МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИИ

«Санкт-Петербургский Политехнический университет Петра Великого» Институт компьютерных наук и технологий Высшая школа искусственного интеллекта

Курс: Методы проектирования баз данных

Отчет по лабораторным работам курса

Выполнила студентка группы 3530201/90101 Андреева Наталия Сергеевна

Принял

Попов Сергей Геннадьевич

Санкт-Петербург, 2022

Содержание

1	Постановка задачи	3
2	Схема базы данных	4
3	Лабораторная №1. View и запрос	5
	3.1 Создание view	5
	3.2 Запрос	6
4	Лабораторная №2. Триггер	9
5	Лабораторная №3. Права пользователей	13
6	Лабораторная №4. Транзакции	16
7	Лабораторная №5. Процедура и функция	18
	7.1 Процедура	18
	7.2 Функция	19
38	аключение	21

1 Постановка задачи

В ходе курса «Методы разработки баз данных» необходимо выполнить следующие задачи, соответствующие 5 лабораторным работам:

- создать view и запрос, использующий как созданную view, так и таблицы базы данных;
- создать таблицу и триггер, который будет вносить в эту таблицу новые данные по событию;
- создать 2 пользователей, один из которых мог бы только просматривать view, а другой мог бы выполнять любые действия с содержимым view;
- в транзакционной модели выбрать уровень изоляции и продемонстрировать одну из проблем, которая возникает или наоборот не возникает на этом уровне;
- создать процедуру и функцию и продемонстрировать работу функции в запросе.

2 Схема базы данных

В лабораторных работах используется база данных ветеринарной клиники. Ее схема на русском и на английском приведена на рисунках 1 и 2 соответственно.

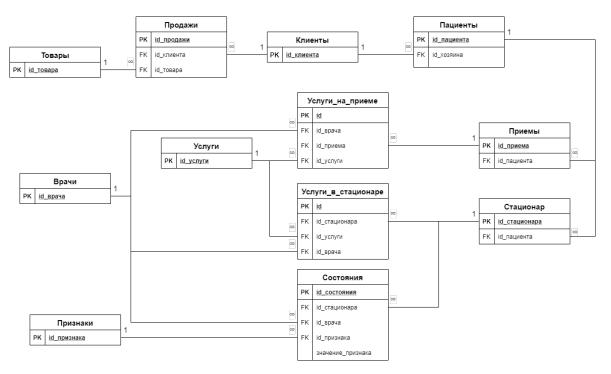


Рис. 1: Схема базы данных на русском языке

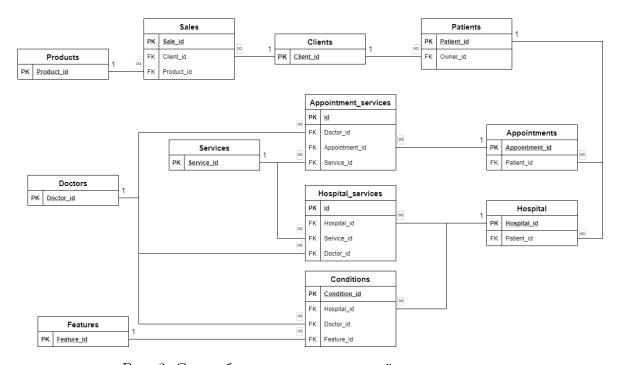


Рис. 2: Схема базы данных на английском языке

3 Лабораторная №1. View и запрос

Формулировка задания:

- 1. Создать view, в которой для каждого врача ведется подсчет числа приемов и числа приемов в стационаре.
- 2. Создать запрос с использованием view, позволяющий для конкретного врача показать число приемов и всех пациентов.

