

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №4 «Запросы на выборку и модификацию данных.
Представления. Работа с индексами»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Фролова К.И.

Факультет: ИКТ

Группа: K3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

1. Запросы к базе данных	3
2. Представления.....	8
3. Кастом запросы.....	11
4. Индексы.....	17
Вывод.....	19

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4.

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

1. Запросы к базе данных

- Составить список дисциплин, которые должны быть сданы заданной группой с указанием дат сдачи и фамилий преподавателей.

```
SELECT group_id, "Session".discipline.discipline_name, date, teacher.surname
FROM "Session".schedule
JOIN "Session".teacher ON "Session".schedule."personal_id" =
"Session".teacher."personal_id"
JOIN "Session"."educational program discipline"
ON "Session".schedule."ed_prog_disc_id" = "Session"."educational program
discipline".ed_discipline_id
JOIN "Session".discipline
ON "educational program discipline".discipline_id = discipline.discipline_id
WHERE group_id = 807
```

Query Query History

```
1 SELECT group_id, "Session".discipline.discipline_name, date, teacher.surname
2 FROM "Session".schedule
3 JOIN "Session".teacher ON "Session".schedule."personal_id" = "Session".teacher."personal_id"
4 JOIN "Session"."educational program discipline"
5 ON "Session".schedule."ed_prog_disc_id" = "Session"."educational program discipline".ed_discipline_id
6 JOIN "Session".discipline
7 ON "educational program discipline".discipline_id = discipline.discipline_id
8 WHERE group_id = 807
9
```

Data Output Messages Notifications

	group_id integer	discipline_name character varying (20)	date timestamp with time zone	surname character varying (20)
1	807	БЖД	2023-10-22 00:00:00+03	Martin
2	807	История	2023-10-22 00:00:00+03	Phillip

- Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.

```
SELECT student_id, teacher.surname FROM "Session"."attestation pass"
JOIN "Session".teacher ON "Session"."attestation pass"."personal_id" =
"Session".teacher."personal_id"
WHERE mark = 2 AND attempt = 1
```

Query Query History

```
1 -- Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересда
2 SELECT student_id, teacher.surname FROM "Session"."attestation pass"
3 JOIN "Session".teacher ON "Session"."attestation pass"."personal_id" = "Session".teacher."personal_id"
4 WHERE mark = 2 AND attempt = 1
```

Data Output Messages Explain X Notifications

	student_id integer	surname character varying (20)
1	8819	Natalie

- Вывести фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине.

```
SELECT DISTINCT student.record_book_id, student.surname, discipline_id, mark
FROM "Session"."attestation pass" a
JOIN "Session"."active student"
ON a.student_id = "Session"."active student"."student_id"
JOIN "Session".student
ON "Session"."active student".record_book_id =
"Session".student."record_book_id"
GROUP BY a.attestation_id, student.record_book_id, student.surname
HAVING a.mark > (
    SELECT AVG(mark)
    FROM "Session"."attestation pass"
    WHERE discipline_id = a.discipline_id
```

Query Query History

```

1 -- Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересда
2 SELECT student_id, teacher.surname FROM "Session"."attestation pass"
3 JOIN "Session".teacher ON "Session"."attestation pass"."personal_id" = "Session".teacher."personal_id"
4 WHERE mark = 2 AND attempt = 1

```

Data Output Messages Explain X Notifications

	student_id integer	surname character varying (20)
1	8819	Natalie

- Создать рейтинговый список групп по заданному направлению по результатам сдачи сессии, упорядочить его по убыванию

```

--
SELECT group_number, ROUND(AVG(mark), 2)
FROM "Session"."attestation pass"
JOIN "Session"."active student"
ON "Session"."attestation pass".student_id = "Session"."active student"."student_id"
JOIN "Session".group
ON "Session"."active student".group_id = "Session".group.group_id
WHERE date BETWEEN '2023-06-10' AND '2023-06-30'
GROUP BY group_number
ORDER BY AVG(mark) DESC

