МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт информационных технологий |
| наименование института (факультета) |
| Кафедра математического и программного обеспечения ЭВМ |
| наименование кафедры  Проектирование баз данных |

наименование дисциплины в соответствии с учебным планом

ЛАБОРАТОРНАЯ РАБОТА №1

«Управление БД Oracle»

|  |  |
| --- | --- |
| Исполнитель |  |
| студент | 1ПИб-02-3оп-22 |
|  | группа |
|  | Маркелов С. А. |
|  | Фамилия, имя, отчество |
| Руководитель | Селяничев О. Л. |
|  | Ф.И.О. преподавателя |
| Оценка |  |
| Подпись |  |

2025 год

Задания выполнить средствами СУБД Oracle.

1. Интерфейс СУБД Oracle

1. Создайте таблицу базы данных с полями, содержание которых – фамилия, год рождения, оценки по математике, информатике, иностранному языку.

В панели вкладок в верхней части экрана выбираем вкладку «SQL Workshop». Далее переходим в раздел «Object Browser» (рис. 1).

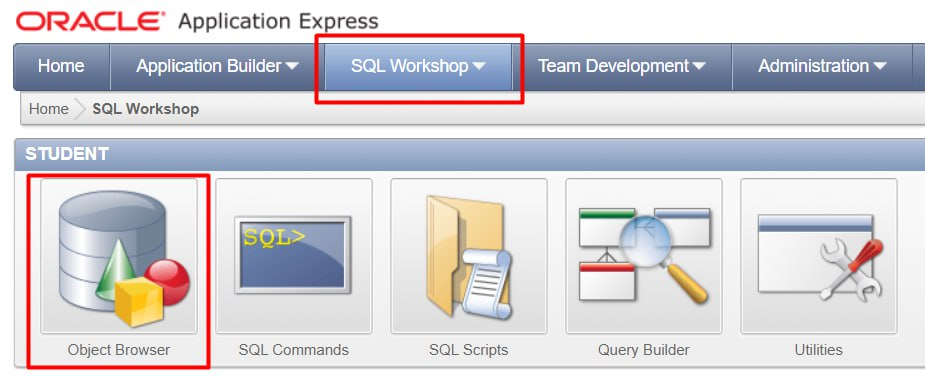


Рис. 1. Открытие Object Browser

В разделе Object Browser в левой части экрана расположена панель со списком существующих таблиц (рис. 2). Нам необходимо создать новую. Для этого нажимаем на кнопку «Create» («Создать») – появляется выпадающий список, предлагающий выбрать, что именно мы хотим создать (рис. 3). Выбираем «Table» - таблицу.

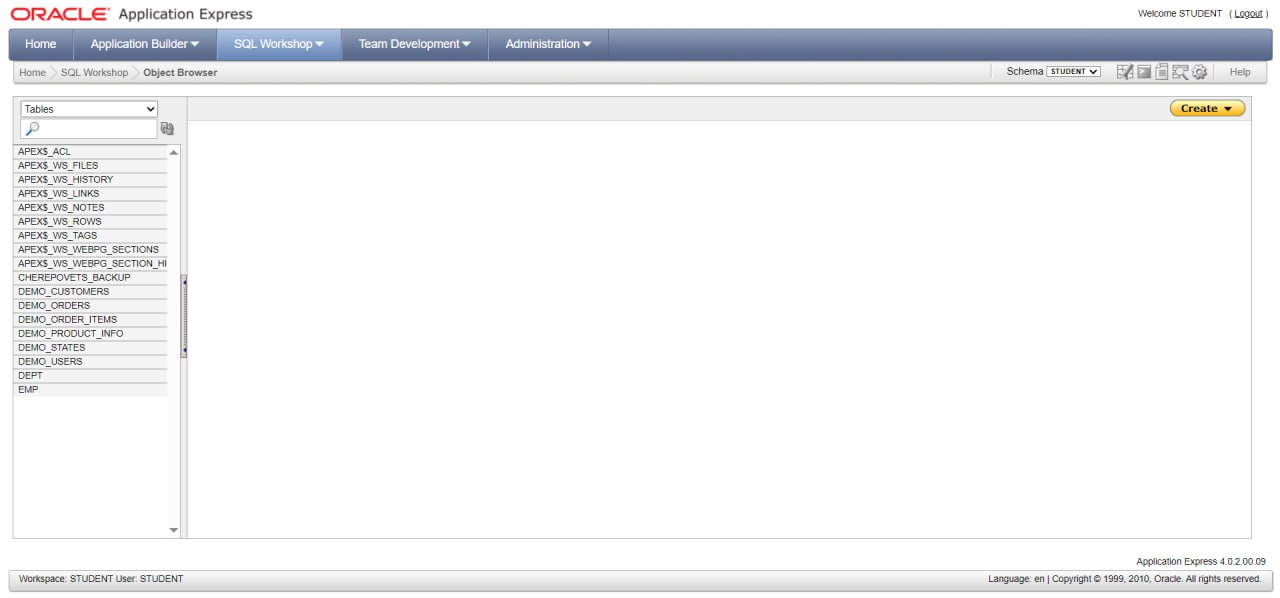


Рис. 2. Object Browser

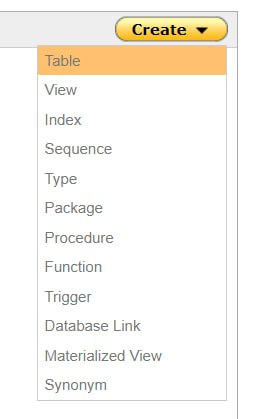


Рис. 3. Кнопка «Create» и выпадающий список

После этого программа перейдет к инструменту создания таблицы «Create table». На шаге «Colums» (дословно переводится как «Столбцы» или «Колонки», хотя корректным названием будет «Поля») программа запросит пользователя ввести имя новой таблицы, названия полей, типы данных и, при необходимости, некоторые другие данные (рис. 4).

В поле «Table name» («Имя таблицы») вводим «PI\_43». Далее заполняем названия полей («Column Name») и выбираем типы данных («Type»):

* «Фамилия» – VARCHAR2, 20 (фамилия студента, строковый тип, длина строки – 20);
* «Дата» – DATE (дата рождения, тип для даты и времени);
* «Мат» – NUMBER (оценка по математике, числовой тип);
* «Инф» – NUMBER (оценка по информатике, числовой тип);
* «ИнЯз» – NUMBER (оценка по иностранному языку, числовой тип).

После заполнения всех полей нажимаем на кнопку «Next >» («Следующий >») для перехода к следующему шагу.

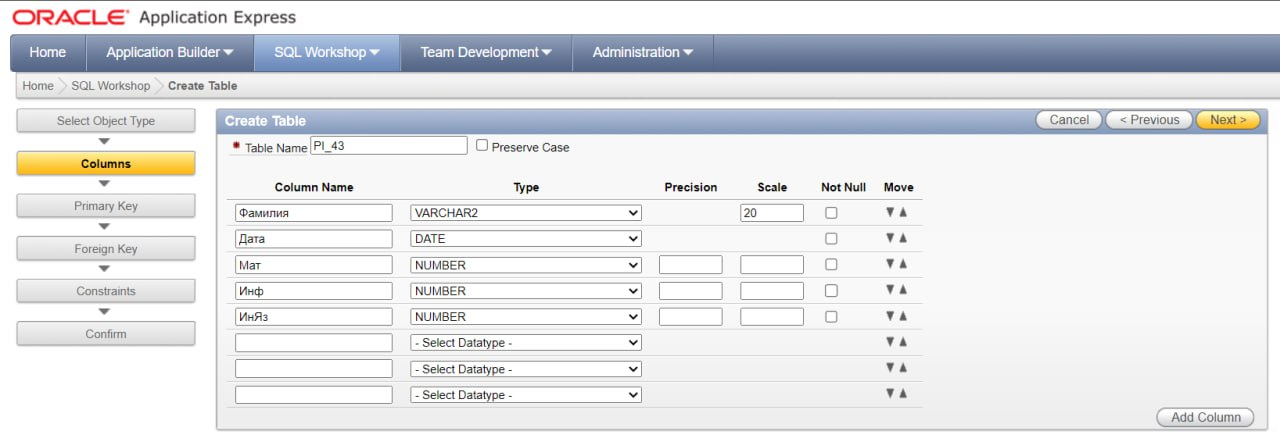


Рис. 4. Инструмент «Create table»: шаг «Columns»

Далее идёт шаг «Primary Key» («Первичный ключ»), на котором задаётся первичный ключ таблицы: выбор поля (или полей), которое будет уникальным идентификатором строки (рис. 5).

