

## Mobile Game Analysis

By [Liad Traube](#)



# Too Long : Didn't Read

### TL; DR

This project goal is to simulate a **mobile game database** and an analysis that is based on data from that database, including **KPI's and game economy**



### 1. Introduction

For many days I wandered around web-sites like "[Kaggle](#)" and "[Maven Analytics](#)" searching for interesting datasets that I could use to create a big and impressive project, downloading files, loading the datasets in MSSQL and Tableau, preparing queries in SQL but It was all for vain.



## Mobile Game Analysis

none of the datasets were good enough for my needs, from my point of view and it seemed to me a lot the datasets in Kaggle were tailored for data-science and machine learning projects and some of the datasets were simply not "rich" enough in data or did not have what I needed for the kind of project that I had in mind.



As a result of my frustration in finding a proper dataset I decided that I have to address this challenge and build a database myself.

As a consequence of having the creative freedom with the mission of building a database by myself I decided to choose a subject that I find interesting and I want to study more from the "developers" side.



## Mobile Game Analysis

As you will see in this project, I tried to simulate a real mobile game database as best as I could, using randomly generated data thanks to website like ["generate data"](#) and various functions in Microsoft Excel.

The database contains data about a mobile game called "Empire Of Crabs" (I tasked chat-GPT to generate me 10 names for mobile games and this was my favorite name).

game_id	game_name	published_date	age_ratings	languages	price	genre 1	genre 2	genre 3	size_mb
1	Empire Of Crabs	2019-08-10 00:00:00.000	4+	English, Hebrew	0	strategy	simulation	NULL	8493



\*This image was randomly found in google and was added to illustrate the idea, this image doesn't belong to me

## 2. Data Base Structure

### 2.1 Database Tables

The data is stored in 9 different tables

#### 1. GameInstall

Holds data about all the users that installed the game at least once

InstallID	user_id	install_date
1	6530	2020-11-07 00:00:00.000
2	11285	2020-01-28 00:00:00.000
3	848	2020-12-21 00:00:00.000
5	7396	2019-11-04 00:00:00.000
6	10694	2020-02-07 00:00:00.000

## Mobile Game Analysis

### 2. Users

Holds data about all the users that also opened an account (the other option is to sign in as "Guest")

user_id	date_registered	country	birth_date
10001	2021-12-29 00:00:00.000	Russian Federation	1992-09-01 00:00:00.000
10002	2020-08-10 00:00:00.000	Chile	2003-01-11 00:00:00.000
10003	2020-03-23 00:00:00.000	France	1967-09-12 00:00:00.000
10004	2020-01-30 00:00:00.000	Chile	1970-10-05 00:00:00.000
10005	2022-06-26 00:00:00.000	India	1967-04-04 00:00:00.000
10006	2021-09-02 00:00:00.000	Mexico	1966-09-18 00:00:00.000

### 3. Log In

Holds data about when different users used the app and for how long

log_id	log_in_date	log_off_date	minutes played	user_id
1	2022-04-05 00:00:00.000	2022-04-05 01:40:00.000	100	17969
3	2021-02-10 00:00:00.000	2021-02-10 01:20:00.000	80	10384
4	2021-10-22 00:00:00.000	2021-10-22 00:40:00.000	40	14386
5	2020-05-03 00:00:00.000	2020-05-03 01:37:00.000	97	8862

### 4. GameRatings

Holds data about what rating each user gave to the app (if he chose to do so)

rating_id	user_id	rating_date	rate
33	7260	2020-09-23 00:00:00.000	3
37	1997	2020-03-19 00:00:00.000	4
42	6747	2020-06-15 00:00:00.000	2
43	5500	2020-04-30 00:00:00.000	1

### 5. Marketing

Hold data about all the money that the company use to advertise the game

marketing_id	campaign	start_date	end_date	cost_per_month
1	facebook_ads	2021-01-01 00:00:00.000	2021-03-01 00:00:00.000	10
2	creative	2021-01-01 00:00:00.000	2021-12-31 00:00:00.000	5
3	app_store_fees	2021-01-01 00:00:00.000	2021-12-31 00:00:00.000	3

