

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт компьютерных наук и кибербезопасности
Высшая школа технологий искусственного интеллекта
Направление **02.03.01** : Математика и компьютерные науки

ОТЧЕТ
О выполнении лабораторных работ
по дисциплине «Теоретические основы баз данных»

Обучающийся: _____

Яшнова Дарья Михайловна
группа 5130201/20002

Руководитель: _____

Попов Сергей Геннадьевич

« ____ » _____ 2024г

Санкт-Петербург, 2024

Содержание

Введение	3
1 Постановка задачи	4
2 Описание предметной области	5
2.1 Описание особенностей конференции	5
2.2 Диаграмма базы данных на русском языке	5
3 Лабораторные работы	7
3.1 Работа 1: Создание представлений	7
3.2 Работа 2: Событийная модель, триггеры	8
3.3 Работа 3: Разграничение прав доступа	11
3.4 Работа 4: Создание функции и процедуры	13
3.5 Работа 5: Уровни изоляции транзакций	17
Заключение	19

Введение

В настоящем отчёте представлены результаты выполнения серии лабораторных работ, направленных на расширение функциональных возможностей базы данных школьной конференции. Эта база данных была разработана в рамках предыдущего семестрового курса «Теоретические основы баз данных».

В процессе выполнения лабораторных работ были изучены и реализованы в системе управления базами данных (СУБД) следующие элементы:

- представления;
- событийная модель (триггеры);
- права доступа;
- процедуры;
- функции;
- транзакции;

1 Постановка задачи

В рамках данного курса предусмотрено выполнение пяти лабораторных работ.

- Разработать представление, отображающее результаты запроса с подсчетом членов жюри и докладов в каждой секции. Подготовить запрос, использующий данное представление. Убедиться в невозможности изменения представления.
- Разработать триггеры для автоматизации сбора статистической информации о количестве членов жюри в каждой секции.

2 Описание предметной области

2.1 Описание особенностей конференции

Данная предметная область я буду рассматривается, на основе информации о конкретной конференции - Всероссийских школьных Харитоновских чтениях.

- Организатором конференции выступает Всероссийский научно-исследовательский институт экспериментальной физики;
- Конференция проводится в 2 этапа: прослушивание докладов в каждой секции, олимпиада.
- Конференция проводится по нескольким секциям: Физика, Математика, Биология, Русский язык, Химия, История. Количество секций, в которых может принимать участие один школьник не ограничено;
- Существует отборочный тур, после прохождения которого лучшие из участников отбираются на конференцию;
- Баллы за каждый этап усредняются и по ним выявляются участники, занявшие 1,2,3 место;
- Участниками конференции могут быть школьники с 7 по 11 класс;
- Каждый год на конференцию приезжают около 100 участников, 20 сопровождающих и 20 членов жюри со всей России;
- Проводится в закрытом городе, всем иногородним участникам оформляются временные пропуска;
- Для проведения конференции используются помещения одной из городских школ;

Рассмотрим подробнее процесс формирования расписания конференции для всех участников:

- В конференции участвует участник и члены жюри. У каждого участника есть сопровождающий, причем один сопровождающий может сопровождать несколько участников;
- Участники и члены жюри делятся на секции. Секция включает в себя олимпиаду и доклад. Участники в каждой секции пишут олимпиаду и делают доклад. При этом можно рассказывать несколько докладов, но олимпиада может быть только одна;
- Сопровождающий регистрируется на секции, чтобы следить за порядком и слушать доклады. Члены жюри оценивают секцию, причем для одной секции может быть несколько членов жюри. Также члены жюри проверяют олимпиаду, несколько членов жюри могут проверять одну олимпиаду.

