

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет инфокоммуникационных технологий

Направление: Интеллектуальные системы в гуманитарной сфере

ОТЧЕТ

О Лабораторной работе №4

по теме: «Построение запросов к БД. Select SQL. Индексы»

по дисциплине: Проектирование и реализация баз данных

Выполнил:

студент группы К3239

Котовщиков Андрей Романович

Проверил:

Говорова М. М. _____

Дата: “ “ ноября 2023 года

Оценка _____

4. Выполнение

1. Вывести все номера групп и программы, где количество слушателей меньше 10

```
SELECT
    g.name,
    c.code,
    c.name
FROM lab_3_2.group AS g
JOIN lab_3_2.implemented_curriculum AS ic
    ON g.implemented_curriculum_id = ic.id
JOIN lab_3_2.curriculum AS c
    ON ic.curriculum_code = c.code
WHERE (SELECT COUNT(*) FROM lab_3_2.student_in_group AS sig
    WHERE sig.group_name = g.name) < 10;
```

	name character varying (20) 🔒	code character varying (15) 🔒	name character varying (70) 🔒
1	K3240	Curriculum-9164	Мобильные Технологии
2	K3241	Curriculum-9164	Мобильные Технологии
3	K3242	Curriculum--307	Ядерная физика
4	K3243	Curriculum-9164	Мобильные Технологии

2. Вывести список преподавателей с указанием количества программ, где они преподавали за истекший год

```
SELECT
    t.name,
    t.surname,
    t.patronymic,
    SUM(CASE WHEN curriculum_code IS NULL THEN 0 ELSE 1 END)
    AS curriculum_count
FROM (
    SELECT DISTINCT
        c.code AS curriculum_code,
        c.name AS curriculum_name,
        s.teacher_id AS teacher_id
    FROM lab_3_2.schedule AS s
    JOIN lab_3_2.group AS g
        ON s.group_name = g.name
    JOIN lab_3_2.implemented_curriculum AS ic
        ON g.implemented_curriculum_id = ic.id
    JOIN lab_3_2.curriculum AS c
        ON ic.curriculum_code = c.code
    WHERE EXTRACT (YEAR FROM s.datetime) =
        EXTRACT (YEAR FROM CURRENT_DATE - INTERVAL '1 Year')
) AS teacher_schedule_for_last_year
RIGHT JOIN lab_3_2.teacher AS t
    ON teacher_id = t.id
GROUP BY t.name, t.surname, t.patronymic;
```

	name character varying (30) 🔒	surname character varying (30) 🔒	patronymic character varying (30) 🔒	curriculum_count bigint 🔒
1	Ирина	Натановна	Евсеева	0
2	Брагина	Алевтина	Феликсовна	0
3	Харитоновна	Наина	Юрьевна	0
4	Зосима	Дорофеевич	Мельников	1
5	Пелагея	Болеславовна	Тетерина	0
6	Петр	Петров	Бездельников	0
7	Екатерина	Ильинична	Кононова	0
8	Иванна	Сергеевна	Карпова	0
9	Изот	Валерьянович	Блохин	0
10	Марфа	Захаровна	Лыткина	0
11	Дементий	Тихонович	Морозов	0

3. Вывести список преподавателей, которые не проводят занятия на третьей паре ни в один из дней недели

```
SELECT
    t.name,
    t.surname,
    t.patronymic
FROM lab_3_2.teacher AS t
WHERE NOT EXISTS (SELECT 1 FROM lab_3_2.schedule AS s
                  WHERE s.teacher_id = t.id
                     AND s.datetime::time = '11:40'::time);
```

	name character varying (30) 🔒	surname character varying (30) 🔒	patronymic character varying (30) 🔒
1	Ирина	Натановна	Евсеева
2	Харитоновна	Наина	Юрьевна
3	Пелагея	Болеславовна	Тетерина
4	Марфа	Захаровна	Лыткина
5	Изот	Валерьянович	Блохин
6	Зосима	Дорофеевич	Мельников
7	Брагина	Алевтина	Феликсовна
8	Екатерина	Ильинична	Кононова
9	Иванна	Сергеевна	Карпова
10	Дементий	Тихонович	Морозов
11	Петр	Петров	Бездельников