3.1 Создание view

Далее приведен скрипт для создания view.

```
CREATE VIEW Appointments_number AS
 SELECT Doctors.Doctor_id AS Doctor_id, Second_name, Number_of_appointments,
        Number_of_hospital_appointments
FROM
 Doctors LEFT JOIN
  (SELECT Doctor_id, count(*) AS Number_of_appointments
   FROM
   (SELECT Doctor_id, Appointment_id
   FROM appointment_services
    GROUP BY Doctor_id, Appointment_id) AS Doctors_and_Appointments
   GROUP BY Doctor_id
   ORDER BY Doctor_id) AS From_appointments
  ON (From_appointments.Doctor_id=Doctors.Doctor_id)
  (SELECT Doctor_id, count(*) AS Number_of_hospital_appointments
  FROM
   (SELECT Doctor_id, Hospital_id
    FROM hospital_services
    GROUP BY Doctor_id, Hospital_id) AS Doctors_and_Hospitals
   GROUP BY Doctor_id) AS From_hospital
  ON (Doctors.Doctor_id=From_hospital.Doctor_id);
  На рисунке 3 приведено содержимое созданной view.
```

На рисунке 4 приведен EXPLAIN для SELECT в скрипте создания view.

Doctor_id	Second_name	Number_of_appointments	Number_of_hospital_appointments
1	Залежнева	2653	2678
2	Соколова	2677	2635
3	Петрова	2725	2628
4	Лебедева	2628	2667
5	Сорокина	2720	2674
6	Петелин	2685	2674
7	Соловьев	2684	2676
8	Цаплина	2736	2647
9	Вареникова	2669	2620
10	Червякова	2686	2638
11	Соловьева	2730	2668
12	Степанов	2653	2752
13	Васильев	2611	2669
14	Бородин	2634	2617
15	Зайцев	2676	2689
16	Червякова	2620	2650
17	Егоров	2635	2566
18	Волков	2616	2625
19	Петелин	2700	2638
20	Виноградов	2658	2697

Рис. 3: Результат выборки из view

	id	select_type	table	partitions	type	possible_keys	key	key_len
•	1	PRIMARY	Doctors	NULL	ALL	NULL	NULL	NULL
	1	PRIMARY	<derived2></derived2>	NULL	ref	<auto_key0></auto_key0>	<auto_key0></auto_key0>	5
	1	PRIMARY	<derived4></derived4>	NULL	ref	<auto_key0></auto_key0>	<auto_key0></auto_key0>	5
	4	DERIVED	<derived5></derived5>	NULL	ALL	NULL	NULL	NULL
	5	DERIVED	hospital_services	NULL	ALL	NULL	NULL	NULL
	2	DERIVED	<derived3></derived3>	NULL	ALL	NULL	NULL	NULL
	3	DERIVED	appointment_services	NULL	ALL	NULL	NULL	NULL

ref	rows	filtered	Extra
NULL	20	100.00	NULL
vetdinic.Doctors.Doctor_id	607	100.00	NULL
vetdinic.Doctors.Doctor_id	678	100.00	NULL
NULL	67890	100.00	Using temporary
NULL	67890	100.00	Using temporary
NULL	60777	100.00	Using temporary; Using filesort
NULL	60777	100.00	Using temporary

Рис. 4: Результат выполнения EXPLAIN'а запроса view

3.2 Запрос

Далее приведен скрипт запроса, использующий view.

SELECT DISTINCT Second_name, Number_of_appointments, Moniker

FROM appointment_services JOIN appointments USING (Appointment_id)
 JOIN Patients USING (Patient_id)
 JOIN Appointments_number USING (Doctor_id)
WHERE Doctor_id=1;

На рисунке 5 приведен результат выполнения запроса.