2.2 Диаграмма базы данных на русском языке

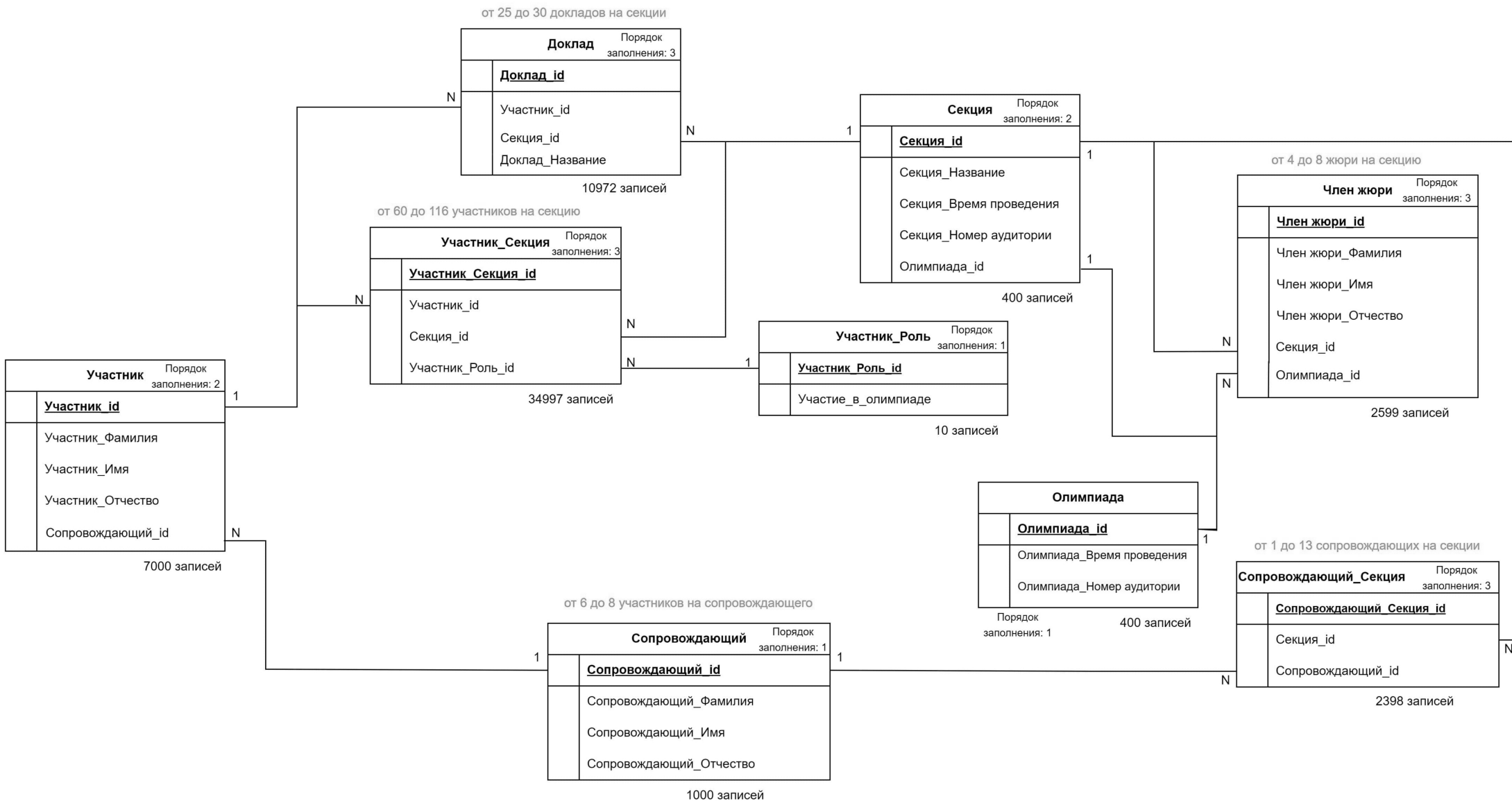


Рис.1 Диаграмма базы данных на русском языке

3 Лабораторные работы

3.1 Работа 1: Создание представлений

Задача: Разработать представление для хранения запроса внутри СУБД

Формулировка задачи: Для каждой секции посчитать количество докладов и количество членов жюри.

Сделано представление sect_mem_rep_stat с посчитанным количеством докладов и членов жюри для каждой секции. Код создания представлен на рис.2.

```
1 CREATE VIEW sect_mem_rep_stat AS select Section_Name, mem_num, rep_num
   from(
2 (
3     select Section_id as section_, count(Jury_Member_id) as mem_num from
   jury_member
4     group by Section_id) as jm
5     join
6     (select Section_id, count(Report_id) as rep_num from report
7     group by Section_id)as rep on rep.Section_id=jm.section_
8     join Section on Section.Section_id = jm.section_
9 )
```

Рис. 2: Создание пользователей и разграничение их прав доступа

Представление создаёт таблицу с колонками Section_Name, rep_num и mem_num. Результат представлен на Рис. 3.

	Section_Name	mem_num	rep_num
►	IQPUINHBTZSKPHOQZIPHKYIOIYOIYOKGRBKY...	5	27
	ZYFRELSUJFWLDQSUQSBBISWVAZNTNPBAFPY	6	28
	DGTIYLCRQDSEMQLBZEABWIIIMYZBBM	7	27
	XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8	28
	HSDB	5	28
	GHDCBFLKNEMXLDE	6	25
	IANXSZWLNONRRQEBM	7	30
	KYXLZMDGEAXMFF	8	26
	LBLUACYOSKFUQTHMUVXCZZLUJKWSAXIJDNI	5	30
	HVFK	6	26
	WDXGSCEXDJTDVXSUWKDDAGSSIKYSYVCWWI...	7	28
	PPMBIKAYGYNPOQSFGHFUI	8	26
	EWIONNDGNLDWF	5	26
	OSHIXKNFNLAYTUXZLMPXFYFVFBFD	6	26
	UKNJXBRLKMGQJSGVVIXNHYSKPZILXWWWYZUM...	7	26
	IWXPOPJO	8	25

Рис. 3: Первые строки вывода представления

Представление используется в следующем запросе на рис.4, в котором добавляется информация о номере аудитории. Результат выполнения запроса с представлением представлен на Рис.5.

```
1 SELECT s.Section_Name, mem_num, rep_num, Section_Audience_Number
```

```

2 FROM Section s
3 JOIN sect_mem_rep_stat sms ON s.Section_Name = sms.Section_Name;

```

Рис. 4: Результат выполнения запроса с представлением

	Section_Name	mem_num	rep_num	Section_Audience_Number
►	IQPUINHBTZSKPHOQZIPHKYIOIYOIYOKGRBK...	5	27	998
	ZYFRELSUJFWLDQSUQSBBSWVAZNTNPBAFPY...	6	28	783
	DGTIYLCRQDSEMLBZEABWIIIMYZBBM	7	27	752
	XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8	28	832
	HSDB	5	28	990
	GHDCBFLKNEMXLDE	6	25	764
	IANXSZWLYNONRRQEBM	7	30	957
	KYXLZMDGEAXMFF	8	26	988
	LBLUACYOSKFUQTHMUVXCZZLUJKWSAXIJDNIG	5	30	839
	HVFK	6	26	964
	WDXGSCXDJTDVXSUWKDDAGSSIKYSYVCWWI...	7	28	874

Рис. 5: Результат выполнения запроса с представлением

3.2 Работа 2: Событийная модель, триггеры

Задача: Разработать триггер, производящий манипуляции над БД при добавлении, удалении и обновлении данных.