По заданию лабораторной работы требуется создание лишь одной таблицы, а значит необходимости в создании первичного ключа нет. По умолчанию на данном шаге выбран пункт «No Primary Key» («Нет первичного ключа») – здесь ничего не меняем, сразу нажимаем на кнопку «Next >».

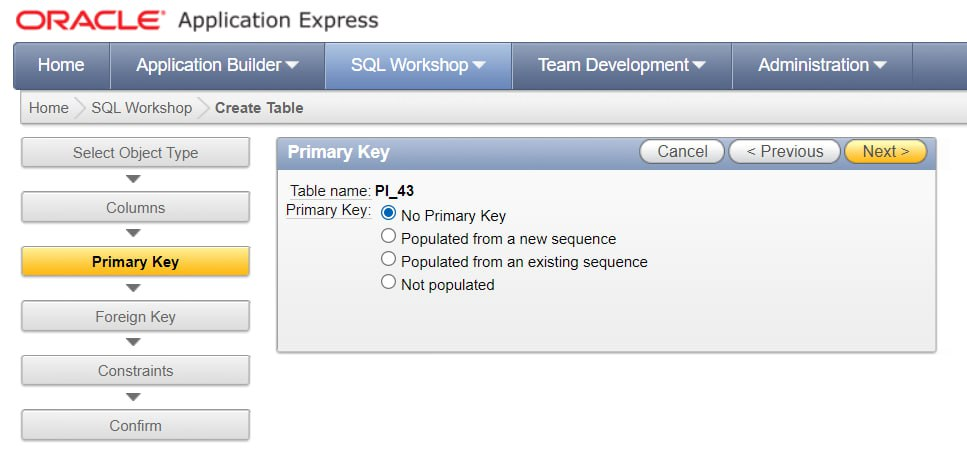


Рис. 5. Инструмент «Create table»: шаг «Primary Key»

Затем идёт шаг «Foreign Key» («Внеший ключ»), на котором задаются внешние ключи: сопоставляются поля текущей таблицы с первичными ключами других таблиц (рис. 6).

Как уже было сказано выше, другие таблицы по заданию лабораторной работы создавать не требуется. Следовательно, данный шаг также необходимо пропустить, не меняя никаких значений, а сразу нажав на кнопку «Next >».

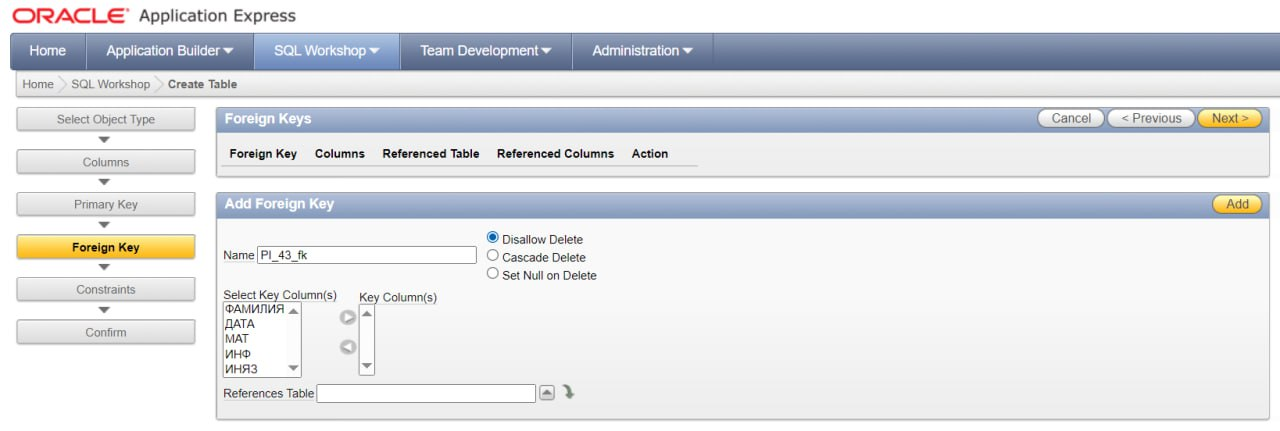


Рис. 6. Инструмент «Create table»: шаг «Foreign Key»

Следом идёт шаг «Constraints» («Ограничения»), на котором настраиваются правила для данных (например, NOT NULL, UNIQUE, CHECK и т. д.) (рис. 7).

По заданию лабораторной работы задавать ограничения также не требуется, поэтому в появившемся поле вводить ничего не нужно, необходимо сразу нажать на кнопку «Next >»



Рис. 7. Инструмент «Create table»: шаг «Constraints»

После этого идёт финальный шаг «Confirm» («Подтвердить»), в котором выводятся все выбранные на предыдущих шагах параметры в виде SQL-запроса, который будет выполнен для создания таблицы (рис. 8). Шаг необходим для проверки – если пользователь заметит ошибку, он может вернуться на один из предыдущих шагов с помощью кнопки «< Previous» («< Предыдущий») и внести какие-либо изменения. Если же всё верно, необходимо нажать на кнопку «Create» («Создать»), после чего таблица будет создана.

SQL-запрос имеет вид:

CREATE table "PI\_43" (

"ФАМИЛИЯ" VARCHAR2(20),

"ДАТА" DATE,

"МАТ" NUMBER,

"ИНФ" NUMBER,

"ИНЯЗ" NUMBER

)

/



Рис. 8. Инструмент «Create table»: шаг «Confirm»

После создания таблица откроется автоматически. В Object Browser для таблицы имеются различные вкладки, позволяющие работать с ней. Изначально открылась вкладка «Table» («Таблица»), в которой содержится информация о структуре таблицы (её полях и их типах данных) и кнопки для работы с этой структурой (рис. 9).

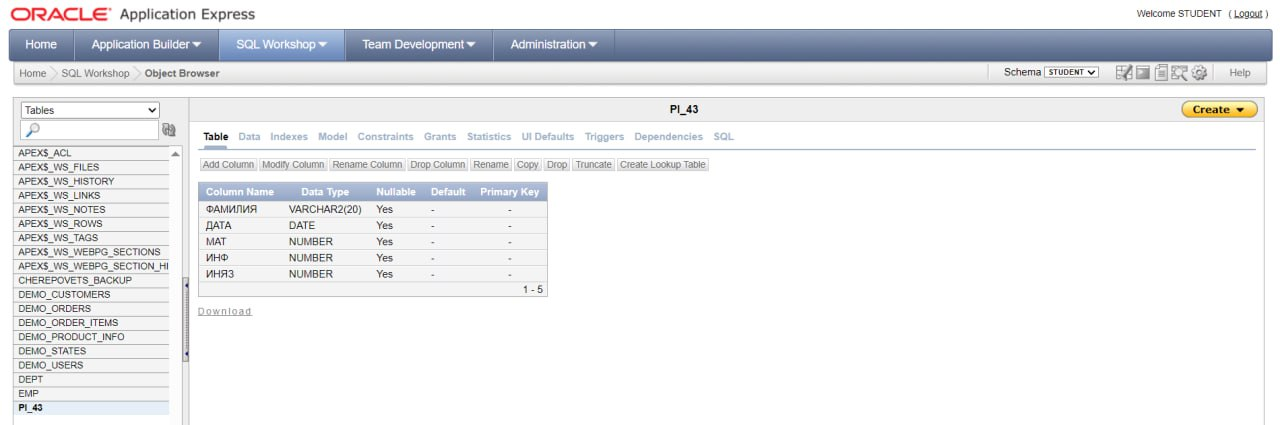


Рис. 9. Просмотр таблицы PI\_43: вкладка «Table»

2. Внесите в таблицу 5 записей.

