#### МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

**БАЗЫ ДАННЫХ И СИСТЕМЫ УПРАВЛЕНИЯ**

**БАЗАМИ ДАННЫХ**

ДОМАШНЯЯ КОНТРОЛЬНАЯ РАБОТА №1

Выполнил Учащийся группы ПО-455

И. А. Дудкин

Шифр 16

2023

**9 Что называют реляционным исчислением? Охарактеризуйте кортежные переменные и правильно построенные формулы.**

Реляционное исчисление — прикладная ветвь формального механизма исчисления предикатов первого порядка. В основе исчисления лежит понятие переменной с определенной для нее областью допустимых значений и понятие правильно построенной формулы, опирающейся на переменные, предикаты и кванторы. Наряду с реляционной алгеброй является способом получения результирующего отношения в реляционной модели данных. В зависимости от того, что является областью определения переменной, различают:

* Исчисление кортежей
* Исчисление доменов

В общем случае все результаты, полученные с помощью операторов реляционного исчисления, могут быть получены с помощью реляционной алгебры. Преимуществом реляционного исчисления перед реляционной алгеброй можно считать то, что пользователю не требуется самому строить алгоритм выполнения запроса, Программа СУБД (при достаточной ее интеллектуальности) сама строит эффективный алгоритм.

Исчисление кортежей - направление реляционного исчисления, где областями определения переменных (операндов) являются отношения базы данных, то есть допустимым значением каждой переменной является кортеж некоторого отношения. В исчислении кортежей, как и в процедурных языках программирования, сначала нужно описать используемые переменные, а затем записать выражения запроса к данным. Описательную часть исчисления можно представить в виде: RANGE OF <переменная> IS <список>. Конструкция RANGE указывает идентификатор переменной кортежа <переменная> и область ее допустимых значений - <список> - последовательность одного или более элементов: x1, ..., xn, каждый из которых является либо отношением, либо выражением над отношением (порядок записи выражений описывается далее). При этом в любой момент <переменная> принимает в качестве значения только один из кортежей <списка> отношений.

Схемы отношений списка должны быть эквивалентными. Область допустимых значений <переменной> образуется путем объединения значений всех элементов списка.

Пример: RANGE OF Студент IS Очный\_студент, Заочный\_студент

Область определения переменной Студент включает в себя все значения из отношения, которое является объединением отношений Очный\_студент и Заочный\_студент.

Выражением реляционного исчисления кортежей называется конструкция вида <целевой\_список > WHERE <WFF>

Значением выражения является отношение, тело (множество кортежей) которого должно удовлетворять WFF (well formulated formula — правильно построенная формула), а схема (набор атрибутов и их имена) определяется целевым списком. Целевой список, по существу, определяет операцию проекции, а формула WFF - селекцию кортежей.

В паре <переменная>.<атрибут> первая составляющая служит для указания переменной кортежа (определенной конструкцией RANGE), а вторая — для определения атрибута отношения, на котором изменяется переменная кортежа. Необязательная часть «AS <атрибут>» используется для переименования атрибута целевого отношения. Если она отсутствует, то имя атрибута целевого отношения наследуется от соответствующего имени атрибута исходного отношения.

Употребление в качестве элемента целевого отношения имени переменной равносильно перечислению в списке всех атрибутов соответствующего отношения.

WFF служат для выражения условий, накладываемых на кортежные переменные. Основой WFF являются простые сравнения, представляющие собой операции сравнения скалярных значений (значений атрибутов переменных или констант). Например, конструкция "СТУДЕНТ.НОМЕР\_ЗАЧЕТНОЙ\_КНИЖКИ = 625432" является простым сравнением. По определению, простое сравнение является WFF, а WFF, заключенная в круглые скобки, является простым сравнением.

Более сложные варианты WFF строятся с помощью логических связок NOT, AND, OR и IF ... THEN. Так, если <формула> - WFF, а <сравнение> - простое сравнение, то NOT <формула> <сравнение> AND <формула> <сравнение> OR <формула> IF <сравнение> THEN <формула> являются WFF.

Допускается построение WFF с помощью кванторов. Если <формула> - это WFF, в которой участвует <переменная>, то конструкции EXISTS <переменная> (<формула>) FORALL <переменная> (<формула>) являются WFF.

В первом случае WFF означает: "Существует по крайней мере одно такое значение <переменной>, что вычисление <формулы> дает значение ИСТИНА".

Во втором случае WFF означает: "Для всех значений переменной <переменной> вычисление <формулы> дает значение ИСТИНА".

Переменные, входящие в WFF, могут быть свободными или связанными. Все переменные, входящие в WFF, при построении которой не использовались кванторы, являются свободными. Фактически, это означает, что если для какого-то набора значений свободных кортежных переменных при вычислении WFF получено значение ИСТИНА, то эти значения кортежных переменных могут входить в результирующее отношение.

Если же имя переменной использовано сразу после квантора при построении WFF вида EXISTS <переменная> (<формула>) или FORALL <переменная> (<формула>), то в этой WFF и во всех WFF, построенных с ее участием, <переменная> — это связанная переменная. Это означает, что такая переменная не видна за пределами минимальной WFF, связавшей эту переменную. При вычислении значения такой WFF используется не одно значение связанной переменной, а вся ее область определения.

Пример. Пусть СОТР1 и СОТР2 - две кортежные переменные, определенные на отношении СОТРУДНИКИ.

СОТРУДНИКИ

СОТР2

|  |  |
| --- | --- |
| ФИО | ЗАРПЛАТА |
| А… | 14000 |
| П… | 9000 |
| В… | 6500 |
| Л… | 12000 |
| Д… | 8300 |

(FOROLL)СОТР1 true

(EXISTS)СОТР1 true

Тогда, WFFEXISTS СОТР2 (СОТР1.ЗАРПЛАТА> СОТР2.ЗАРПЛАТА) для текущего кортежа переменной СОТР1 принимает значение true в том и только в том случае, если во всем отношении СОТРУДНИКИ найдется кортеж (связанный с переменной СОТР2) такой, что значение его атрибута ЗАРПЛАТА удовлетворяет внутреннему условию сравнения.

WFF FORALL СОТР2 (СОТР1.ЗАРПЛАТА> СОТР2.ЗАРПЛАТА) для текущего кортежа переменной СОТР1 принимает значение true в том и только в том случае, если для всех кортежей отношения СОТРУДНИКИ (связанных с переменной СОТР2) значения атрибута СОТР\_ЗАРП удовлетворяют условию сравнения.

Описанное исчисление не обладает вычислительной полнотой, так как не позволяет выполнять вычисления. Добавление вычислительных функций в исчисление можно реализовать путем расширения определения операндов сравнения и элементов целевого списка таким образом, чтобы они допускали использование скалярных выражений с литералами, ссылками на атрибуты и итоговыми функциями.

качестве итоговых могут выступать следующие функции: COUNT (количество), SUMM (сумма), АVG (среднее), МАХ (максимальное), MIN (минимальное).

Для данных элементов целесообразно использовать спецификацию вида "AS <имя атрибута>, где можно явно задать имя результирующему атрибуту.