Формулировка задачи: Необходимо собирать статистику о числе членов жюри в одной секции. Событийная модель должна поддерживать актуальность данных в таблицах со статистикой при изменении данных.

В листинге на рис.6 создается и заполняется таблица для работы с триггерами. Результат создания таблицы для триггера представлен на Рис.7.

```

1 CREATE TABLE 'mydb'.'trig_tbl' (
2     'Olympiad_id' INT NOT NULL,
3     'Olympiad_Name' VARCHAR(45) NULL,
4     'Jmem_num' INT NULL,
5     PRIMARY KEY ('Olympiad_id'));
6
7 INSERT INTO trig_tbl (Olympiad_id, Olympiad_Name, Jmem_num)
8 SELECT Section.Section_id AS Olympiad_id,
9        Section.Section_Name AS Olympiad_Name,
10       COUNT(Jury_Member_id) AS Jmem_num
11 FROM jury_member
12 JOIN Section
13      ON Section.Section_id = jury_member.Section_id
14 GROUP BY Section.Section_id, Section.Section_Name;

```

Рис. 6: Результат создания таблицы для триггера

	Olympiad_id	Olympiad_Name	Jmem_num
►	1	IQPUINHBTZSKPHOQZIPHKYOIYOIYOKGRBKY...	5
	2	ZYFRELSUJFWLDQSUQSBBISWVAZNTNPBAFPY	8
	3	DGTIYLCRQDSEMLBZEABWIIMYZBBM	7
	4	XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8
	5	HSDB	5
	6	GHDCBFLKNEMXLDE	6
	7	IANXSZWLYNONRRQEBM	7
	8	KYXLZMDGEAXMFF	8
	9	LBLUACYOSKFUQTHMUVXCZZLUJKWSAXIJDNIG	5
	10	HVFK	6
	11	WDXGSCEXDJTDVXSUWKDDAGSSIKYSYVCWWI...	7

Рис. 7: Результат создания таблицы для триггера

В листинге на рис.8 создается триггер inst_trigger, который выполняется после вставки каждой новой записи в таблицу jury_member. Триггер обновляет таблицу trig_tbl, увеличивая значение Jmem_num на 1 для строк, где Olympiad_id совпадает со значением NEW.Olympiad_id (значение Olympiad_id в только что добавленной записи)

```

1 delimiter $$
2 CREATE TRIGGER inst_trigger AFTER INSERT ON jury_member
3   FOR EACH ROW BEGIN
4     UPDATE trig_tbl
5     SET Jmem_num = Jmem_num + 1
6     WHERE trig_tbl.Olympiad_id = NEW.Olympiad_id;
7   END $$
8 delimiter ;

```

Рис. 8: Код создания триггера вставки

Код из рис.9 создает триггер в базе данных, который срабатывает после удаления записи из таблицы jury_member. Когда запись удаляется, триггер обновляет таблицу trig_tbl, уменьшая значение столбца Jmem_num на 1 для строки, где Olympiad_id соответствует значению Olympiad_id удаленной записи

```

1 delimiter $$
2 CREATE TRIGGER del_trigger AFTER DELETE ON jury_member
3   FOR EACH ROW BEGIN
4     UPDATE trig_tbl tt
5     SET Jmem_num = Jmem_num - 1
6     WHERE tt.Olympiad_id = OLD.Olympiad_id;
7   END $$
8 delimiter ;

```

Рис. 9: Код создания триггера удаления

Код из рис.10 создает триггер upd_trigger в базе данных, который срабатывает после обновления записи в таблице jury_member. Триггер изменяет количество членов жюри (Jmem_num) в таблице trig_tbl, уменьшая значение на 1 для старого идентификатора Олимпиады (OLD.Olympiad_id) и увеличивая на 1 для нового идентификатора Олимпиады (NEW.Olympiad_id).

```

1 delimiter $$
2 CREATE TRIGGER upd_trigger AFTER UPDATE ON jury_member
3     FOR EACH ROW BEGIN
4         UPDATE trig_tbl
5         SET Jmem_num = Jmem_num - 1
6         WHERE trig_tbl.Olympiad_id = OLD.Olympiad_id;
7         UPDATE trig_tbl
8         SET Jmem_num = Jmem_num + 1
9         WHERE trig_tbl.Olympiad_id = NEW.Olympiad_id;
10    END $$
11 delimiter ;

```

Рис. 10: Код создания триггера обновления

После выполнения запроса на удаление из таблицы jury_member из листинга на рис.11 количество членов жюри в первой секции уменьшилось на 1. Таблица после удаления представлена на Рис.12.

```

1 delete from jury_member where Jury_Member_id=2602

```

Рис. 11: Запрос на удаление

	Olympiad_id	Olympiad_Name	Jmem_num
►	1	IQPUINHBTZSKPHOQZIPHKYOIYOIYOKGRBKY...	4
	2	ZYFRELSUJFWLDQSUQSBBISWVAZNTNPBAFPY	8
	3	DGTIYLCRQDSEMQLBZEABWIMYZBBM	7
	4	XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8

Рис. 12: Таблица после удаления

После выполнения запроса (рис.13) на добавление в таблицу увеличивается количество членов жюри на 1. Таблица после добавления представлена на Рис.14.

```

1 insert INTO jury_member
2     (Jury_Member_id,
3     Jury_Member_Surname,
4     Jury_Member_Name,
5     Jury_Member_Patronymic,
6     Section_id,
7     Olympiad_id)
8     VALUES (2602, "Andreev", "Andrey", "Andreevich", 1, 1);

