НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Базы Данных

Лабораторная работа № 4

Вариант 888

Выполнил студент: Маликов Глеб Игоревич

Группа № P3124

Преподаватель: Королёва Юлия Александровна

г. Санкт-Петербург

2023

Задание

Составить запросы на языке SQL (пункты 1-2).

Для каждого запроса предложить индексы, добавление которых уменьшит время выполнения запроса (указать таблицы/атрибуты, для которых нужно добавить индексы, написать тип индекса; объяснить, почему добавление индекса будет полезным для данного запроса).

Для запросов 1-2 необходимо составить возможные планы выполнения запросов. Планы составляются на основании предположения, что в таблицах отсутствуют индексы. Из составленных планов необходимо выбрать оптимальный и объяснить свой выбор.

Изменятся ли планы при добавлении индекса и как?

Для запросов 1-2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]

Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете (планы выполнения запросов должны быть нарисованы, ответы на вопросы - представлены в текстовом виде).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
    - Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.
    - Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД, Н_ВЕДОМОСТИ.ЧЛВК_ИД.
    - Фильтры (AND):
        - Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = 2.
        - Н_ВЕДОМОСТИ.ЧЛВК_ИД > 105590.
    - Вид соединения: RIGHT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
    - Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ, Н_СЕССИЯ.
    - Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.ЧЛВК_ИД.
    - Фильтры (AND):
        - Н_ЛЮДИ.ФАМИЛИЯ > Ёлкин.
        - Н_ВЕДОМОСТИ.ИД < 1490007.
        - Н_СЕССИЯ.ДАТА > 2002-01-04.
    - Вид соединения: RIGHT JOIN.

Запросы

```
SELECT NTV."ИД", NV."ЧЛВК_ИД" FROM "Н_ТИПЫ_ВЕДОМОСТЕЙ" NTV RIGHT JOIN "Н_ВЕДОМОСТИ" NV ON "ТВ_ИД" = NTV."ИД" WHERE NTV."ИД" = 2 AND NV."ЧЛВК_ИД"> 105590;
```

```
SELECT NL."ИМЯ", NV."ДАТА", NS."ЧЛВК_ИД" FROM "Н_ЛЮДИ" NL RIGHT JOIN "Н_ВЕДОМОСТИ" NV on NL."ИД" = NV."ЧЛВК_ИД" RIGHT JOIN "Н_СЕССИЯ" NS ON NL."ИД" = NS."ЧЛВК_ИД" WHERE NL."ФАМИЛИЯ" > 'Ёлкин' AND NV."ИД" < 1490007 AND NS."ДАТА" > TIMESTAMP '2002-01-04 00:00:00';
```

Вывод EXPLAIN ANALYZE

1.

```
Nested Loop (cost=243.51..4849.91 rows=21570 width=8) (actual time=1.180..10.429 rows=21468 loops=1)
"  ->  Seq Scan on ""Н_ТИПЫ_ВЕДОМОСТЕЙ"" ntv  (cost=0.00..1.04 rows=1 width=4) (actual time=0.011..0.012 rows=1 loops=1)"
"        Filter: (""ИД"" = 2)"
        Rows Removed by Filter: 2
"  ->  Bitmap Heap Scan on ""Н_ВЕДОМОСТИ"" nv  (cost=243.51..4633.17 rows=21570 width=8) (actual time=1.164..7.881 rows=21468 loops=1)"
"        Recheck Cond: (""ТВ_ИД"" = 2)"
"        Filter: (""ЧЛВК_ИД"" > 105590)"
        Heap Blocks: exact=2274
"        ->  Bitmap Index Scan on ""ВЕД_ТВ_FK_I""  (cost=0.00..238.12 rows=21577 width=0) (actual time=0.834..0.834 rows=21468 loops=1)"
"              Index Cond: (""ТВ_ИД"" = 2)"
Planning Time: 0.231 ms
Execution Time: 11.477 ms
```