Пример. Определить студента с максимальным рейтингом Студент.ФИО, MAX(Рейтинг) AS Максимальный\_Рейтинг WHERE Студент.Номер\_зачетной\_книжки=Рейтинг.Номер\_ зачетной\_книжки \_

В исчислении доменов областью определения переменных являются не отношения, а домены. Применительно к базе данных Рейтинг студентов можно говорить, например, о доменных переменных ИМЯ (значения - допустимые имена) или Номер\_зачетной\_книжки (значения - допустимые номера зачетных книжек студентов).

Основным формальным отличием исчисления доменов от исчисления кортежей является наличие дополнительного набора предикатов, позволяющих выражать так называемые условия членства. Если R - это n-арное отношение с атрибутами a1, a2, ..., an, то условие членства имеет вид R(a1i:v1i...aim:vim)(m<=n) где vij - это либо литерально задаваемая константа, либо имя кортежной переменной. Условие членства принимает значение true в том и только в том случае, если в отношении R существует кортеж, содержащий указанные значения указанных атрибутов.

Если vij - константа, то на атрибут aij задается жесткое условие, не зависящее от текущих значений доменных переменных; если же - имя доменной переменной, то условие членства может принимать разные значения при разных значениях этой переменной.

Во всем остальном формулы и выражения исчисления доменов выглядят похожими на формулы и выражения исчисления кортежей. В частности, конечно, различаются свободные и связанные вхождения доменных переменных.

Для примера сформулируем с использованием исчисления доменов запрос "Выдать номера и имена студентов сотрудников, не получающих минимальную заработную плату" (будем считать для простоты, что мы определили доменные переменные, имена которых совпадают с именами атрибутов отношения СОТРУДНИКИ, а в случае, когда требуется несколько доменных переменных, определенных на одном домене, мы будем добавлять в конце имени цифры): СОТР\_НОМ, СОТР\_ИМЯ WHERE EXISTS СОТР\_ЗАРП1 (СОТРУДНИКИ (СОТР\_ЗАРП1) AND СОТРУДНИКИ (СОТР\_НОМ, СОТР\_ИМЯ, СОТР\_ЗАРП) AND СОТР\_ЗАРП > СОТР\_ЗАРП1)

**41 Найдите преподавателей, которые родились в 70-е годы и имеют домашний телефон.**

Заполним таблицу преподавателей тестовыми данными и выведем её на экран. В нашей таблице есть три преподавателя в 70 годах рождения и только у одного из них есть домашний телефон. Содержимое таблицы представлено на рисунке 1.

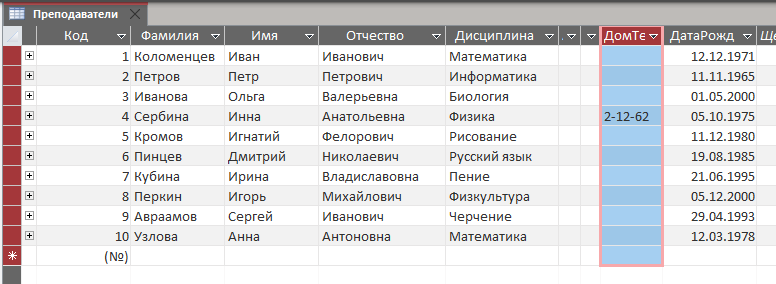


Рисунок 1. Содержимое таблицы Преподаватели.

Создадим простой запрос, этапы создания запроса представлены на рисунках 2-4.

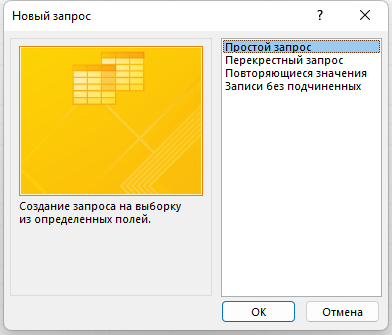


Рисунок 2. Выбор типа запроса.

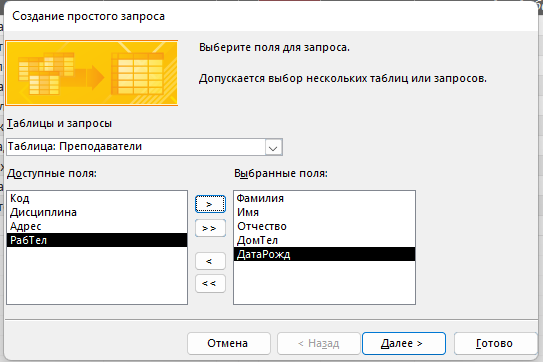


Рисунок 3. Выбор полей для запроса

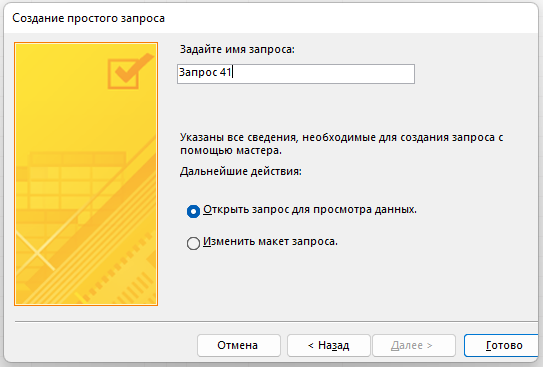


Рисунок 4. Задание имени запроса.

Откроем запрос в режиме конструктора и наложим ограничение на интересующие нас поля – даты рождения и наличие домашнего телефона. Запрос в режиме конструктора представлен на рисунке 5.

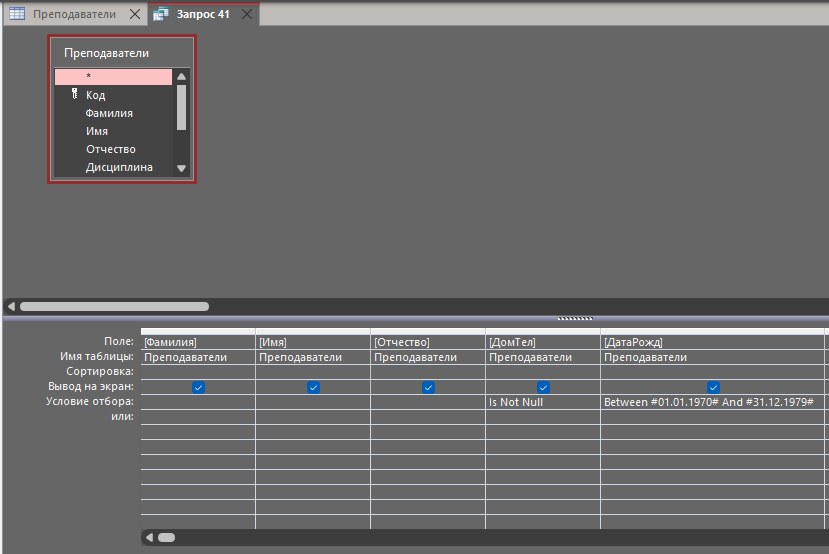


Рисунок 5. Запрос в режиме конструктора.

Запустим запрос на выполнение и сравним полученные результаты с ожидаемыми, результат выполнения запроса представлен на рисунке 6.

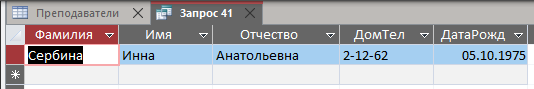


Рисунок 6. Результат выполнения запроса.

**83 Найдите номера комнат, в которых живет четыре девушки.**

Заполним таблицу тестовыми данными, содержимое таблицы представлено на рисунке 7.