Для работы с данными нам необходимо перейти во вкладку «Data» («Данные»). Изначально данная вкладка будет содержать всего лишь 3 кнопки (рис. 10):

* «Query» («Запрос») – выполнить запрос;
* «Count Rows» (дословно – «Количество строк», корректнее – «Количество записей») – посчитать количество записей;
* «Insert Row» (дословно – «Вставить строку», корректнее – «Добавить запись») – добавить новую запись.

После добавления записей они будут отображаться под этими кнопками. А пока вместо них отображается надпись «This table contains no data» – «Эта таблица не содержит данных».

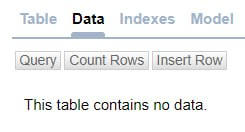


Рис. 10. Просмотр данных таблицы PI\_43: вкладка «Data» до добавления записей

Для добавления записи необходимо нажать на кнопку «Insert Row». Откроется форма для заполнения значений полей записи (рис. 11). После её заполнения необходимо нажать на кнопку «Create» («Создать»). После нажатия на эту кнопку создастся запись, а пользователь вернется к просмотру таблицы. Если нужно добавить сразу несколько полей, можно нажать на другую кнопку – «Create and Create Another» («Создать и создать другое»). В таком случае после создания записи мы останемся в этой же форме и сможем создавать новые записи без лишних нажатий кнопок.

Создадим первую запись, и добавим себя:

* «Фамилия» – Маркелов;
* «Дата» – 07.26.2004 – в американском формате «ММ.ДД.ГГГГ»;
* «Мат» – 5;
* «Инф» – 5;
* «ИнЯз» – 5.

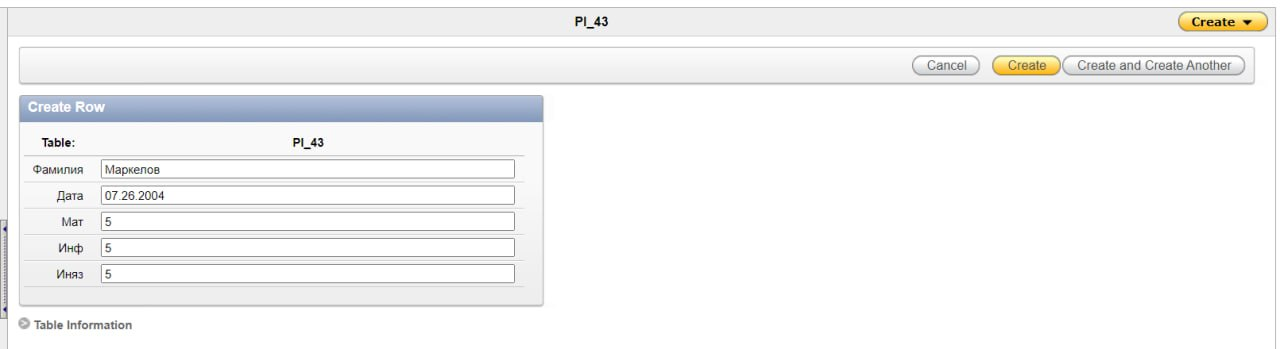


Рис. 11. Создание записи

Аналогичным образом создаем ещё 4 записи, после чего возвращаемся на вкладку «Data», где теперь появились записи (рис. 12).

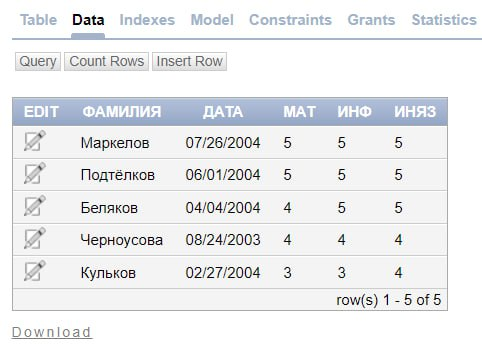


Рис. 12. Просмотр данных таблицы PI\_43: вкладка «Data» после добавления записей

3. Внесите изменения в структуру таблицы – создайте поле, которое будет содержать пол студента.

Для изменения структуры таблицы возвращаемся во вкладку «Table» («Таблица»). Помимо описания структуры данных она содержит кнопки:

* «Add Column» (дословно – «Добавить столбец», корректнее – «Добавить поле») – добавить новое поле в таблице;
* «Modify Column» (дословно – «Изменить столбец», корректнее – «Изменить поле») – изменить существующее поле в таблице (его тип данных, длину и др. параметры, кроме имени);
* «Rename Column» (дословно – «Переименовать столбец», корректнее – «Переименовать поле») – переименовать поле в таблице;
* «Drop Column» (дословно – «Удалить столбец», корректнее – «Удалить поле») – удалить поле в таблице;
* «Rename» («Переименовать») – переименовать таблицу;
* «Copy» («Копировать») – копировать таблицу;
* «Drop» («Удалить») – удалить таблицу целиком;
* «Truncate» («Урезать») – удалить все записи в таблице;
* «Create Lookup Table» - создать таблицу подстановки.

Для добавления нового поля нам необходимо нажать на кнопку «Add Column». Откроется форма для заполнения параметров поля (рис. 13). Вводим название поля – «Пол», выбираем тип данных – VARCHAR2 (строковый), Lenght (длина) – 2 (для хранения 1 буквы «М» или «Ж» необходимо 2 байта). После этого нажимаем на кнопку «Next >»

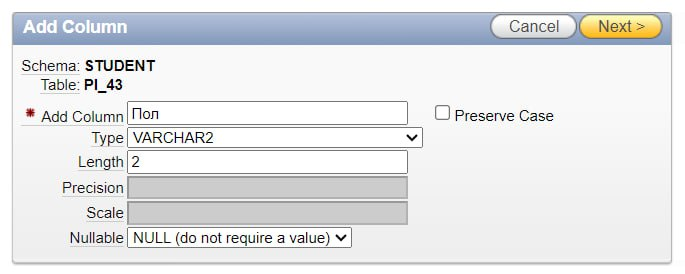


Рис. 13. Создание нового поля

Программа выведет информацию о создаваемом поле для проверки (рис. 14). Если пользователь заметит ошибку, он может вернуться на один из предыдущих шагов с помощью кнопки «< Previous» и внести какие-либо изменения. Если же всё верно, необходимо нажать на кнопку «Finish» («Финиш»), после чего поле будет создано.



Рис. 14. Проверка перед созданием поля

Возвращаемся на вкладку «Data», чтобы проверить успешность создания поля (рис. 15).

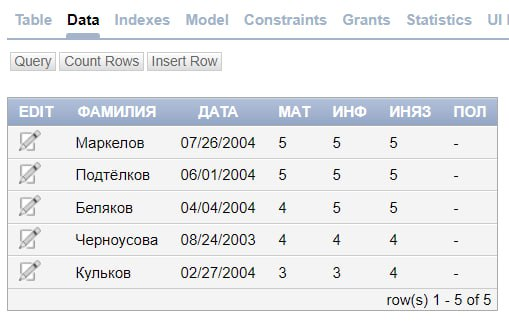


Рис. 15. Таблица PI\_43 с новым полем «Пол»

4. Дополните данными таблицу, указав пол каждого студента.

Внести изменения в существующие записи можно во вкладке «Data». Для этого необходимо нажать на иконку в виде листа бумаги и карандаша напротив соответствующей записи (см. рис. 15).

Откроется форма редактирования значений полей (рис. 16), похожая на форму создания записи (см. рис. 11). В каждой записи заполняем поле «Пол» значениями «М» или «Ж», после чего нажимаем на кнопку «Apply Changes» («Применить изменения»).

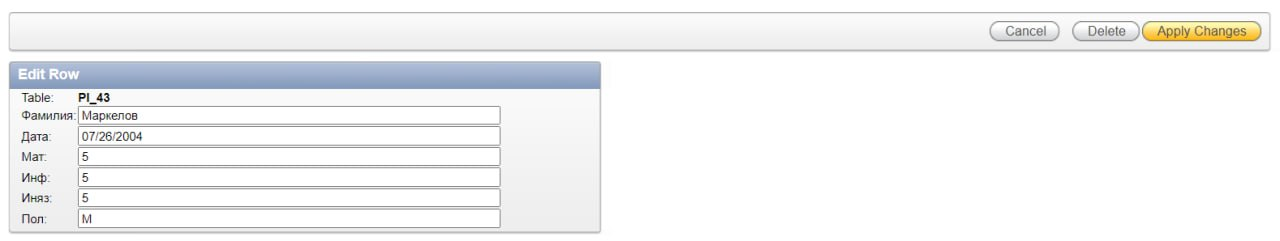


Рис. 16. Изменение записи

Таблица PI\_43 после заполнения поля «Пол» представлена на рис. 17.