4. Вывести список свободных лекционных аудиторий на ближайший понедельник

```
SELECT
    c.code,
    c.type,
    c.capacity
FROM lab_3_2.classroom AS c
WHERE NOT EXISTS (SELECT 1 FROM lab_3_2.schedule AS s
                  WHERE s.classroom_code = c.code
                     AND s.datetime = DATE_TRUNC('WEEK', NOW()) + INTERVAL '7 Days');
```

	code [PK] character varying (15)	type lab_3_2.classroom_type	capacity smallint
1	Aud-1000	Практическая	175
2	Aud-1001	Практическая	60
3	Aud-1002	Практическая	82
4	Aud-1003	Лабораторная	69
5	Aud-1004	Лекционная	126
6	Aud-1005	Лабораторная	153
7	Aud-1006	Лекционная	182
8	Aud-1007	Лекционная	171
9	Aud-1008	Практическая	83
10	Aud-1009	Лабораторная	163
11	Aud-1010	Лабораторная	61

5. Вычислить общее количество обучающихся по каждой программе за последний год

```

SELECT
    c.code,
    c.name,
    SUM(CASE WHEN student_count IS NULL THEN 0 ELSE student_count END)
        AS total_student_count
FROM (
    SELECT
        ic.id AS implemented_curriculum_id,
        ic.curriculum_code,
        COUNT (*) AS student_count
    FROM lab_3_2.student_in_group AS sig
    JOIN lab_3_2.group AS g
        ON sig.group_name = g.name
    JOIN lab_3_2.implemented_curriculum AS ic
        ON g.implemented_curriculum_id = ic.id
    WHERE (CURRENT_DATE - INTERVAL '1 Year', CURRENT_DATE)
        OVERLAPS (sig.start_date, sig.end_date)
    GROUP BY ic.id, ic.curriculum_code
) AS implemented_curriculum_student_count
RIGHT JOIN lab_3_2.curriculum AS c
    ON curriculum_code = c.code
GROUP BY c.code, c.name;

```

	code [PK] character varying (15)	name character varying (70)	total_student_count numeric
1	Curriculum--307	Ядерная физика	0
2	Curriculum-6964	Сетевые технологии	0
3	Curriculum--599	Облачные технологии	0
4	Curriculum-9164	Мобильные Технологии	16

6. Вычислить среднюю загруженность компьютерных классов в неделю за последний месяц (в часах)

```

SELECT
    (COUNT(*) * 1.5 / 4)::numeric(5, 1)
    AS avg_workload_per_week
FROM lab_3_2.classroom AS cr
JOIN lab_3_2.schedule AS s
    ON s.classroom_code = cr.code
WHERE cr.type = 'Практическая'
    AND s.datetime BETWEEN NOW() - INTERVAL '1 Month' AND NOW();

```

	avg_workload_per_week
	numeric (5,1)
1	13.9

7. Найти самые популярные программы за последние 3 года

```

SELECT
    code,
    name,
    total_student_count
FROM
(
    SELECT
        c.code,
        c.name,
        SUM (CASE WHEN student_count IS NULL THEN 0 ELSE student_count END)
        AS total_student_count
    FROM (
        SELECT
            ic.id,
            ic.curriculum_code AS curriculum_code,
            COUNT(*) AS student_count
        FROM lab_3_2.student_in_group AS sig
        JOIN lab_3_2.group AS g
            ON sig.group_name = g.name
        JOIN lab_3_2.implemented_curriculum AS ic
            ON g.implemented_curriculum_id = ic.id
        WHERE sig.start_date BETWEEN CURRENT_DATE - INTERVAL '3 Year' AND CURRENT_DATE
        GROUP BY ic.id, curriculum_code
    ) AS implemented_curriculum_student_count
    RIGHT JOIN lab_3_2.curriculum AS c
        ON curriculum_code = c.code
    GROUP BY c.code, c.name
    ORDER BY total_student_count DESC
) AS tsc
WHERE total_student_count = (

