Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант №1501

Лабораторная работа №3

По дисциплине

Базы Данных

Выполнил студент группы P3115:

Барсуков Максим

Преподаватель:

Горбунов Михаил Витальевич

Николаев Владимир Вячеславович

Санкт-Петербург 2023 г.

1. **Текст задания**

По варианту, выданному преподавателем, составить и выполнить запросы к базе данных "Учебный процесс".

Составить запросы на языке SQL (пункты 1-7).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ДАТА.  
   Фильтры (AND):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 163249.  
   Вид соединения: RIGHT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ИМЯ < Александр.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276.  
   c) Н\_УЧЕНИКИ.ГРУППА = 3100.  
   Вид соединения: INNER JOIN.
3. Вывести число фамилий без учета повторений.  
   При составлении запроса нельзя использовать DISTINCT.
4. Найти группы, в которых в 2011 году было ровно 10 обучающихся студентов на ФКТИУ.  
   Для реализации использовать подзапрос.
5. Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст), где средний возраст меньше среднего возраста в группе 1100.
6. Получить список студентов, отчисленных до первого сентября 2012 года с очной или заочной формы обучения (специальность: Программная инженерия). В результат включить:  
   номер группы;  
   номер, фамилию, имя и отчество студента;  
   номер пункта приказа;  
   Для реализации использовать подзапрос с EXISTS.
7. Вывести список людей, не являющихся или не являвшихся студентами ФКТИУ (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ). В запросе нельзя использовать DISTINCT.
8. **Реализация запросов на SQL**

