



ИНСТИТУТ
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС



УНИВЕРСИТЕТ
ИННОПОЛИС

Игровая перспектива 2025:

Исследование ключевых факторов разработки новой компьютерной игры

Итоговый проект в рамках аттестации по курсу:

Программная инженерия. Разработка программного обеспечения
(SQL для начинающих специалистов)

✉ Исполнитель: Боталов Константин Владимирович

📅 2024



Цель:

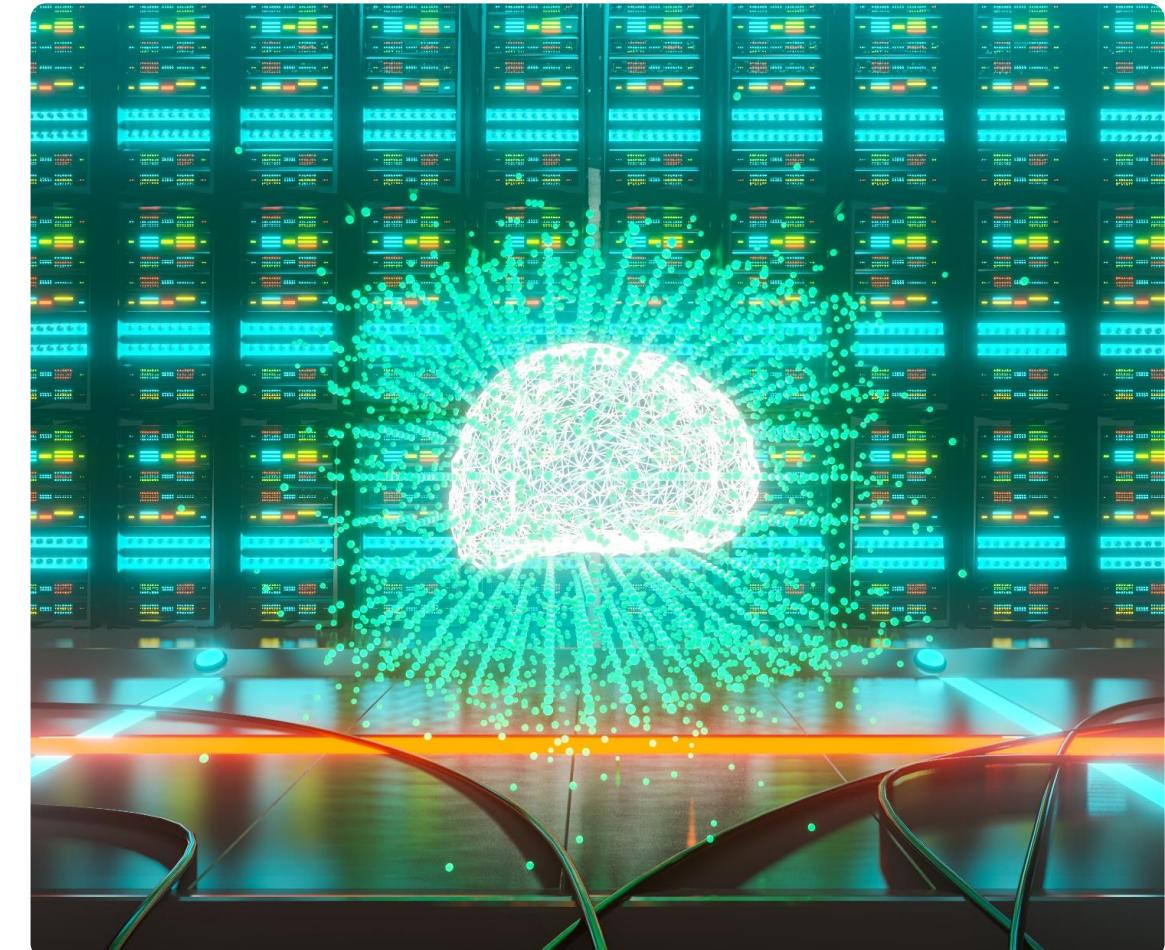
- Определить наиболее перспективные жанры и платформы для разработки компьютерных игр в 2025 году
- Рекомендовать подходящих издателей и выявить успешные игровые серии
- Использовать SQL и методы программирования

Задачи:

1. Первичная обработка данных
2. Первичное исследование данных
3. Импорт данных в базу данных
4. Подготовка и первичный анализ данных средствами PostgreSQL
5. Проектирование и создание базы данных
6. Анализ ключевых факторов, влияющих на успех разработки компьютерной игры
7. Интерпретация полученных результатов

Основные инструменты проекта

- **ОС:** Windows 11 Home 23H2
- **Среда контейнеризации приложений:**
Docker Desktop
- **СУБД:** PostgreSQL / pgAdmin
- **Языки:** Python, SQL, PL/pgSQL
- **Система управления версиями:** GIT
- **Интерактивное веб-приложение:**
Jupyter Server / Jupiter Notebook
- **Библиотеки:** Pandas, SQLAlchemy, psycopg