```

Рис. 13: Запрос на вставку

	Olympiad_id	Olympiad_Name	Jmem_num
▶	1	IQPUINHBTZSKPHOQZIPHKYOIYOIYOKGRBKY...	4
	2	ZYFRELSUJFWLDQSUQSBBISWVAZNTNPBAFPY EY	8
	3	DGTIYLCRQDSEMQLBZEABWIIMYZBBM	7
	4	XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8

Рис. 14: Таблица после добавления

После выполнения запроса на изменение в одной секции (рис.15) количество членов жюри увеличивается на 1, а в другой уменьшается. Результат выполнения запроса на изменение таблицы представлен на Рис.16.

```
1 update jury_member set Olympiad_id = 2 where Jury_Member_id=2506
```

Рис. 15: Запрос на изменение

	Olympiad_id	Olympiad_Name	Jmem_num
▶	1	IQPUINHBTZSKPHOQZIPHKYOIYOIYOKGRBKY...	4
	2	ZYFRELSUJFWLDQSUQSBBISWVAZNTNPBAFPY EY	9
	3	DGTIYLCRQDSEMQLBZEABWIIMYZBBM	6
	4	XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8

Рис. 16: Результат выполнения запроса на изменение таблицы

3.3 Работа 3: Разграничение прав доступа

Необходимо организовать работу двух пользователей: guinea и master. Пользователь guinea сможет только читать представление, а master получит возможность не только читать представление, но и редактировать таблицы, которые в него входят.

Для реализации этого необходимо создать двух пользователей — guinea и master. Затем нужно предоставить пользователю guinea права на просмотр представления, а пользователю master — права на просмотр представления и редактирование определённых таблиц.

Процесс создания пользователей и назначения им прав продемонстрирован на рис.17. Сравнение прав доступа пользователей представлено в таблице 1.

```
1 CREATE USER 'guinea'@'localhost' IDENTIFIED BY 'oinkoink';
2 GRANT SELECT ON sect_mem_rep_stat TO 'guinea'@'localhost';
3
4 CREATE USER 'master'@'localhost' IDENTIFIED BY 'master';
5 GRANT SELECT ON sect_mem_rep_stat TO 'master'@'localhost';
6 GRANT SELECT, INSERT, DELETE, UPDATE
7     ON jury_member TO 'master'@'localhost';
8 GRANT SELECT, INSERT, DELETE, UPDATE
9     ON report TO 'master'@'localhost';
```

Рис. 17: Код создания пользователей

В таблице 1 представлены результаты выполнения различных запросов от лица guinea и master.

