МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение

высшего образования

**«Московский технический университет связи и информатики»**

**(МТУСИ)**

Кафедра

«Сетевые информационные технологии и сервисы»

Лабораторная работа 4

По дисциплине «Интеллектуальные базы данных»

«Создание простых запросов на выборку»

Выполнил студент:

М092301(75) Леонов Н.Н.

Вариант 17

Проверил:

Ст.пр. Фатхулин Т.Д.

Москва 2024

1. **Цель работы**

В данной лабораторной работе необходимо выполнить запросы на выборку с использованием операторов и функций.

1. **Задание**

При выполнении лабораторной работы необходимо для заданной предметной области средствами MySQL:

* построить несколько простых запросов на выборку с использованием операторов и функций MySQL;
* составить отчет по лабораторной работе.

1. **Теоретические сведения**

Рассмотрим следующие вопросы:

* выборка данных из одной таблицы с помощью оператора *SELECT*;
* использование в запросах операторов и встроенных функций MySQL.

Для выполнения запросов (извлечения строк из одной или нескольких таблиц БД) используется оператор *SELECT*. Результатом запроса всегда является таблица. Результаты запроса могут быть использованы для создания новой таблицы. Таблица, полученная в результате запроса, может стать предметом дальнейших запросов.

Общая форма оператора *SELECT*:

*SELECT столбцы FROM таблицы*

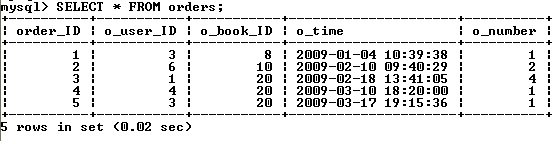
*[WHERE условия]*

*[GROUP BY группа [HAVING групповые\_условия] ]*

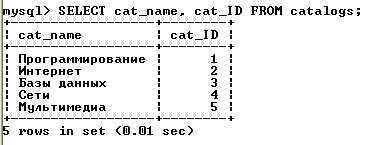
*[ORDER BY имя\_поля]*

*[LIMIT пределы];*

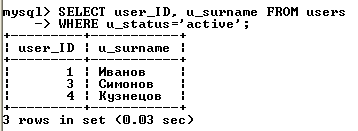
Оператор *SELECT* имеет много опций. Их можно использовать или не использовать, но они должны указываться в том порядке, в каком они приведены. Если требуется вывести все столбцы таблицы, необязательно перечислять их после ключевого слова *SELECT*, достаточно заменить этот список символом \*.



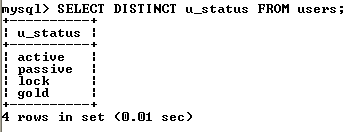
Список столбцов в операторе *SELECT* используют, если нужно изменить порядок следования столбцов в результирующей таблице или выбрать часть столбцов.



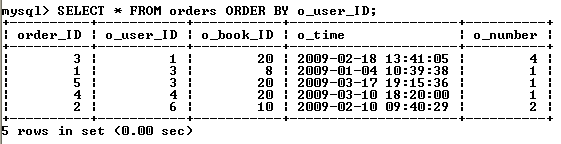
**Условия выборки**. Гораздо чаще встречается ситуация, когда необходимо изменить количество выводимых строк. Для выбора записей, удовлетворяющих определенным критериям поиска, можно использовать конструкцию *WHERE*.



В запросе можно использовать ключевое слово *DISTINCT*, чтобы результат не содержал повторений уже имеющихся значений, например:

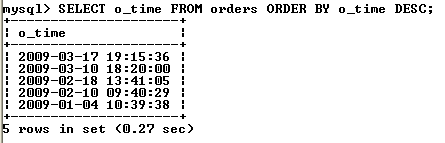


**Сортировка**. Результат выборки – записи, расположенные в том порядке, в котором они хранятся в БД. Чтобы отсортировать значения по одному из столбцов, необходимо после конструкции *ORDER BY* указать этот столбец, например:



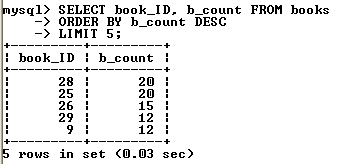
Сортировку записей можно производить по нескольким столбцам (их следует указать после слов *ORDER BY* через запятую). Число столбцов, указываемых в конструкции *ORDER BY*, не ограничено.

По умолчанию сортировка производится в прямом порядке (записи располагаются от наименьшего значения поля сортировки до наибольшего). Обратный порядок сортировки реализуется с помощью ключевого слова *DESC*:

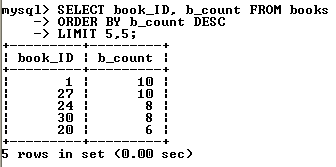


Для прямой сортировки существует ключевое слово *ASC*, но так как записи сортируются в прямом порядке по умолчанию, данное ключевое слово опускают.