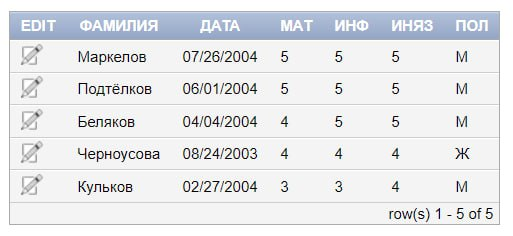


Рис. 17. Таблица PI\_43 с заполненным полем «Пол»

5. Создайте таблицу с теми же полями – она будет содержать сведения о студентах параллельной группы. Решите это задание способом, отличным от того, каким было выполнено задание 1.

Наиболее эффективным способом создания таблицы с такими же полями является копирование её структуры. Для этого необходимо перейти во вкладку «Table» и нажать на кнопку «Copy».

Откроется форма копирования таблицы (рис. 18). В этой форме необходимо ввести имя новой таблицы, а также выбрать, какие поля и записи нужно копировать.

Назовём новую таблицу «PI\_41». Так как от нас требуется создать таблицу с теми же полями, в выпадающем списке «Columns» (~~«Столбцы»~~ «Поля») выбираем вариант «All Columns» («Все ~~столбцы~~ поля»). При этом она будет содержать сведения о студентах параллельной группы, т. е. записи о студентах группы ПИ-43 там будут не нужны, поэтому в выпадающем списке «Display» («Отображать») выбираем вариант «No Rows» («Нет ~~строк~~ записей»). После этого нажимаем на кнопку «Next >».

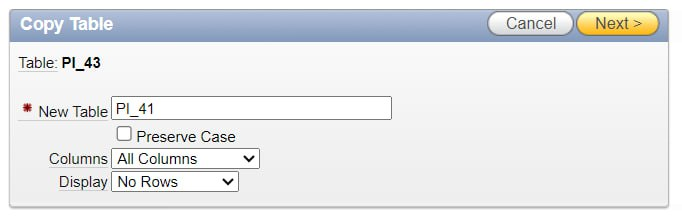


Рис. 18. Копирование структуры таблицы PI\_43 в новую таблицу PI\_41

Программа вновь попросит подтвердить корректность выполняемых действий (рис. 19). Нажимаем на кнопку «Finish».

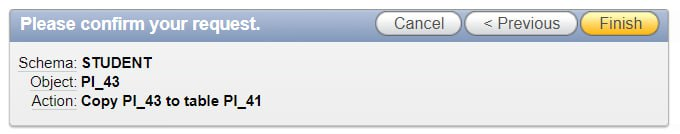


Рис. 19. Проверка перед копированием структуры таблицы

После создания новой таблицы откроется её структура, скопированная из таблицы PI\_43 (рис. 20). Если мы откроем вкладку «Data», записей мы не увидим, т. к. мы выбрали не копировать их (рис. 21).

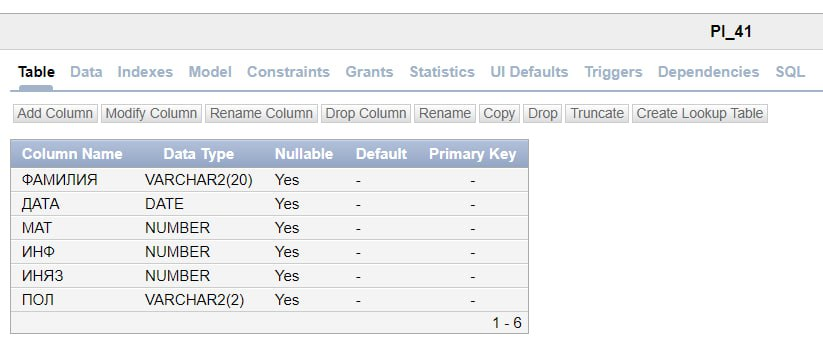


Рис. 20. Структура таблицы PI\_41

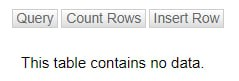


Рис. 21. Содержание таблицы PI\_41 до внесения записей

6. Внесите во вторую таблицу 5 записей.

Делаем это способом, аналогичным заданию № 2 (см. рис. 11), за тем лишь исключением, что здесь поле «Пол» заполняется сразу при создании записи (рис. 22).

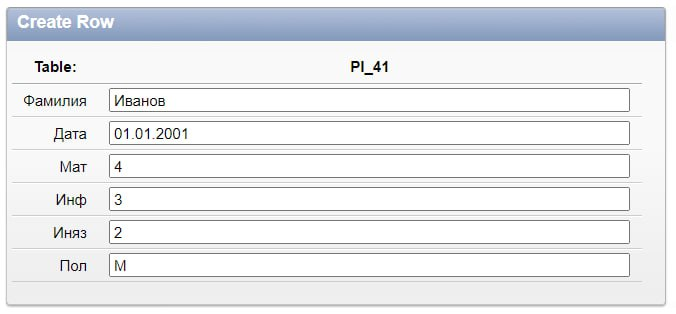


Рис. 22. Создание записей в таблице PI\_41

Таблица PI\_41 после внесения записей представлена на рис. 23.



Рис. 23. Таблица PI\_41 после внесения записей

7. Осуществите сортировку записей первой таблицы в алфавитном порядке поля фамилий.

Для осуществления сортировки записей по алфавиту выполним запрос. Для этого перейдем на вкладку «Data» и нажмем на кнопку «Query».

Откроется инструмент создания запроса (рис. 24). В списке полей проставляем галочки во всех чекбоксах, чтобы в результат запроса попали все поля.

Проводить фильтрацию записей в данном случае не требуется, поэтому все графы «Column Condition» («Условие ~~столбца~~ поля») оставляем пустыми.

Под списком находятся 3 выпадающих списка «Order by» («Сортировать по»). В первом списке выбираем поле «Фамилия», остальные не трогаем.

Под выпадающем полем содержится 2 варианта сортировки: «Ascending» («По возрастанию») и «Descending» («По убыванию»). Выбираем вариант «Ascending», т. к. по алфавиту = по возрастанию.

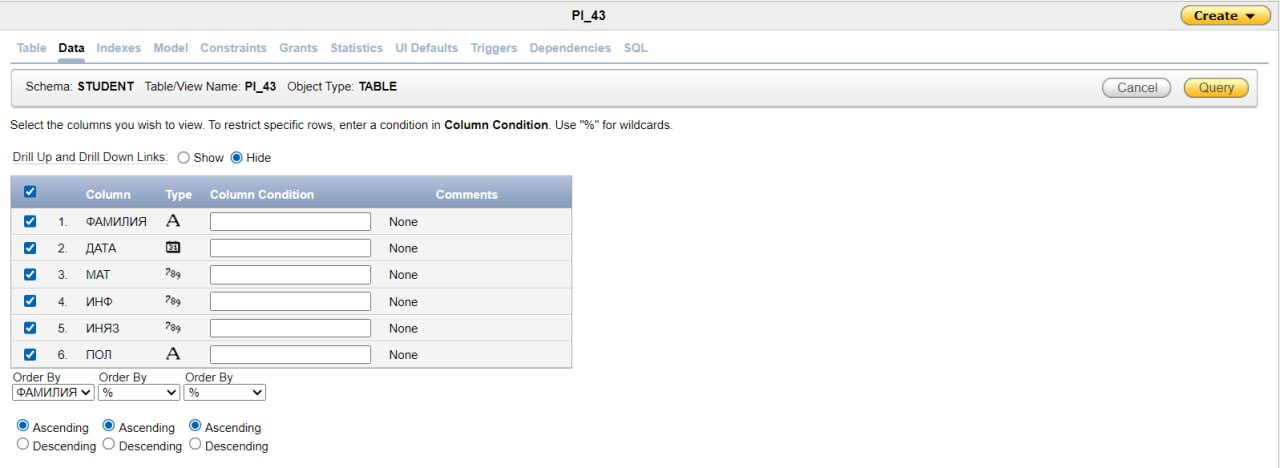


Рис. 24. Создание запроса на сортировку по алфавиту

После этого нажимаем на кнопку «Query» («Запрос») для выполнения запроса. Результат запроса представлен на рис. 25. Записи отсортированы верно – от Б до Ч.