+	+	+	F	Залежнева	26	553	Фокси
Second_name	Number_of_appointments	Moniker		Залежнева	26	553	Симона
+	+	+	٠	Залежнева	26	553	Фред
Залежнева	2653	Берта		Залежнева	26	553	Пип
Залежнева	2653	Оливия		Залежнева	26	553	Феликс
Залежнева	2653	Фея		Залежнева	26	553	Ник
Залежнева	2653	Граф		Залежнева	26	553	Рекс
Залежнева	2653	Снежка		Залежнева	26	553	Дилли
Залежнева	2653	Кейк		Залежнева	26	553	Маркиза
Залежнева	2653	Дина		Залежнева	26	553	Трикси
Залежнева	2653	Дюк		Залежнева	26	553	Милка
Залежнева	2653	Ночка		Залежнева	26	553	Чейз
Залежнева	2653	Лея		Залежнева	26	553	Tecca
Залежнева	2653	Вилли		Залежнева	26	553	Туз
Залежнева	2653	Ава		Залежнева	26	553	Билли
Залежнева	2653	Хэнк		Залежнева	26	553	Барни
Залежнева	2653	Арчи		Залежнева	26	553	Бинго
Залежнева	2653	Лука		Залежнева	26	553	Бруно
Залежнева	2653	Руби		Залежнева	26	553	Тинг
Залежнева	2653	Дора		Залежнева	26	553	Лео
Залежнева	2653	Фиона		Залежнева	26	553	Честер
Залежнева	2653	Кэт		Залежнева	26	553	Салли
Залежнева	2653	Лили		Залежнева	26	553	Бони
Залежнева	2653	Лулу		Залежнева	26	553	Дымка
Залежнева	2653	Тедди		Залежнева	26	553	Айс
Залежнева	2653	Альф		Залежнева	26	553	Ленни
Залежнева	2653	Кора		Залежнева	26	553	Ромашка
Залежнева	2653	Арто		Залежнева	26	553	Полли
Залежнева	2653	Рокси		Залежнева	26	553	Пенелопа
Залежнева	2653	Рик		Залежнева	26	553	Джек
Залежнева	2653	Барон		Залежнева	26	553	Микки
Залежнева	2653	Люк		Залежнева	26	553 İ	Локи
Залежнева	2653	Буян	-	+	+	+	+
Залежнева	2653	Стелла	6	3 rows in set	(0.1676 sec)		

Рис. 5: Результат выполнения запроса

На рисунке 6 приведен результат выполнения EXPLAIN для запроса.

				_			
id	select_type	table	partitions	type	possible_keys	key	key_len
1	PRIMARY	doctors	NULL	const	PRIMARY	PRIMARY	4
1	PRIMARY	<derived5></derived5>	NULL	ref	<auto_key0></auto_key0>	<auto_key0></auto_key0>	5
1	PRIMARY	<derived3></derived3>	NULL	ref	<auto_key0></auto_key0>	<auto_key0></auto_key0>	5
1	PRIMARY	appointment_services	NULL	ref	Doctor_id,Appointment_id	Doctor_id	5
1	PRIMARY	appointments	NULL	eq_ref	PRIMARY,Patient_id	PRIMARY	4
1	PRIMARY	Patients	NULL	eq_ref	PRIMARY	PRIMARY	4
3	DERIVED	<derived4></derived4>	NULL	ALL	NULL	NULL	NULL
4	DERIVED	hospital_services	NULL	ALL	NULL	NULL	NULL
5	DERIVED	<derived6></derived6>	NULL	ALL	NULL	NULL	NULL
6	DERIVED	appointment services	NULL	ALL	HULL	NULL	NULL

ref	rows	filtered	Extra
const	1	100.00	Using temporary
const	10	100.00	HULL
const	10	100.00	HULL
const	3079	100.00	Using where
vetclinic.appointment_services.Appointment_id	1	100.00	Using where
vetclinic.appointments.Patient_id	1	100.00	HULL
NULL	67893	100.00	Using temporary
NULL	67893	100.00	Using temporary
NULL	60779	100.00	Using temporary; Using filesort
NULL	60779	100.00	Using temporary

Рис. 6: Результат выполнения EXPLAIN'а запроса

4 Лабораторная №2. Триггер

Формулировка задания: создать триггер, который позволяет учитывать все услуги врача по их количеству на приеме и в стационаре.

Сначала для триггера была создана таблица Doctor_services. Далее приведен скрипт создания этой таблицы.

```
CREATE TABLE VetClinic.Doctor_services
(
    Doctor_id INT PRIMARY KEY,
    Second_name VARCHAR(30),
    Services_number_on_appointment INT,
    Services_number_in_hospital INT
);

    3arom 6ulum costants Transport and salvon.
```