**Ограничение выборки**. Результат выборки может содержать тысячи записей, вывод и обработка которых занимают значительное время. Поэтому информацию часто разбивают на страницы и предоставляют ее пользователю частями. Постраничная навигация используется при помощи ключевого слова *LIMIT*, за которым следует число выводимых записей. Следующий запрос извлекает первые 5 записей, при этом осуществляется обратная сортировка по полю *b\_count*:



Для извлечения следующих пяти записей используется ключевое слово *LIMIT* с двумя цифрами. Первая указывает позицию, начиная с которой необходимо вернуть результат, вторая цифра – число извлекаемых записей, например:



При определении смещения нумерация строк начинается с нуля (поэтому в последнем примере для шестой строки указано смещение 5).

**Группировка записей**. Конструкция *GROUP ВУ* позволяет группировать извлекаемые строки. Она полезна в комбинации с функциями, применяемыми к группам строк. Эти функции (табл. 6) называются агрегатами (суммирующими функциями) и вычисляют одно значение для каждой группы, создаваемой конструкцией *GROUP BY*. Функции позволяют узнать число строк в группе, подсчитать среднее значение, получить сумму значений столбцов. Результирующее значение рассчитывается для значений, не равных *NULL* (исключение – функция *COUNT(\*)*). Допустимо использование этих функций в запросах без группировки (вся выборка – одна группа).

Пример использования функции *COUNT( )*, которая возвращает число строк в таблице, значения указанного столбца для которых отличны от *NULL*:

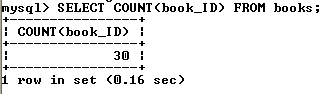
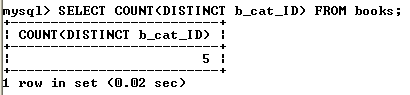


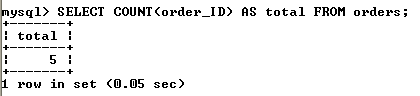
Таблица 6

|  |  |
| --- | --- |
| **Обозначение** | **Описание** |
| *AVG ( [DISTINCT]  expr)* | Возвращает среднее значение аргумента *expr*. В качестве аргумента обычно выступает имя столбца. Необязательное слово *DISTINCT* позволяет обрабатывать только уникальные значения столбца *expr* |
| *COUNT ( )* | Подсчитывает число записей и имеет несколько форм. Форма *COUNT (выражение)* возвращает число записей в таблице, поле *выражение* для которых не равно *NULL*. Форма *COUNT(\*)* возвращает общее число строк в таблице независимо от того, принимает какое-либо поле значение *NULL* или нет. Форма *COUNT (DISTINCT выражение1, выражение2, ... )* позволяет использовать ключевое слово *DISTINCT*, которое позволяет подсчитать только уникальные значения столбца |
| *MIN ( [DISTINCT]  expr)* | Возвращает минимальное значение среди всех непустых значений выбранных строк в столбце *expr*. Необязательное слово *DISTINCT* позволяет обрабатывать только уникальные значения столбца *expr* |
| *MAX ( [DISTINCT]  expr)* | Возвращает максимальное значение среди всех непустых значений выбранных строк в столбце *expr*. Необязательное слово *DISTINCT* позволяет обрабатывать только уникальные значения столбца *expr* |
| *STD (expr)* | Возвращает стандартное среднеквадратичное отклонение в аргументе *expr* |
| *STDDEV\_SAMP (expr)* | Возвращает выборочное среднеквадратичное отклонение в аргументе *expr* |
| *SUM ( [DISTINCT]  expr)* | Возвращает сумму величин в столбце *expr*. Необязательное слово *DISTINCT* позволяет обрабатывать только уникальные значения столбца *expr* |

Использование ключевого слова *DISTINCT* с функцией *COUNT( )* позволяет вернуть число уникальных значений *b\_cat\_ID* в таблице *books*, например:



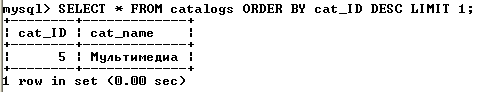
В *SELECT*-запросе столбцу можно назначить новое имя с помощью оператора *AS*. Например, результату функции *COUNT( )* присваивается псевдоним *total*:



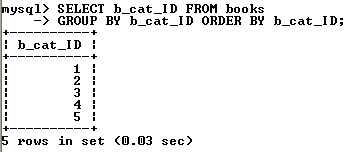
Использование функций в конструкции *WHERE* приведет к ошибке. В следующем примере показана попытка извлечения из таблицы *catalogs* записи с максимальным значением поля *cat\_ID*:



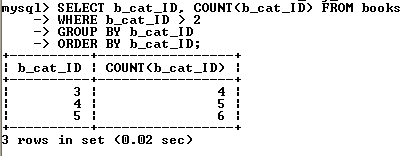
Решение задачи следует искать в использовании конструкции *ORDER BY*:



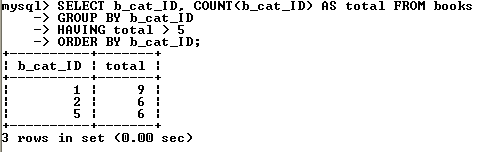
Для извлечения уникальных записей используют конструкцию *GROUP BY* с именем столбца, по которому группируется результат:



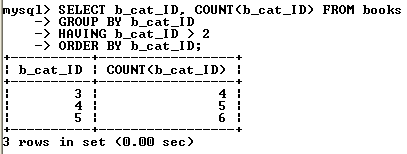
При использовании *GROUP BY* возможно использование условия *WHERE*:



Часто при задании условий требуется ограничить выборку по результату функции (например, выбрать каталоги, где число товарных позиций больше 5). Использование для этих целей конструкции *WHERE* приводит к ошибке. Для решения этой проблемы вместо ключевого слова *WHERE* используется ключевое слово *HAVING*, располагающееся за конструкцией *GROUP BY*:



Запрос, извлекающий уникальные значения столбца *b\_cat\_ID*, большие двух:



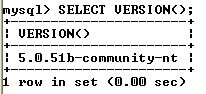
При этом в случае использования ключевого слова *WHERE* сначала производится выборка из таблицы с применением условия и лишь затем группировка результата, а в случае использования ключевого слова *HAVING* сначала происходит группировка таблицы и лишь затем выборка с применением условия. Допускается использование условия *HAVING* без группировки *GROUP BY*.

**Использование функций**. Для решения специфических задач при выборке удобны встроенные функции MySQL. Большинство функций предназначено для использования в выражениях *SELECT* и *WHERE*. Существуют также специальные функции группировки для использования в выражении *GROUP BY* (см. выше).

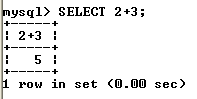
Каждая функция имеет уникальное имя и может иметь несколько аргументов (перечисляются через запятую в круглых скобках). Если аргументы отсутствуют, круглые скобки все равно следует указывать. Пробелы между именем функции и круглыми скобками недопустимы.

Число доступных для использования функций велико, в приложениях приведены наиболее полезные из них.

Пример использования функции, возвращающей версию сервера MySQL:



Отметим также возможность использования оператора *SELECT* без таблиц вообще. В такой форме *SELECT* можно использовать как калькулятор:



Можно вычислить любое выражение без указания таблиц, получив доступ ко всему разнообразию математических и других операторов и функций. Возможность выполнять математические расчеты на уровне *SELECT* позволяет проводить финансовый анализ значений таблиц и отображать полученные результаты в отчетах. Во всех выражениях MySQL (как в любом языке программирования) можно использовать скобки, чтобы контролировать порядок вычислений.

**Операторы**. Под операторами подразумеваются конструкции языка, которые производят преобразование данных. Данные, над которыми совершается операция, называются операндами.

В MySQL используются три типа операторов:

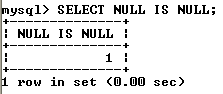
* арифметические операторы;
* операторы сравнения;
* логические операторы.

*Арифметические операции.*В MySQL используются обычные арифметические операции: сложение (+), вычитание (–), умножение (\*), деление (/) и целочисленное деление *DIV* (деление и отсечение дробной части). Деление на 0 дает безопасный результат *NULL*.

*Операторы сравнения.* При работе с операторами сравнения необходимо помнить о том, что, за исключением нескольких особо оговариваемых случаев, сравнение чего-либо со значением *NULL* дает в результате *NULL*. Это касается и сравнения значения *NULL* со значением *NULL*:



Корректнее использовать следующий запрос:



Поэтому следует быть предельно внимательными при работе с операторами сравнения, если операнды могут принимать значения *NULL*.

Наиболее часто используемые операторы сравнения приведены в табл. 7.

*Логические операторы.* MySQL поддерживает все обычные логические операции, которые можно использовать в выражениях. Логические выражения в MySQL могут принимать значения 1 (истина), 0 (ложь) или *NULL*.

Кроме того, следует учитывать, что MySQL интерпретирует любое ненулевое значение, отличное от *NULL*, как значение «истина».

1. **Выполнение лабораторной работы**

Выполним несколько запросов для выборки данных из БД Cost Accounting.

Выведем все столбцы таблицы Employee с помощью команды SELECT \* FROM Employee (рис. 1).