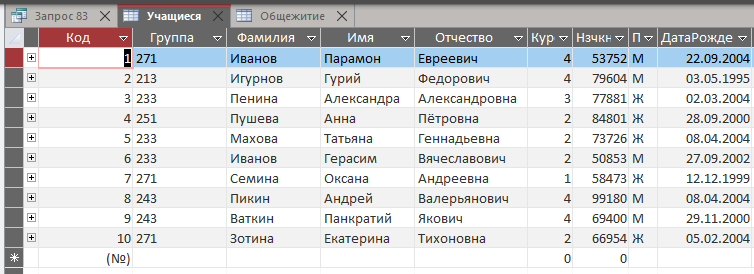


Рисунок 7. Содержимое таблицы Учащиеся.

Создадим простой запрос, который потом будем использовать как базу, этапы создания запроса представлены на рисунках 8-11

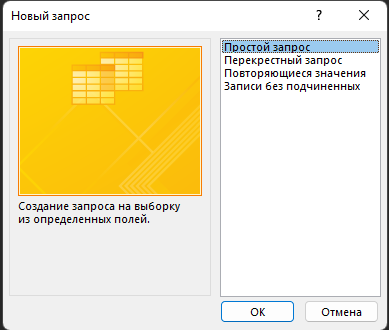


Рисунок 8. Выбор типа запроса.

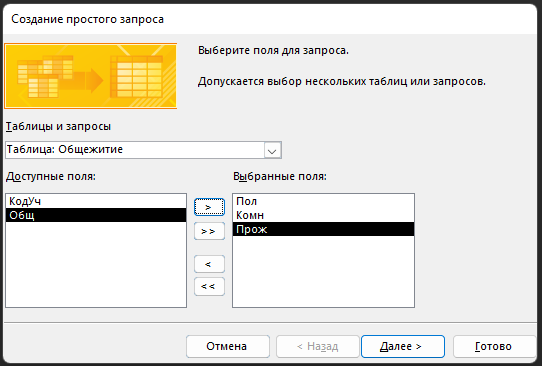


Рисунок 9. Выбор полей для запроса.

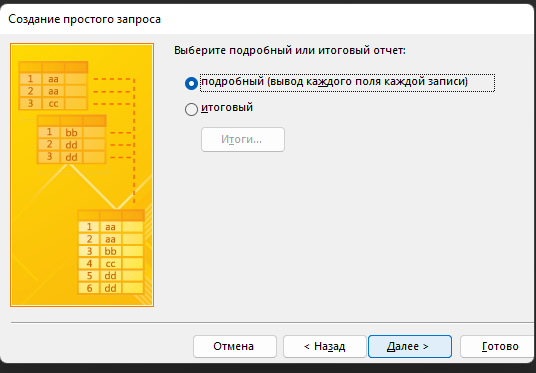


Рисунок 10. Выбор типа отчета.

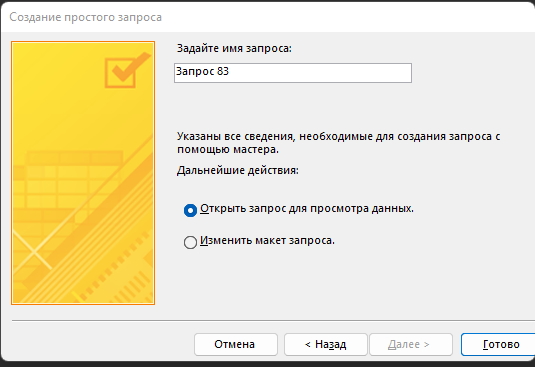


Рисунок 11. Указание имени запроса.

Запустим запрос на выполнение, результаты представлены на рисунке 12.

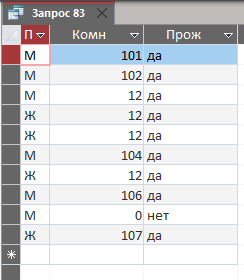


Рисунок 12. Информация об учащихся в общежитии.

Переключимся в режим конструктора и отфильтруем записи проживающих, выберем только проживающих женского пола. Конструктор запроса представлен на рисунке 13.

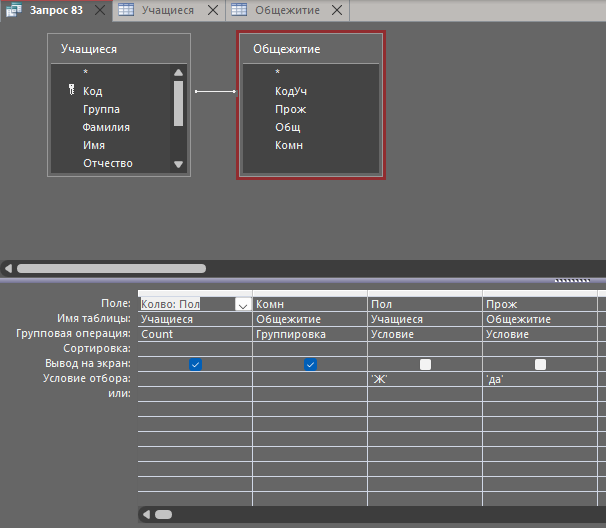


Рисунок 13. Фильтрация базового запроса.

Переключимся в режим SQL и подсчитаем количество проживающих покомнатно. Отредактированный запрос представлен на рисунке 14.

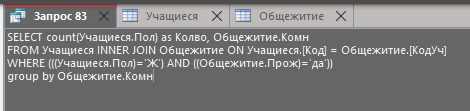


Рисунок 14. SQL запрос.

Результат выполнения запроса – количество проживающих женщин в каждой комнате представлен на рисунке 15.

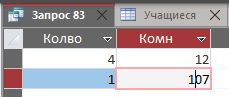


Рисунок 15. Результат итогового запроса.

**92 Найдите десять лучших (имеющих максимальную сумму баллов) учащихся курса. Номер курса - параметр запроса.**

Модифицируем таблицу учащихся для этого задания, разделим их на 2 курса. Таблица для выполнения задания представлена на рисунке 16.

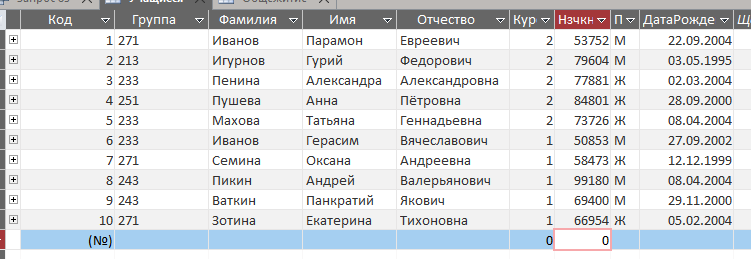


Рисунок 16. Содержимое таблицы Учащиеся.

Создадим запрос при помощи мастера запросов. Этапы создания запроса представлены на рисунках 17-20.

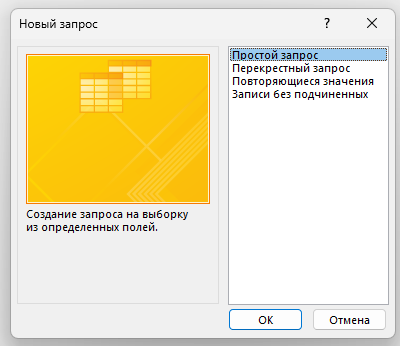


Рисунок 17. Задание типа запроса.

