GAME↑UP

GROUP 2
Chenyi Xu
Wei Lin
Sahira Zaheen
B.Royal Nebeker

GameUP MISSION STATEMENT:

Taking back the gaming industry by connecting gamers to publishers from every corner of the world using honest gamer-generated information to unlock all achievements.

PROBLEM SUMMARY:

· Massive marketing budgets drowning out small publishers

· Microtransactions required to be competitive

· Reviews are purchased and paid for

· The same franchise repeatedly

· Innovation is lacking

In 2013, according to Guinness World Records, the video game Grand Theft Auto V broke six world records. GTAV is the fastest selling entertainment product, ever, i.e. since-the-beginning-of-time. In 2018, Red Dead Redemption II took the title of second-highest grossing entertainment launch, ever. Both games were published by Rockstar Games. How can small video game publishers begin to compete against production/marketing budgets of $265M?

Some could say that Rockstar Games is simply an outlier. However, looking at the revenues of major game publishers it is easy to see that when it comes to publishing video games, one must "pay to play". Since 2010, Tencent, a Chinese-headquartered video game publisher, has seen revenue of $92.4B which is more than double that of major video game publisher Electronic Arts ($40B). Tencent's annual revenue is about $10.9B compared to Microsoft's annual gaming revenue of $7.4B.

With massive budgets these major publishers have stifled innovation by focusing only on their blockbuster hits creating franchises that have no end in sight. Reviews and once trusted video game sites are all paid to promote, purchased by the highest bidder. Offshoots of major hits can be found on mobile platforms riddled with microtransactions squeezing every penny out of gamers. GTAV online, 7 years later, sees regular updates and added in-game items that are available for purchase. Small innovative game publishers are more akin to an ant than a David facing Goliath.

GAMEUP is the voice that small publishers and gamers alike have been searching for. By providing real playing experiences that are not for purchase, gamers can search genres, read/review, and follow the links to new innovative games which will reignite their love for the industry. In addition, reviews include honest and upfront information on in-game costs, be it microtransactions to expansion packs. While GAMEUP has sponsorships, they are clearly noted and accepted strictly based on review scores, not on their marketing budgets.

Globalized gaming has brought many gamers into contact with foreign publishers but finding reviews in the gamer's language is a hassle. GAME**UP** believes gaming should transcend cultural barriers as all content is provided in the language preferred by the user. By giving both publishers and users a voice, GAME**UP** connects gamers worldwide to what they've always wanted.

GameUP SQL Queries:

1. **Games from different genre**
   This shows information to the user about the game's ID, name, platform, and genre name by the descending order of rating. Users can search for games from specific genres for a specific platform which have a star rating greater than 3.

SQL Code:
select G.game_id, G.name as game_name, D.platform, GN.name as genre, R.score as rating
from
Game as G,
Download as D,
Genre as GN,
Review as R
where G.game_id = D.game_id
and GN.genre_id = G.genre_id
and G.game_id = R.game_id
and D.platform = 'Android'
and GN.name = 'Sports' and R.score > 3
order by R.score DESC;

Output:

game_id	game_name	platform	genre	rating
71	FIFA 14	Android	Sports	5
71	FIFA 14	Android	Sports	5
78	NBA 2K13	Android	Sports	5
80	Madden NFL 09	Android	Sports	4
80	Madden NFL 09	Android	Sports	4

2. Games which have at least 3 languages

This shows the games that have at least 3 languages and sort by the language count.

SQL Code:

```
select G.game_id, G.name as game_name, count(L.Language_id) as language_count
from Game G
join Game_has_Language GL on G.game_id = GL.Game_id
join Language L on GL.Language_id = L.Language_id
group by G.game_id, game_name
having count(L.Language_id) >= 3
order by language_count desc;
```

Output:

game_id	game_name	language_count
59	Ninja Gaiden Sigma	5
73	Disgaea D2: A Brighter Darkness	4
87	Puzzle Craft	3
21	Batman: Arkham City	3
39	Okami HD	3
69	The Elder Scrolls IV: Oblivion	3
89	Deus Ex: The Fall	3
12	Minecraft	3
29	World of Goo	3
76	Mutant Mudds	3
32	Uncharted 2: Among Thieves	3
23	FIFA Soccer 12	3
18	Mass Effect 3	3
38	Ni no Kuni: Wrath of the White...	3
60	Prince of Persia	3
6	Red Dead Redemption -- Undea...	3
27	Portal 2	3
47	Rock Band (Special Edition)	3
54	Call of Duty: Black Ops II	3
22	NBA 2K12	3
80	Madden NFL 09	3
14	Rayman Legends	3
35	Resistance 2	3

3. Regions with total number of users

This creates a view that displays the total number of users by region. Employees can use this information to better plan their strategy around the regions.