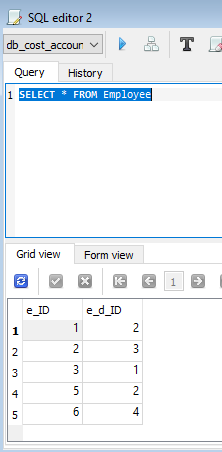


Рисунок 1 – Выборка всех столбцов из таблицы Employee

Выведем название отдела, в котором числится больше 10 сотрудников с помощью команды SELECT d\_name FROM Department WHERE d\_count\_of\_employees > 10 (рис. 2).

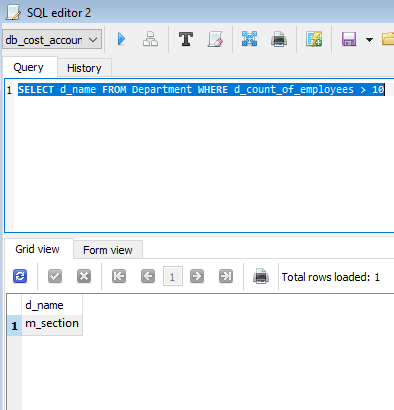


Рисунок 2 – Получение названия отдела, в котором больше 10 сотрудников

Отсортируем все столбцы данной таблицы по количеству сотрудников с помощью команды SELECT \* FROM Department ORDER BY d\_count\_of\_employees (рис. 3).

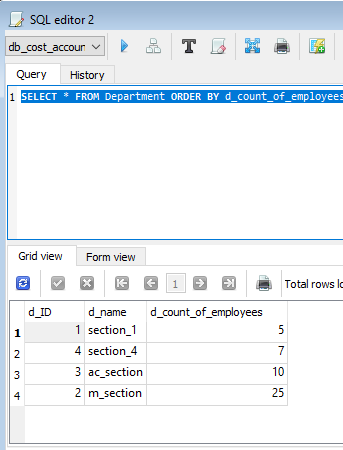


Рисунок 3 – Результат сортировки таблицы Department по количеству сотрудников

Также произведем сортировку столбцов таблицы Purchase по дате, начиная с самых свежих заказов с помощью команды SELECT \* FROM Purchase ORDER BY p\_date DESC (рис. 4).

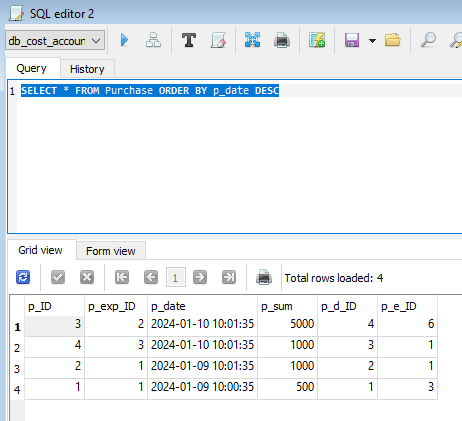


Рисунок 4 – Результат сортировки таблицы Purchase по дате в обратном порядке

Сделаем выборку стоимости двух наиболее свежих заказов с помощью команды SELECT p\_sum FROM Purchase ORDER BY p\_date DESC LIMIT 2 (рис. 5).

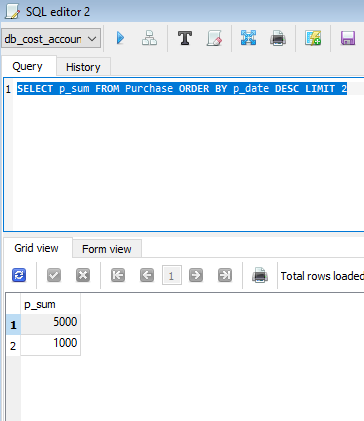


Рисунок 5 – Результат выборки стоимости двух наиболее свежих заказов

Выведем максимальный и минимальный денежный лимит из таблицы Expenses с помощью команды SELECT MAX(exp\_limit) AS max, MIN(exp\_limit) AS min FROM Expenses (рис. 6).

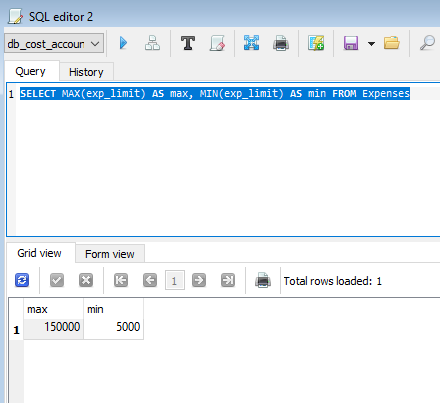


Рисунок 6 – Выборка макс. и мин. денежного лимита

# **Выводы**

В ходе лабораторной работы были произведены простые запросы на выборки с использованием операторов и функций MySQL на основе заполненной базы данных «db\_cost\_accounting» в предыдущей лабораторной работе №3.