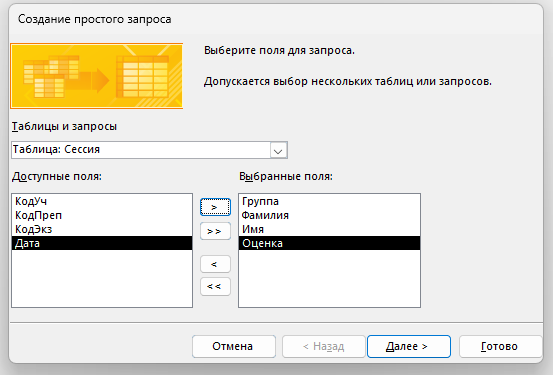


Рисунок 18. Выбор полей запроса.

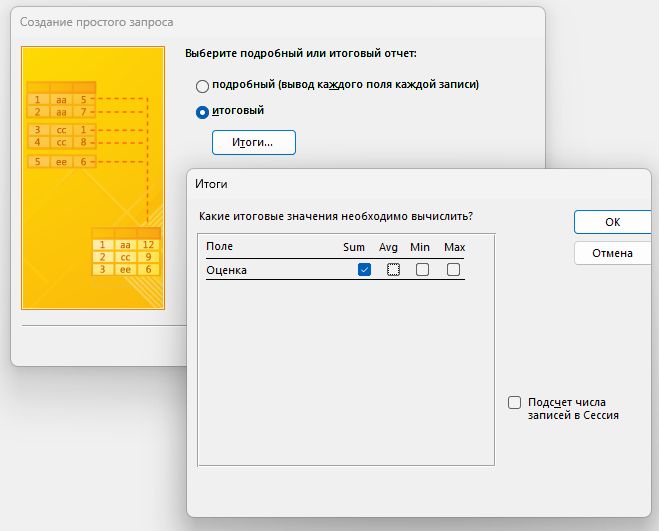


Рисунок 19. Выбор типа итогов запроса.

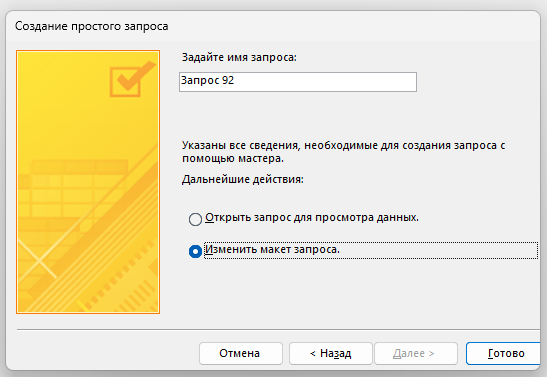


Рисунок 20. Завершение работы мастера.

Откроем запрос в режиме конструктора и модифицируем его, зададим сортировку по баллам, и параметр запроса - Курс. Вид запроса представлен на рисунке 21.

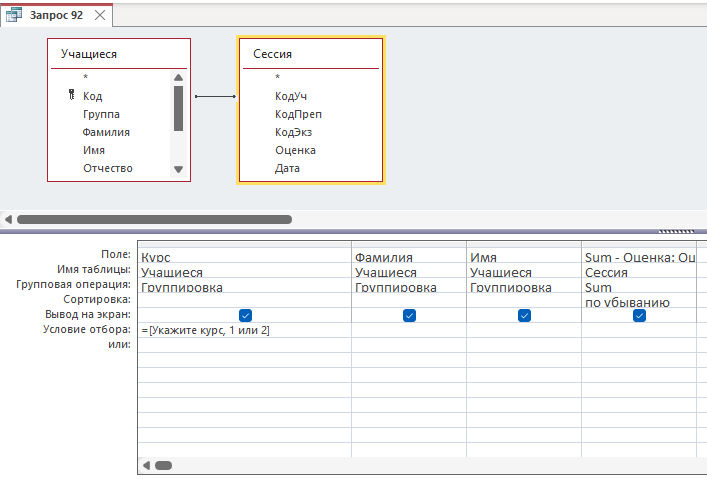


Рисунок 21. Модифицированный запрос.

Откроем запрос в режиме SQL и зададим выбор необходимого количества записей командой TOP после которой укажем необходимое количество записей, у нас это 2. Вид запроса представлен на рисунке 22.

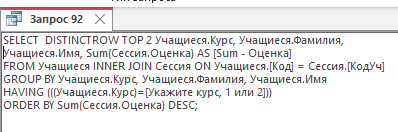


Рисунок 22. SQL запрос.

Запустим запрос на выполнение. Окно ввода параметра представлено на рисунке 23.

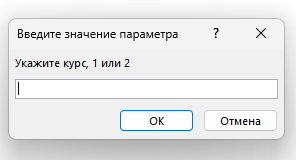


Рисунок 23. Запрос параметра.

Результат запроса с указанием TOP 2 представлен на рисунке 24.

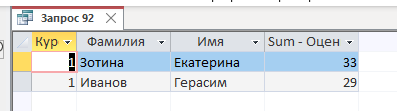


Рисунок 24. Результат запроса TOP 2.

Результат запроса с указанием TOP 10 представлен на рисунке 25.

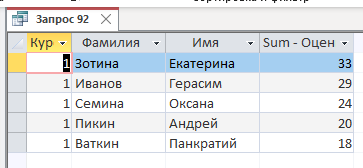


Рисунок 25. Результат запроса TOP 10.

**139 Постройте запрос на создание таблицы «Список отличников», содержащей фамилии и номера групп учащихся, сдавших сессию на 9 и 10.**

Создадим запрос в режиме мастера. Этапы создания запроса представлены на рисунках 26-29.

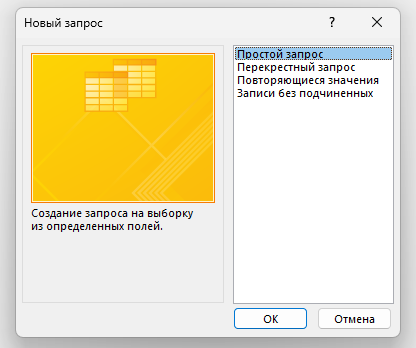


Рисунок 26. Выбор типа запроса.

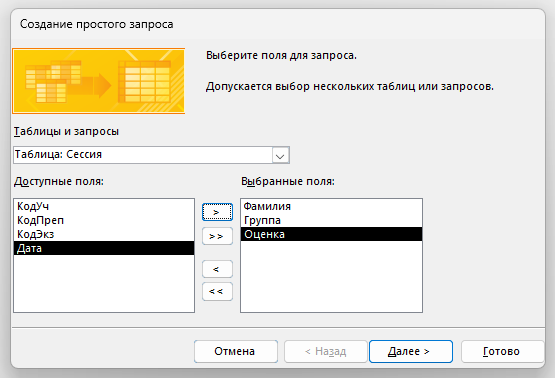


Рисунок 27. Выбор полей запроса.