```

```

WHERE total_student_count = (
    SELECT MAX(total_student_count) FROM (
        SELECT
            c.code,
            c.name,
            SUM (CASE WHEN student_count IS NULL THEN 0 ELSE student_count END)
            AS total_student_count
        FROM (
            SELECT
                ic.id,
                ic.curriculum_code AS curriculum_code,
                COUNT(*) AS student_count
            FROM lab_3_2.student_in_group AS sig
            JOIN lab_3_2.group AS g
            ON sig.group_name = g.name
            JOIN lab_3_2.implemented_curriculum AS ic
            ON g.implemented_curriculum_id = ic.id
            WHERE sig.start_date BETWEEN CURRENT_DATE - INTERVAL '3 Year'
            AND CURRENT_DATE
            GROUP BY ic.id, curriculum_code
        ) AS implemented_curriculum_student_count
        RIGHT JOIN lab_3_2.curriculum AS c
        ON curriculum_code = c.code
        GROUP BY c.code, c.name
        ORDER BY total_student_count DESC
    ) AS tsc_max
);

```

	code [PK] character varying (15)	name character varying (70)	total_student_count numeric
1	Curriculum-9164	Мобильные Технологии	16

8. для потенциальных слушателей, содержащее перечень специальностей, изучаемых на них дисциплин и количество часов

```

CREATE VIEW get_curriculum_with_subjects_and_hours AS
SELECT DISTINCT
    c.code AS curriculum_code,
    c.name AS curriculum_name,
    c.description,
    s.name AS subject_name,
    s.practical_hours,
    s.lecture_hours,
    s.laboratory_hours,
    s.consultation_hours
FROM lab_3_2.curriculum AS c
JOIN lab_3_2.implemented_curriculum AS ic
    ON ic.curriculum_code = c.code
JOIN lab_3_2.curriculum_subject AS cs
    ON cs.implemented_curriculum_id = ic.id
JOIN lab_3_2.subject AS s
    ON cs.subject_code = s.code
ORDER BY curriculum_name;

```

```
SELECT * FROM get_curriculum_with_subjects_and_hours;
```

Output Messages Notifications

curriculum_code character varying (15)	curriculum_name character varying (70)	description text	subject_name character varying (50)	practical_hours smallint	lecture_hours smallint	laboratory_hours smallint
Curriculum-9164	Мобильные Технологии	Описание программы...	Программирование	10	9	9
Curriculum-9164	Мобильные Технологии	Описание программы...	Теория и Реализация Баз Данных	12	6	11
Curriculum--599	Облачные технологии	Описание программы...	Теория и Реализация Баз Данных	12	6	11
Curriculum--307	Ядерная физика	Описание программы...	Английский Язык	12	12	12
Curriculum--307	Ядерная физика	Описание программы...	Программирование	10	9	9

9. общий доход по каждой программе за последний год

```
CREATE VIEW get_total_income_for_each_curriculum AS
SELECT
    c.code AS curriculum_code,
    c.name AS curriculum_name
    SUM (CASE WHEN id IS NULL THEN 0 ELSE total_cost END) AS income
FROM (
    SELECT
        ic.id,
        ic.curriculum_code,
        SUM (CASE WHEN sig.code IS NULL THEN 0 ELSE 1 END) * cost_per_year AS total_cost
    FROM lab_3_2.implemented_curriculum AS ic
    JOIN lab_3_2.group AS g
        ON g.implemented_curriculum_id = ic.id
    JOIN lab_3_2.student_in_group AS sig
        ON sig.group_name = g.name
    WHERE sig.start_date BETWEEN NOW() - INTERVAL '1 Year' AND NOW()
    GROUP BY ic.id, ic.curriculum_code
) AS student_money
RIGHT JOIN lab_3_2.curriculum AS c
    ON curriculum_code = c.code
GROUP BY c.code, c.name;
```

```
SELECT * FROM get_total_income_for_each_curriculum;
```

Output Messages Notifications

curriculum_code character varying (15)	curriculum_name character varying (70)	income numeric
Curriculum--307	Ядерная физика	0
Curriculum-6964	Сетевые технологии	0
Curriculum--599	Облачные технологии	0
Curriculum-9164	Мобильные Технологии	1600000.00

