PostgreSQL credentials:

[redacted]

Task 1 - SQL

- **1.1 Select list of distinct users from table `ads` who posted ads in March 2020, but didn't post in April 2020.**
- **1.2 Write a query using table `ads`, which shows how many ads were posted in certain category and rank of this category for each user.** See explanation table below, if needed.

Expected table example task 1.2

user_id	category_id	category_name	ads	category_rank
27021488	1	children	25	1
27021488	2	hobby, rest and sport	10	2
27021488	3	home and garden	5	3
27021488	4	animals	1	4

Task 2 - R/Python

Using the PostgreSQL credentials, create a R/Python script following next few steps:

- **2.1 Connect to the database;**
- **2.2 Get data from query in task 1.2;**
- **2.3 Calculate users overlap matrix by categories;**

> Every user can post an ad in any category, you have to calculate HOW MANY DISTINCT USERS posted an ad in catagory A and at the same time posted ads in category B, C, D etc.

Expected table example:

category_name	A	B	C	D
A	1000	392	144	219
B	392	1500	500	121
C	144	500	700	65
D	219	121	65	2000

- **2.4 Create new Google Sheets workbook with name** `[OLX] <your_surname> <your_name> - <current_date>`**;**
- **2.5 Create new sheet with name** `Test task` **and send final data from task 2.3 to it.**

Task 3 - Google Sheets

Use GS Workbook generated in task 2.
As you might noticed - **data** from different sides of the main matrix diagonal **are the same**.

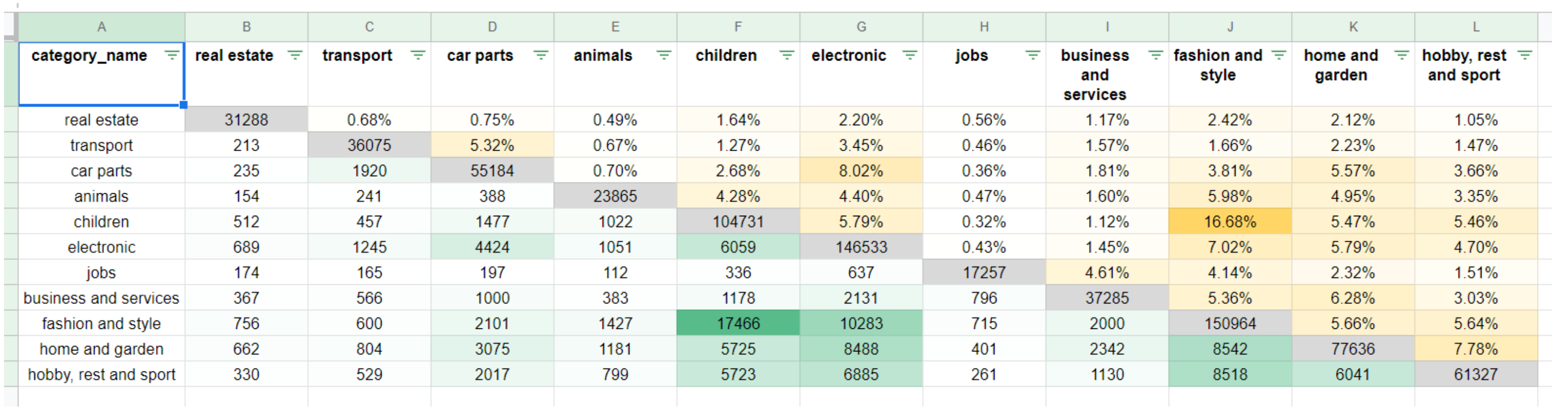
- **3.1** So let`s remove all numbers which are ABOVE of matrix main diagonal and write the UNIQUE formula(write only one and then copy-paste it for whole range) in cleaned cells to calculate share of overlapped users. See explanation table below, if needed.

Explanation table:

category_name	A	B	C	D
A	1000	392/1000=39.2%	14.4%	21.9%
B	392	1500	33.3%	121/1500=8.1%
C	144	500	700	9.3%
D	219	121	65	2000

- **3.2** Add color scale cells filling, different colors for 2 different sides of diagonal.
- **3.3** Send all your results (queries, code and GS link) to HR.

Expected table example:



A	B	C	D	E	F	G	H	I	J	K	L
category_name	**real estate**	**transport**	**car parts**	**animals**	**children**	**electronic**	**jobs**	**business and services**	**fashion and style**	**home and garden**	**hobby, rest and sport**
real estate	31288	0.68%	0.75%	0.49%	1.64%	2.20%	0.56%	1.17%	2.42%	2.12%	1.05%
transport	213	36075	5.32%	0.67%	1.27%	3.45%	0.46%	1.57%	1.66%	2.23%	1.47%
car parts	235	1920	55184	0.70%	2.68%	8.02%	0.36%	1.81%	3.81%	5.57%	3.66%
animals	154	241	388	23865	4.28%	4.40%	0.47%	1.60%	5.98%	4.95%	3.35%
children	512	457	1477	1022	104731	5.79%	0.32%	1.12%	16.68%	5.47%	5.46%
electronic	689	1245	4424	1051	6059	146533	0.43%	1.45%	7.02%	5.79%	4.70%
jobs	174	165	197	112	336	637	17257	4.61%	4.14%	2.32%	1.51%
business and services	367	566	1000	383	1178	2131	796	37285	5.36%	6.28%	3.03%
fashion and style	756	600	2101	1427	17466	10283	715	2000	150964	5.66%	5.64%
home and garden	662	804	3075	1181	5725	8488	401	2342	8542	77636	7.78%
hobby, rest and sport	330	529	2017	799	5723	6885	261	1130	8518	6041	61327

Task 4 - ADVANCED

For this part use table `advanced_task`.

Table data structure:

- `id` - unique identifier of the complaint;
- `ad_id` - unique identifier of ad;
- `content` - complaint's text;
- `type` - category of the complaint (predefined list of purposes);
- `ip` - hash of complaining user's IP-address;
- `checked_at` \ `checked_by` - was this complaint review by the OLX's employee with timestamp;
- some extra info-columns.

Task:

Could you, please, provide some insights about data? And could you please propose some actions, solutions or recommendations for business?

For example (but not limit yourself):

- What is the key reasons and phrases in complains type-wised?
- How fast is our reaction on complaint?
- Do we have any most frequent complainers?

Feel free to choose any tool, language or framework. As a result we're expecting presentation in pdf-format and the sources of your solution.