## Mobile Game Analysis

### 6. PremiumItems

Holds data about the special items or features that the user can purchase with real life money

premuim_item_id	premuim_item_name	item genre	cost
1	Green Crab	skin	2.3
2	Crown	power-up	3
3	Aurora Shell Shimmer	skin	4
4	Oceanic Elegance	skin	6.5
5	Enigmatic Moonstone Claw	power-up	2.3
6	Cursed Relic	power-up	4.99
7	Crystal Cavern Chateau	house	5.3

### 7. Purchases

Holds data about the purchases users made (using real life money)

purchase_id	purchased_item_id	user_purchased	purchase_date
1	15	12164	2020-06-05 00:00:00.000
25	15	5274	2020-08-20 00:00:00.000
50	15	5071	2020-05-28 00:00:00.000
52	15	4485	2020-03-01 00:00:00.000
55	15	6931	2020-07-09 00:00:00.000
57	15	4812	2020-01-09 00:00:00.000

### 7. game economy

Hold data about the digital in app currency inflows and outflows.

the digital currency we can call "CrabCoins", so the "CrabCoins" that were generated by various actions made and milestone achieved by the users actions in game and the "CrabCoins" that were used by the users and disappeared into the void (or sink, whatever term you like).

action_id	user_id	action_date	action_desc	effect
1	10001	2020-01-01 00:00:00.000	first_log_in reward	25000
2	10001	2020-01-01 00:00:00.000	battle	5000
3	10001	2020-01-01 00:00:00.000	purchased blue crab	-10000
4	10001	2020-01-01 00:00:00.000	purchased iron sword	-2500
5	10001	2020-01-01 00:00:00.000	battle	7500
6	10001	2020-01-01 00:00:00.000	purchased blue crab	-10000
7	10001	2020-01-01 00:00:00.000	battle	1250
8	10001	2020-01-01 00:00:00.000	battle	5000
9	10001	2020-01-01 00:00:00.000	milestone - tier 2 clan	50000
10	10001	2020-01-01 00:00:00.000	battle	7500
11	10002	2020-02-01 00:00:00.000	purchased blue crab	-10000
12	10002	2020-02-01 00:00:00.000	milestone - tier 3 clan	75000
13	10002	2020-02-01 00:00:00.000	found magic sea-shell	10000



## 2.2 Data Cleaning and Validation



As consequence of the data being randomly generated, I had to build queries in SQL to make sure that data make sense, for example:

a user cannot have data in the log-in table on a date that is earlier than the date he installed the app for the first time.

Installs table -

InstallID	user_id	install_date
1	6530	2020-11-07 00:00:00.000
2	11285	2020-01-28 00:00:00.000
3	848	2020-12-21 00:00:00.000

Log\_in table -

	user_id	log_in_date	log_off_date	minutes played
1	6530	2021-11-29 00:00:00.000	2021-11-29 00:28:00.000	28
2	6530	2019-12-20 00:00:00.000	2019-12-20 00:12:00.000	12
3	6530	2019-12-21 00:00:00.000	2019-12-21 00:32:00.000	32
4	11285	2021-02-23 00:00:00.000	2021-02-23 00:10:00.000	10
5	11285	2022-04-12 00:00:00.000	2022-04-12 00:43:00.000	43
6	11285	2020-12-22 00:00:00.000	2020-12-22 00:10:00.000	10

Query to validate the data -

```
WITH cte_first_install
AS
(
    SELECT user_id, min(install_date) AS "FirstInstall"
    FROM GameInstalls
    GROUP BY user_id
)
DELETE FROM Log_In WHERE log_id IN
(
    SELECT L.log_id
    FROM cte_first_install FI JOIN Log_In L ON FI.user_id = L.user_id
    WHERE FI.FirstInstall > L.log_in_date
)
```

