

Итоговый проект по SQL.

Выполнил: Александр Косолапов

БД состоит из 5 таблиц, логически связанных между собой.

Данные собраны в сервисе [Hazanda.com](https://hazanda.com) - интернет-площадка формата хелпдеск, где любой желающий может оставить заявку(тикет) на распознавание, зачастую долго разыскиваемого, музыкального трека в любом виде (текст, ссылка, файл или сделать аудиозапись на сайте). Музыка, как сегмент, выбрана из-за очевидной проблематики распознавания треков различными сервисами. В сервисе работают эксперты - меломаны с феноменальной памятью, способные определять названия треков, которые, например не смог определить Shazam. Идеальная модель сервиса может выглядеть, как универсальный сервис для разнородных типов тикетов, которые обрабатываются не только экспертами, но технологиями цифровых отпечатков внутренней базы, так и через API ко внешним источникам, типа echonest. Сервис находится на пилотной стадии и развивается в свободное время.

Структура таблиц в БД:

```
[postgres=# \d tickets
```

Table "public.tickets"				
Column	Type	Collation	Nullable	Default
id	bigint			
has_attachment	boolean			
status	character varying(255)			
text	character varying(5000)			
link	character varying(255)			
user_id	bigint			
resolved_by_id	bigint			
target_id	bigint			
visits_count_admin	integer			
visits_count_expert	integer			
visits_count_user	integer			

```
[postgres=# \d users
```

Table "public.users"				
Column	Type	Collation	Nullable	Default
id	bigint			
email	character varying(255)			
login	character varying(255)			
mailing_enable	boolean			
name	character varying(255)			
status	character varying(255)			
type	character varying(255)			
oauth_source	character varying(255)			

```
[postgres=# \d notices
```

Table "public.notices"				
Column	Type	Collation	Nullable	Default
id	bigint			
ticket_id	bigint			
related_type	character varying(255)			
text	character varying(5000)			
type	character varying(255)			
user_id	bigint			

```
[postgres=# \d chat_messages
```

Table "public.chat_messages"				
Column	Type	Collation	Nullable	Default
id	bigint			
text	character varying(5000)			
author_id	bigint			
chat_id	bigint			

```
[postgres=# \d chat
```

Table "public.chat"				
Column	Type	Collation	Nullable	Default
id	bigint			
target_id	bigint			
ticket_id	bigint			

SQL-запросы:

1) Получение статистики результатов работы экспертов с расчетом премий за проделанную работу за учетный период при бюджете 5 000 рублей :

WITH temp as (SELECT resolved_by_id as expert_id, count(id) as count FROM tickets WHERE resolved_by_id IS NOT NULL GROUP BY resolved_by_id) SELECT name, expert_id, count, round(100*count/(SELECT sum(count) FROM temp), 2) as percent, round(5000*(count/(SELECT sum(count) FROM temp))) as earnings FROM temp JOIN users ON temp.expert_id=users.id ORDER BY count DESC;

name	expert_id	count	percent	earnings
Кир	833	2021	44.06	2203
Melody	712	1054	22.98	1149
M3mph1s	43	681	14.85	742
Dis	788	270	5.89	294
Настена	60	223	4.86	243
Annie	708	58	1.26	63
Franz	99	46	1.00	50
Юрий	5	41	0.89	45
Александр	2	34	0.74	37
Ishtar	711	34	0.74	37
Станислав	22	25	0.55	27
Камиль	723	24	0.52	26
Plaxou	716	20	0.44	22
Icluber	763	12	0.26	13
Ivy Mike	1	9	0.20	10
	220	6	0.13	7
Анастасия	819	4	0.09	4
Владимир	776	3	0.07	3
Monah	791	3	0.07	3
Виталий	775	2	0.04	2
	766	2	0.04	2
Никита	727	2	0.04	2
Nikita	357	2	0.04	2
AlexandrS	721	2	0.04	2
Шульц	1238	1	0.02	1
Phoenix	255	1	0.02	1
Анжелика	736	1	0.02	1
Глеб Родин	200	1	0.02	1
Гаврош	778	1	0.02	1
Sonny	45	1	0.02	1
Павел	7	1	0.02	1
Бодя	838	1	0.02	1
Настя	668	1	0.02	1

Бизнес-задача: Определение самых эффективных экспертов и расчет денежных премий, необходимых для мотиваций и развития экспертсорсинга.

