Кафедра вычислительной техники
Рефакторинг баз данных и приложений

Рефакторинг мобильного приложения “Nomeragram”
Этап 3

Преподаватель:
Логинов Иван Павлович

Выполнили:
Полуянов Александр Михайлович
Девяткин Арсений Юрьевич
P34141

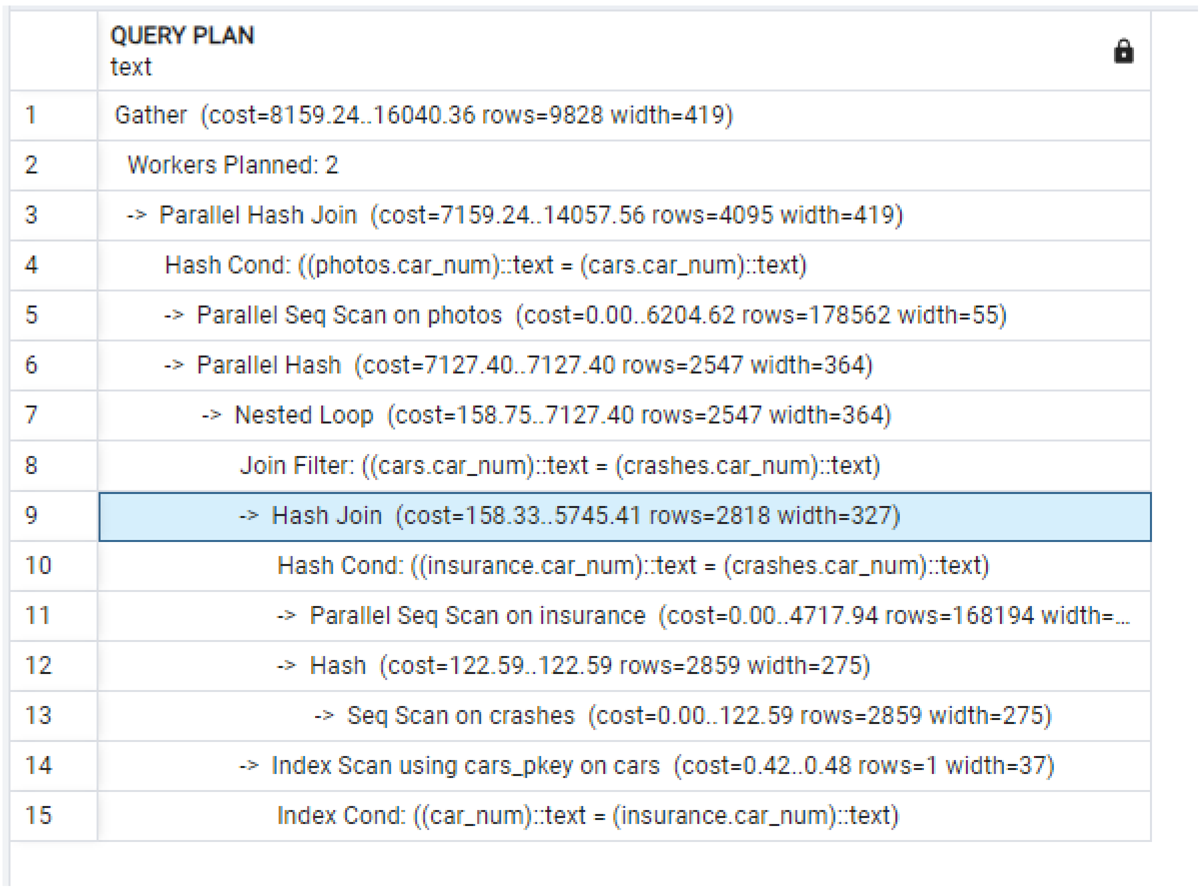
Санкт-Петербург
2024

В данной итерации пытались ускорить работу БД сделали 2 процедуры для ускорения работы.

```
-- Функция для получения всех фото по номеру
CREATE OR REPLACE FUNCTION get_car_photos(car_id VARCHAR)
RETURNS TABLE (
  photo_id INT,
  link VARCHAR(256),
  date DATE
) AS $$
BEGIN
  RETURN QUERY
  SELECT
    photos.id AS photo_id,
    photos.link,
    photos.date
  FROM photos
  WHERE photos.car_num = car_id;
END;
$$ LANGUAGE plpgsql;

-- Функция для получения всех аварий по номеру
CREATE OR REPLACE FUNCTION get_car_crashes(car_id VARCHAR)
RETURNS TABLE (
  id INT,
  date DATE,
  description text
) AS $$
BEGIN
  RETURN QUERY
  SELECT
    crashes.id AS crash_id,
    crashes.date,
    crashes.description
  FROM photos
  WHERE crashes.car_num = car_id;
END;
$$ LANGUAGE plpgsql;
```

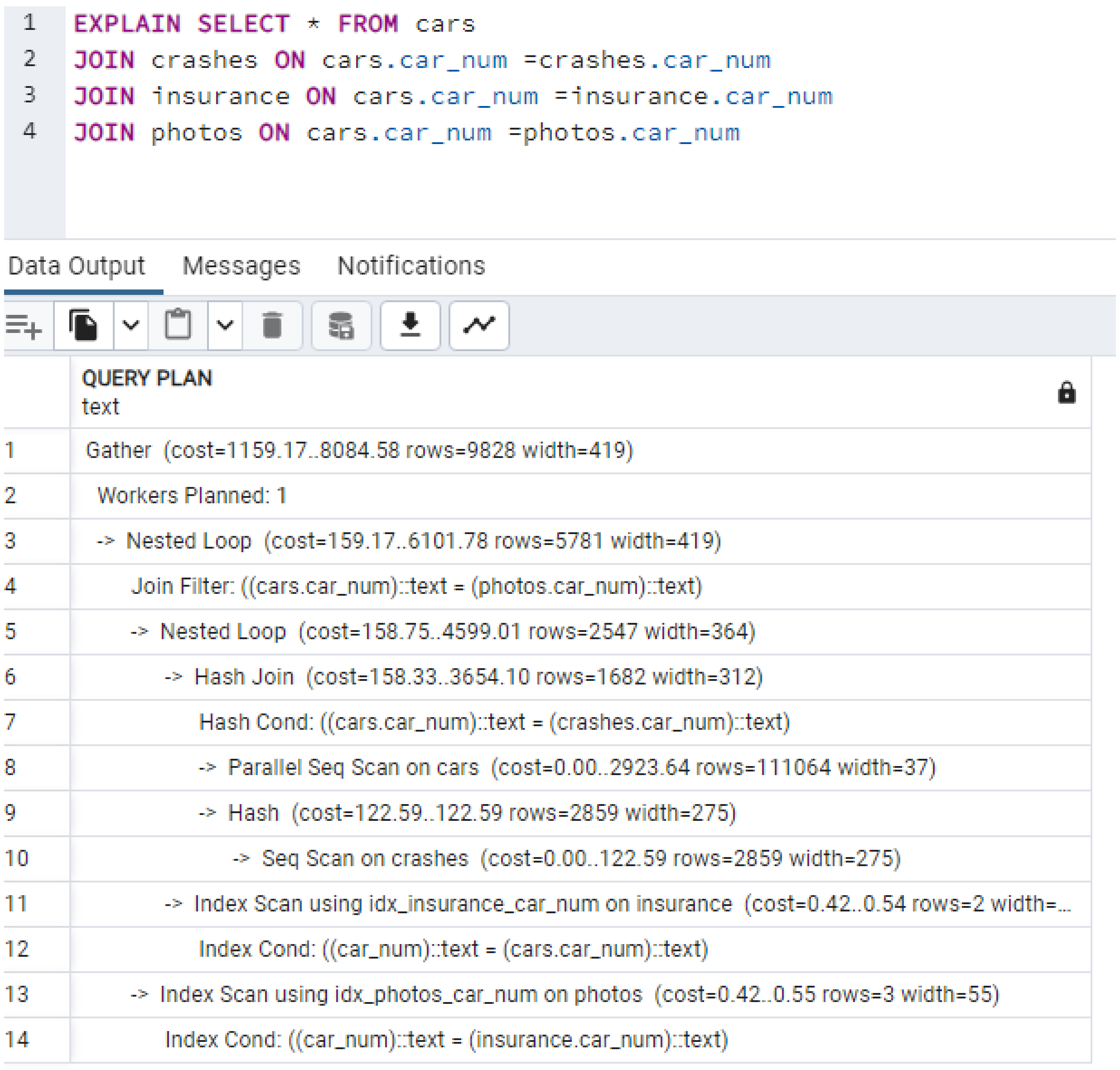
Также с помощью EXPLAIN проанализировали запросы и добавили индексы для их ускорения.

```
CREATE INDEX idx_cars_car_num ON cars (car_num);
CREATE INDEX idx_cars_year_of_issue ON cars (year_of_issue);
CREATE INDEX idx_cars_color_body_type ON cars (color,
car_body_type);

CREATE INDEX idx_photos_car_num ON photos (car_num);
CREATE INDEX idx_photos_date ON photos (date);