Затем были созданы триггеры для заполнения созданной таблицы по событию добавления строк в одну из трех таблиц: Врачи, Услуги_на_приеме и Услуги_в_стационаре. Скрипт триггеров приведен далее.

```
CREATE TRIGGER New_appointment_services AFTER INSERT
ON Appointment_services FOR EACH ROW
 INSERT INTO Doctor_services
  SELECT Doctor_id, Second_name, Services_number_on_appointment,
         Services_number_in_hospital
  FROM (
   Doctors LEFT JOIN
   (SELECT Doctor_id, count(*) AS Services_number_on_appointment
   FROM Appointment_services
   GROUP BY Doctor_id) AS From_appointments
  USING (Doctor_id)
  LEFT JOIN
   (SELECT Doctor_id, count(*) AS Services_number_in_hospital
  FROM Hospital_services
  GROUP BY Doctor_id) AS From_hospital
  USING (Doctor id) )
ON DUPLICATE KEY UPDATE Services_number_on_appointment=
                           From_appointments.Services_number_on_appointment;
CREATE TRIGGER New_hospital_services AFTER INSERT
ON Hospital_services FOR EACH ROW
 INSERT INTO Doctor_services
  SELECT Doctor_id, Second_name, Services_number_on_appointment,
         Services_number_in_hospital
 FROM (
   Doctors LEFT JOIN
   (SELECT Doctor_id, count(*) AS Services_number_on_appointment
```

```
FROM Appointment_services
  GROUP BY Doctor_id) AS From_appointments
  USING (Doctor_id)
  LEFT JOIN
   (SELECT Doctor_id, count(*) AS Services_number_in_hospital
  FROM Hospital_services
   GROUP BY Doctor_id) AS From_hospital
  USING (Doctor id) )
 ON DUPLICATE KEY UPDATE Services_number_in_hospital=
                            From_hospital.Services_number_in_hospital;
CREATE TRIGGER New_doctors AFTER INSERT
ON Doctors FOR EACH ROW
 INSERT INTO Doctor_services
  SELECT Doctor_id, Second_name, Services_number_on_appointment,
         Services_number_in_hospital
  FROM (
   Doctors LEFT JOIN
   (SELECT Doctor_id, count(*) AS Services_number_on_appointment
   FROM Appointment_services
   GROUP BY Doctor_id) AS From_appointments
  USING (Doctor id)
  LEFT JOIN
   (SELECT Doctor_id, count(*) AS Services_number_in_hospital
  FROM Hospital_services
  GROUP BY Doctor_id) AS From_hospital
  USING (Doctor_id) )
  ON DUPLICATE KEY UPDATE Second_name=Doctors.Second_name,
                          Services_number_on_appointment=
                             From_appointments.Services_number_on_appointment,
                          Services_number_in_hospital=
                             From_hospital.Services_number_in_hospital;
  Изначально таблица пуста (рис. 7). На рисунке 8 приведено содержимое таблицы
после выполнения следующего скрипта:
INSERT INTO doctors (Doctor_id, Second_name, First_name, Patronymic, Phone_number)
           VALUE (21, 'Бортникова', 'Валентина', 'Алексеевна', '+7(956)245-91-46');
MySQL 127.0.0.1:33060+ ssl vetclinic SQL SELECT * FROM doctor_services;
```

Рис. 7: Изначальное содержание таблицы

Empty set (0.0032 sec)

Doctor_id	Second_name	Services_number_on_appointment	Services_number_in_hospital
1	Залежнева	3079	3423
2	Соколова	3088	3347
3	Петрова	3157	3295
4	Лебедева	3037	3401
5	Сорокина	3153	3395
6	Петелин	3114	3397
7	Соловьев	3100	3397
8	Цаплина	3159	3386
9	Вареникова	3089	3338
10	Червякова	3114	3360
11	Соловьева	3151	3377
12	Степанов	3079	3492
13	Васильев	3013	3419
14	Бородин	3075	3346
15	Зайцев	3110	3392
16	Червякова	3045	3350
17	Егоров	3042	3234
18	Волков	3024	3318
19	Петелин	3123	3390
20	Виноградов	3123	3443
21	Бортникова	NULL	NULL

Рис. 8: Таблица после вставки в таблицу «Врачи»

Далее была сделана всавка в таблицу «Услуги_на_приеме» с помощью следующего скрипта:

Результат работы триггера приведен на рисунке 9.