```

Query Query History

```

1 -- Создать рейтинговый список групп по заданному направлению по результатам сдачи сессии, упорядочить его по убываник
2 SELECT group_number, ROUND(AVG(mark), 2)
3 FROM "Session"."attestation pass"
4 JOIN "Session"."active student"
5 ON "Session"."attestation pass".student_id = "Session"."active student"."student_id"
6 JOIN "Session".group
7 ON "Session"."active student".group_id = "Session".group.group_id
8 WHERE date BETWEEN '2023-06-10' AND '2023-06-30'
9 GROUP BY group_number
10 ORDER BY AVG(mark) DESC

```

Data Output Messages Notifications

	group_number character varying (10)	round numeric
1	K3323	5.00
2	K3321	4.33
3	K3143	4.00
4	K3140	3.50
5	K3122	3.00
6	K3142	3.00
7	K3239	2.00

- Создайте списки студентов, упорядоченные по группам и фамилиям студентов, содержащие данные о средних баллах и назначении на стипендии. Студент получает стипендию, если он сдал сессию без троек. Если студент не назначен на стипендию, указать 0, если назначен – 1.

```
SELECT a.group_id, a.student_id, s.surname, ROUND(AVG(at.mark), 2) AS
average_mark,
CASE
    WHEN COUNT(CASE WHEN at.mark <= '3' THEN 1 END) = 0 THEN 1
    ELSE 0
END AS scholarship
FROM "Session"."attestation pass" at
JOIN "Session"."active student" a ON at.student_id = a.student_id
JOIN "Session".student s ON a.record_book_id = s.record_book_id
GROUP BY a.group_id, a.student_id, s.surname, mark;
```

Query Query History

```
1  -- Создайте списки студентов, упорядоченные по группам и фамилиям студентов,
2  -- содержащие данные о средних баллах и назначении на стипендии.
3  -- Студент получает стипендию, если он сдал сессию без троек.
4  -- Если студент не назначен на стипендию, указать 0, если назначен – 1.
5
6  SELECT a.group_id, a.student_id, s.surname, ROUND(AVG(at.mark), 2) AS average_mark,
7  CASE
8      WHEN COUNT(CASE WHEN at.mark <= '3' THEN 1 END) = 0 THEN 1
9      ELSE 0
10 END AS scholarship
11 FROM "Session"."attestation pass" at
12 JOIN "Session"."active student" a ON at.student_id = a.student_id
13 JOIN "Session".student s ON a.record_book_id = s.record_book_id
14 GROUP BY a.group_id, a.student_id, s.surname, mark;
```

Data Output Messages Notifications



	group_id integer	student_id integer	surname character varying (20)	average_mark numeric	scholarship integer
1	860	5085	John	5.00	1
2	732	5701	Kristin	4.00	1
3	335	1895	Brett	2.00	0
4	807	8202	Kristin	5.00	1
5	807	1101	Sara	5.00	1
6	807	1101	Sara	4.00	1
7	443	9105	Sara	2.00	0
8	443	9105	Sara	5.00	1
9	441	3514	Ricky	3.00	0
10	335	1895	Brett	5.00	1
11	807	8202	Kristin	2.00	0
12	228	1795	Kristen	3.00	0

- Вывести список студентов, сдавших все положенные экзамены.

```
--
SELECT , a.student_id, s.surname
FROM "Session".student s
```

```
JOIN "Session"."active student" a ON s.record_book_id = a.record_book_id
LEFT JOIN "Session"."attestation pass" ap ON a.student_id = ap.student_id
GROUP BY a.student_id, s.surname
HAVING COUNT(DISTINCT ap.discipline_id) = (SELECT COUNT(DISTINCT
ed_prog_disc_id) FROM "Session".schedule)
```

Query Query History

```
1 SELECT s.surname
2 FROM "Session".student s
3 JOIN "Session"."active student" a ON s.record_book_id = a.record_book_id
4 LEFT JOIN "Session"."attestation pass" ap ON a.student_id = ap.student_id
5 GROUP BY s.surname
6 HAVING COUNT(DISTINCT ap.discipline_id) = (SELECT COUNT(DISTINCT ed_prog_disc_id) FROM "Session".schedule)
```