Таблица 1: Сравнение прав пользователей

№	guinea	master																																				
1	Запрос на выборку из представления Запрос выполняет выборку 5 записей из таблицы: <i>SELECT * FROM sect_mem_rep_stat limit 5</i>																																					
	<pre>mysql> select * from sect_mem_rep_stat limit 5;</pre> <table><thead><tr><th>Section_Name</th><th>mem_num</th><th>rep_num</th></tr></thead><tbody><tr><td>IQPUINHBTZSKPHOQZIPHKYIOIYOKGRBKYQWZ</td><td>4</td><td>27</td></tr><tr><td>ZYFRELSUJFWLDQSUQSBBSISWVAZNTNPBAFPYEY</td><td>6</td><td>28</td></tr><tr><td>DGTIYLCRODSEMOQLBZEABWIMYZBBM</td><td>7</td><td>27</td></tr><tr><td>XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY</td><td>8</td><td>28</td></tr><tr><td>HSDB</td><td>5</td><td>31</td></tr></tbody></table> <p>5 rows in set (0.01 sec)</p>	Section_Name	mem_num	rep_num	IQPUINHBTZSKPHOQZIPHKYIOIYOKGRBKYQWZ	4	27	ZYFRELSUJFWLDQSUQSBBSISWVAZNTNPBAFPYEY	6	28	DGTIYLCRODSEMOQLBZEABWIMYZBBM	7	27	XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8	28	HSDB	5	31	<pre>mysql> select * from sect_mem_rep_stat limit 5;</pre> <table><thead><tr><th>Section_Name</th><th>mem_num</th><th>rep_num</th></tr></thead><tbody><tr><td>IQPUINHBTZSKPHOQZIPHKYIOIYOKGRBKYQWZ</td><td>4</td><td>27</td></tr><tr><td>ZYFRELSUJFWLDQSUQSBBSISWVAZNTNPBAFPYEY</td><td>6</td><td>28</td></tr><tr><td>DGTIYLCRODSEMOQLBZEABWIMYZBBM</td><td>7</td><td>27</td></tr><tr><td>XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY</td><td>8</td><td>28</td></tr><tr><td>HSDB</td><td>5</td><td>31</td></tr></tbody></table> <p>5 rows in set (0.01 sec)</p>	Section_Name	mem_num	rep_num	IQPUINHBTZSKPHOQZIPHKYIOIYOKGRBKYQWZ	4	27	ZYFRELSUJFWLDQSUQSBBSISWVAZNTNPBAFPYEY	6	28	DGTIYLCRODSEMOQLBZEABWIMYZBBM	7	27	XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8	28	HSDB	5	31
Section_Name	mem_num	rep_num																																				
IQPUINHBTZSKPHOQZIPHKYIOIYOKGRBKYQWZ	4	27																																				
ZYFRELSUJFWLDQSUQSBBSISWVAZNTNPBAFPYEY	6	28																																				
DGTIYLCRODSEMOQLBZEABWIMYZBBM	7	27																																				
XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8	28																																				
HSDB	5	31																																				
Section_Name	mem_num	rep_num																																				
IQPUINHBTZSKPHOQZIPHKYIOIYOKGRBKYQWZ	4	27																																				
ZYFRELSUJFWLDQSUQSBBSISWVAZNTNPBAFPYEY	6	28																																				
DGTIYLCRODSEMOQLBZEABWIMYZBBM	7	27																																				
XMESHWJPNCBWSXXSWTKKXBFVTRFIXGECFDKLY	8	28																																				
HSDB	5	31																																				
2	Запрос на выборку из таблицы jury_member Выбрать все столбцы из таблицы jury_member и вернуть только первые 5 строк <i>SELECT * FROM jury_member limit 5</i>																																					
	<pre>mysql> SELECT * FROM jury_member;</pre> <p>ERROR 1142 (42000): SELECT command denied to user 'guinea'@'localhost' for table 'jury_member'</p>	<pre>mysql> SELECT * FROM jury_member limit 5;</pre> <table><thead><tr><th>Jury_Member_id</th><th>Jury_Member_Surname</th><th>Jury_Member_Name</th></tr></thead><tbody><tr><td>2</td><td>HWBGFKEPWHQZFVX</td><td>IRBFICUMOOKMCTDMCJAQUFQVZIVKNUFP</td></tr><tr><td>3</td><td>HWBKGGBGSHKQP</td><td>JFRFQVQKASFSFUDQJONV</td></tr><tr><td>4</td><td>NIXEYSPSTSEU</td><td>IDQZMHSJLMLJQREPLJZYPMHUSJDBUEWYH</td></tr><tr><td>5</td><td>NG</td><td>CTMECKEPELBNPMLVZJKEHRIEFBYPUL</td></tr><tr><td>6</td><td>HRPX</td><td>VLBYGOYMFJQSPS</td></tr></tbody></table> <p>5 rows in set (0.00 sec)</p>	Jury_Member_id	Jury_Member_Surname	Jury_Member_Name	2	HWBGFKEPWHQZFVX	IRBFICUMOOKMCTDMCJAQUFQVZIVKNUFP	3	HWBKGGBGSHKQP	JFRFQVQKASFSFUDQJONV	4	NIXEYSPSTSEU	IDQZMHSJLMLJQREPLJZYPMHUSJDBUEWYH	5	NG	CTMECKEPELBNPMLVZJKEHRIEFBYPUL	6	HRPX	VLBYGOYMFJQSPS																		
Jury_Member_id	Jury_Member_Surname	Jury_Member_Name																																				
2	HWBGFKEPWHQZFVX	IRBFICUMOOKMCTDMCJAQUFQVZIVKNUFP																																				
3	HWBKGGBGSHKQP	JFRFQVQKASFSFUDQJONV																																				
4	NIXEYSPSTSEU	IDQZMHSJLMLJQREPLJZYPMHUSJDBUEWYH																																				
5	NG	CTMECKEPELBNPMLVZJKEHRIEFBYPUL																																				
6	HRPX	VLBYGOYMFJQSPS																																				
3	Запрос на обновление таблицы jury_member Обновить данные в таблице jury_member. Изменить значение столбца Olympiad_id на 100, где Jury_Member_id равен 12 <i>UPDATE jury_member set Olympiad_id = 100 where Jury_Member_id = 12</i>																																					
	<pre>mysql> UPDATE jury_member set Olympiad_id = 100 where Jury_Member_id = 12;</pre> <p>ERROR 1142 (42000): UPDATE command denied to user 'guinea'@'localhost' for table 'jury_member'</p>	<pre>mysql> UPDATE jury_member set Olympiad_id = 100 where Jury_Member_id = 12;</pre> <p>Query OK, 0 rows affected (0.06 sec) Rows matched: 1 Changed: 0 Warnings: 0</p>																																				
	Проверка запроса обновления таблицы. <i>select Olympiad_id from jury_member where Jury_Member_id=12;</i>																																					
	<table><thead><tr><th>Olympiad_id</th></tr></thead><tbody><tr><td>233</td></tr></tbody></table>	Olympiad_id	233	<table><thead><tr><th>Olympiad_id</th></tr></thead><tbody><tr><td>100</td></tr></tbody></table>	Olympiad_id	100																																
Olympiad_id																																						
233																																						
Olympiad_id																																						
100																																						
4	Запрос на удаление из таблицы jury_member Удалить из таблицы jury_member все строки, где Jury_Member_id равен 2602 <i>DELETE from jury_member where Jury_Member_id = 2602</i>																																					
	<pre>mysql> DELETE from jury_member where Jury_Member_id = 2602;</pre> <p>ERROR 1142 (42000): DELETE command denied to user 'guinea'@'localhost' for table 'jury_member'</p>	<pre>mysql> DELETE from jury_member where Jury_Member_id = 2600;</pre> <p>Query OK, 1 row affected (0.01 sec)</p>																																				
	Проверка выполнения запроса: выбираем из таблицы jury_member члена жюри с id = 2602 <i>select * from jury_member where Jury_Member_id=2602;</i>																																					
	<table><thead><tr><th>Jury_Member_id</th><th>Jury_Member_S</th><th>Jury_Mem</th><th>Jury_Member_S</th><th>Section_id</th><th>Olympiad_id</th></tr></thead><tbody><tr><td>2602</td><td>Valul</td><td>Uliana</td><td>19</td><td>19</td><td></td></tr></tbody></table>	Jury_Member_id	Jury_Member_S	Jury_Mem	Jury_Member_S	Section_id	Olympiad_id	2602	Valul	Uliana	19	19		<table><thead><tr><th>Jury_Member_id</th><th>Jury_Member_S</th><th>Jury_Mem</th><th>Jury_Member_S</th><th>Section_id</th><th>Olympiad_id</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>	Jury_Member_id	Jury_Member_S	Jury_Mem	Jury_Member_S	Section_id	Olympiad_id																		
Jury_Member_id	Jury_Member_S	Jury_Mem	Jury_Member_S	Section_id	Olympiad_id																																	
2602	Valul	Uliana	19	19																																		
Jury_Member_id	Jury_Member_S	Jury_Mem	Jury_Member_S	Section_id	Olympiad_id																																	