CREATE INDEX idx_insurance_car_num ON insurance (car_num);
CREATE INDEX idx_insurance_date_range ON insurance (start_date,
end_date);
CREATE INDEX idx_insurance_company ON insurance (company);

CREATE INDEX idx_crashes_car_num ON crashes (car_num);
CREATE INDEX idx_crashes_date ON crashes (date);
```



```
QUERY PLAN
text
Gather  (cost=8159.24..16040.36 rows=9828 width=419)
  Workers Planned: 2
  ->  Parallel Hash Join  (cost=7159.24..14057.56 rows=4095 width=419)
        Hash Cond: ((photos.car_num)::text = (cars.car_num)::text)
        ->  Parallel Seq Scan on photos  (cost=0.00..6204.62 rows=178562 width=55)
        ->  Parallel Hash  (cost=7127.40..7127.40 rows=2547 width=364)
              ->  Nested Loop  (cost=158.75..7127.40 rows=2547 width=364)
                    Join Filter: ((cars.car_num)::text = (crashes.car_num)::text)
                    ->  Hash Join  (cost=158.33..5745.41 rows=2818 width=327)
                          Hash Cond: ((insurance.car_num)::text = (crashes.car_num)::text)
                          ->  Parallel Seq Scan on insurance  (cost=0.00..4717.94 rows=168194 width=...
                          ->  Hash  (cost=122.59..122.59 rows=2859 width=275)
                                ->  Seq Scan on crashes  (cost=0.00..122.59 rows=2859 width=275)
                    ->  Index Scan using cars_pkey on cars  (cost=0.42..0.48 rows=1 width=37)
                          Index Cond: ((car_num)::text = (insurance.car_num)::text)
```