После этого в таблицу «Услуги_в_стационаре» была добавлена запись с помощью скрипта:

Результат работы триггера приведен на рисунке 10.

Doctor_id	Second_name	Services_number_on_appointment	Services_number_in_hospital
1	Залежнева	3079	3423
2	Соколова	3088	3347
3	Петрова	3157	3295
4	Лебедева	3037	3401
5	Сорокина	3153	3395
6	Петелин	3114	3397
7	Соловьев	3100	3397
8	Цаплина	3159	3386
9	Вареникова	3089	3338
10	Червякова	3114	3360
11	Соловьева	3151	3377
12	Степанов	3079	3492
13	Васильев	3013	3419
14	Бородин	3075	3346
15	Зайцев	3110	3392
16	Червякова	3045	3350
17	Егоров	3042	3234
18	Волков	3024	3318
19	Петелин	3123	3390
20	Виноградов	3123	3443
21	Бортникова	1	NULL

Рис. 9: Таблица после вставки в таблицу «Услуги_на_приеме»

Doctor_id	Second_name	Services_number_on_appointment	Services_number_in_hospital
1	Залежнева	3079	3423
2	Соколова	3088	3347
3	Петрова	3157	3295
4	Лебедева	3037	3401
5	Сорокина	3153	3395
6	Петелин	3114	3397
7	Соловьев	3100	3397
8	Цаплина	3159	3386
9	Вареникова	3089	3338
10	Червякова	3114	3360
11	Соловьева	3151	3377
12	Степанов	3079	3492
13	Васильев	3013	3419
14	Бородин	3075	3346
15	Зайцев	3110	3392
16	Червякова	3045	3350
17	Егоров	3042	3234
18	Волков	3024	3318
19	Петелин	3123	3390
20	Виноградов	3123	3443
21	Бортникова	1	1

Рис. 10: Таблица после вставки в таблицу «Услуги_в_стационаре»

5 Лабораторная №3. Права пользователей

Формулировка задания: создать 2 пользователей:

- 1. Первый пользователь может только делать запросы ко view.
- 2. Второй пользователь должен иметь возможность совершать любые операции с данными во view: вставка, изменение, удаление.

Сначала создаются 2 пользователя без прав: cavy сможет только просматривать view, a viewer сможет вносить изменения во view.

```
CREATE USER 'cavy'
DEFAULT ROLE cavy;
CREATE USER 'viewer'
DEFAULT ROLE viewer;
```

Затем пользователи наделяются правами наделяются. Первый пользователь получает права SELECT для таблицы(view) appointments_number.

```
GRANT SELECT
ON TABLE appointments_number
TO 'cavy';
```

Второй пользователь, кроме прав SELECT из view, получает права на SELECT, INSERT, UPDATE и DELETE для таблиц, от которых зависит view. Эти таблицы — таблица врачей и таблицы услуг на приеме и в стационаре.

```
CREATE USER 'viewer',

DEFAULT ROLE viewer;

GRANT SELECT

ON TABLE vetclinic.appointments_number

TO 'viewer';

GRANT SELECT, UPDATE, DELETE, INSERT

ON TABLE vetclinic.appointment_services

TO 'viewer';

GRANT SELECT, UPDATE, DELETE, INSERT

ON TABLE vetclinic.hospital_services

TO 'viewer';

GRANT SELECT, UPDATE, DELETE, INSERT

ON TABLE vetclinic.doctors

TO 'viewer';
```

Теперь от имени **cavy** пробуем вставить значение в таблицу врачей. Попытка неудачна (рис. 11).

```
MySQL 127.0.0.1:33060+ ssl vetclinic SQL INSERT INTO doctors (Doctor_id, Second_name, First_name, Patronymic, Phone_number)
VALUE (21, 'Бортникова', 'Валерия', 'Витальевна', '+7(921)426-71-37');
ERROR: 1142: INSERT command denied to user 'cavy'@'localhost' for table 'doctors'
MySQL 127.0.0.1:33060+ ssl vetclinic SQL
```

Рис. 11: Попытка вставки в таблицу врачей пользователем сачу

Сначала просмотрим содержимое view от имени пользователя viewer (рис. 12).