5	Запрос на вставку в таблицу jury_member																			
	Вставить в таблицу jury_member новую строку с указанными значениями insert INTO jury_member (Jury_Member_id, Jury_Member_Surname, Jury_Member_Name, Jury_Member_Patronymic, Section_id, Olympiad_id) VALUES (2602, «Andreev», «Andrey», «Andreevich», 1, 1);																			
	<pre>mysql> insert INTO jury_member(Jury_Member_id, Jury_Member_Surname, Jury_Member_Name, Jury_Member_Patronymic, Section_id, Olympiad_id)VALUES (2602, "Andreev", "Andrey", "Andreevich", 1, 1); ERROR 1142 (42000): INSERT command denied to user 'guinea'@'localhost' for table 'jury_member'</pre>	<pre>mysql> insert INTO jury_member(Jury_Member_id, Jury_Member_Surname, Jury_Member_Name, Jury_Member_Patronymic, Section_id, Olympiad_id)VALUES (2600, "Andreev", "Andrey", "Andreevich", 1, 1); Query OK, 1 row affected (0.01 sec)</pre>																		
	Вставить в таблицу jury_member новую строку с указанными значениями insert INTO jury_member (Jury_Member_id, Jury_Member_Surname, Jury_Member_Name, Jury_Member_Patronymic, Section_id, Olympiad_id) VALUES (2602, «Andreev», «Andrey», «Andreevich», 1, 1);																			
	<pre>mysql> DELETE from jury_member where Jury_Member_id = 2600; Query OK, 1 row affected (0.01 sec)</pre>	<table><thead><tr><th>Jury_Member_id</th><th>Jury_Member_Surname</th><th>Jury_Member_Name</th><th>Jury_Member_Patronymic</th><th>Section_id</th><th>Olympiad_id</th></tr></thead><tbody><tr><td>2602</td><td>Andreev</td><td>Andrey</td><td>Andreevich</td><td>1</td><td>1</td></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></tbody></table>	Jury_Member_id	Jury_Member_Surname	Jury_Member_Name	Jury_Member_Patronymic	Section_id	Olympiad_id	2602	Andreev	Andrey	Andreevich	1	1	*	NULL	NULL	NULL	NULL	NULL
Jury_Member_id	Jury_Member_Surname	Jury_Member_Name	Jury_Member_Patronymic	Section_id	Olympiad_id															
2602	Andreev	Andrey	Andreevich	1	1															
*	NULL	NULL	NULL	NULL	NULL															
6	Запрос на удаление таблицы jury_member																			
	Удалить таблицу jury_member из базы данных DROP table jury_member																			
	<pre>mysql> DROP table jury_member; ERROR 1142 (42000): DROP command denied to user 'guinea'@'localhost' for table 'jury_member'</pre>	<pre>mysql> DROP table jury_member; ERROR 1142 (42000): DROP command denied to user 'guinea'@'localhost' for table 'jury_member'</pre>																		

3.4 Работа 4: Создание функции и процедуры

Задача: реализовать функцию, принимающую в качестве аргументов фамилию, имя и отчество и возвращающая строку формата «Фамилия.И.О.», если у человека есть отчество, и строку формата «Фамилия.И.», если нет отчества.

```

1
2 DELIMITER //
3
4 CREATE FUNCTION format_name(
5     Psurname VARCHAR(45),
6     Pname VARCHAR(45),
7     Ppatronymic VARCHAR(45)
8 )
9 RETURNS VARCHAR(50)
10 DETERMINISTIC
11 BEGIN
12     DECLARE formatted_name VARCHAR(50);
13     IF Ppatronymic IS NOT NULL THEN
14         SET formatted_name = CONCAT(Psurname, '. ', LEFT(Pname, 1), '. ', LEFT(
15             Ppatronymic, 1), '.');
16     ELSE
17         SET formatted_name = CONCAT(Psurname, '. ', LEFT(Pname, 1), '.');
18     END IF;
19     RETURN formatted_name;
20 END //
21 DELIMITER ;

```

Рис. 18: Реализация функции

```

1     select Jury_Member_Surname, Jury_Member_Name, Jury_Member_Patronymic,
2     format_name(Jury_Member_Surname, Jury_Member_Name,
3     Jury_Member_Patronymic)

```

```
2 from jury_member where Jury_Member_id>2599
```

Рис. 19: Пример использования функции

	Jury_Member_Surname	Jury_Member_Name	Jury_Member_Patronymic	format_name(Jury_Member_Surname, Jury_Member_Name, Jury_Member_Patronymic)
▶	Andreev	Andrey	Andreevich	Andreev. A. A.
	shhhh	xuxa	NULL	shhhh. x.
	Vakul	Uliana	NULL	Vakul. U.