Data Output Messages Notifications



	surname character varying (20)
1	Sara

- Вывести список студентов, получивших максимальный средний балл в своей группе.

```
--
SELECT s.record_book_id, s.surname, a.group_id, MAX(sub.avg_mark) AS
max_average_mark
FROM "Session".student s
JOIN "Session"."active student" a ON s.record_book_id = a.record_book_id
JOIN (
    SELECT ap.student_id, ROUND(AVG(ap.mark), 2) AS avg_mark
    FROM "Session"."attestation pass" ap
    GROUP BY ap.student_id
) sub ON a.student_id = sub.student_id
GROUP BY s.record_book_id, s.surname, a.group_id
```

```

1  -- Вывести список студентов, получивших максимальный средний балл в своей группе.
2  SELECT s.record_book_id, s.surname, a.group_id, MAX(sub.avg_mark) AS max_average_mark
3  FROM "Session".student s
4  JOIN "Session"."active student" a ON s.record_book_id = a.record_book_id
5  JOIN (
6      SELECT ap.student_id, ROUND(AVG(ap.mark), 2) AS avg_mark
7      FROM "Session"."attestation pass" ap
8      GROUP BY ap.student_id
9  ) sub ON a.student_id = sub.student_id
10 GROUP BY s.record_book_id, s.surname, a.group_id
11 |

```

Data Output Messages Notifications



	record_book_id integer	surname character varying (20)	group_id integer	max_average_mark numeric
1	39784	John	860	5.00
2	26163	Ricky	441	3.00
3	37086	Brett	335	3.50
4	26163	Ricky	583	2.00
5	25641	Kristin	807	3.50
6	2722	Sara	443	3.75
7	835	Kristen	228	3.00
8	25641	Kristin	732	4.00
9	2722	Sara	807	4.67

2. Представления

- Список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен;

```

CREATE VIEW "Session".Retake AS
SELECT s.surname, teacher.surname AS teacher_surname
FROM "Session".student s
JOIN "Session"."active student" a ON s.record_book_id = a.record_book_id
JOIN "Session"."attestation pass" ap ON a.student_id = ap.student_id
JOIN "Session".teacher ON ap.personal_id = teacher.personal_id
WHERE ap.mark = 2 AND ap.attempt = 1

```

Query Query History

```

1  -- список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзам
2  CREATE VIEW "Session".Retake AS
3  SELECT s.surname, teacher.surname AS teacher_surname
4  FROM "Session".student s
5  JOIN "Session"."active student" a ON s.record_book_id = a.record_book_id
6  JOIN "Session"."attestation pass" ap ON a.student_id = ap.student_id
7  JOIN "Session".teacher ON ap.personal_id = teacher.personal_id
8  WHERE ap.mark = 2 AND ap.attempt = 1

```



```

1 SELECT * FROM "Session".retake2
2

```

Data Output Messages Notifications

	surname character varying (20)	teacher_surname character varying (20)
1	Ricky	Natalie

- Данные о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки).

```

--
CREATE VIEW "Session".remand1 AS
SELECT a.student_id, s.surname
FROM "Session".student s
JOIN "Session"."active student" a ON s.record_book_id = a.record_book_id
JOIN "Session"."attestation pass" ap ON a.student_id = ap.student_id
JOIN "Session".teacher t ON ap.personal_id = t.personal_id
WHERE ap.mark = 2 AND ap.attempt = 3

```

Query

Query History

1

-- данных о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки).

2

3

CREATE VIEW "Session".remand1 AS

4

SELECT a.student_id, s.surname

5

FROM "Session".student s

6

JOIN "Session"."active student" a ON s.record_book_id = a.record_book_id

7

JOIN "Session"."attestation pass" ap ON a.student_id = ap.student_id

8

JOIN "Session".teacher t ON ap.personal_id = t.personal_id

9

WHERE ap.mark = 2 AND ap.attempt = 3

Data Output

Messages

Notifications

CREATE VIEW

Query returned successfully in 20 msec.