10. создать занятие на ближайший понедельник в 00:00 со свободной аудиторией с самой большой вместимостью


```

INSERT INTO lab_3_2.schedule VALUES
(
    DEFAULT,
    'пн',
    true,
    NOW(),
    (SELECT c.code FROM lab_3_2.classroom AS c
     WHERE NOT EXISTS (SELECT 1 FROM lab_3_2.schedule AS s
                       WHERE s.classroom_code = c.code
                           AND s.datetime =
                               DATE_TRUNC('WEEK', NOW()) + INTERVAL '7 Days')

    ORDER BY capacity DESC
    LIMIT 1),
    (SELECT id FROM lab_3_2.curriculum_subject AS cs
     JOIN lab_3_2.subject AS s
     ON cs.subject_code = s.code
     WHERE s.name = 'Программирование'
     LIMIT 1),
    (SELECT id FROM lab_3_2.teacher AS t
     WHERE t.position = 'Преподаватель программирования'
     LIMIT 1),
    (SELECT name FROM lab_3_2.group LIMIT 1)
);

```

до:

	id [PK] integer	week_day lab_3_2.schedule_week_day	is_even boolean	datetime timestamp without time zone	classroom_code character varying (15)	curriculum_subject_id integer	teacher_id integer	group_name character var
98		пн	false	2023-10-08 10:34:41.950087	Aud-3970	20	82	K3239
99	100	пн	false	2023-10-04 10:34:41.957088	Aud-3970	12	84	K3239
100	102	пн	true	2022-10-27 02:01:51.491161	Aud-1003	18	80	K3239
101	103	пн	true	2022-10-27 02:02:32.152942	Aud-1004	18	80	K3239
102	104	пн	true	2022-10-27 02:03:03.248844	Aud-1004	18	80	K3239
103	105	пн	true	2023-11-01 18:27:38.546323	Aud-1038	11	75	K3239
104	107	пн	true	2023-11-01 18:40:05.219641	Aud-1038	11	75	K3239
105	111	пн	true	2023-11-11 17:26:39.553587	Aud-1038	11	75	K3239

после:

	id [PK] integer	week_day lab_3_2.schedule_week_day	is_even boolean	datetime timestamp without time zone	classroom_code character varying (15)	curriculum_subject_id integer	teacher_id integer	group_name character var
99	100	пн	true	2023-10-04 10:34:41.957088	Aud-3970	12	84	K3239
100	102	пн	true	2022-10-27 02:01:51.491161	Aud-1003	18	80	K3239
101	103	пн	true	2022-10-27 02:02:32.152942	Aud-1004	18	80	K3239
102	104	пн	true	2022-10-27 02:03:03.248844	Aud-1004	18	80	K3239
103	105	пн	true	2023-11-01 18:27:38.546323	Aud-1038	11	75	K3239
104	107	пн	true	2023-11-01 18:40:05.219641	Aud-1038	11	75	K3239
105	111	пн	true	2023-11-11 17:26:39.553587	Aud-1038	11	75	K3239
106	114	пн	true	2023-11-13 00:00:00	Aud-1038	11	75	K3239

- установить статус (действующая) для программ, у которых есть хотя бы 1 реализованная программа

```

UPDATE lab_3_2.curriculum AS c
SET status = 'Действующая'
WHERE EXISTS (SELECT 1 FROM lab_3_2.implemented_curriculum AS ic
              WHERE ic.curriculum_code = c.code);

```

до:

	code [PK] character varying (15)	name character varying (70)	description text	status lab_3_2.curriculum_status	certificate character varying (50)
1	Curriculum-6964	Сетевые технологии	Описание программы...	Недействующая	Разработчик интернета
2	Curriculum--599	Облачные технологии	Описание программы...	Действующая	Devops инженер
3	Curriculum--307	Ядерная физика	Описание программы...	Действующая	Физик ядерщик
4	Curriculum-9164	Мобильные Технологии	Описание программы...	Недействующая	Разработчик мобильных телефонов

после:

	code [PK] character varying (15)	name character varying (70)	description text	status lab_3_2.curriculum_status	certificate character varying (50)
1	Curriculum-6964	Сетевые технологии	Описание программы...	Недействующая	Разработчик интернета
2	Curriculum--599	Облачные технологии	Описание программы...	Действующая	Devops инженер
3	Curriculum--307	Ядерная физика	Описание программы...	Действующая	Физик ядерщик
4	Curriculum-9164	Мобильные Технологии	Описание программы...	Действующая	Разработчик мобильных телефонов