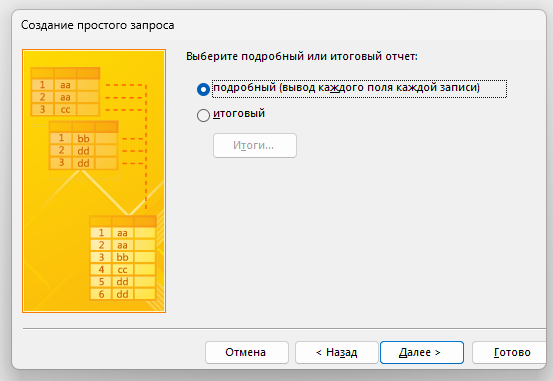


Рисунок 28. Выбор типа отчета.

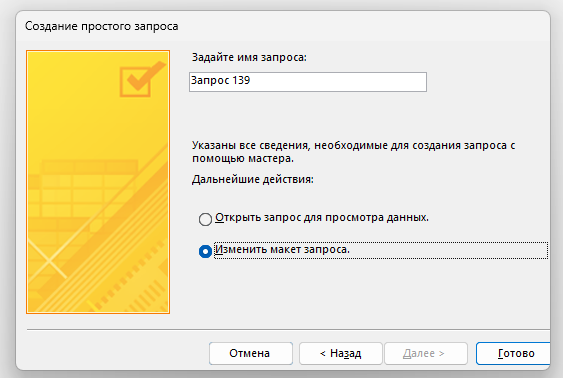


Рисунок 29. Завершение работы мастера.

При завершении работы мастера сразу укажем открыть запрос в режиме макета и укажем ограничение на оценки и вывод необходимых по условию задания полей. Вид макета представлен на рисунке 30.

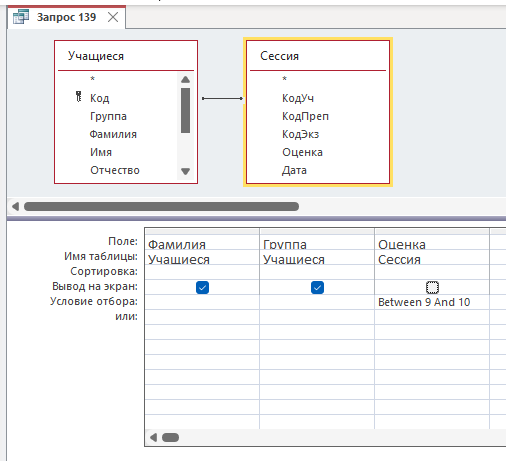


Рисунок 30. Макет запроса.

Переключимся в режим SQL и удалим дубликаты записей командой DISTICNT. Вид в режиме SQL представлен на рисунке 31.

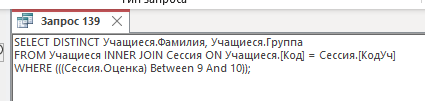


Рисунок 31. Вид в режиме SQL.

В режиме конструктора укажем создание таблицы как результат выполнения запроса. Настройка названия таблицы представлена на рисунке 32.

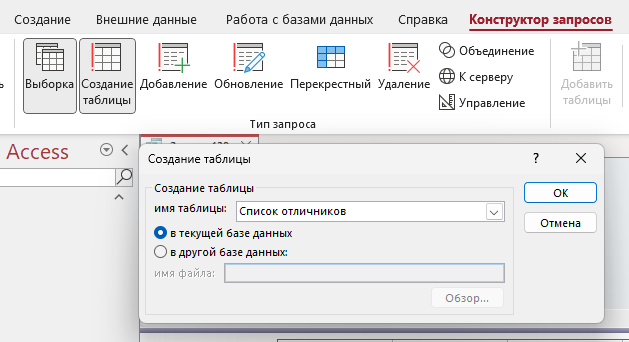


Рисунок 32. Задание имени таблицы для запроса.

Запустим запрос на выполнение. Предупреждение базы данных при запуске представлено на рисунке 33.

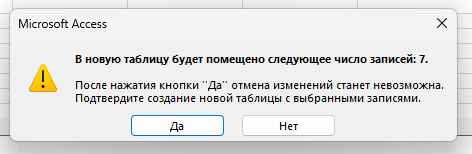


Рисунок 33. Предупреждение базы данных при запуске.

Созданная таблица и её содержимое представлено на рисунке 34.

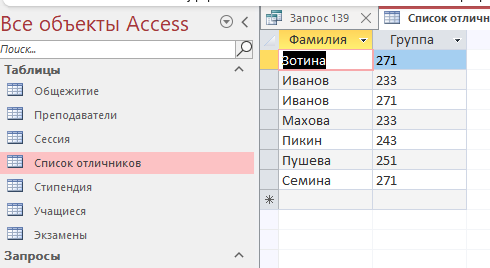


Рисунок 34. Созданная запросом таблица с содержимым.

**154 На базе таблицы «Учащиеся» и «Общежитие» постройте табличную форму «Девушки» с полями «ФИО», «Курс», «Группа», «Адрес» и «Дата рождения». Поле «ФИО» должно содержать фамилию и имя учащихся; а поле «Адрес» - сведения об общежитии и комнате. Список должен быть отсортирован по группам, а затем по фамилиям**

Сначала создадим простой запрос, этапы создания представлены на рисунках 35-38.

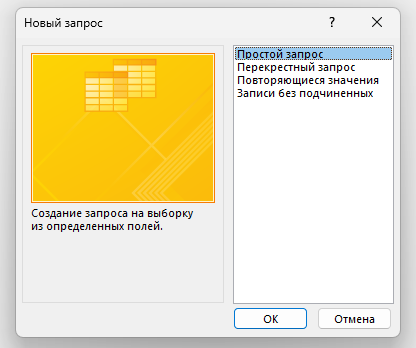


Рисунок 35. Выбор типа запроса.

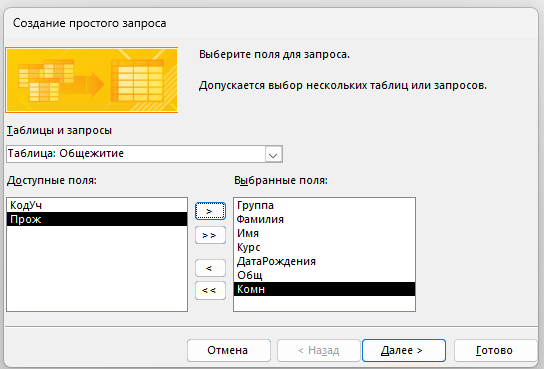


Рисунок 36. Указание таблиц и полей.

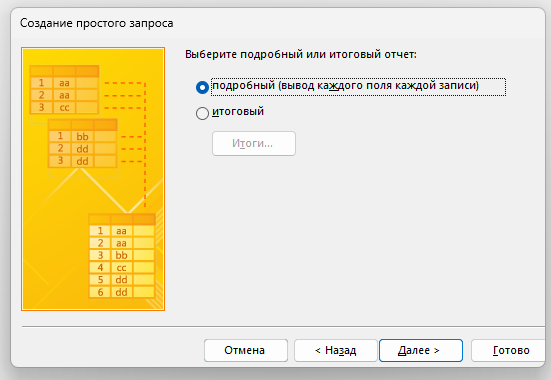


Рисунок 37. Выбор типа отчета.