Query

Query History

1

2

SELECT

*

FROM

"Session".remand1

Data Output

Messages

Notifications

≡+

▼

▼

	<div>student_id</div> <div>integer</div> <div></div>	<div>surname</div> <div>character varying (20)</div> <div></div>
1	9105	Sara
2	1895	Brett
3	8202	Kristin

3. Кастом запросы

- Добавить данные по новой аттестации по фамилии преподавателя и номеру дисциплины

```
INSERT INTO "Session"."attestation pass"
(attestation_id, personal_id, student_id, discipline_id, date, mark, attempt)
VALUES (21,
        (SELECT personal_id FROM "Session".teacher WHERE teacher.surname =
'Phillip'),
        1101,
        (SELECT discipline_id FROM "Session"."educational program discipline"
WHERE discipline_id = 4),
        '2021-01-01',
        5,
        1);
```

Query

Query History

1

2

3

4

5

6

7

8

9

10

INSERT INTO "Session"."attestation pass"

(attestation_id, personal_id, student_id, discipline_id, date, mark, attempt)

VALUES (21,

(SELECT personal_id FROM "Session".teacher WHERE teacher.surname = 'Phillip'),

1101,

(SELECT discipline_id FROM "Session"."educational program discipline" WHERE discipline_id = 4),

'2021-01-01',

5,

1);

Data Output

Messages

Notifications

INSERT 0 1

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗄

⬇

📈

personal_id

integer

🖋

student_id

integer

🖋

discipline_id

integer

🖋

date

date

🖋

mark

integer

🖋

attempt

integer

🖋

attestation_id

[PK] integer

🖋

10

4146

8202

87

2023-10-22

5

3

10

11

7782

1895

92

2023-10-22

5

3

11

12

7782

7936

87

2023-10-22

4

2

12

13

7782

9105

30

2023-10-22

5

2

13

14

8201

1101

4

2023-10-22

4

3

14

15

8201

1895

15

2023-10-22

2

3

15

16

8201

3514

30

2023-10-22

3

1

16

17

8201

5085

15

2023-10-22

5

3

17

18

8201

5701

40

2023-10-22

4

3

18

19

8201

9105

15

2023-10-22

5

1

19

20

8201

9105

22

2023-10-22

2

3

20

21

1928

1101

4

2021-01-01

5

1

21

- Изменить оценку за экзамен у определённого студента по определённому предмету

```
UPDATE "Session"."attestation pass"  
SET mark = 4  
WHERE student_id = (SELECT a.student_id FROM "Session".student  
JOIN "Session"."active student" a ON student.record_book_id =  
a.record_book_id  
WHERE student.surname = 'Sara'  
LIMIT 1)  
AND discipline_id = 30;
```

Query Query History

```
1 UPDATE "Session"."attestation pass"  
2 SET mark = 4  
3 WHERE student_id = (SELECT a.student_id FROM "Session".student  
4 JOIN "Session"."active student" a ON student.record_book_id = a.record_book_id  
5 WHERE student.surname = 'Sara'  
6 LIMIT 1)  
7 AND discipline_id = 30;  
8
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 18 msec.

Query Query History

```
1 SELECT * FROM "Session"."attestation pass"
2 ORDER BY attestation_id ASC
```

Data Output Messages Notifications

	personal_id integer	student_id integer	discipline_id integer	date date	mark integer	attempt integer	attestation_id [PK] integer
1	1928	1101	22	2023-10-22	5	2	1
2	1928	9105	10	2023-10-22	4	2	2
3	2413	3514	87	2023-10-22	3	2	3
4	2413	8202	68	2023-10-22	2	3	4
5	2413	8819	87	2023-10-22	2	2	5
6	3758	8819	88	2023-10-22	2	1	6
7	4146	1795	30	2023-10-22	3	2	7
8	4146	1795	88	2023-10-22	3	1	8
9	4146	3514	15	2023-10-22	3	3	9
10	4146	8202	87	2023-10-22	5	3	10
11	7782	1895	92	2023-10-22	5	2	11
12	7782	7936	87	2023-10-22	4	2	12

✓ Successfully run. Total c

Data Output Messages Notifications

	personal_id integer	student_id integer	discipline_id integer	date date	mark integer	attempt integer	attestation_id [PK] integer
8	4146	1795	88	2023-10-22	3	1	8
9	4146	3514	15	2023-10-22	3	3	9
10	4146	8202	87	2023-10-22	5	3	10
11	7782	1895	92	2023-10-22	5	3	11
12	7782	7936	87	2023-10-22	4	2	12
13	7782	9105	30	2023-10-22	4	2	13
14	8201	1101	4	2023-10-22	4	3	14

- Удалить данные о результате аттестации у определённого студента