Рис. 25. Результат запроса: сортировка по алфавиту

При этом выполнение запросов никак не влияет на содержание таблицы. Если мы вернемся к исходной таблице, там будет всё по-старому (рис. 26).

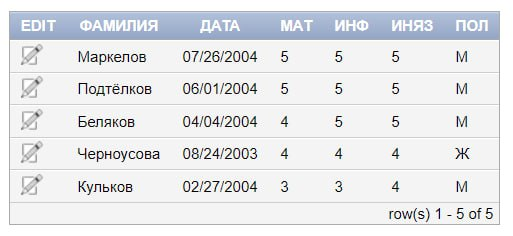


Рис. 26. Таблица PI\_43 после выполнения запроса

8. Сформируйте запросы на выборку:

а) студентов с указанием фамилии и даты рождения

Создаем запрос аналогично заданию № 7 (см. рис. 24). Но на этот раз с помощью чекбоксов выбираем только 2 поля: «Фамилия» и «Дата» (рис. 27). В графе «Column Condition» по-прежнему ничего не пишем. «Order by» здесь не заполняем.

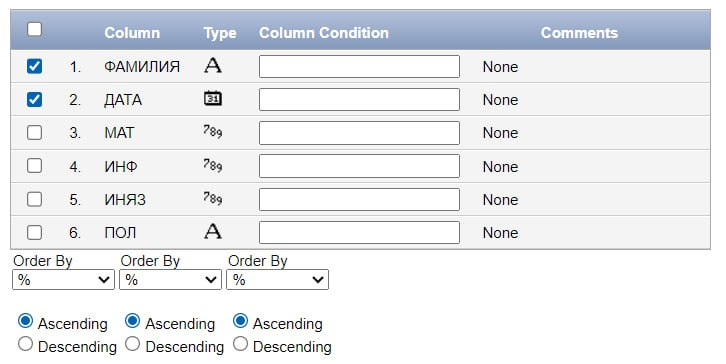


Рис. 27. Создание запроса на вывод фамилий и дат рождения

Результат запроса представлен на рис. 28. Программа вывела только поля «Фамилия» и «Дата рождения». Поля «Мат», «Инф», «ИнЯз», «Пол» в результатах запроса не отобразились, при этом из таблицы они не удалены.

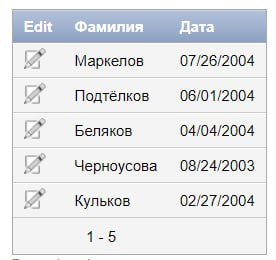


Рис. 28. Результат запроса: вывод фамилий и дат рождения

б) студентов-отличников по математике

Создаем запрос аналогично заданиям выше. Выбираем поля «Фамилия» и «Мат». У поля «Мат» в графе «Column Condition» пишем «5» (рис. 29). «Order by» не заполняем.

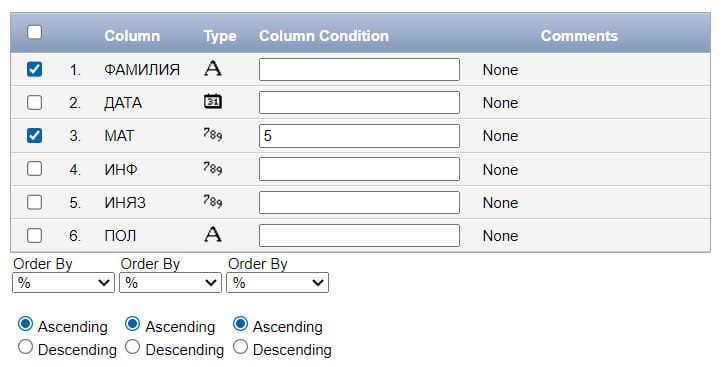


Рис. 29. Создание запроса на вывод отличников по математике

Результат запроса представлен на рис. 30. Сверяя с исходной таблицей (см. рис. 26), убеждаемся, что оценка «5» по математике действительно стоит только у данных студентов.

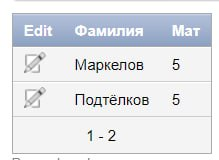


Рис. 30. Результат запроса: отличники по математике

в) студентов-отличников по всем предметам

Запрос почти полностью аналогичен предыдущему, за тем лишь исключением, что чекбоксы также проставляются у полей «Инф» и «ИнЯз», а в графе «Column Condition» у них также пишется «5» (рис. 31).

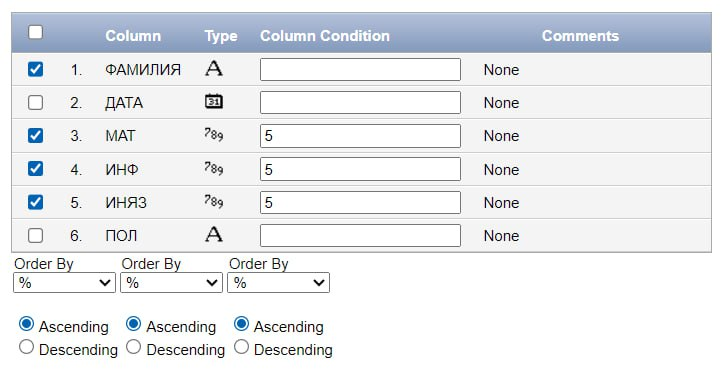


Рис. 31. Создание запроса на вывод отличников по всем предметам

Результат запроса представлен на рис. 32. Сверяя с исходной таблицей (см. рис. 26), убеждаемся, что одновременно оценка «5» по математике, информатике и иностранному языку действительно стоит только у данных студентов.



Рис. 32. Результат запроса: отличники по всем предметам

г) студентов, чей возраст старше 20 лет

На данный момент в таблице содержатся данные только о студентах старше 20 лет. Следовательно, данный запрос выведет всех студентов, что не позволит объективно проверить его правильность – возможно, он работает некорректно и выводит вообще все записи.

Для проведения объективного тестирования намеренно исказим данные в таблице и установим 2 студентам 2006 год рождения (рис. 33). Тем самым, на день выполнения лабораторной работы (08.12.2025) им будет 19 лет.

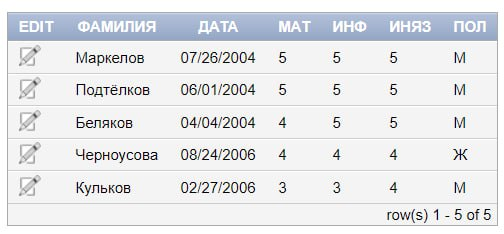


Рис. 33. Таблица PI\_43 со студентами младше 20 лет

Выполнить данный запрос средствами интерфейса Oracle DB не удалось. Попытка выполнения была произведена следующим образом. Галочки были проставлены в чекбоксах полей «Фамилия» и «Год». В графе «Column Condition» для поля «Дата» было указано значение «<= 12.08.2005» (рис. 34). «Order by» не заполнялось.