2) Получение списка id экспертов, администраторов и юзеров, которые были наиболее активными в чатах тикетов за учетный период:

with t1 as (select author_id, count(author_id) as chat_messages FROM chat_messages GROUP BY author_id) select author_id, type, chat_messages FROM t1 JOIN users ON t1.author_id=users.id ORDER BY chat_messages DESC LIMIT 20;

author_id	type	chat_messages
833	EXPERT	7895
712	EXPERT	3262
43	EXPERT	1649
788	EXPERT	716
60	EXPERT	652
2	ADMIN	406
7175	USER	241
1	ADMIN	220
711	EXPERT	181
99	EXPERT	169
2871	USER	166
5	EXPERT	164
708	EXPERT	137
716	USER	129
7190	USER	114
1051	USER	107
22	EXPERT	92
6247	USER	70
6044	USER	58
723	EXPERT	55

Бизнес-задача: Анализ диалогов с целью поиска инсайтов по развитию системы чатов, а также выработки правил общения для экспертов.

3) Получение списка чатов с наибольшим количеством сообщений.

```
select chat_id, count(chat_id) as count FROM chat_messages GROUP BY chat_id ORDER BY count DESC LIMIT 10;
```

chat_id	count
10146	122
10049	68
3578	68
10110	64
10062	53
18	53
366	49
3579	35
1917	35
7106	33

Бизнес-задача: поиск инсайтов для расширения возможностей сервиса элементами соцсети/мессенджера в первую очередь для возможной мобильной версии сервиса.

4)Получение списков самых просматриваемых тикетов, которые просматривали: а) эксперты, б) юзеры.

```
SELECT id, status, text, visits_count_expert, visits_count_user FROM tickets WHERE visits_count_expert IS NOT NULL ORDER BY visits_count_expert DESC LIMIT 10;
```

```
SELECT id, status, text, visits_count_user, visits_count_expert FROM tickets WHERE visits_count_user IS NOT NULL ORDER BY visits_count_user DESC LIMIT 10;
```

```
postgres=# SELECT id, status, text, visits_count_expert, visits_count_user FROM tickets WHERE visits_count_expert IS NOT NULL ORDER BY visits_count_expert
```

id	status	text	visits_count_expert	visits_count_user
2310	CLOSED	Помогите пжл найти песню из видео	35	10
3034	CLOSED	трек?	33	
2749	CLOSED	трек?	28	5
7106	REOPEN	Что за музыка играет фоном с 52сек по 2:40 - https://youtu.be/iijkJLpnLKI?t=52s	26	47
1441	CLOSED	Подскажите пожалуйста =)	25	2
1600	OPEN	Сериал Бесстыжие 3 сезон 10 серия песня на 17-ой минуте когда Лип пришел к Карен в больницу	25	7
2549	RESOLVED	можно трек	25	8
1610	OPEN	Н	24	5
1565	RESOLVED	Нужен трек которая начинается на 22-ой минуте фильма за ранее спасибо) 9 серия	24	23
1795	CLOSED		23	29

(10 rows)

```
postgres=# SELECT id, status, text, visits_count_user, visits_count_expert FROM tickets WHERE visits_count_user IS NOT NULL ORDER BY visits_count_user DESC
```

id	status	text	visits_count_user	visits_count_expert
11645	OPEN	Здрасте мне бы узнать песнь с видео https://www.youtube.com/watch?v=IcmDEGbpMNU	126	
10967	OPEN	Трек из одного видео от 0:03 до 0:19	104	
4638	CLOSED	Заранее благодарю! :*	103	3
6497	OPEN	Какая песня играет на фоне? https://www.youtube.com/watch?v=ukRI65Zygro	96	3
6123	RESOLVED	помогите пожалуйста найти трек. (ссылка прикреплена)	96	2
2507	CLOSED		82	5
2904	OPEN	Что за музыка с 4 секунды видео. https://www.youtube.com/watch?v=SeNuif-pgao	76	8
3296	OPEN	трек	73	3
9832	CLOSED	типа этот)	72	2
5178	OPEN	В конце это аудиозаписи есть отрывок где поет девушка хотелось бы узнать название	68	9

(10 rows)

Бизнес-задача: б) анализ тикетов с целью определения причин высокой посещаемости тикетов пользователями, а не экспертами. Определение природы высокой вовлеченности.

