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 5

По дисциплине «Интеллектуальные базы данных»

«Создание сложных 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 объединения и вложенных запросов.

1. **Задание**

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

* для заданной предметной области построить многотабличный запрос на выборку с использованием объединения;
* для заданной предметной области построить запрос на выборку, содержащий вложенный запрос;
* составить отчет по лабораторной работе.

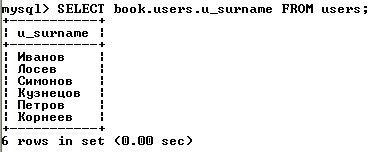
1. **Теоретические сведения**

Рассмотрим следующие вопросы:

* использование объединений в запросах к нескольким таблицам;
* создание вложенных запросов.

В реаль­ных приложениях часто требуется использовать сразу несколько таблиц БД. Запросы, которые обращаются одновременно к нескольким таблицам, называются многотабличными или сложными запросами.

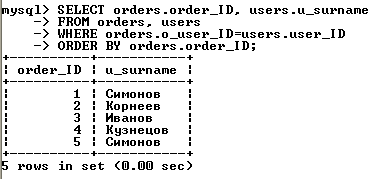
**Абсолютные ссылки на базы данных и таблицы**. В запросе мож­но прямо указывать необходимую БД и таблицу. Напри­мер, можно представить ссылку на столбец *u\_surname* из таблицы *users* в виде *users.u\_surname*. Аналогично можно уточнить БД, таблица из которой упоминается в запросе. Если необходимо, то вместе с БД и таблицей можно указать и столбец, например:



При использовании сложных запросов это позволяет избежать двусмысленности при указании источника необходи­мой информации.

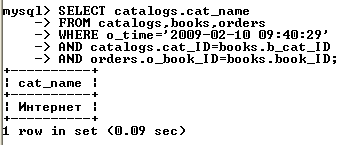
**Использование объединений для запросов к нескольким таблицам**. Хорошо спроектированная реляционная БД эффективна из-за связей между таблицами. При выборе информации из нескольких таблиц такие связи называют объединениями.

В качестве примера объединения двух таблиц рассмотрим запрос, извлекающий из БД *book* фамилии покупателей вместе с номерами сделанных ими заказов:

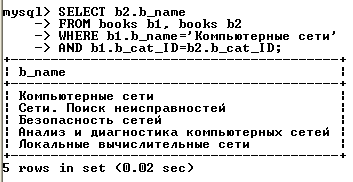


Выражение *WHERE* важно с точки зрения получения ре­зультата. Набор условий, используемых для объединения таб­лиц, называют условием объединения. В данном примере условие связы­вает таблицы *orders* и *users* по внешним ключам.

Объединение нескольких таблиц аналогично объединению двух таблиц. Например, необходимо выяснить, какому каталогу принадлежит товарная позиция из заказа, сделанного 10 февраля 2009 г. в 09:40:29:



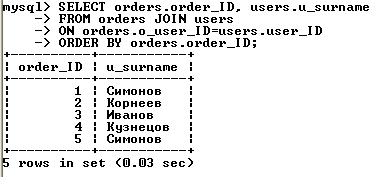
**Самообъединение таблиц**. Можно объединить таб­лицу саму с собой (когда интересуют связи между строками одной и той же таблицы). Пусть нужно выяснить, какие книги есть в каталоге, содержащем книгу с названием «Компьютерные сети». Для этого необходимо найти в таблице *books* номер каталога (*b\_cat\_ID*) с этой книгой, а затем посмотреть в таблице *books* книги этого каталога.



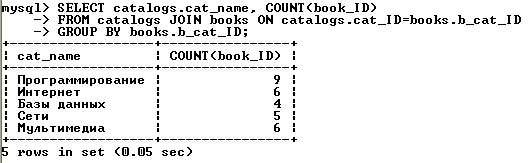
В этом запросе для таблицы *books* определены два разных псев­донима (две от­дельных таблицы *b1* и *b2*, которые должны содержать одни и те же данные). После этого они объединяются, как любые другие таблицы. Сна­чала ищется строка в таблице *b1*, а затем в таблице *b2* – строки с тем же значением номера каталога.

**Основное объединение**. Набор таблиц, перечисленных в выражении *FROM* и разделенных запятыми, – это декартово произведение (полное или перекрестное объединение), которое возвращает полный набор ком­бинаций. Добавление к нему условного выражения *WHERE* превраща­ет его в объединение по эквивалентности, ограни­чивающее число возвращаемых запросом строк.

