

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 3

з дисципліни “Бази даних”

тема “ Засоби оптимізації роботи СУБД PostgreSQL”

варіант - 20

Виконав

студент 2-го курсу

групи КП-93

Торговських Олександр Олегович

(прізвище, ім'я, по батькові)

Перевірів

“ ____ ” “ _____ ” 20__ р.

Асистент

Замекула Олексій Ігорович

(прізвище, ім'я, по батькові)

Варіант: Студент в університеті
(студент, група, факультет)

Київ 2020

Метою роботи є здобуття практичних навичок використання засобів оптимізації СУБД PostgreSQL.

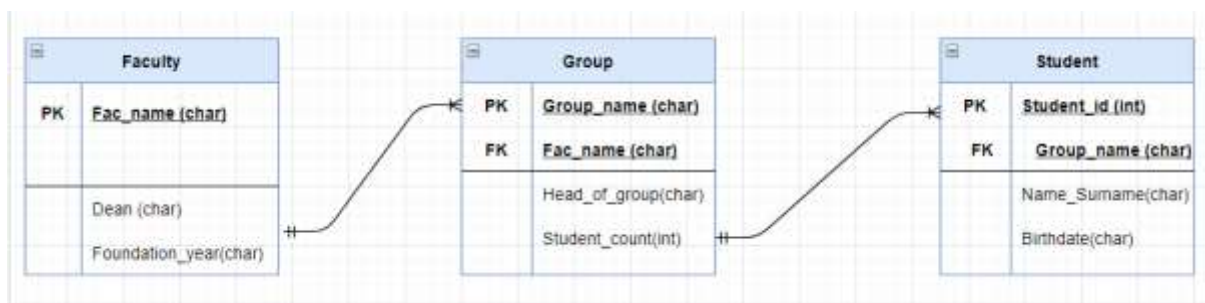
Завдання роботи полягає у наступному:

1. Перетворити модуль “Модель” з шаблону MVC лабораторної роботи №2 у вигляд об’єктно-реляційної проекції (ORM).
2. Створити та проаналізувати різні типи індексів у PostgreSQL.
3. Розробити тригер бази даних PostgreSQL.

URL репозиторію - <https://github.com/alex123411/Data-Base>

Пункт 1:

Таблиці та зв'язки між ними



ORM класи

```
class faculty(Base):
    __tablename__ = 'faculty'
    fac_name = Column(String(50), primary_key = True)
    foundation_year = Column(Integer)
    dean = Column(String(70))

class group(Base):
    __tablename__ = 'group'
    group_name = Column(String(111), primary_key = True)
    fac_name = Column(String(111), ForeignKey('faculty.fac_name'))
    head_of_group = Column(String(70))
    student_count = Column(Integer)

class student(Base):
    __tablename__ = 'student'
    student_id = Column(Integer, primary_key = True)
    group_name = Column(String(10), ForeignKey('group.group_name'))
    name_surname = Column(String(70))
    birthdate = Column(String(15))
```

Приклади запитів:

Додавання в базу

```
def add_faculty(self, x, y, z):
    try:
        faculty_1 = faculty(fac_name = x, foundation_year = y, dean = z)
        self.session_.add(faculty_1)
        self.session_.commit()
    except (Exception, psycopg2.Error) as er :
        print('Oops')
```

Видалення з бази

```
def delete_faculty(self, fac_name):
    try:
        groups = self.session_.query(group)
        for group_ in groups:
            if group_.fac_name == fac_name:
                self.session_.query(student).filter(student.group_name == group_.group_name).\
                    delete()
                self.session_.commit()

        self.session_.query(group).filter(group.fac_name == fac_name).\
            delete()
        self.session_.commit()

        faculty_ = self.session_.query(faculty).filter(faculty.fac_name == fac_name).first()
        self.session_.delete(faculty_)
        self.session_.commit()
    except (Exception, psycopg2.Error) as er :
        print(er)
```

