Steam DB 🚀

Group members

Name	email	github_username
Qihang Dai	ahgdyycc@seas.upenn.edu	qihang-dai
JingXuan Bao	bjx@seas.upenn.edu	Jingxuan-Bao
Peihan Li	peihanli@seas.upenn.edu	Yourcody
JingRu Wang	wjingru@seas.upenn.edu	JrWang0930

Application Description

We are going to build a Steam Store application, which is a video game digital distribution platform. The database will include the information of games, users, and developers. We will also provide some basic functions for users to search for games and developers, and if time permits, we will also implement some advanced functions such as recommendation system and game review system.

We plan to hold the databse on AWS RDS and running the web application on AWS EC2. The web application is planned to be built with Flask and React. The database will be built with MySQL. And we wish the application would support as many as functions like that in the Steam Store.

Dataset Description

The dataset we are going to use is from Kaggle. It contains 27075 games and 18 features. The features include game name, developer, publisher, release date, price, tags, and so on. The dataset is updated on 2019-06-12 and it is a clean dataset, which saves us a lot of time on data cleaning (doesn't mean we don't need to do more EDA). The overall dataset is 252.04MB. Its a big enough dataset for us to build a database and a web application. For exapmle, We would use steam.csv as our main dataset, which has 27075 rows and 18 columns. We would also use steam_description_data.csv as our secondary dataset, which has 27075 rows and 2 columns. We would use the game name as the primary key to join the two datasets. The overall order of the

dataset shall meet the requiredment of meaningful optimization cause the datasets has 6 csv files in total that we may optionally use.

Five Sudo SQL Query (aggregate, subquery, join, nested aggregation, nested subquery, etc.)

1. Search for games with 3d tag publiced after 2020.
   "*SELECT NAME FROM steam S*
   JOIN steamspy_tag_data T ON S.appid = T.appid
   WHERE T.3d > 0 AND S.realse_date > 2020-00-00"
2. Search for the games with game short description which publiced after 2020.
   "*SELECT S.NAME, D.short_description FROM steam S*
   JOIN steam_description_data D ON S.appid = D.appid
   WHERE S.realse_date > 2020-00-00"
3. Search for the percentage of game in English.
   "*SELECT COUNT(*) FROM steam S*
   WHERE english = 1 /
   SELECT COUNT(*) FROM steam S
   WHERE english <> 1"
4. Search for games which are described as action game and have 3d view, and sort these games by their release date.
   "*SELECT t1.appid AS app_id, name FROM Steam t1*
   JOIN Steam_description t2 ON t1.appid = t2.appid
   JOIN Steamspy_tag t3 ON t1.appid = t3.appid
   WHERE short_description LIKE '%action%' AND 3d = 1
   GROUP BY t1.appid, name
   ORDER BY release_date DESC“
5. Search for the top10 games with highest average rating in each year (Using window functions)
   "*SELECT t1.appid AS app_id, name, release_year, rnk*
   FROM (SELECT RANK() OVER(partition by YEAR(release_year) ORDER BY average_rating DESC) rnk,
   t1.appid AS app_id, name
   FROM Steam t1 JOIN Steam_description t2 ON t1.appid = t2.appid) t
   WHERE rnk <= 10
   ORDER BY release_year, rnk"