5) Получение распределения авторизаций в сервисе через профили соцсетей уникальными пользователями:

```
SELECT oauth_source, count(oauth_source) FROM users WHERE oauth_source IS NOT NULL GROUP BY oauth_source;
```

oauth_source	count
VK	1790
FB	473

(2 rows)

Бизнес-задача: Отслеживание динамики изменения соотношения при изменении рекламных и маркетинговых кампаний продвижения сервиса в данных соцсетях.

6) Получение списка наиболее сложных тикетов, которые протеггировали более чем один эксперт:

```
ALTER TABLE tickets RENAME COLUMN resoved_by_id TO resolved_by_id;
```

```
DROP VIEW ranks, places;
```

```
CREATE VIEW places AS
```

```
SELECT resolved_by_id as expert_id, count(id) as count FROM tickets WHERE resolved_by_id IS NOT NULL GROUP BY resolved_by_id ORDER BY count DESC;
```

```
CREATE VIEW ranks AS
```

```
SELECT name, expert_id, count, row_number() over (order by count asc) as place FROM places JOIN users ON places.expert_id=users.id;
```

```
with t1 as (SELECT * FROM notices) SELECT ticket_id, count(type) as count_tag, array_agg(type) as types, array_agg(text) as notices, array_agg(user_id) as experts, round(avg(place),1) as difficulty FROM t1 JOIN ranks ON user_id=expert_id WHERE type <> 'SOLVED' GROUP BY ticket_id ORDER BY count_tag DESC, difficulty DESC LIMIT 200;
```


hard_ticket	count_tag	types	notices	experts	difficulty
4719	2	{WAITING, HARD}	{Отложено, Затрудняюсь}	{43, 712}	31.5
4981	2	{WAITING, OBSCURE}	{Отложено, Непонятно}	{43, 712}	31.5
4385	2	{HARD, WAITING}	{Затрудняюсь, Отложено}	{712, 43}	31.5
4727	2	{HARD, WAITING}	{Затрудняюсь, Отложено}	{712, 43}	31.5
942	2	{OBSCURE, OBSCURE}	{Непонятно, "Непонятно пустышка"}	{788, 712}	31.0
1991	2	{WAITING, OBSCURE}	{Отложено, Непонятно}	{708, 712}	30.0
910	2	{WAITING, HARD}	{Отложено, хз}	{99, 712}	29.5
873	2	{WAITING, HARD}	{"S.T.A.L.K.E.R. WIKI", хз}	{99, 712}	29.5
917	2	{HARD, WAITING}	{хз, "Холостяк 4"}	{712, 99}	29.5
823	2	{WAITING, HARD}	{эпик, эпическая}	{99, 712}	29.5

Бизнес-задача: Вывести разделе “Угадай трек” в виде маркетплейса с ценниками на сложные тикеты. Цель данного раздела увеличить посещаемость сервиса (привлечь трафик меломанов), а также определить потенциальных экспертов среди самых активных юзеров.

7) Получение списка почтовых сервисов пользователей.

with tmp as (select substring(email from '@(.*)\$'), count(*) from users group by substring(email from '@(.*)\$') ORDER by count DESC) select *, round(100*count/(sum(count) over())) as percent FROM tmp LIMIT 10;

substring	count	percent
mail.ru	5887	46
gmail.com	2838	22
yandex.ru	1531	12
bk.ru	553	4
inbox.ru	250	2
list.ru	241	2
	227	2
rambler.ru	154	1
ukr.net	96	1
mail.ru	84	1
(10 rows)		

Бизнес-задача: определить важность той или иной почтовой службы при выявлении случаев, когда письма из сервиса не доходят или попадают в спам. Пересмотр беклога проблем.

8) Получение списка пользователей, создавших больше всего тикетов.