2.

```
Nested Loop  (cost=0.60..3719.79 rows=95191 width=25) (actual time=0.397..6.239 rows=5891 loops=1)
"  Join Filter: (nl.""ИД"" = nv.""ЧЛВК_ИД"")"
   ->  Nested Loop  (cost=0.29..283.14 rows=2190 width=21) (actual time=0.031..2.792 rows=2167 loops=1)
"        ->  Seq Scan on ""Н_СЕССИЯ"" ns  (cost=0.00..117.90 rows=3447 width=4) (actual time=0.011..0.761 rows=3445 loops=1)"
"              Filter: (""ДАТА"" > '2002-01-04 00:00:00'::timestamp without time zone)"
              Rows Removed by Filter: 307
        ->  Memoize  (cost=0.29..0.45 rows=1 width=17) (actual time=0.000..0.000 rows=1 loops=3445)
"              Cache Key: ns.""ЧЛВК_ИД"""
              Cache Mode: logical
```

```
                    Hits: 3267  Misses: 178  Evictions: 0  Overflows: 0
Memory Usage: 19kB
      "               ->  Index Scan using ""ЧЛВК_PK"" on ""Н_ЛЮДИ"" nl
(cost=0.28..0.44 rows=1 width=17) (actual time=0.003..0.003 rows=1
loops=178)"
      "                     Index Cond: (""ИД"" = ns.""ЧЛВК_ИД"")"
      "                     Filter: ((""ФАМИЛИЯ""):text > 'Ёлкин'::text)"
                           Rows Removed by Filter: 0
        ->  Memoize  (cost=0.30..6.51 rows=68 width=12) (actual
time=0.000..0.001 rows=3 loops=2167)
      "         Cache Key: ns.""ЧЛВК_ИД"""
              Cache Mode: logical
              Hits: 2035  Misses: 132  Evictions: 0  Overflows: 0  Memory
Usage: 52kB
      "         ->  Index Scan using ""ВЕД_ЧЛВК_FK_IFK"" on ""Н_ВЕДОМОСТИ""
nv  (cost=0.29..6.50 rows=68 width=12) (actual time=0.002..0.009 rows=8
loops=132)"
      "               Index Cond: (""ЧЛВК_ИД"" = ns.""ЧЛВК_ИД"")"
      "               Filter: (""ИД"" < 1490007)"
      Planning Time: 1.459 ms
      Execution Time: 6.634 ms
```

Возможное добавление индексов

1.

Добавление hash индекса для Н_ТИПЫ_ВЕДОМОСТЕЙ.ид так как производится операция =.

```
CREATE INDEX "TV_ID_HASH_INDEX" ON public."Н_ТИПЫ_ВЕДОМОСТЕЙ" USING hash
("ИД")
```

Добавление btree индекса для Н_ВЕДОМОСТИ.ЧЛВК_ИД так как производится операция >.

```
"CREATE INDEX ""NV_BTREE_PEOPLE_ID"" ON public.""Н_ВЕДОМОСТИ"" USING btree
(""ЧЛВК_ИД"")"
```

Добавление hash индекса для ВЕДОМОСТИ.ТВ_ИД так как производится операция =.

```
"CREATE INDEX ""NV_HASH_TV_ID"" ON public.""Н_ВЕДОМОСТИ"" USING hash
(""ТВ_ИД"")"
```

2.

Добавление hash индекса для Н_ЛЮДИ.ид так как производится операция =.

Добавление hash индекса для Н_ВЕДОМОСТИ.ЧЛВК_ИД так как производится операция =.

Добавление hash индекса для Н_СЕССИЯ.ЧЛВК_ИД так как производится операция =.

Добавление btree индекса для Н_ЛЮДИ.ФАМИЛИЯ так как производится операция >.
Добавление btree индекса для Н_ТИПЫ_ВЕДОМОСТЕЙ.ид так как производится операция <.
Добавление btree индекса для Н_СЕССИЯ.ДАТА так как производится операция >.

Существующие индексы

Н_ТИПЫ_ВЕДОМОСТЕЙ

```
CREATE UNIQUE INDEX "ТВ_PK" ON public."Н_ТИПЫ_ВЕДОМОСТЕЙ" USING btree
("ИД")
```

Н_ВЕДОМОСТИ

```
"CREATE UNIQUE INDEX ""ВЕД_PK"" ON public.""Н_ВЕДОМОСТИ"" USING btree
(""ИД"")"
"CREATE INDEX ""ВЕД_ДАТА_I"" ON public.""Н_ВЕДОМОСТИ"" USING btree
(""ДАТА"")"
"CREATE INDEX ""ВЕД_ИП_FK_I"" ON public.""Н_ВЕДОМОСТИ"" USING btree
(""СЭС_ИД"")"
"CREATE INDEX ""ВЕД_ОТД_I"" ON public.""Н_ВЕДОМОСТИ"" USING btree
(""ОТД_ИД"")"
"CREATE INDEX ""ВЕД_ОЦЕНКА_I"" ON public.""Н_ВЕДОМОСТИ"" USING btree
(""ОЦЕНКА"")"
"CREATE INDEX ""ВЕД_ТВ_FK_I"" ON public.""Н_ВЕДОМОСТИ"" USING btree
(""ТВ_ИД"")"
"CREATE INDEX ""ВЕД_ЧЛВК_FK_IFK"" ON public.""Н_ВЕДОМОСТИ"" USING btree
(""ЧЛВК_ИД"")"
```

