

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

по базам данных

Вариант № 6285

Выполнила: Гафурова Фарангиз

Фуркатовна

Группа: Р3120

Принял: Николаев Владимир Вячеславович

Санкт-Петербург. 2024

Текст задания

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

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ДАТА.
Фильтры (AND):
а) Н_ЛЮДИ.ИД = 152862.
б) Н_ВЕДОМОСТИ.ЧЛВК_ИД < 153285.
с) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 117219.
Вид соединения: LEFT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ, Н_СЕССИЯ.
Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.УЧГОД.
Фильтры (AND):
а) Н_ЛЮДИ.ИД = 152862.
б) Н_ВЕДОМОСТИ.ИД > 39921.
Вид соединения: LEFT JOIN.
3. Составить запрос, который ответит на вопрос, есть ли среди студентов вечерней формы обучения люди без ИНН.
4. В таблице Н_ГРУППЫ_ПЛАНОВ найти номера планов, по которым обучается (обучалось) более 2 групп на заочной форме обучения.
Для реализации использовать соединение таблиц.
5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср_оценка), у которых средняя оценка равна минимальной оценк(е|и) в группе 1100.
6. Получить список студентов, отчисленных ровно первого сентября 2012 года с заочной формы обучения (специальность: Программная инженерия). В результат включить:
номер группы;
номер, фамилию, имя и отчество студента;
номер пункта приказа;
Для реализации использовать соединение таблиц.
7. Сформировать запрос для получения числа в СПбГУ ИТМО троечников.

Решение

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ДАТА.

Фильтры (AND):

а) Н_ЛЮДИ.ИД = 152862.

б) Н_ВЕДОМОСТИ.ЧЛВК_ИД < 153285.

с) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 117219.

Вид соединения: LEFT JOIN.

```
select Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ДАТА from Н_ЛЮДИ
left join Н_ВЕДОМОСТИ ON (Н_ВЕДОМОСТИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД)
where Н_ЛЮДИ.ИД = 152862 and Н_ВЕДОМОСТИ.ЧЛВК_ИД < 153285
and Н_ВЕДОМОСТИ.ЧЛВК_ИД > 117219;
```

Вывод:

ИД	ДАТА
152862	2010-04-30 00:00:00
152862	2011-01-12 00:00:00
152862	2010-05-25 00:00:00
152862	2011-06-08 00:00:00
152862	2010-12-30 00:00:00
152862	2010-12-28 00:00:00
.....	
152862	2010-06-21 00:00:00
152862	2010-06-28 00:00:00
152862	2010-06-17 00:00:00

2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ, Н_СЕССИЯ.

Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.УЧГОД.

Фильтры (AND):

а) Н_ЛЮДИ.ИД = 152862.

б) Н_ВЕДОМОСТИ.ИД > 39921.

Вид соединения: LEFT JOIN.

```
select Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.УЧГОД
from Н_ЛЮДИ left join Н_ВЕДОМОСТИ ON (Н_ВЕДОМОСТИ.ЧЛВК_ИД =
Н_ЛЮДИ.ИД)
left join Н_СЕССИЯ ON (Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД)
where Н_ЛЮДИ.ИД = 152862 and Н_ВЕДОМОСТИ.ИД > 39921;
```

Вывод:

ИМЯ	ДАТА	УЧГОД
Александр	2010-04-30 00:00:00	
Александр	2011-01-12 00:00:00	
Александр	2010-05-25 00:00:00	
Александр	2011-06-08 00:00:00	
Александр	2010-12-30 00:00:00	
Александр	2010-12-28 00:00:00	
Александр	2010-12-27 00:00:00	
Александр	2011-02-03 00:00:00	
Александр	2011-01-19 00:00:00	
Александр	2011-02-02 00:00:00	
Александр	2011-01-24 00:00:00	
Александр	2011-01-15 00:00:00	

.....

Александр	2010-06-28 00:00:00	
Александр	2010-06-21 00:00:00	
Александр	2010-06-28 00:00:00	
Александр	2010-06-17 00:00:00	

(43 троки)

3. Составить запрос, который ответит на вопрос, есть ли среди студентов вечерней формы обучения люди без ИНН.