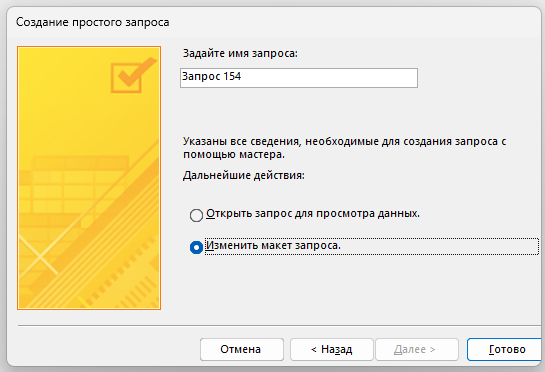


Рисунок 38. Завершение работы мастера.

Вид запроса в режиме конструктора представлен на рисунке 39.

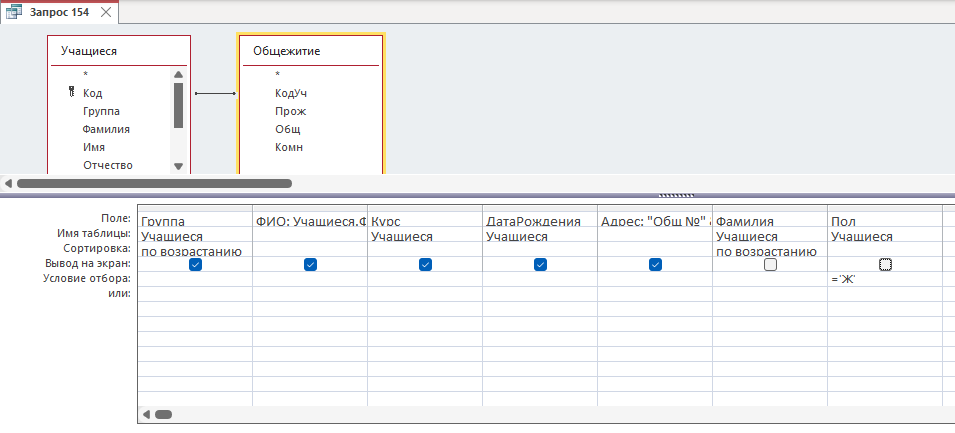


Рисунок 39. Запрос в режиме конструктора.

Переключимся в режим SQL и конкатенируем необходимые по условию поля, а также укажем поля для сортировки данных. Вид запроса в режиме SQL представлен на рисунке 40.

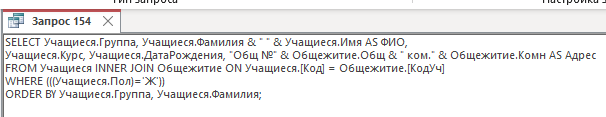


Рисунок 40. Вид запроса в режиме SQL.

Запустим запрос, результат работы запроса представлен на рисунке 41.

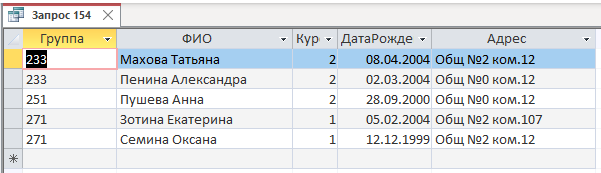


Рисунок 41. Результат работы запроса.

На базе этого запроса создадим форму при помощи мастера. Этапы создания формы представлены на рисунках 42-

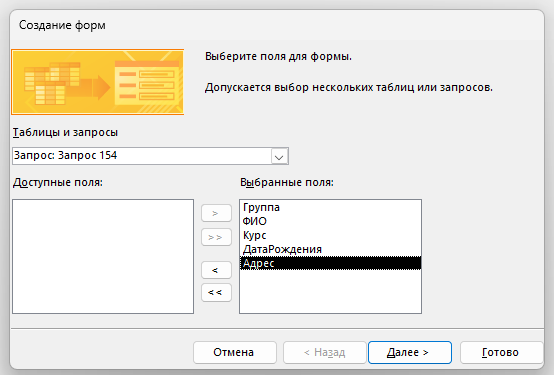


Рисунок 42. Выбор источника и полей для формы.

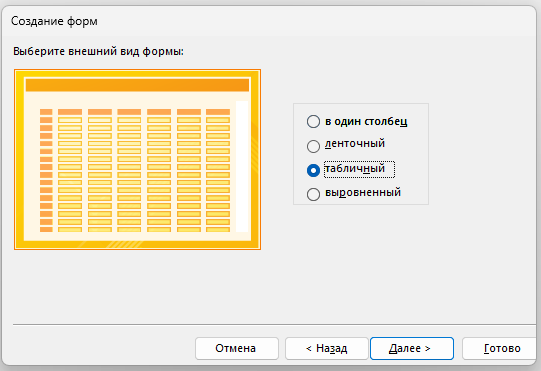


Рисунок 43. Выбор типа формы.

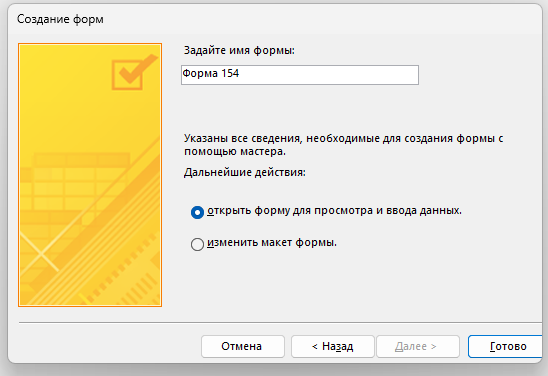


Рисунок 44. Задание имени форме.

Откроем форму. Данные формы представлены на рисунке 45.

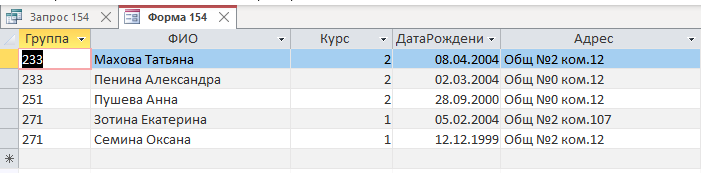


Рисунок 45. Данные готовой формы.

**187 Постройте отчет «Список девушек по группам», включающий поля «№ зачкн.», «ФИО», «Общежитие» и «Комната». Он должен быть сгруппирован по учебным группам, а список каждой группы отсортирован по фамилиям. Номер учебной группы вынести в заголовок группы. После каждой группы указать общее число** **девушек в группе, а в конце списка - общее число девушек на курсе**

Создадим запрос для отчета. Этапы создания запроса представлены на рисунках 46 – 49.

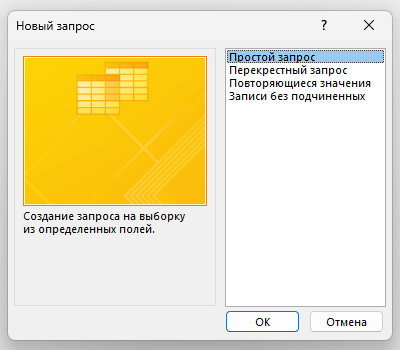


Рисунок 46. Выбор типа запроса.