SQL Code:

```
CREATE  VIEW `Overview_regionsWithTotalNumberOfUsers` AS
select R.region_id as region_id, R.name as region_name, count(user_id) as
total_number_of_user
from Region R,
User U
where R.Region_id = U.Region
group by R.Region_id, R.name
order by count(user_id) desc;
```

Output:

region_id	region_name	total_number_of_user
6	Africa	12
2	North America	10
1	Asia	9
4	Australia	9
3	Europe	5
5	South America	5

4. Recent sponsorship updates

This provides information about the most recent sponsorship type, sponsor name, and which game is sponsored by recent date. Marketing and sales department could use this information to analyze the expenses, revenue, etc.

SQL Code:

```
select S.type as sponsor_type, S.end_date as recenr_sponsor, P.name as sponsor_name, G.name
as sponsored_game
from Sponsorship S,
Publisher P,
Game G
where S.game_id = G.game_id
and G.publisher_id = P.publisher_id
group by S.type, P.name, S.end_date, G.name
having year(S.end_date) > 2019
order by S.end_date DESC
```

Output:

	sponsor_type	recent_sponsor	sponsor_name	sponsored_game
▶	financial	2020-02-24	Namco Bandai Games	Ni no Kuni: Wrath of the White Witch
	financial	2020-02-15	MTV Games	Rock Band (Special Edition)
	financial	2020-02-10	Take-Two Interactive	Grand Theft Auto V
	financial	2020-01-27	Namco Bandai Games	Pac-Man Championship Edition DX
	event	2020-01-11	Capcom	Bionic Commando Rearmed

5. **Users who downloaded most games**
   This displays user's id, name, and region who have downloaded most games in January 2020, providing the information about user activity.

SQL Code:

select D.User_id, U.name, R.name as region

from Download D

left join User U

on D.User_id = U.User_id

left join Region R

on U.region = R.region_id

where month(D.download_date) = 1 and year(D.download_date) = 2020

group by D.User_id

having count(D.game_id) =

      (select count(D.Game_id) as game_count

      from Download D

      where month(D.download_date) = 1 and year(D.download_date)=2020

      group by D.User_id

      order by game_count desc limit 1);

Output:

	User_id	name	region
▶	2	bwhymark1	Africa

6. **Numbers of download, number of recommendations, and the recommendation download transfer rate for specific time period**

    This will be used by employees to see how many recommendations have been turned into actual downloads by the users.

    Here two stored procedures were created to provide the number of downloads and number of recommendations in a given time period.

SQL Code:
 # giving the recommendation_download transfer rate for Jan 2020
CALL recommenation_download_count('2020-01-10',@recommendation_count,@download_count;
CALL recommenation_download_transfer_rate('2020-01-10');

Output:

@recommendation_count/@download_count
► 0.3000

7. **Check invoice for publishers**

    Check invoice for Sony in Jan 2020, will be used by the employees to see how much money has flowed into the system from publisher during said time period.

SQL Code:
select G.Publisher_id,P.name, sum(G.publisher_charged_money) from Download D
join Game G on D.game_id = G.Game_id
join Publisher P on G.Publisher_id = P.Publisher_id
where month(D.download_date) = month('2010-01-10') and year(D.download_date) = year('2020-01-10') and P.name like '%sony%'
group by G.Publisher_id;

Output:

Publisher_id	name	sum(G.publisher_charged_money)
► 2	Sony Computer Entertainment	284.08

8. **Give the information about department and group that operate the sponsorship about the game that was downloaded most in Jan 2020**

    This will be useful when tracking the sponsorship operations of the top downloaded game.

    Here a stored procedure was created to provide the game that has been downloaded most in given time period.