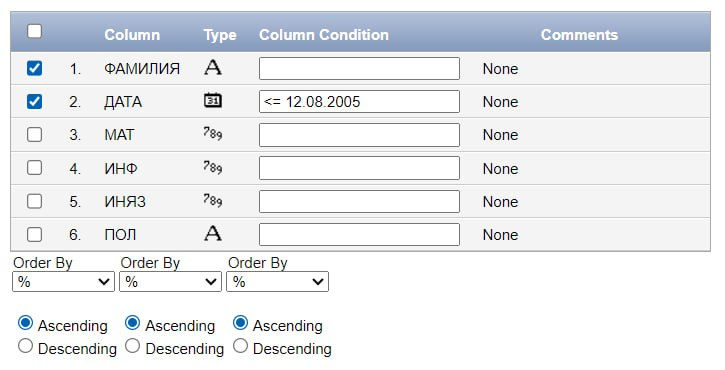


Рис. 34. Создание запроса на вывод студентов старше 20 лет

В результате попытки выполнения запроса была получена ошибка (рис. 35):

*report error: ORA-01858: a non-numeric character was found where a numeric was expected*

или в переводе на русский:

*сообщение об ошибке: ORA-01858: вместо ожидаемого числа обнаружен нецифровой символ.*

Это говорит о том, что программа не принимает в графе «Column Condition» никаких значений, кроме числовых.

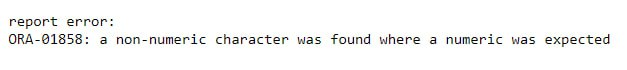


Рис. 35. Ошибка при выполнении запроса

Перебирались и многие другие варианты записи даты в «Column Condition», но все они не увенчались успехом. Удалось лишь выводить записи по конкретным датам (рис. 36-37).

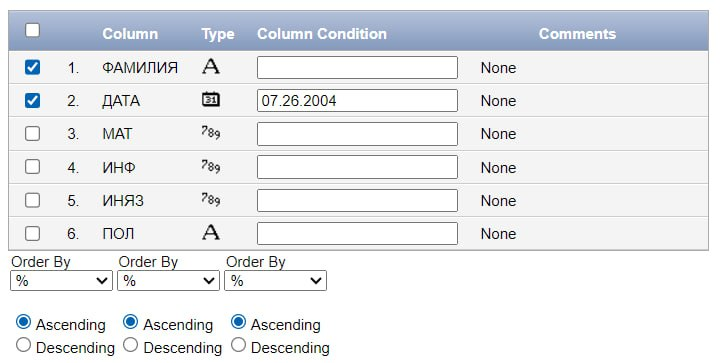


Рис. 36. Создание запроса на вывод студентов, родившихся 26 июля 2004 года

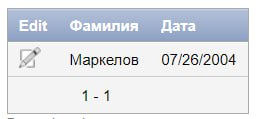


Рис. 37. Результат запроса: студенты, родившиеся 26 июля 2004 года

Попробуем выполнить задание средствами SQL. На верхней панели вкладок выбираем «SQL Workshop», затем переходим в раздел «SQL Commands» (рис. 38).

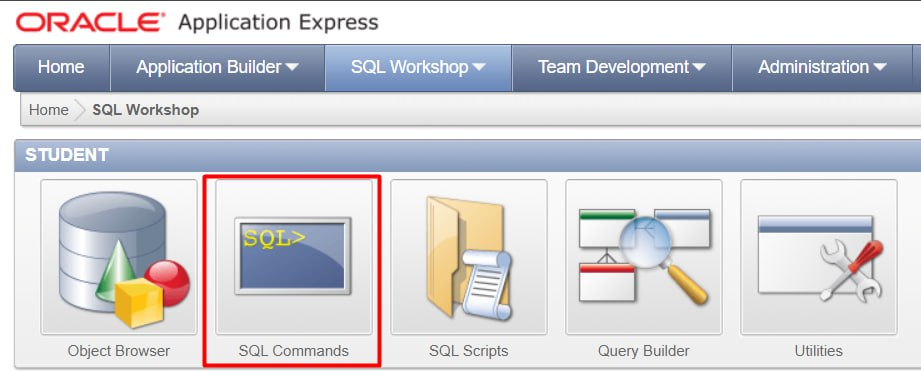


Рис. 38. Открытие SQL Commands

В открывшемся поле вводим запрос (рис. 39):

SELECT \* FROM PI\_43

WHERE "Дата" < ADD\_MONTHS(SYSDATE, -12\*20);



Рис. 39. Поле для ввода SQL-запросов

Результат запроса представлен на рис. 40. Студенты 2006 года рождения в данной таблице отсутствуют, что подтверждает корректность работы запроса.

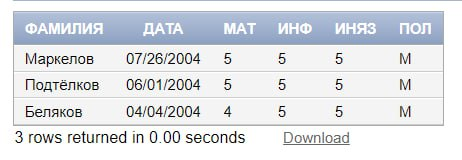


Рис. 40. Результат запроса: студенты старше 20 лет

9. Сформируйте параметрический запрос:

а) студентов с фамилией «Иванов»

Выполнить запрос средствами интерфейса Oracle DB также не удалось, поэтому вновь будем использовать SQL. Параметризированный SQL-запрос в Oracle DB имеет следующий вид:

SELECT \* FROM Имя\_таблицы

WHERE Имя\_колонки = :Параметр;

где **:Параметр** – специальная метка, которую пользователь заменяет своим значением при запуске запроса.

Вводим следующий запрос (рис. 41):

SELECT \* FROM PI\_43

WHERE "ФАМИЛИЯ" = :Surname;



Рис. 41. Создание параметрического запроса по фамилии

После запуска запроса откроется всплывающее окно, содержащее текстовое поле, в котором необходимо ввести значение параметра (рис. 42). Как и требовалось по условию, вводим фамилию «Иванов», а затем нажимаем на кнопку «Submit» («Утвердить»).

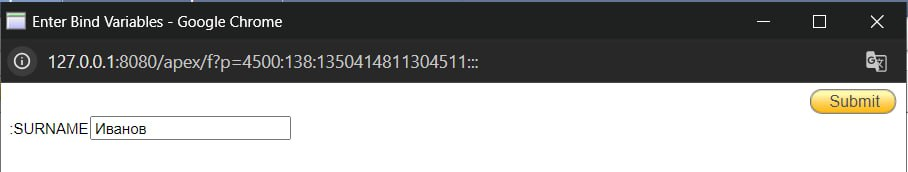


Рис. 42. Ввод параметра для параметризированного запроса

Программа выдала сообщение «No data found» – «Данные не найдены». Это не ошибка. В таблице PI\_43 действительно нет студентов с фамилией «Иванов». Зато такой студент есть в таблице PI\_41. Для проверки правильности запроса повторим его, но уже для таблицы PI\_41 (рис. 43-44):

SELECT \* FROM PI\_41

WHERE "ФАМИЛИЯ" = :Surname;



Рис. 43. Ввод запроса

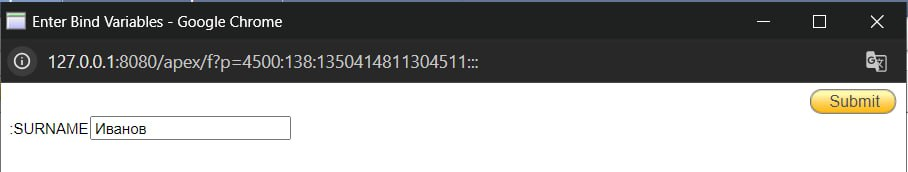


Рис. 44. Ввод параметра

Результат запроса представлен на рис. 45. Действительно, студент с фамилией «Иванов» был выведен. Запрос работает корректно.

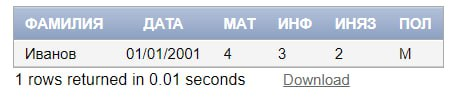


Рис. 45. Результат параметрического запроса по фамилии

б) студентов Ивановых, имеющих «5» по математике

SQL-запрос будет иметь вид (рис. 46):

SELECT \* FROM PI\_41

WHERE "ФАМИЛИЯ" = :Surname AND "МАТ" = :Math;

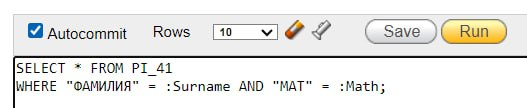


Рис. 46. Ввод запроса

Перед выполнением запроса заменим оценку по математике у Иванова на 5, чтобы запрос выдал результат (рис. 47).