12. Удалить 1 группу, где нет ни одного студента.

```
DELETE FROM lab_3_2.group
WHERE name IN (
    SELECT g.name
    FROM lab_3_2.group AS g
    WHERE (
        SELECT
            COUNT(*)
        FROM lab_3_2.student_in_group AS sig
        WHERE sig.group_name = g.name
    ) = 0
LIMIT 1
);|
```

до:

	name [PK] character varying (20)	start_date date	end_date date	implemented_curriculum_id integer
1	K3239	2023-10-26	2027-10-26	6
2	K3240	2023-10-26	2027-10-26	6
3	K3241	2023-10-26	2027-10-26	7
4	K3242	2023-10-26	2027-10-26	8
5	K3243	2023-10-26	2027-10-26	7

после:


	name [PK] character varying (20) 	start_date date 	end_date date 	implemented_curriculum_id integer 
1	K3239	2023-10-26	2027-10-26	6
2	K3240	2023-10-26	2027-10-26	6
3	K3241	2023-10-26	2027-10-26	7
4	K3243	2023-10-26	2027-10-26	7

13. Создание индексов


```
|  
CREATE INDEX ON lab_3_2.student USING HASH (email);  
CREATE INDEX ON lab_3_2.classroom (capacity);
```

до:

```
EXPLAIN SELECT * FROM lab_3_2.student  
WHERE email = 'hemeljanov@yandex.ru';
```


	QUERY PLAN text 
1	Seq Scan on student (cost=0.00..26.50 rows=1 wi...
2	Filter: ((email)::text = 'hemeljanov@yandex.ru':te...
Total rows: 2 of 2 Query complete 00:00:00.208	

```
EXPLAIN SELECT * FROM lab_3_2.classroom
WHERE capacity < 20
ORDER BY capacity DESC;
```

	QUERY PLAN	
	text	
1	Sort (cost=61.52..61.53 rows=1 width=24)	
2	Sort Key: capacity DESC	
3	-> Seq Scan on classroom (cost=0.00..61.51 rows=1 width=2...	
4	Filter: (capacity < 20)	
Total rows: 4 of 4 Query complete 00:00:00.167		


ПОСЛЕ:

```
EXPLAIN SELECT * FROM lab_3_2.student
WHERE email = 'hemeljanov@yandex.ru';
```

	QUERY PLAN	
	text	
1	Index Scan using student_email_idx1 on student (cost=0.00..8.02 rows=1 width=...	
2	Index Cond: ((email)::text = 'hemeljanov@yandex.ru'::text)	

Total rows: 2 of 2 Query complete 00:00:00.161

```
EXPLAIN SELECT * FROM lab_3_2.classroom
WHERE capacity < 20
ORDER BY capacity DESC;
```

	QUERY PLAN	
	text	
1	Index Scan Backward using classroom_capacity_idx on classroom (cost=0.28..8.30 rows=1 width=...	
2	Index Cond: (capacity < 20)	
Total rows: 2 of 2 Query complete 00:00:00.122		

удаление индексов:

```
DROP INDEX lab_3_2.student_email_idx;
DROP INDEX lab_3_2.classroom_capacity_idx;
```

5. Вывод

В рамках данной лабораторной работы были написаны различные вариации запросов на выборку, включающие в себя группировку, фильтрацию, агрегацию, сортировку и присоединение (join). Удалось поработать с подзапросами, причем они использовались как в запросах на чтение, так и в запросах на запись, удаление и обновление. Помимо этого, было проведено сравнение скорости выполнения запросов до и после создания индексов. Результаты показали прирост скорости выполнения запросов на чтение после использования индексов, однако запросы, связанные со вставкой, были немного замедлены, по причине перестроения структуры индекса во время изменения состояния таблицы. Также было установлено, что HASH индекс является хорошим решением для точного поиска на равенство по какому-либо строковому полю (например email). Время поиска с ним является константным $O(1)$. Индекс по умолчанию (Btree) является отличным решением для поиска числовых значений, включающего в себя различные операции сравнения (не только равно). Его сложность $O(\log n)$.