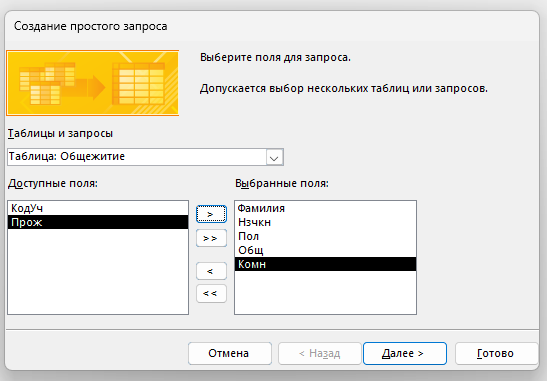


Рисунок 47. Выбор таблиц и полей

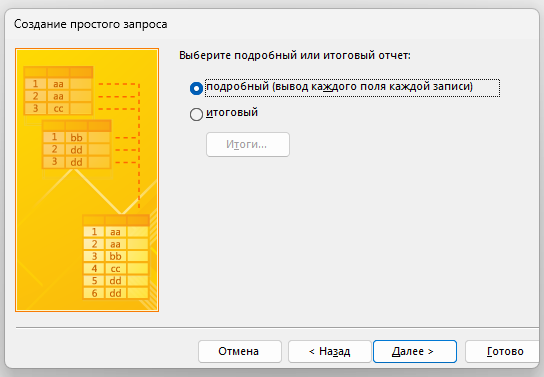


Рисунок 48. Указание вида отчета.

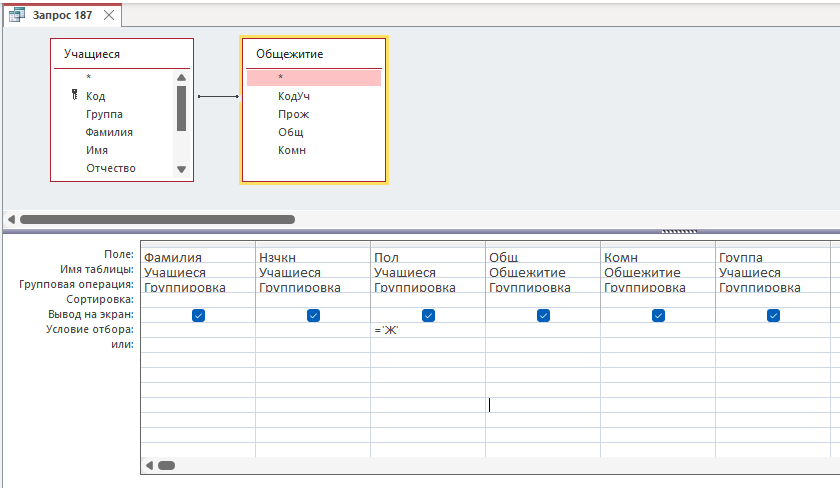


Рисунок 49. Запрос в режиме конструктора с указанием пола учащегося.

Переключимся в режим SQL и модифицируем его для отображения ФИО одним полем и задания группировки и сортировки. Вид запроса в режиме SQL представлен на рисунке 50.

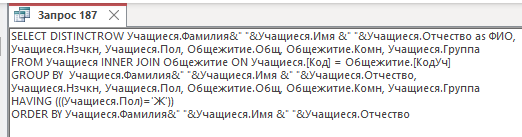


Рисунок 50. Вид запроса в режиме SQL.

Просмотрим результат выполнения, он представлен на рисунке 51.

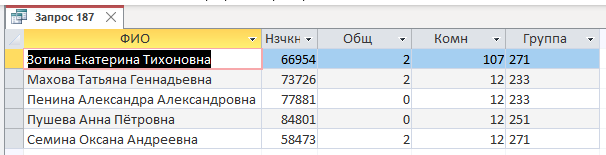


Рисунок 51. Результат выполнения запроса.

Этап предварительного создания отчета представлен на рисунках 52 – 57.

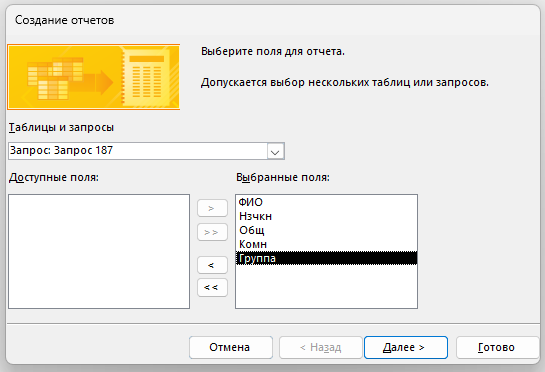


Рисунок 52. Выбор полей для отчета.

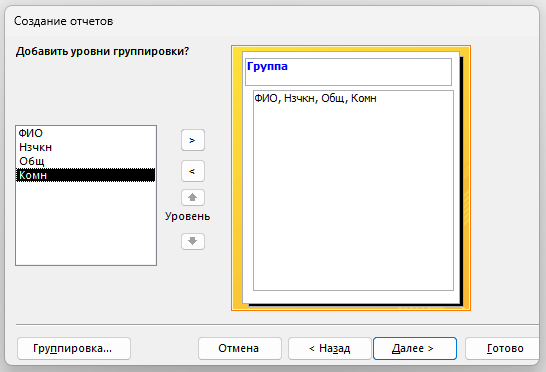


Рисунок 53. Указание группировки.

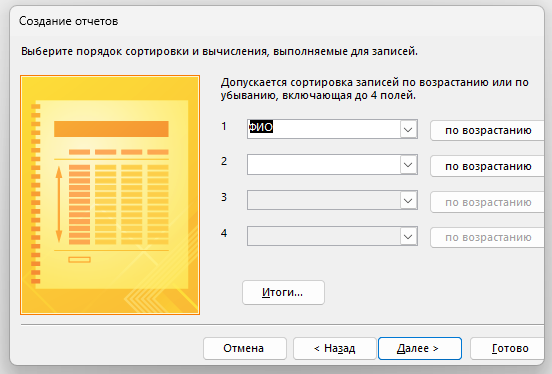


Рисунок 54. Указание сортировки и итогов.

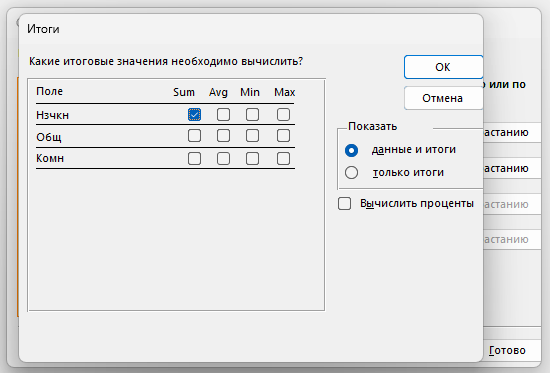


Рисунок 55. Указание итогов, сумма.

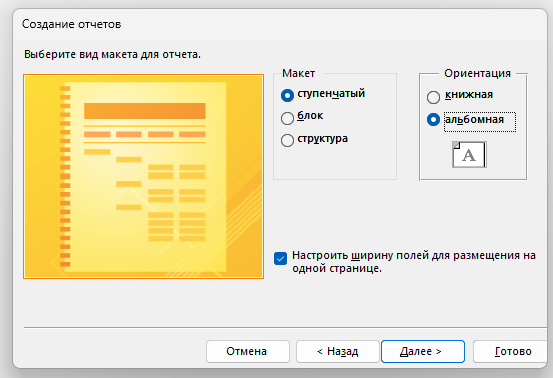


Рисунок 56. Выбор макета отчета.