Рис. 20: Результат использования функции

Если вызвать функцию для неверной таблицы, она не будет выполняться (рис.21-22).

```
1 select Jury_Member_Name, Jury_Member_Surname, format_name(  
Jury_Member_Name, Jury_Member_Surname, Jury_Member_Patronymic) as fn  
from report
```

Рис. 21: Пример некорректного использования функции

Error Code: 1054. Unknown column 'Jury_Member_Name' in 'field list'

Рис. 22: Некорректный вызов функции

Задача: Реализовать процедуру, которая будет принимать значения фамилии, имени и отчества участника, фамилии, имени и отчества сопровождающего, название секции и название доклада. Если участника нет в таблице participant нужно добавить его и его доклад в секцию, иначе просто добавить доклад в секцию. Если сопровождающего нет в таблице charerone его также нужно добавить в эту таблицу, при условии, что за участником уже не закреплен другой сопровождающий.

```
1  
2 select * from participant  
3 drop procedure AddParticipantAndReport  
4 call AddParticipantAndReport("Ulianna","Shhhhhh","Alexeevna" , "Aboba", "  
Abobov",null, "HVFK", "VERY Important Report")  
5 DELIMITER //  
6 CREATE PROCEDURE AddParticipantAndReport (  
7     IN PName VARCHAR(45),  
8     IN PSurname VARCHAR(45),  
9     IN PPatronymic VARCHAR(45) ,  
10    IN CName VARCHAR(45),  
11    IN CSurname VARCHAR(45),  
12    IN CPatronymic VARCHAR(45) ,  
13    IN SectionName VARCHAR(45),  
14    IN ReportTitle VARCHAR(100)  
15 )  
16 BEGIN  
17     declare pt int default null;  
18     declare ch int default null;  
19     declare sec int default null;  
20     if (PName is null or PSurname is null or CName is null or SectionName is  
21         null or ReportTitle is null) then  
22         SIGNAL SQLSTATE '23000'  
23         SET MESSAGE_TEXT = 'PName, PSurname, CName,SectionName,  
24         ReportTitle cant be null.';
```

```

23 end if;
24
25 select Section_id into sec from section
26     where Section_Name = SectionName;
27 select Participant_id into pt from participant
28     where Participant_Surname = PSurname
29     and Participant_Name = PName
30     and (( Participant_Patronymic is null and PPatronymic is null)
31         or Participant_Patronymic = PPatronymic);
32 select Chaperone_id into ch from chaperone
33     where Chaperone_Surname = CSurname
34     and Chaperone_Name = CName
35     and (( Chaperone_Patronymic is null and CPatronymic is null)
36         or Chaperone_Patronymic = CPatronymic);
37 if not sec is null then
38     if not pt is null then
39         select Chaperone_id into ch from participant where Participant_id =
40             pt;
41     end if;
42     if pt is null then
43         if ch is null then
44             insert into chaperone (Chaperone_Surname, Chaperone_Name,
45                 Chaperone_Patronymic)
46                 values (CSurname, CName, CPatronymic);
47
48             select Chaperone_id into ch from chaperone
49                 where Chaperone_Surname = CSurname
50                 and Chaperone_Name = CName
51                 and (( Chaperone_Patronymic is null and CPatronymic is
52                     null)
53                     or Chaperone_Patronymic = CPatronymic);
54             end if;
55             insert into participant (Participant_Surname,
56                 Participant_Name,
57                 Participant_Patronymic,
58                 Chaperone_id)
59                 values (PSurname, PName, PPatronymic, ch);
60
61         end if;
62         select Participant_id into pt from participant
63             where Participant_Surname = PSurname
64             and Participant_Name = PName
65             and (( Participant_Patronymic is null and PPatronymic is null)
66                 or Participant_Patronymic = PPatronymic);
67         insert into participant_section ( Participant_id, Section_id,
68             Participant_Role_id)
69             values(pt, sec, 1);
70         insert into report (Participant_id,Section_id,Report_name)
71             values (pt, sec, ReportTitle);
72     else
73         SIGNAL SQLSTATE 'HY000'
74         SET MESSAGE_TEXT = 'No such section.';
75     end if;
76 END //
77 DELIMITER ;

```

Рис. 23: Реализация процедуры


```

1 call AddParticipantAndReport("AAAa","BBBb", "CCCc", "XXXx","YYYy","PPPp",
   "HSDB","Important Report")
2
3 call AddParticipantAndReport("AAAa","BBBb", null, "XXXx","YYYy","PPPp", "
   HSDB","Important Report")
4
5 call AddParticipantAndReport("Ulianna","Shhhhh","Alexeevna" , "Aboba", "
   Abobov",null, "HVFK", "VERY Important Report")

```

Рис. 24: Примеры вызова процедуры

	Participant_id	Participant_Surname	Participant_Name	Participant_Patron	Chaperone_id
►	7007	Shhhhh	Ulianna	Alexeevna	2003
	7006	BBBb	AAAa	NULL	2002
	7005	BBBb	AAAa	CCCc	2002

Рис. 25: Результат выполнения трех вызовов процедуры. Изменения в таблице participant.

	Participant_Section_id	Participant_id	Section_id	Participant_Role_id
►	35003	7007	10	1
	35002	7006	5	1
	35001	7005	5	1

Рис. 26: Результат выполнения трех вызовов процедуры. Изменения в таблице participant_section.

	Chaperone_id	Chaperone_Surname	Chaperone_Name	Chaperone_Patronymic
►	2003	Abobov	Aboba	NULL
	2002	YYYy	XXXx	PPPp

Рис. 27: Результат выполнения трех вызовов процедуры. Изменения в таблице chaperone.

	Report_id	Participant_id	Section_id	Report_name
	10978	7007	10	VERY Important Report
	10977	7006	5	Important Report2
	10976	7005	5	Important Report

Рис. 28: Результат выполнения трех вызовов процедуры. Изменения в таблице report.

