SQL Queries

1. Add template_id B-tree index to the Comment table to speed up the fetch

1.2 Create Temp Table and save duplicate template data with comments

1.3 Add two fields to the template for the sound result of its sections and comments plus save and update the number of comments and sections in the creation and deletion of sections and comments

1) also, we can use INNER JOIN and CREATE VIEW but as data grows it's not gonna work slow
2) I'm trying to avoid using the COUNT (its cost is high) function unless it gets necessary

```
WITH template_list AS (
    SELECT id,
        COUNT(template_id) section_count
    FROM   rental
    GROUP  BY staff_id
)
SELECT templates,
    name,
    section
FROM templates
    INNER JOIN template_list USING (template_id);
```

2. Use Load More pagination for templates in Homepage

User LIMIT and OFFSET

```
EXPLAIN (ANALYZE) SELECT id, name, section_count, comment_count FROM templates
WHERE users.id = 1 ORDER BY 'created_at' OFFSET ((page * per) - per)  LIMIT 10;
```

SELECT * FROM templates WHERE

3. Use EXPLAIN query plan with ANALYZE key to finding slow queries

```
EXPLAIN (ANALYZE) SELECT id, name, section_count, comment_count FROM templates WHERE users.id = 1
```

5. Use pg_stat_* modules to check queries activities

EX: The pg_stat_statements module is used for tracking the execution statistics of SQL statements

6. Run VACUUM periodically to clean up obsolete data and reclaims storage occupied by dead tuples

To clean a single table `template`, analyze it for the optimizer, and print a detailed vacuum activity report:

```
VACUUM (VERBOSE, ANALYZE) templates;
```

```
SELECT * FROM templates WHERE user_id = $user_id
SELECT COUNT(*) FROM sections WHERE template_id = $template_id
SELECT COUNT(*) FROM comments WHERE template_id = 1 AND section_id = $section_id
```

7. Use auto_explain is used for logging execution plans of slow statements automatically

```
     LOAD 'auto_explain';
     SET auto_explain.log_min_duration = 0
     SET auto_explain.log_analyze = true;
```

Dependencies

System Dependencies

Ubuntu Server

Load Balancer
PG Server Cluster
Index Cluster

Application Dependencies

Apps cases

Devise
Devise-Jwt
Rack-cors
Virtus
Interactor
Active_model_serializers
Hattparty
PG
Puma

Test Cases

Factory_bot_rails
RSpes_rails
Faker

Third-Party Services

Stripe