**Google play store apps**

In this project I'm going to evaluate different app attributes to decide which characteristics are better perceived by the market. Attributes that's going to be evaluated are app rating, rating count, category. Evaluation parameters are app's average number of reactions (count of rating scores assigned) and / or number installs, and correlation between rating and the percentage of rating count to installs. High number of rating count tells us about the demand for the application and high potential income. High number of rating count tells us about the demand for the application and high potential income indicates that the audience is more loyal to such type of applications.

The final result will be a full set of application characteristics, determined by indirect evidence.

My work consisted of several parts:

- search for a suitable dataset;

- loading the date of the set into the database in denormalized form;

- data processing;

- development and creation of a database schema;

- loading data into the database;

- writing SQL queries to the database;

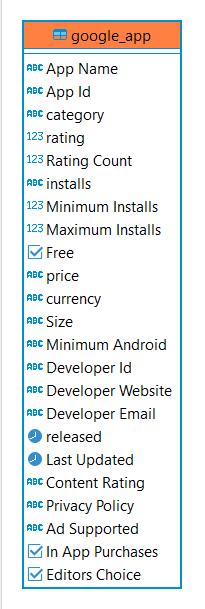
- data visualization.

**Search for a suitable dataset.**

I found the basic data for analysis on [kaggle](https://www.kaggle.com/gauthamp10/google-playstore-apps). The date set includes 23 columns with various information about google play store apps, and also contains more than 1 million lines, but I decided to cut it to 3,5 thousand lines.

**Loading the date of the set into the database in denormalized form.**

I downloaded all data from csv with help of automatic import of data into table in DBeaver. Tables in which the data was downloaded on the Pic.1.



Pic. 1: Table for insert data from csv

**Data processing.**

At the moment of processing the data, I faced a lot of problems. This stage took me about half of the time spent on the project. Sometimes, having passed this stage, I returned to it already in the process of data analysis. Typical problems:

- non-obvious duplicates;

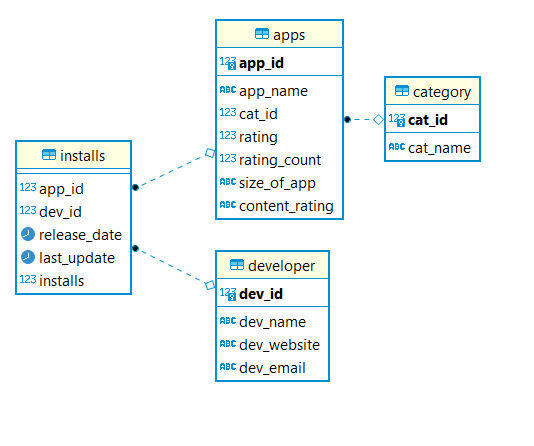
- unsuitable data type;

- NULL value in those rows where they should not logically be.

Data processing script is in .rar archive (DDL\_script.sql).

**Development and creation of a database schema.**

At this stage, I thought about which data from the dataset I might need, and which ones do not need to be stored. By this way i created the following db schema. (Pic.2)



Pic. 2: Table for insert data from csv

**Database description**

The created database has one schema. The database schema is a snowflake. It contains one fact table and 3 dimension tables. Description of the database in the table below.

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Column description** |
| category | cat\_id | PK |
| cat\_nane | description of the category |
| apps | app\_id | PK |
| app\_name | full name of the app |
| cat\_id | category id, references to category.cat\_id |
| rating | rating of the app |
| rating\_count | count of rating scores assigned |
| size\_of\_app | size of the app |
| content\_rating | age restrictions |
| developer | dev\_id | PK |
| dev\_name | full name of the developer |
| dev\_website | developer’s website |
| dev\_email | developer’s email |
| installs | app\_id | app Id, references apps.app\_id |
| dev\_id | developer Id, references developer.dev\_id |
| release\_date | released date of the app |
| last\_update | last update date of the app |
| installs | amount of installs |