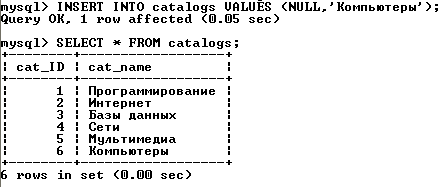
Вместо запятой в выражении *FROM* можно использовать ключевое слово *JOIN*. В этом случае вместо *WHERE* лучше использовать ключевое слово *ON*:



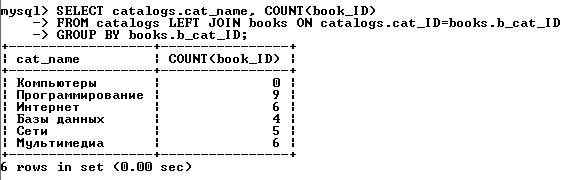
Вместо *JOIN* с тем же результатом можно использовать *CROSS JOIN* (перекрестное объединение) или *INNER JOIN* (внутреннее объединение). Пример запроса, выдающего число товарных позиций в каталогах:



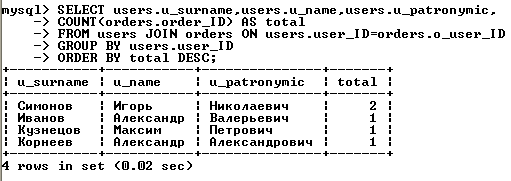
Допустим, происходит расширение ассортимента и в списке каталогов появляется новый каталог «Компьютеры»:



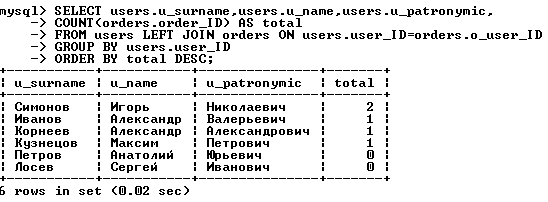
Предыдущий запрос не отразит наличие нового каталога (таблица *books* не содержит записей, относящихся к новому каталогу). Выходом является использование левого объединения (таблица *catalogs* должна быть левой таблицей):



Пусть нужно вывести список покупателей и число осуществленных ими покупок, причем покупателей необходимо отсортировать по убыванию числа заказов:

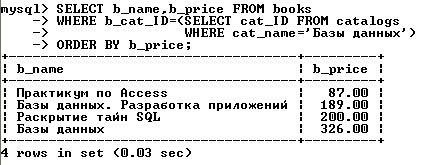


В список не входят покупатели, которые не сделали ни одной покупки. Чтобы вывести полный список покупателей, необходимо вместо перекрестного объединения таблиц *users* и *orders* использовать левое объединение (левой таблицей должна быть таблица *users*):



**Вложенный запрос**. Позволяет использовать результат, возвращаемый одним запросом, в другом запросе. Так как результат возвращает только оператор *select*, то в качестве вложенного запроса всегда выступает *SELECT*-запрос. В качестве внешнего запроса может выступать запрос с участием любого SQL-оператора: *select, insert, update, delete, create table* и др.

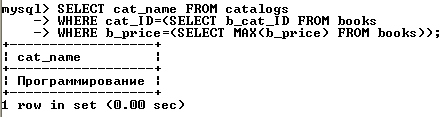
Пусть требуется вывести названия и цены товарных позиций из таблицы *books* для каталога «Базы данных» таблицы *catalogs*:



Получить аналогичный результат можно при помощи многотабличного запроса, но имеется ряд задач, которые решаются только при помощи вложенных запросов. Вложенный запрос может применяться не только с условием *WHERE*, но и в конструкциях *DISTINCT, GROUP BY, ORDER BY, LIMIT* и т. д. Различают:

* вложенные запросы, возвращающие одно значение;
* вложенные запросы, возвращающие несколько строк.

В первом случае вложенный запрос возвращает скалярное значение или литерал, которое используется во внешнем запросе (подставляет результат на место своего выполнения). Например, необходимо определить название каталога, содержащего самую дорогую товарную позицию:



Наиболее часто вложенные запросы используются в операциях сравнения в условиях, которые задаются ключевыми словами *WHERE, HAVING* или *ON*.