|  |
| --- |
| -- 1.  -- Сделать запрос для получения атрибутов из указанных таблиц,  -- применив фильтры по указанным условиям:  -- Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  -- Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ДАТА.  -- Фильтры (AND):  -- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.  -- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 163249.  -- Вид соединения: RIGHT JOIN.  **SELECT** "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ВЕДОМОСТИ"."ДАТА"  **FROM** "Н\_ВЕДОМОСТИ"  **RIGHT** **JOIN** "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" **ON** "Н\_ВЕДОМОСТИ"."ВЕД\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД"  **WHERE** "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" = 'Ведомость'  **AND** "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 163249;  ---------------------------------------------------------  -- 2.  -- Сделать запрос для получения атрибутов из указанных таблиц,  -- применив фильтры по указанным условиям:  -- Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  -- Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД.  -- Фильтры: (AND)  -- a) Н\_ЛЮДИ.ИМЯ < Александр.  -- b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276.  -- c) Н\_УЧЕНИКИ.ГРУППА = 3100.  -- Вид соединения: INNER JOIN.  **SELECT** "Н\_ЛЮДИ"."ИМЯ", "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД", "Н\_УЧЕНИКИ"."ИД"  **FROM** "Н\_ЛЮДИ"  **JOIN** "Н\_ОБУЧЕНИЯ" **ON** "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  **JOIN** "Н\_УЧЕНИКИ" **ON** "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  **WHERE** "Н\_ЛЮДИ"."ИМЯ" < 'Александр'  **AND** "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" > 163276  **AND** "Н\_УЧЕНИКИ"."ГРУППА" = '3100';  ---------------------------------------------------------  -- 3.  -- Вывести число фамилий без учета повторений.  -- При составлении запроса нельзя использовать DISTINCT.  **SELECT** count("ФАМИЛИЯ") **FROM**  (**SELECT** "Н\_ЛЮДИ"."ФАМИЛИЯ" **AS** "ФАМИЛИЯ"  **FROM** "Н\_ЛЮДИ"  **GROUP** **BY** "Н\_ЛЮДИ"."ФАМИЛИЯ")  **AS** "Число фамилий без повторений";  -- Проверка: SELECT COUNT(DISTINCT "Н\_ЛЮДИ"."ФАМИЛИЯ") FROM "Н\_ЛЮДИ";  ---------------------------------------------------------  -- 4.  -- Найти группы, в которых в 2011 году было ровно 10 обучающихся студентов на ФКТИУ.  -- Для реализации использовать подзапрос.  **SELECT** "ГРУППЫ\_КТиУ\_2011"."ГРУППА", "ГРУППЫ\_КТиУ\_2011"."КОЛИЧЕСТВО" **FROM**  (**SELECT** "Н\_УЧЕНИКИ"."ГРУППА", count("Н\_УЧЕНИКИ"."ИД") **AS** "КОЛИЧЕСТВО" **FROM** "Н\_УЧЕНИКИ"  **JOIN** "Н\_ПЛАНЫ"  **ON** "Н\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД"  **AND** "Н\_ПЛАНЫ"."УЧЕБНЫЙ\_ГОД" = '2010/2011'  **JOIN** "Н\_ОТДЕЛЫ"  **ON** "Н\_ОТДЕЛЫ"."ИД" = "Н\_ПЛАНЫ"."ОТД\_ИД"  **AND** "Н\_ОТДЕЛЫ"."КОРОТКОЕ\_ИМЯ" = 'КТиУ'  **GROUP** **BY** "Н\_УЧЕНИКИ"."ГРУППА"  ) **AS** "ГРУППЫ\_КТиУ\_2011"  **WHERE** "ГРУППЫ\_КТиУ\_2011"."КОЛИЧЕСТВО" = 10;  ---------------------------------------------------------  -- 5.  -- Выведите таблицу со средним возрастом студентов во всех группах (Группа, Средний возраст),  -- где средний возраст меньше среднего возраста в группе 1100.  **SELECT** "Н\_УЧЕНИКИ"."ГРУППА", avg(date\_part('year', age("Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ")))  **FROM** "Н\_ЛЮДИ"  **JOIN** "Н\_УЧЕНИКИ" **ON** "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  **GROUP** **BY** "Н\_УЧЕНИКИ"."ГРУППА"  **HAVING** avg(date\_part('year', age("Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ"))) < (  **SELECT** avg(date\_part('year', age("Н\_ЛЮДИ"."ДАТА\_РОЖДЕНИЯ")))  **FROM** "Н\_ЛЮДИ"  **JOIN** "Н\_УЧЕНИКИ" **ON** "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  **WHERE** "Н\_УЧЕНИКИ"."ГРУППА" = '1100'  );  ---------------------------------------------------------  -- 6.  -- Получить список студентов, отчисленных до первого сентября 2012 года  -- с очной или заочной формы обучения (специальность: Программная инженерия).  -- В результат включить:  -- номер группы;  -- номер, фамилию, имя и отчество студента;  -- номер пункта приказа;  -- Для реализации использовать подзапрос с EXISTS.  **SELECT** "ВНЕШ\_УЧЕНИКИ"."ГРУППА",  "ВНЕШ\_УЧЕНИКИ"."ИД",  "Н\_ЛЮДИ"."ФАМИЛИЯ",  "Н\_ЛЮДИ"."ИМЯ",  "Н\_ЛЮДИ"."ОТЧЕСТВО",  "ВНЕШ\_УЧЕНИКИ"."П\_ПРКОК\_ИД"  **FROM** "Н\_УЧЕНИКИ" "ВНЕШ\_УЧЕНИКИ"  **JOIN** "Н\_ЛЮДИ" **ON** "Н\_ЛЮДИ"."ИД" = "ВНЕШ\_УЧЕНИКИ"."ЧЛВК\_ИД"  **JOIN** "Н\_ПЛАНЫ" **ON** "ВНЕШ\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД"  **JOIN** "Н\_ФОРМЫ\_ОБУЧЕНИЯ" **ON** "Н\_ПЛАНЫ"."ФО\_ИД" = "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."ИД"  **AND** ("Н\_ФОРМЫ\_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Заочная' **OR** "Н\_ФОРМЫ\_ОБУЧЕНИЯ"."НАИМЕНОВАНИЕ" = 'Очная')  **JOIN** "Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ" **ON** "Н\_ПЛАНЫ"."НАПС\_ИД" = "Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ"."ИД"  **JOIN** "Н\_НАПР\_СПЕЦ" **ON** "Н\_НАПР\_СПЕЦ"."ИД" = "Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ"."НС\_ИД"  **AND** "Н\_НАПР\_СПЕЦ"."НАИМЕНОВАНИЕ" = 'Программная инженерия'  **WHERE** **EXISTS** (  **SELECT** \*  **FROM** "Н\_УЧЕНИКИ" "ВНУТР\_УЧЕНИКИ"  **WHERE** "ВНУТР\_УЧЕНИКИ"."ПРИЗНАК" = 'отчисл'  **AND** "ВНУТР\_УЧЕНИКИ"."СОСТОЯНИЕ" = 'утвержден'  **AND** "ВНУТР\_УЧЕНИКИ"."ИД" = "ВНЕШ\_УЧЕНИКИ"."ИД"  **AND** DATE("ВНУТР\_УЧЕНИКИ"."КОНЕЦ") < '2012-09-01'  );  ---------------------------------------------------------  -- 7.  -- Вывести список людей, не являющихся или не являвшихся студентами ФКТИУ  -- (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ).  -- В запросе нельзя использовать DISTINCT.  **SELECT** "people"."ИД",  "people"."ФАМИЛИЯ",  "people"."ИМЯ",  "people"."ОТЧЕСТВО"  **FROM** "Н\_ЛЮДИ" **AS** people  **WHERE** **NOT** **EXISTS** (  **SELECT** \*  **FROM** "Н\_УЧЕНИКИ"  **JOIN** "Н\_ПЛАНЫ" **ON** "Н\_УЧЕНИКИ"."ПЛАН\_ИД" = "Н\_ПЛАНЫ"."ИД"  **JOIN** "Н\_ОТДЕЛЫ" **ON** "Н\_ПЛАНЫ"."ОТД\_ИД" = "Н\_ОТДЕЛЫ"."ИД"  **AND** "Н\_ОТДЕЛЫ"."КОРОТКОЕ\_ИМЯ" = 'КТиУ'  **WHERE** "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "people"."ИД"  ); |

1. **Вывод**

При выполнении лабораторной работы я познакомился с основными функциями языка SQL и диалекта PostgreSQL. Научился писать запросы, получать, агрегировать, отсеивать и сортировать полученные данные с использованием различных синтаксических конструкций языка. В результате был освоен язык DML SQL, предназначенный для работы с данными, хранящимися внутри базы данных.