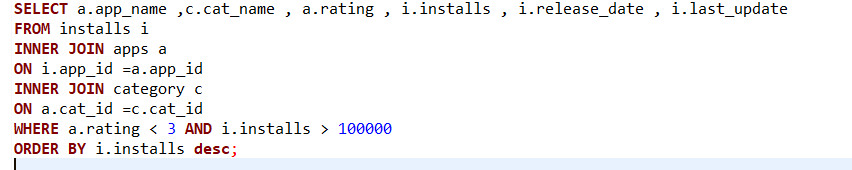
**Loading data into the database**

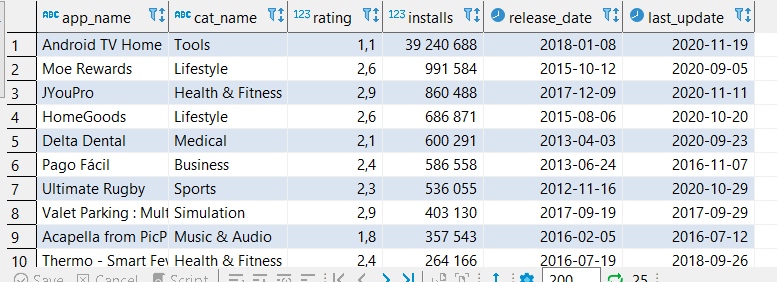
After creating the database, I loaded the data into it. Basically, to load data, I used "sourse id" columns.

**Writing SQL queries to the database and data visualization.**

All sql queries are in the file: DML\_Google\_apps.sql.

The first thing I did was get popular apps (100k+ downloads) with a bad rating (< 3). This way you can easily find the apps people are interested in, but dissapointed in the realization. And use that information for creating your own projects.(Pic. 3)



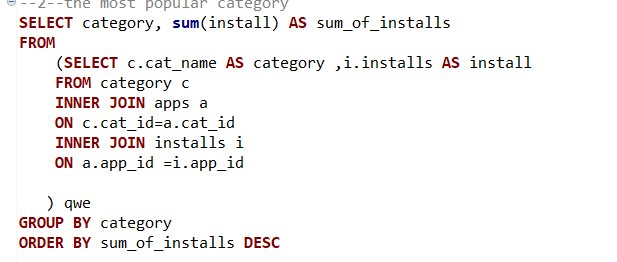


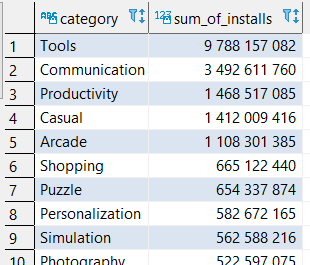
Pic. 3: popular apps with bad realization

**Conclusion:**

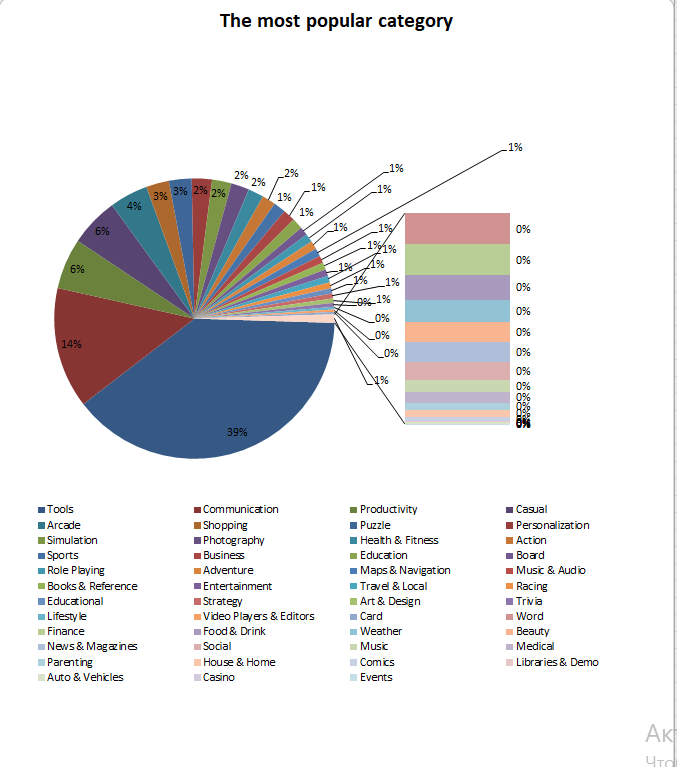
Surprisingly, the most downloaded app came out with the lowest rating, and this app is from the category of tools. Thus, we learned about the prospect of developing a quality application from the category of tools.