Раньше запрос на полную выборку проходил за 0.7 сек



```sql
EXPLAIN SELECT * FROM cars
JOIN crashes ON cars.car_num =crashes.car_num
JOIN insurance ON cars.car_num =insurance.car_num
JOIN photos ON cars.car_num =photos.car_num
```

```
QUERY PLAN
text
Gather  (cost=1159.17..8084.58 rows=9828 width=419)
  Workers Planned: 1
  ->  Nested Loop  (cost=159.17..6101.78 rows=5781 width=419)
        Join Filter: ((cars.car_num)::text = (photos.car_num)::text)
        ->  Nested Loop  (cost=158.75..4599.01 rows=2547 width=364)
              ->  Hash Join  (cost=158.33..3654.10 rows=1682 width=312)
                    Hash Cond: ((cars.car_num)::text = (crashes.car_num)::text)
                    ->  Parallel Seq Scan on cars  (cost=0.00..2923.64 rows=111064 width=37)
                    ->  Hash  (cost=122.59..122.59 rows=2859 width=275)
                          ->  Seq Scan on crashes  (cost=0.00..122.59 rows=2859 width=275)
              ->  Index Scan using idx_insurance_car_num on insurance  (cost=0.42..0.54 rows=2 width=...
                    Index Cond: ((car_num)::text = (cars.car_num)::text)
        ->  Index Scan using idx_photos_car_num on photos  (cost=0.42..0.55 rows=3 width=55)
              Index Cond: ((car_num)::text = (insurance.car_num)::text)
```

Теперь запрос на полную выборку проходит за 0.019 сек