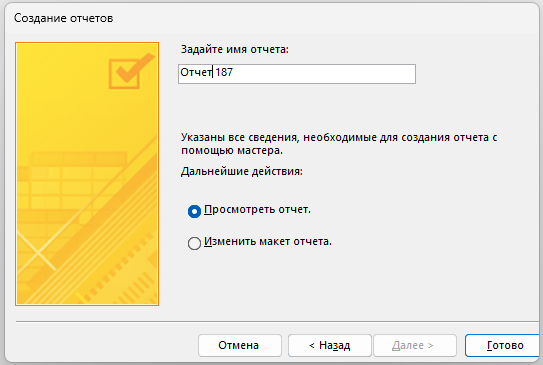


Рисунок 57. Завершение работы мастера отчетов.

Результирующий отчет представлен на рисунке 58.

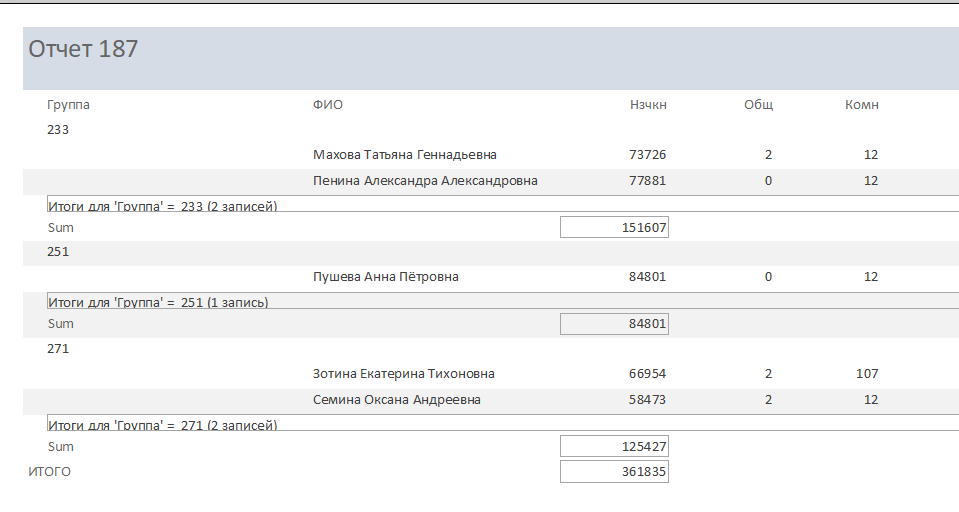


Рисунок 58. Предварительный просмотр отчета.

Так как в итогах мы указали суммирование по полю номера зачетной книжки, то необходимо изменить итоговую функцию SUM на COUNT. Сумма была необходима для того чтобы мастер красиво расставил поля в макете. Откроем макет, рисунок 59, и изменим итоговую функцию.

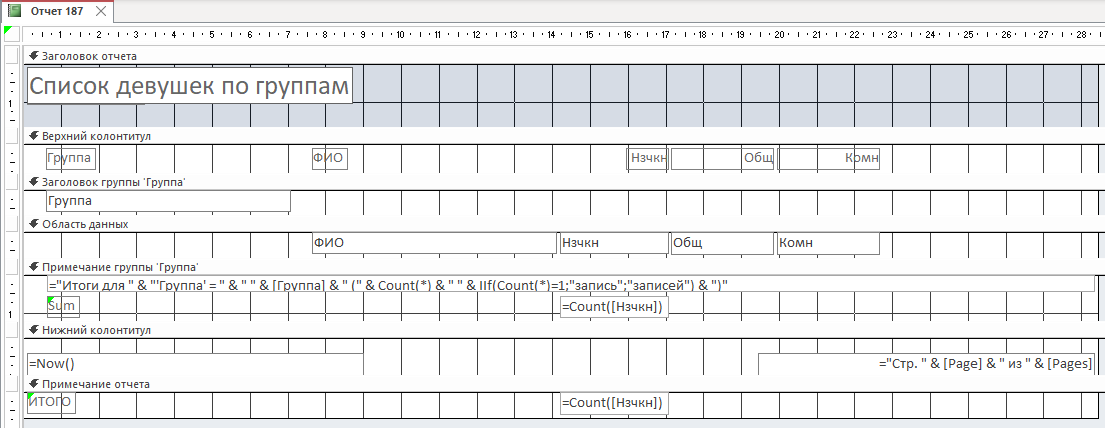


Рисунок 59. Изменение итоговой функции в макете.

Внешний вид результирующего отчета представлен на рисунке 60.

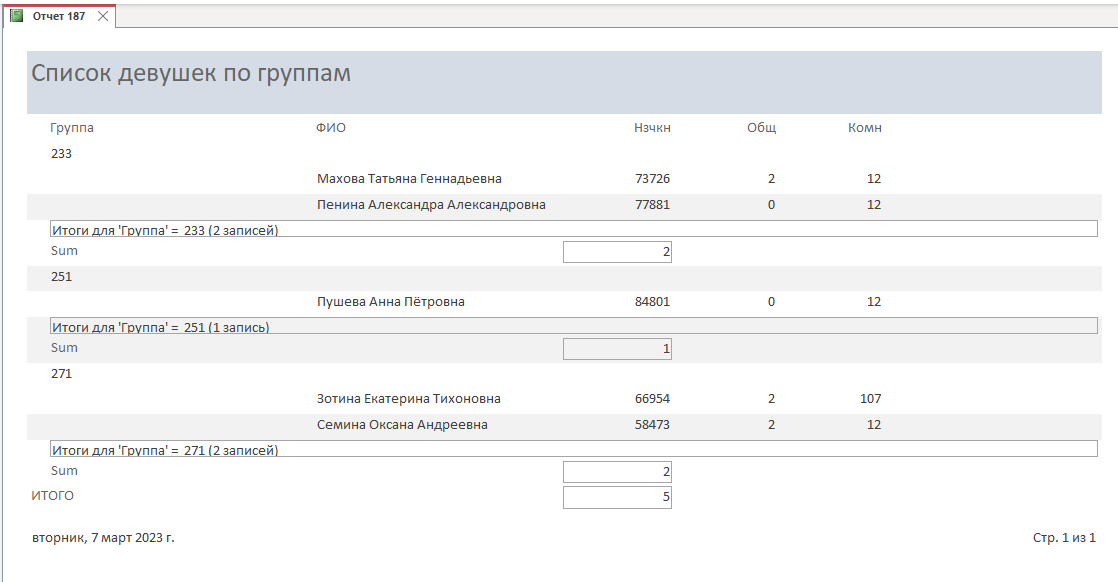


Рисунок 60. Отчет «Список девушек по группам».

**Список использованных источников**

1 Бен-Ган, И. Microsoft SQL Server 2012. Основы T-SQL / И. Бен-Ган. М. : Эксмо, 2015. 400 с.

2 Дейт, К.Дж. Введение в системы баз данных / К.Дж. Дейт. М. : Вильямс, 2018. 1382 с.

3 Лазицкас, Е.А. Базы данных и системы управления базами данных / Е.А. Лазицкас, И.Н. Загуменникова, П.Г. Гилевский. Минск : РИПО, 2016. 268 с.

4 Федорова, Г. Разработка и администрирование баз данных / Г. Федорова. М. : Академия, 2015. 313 с.

5 SQL справочник / К. Кляйн [и др.]. СПб. : Символ-плюс, 2016. 56 с.