SQL Code:
 # give the game that has been downloaded most in Jan 2020
call max_download_game('2020-01-10',@max_download_game_id);
 # give the department and group who operates the game
select sp.sponsorship_id,sp.type,D.name as department,sh.group_id from Sponsorship sp
join Sponsorship_has_Group sh on sp.sponsorship_id = sh.Sponsorship_id
join mm_cpsc5910team02.Group G on G.group = sh.group_id and G.department_id = sh.department_id
join Department D on D.department_id = G.department_id
where sp.Game_id = @max_download_game_id;

Output:

sponsorship_id	type	departme...	group_id
29	financial	Marketing	1
29	financial	Legal	1
29	financial	Marketing	2

9. **Check the download count of max_download_game before sponsorship**
   Here a stored procedure was created to provide the recent sponsorship id of the max_download_game.

SQL Code:
#for the max_download_game, give the total download count within 60 days before the most recent sponsorship
CALL
recent_sponsor_for_max_download_game(@max_download_game_id,@sponsor_end_date);
select D.Game_id, count(*) from Download D
where datediff(@sponsor_end_date,D.download_date) < 60 and
datediff(@sponsor_end_date,D.download_date) >0 and D.game_id = @max_download_game_id;

Output:

Game_id	count(*)
4	1

10. For the max_download_game, give the total download count after the most recent sponsorship

It will show after sponsorship if the publisher's game download has been increased.

SQL Code:

select D.Game_id, count(*) from Download D
where datediff(D.download_date,@sponsor_end_date) >0 and D.game_id =
@max_download_game_id;

Output:

Game_id	count(*)
4	4

GameUP Stored Procedures:

1. Create procedure of retrieving recommendation and download count given specific time period

DELIMITER $$
CREATE PROCEDURE recommenation_download_count(IN time date, out recommendation_count int, out download_count int)
BEGIN
    select count(*) into recommendation_count from Recommendation R where month(R.review_date) = month(time) and year(R.ewview_date)= year(time);
  select count(*) into download_count from Download D where month(D.download_date) = month(time) and year(D.download_date)= year(time);
END$$
DELIMITER ;

2. Create procedure of retrieving recommendation and download transfer rate given specific time period

DELIMITER $$
CREATE PROCEDURE recommenation_download_transfer_rate(IN time date)
BEGIN
    call recommenation_download_count(time,@recommendation_count,@download_count);
  select @recommendation_count/@download_count;
END$$
DELIMITER ;

3. Create procedure retrieving the game id that has been download most in specific time period

DELIMITER $$
CREATE PROCEDURE max_download_game(IN time date, out max_download_game int)
BEGIN
    select D2.game_id into max_download_game from Download D2
  where month(D2.download_date) = month(time) and year(D2.download_date)= year(time)
  group by D2.game_id
  having count(D2.download_id) = (
   select count(D1.download_id) from Download D1
  where month(D1.download_date) = month(time) and year(D1.download_date)= year(time)

```
    group by D1.game_id
    order by count(D1.download_id) desc limit 1);
    END$$
    DELIMITER ;
```

4. Create procedure that retrieves the most recent sponsorship end date for the max_download_game

```
DELIMITER $$
create procedure recent_sponsor_for_max_download_game(in max_download_game_id
int,OUT sponsor_end_date date)
BEGIN
        select S.end_date into sponsor_end_date
    from Sponsorship S
    where S.game_id = @max_download_game_id
    order by S.end_date desc limit 1;
END$$
DELIMITER ;
```