### 3. Analysis

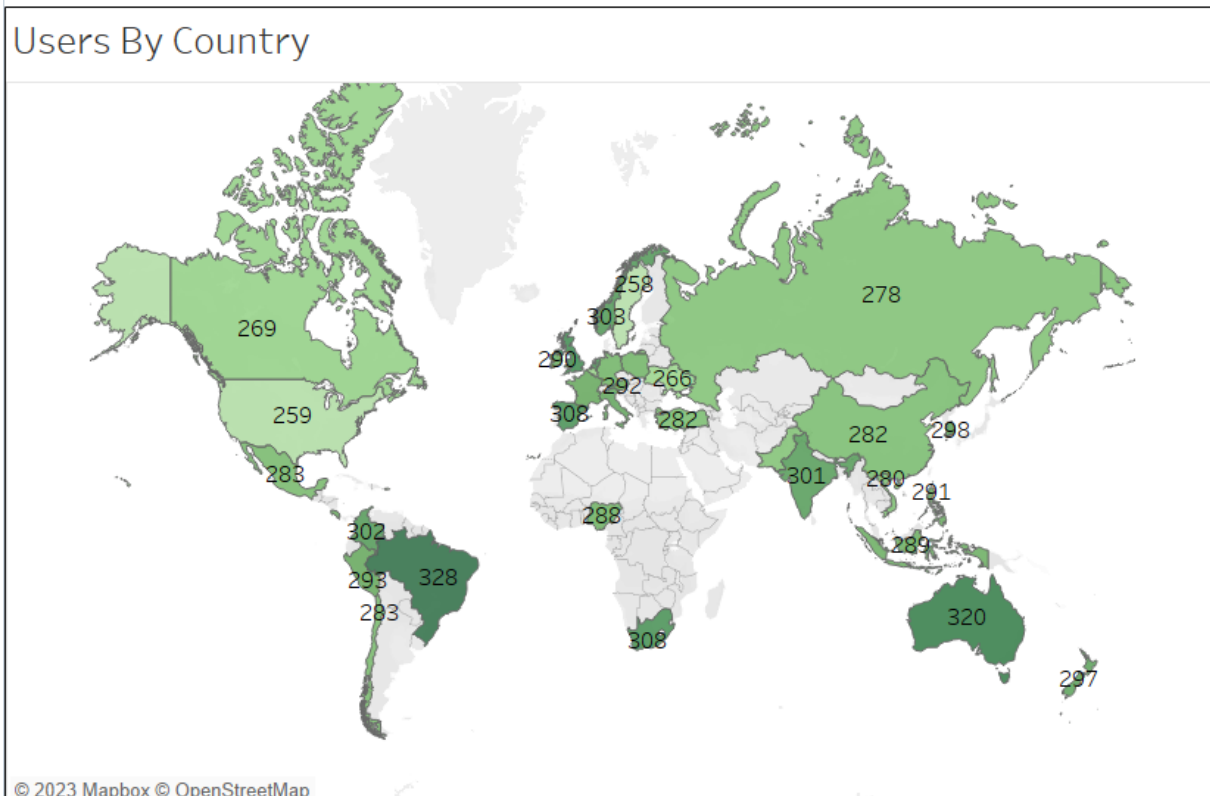
The database contains dates from the years 2019-2022 but the analysis will be based on data from the year **2020 only**, a yearly analysis you could say.

### 3.1 Descriptive Statics

I will start the analysis section with some simple analysis to give a little snapshot about the mobile app performance in 2020.

just to clarify I will add and say that the app was made by 1 person and the only expenses are spent on marketing.

Users Statics	Ratings Statics	Revenue Statics
Number Of Distinct Users That Installed The App: 32,424	Average Rating: 2.9 ★	Total Revenue: \$1,332
Number Of Users That Opened An Account: 3,420	Number Of Ratings: 368	Total Expenses: \$116
Number Of Diffrent Countries (Registered Users Only): 35		Number Of Purchases: 166
Average Age (Registered Users Only): 18		Avg Purchase Amount: \$8.03



## Mobile Game Analysis

### 3.2 User Activity

DAU, daily active users graph shows the number of **distinct** users the logged in in each day in 2020.