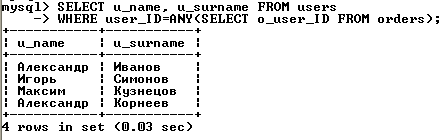
Однако следующий вложенный запрос вернет ошибку:



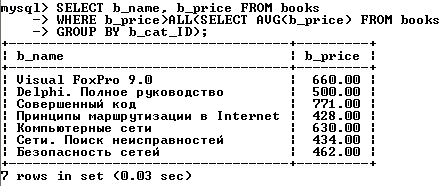
Чтобы выбрать строки из таблицы *catalogs*, у которых первичный ключ совпадает с одним из значений, возвращаемых вложенным запросом, следует воспользоваться конструкцией *IN*:



Ключевое слово *ANY* может применяться с использованием любого оператора сравнения. Используется логика *ИЛИ*, т. е. достаточно, чтобы срабатывало хотя бы одно из многих условий. Запрос вида *WHERE X > ANY (SELECT Y …)* можно интерпретировать как «где *X* больше хотя бы одного выбранного *Y*». Соответственно, запрос вида *WHERE X < ANY (SELECT Y …)* интерпретируется как «где *X* меньше хотя бы одного выбранного *Y*». Рассмотрим запрос, возвращающий имена и фамилии покупателей, совершивших хотя бы одну покупку:

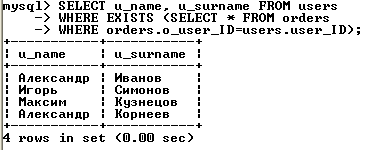


Ключевое слово *ALL* также может применяться с использованием любого оператора сравнения, но при этом используется логика *И*, то есть должны срабатывать все условия. Запрос вида *WHERE X > ALL (SELECT Y …)* интерпретируется как «где *X* больше любого выбранного *Y*». Соответственно, запрос вида *WHERE X < ALL (SELECT Y …)* интерпретируется как «где *X* меньше, чем все выбранные *Y*». Рассмотрим запрос, возвращающий все товарные позиции, цена которых превышает среднюю цену каждого из каталогов:



Результирующая таблица, возвращаемая вложенным запросом, может не содержать ни одной строки. Для проверки этого факта могут использоваться ключевые слова *EXISTS* и *NOT EXISTS*.

Запрос, формирующий список покупателей, совершивших хотя бы одну покупку, можно записать следующим образом:



1. **Выполнение лабораторной работы**

Выполним запрос с объединением двух таблиц. Необходимо получить данные о дате закупок в соответствии с названием отделов, отсортированных по их новизне (рис. 1). Для этого выполним команду

*SELECT d\_name AS ‘Отдел’, p\_date AS’Дата’ FROM Department, Purchase WHERE Purchase.p\_d\_ID = Department.d\_ID ORDER BY p\_date DESC*

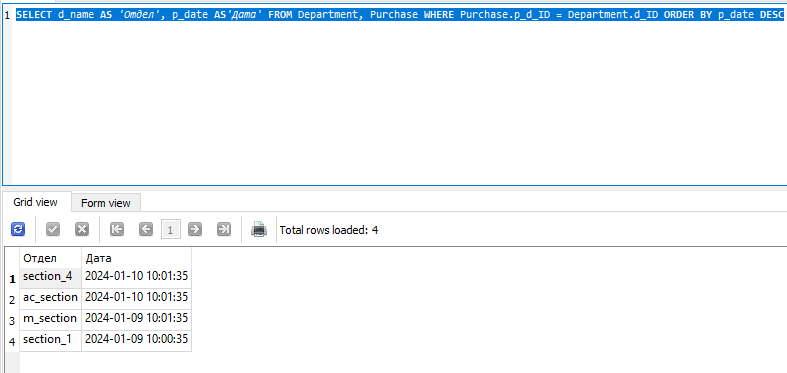


Рисунок 1 – Результат выборки с объединением таблиц

Также выполним запрос с использованием вложенности. Необходимо вывести данные о затратах, которые достигают или превышают минимальный лимит из всех видов затрат, предоставляемых компанией (рис. 2). Для этого выполним команду

*SELECT p\_ID AS 'Номер заказа' FROM Purchase WHERE p\_sum >= (SELECT MIN(exp\_limit) FROM Expenses)*

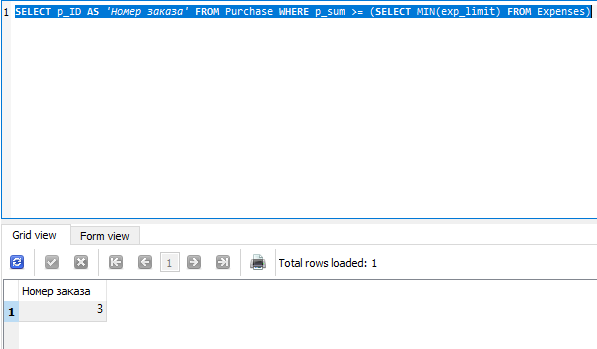


Рисунок 2 – Результат выборки с использованием вложенного запроса

# **Выводы**

В ходе лабораторной работы были произведены сложные запросы на выборки с использованием объединения и вложенных запросов на основе заполненной базы данных «db\_cost\_accounting» в лабораторной работе №3.