При вызове процедуры с ошибкой, например с null вместо обязательных аргументов, как в коде на рис.27 возникает ошибка (рис.28).

```

1 call AddParticipantAndReport
2 (null,"vvv",null,"aaaa","aaaa","aaaa","aaaa","aaaaa")

```

Рис. 29: Вызов процедуры с неверными аргументами

Error Code: 1644. PName, PSurname, CName, SectionName, ReportTitle cant be null.

Рис. 30: Ошибка при неверном вызове процедуры

3.5 Работа 5: Уровни изоляции транзакций

Задача: Необходимо установить уровень изоляции транзакций, соответствующий стандарту повторяемого чтения (repeatable read).

Repeatable read (повторяющееся чтение) Уровень, при котором читающая транзакция «не видит» изменения данных, которые были ею ранее прочитаны. При этом никакая другая транзакция не может изменять данные, читаемые текущей транзакцией, пока та не окончена.

Блокировки в разделяющем режиме применяются ко всем данным, считываемым любой инструкцией транзакции, и сохраняются до её завершения. Это запрещает другим транзакциям изменять строки, которые были считаны незавершённой транзакцией. Однако другие транзакции могут вставлять новые строки, соответствующие условиям поиска инструкций, содержащихся в текущей транзакции. При повторном запуске инструкции текущей транзакцией будут извлечены новые строки, что приведёт к фантомному чтению. В таблице 2 представлен пример появления фантома.

Таблица 2: Пример выполнения транзакций при уровне изоляции repeatable read

№	user1	user2
0	Установка уровня изоляции для транзакций	
	SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;	
1	Оба пользователя начинают транзакцию	
	start transaction;	
	<pre>mysql> start transaction; Query OK, 0 rows affected (0.00 sec)</pre>	<pre>mysql> start transaction; Query OK, 0 rows affected (0.00 sec)</pre>
2	Второй пользователь выполняет запрос на выборку из таблицы jury_member	
		SELECT * FROM jury_member where Jury_Member_id>2599
		<pre>mysql> select * from jury_member where Jury_Member_id > 2599; +-----+-----+-----+-----+-----+-----+ Jury_Member_id Jury_Member_Surname Jury_Member_Name Jury_Member_Patronymic Section_id Olympiad_id +-----+-----+-----+-----+-----+-----+ 2600 Andreev Andrey Andreevich 1 1 2602 Andreev Andrey Andreevich 1 1 2604 shhh xuxa NULL 10 10 +-----+-----+-----+-----+-----+-----+ </pre>
3	Первый пользователь делает вставку в таблицу jury_member	
	insert into jury_member (Jury_Member_Surname, Jury_Member_Name, Jury_Member_Patronymic, Section_id, Olympiad_id) values («Ivanov», «Ivan», «Ivanovich», 100, 100);	
	<pre>mysql> insert into jury_member (Jury_Member_Surname, Jury_Member_Name, Jury_Member_Patronymic, Section_id, Olympiad_id) values ("Ivanov", "Iva n", "Ivanovich", 100,100); Query OK, 1 row affected (0.05 sec)</pre>	
4	Первый пользователь сохраняет изменения в базу данных	
	commit;	
	<pre>mysql> commit; Query OK, 0 rows affected (0.01 sec)</pre>	
5	Второй пользователь выполняет запрос на выборку из таблицы jury_member и читает фантом, в котором не отображены изменения	
		SELECT * FROM jury_member where Jury_Member_id>2599
		<pre>mysql> select * from jury_member where Jury_Member_id > 2599; +-----+-----+-----+-----+-----+-----+ Jury_Member_id Jury_Member_Surname Jury_Member_Name Jury_Member_Patronymic Section_id Olympiad_id +-----+-----+-----+-----+-----+-----+ 2600 Andreev Andrey Andreevich 1 1 2602 Andreev Andrey Andreevich 1 1 2604 shhh xuxa NULL 10 10 +-----+-----+-----+-----+-----+-----+ </pre>

Заключение

В процессе выполнения пяти лабораторных работ были получены различные навыки.

Реализовано представление на основе запроса. Для каждой секции подсчитано количество докладов и количество членов жюри. Сформулирован запрос, использующий представление, а также продемонстрирован результат попытки изменения, добавления и удаления данных в созданное представление.

Созданы 3 триггера на добавление, удаление и обновление данных в таблицах, на основе которых была создана таблица для хранения статистических данных. Необходимо собирать статистику о количестве членов жюри в одной секции.

Созданы два пользователя с различными правами. Первый пользователь может просматривать представление, второй — имеет полный доступ к таблицам, использованным в создании представления. Сформирована таблица сравнения прав доступа двух пользователей для 6 различных операций: запрос на выборку из представления, запрос на выборку из таблицы `jury_member`, запрос на обновление таблицы `jury_member`, запрос на удаление из таблицы `jury_member`, запрос на вставку в таблицу `jury_member`, запрос на удаление таблицы `jury_member`.

Реализована функция, которая принимает на вход фамилию, имя и отчество персоны и возвращает конкатенацию инициалов и фамилии персоны в формате «И.О. Фамилия». При отсутствии отчества результат возвращается в формате «И. Фамилия».

Создана процедура, которая будет принимать значения фамилии, имени и отчества участника, фамилии, имени и отчества сопровождающего, название секции и название доклада. Если участника нет в таблице `participant`, нужно добавить его и его доклад в секцию. Если сопровождающего нет в таблице `chaperone`, его также нужно добавить в эту таблицу, при условии, что за участником уже не закреплён другой сопровождающий.

Установлен уровень изоляции транзакций, соответствующий стандарту повторяемого чтения (`repeatable read`), и выявлен случай, когда появляется фантом. Приведена таблица, описывающая 6 шагов, для выявления фантома.