Оновлення в базі

```
def update_group(self, group_name, fac_name, head_of_group, student_count):
    try:
        group_ = self.session_.query(group).filter(group.group_name == group_name).first()
        group_.group_name = group_name
        group_.fac_name = fac_name
        group_.head_of_group = head_of_group
        group_.student_count = student_count
        self.session_.commit()
    except (Exception, psycopg2.Error) as er :
        print(er)
```

Пункт 2:

2.1: Команда створення GIN індексу

```
create index head_of_group_idx on public.group using gin (head_of_group)
```

Query Editor История запросов

```
1 SELECT * FROM public.group where head_of_group = '{FW}'
```

Результат План выполнения Сообщения Notifications

	group_name [PK] character varying (111)	fac_name character varying (111)	student_count integer	head_of_group text[]
1	9a5f7e44007228e2fa03810f73e33c3c	FPM2	49263	{FW}
2	a7830f4bd7b09ef0914707a763ccea08	FPM2	11829	{FW}
3	3d00f48344ad93b7c41d9f20fc295275	FPM2	589	{FW}
4	dcf5db3298f2d07a4f2f28bd705499b8	FPM2	42609	{FW}
5	6972d22109cca737afb67e7ae2379...	FPM2	47456	{FW}
6	07854342e3cdd26c1a370c0bee2e9e...	FPM2	16770	{FW}
7	fe7938ab64e7d8f067c6a4203c2b32da	FPM2	13254	{FW}
8	7dc4cb186dff5b38f49a8caa78d04ec	FPM2	27906	{FW}
9	f90d80da0ffbf95c7bc1862d0629540	FPM2	11972	{FW}
10	460bc9ac2bb3af69654c197d49b75c...	FPM2	38830	{FW}
11	27c6ee6bcf0996eb67b25bf3e96ed89f	FPM2	25925	{FW}
12	...	FPM2	...	{FW}

QUERY PLAN

text

- 1 Bitmap Heap Scan on "group" (cost=12.60..233.11 rows=77 width=71)
- 2 Recheck Cond: (head_of_group = '{FW}')::text[])
- 3 -> Bitmap Index Scan on head_of_group_idx (cost=0.00..12.58 rows=77 width=0)
- 4 Index Cond: (head_of_group = '{FW}')::text[])

```

1 SELECT * FROM public.group where head_of_group[1] < 'SS' order by student_count;
2

```

Результат План выполнения Сообщения Notifications

	group_name [PK] character varying (111)	fac_name character varying (111)	student_count integer	head_of_group text[]
1	892112f5b7f277c1bfefe078be75b397	FPM2	7	{OQ}
2	9ade905ee173f3b3edcb16b4a727d1...	FPM2	8	{JD}
3	c634ac3d0354f10e505fbf1e96ea9c33	FPM2	9	{QL}
4	d49c5f865de3f7d851deec3b91f868c	FPM2	10	{OE}
5	a3afc1773749bf11d259b5800f8ba096	FPM2	10	{HM}
6	8f989e09056ba4f0d75ad317bb7bfe49	FPM2	11	{CX}
7	7ea5b8ddb1f0f216fb694d7c9b365cda	FPM2	14	{LA}
8	858442624435a9576fc89cb4d91e42...	FPM2	17	{QI}
9	ceec02854adc950daf9a2c61217d5c...	FPM2	17	{NJ}
10	028763058e93a6aed0944f9ee3d1442f	FPM2	17	{QL}
11	570c00bf4935646ca9c0a0411bec7f05	FPM2	19	{QJ}
12	689ac6a895f7d0ef1f84027c571123e3	FPM2	20	{SL}
13	6f6c838a6474eb6cacb16bf17c926150	FPM2	22	{NK}
14	91f0e9fc6b103e1c77ca80b9ff0574f7	FPM2	22	{NQ}

QUERY PLAN		🔒
	text	
1	Sort (cost=2460.75..2502.42 rows=16667 width=71)	
2	Sort Key: student_count	
3	-> Seq Scan on "group" (cost=0.00..1292.00 rows=16667 width=71)	
4	Filter: (head_of_group[1] < 'SS'::text)	

