождения сотрудников из таблицы *table*\_1 и дату за ***x*** дней до даты рождения курсантов и задать псевдоним результирующему столбцу «*new date*».

***Задание 2.*** Сортировка строк с помощью предложения ORDER BY.

Выбрать должности, фамилии и даты рождения сотрудников из таблицы *table*\_1, провести сортировку по должностям в порядке обратном алфавитному, а в должностях по дате рождения в порядке убывания (начиная с самого молодого).

***Задание 3.*** Поиск максимума посредством функции MAX.

**а)** найти самый большой номер группы в таблице *table*\_1;

**б)** найти дату рождения самого младшего по возрасту сотрудников ***y*** группы;

**в)** найти в списке сотрудников ***y*** группы, составленном в алфавитном порядке, последнюю фамилию.

***Задание 4.*** По заданию 2.6 (а) провести сортировку по количеству сотрудников в порядке возрастания;

Варианты по списку:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ***x*** | ***y*** |  | ***x*** | ***y*** |
| **1** | 1 | 123 | **16** | 8 | 123 |
| **2** | 2 | 456 | **17** | 1 | 456 |
| **3** | 3 | 789 | **18** | 2 | 789 |
| **4** | 4 | 123 | **19** | 3 | 123 |
| **5** | 5 | 456 | **20** | 4 | 456 |
| **6** | 6 | 789 | **21** | 5 | 789 |
| **7** | 7 | 123 | **22** | 6 | 123 |
| **8** | 8 | 456 | **23** | 7 | 456 |
| **9** | 1 | 789 | **24** | 8 | 789 |
| **10** | 2 | 123 | **25** | 1 | 123 |
| **11** | 3 | 456 | **26** | 2 | 456 |
| **12** | 4 | 789 | **27** | 3 | 789 |
| **13** | 5 | 123 | **28** | 4 | 123 |
| **14** | 6 | 456 | **29** | 5 | 456 |
| **15** | 7 | 789 | **30** | 6 | 789 |