Н_ЛЮДИ

```
"CREATE UNIQUE INDEX ""ИНН_UK"" ON public.""Н_ЛЮДИ"" USING btree (""ИНН"")"
"CREATE UNIQUE INDEX ""ПИН_UK"" ON public.""Н_ЛЮДИ"" USING btree (""ПИН"")"
"CREATE INDEX ""ФАМ_ЛЮД"" ON public.""Н_ЛЮДИ"" USING btree (""ФАМИЛИЯ"")"
"CREATE UNIQUE INDEX ""ЧЛВК_PK"" ON public.""Н_ЛЮДИ"" USING btree (""ИД"")"
```

Н_СЕССИЯ

```
"CREATE INDEX ""SYS_C003500_IFK"" ON public.""Н_СЕССИЯ"" USING btree
(""ЧЛВК_ИД"")"
"CREATE INDEX ""СЕС_СЭС_FK"" ON public.""Н_СЕССИЯ"" USING btree
(""СЭС_ИД"")"
```

Доп. Задания

Вызов плана с запросом, использующий оконные функции.

```
EXPLAIN ANALYZE
SELECT
    help.num, help."ГРУППА", help."ФАМИЛИЯ", help."НАИМЕНОВАНИЕ",
help."ОЦЕНКА",
    sum(help."ОЦЕНКА"::integer) OVER (PARTITION BY help."ГРУППА",
help."ФАМИЛИЯ" ORDER BY help.num) AS sum_оценка,
    sum(help."ОЦЕНКА"::integer) OVER (PARTITION BY help."ГРУППА" ORDER BY
help.num) AS sum_оценка_группа
FROM (
         SELECT
                     row_number() OVER (ORDER BY уч."ГРУППА", чл."ФАМИ-
ЛИЯ", дис."НАИМЕНОВАНИЕ") AS num,
                     уч."ГРУППА", чл."ФАМИЛИЯ", дис."НАИМЕНОВАНИЕ",
вед."ОЦЕНКА"
         FROM
             "Н_УЧЕНИКИ" уч
                 JOIN "Н_ЛЮДИ" чл ON уч."ЧЛВК_ИД" = чл."ИД"
                 JOIN "Н_ВЕДОМОСТИ" вед ON (вед."ЧЛВК_ИД" = чл."ИД" AND
вед."ОЦЕНКА" IN ('1', '2', '3', '4', '5'))
                 JOIN "Н_СОДЕРЖАНИЯ_ЭЛЕМЕНТОВ_СТРОК" сэс ON сэс."ИД" =
вед."СЭС_ИД"
                 JOIN "Н_ЭЛЕМЕНТЫ_СТРОК" эс ON эс."ИД" = сэс."ЭСТ_ИД"
                 JOIN "Н_СТРОКИ_ПЛАНОВ" спл ON спл."ИД" = эс."СПЛ_ИД"
                 JOIN "Н_ДИСЦИПЛИНЫ" дис ON дис."ИД" = спл."ДИС_ИД"
     ) AS help
ORDER BY help.num;
```