2		Очно-заочная(вечерняя)		ISU_UCHEB		2011-11-08 15:55:54		ISU_UCHEB	
2011-11-08 15:55:54		вечерняя		вечерней		вечерней		вечернюю	
	вечерней		вечерней						

```
SELECT COUNT(*) AS cnt FROM Н_ЛЮДИ
LEFT JOIN Н_УЧЕНИКИ ON (Н_УЧЕНИКИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД)
```

```

LEFT JOIN Н_ПЛАНЫ ON (Н_УЧЕНИКИ.ПЛАН_ИД = Н_ПЛАНЫ.ИД)
LEFT JOIN Н_ФОРМЫ_ОБУЧЕНИЯ ON (Н_ПЛАНЫ.ФО_ИД = Н_ФОРМЫ_ОБУЧЕНИЯ.ИД)
WHERE Н_ЛЮДИ.ИНН IS NULL AND Н_ФОРМЫ_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очно-
заочная(вечерняя)';

```

Вывод:

```

cnt
-----
1142
(1 строка)

```

- В таблице Н_ГРУППЫ_ПЛАНОВ найти номера планов, по которым обучается (обучалось) более 2 групп на очной форме обучения.
Для реализации использовать соединение таблиц.
ИД заочная очная форма 1

```

select Н_ГРУППЫ_ПЛАНОВ.ПЛАН_ИД from Н_ГРУППЫ_ПЛАНОВ
left join Н_ПЛАНЫ ON (Н_ГРУППЫ_ПЛАНОВ.ПЛАН_ИД = Н_ПЛАНЫ.ИД)
JOIN Н_ФОРМЫ_ОБУЧЕНИЯ ON Н_ПЛАНЫ.ФО_ИД = Н_ФОРМЫ_ОБУЧЕНИЯ.ИД AND
Н_ФОРМЫ_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очная'
GROUP BY Н_ГРУППЫ_ПЛАНОВ.ПЛАН_ИД HAVING COUNT(*) > 2;

```

Вывод:

```

ПЛАН_ИД
-----
6521
273
272
874
40

.....
871
7438
7583
880
(69 трок)

```

- Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср_оценка), у которых средняя оценка равна минимальной оценк(е|и) в группе 1100.

```

SELECT * FROM(
    SELECT Н_ЛЮДИ.ИД,
           Н_ЛЮДИ.ФАМИЛИЯ,
           Н_ЛЮДИ.ИМЯ,
           Н_ЛЮДИ.ОТЧЕСТВО,
           AVG(CAST("Н_ВЕДОМОСТИ"."ОЦЕНКА" AS FLOAT))
           as avrg FROM
    Н_ЛЮДИ
    JOIN Н_ВЕДОМОСТИ
      ON (Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД)
    JOIN Н_УЧЕНИКИ
      ON (Н_УЧЕНИКИ.ЧЛВК_ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД)

```

```

WHERE H_ВЕДОМОСТИ.ОЦЕНКА ~ '[0-5]'
AND H_УЧЕНИКИ.ГРУППА = '4100'
GROUP BY H_ЛЮДИ.ИД, H_ЛЮДИ.ФАМИЛИЯ, H_ЛЮДИ.ИМЯ,
H_ЛЮДИ.ОТЧЕСТВО
) AS tmp
WHERE avrg = (
SELECT MIN(CAST(H_ВЕДОМОСТИ.ОЦЕНКА AS FLOAT))
FROM H_УЧЕНИКИ
JOIN H_ВЕДОМОСТИ
ON (H_УЧЕНИКИ.ЧЛВК_ИД = H_ВЕДОМОСТИ.ЧЛВК_ИД)
WHERE H_ВЕДОМОСТИ.ОЦЕНКА ~ '[0-5]'
AND H_УЧЕНИКИ.ГРУППА = '1100'
);

```

Вывод:

ИД | ФАМИЛИЯ | ИМЯ | ОТЧЕСТВО | avrg

-----+-----+-----+-----+-----

(0 строк)

- Получить список студентов, отчисленных ровно первого сентября 2012 года с заочной формы обучения (специальность: Программная инженерия). В результат включить: номер группы; номер, фамилию, имя и отчество студента; номер пункта приказа; Для реализации использовать соединение таблиц.

```

SELECT H_ЛЮДИ.ИД, H_УЧЕНИКИ.ГРУППА, H_ЛЮДИ.ФАМИЛИЯ, H_ЛЮДИ.ИМЯ,
H_ЛЮДИ.ОТЧЕСТВО, H_УЧЕНИКИ.КОНЕЦ_ПО_ПРИКАЗУ
FROM H_УЧЕНИКИ
JOIN H_ПЛАНЫ ON (H_УЧЕНИКИ.ПЛАН_ИД = H_ПЛАНЫ.ИД)
JOIN H_НАПРАВЛЕНИЯ_СПЕЦИАЛ ON (H_ПЛАНЫ.НАПС_ИД = H_НАПРАВЛЕНИЯ_СПЕЦИАЛ.ИД)
JOIN H_ЛЮДИ ON (H_УЧЕНИКИ.ЧЛВК_ИД = H_ЛЮДИ.ИД)
JOIN H_НАПР_СПЕЦ ON (H_НАПРАВЛЕНИЯ_СПЕЦИАЛ.НС_ИД = H_НАПР_СПЕЦ.ИД)
JOIN H_ФОРМЫ_ОБУЧЕНИЯ ON (H_ПЛАНЫ.ФО_ИД = H_ФОРМЫ_ОБУЧЕНИЯ.ИД AND
H_ФОРМЫ_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очная')
WHERE CAST(H_УЧЕНИКИ.КОНЕЦ_ПО_ПРИКАЗУ as DATE) < '2012-09-01'
AND H_НАПР_СПЕЦ.НАИМЕНОВАНИЕ = 'Программная инженерия';

```

Вывод:

ИД	ГРУППА	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	КОНЕЦ_ПО_ПРИКАЗУ
-----	-----	-----	-----	-----	-----
154927	1101	Богданов	Евгений	.	2012-08-31 00:00:00
158040	1101	Агафонов	Иван	Вячеславович	2012-08-31 00:00:00
161875	1100	Быховский	Михаил	Валентинович	2012-08-31 00:00:00
162093	1100	Кучеренко	Сергей	Владимирович	2012-08-31 00:00:00
162103	1100	Андреев	Иван	Ильич	2012-08-31 00:00:00
154956	1100	Степанова	Александр	Валерьевич	2012-08-31 00:00:00
161903	1100	Кирсанов	Владимир	Валерьевич	2012-08-31 00:00:00

161906 | 1100 | Белых | Владимир | Михайлович | 2012-08-31 00:00:00
161966 | 1100 | Хуан Лэ | Алексей | Дмитриевич | 2012-08-31 00:00:00
160739 | 1100 | Федотова | Олег | Владимирович | 2012-08-31 00:00:00
160960 | 1100 | Ярославкин | Александр | Олегович | 2012-08-31 00:00:00

7. Сформировать запрос для получения числа в СПбГУ ИТМО троечников.

```
select COUNT(*) from Н_ЛЮДИ join(  
  select Н_ВЕДОМОСТИ.ЧЛВК_ИД  
  from Н_ВЕДОМОСТИ  
  WHERE Н_ВЕДОМОСТИ.ОЦЕНКА ~ '[0-5]'  
  GROUP BY Н_ВЕДОМОСТИ.ЧЛВК_ИД  
  HAVING AVG(CAST(Н_ВЕДОМОСТИ.ОЦЕНКА as FLOAT)) = 3  
) as troe ON Н_ЛЮДИ.ИД = troe.ЧЛВК_ИД;
```

Вывод:

```
count  
-----  
133  
(1 строка)
```

Вывод

При выполнении лабораторной работы я познакомилась с основными функциями PostgreSQL. Научилась 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.