

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

ЗАПРОСЫ НА ВЫБОРКУ ДАННЫХ

Отчёт о лабораторной работе № 3
по дисциплине «Базы данных»

Выполнил: студентка гр. 430-2

_____ Лузинсан А.А.

«___»_____ 2022 г.

Проверил: ассистент каф. АСУ

_____ Яблонский Я. В.

«___»_____ 2022 г.

Томск 2022

Оглавление

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ.....	2
2 ОПИСАНИЕ РАБОТЫ.....	3
Выводы.....	8

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

Цель работы: научиться создавать и редактировать запросы на выборку данных на языке SQL.

2 ОПИСАНИЕ РАБОТЫ

В ходе выполнения данной лабораторной работы в соответствии с индивидуальным заданием были созданы запросы на языке SQL, которые были выполнены в среде pgAdmin и соответствуют следующим формулировкам:

1. Выбрать всех подписчиков указанного издания.
2. Для каждого издания подсчитать количество подписчиков.
3. Получить действительный на указанную дату список подписчиков с указанием адресов и изданий, на которые они подписаны.
4. Получить группы подписчиков, проживающих на одной улице в указанном городе, и отсортировать их по номерам домов.

Текст запроса SQL, реализующий формулировку «Выбрать всех подписчиков указанного издания» представлен далее:

```
CREATE VIEW t_subs_pub_donkey  
AS SELECT fio as Подписчики_на_Мин_говор_ослы  
FROM t_subs  
JOIN t_publishing on idpub=id  
JOIN t_pod on t_subs.idpod=t_pod.id  
WHERE pubname='Миниатюрные говорящие ослы';
```

В данном запросе нас интересуют подписчики некоторого издания (для примера взято издание «Миниатюрные говорящие ослы»), поэтому ключевой таблицей будет фигурировать таблица t_subs, так как она содержит ссылки на подписчиков и соответствующие издания, на которые они подписаны. Таким образом, чтобы получить ФИО подписчика производится соединение отношения t_subs с отношением t_pod по условию совпадения внешнего ключа t_subs.idpod с ключом t_pod.id. Далее, чтобы отсеять подписчиков по определённому изданию, производится выборка с помощью предложения WHERE с условием совпадения названия издания с определённой строкой: pubname='Миниатюрные говорящие ослы'. Но для того, чтобы иметь доступ к

строке названия издания, производится соединение отношения t_subs с отношением t_publishing по условию совпадения внешнего ключа t_subs.idpub с ключом t_publishing.id. В конце запрос оформляется в виде представления с помощью команды CREATE VIEW t_subs_pub_donkey AS. В результате выполнения вышеприведённого запроса и последующего вызова запроса на вывод столбца созданного представления в виде: SELECT * FROM t_subs_pub_donkey;, получаем результат, представленный на рисунке 2.1.

	Подписчики_на_Мин_говор_ослы character varying (85)
1	Добрыйвечер Добромир Миронов
2	Камрад Артём Златоустович
3	Пьяных Татьяна Николаевна
4	Какаев Аркадий Акакиевич

Рисунок 2.1 — Содержание представления t_subs_pub_donkey

Текст запроса SQL, реализующий формулировку «Для каждого издания подсчитать количество подписчиков» представлен далее:

```
CREATE VIEW t_cpod_publishings
AS SELECT
pubname AS Издание,
COUNT(idpod) AS Количество_подписчиков
FROM t_subs
JOIN t_publishing ON t_subs.idpub=t_publishing.id
GROUP BY idpub, pubname;
```

Чтобы реализовать данный запрос, нам потребуется отношение t_subs, соединяющее сведения об изданиях, содержащихся в отношении t_publishing и подписчиках, которые на эти издания подписаны. Для подсчёта подписчиков нам потребуется только идентификационный номер, который уже содержится в отношении t_subs, поэтому в качестве дополнительной таблицы

используем только отношение `t_publishing`. Соединяя таблицу `t_subs` с таблицей `t_publishing` по условию совпадения внешнего ключа `t_subs.idpub` с первичным ключом `t_publishing.id` получаем объединённую таблицу для взятия в последующем столбца с названием издания. Далее группируем отношение по индексу издания и названию издания, получая при этом сгруппированные по изданиям записи подписчиков. В качестве выборки указываются столбец с названием издания, который переименовывается в строку «Издание» и столбец «Количество_подписчиков», формируемый с помощью агрегированной функции `COUNT`, подсчитывающей количество уникальных подписчиков. В конце запрос оформляется в виде представления командой `CREATE VIEW t_cpod_publishings AS`. Таким образом получаем содержание созданного представления командой `SELECT * FROM t_cpod_publishings;`, изображение которого представлено на рисунке 2.2.