GameUP Physical model diagram:
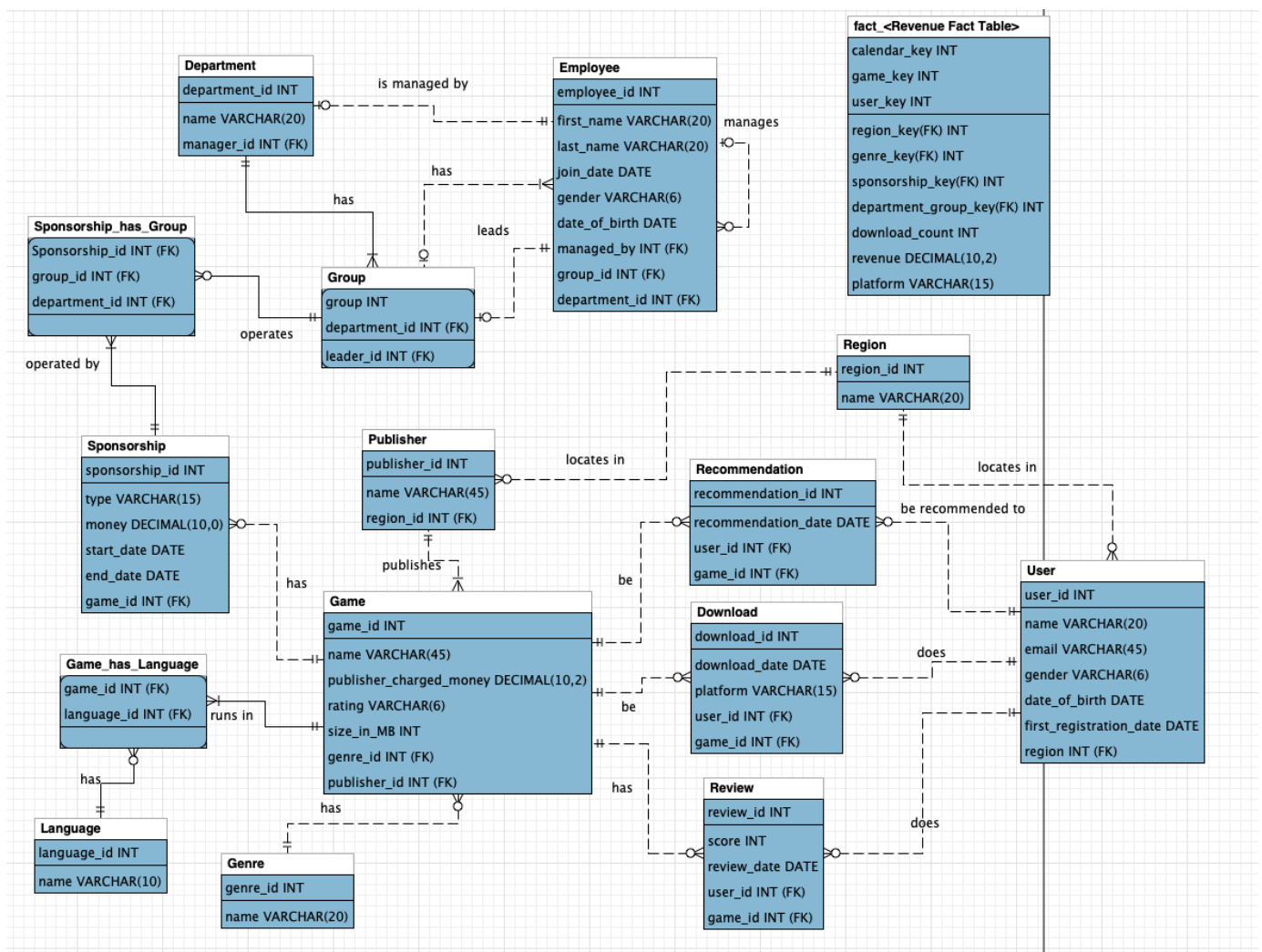
This Revenue Fact Table records the download count, revenue(sum of publisher_charged_money) and platform for each game, user and day. It has three primary keys and four foreign keys referencing seven dimension tables, Calendar Dimension, Game Dimension, User Dimension, Region Dimension, Genre Dimension, Sponsorship Dimension and Department_group Dimension.

For the calendar dimension table, extract the download_date from the download table and then transfer it to fill in the information of date, like day of week, day of month, month, quarter and year.

For the game dimension table, extract the information of game name, rating and size from the game table and join with the publisher table to get game and publisher information.

For the user dimension table, extract the information of user name, email, birth date and first_registration_date in the user table to get user information.

For the region dimension table, extract the information of region id and name in region table.

For the genre dimension table, extract the information of genre id and name in the region table.

For the sponsorship dimension table, extract the information of sponsorship type and money and its start date and end date from the sponsorship table.

For department_group dimension table, join group table and department table and employee table to get the department name, group name and employee information.

Access MYSQL database in the cloud:

Hostname: cpsc5910team02.c5hkxwwgu62q.us-east-1.rds.amazonaws.com
Username: admin
Password: adminadmin

Access tableau public site online site:

https://public.tableau.com/profile/szaheen#!/vizhome/GameUP1_Group02_CPSC_Milestone03/userregion?publish=yes

Upload in the Github

We have uploaded the project on github.