Запросы для вывода суммы оценок студента и суммы оценок всех студентов в каждой группе.

```
--Grouping sets
Explain
SELECT
    help."ГРУППА", help."ФАМИЛИЯ",
    sum(help."ОЦЕНКА"::integer) AS sum_оценка
FROM (
         SELECT
                     уч."ГРУППА", чл."ФАМИЛИЯ", вед."ОЦЕНКА"
         FROM
             "Н_УЧЕНИКИ" уч
                 JOIN "Н_ЛЮДИ" чл ON уч."ЧЛВК_ИД" = чл."ИД"
                 JOIN "Н_ВЕДОМОСТИ" вед ON (вед."ЧЛВК_ИД" = чл."ИД" AND
вед."ОЦЕНКА" IN ('1', '2', '3', '4', '5'))
                 JOIN "Н_СОДЕРЖАНИЯ_ЭЛЕМЕНТОВ_СТРОК" сэс ON сэс."ИД" =
вед."СЭС_ИД"
                 JOIN "Н_ЭЛЕМЕНТЫ_СТРОК" эс ON эс."ИД" = сэс."ЭСТ_ИД"
                 JOIN "Н_СТРОКИ_ПЛАНОВ" спл ON спл."ИД" = эс."СПЛ_ИД"
                 JOIN "Н_ДИСЦИПЛИНЫ" дис ON дис."ИД" = спл."ДИС_ИД"
     ) AS help
group by Grouping Sets ((help."ФАМИЛИЯ", help."ГРУППА"), (help."ГРУППА"))
ORDER BY help."ГРУППА", help."ФАМИЛИЯ";
```

```
--CUBE
Explain
SELECT
    help."ГРУППА", help."ФАМИЛИЯ",
    sum(help."ОЦЕНКА"::integer) AS sum_оценка
FROM (
         SELECT
             уч."ГРУППА", чл."ФАМИЛИЯ", вед."ОЦЕНКА"
         FROM
             "Н_УЧЕНИКИ" уч
                 JOIN "Н_ЛЮДИ" чл ON уч."ЧЛВК_ИД" = чл."ИД"
                 JOIN "Н_ВЕДОМОСТИ" вед ON (вед."ЧЛВК_ИД" = чл."ИД" AND
вед."ОЦЕНКА" IN ('1', '2', '3', '4', '5'))
                 JOIN "Н_СОДЕРЖАНИЯ_ЭЛЕМЕНТОВ_СТРОК" сэс ON сэс."ИД" =
вед."СЭС_ИД"
                 JOIN "Н_ЭЛЕМЕНТЫ_СТРОК" эс ON эс."ИД" = сэс."ЭСТ_ИД"
                 JOIN "Н_СТРОКИ_ПЛАНОВ" спл ON спл."ИД" = эс."СПЛ_ИД"
                 JOIN "Н_ДИСЦИПЛИНЫ" дис ON дис."ИД" = спл."ДИС_ИД"
     ) AS help
group by CUBE(help."ФАМИЛИЯ", help."ГРУППА")
ORDER BY help."ГРУППА", help."ФАМИЛИЯ";


--ROLLUP
EXPLAIN
SELECT
    help."ГРУППА", help."ФАМИЛИЯ",
    sum(help."ОЦЕНКА"::integer) AS sum_оценка
FROM (
         SELECT
             уч."ГРУППА", чл."ФАМИЛИЯ", вед."ОЦЕНКА"
         FROM
             "Н_УЧЕНИКИ" уч
                 JOIN "Н_ЛЮДИ" чл ON уч."ЧЛВК_ИД" = чл."ИД"
                 JOIN "Н_ВЕДОМОСТИ" вед ON (вед."ЧЛВК_ИД" = чл."ИД" AND
вед."ОЦЕНКА" IN ('1', '2', '3', '4', '5'))
                 JOIN "Н_СОДЕРЖАНИЯ_ЭЛЕМЕНТОВ_СТРОК" сэс ON сэс."ИД" =
вед."СЭС_ИД"
                 JOIN "Н_ЭЛЕМЕНТЫ_СТРОК" эс ON эс."ИД" = сэс."ЭСТ_ИД"
                 JOIN "Н_СТРОКИ_ПЛАНОВ" спл ON спл."ИД" = эс."СПЛ_ИД"
                 JOIN "Н_ДИСЦИПЛИНЫ" дис ON дис."ИД" = спл."ДИС_ИД"
     ) AS help
group by ROLLUP(help."ГРУППА", help."ФАМИЛИЯ")
ORDER BY help."ГРУППА", help."ФАМИЛИЯ";
```

Функция для создания таблицы с возможностью добавления столбца с Foreign Key.

```
create or replace procedure s372819.createtablewithfk(IN table_name char-
acter varying, IN column_name character varying DEFAULT NULL::character
varying, IN foreign_key_table_name character varying DEFAULT NULL::charac-
ter varying, IN foreign_key_column_name character varying DEFAULT
NULL::character varying)
```

```
    language plpgsql
as
$$
BEGIN
    EXECUTE 'CREATE TABLE IF NOT EXISTS ' || table_name || ' (ID INTEGER
PRIMARY KEY);';

    IF column_name IS NOT NULL AND foreign_key_table_name IS NOT NULL AND
foreign_key_column_name IS NOT NULL THEN
        EXECUTE 'ALTER TABLE ' || table_name || ' ADD COLUMN ' || col-
umn_name || 'INTEGER REFERENCES ' || foreign_key_table_name ||'('|| for-
eign_key_column_name|| ');';
    END IF;
END
$$;
```

Заключение

В процессе выполнения лабораторной работы, были приобретены значительные знания о индексах и их важности в оптимизации запросов в PostgreSQL. Индексы являются мощным инструментом для ускорения выполнения запросов, позволяя базе данных быстро находить и выбирать нужные данные. Также я приобрёл знания о команде EXPLAIN и параметр ANALYZE которые являются важными инструментами для анализа и оптимизации выполнения запросов.