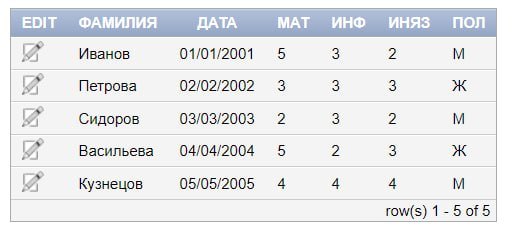


Рис. 47. Таблица PI\_41 перед выполнением запроса

Выполняем запрос, вводим параметры – «Иванов» и «5» (рис. 48) и получаем результат (рис. 49). Запрос работает корректно.



Рис. 48. Ввод параметра

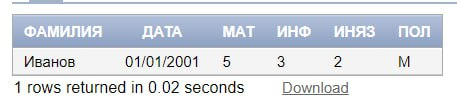


Рис. 49. Результат параметрического запроса по фамилии и оценке

1. Командная строка

Работа в командной строке осуществляется с помощью консоли «Run SQL Command Line».

1. Создайте таблицу базы данных с полями, содержание которых – фамилия, год рождения, оценки по математике, информатике, иностранному языку (рис. 50).

CREATE TABLE PI43 (

"Фамилия" VARCHAR2(40),

"Дата" DATE,

"Мат" NUMBER,

"Инф" NUMBER,

"ИнЯз" NUMBER

);

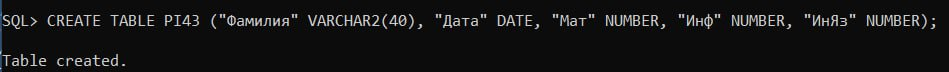


Рис. 50. Создание таблицы

1. Внесите в таблицу 5 записей (рис. 51-52).

INSERT INTO PI43 ("Фамилия","Дата","Мат","Инф","ИнЯз")

VALUES ('Маркелов', DATE '2004-07-26', 5, 5, 5);

INSERT INTO PI43 ("Фамилия","Дата","Мат","Инф","ИнЯз")

VALUES ('Подтёлков', DATE '2004-06-01', 5, 5, 5);

INSERT INTO PI43 ("Фамилия","Дата","Мат","Инф","ИнЯз")

VALUES ('Беляков', DATE '2004-04-04', 4, 5, 5);

INSERT INTO PI43 ("Фамилия","Дата","Мат","Инф","ИнЯз")

VALUES ('Черноусова', DATE '2003-08-24', 4, 4, 4);

INSERT INTO PI43 ("Фамилия","Дата","Мат","Инф","ИнЯз")

VALUES ('Кульков', DATE '2004-02-27', 3, 3, 3);

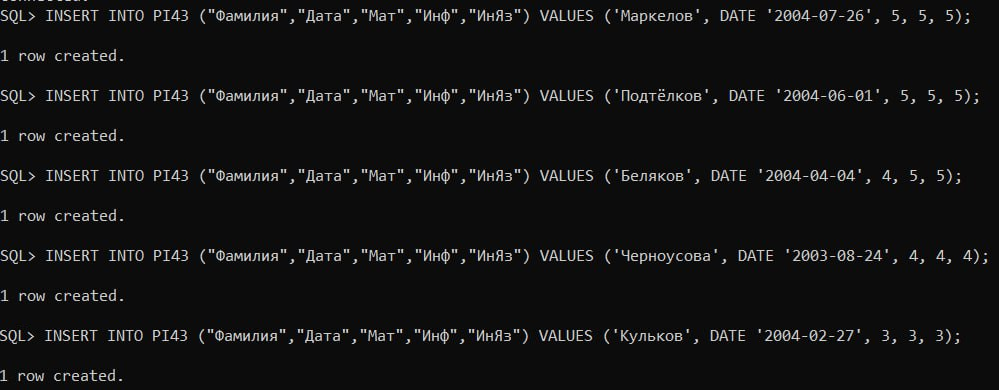


Рис. 51. Внесение записей

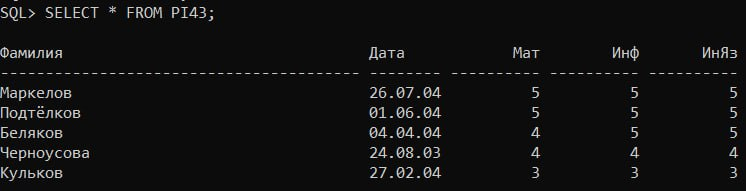


Рис. 52. Вывод таблицы

1. Внесите изменения в структуру таблицы – создайте поле, которое будет содержать пол студента (рис. 53).

ALTER TABLE PI43

ADD ("Пол" VARCHAR2(6));

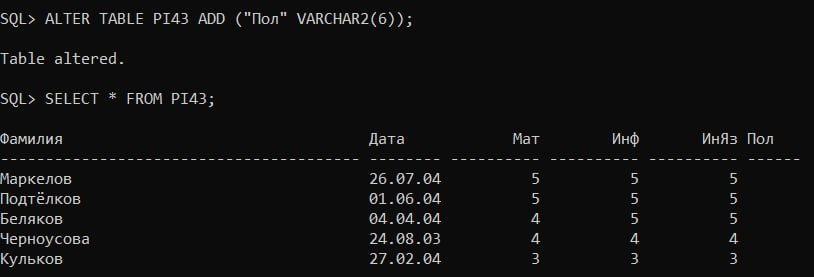


Рис. 53. Добавление поля «Пол»

1. Дополните данными таблицу, указав пол каждого студента (рис. 54-55)

UPDATE PI43

SET "Пол" = 'М'

WHERE "Фамилия" IN ('Маркелов', 'Подтёлков', 'Беляков', 'Кульков');

UPDATE PI43

SET "Пол" = 'Ж'

WHERE "Фамилия" = 'Черноусова';

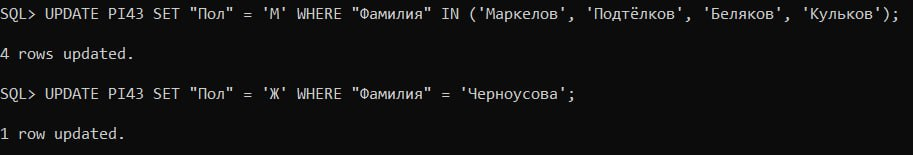


Рис. 54. Добавление значений поля «Пол»

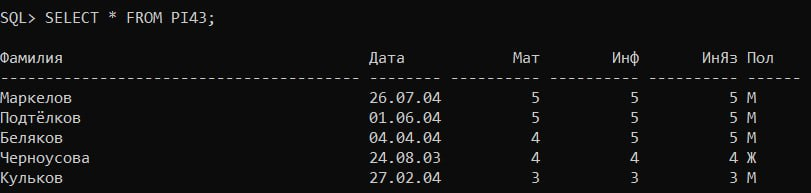


Рис. 55. Вывод таблицы

1. Создайте таблицу с теми же полями – она будет содержать сведения о студентах параллельной группы. Решите это задание способом, отличным от того, каким было выполнено задание 1 (рис. 56).

CREATE TABLE PI41 AS

SELECT \* FROM PI43

WHERE 1 = 0;

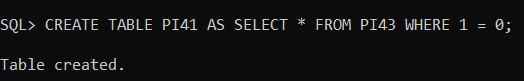


Рис. 56. Копирование структуры таблицы

1. Внесите во вторую таблицу 5 записей (рис. 57-58).