	+	Normban	N
octor_id	Secona_name +	Number_ot_appointments +	Number_of_hospital_appointment
1	Залежнева	2653	267
2	Соколова	2677	263
3	Петрова	2725	262
4	Лебедева	2628	266
5	Сорокина	2720	267
6	Петелин	2685	267
7	Соловьев	2684	267
8	Цаплина	2736	264
9	Вареникова	2669	262
10	Червякова	2686	263
11	Соловьева	2730	266
12	Степанов	2653	275
13	Васильев	2611	266
14	Бородин	2634	261
15	Зайцев	2676	268
16	Червякова	2620	265
17	Егоров	2635	256
18	Волков	2616	262
19	Петелин	2700	263
20	Виноградов	2658	269

Рис. 12: Просмотр view пользователем viewer

Далее производим вставку от имени пользователя viewer (рис. 13).

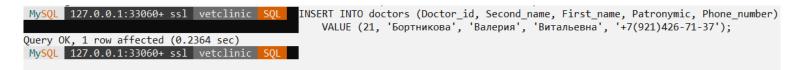


Рис. 13: Вставка в таблицу врачей пользователем viewer

Проверяем изменения от имени пользователя сауу(рис. 14).

Ooctor_id	Second_name	Number_of_appointments	Number_of_hospital_appointment
1	+ Залежнева	+ 2653	+ 267
2	Соколова	2677	263
3	Петрова	2725	262
4	Лебедева	2628	266
5	Сорокина	2720	267
6	Петелин	2685	267
7	Соловьев	2684	267
8	Цаплина	2736	264
9	Вареникова	2669	262
10	Червякова	2686	263
11	Соловьева	2730	266
12	Степанов	2653	275
13	Васильев	2611	266
14	Бородин	2634	261
15	Зайцев	2676	268
16	Червякова	2620	265
17	Егоров	2635	256
18	Волков	2616	262
19	Петелин	2700	263
20	Виноградов	2658	269
21	Бортникова	NULL	NUL

Рис. 14: Просмотр view пользователем саvy

6 Лабораторная №4. Транзакции

Формулировка задания: выбрать уровень изоляции транзакций и продемонстрировать проблему, которая присутствует или отсутствует на выбранном уровне изоляции.

В данной работе был выбран уровень изоляции READ COMMITTED и продемонстрировано отсутствие такого феномена, как «грязное чтение». На уровне READ COMMITTED изменения, которые производятся внутри транзакции видны только внутри этой транзакции: в другой транзакции или вне ее изменения будут не доступны, пока не будет сделан COMMIT. «Грязное чтение» — это феномен, при котором изменения внутри транзакции доступны еще до завершения транзакции.

Для демонстрации был совершен вход от двух пользователей и установлен уровень изоляции READ COMMITTED с помощью команды:

SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED

Также была осуществлена проверка того, что выбранный уровень изоляции был установлен. Результат приведен на рисунке 15.



Рис. 15: Задание уровня изоляции

Далее оба пользователя начинают транзакцию. Первый пользователь вставляет в таблицу Клиенты строку с помощью команды

INSERT INTO Clients (Client_id, Second_name, First_name, Patronymic, Phone_number)
VALUE (501, 'Алексеев', 'Петр', 'Владимирович', '+7(342)671-73-82')

Далее оба пользователя выбирают строки из таблицы Клиенты. На рисунке 16 приведен результат для первого пользователя, вставляющего строку, а на рисунке 17 — результат для второго пользователя.

MySQL 127.0 Ouery OK, 0 re	.0.1:33060+ ss		SQL START TR	ANSACTION;	
	.0.1:33060+ ss		(Client		irst_name, Patronymic, Phone_number) eтp', 'Владимирович', '+7(342)671-73-82');
Query OK, 1 re MySQL 127.0			SQL SELECT *	FROM Clients WHERE	Client_id > 490;
Client_id	Second_name	First_name	Patronymic	Phone_number	
491	Павлова	Екатерина	Борисовна	+7(027)420-84-57	
492	Федоров	Илья	Олегович	+7(715)830-67-22	
493	Работин	Валерий	Борисович	+7(851)467-24-65	
494	Ларина	Кристина	Николаевна	+7(708)363-04-74	
495	Захаров	Кирилл	Анатольевич	+7(138)657-08-33	
496	Сидорова	Марина	Андреевна	+7(772)473-65-75	
497	Табакова	Вера	Александровна	+7(852)083-23-66	
498	Степанова	Елена	Олеговна	+7(884)580-38-38	
499	Гусев	Георгий	Олегович	+7(078)355-07-50	
500	Кузнецов	Максим	Евгеньевич	+7(514)385-52-13	
j 501 j	Алексеев	Петр	Владимирович	+7(342)671-73-82	
++		·		+	-