	Издание character varying (50)	Количество_подписчиков bigint
1	Миниатюрные говорящие ослы	4
2	Разыскивается	2
3	Медвежья жизнь	3
4	Современный Пьяница	4
5	Новости трубочиста	1

Рисунок 2.2 — Содержание представления `t_cpod_publishings`

Текст запроса SQL, реализующий формулировку «Получить действенный на указанную дату список подписчиков с указанием адресов и изданий, на которые они подписаны.» представлен далее:

```
CREATE VIEW t_pods_2017_07_21
AS SELECT fio as ФИО,
'Регион ' || region || ', город ' || city
|| ', улица ' || street || ', дом ' || house
|| ', квартира ' || apartment as Адрес,
```

```

t_publishing.pubname as Издание
FROM t_subs
JOIN t_pod ON t_pod.id=idpod
JOIN t_publishing ON t_publishing.id=idpub
JOIN t_address ON t_address.id=t_pod.idaddr
WHERE '2017-07-21' >= datestart
AND '2017-07-21' <= datestart + make_interval(months => idperiod);

```

Данный запрос обращается к таблице `t_subs`, и соединяясь с таблицами `t_pod`, `t_publishing` и `t_address` по первичным ключам `t_pod.id`, `t_publishing.id` и `t_address.id` соответственно, получаем подготовленную таблицу для дальнейших действий, реализуемых в предложении `WHERE`. В условии `WHERE` указывается временной промежуток, который задаётся как нижняя граница в виде `'2017-07-21' >= datestart` и верхняя граница, как `'2017-07-21' <= datestart + make_interval(months => idperiod)`, что указывает на соответствующую дату от начала действия подписки с учётом выделенного количества месяцев, в рамках данного плана подписки. Профильтровав по этому условию, обращаемся к полям `region`, `city`, `street`, `house`, `apartment`, которые образуют адрес подписчика путём склеивания столбцов, а также к полю отношения `t_publishing pubname`, именуя его как `Издание`. Оформляем таблицу через представление `t_pods_2017_07_21`, содержание которого представлено на рисунке 2.3.

	ФИО character varying (85)	Адрес text	Издание character varying (50)
1	Добрыивечер Добромир Миронов	Регион Досадный регион, город Новоголвилль, улица Провал Синицина, дом 500, квартира ...	Разыскивается
2	Фамильяров Геннадий Иннокентьевич	Регион Досадный регион, город Новоголвилль, улица Горшок, дом 77, квартира 47	Медвежья жизнь
3	Шорохова Евгения Олеговна	Регион Досадный регион, город Новоголвилль, улица Золотова, дом 38, квартира 26	Разыскивается
4	Шорохов Олег Евгеньевич	Регион Досадный регион, город Новоголвилль, улица Золотова, дом 39, квартира 47	Современный Пьяница
5	Какаев Аркадий Акакиевич	Регион Стекланный округ, город Скрежетальск, улица Артефактов, дом 336, квартира 77	Миниатюрные говорящие ослы

Рисунок 2.3 — Содержание представления `t_pods_2017_07_21`

Последний SQL запрос, реализующий формулировку «Получить группы подписчиков, проживающих на одной улице в указанном городе, и отсортировать их по номерам домов.» выглядит следующим образом:

```

CREATE VIEW t_novogol vill
AS SELECT street as Улица_города_Новоголвилль, fio as ФИО
FROM t_subs
JOIN t_pod ON t_pod.id=t_subs.idpod
JOIN t_address ON t_address.id=t_pod.idaddr
WHERE city='Новоголвилль'
GROUP BY street, house, fio
ORDER BY house;

```

В данном запросе необходимо обратиться к отношению t_subs, соединив его с t_pod по внешнему ключу t_subs.idpod с первичным ключом t_pod.id, а также соединить отношение t_pod с t_address по внешнему ключу t_pod.idaddr с первичным ключом t_address.id. Далее группируем, в соответствии с заданием, таблицу по столбцам street, house и fio. Указываем фильтр по городу «Новоголвилль», и теперь остаётся только отсортировать строки по номерам домов, что реализуется посредством предложения ORDER BY house. Выделяем отдельный столбец street, именуя как Улица_города_Новоголвилль, и столбец fio, обозначив как ФИО. В конечном счёте результат работы оформляется в представление t_novogol vill, содержание которого представлено на рисунке 2.4.

	Улица_города_Новоголвилль character varying (50)	ФИО character varying (85)
1	Золотова	Шорохова Евгения Олеговна
2	Золотова	Шорохов Олег Евгеньевич
3	Горшок	Фамильяров Геннадий Иннокентьевич
4	Провал Синицина	Камрад Артём Златоустович
5	Провал Синицина	Добрыйвечер Добромир Миронов

Рисунок 2.4 — Содержание представления t_novogol vill

Выводы

В результате выполнения данной лабораторной работы я научилась создавать и редактировать запросы на выборку данных на языке SQL, а также получила практические навыки создания запросов в среде pgAdmin в соответствие с формулировками, написанными на естественном языке.