```
DELETE FROM "Session"."attestation pass"
WHERE student_id = (SELECT a.student_id FROM "Session".student
JOIN "Session"."active student" a ON student.record_book_id =
a.record_book_id
WHERE student.surname = 'Jon'
LIMIT 1)
```

Query Query History

```
1 DELETE FROM "Session"."attestation pass"
2 WHERE student_id = (SELECT a.student_id FROM "Session".student
3 JOIN "Session"."active student" a ON student.record_book_id = a.record_book_id
4 WHERE student.surname = 'Ricky'
5 LIMIT 1)
```

Data Output Messages Explain X Notifications

DELETE 3

Query returned successfully in 22 msec.

Data Output Messages Notifications



	personal_id integer	student_id integer	discipline_id integer	date date	mark integer	attempt integer	attestation_id [PK] integer
10	4146	8202	87	2023-06-13	5	3	10
11	7782	1895	92	2023-06-13	5	3	11
12	7782	9105	30	2023-06-13	4	2	13
13	8201	1101	4	2023-06-15	4	3	14
14	8201	1895	15	2023-06-13	2	3	15
15	8201	3514	30	2023-10-22	3	1	16
16	8201	5085	15	2023-06-13	5	3	17
17	8201	5701	40	2023-10-22	4	3	18
18	8201	9105	15	2023-10-22	5	1	19
19	8201	9105	22	2023-10-22	2	3	20
20	1928	1101	4	2023-06-15	5	1	21
21	1928	1101	4	2023-06-15	5	1	22

Query Query History

```
1 SELECT * FROM "Session"."attestation pass"
2 ORDER BY attestation_id ASC
```

Data Output Messages Notifications



	personal_id integer	student_id integer	discipline_id integer	date date	mark integer	attempt integer	attestation_id [PK] integer
1	1928	1101	22	2023-06-10	5	2	1
2	1928	9105	10	2023-06-30	4	2	2
3	2413	8202	68	2023-06-10	2	3	4
4	2413	8819	87	2023-06-25	2	2	5
5	3758	8819	88	2023-06-25	2	1	6
6	4146	1795	30	2023-06-25	3	2	7
7	4146	1795	88	2023-06-25	3	1	8
8	4146	8202	87	2023-06-13	5	3	10
9	7782	1895	92	2023-06-13	5	3	11
10	7782	9105	30	2023-06-13	4	2	13
11	8201	1101	4	2023-06-15	4	3	14
12	8201	1895	15	2023-06-13	2	3	15
13	8201	5085	15	2023-06-13	5	3	17
14	8201	5701	40	2023-10-22	4	3	18
15	8201	9105	15	2023-10-22	5	1	19
16	8201	9105	22	2023-10-22	2	3	20
17	1928	1101	4	2023-06-15	5	1	21
18	1928	1101	4	2023-06-15	5	1	22

QueryHistory

Query

Query History

Show queries generated internally by pgAdmin? ☒

Remove

Remove All

Today - 13.11.2023

SELECT * FROM "Session"."attestation pass" ORDER BY at...
15:04:33

DELETE FROM "Session"."attestation pass" WHERE student...
15:04:31

SELECT * FROM "Session"."active student" ORDER BY stud...
15:03:57

SELECT * FROM "Session".student ORDER BY record_book_i...
15:03:18

SELECT * FROM "Session".schedule ORDER BY schedule_id ...
15:00:24

SELECT * FROM "Session".schedule ORDER BY schedule_id ...
14:59:55

SELECT * FROM "Session"."attestation pass" ORDER BY at...
14:56:02

SELECT * FROM "Session"."attestation pass" ORDER BY at...
14:56:01

SELECT * FROM "Session"."attestation pass" ORDER BY at...
14:55:40

UPDATE "Session"."attestation pass" SET mark = 4 WHERE...
14:55:19

UPDATE "Session"."attestation pass" SET mark = 4 WHERE...
14:53:33

UPDATE "Session"."attestation pass" SET mark = 4 WHERE...
14:53:31

UPDATE "Session"."attestation pass" SET mark = 4 WHERE...
14:50:52

SELECT * FROM "Session"."attestation pass" ORDER BY at...
14:50:33

13.11.2023 15:04:33

20

34 msec

Date

Rows affected

Duration

Copy

Copy to Query Editor

SELECT * FROM "Session"."attestation pass"
ORDER BY attestation_id ASC

Messages

Successfully run. Total query runtime: 34 msec. 20 rows affected.

4. Индексы