Рис. 16: Вставка в таблицу Клиенты и выборка из нее

Query OK, 0 r	0.0.1:33060+ serows affected ((0.0005 sec)		ANSACTION; FROM Clients WHERE Client_id > 490
Client_id	Second_name	First_name	Patronymic	
491	Павлова	Екатерина	Борисовна	+ +7(027)420-84-57
492	Федоров	Илья	Олегович	+7(715)830-67-22
493	Работин	Валерий	Борисович	+7(851)467-24-65
494	Ларина	Кристина	Николаевна	+7(708)363-04-74
495	Захаров	Кирилл	Анатольевич	+7(138)657-08-33
496	Сидорова	Марина	Андреевна	+7(772)473-65-75
497	Табакова	Вера	Александровна	+7(852)083-23-66
498	Степанова	Елена	Олеговна	+7(884)580-38-38
499	Гусев	Георгий	Олегович	+7(078)355-07-50
500	Кузнецов	Максим	Евгеньевич	+7(514)385-52-13
10 rows in se	+ et (0 0023 sec)	++ \		++

Рис. 17: Выборка из таблицы Клиенты во второй транзакции

После того, как первый пользователь сделал COMMIT, второй пользователь может увидеть изменения в таблице Клиенты (рис. 18).

	et (0.0023 sec) 0.0.1:33060+ sec		SQL SELECT *	FROM Clients WHERE Client_id > 490;
Client_id	Second_name	First_name	Patronymic	Phone_number
491	Павлова	Екатерина	Борисовна	+7(027)420-84-57
492	Федоров	Илья	Олегович	+7(715)830-67-22
493	Работин	Валерий	Борисович	+7(851)467-24-65
494	Ларина	Кристина	Николаевна	+7(708)363-04-74
495	Захаров	Кирилл	Анатольевич	+7(138)657-08-33
496	Сидорова	Марина	Андреевна	+7(772)473-65-75
497	Табакова	Вера	Александровна	+7(852)083-23-66
498	Степанова	Елена	Олеговна	+7(884)580-38-38
499	Гусев	Георгий	Олегович	+7(078)355-07-50
500	Кузнецов	Максим	Евгеньевич	+7(514)385-52-13
501	Алексеев	Петр	Владимирович	+7(342)671-73-82
+	+	+	·	++
11 rows in se	et (0.0008 sec)		

Рис. 18: Выборка из таблицы Клиенты во второй транзакции после СОММІТ

7 Лабораторная №5. Процедура и функция

7.1 Процедура

Формулировка задания: вывести 2 таблицы: услуг на приеме и услуг в стационаре для определенного клиентов, проводившихся в указанном временном промежутке.

Вход процедуры: id клиента, 2 даты.

Выход: 2 соответствующие таблицы.

Процедура состоит из 2 запросов. В первом запросе объединяются таблицы Клиенты, Пациенты, Приемы, Услуги_на_приеме и Услуги. id_клиента выбирается из таблицы Клиенты, дата — из таблицы Приемы. Запрос выбирает наименование услуги и дату ее проведения. Во втором запросе объединяются таблицы Клиенты, Пациенты, Стационар, Услуги_в_стационаре и Услуги. id_клиента выбирается из таблицы Клиенты, дата — из таблицы Услуги_в_стационаре. Как и в предыдущем случае запрос выбирает наименование услуги и дату ее проведения.