2.2 : Команда створення BRIN індексу

```
create index student_cnt_idx on public.group using brin (student_count)
```

```
1 SELECT * FROM public.group where student_count < 112 group by group_name, head_of_group;
2
```

Результат План выполнения Сообщения Notifications

	group_name [PK] character varying (111)	fac_name character varying (111)	student_count integer	head_of_group text[]	
1	004ed4db242f864141386b7056782d...	FPM2	74	{QJ}	
2	028763058e93a6aed0944f9ee3d1442f	FPM2	17	{QL}	
3	02db0f2697c75835e72ebee0e6b298...	FPM2	37	{JW}	
4	0497a4982676d098fd8e8f82be57bb...	FPM2	69	{NT}	
5	0549d6eb6753e483a2ca136b06449d...	FPM2	31	{EN}	
6	06a13c92dd7ee7391bb61ab907289a...	FPM2	85	{IJ}	
7	0e191765f778deb38d44c4f2ffbdec29	FPM2	35	{AK}	
8	118df616c9426bb8133416be99df2c...	FPM2	55	{KG}	
9	1bf17b94a9f977fdc5a631ca45043e18	FPM2	48	{UY}	
10	1d49b66402d0f38bf1d90a69b2ba03...	FPM2	67	{HY}	
11	27276a2ef7febc93cf6d886069debc19	FPM2	81	{JW}	
12	33f08a91e5f7dac6d715faea62d9179d	FPM2	40	{MY}	
13	35184f47b96edb935557cd52227da1...	FPM2	65	{GX}	
14	39d975e093997d6757cd4a15ac6219	FPM2	49	{FC}	

Результат План выполнения Сообщения Notifications

	QUERY PLAN text	
1	Group (cost=1295.52..1296.05 rows=105 width=71)	
2	Group Key: group_name	
3	-> Sort (cost=1295.52..1295.79 rows=105 width=71)	
4	Sort Key: group_name	
5	-> Seq Scan on "group" (cost=0.00..1292.00 rows=105 width=71)	
6	Filter: (student_count < 112)	

1 SELECT * FROM public.group where "student_count" < 11 and "student_count" > 1 order by head_of_group;

Результат

План выполнения

Сообщения

Notifications

	group_name [PK] character varying (111)	fac_name character varying (111)	student_count integer	head_of_group text[]	time_timezone timestamp without time zone
1	0a766f929759ac7d97068666c055cd8f	FPM4	4	{AG}	[null]
2	c7824a180b5cbb117352c888a0632e...	FPM4	10	{BG}	[null]
3	5750ef5bf395420181a7a5281291dd1f	FPM4	5	{CX}	[null]
4	322326ff42cfa85da9f2f10eadc12337	FPM4	2	{DA}	[null]
5	a1af78184d6d6b808d8cbfa724c450e3	FPM4	3	{EF}	[null]
6	fbe1539beac4b3f494399d0b5bb100f3	FPM4	8	{EJ}	[null]
7	4606a94991dc5ac76c6fb31a9cb87d...	FPM4	10	{EY}	[null]
8	3b1c0183ee8d3cc5c8fa01232a05abf7	FPM4	9	{GX}	[null]
9	0941db431364699ad2dac32b51f271...	FPM4	9	{HU}	[null]
10	dbd14df17781a47a8a27e365effe57bb	FPM4	5	{IQ}	[null]
11	381a638136d35ee77edd9ea3403cbb...	FPM4	2	{LH}	[null]
12	ece352818f971e432fe2ec0bd56e3a09	FPM4	5	{LI}	[null]
13	4cbc5f34e32e201eee9a8c30927bf0ca	FPM4	7	{MU}	[null]
14	801bd5bb3f260a1eb3eae96d0d431d1	FPM4	10	{OR}	[null]

QUERY PLAN

text

1	Sort (cost=3424.29..3424.33 rows=15 width=79)
2	Sort Key: head_of_group
3	-> Seq Scan on "group" (cost=0.00..3424.00 rows=15 width=79)
4	Filter: ((student_count < 11) AND (student_count > 1))

В наведених прикладах індекси прискорюють виконання запиту SELECT, оскільки були обрані раціональні поля для індексування.