Next thing that I decide to account was what is the most popular app category by the amount of its installed. (Pic. 4 and 5)





Pic. 4: the most popular category

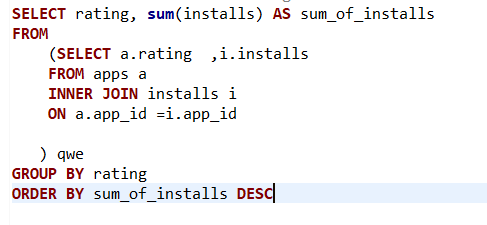


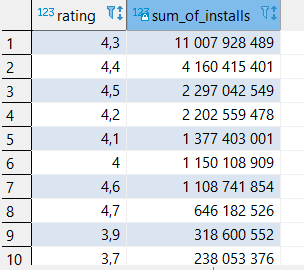
Pic. 5: Circle diagram of the most popular category

**Conclusion:**

The most popular and perspective app category is tools, communication apps are on the second place.

Then I counted the correlation between rating and installs, which will help us with the next question.(Pic. 6 )



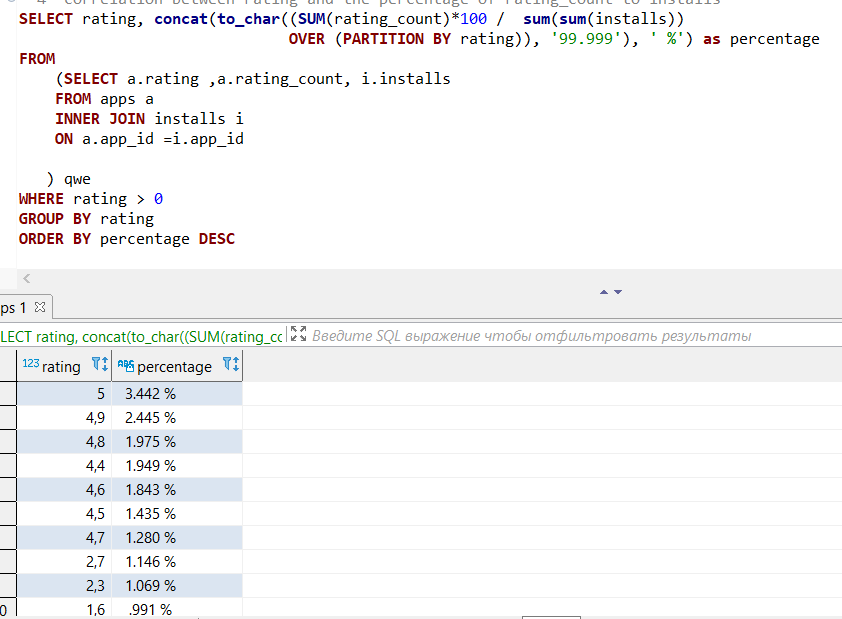


Pic. 6: correlation between rating and installs

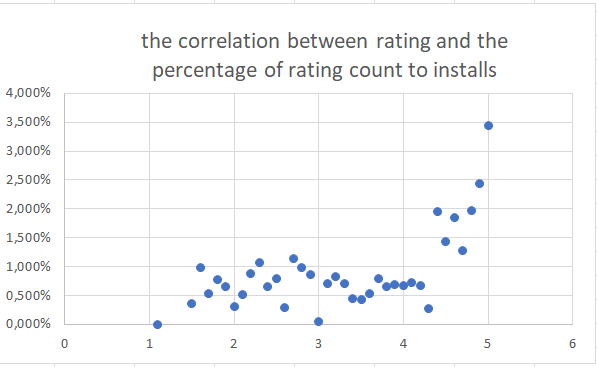
**Conclusion:**

In this query we can see that the most installed apps have pretty high rating.

At the end of the project, I decided to calculate what is the correlation between rating and the percentage of rating count to installs. (Pic. 7 and 8)



Pic. 7: correlation between rating and the percentage of rating count to installs



Pic. 8: scatter chart of correlation between rating and the percentage of rating count to installs

**Conclusion:**

In conclusion, we can note that the highest rated apps have the highest percentage of feedback, but the number of people who have rated the app is still very small relative to the number of installs.