Далее приведен скрипт создания процедуры:

```
delimiter //
CREATE PROCEDURE services_between_dates
     (IN cl_id INT, IN start_date DATE, IN finish_date DATE)
BEGIN
SELECT Service_name AS Service_on_appointment, Appointment_date AS Service_date
FROM (clients JOIN patients ON (clients.Client_id = patients.Owner_id)
  JOIN appointments ON (patients.Patient_id = appointments.Patient_id)
  JOIN appointment_services ON (appointments.Appointment_id =
                                appointment_services.Appointment_id)
  JOIN services ON (appointment_services.Service_id = services.Service_id) )
WHERE (clients.Client_id=cl_id AND Appointment_date >= start_date AND
        Appointment_date <= finish_date);
 SELECT Service_name AS Service_in_hospital, Service_date
FROM (clients JOIN patients ON (clients.Client_id = patients.Owner_id)
  JOIN hospital ON (patients.Patient_id = hospital.Patient_id)
  JOIN hospital_services ON (hospital.Hospital_id = hospital_services.Hospital_id)
  JOIN services ON (hospital_services.Service_id = services.Service_id) )
WHERE (clients.Client_id=cl_id AND Service_date >= start_date AND
        Service_date <= finish_date);</pre>
END//
```

На рисунке 19 приведен пример выполнения этой процедуры для клиента с id 10.



Рис. 19: Пример использования процедуры

7.2 Функция

Формулировка задания: функция получает на вход фамилию, имя и отчество и возвращает фамилию вместе с инициалами.

Скрипт создания функции приведен далее.

```
CREATE FUNCTION reduce(Second_name VARCHAR(30), First_name VARCHAR(30),

Patronymic VARCHAR(35))

RETURNS VARCHAR(36) DETERMINISTIC

RETURN CONCAT(Second_name, '', SUBSTRING(First_name, 1, 1), '.',

SUBSTRING(Patronymic, 1, 1), '.');
```

Для проверки функции был создан запрос к таблице Врачи. Скрипт этого запроса приведен далее.

Результат выполнения данного запроса приведен на рисунке 20.

Initials	Second_name	First_name	Patronymic
Романова К. Г.	Романова	 Кристина	Глебовна
Гусев Г. Б.	Гусев	Григорий	Борисович
Ларина Н. К.	Ларина	Надежда	Кирилловна
Егорова Е. В.	Егорова	Елизавета	Витальевна
Лебедева М. Г.	Лебедева	Маргарита	Григоревна
Павлова Е. В.	Павлова	Екатерина	Витальевна
Ошурков А. Г.	Ошурков	Артем	Георгиевич
Виноградова Е. А.	Виноградова	Екатерина	Александровна
Борисова В. Г.	Борисова	Варвара	Георгиевна
Шариков А. С.	Шариков	Алексей	Семенович
Демидова Е. С.	Демидова	Екатерина	Семеновна
Цаплина М. Е.	Цаплина	Маргарита	Егоровна
Ларин Д. Г.	Ларин	Дмитрий	Глебович
Николаев М. О.	Николаев	Максим	Олегович
Ларина Е. В.	Ларина	Елена	Витальевна
Воробьева К. Д.	Воробьева	Ксения	Дмитриевна
Назимов В. М.	Назимов	Виталий	Михайлович
Демидов В. Н.	Демидов	Виталий	Николаевич
Морозова С. В.	Морозова	Светлана	Викторовна
Павлов М. С.	Павлов	Максим	Семенович

Рис. 20: Пример использования функции

Заключение

В результате данной работы были выполнены следующие задания:

- создана view для подсчета приемов и числа приемов в стационаре для каждого врача;
- создан запрос, использующий view и показывающий число приемов и всех пациентов конкретного врача;
- создан триггер, позволяющий учитывать услуги врача по их количеству на приеме и в стационаре;
- создан пользователь, который может только просматривать view;
- создан пользователь, который может выполнять любые операции с данными view;
- продемонстрировано отсутствие феномена «грязного чтения» на уровне изоляции READ COMMITTED;
- создана процедура для вывода услуг на приеме и в стационаре, предоставленных определенному клиенту в определенный временной промежуток;
- создана функция для задания фамилии и инициалов.

Список литературы

[1] Документация по MySQL — https://dev.mysql.com/doc/ (последнее обращение: 11.12.2022)