WITH t1 as (SELECT id as id1, user_id as us1 FROM tickets) SELECT us1 as user, count(id1) FROM t1 JOIN users ON us1=users.id GROUP BY us1 ORDER BY count(id1) DESC LIMIT 20;

user	count
1877	145
2871	142
1051	105
3283	92
324	56
6247	56
2470	55
1175	53
3052	46
323	44
4437	40
6044	40
6181	38
2667	37
3232	37
1457	37
988	35
1543	32
678	30
4883	29

Бизнес-задача: Определить девиантных пользователей на основе их “агрессивного” создания тикетов. Пересмотр механизма создания тикетов для автоматического предотвращения повторных заявок.

9) Получение списка пользователей, не создавших тикет в сервисе.

```
WITH tmp as (SELECT user_id FROM tickets) SELECT DISTINCT id FROM tmp RIGHT
JOIN users ON user_id=users.id WHERE user_id is null ORDER BY id asc LIMIT 10;
WITH tmp as (SELECT user_id FROM tickets) SELECT Count(DISTINCT id) as
Count_users_without_tickets FROM tmp RIGHT JOIN users ON user_id=users.id WHERE
user_id is null;
```

```
id
----
 6
 9
10
12
13
14
22
34
36
40
(10 rows)

postgres=# WITH tmp as (SELECT
count_users_without_tickets
-----
3419
```

Бизнес-задача: уведомить письмом тех, пользователей, которые зарегистрировались в сервисе, но не оставили тикет за учетный период.

10) Получение списка таблиц БД:

```
SELECT TABLE_NAME, pg_size_pretty(total_size) AS total_size
FROM (SELECT TABLE_NAME, pg_total_relation_size(TABLE_NAME) AS total_size
FROM (SELECT (" || table_schema || '.' || TABLE_NAME || ") AS TABLE_NAME FROM
information_schema.tables ) AS all_tables ORDER BY total_size DESC) AS ps LIMIT 20;
```

table_name	total_size
public.tickets	2744 kB
public.chat_messages	1920 kB
public.users	1184 kB
pg_catalog.pg_depend	1168 kB
pg_catalog.pg_proc	976 kB
public.chat	744 kB
pg_catalog.pg_rewrite	640 kB
pg_catalog.pg_attribute	624 kB
pg_catalog.pg_description	552 kB
public.notices	504 kB
pg_catalog.pg_collation	304 kB
pg_catalog.pg_statistic	296 kB
pg_catalog.pg_class	240 kB
pg_catalog.pg_operator	240 kB
pg_catalog.pg_amop	192 kB
pg_catalog.pg_type	176 kB
pg_catalog.pg_amproc	128 kB
pg_catalog.pg_constraint	112 kB
pg_catalog.pg_conversion	104 kB
pg_catalog.pg_index	96 kB

11) Получение списка тикетов с одинаковым текстом:

DROP VIEW duplicates;

CREATE VIEW duplicates as

SELECT (tickets.text)::text, count(*) FROM tickets GROUP BY tickets.text HAVING count(*) > 1 ORDER BY count(*) DESC;

SELECT * FROM duplicates LIMIT 30;

text	count
	544
	274
Подскажите трек	86
трек?	59
???	42
+	42
Что за трек?	34
трек	32
что за трек?	32
Трек	31
?	26
подскажите трек	20
Подскажите трек пожалуйста	20
Трек?	20
помогите найти трек	19
??	19
музыка	17
Что за песня?	17
Подскажите трек	17
Помогите найти трек	17
1	16
Здравствуйте.	15
помогите найти)	14
????	14
подскажите песню пожалуйста	14
найдите пожалуйста)	13
Будь со мной будь со мной	13
Подскажите трек	12
https://coub.com/view/8mmij	12
Какая музыка с 0:00 до 3:40 подскажите пожалуйста https://www.youtube.com/watch?v=82yA1Ra5Gh0	12

Бизнес-задача: Быстрое получение списка тикетов с одинаковым текстом.

PS: Текст справа порезан при подготовке csv-файлов для данного проекта, поэтому пропала уникальность и образовались агрегации высокочастотных фраз.