INSERT INTO PI41 ("Фамилия","Дата","Мат","Инф","ИнЯз", "Пол")

VALUES ('Иванов', DATE '2001-01-21', 4, 3, 2, 'М');

INSERT INTO PI41 ("Фамилия","Дата","Мат","Инф","ИнЯз", "Пол")

VALUES ('Петрова', DATE '2002-02-02', 3, 3, 3, 'Ж');

INSERT INTO PI41 ("Фамилия","Дата","Мат","Инф","ИнЯз", "Пол")

VALUES ('Сидоров', DATE '2003-03-03', 2, 3, 2, 'М');

INSERT INTO PI41 ("Фамилия","Дата","Мат","Инф","ИнЯз", "Пол")

VALUES ('Васильева', DATE '2004-04-04', 5, 2, 3, 'Ж');

INSERT INTO PI41 ("Фамилия","Дата","Мат","Инф","ИнЯз", "Пол")

VALUES ('Кузнецов', DATE '2005-05-05', 4, 4, 4, 'М');

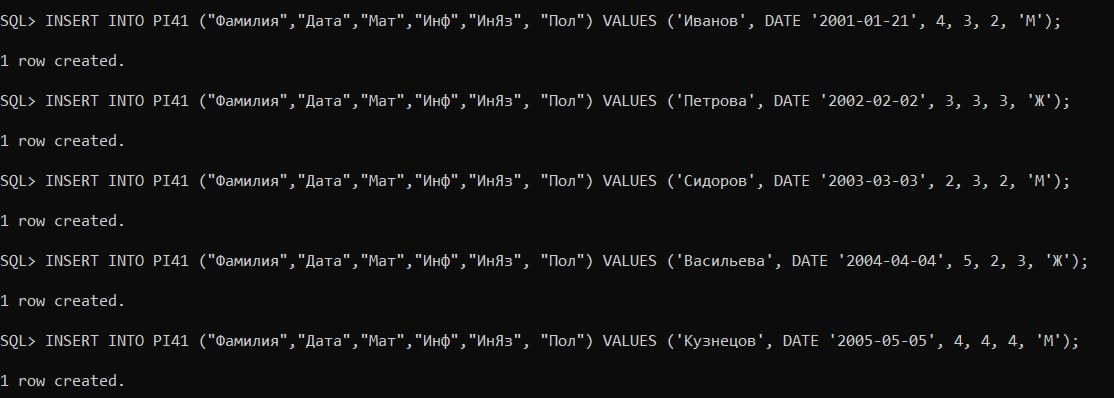


Рис. 57. Внесение записей во вторую таблицу

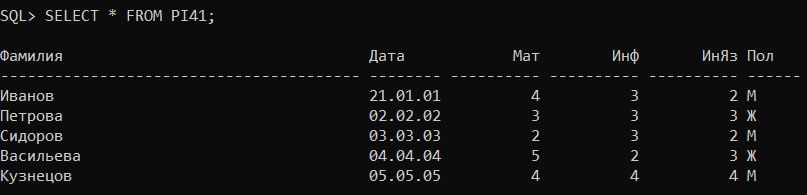


Рис. 58. Вывод второй таблицы

1. Осуществите сортировку записей первой таблицы в алфавитном порядке поля фамилий.

SELECT \* FROM PI43

ORDER BY "Фамилия" ASC;

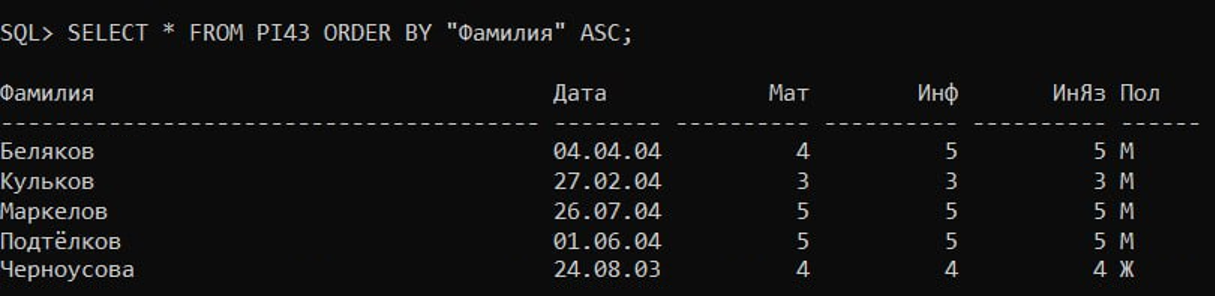


Рис. 59. Сортировка таблицы по алфавиту

1. Сформируйте запросы на выборку:

а) студентов с указанием фамилии и даты рождения (рис. 60)

SELECT "Фамилия", "Дата"

FROM PI43;

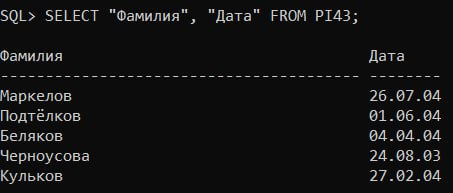


Рис. 60. Запрос на вывод фамилий и дат рождения

б) студентов-отличников по математике (рис. 61)

SELECT \* FROM PI43

WHERE "Мат" = 5;

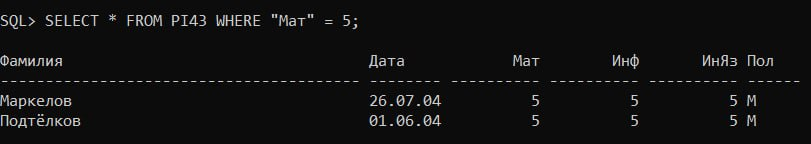


Рис. 61. Запрос отличников по математике

в) студентов-отличников по математике (рис. 62)

SELECT \* FROM PI43

WHERE "Мат" = 5 AND "Инф" = 5 AND "ИнЯз" = 5;

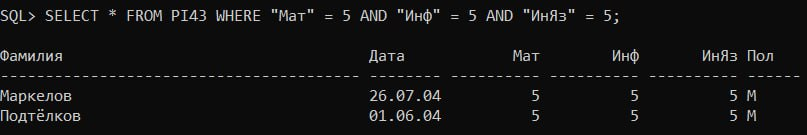


Рис. 62. Запрос отличников по всем предметам

г) студентов, чей возраст старше 20 лет (рис. 63-64)



Рис. 63. Таблица PI43 со студентами младше 20 лет

SELECT \* FROM PI43

WHERE "Дата" < ADD\_MONTHS(SYSDATE, -12\*20);

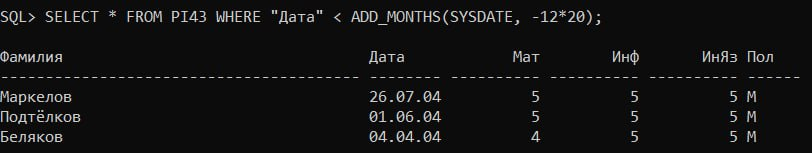


Рис. 64. Запрос студентов старше 20 лет

1. Сформируйте параметрический запрос:

а) студентов с фамилией «Иванов» (рис. 65)

SELECT \* FROM PI41

WHERE "Фамилия" = '&Surname';

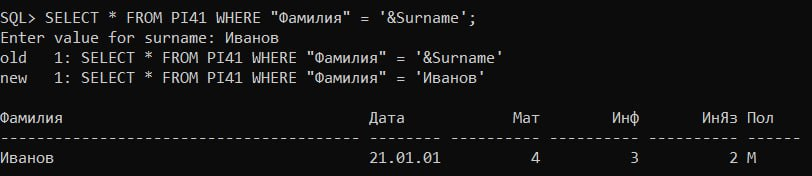


Рис. 65. Параметрический запрос по фамилии

б) студентов Ивановых, имеющих «5» по математике (рис. 66-67)

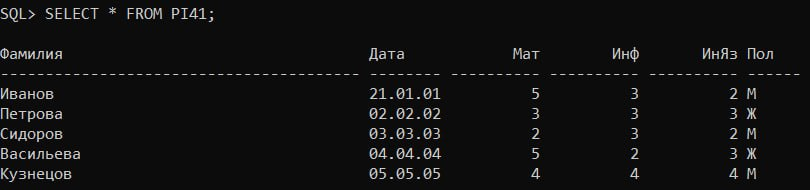


Рис. 66. Таблица PI41 с оценкой «5» по математике у Иванова

SELECT \* FROM PI41

WHERE "Фамилия" = '&Surname' AND "Мат" = &Math;

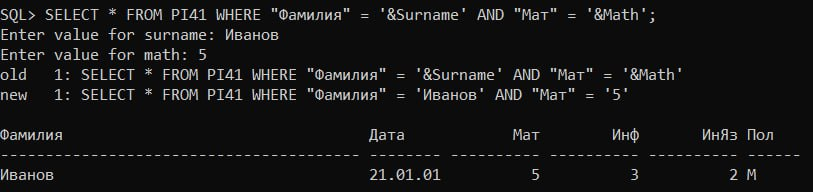


Рис. 67. Параметрический запрос по фамилии и оценке