```
CREATE INDEX idx_teacher_surname ON "Session"."teacher" (surname);  
CREATE INDEX idx_attestation_results ON "Session"."attestation pass" (mark,  
attempt);  
CREATE INDEX idx_student_surname ON "Session"."student" (surname);
```

```
-- Вывести список студентов, получивших двойки на первой попытке с указанием  
фамилии преподавателя, которым они должны пересдать экзамен.  
SELECT student_id, teacher.surname FROM "Session"."attestation pass"  
JOIN "Session".teacher ON "Session"."attestation pass"."personal_id" =  
"Session".teacher."personal_id"  
WHERE mark = 2 AND attempt = 1
```

Без индекса:

Query Query History

```

1 -- Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они д
2 SELECT student_id, teacher.surname FROM "Session"."attestation pass"
3 JOIN "Session".teacher ON "Session"."attestation pass"."personal_id" = "Session".teacher."personal_id"
4 WHERE mark = 2 AND attempt = 1

```

Data Output Messages Notifications

Successfully run. Total query runtime: 138 msec.
1 rows affected.

С индексом:

Query Query History

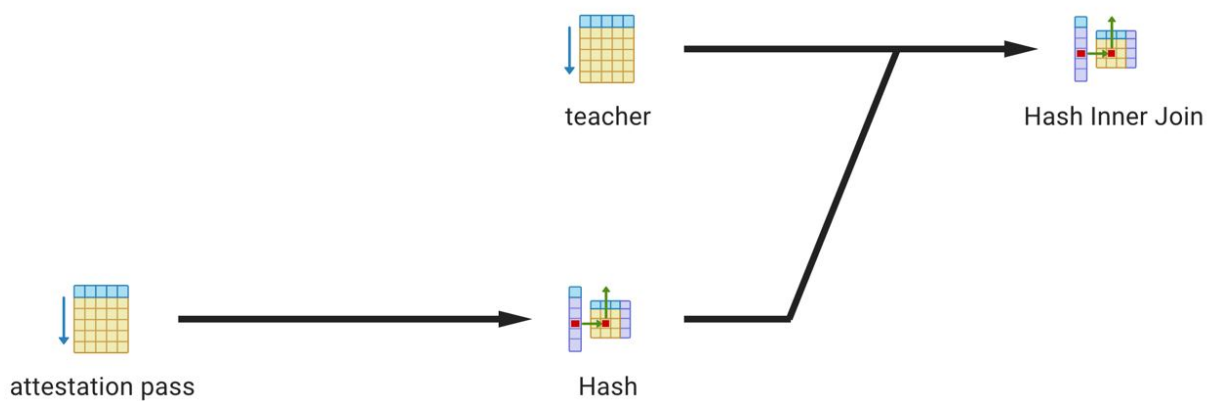
```

1 -- Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересда
2 SELECT student_id, teacher.surname FROM "Session"."attestation pass"
3 JOIN "Session".teacher ON "Session"."attestation pass"."personal_id" = "Session".teacher."personal_id"
4 WHERE mark = 2 AND attempt = 1

```

Data Output Messages Notifications

Successfully run. Total query runtime: 24 msec.
1 rows affected.



```
DROP INDEX idx_teacher_surname;  
DROP INDEX idx_attestation_results;  
DROP INDEX idx_student_surname;
```

Вывод

В ходе лабораторной работы я освоила работу с Query Tool, а именно с выборкой данных, добавлением, обновлением и их удалением. Также научилась делать представление данных и индексировать поля, в следствии чего узнала, как можно улучшить скорость работы базы данных.