GIN індексує не атомарні дані, а ті, які складаються із декількох елементів. Індексуються окремі елементи. Доречно використовувати його, наприклад у повнотекстовому пошуку.

Натомість BRIN створює «сторінки» значень де помічає мінімальне та максимальне значення на сторінці, що пришвидшує пошук відсортованих даних, та даних де можна чітко виділити діапазон на мін-макс. Недоречно використовувати якщо дані неможливо укласти у сторінки(розкиданий діапазон значень).

Пункт 3:

3.1)

after insert,

Текст триггеру

```
CREATE OR REPLACE FUNCTION after_insert_new() RETURNS trigger AS $after_insert_new_$
DECLARE
    c cursor for
    select group_name, fac_name, student_count, head_of_group from public.group;
BEGIN
    for record in c loop
        IF new.head_of_group = 'qweqwe' then
            raise notice 'head`s of group name doesn`t fit!';
            delete from public.group where head_of_group = 'qweqwe';
            return null;
        end if;
        if new.student_count > 100 then
            raise notice 'Can`t be such a big group!';
            delete from public.group where student_count = new.student_count;
            return null;
        end if;
    end loop;
    return new;
END;
$after_insert_new_$ LANGUAGE plpgsql;
CREATE TRIGGER insert_trigger after INSERT ON public.group
FOR EACH ROW EXECUTE PROCEDURE after_insert_new();
```

Запит на вставку

```
1 INSERT INTO public."group"(
2     group_name, fac_name, student_count, head_of_group)
3     VALUES ('fiot221', 'FPM2', 11, 'qweqwe') ,('fiot222', 'FPM2', 111, 'qwe') ;
```

Результат План выполнения Сообщения Notifications

NOTICE: head`s of group name doesn`t fit!
NOTICE: Can`t be such a big group!
INSERT 0 2

так як вставили не підходящі дані вони були одразу ж видалені з таблиці
тому таблиця залишилась пустою

```
1 SELECT * FROM public."group"
2 ORDER BY group_name ASC
```

Результат План выполнения Сообщения Notifications

	group_name [PK] character varying (111)	fac_name character varying (111)	student_count integer	head_of_group character varying (111)

3.2)

update

Текст триггеру

```
CREATE OR REPLACE FUNCTION update_trg() RETURNS trigger AS $update_trg$
DECLARE
    c cursor for
    select group_name, fac_name, student_count, head_of_group from public.group;
BEGIN
    IF new.student_count > 100 or new.student_count < 0 then
        raise notice 'student count can't be more then 100 or less then 0!';
        new.student_count = 25;
    end if;
    for record in c loop
        if record.fac_name = record.group_name then
            raise notice 'group and faculty can't have same names!';
            new.group_name = MD5(random()::text);
        end if;
    end loop;
    return new;
END;
$update_trg$ LANGUAGE plpgsql;
CREATE TRIGGER update_trigger before UPDATE ON public.group
FOR EACH ROW EXECUTE PROCEDURE update_trg();
```

ДО ЗМІН

	group_name [PK] character varying (111)	fac_name character varying (111)	student_count integer	head_of_group character varying (111)
1	fiot222	fiot222	11	eeq

запит

```
1 UPDATE public."group"
2     SET group_name='fiot222', fac_name='fiot222', student_count=411111, head_of_group='eeq'
3     WHERE fac_name = 'fiot222';
```

Результат План выполнения Сообщения Notifications

NOTICE: student count can't be more then 100 or less then 0!
NOTICE: group and faculty can't have same names!
UPDATE 1

результат змін

	group_name [PK] character varying (111)	fac_name character varying (111)	student_count integer	head_of_group character varying (111)
1	6d4ddb06b9ec28c343e3e0c49bed...	fiot222	25	eeq