Загрузка данных



```
[1]: # Импорт необходимых библиотек
import os.path # os - для работы с файлами операционной системы
import pandas as pd # pandas - для работы с табличными данными

[2]: # Путь к исходному CSV-файлу
file_path = '../src/vgchartz-2024.csv'

# Проверка существования файла по заданному пути
os.path.exists(file_path)

[2]: True

[3]: # Загрузка CSV файла в DataFrame Pandas
try:
    df = pd.read_csv(file_path)
    display('Файл загружен!')
except FileNotFoundError:
    display('Файл отсутствует!')

'Файл загружен!'
```



Первичное исследование данных



```
[6]: # Получение размера набора данных (количество строк и столбцов)
display(df.shape)
```

```
# Получение списка названий столбцов
display(df.columns)
```

```
(64016, 14)
Index(['img', 'title', 'console', 'genre', 'publisher', 'developer',
       'critic_score', 'total_sales', 'na_sales', 'jp_sales', 'pal_sales',
       'other_sales', 'release_date', 'last_update'],
      dtype='object')
```

```
[10]: # Определение минимумов и максимумов для числовых значений
df.describe()
```

	critic_score	total_sales	na_sales	jp_sales	pal_sales	other_sales
count	6678.000000	18922.000000	12637.000000	6726.000000	12824.000000	15128.000000
mean	7.220440	0.349113	0.264740	0.102281	0.149472	0.043041
std	1.457066	0.807462	0.494787	0.168811	0.392653	0.126643
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	6.400000	0.030000	0.050000	0.020000	0.010000	0.000000
50%	7.500000	0.120000	0.120000	0.040000	0.040000	0.010000
75%	8.300000	0.340000	0.280000	0.120000	0.140000	0.030000
max	10.000000	20.320000	9.760000	2.130000	9.850000	3.120000

```
[9]: # Подсчет количества null-значений в наборе данных
df.isnull().sum()
```

```
img          0
title        0
console      0
genre         0
publisher     0
developer    17
critic_score 57338
total_sales  45094
na_sales     51379
jp_sales     57290
pal_sales    51192
other_sales  48888
release_date 7051
last_update   46137
dtype: int64
```



Импорт данных в базу данных

```
[11]: # Импорт необходимых библиотек для переноса данных
from sqlalchemy import create_engine # Набор инструментов для работы с SQL из Python
import psycopg2 # Коннектор к СУБД PostgreSQL из Python
```

```
[12]: # Импорт дата-фрейма в PostgreSQL средствами Pandas и SQLAlchemy

# Стока подключения к базе данных SALES
# Пример: 'postgresql://username:password@localhost:5432/your_database'
db_connection = create_engine('postgresql://postgres:changeme@localhost:5432/sales')

# Перенос данных из дата-фрейма в таблицу базы данных
df.to_sql(
    name='sales_import', # Имя целевой таблицы
    con=db_connection, # Подключение к базе данных
    if_exists='replace', # если таблица уже существует, перезапись
    index=False # без индекса, он нам пока не требуется
)
```

[12]: 16

```
/* Создание чистой базы данных SALES */
CREATE DATABASE sales;

/* Проверяем, что создалась база данных SALES */
SELECT datname FROM pg_database
WHERE datname='sales';

/* Проверка количества импортированных данных в таблицу SALES_IMPORT */
SELECT COUNT(*) AS IMPORT_COUNT
FROM sales_import;
```

	import_count	bigint
1	64016	



Подготовка и первичный анализ данных средствами PostgreSQL (1)



```
/* Вывод таблицы с количеством уникальных значений по каждому столбцу */
DO $$

DECLARE
cols TEXT; -- Переменная для хранения агрегированного списка
BEGIN
/*
Очищаем (удаляем) временную таблицу counter перед ее заполнением
*/
DROP TABLE IF EXISTS temp_counter;
cols := string_agg( -- объединяем строки в одну строку с разделителем (запятая)
    'COUNT(DISTINCT ' || column_name::TEXT || ')' || column_name::TEXT, ','
)
FROM (
/* Подзапрос для получения всех имен колонок из таблицы SALES_IMPORT */
SELECT column_name
    FROM information_schema.columns
    WHERE
        table_name = 'sales_import'
) AS c;
/*
Создаем временную таблицу,
где в качестве столбцов будут строки
из переменной COLS (названия столбцов)
*/
EXECUTE format(
    'CREATE TEMP TABLE temp_counter AS SELECT %s FROM sales_import;',
    cols
);
END $$;
SELECT * FROM temp_counter; -- Вывод данных из временной таблицы
```

	na_sales	jp_sales	pal_sales	other_sales	critic_score	total_sales	release_date	img	last_update	title	console	genre	published	developer
	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint
1	320	121	256	133	89	482	7922	56177	1545	39798	81	20	3383	8862



PostgreSQL

Подготовка и первичный анализ данных средствами PostgreSQL (2)



```
/* Подсчет количества NULL-значений по каждому столбцу */
DO $$$
DECLARE
cols TEXT; -- Переменная для хранения агрегированного списка
BEGIN
-- Удаляем временную таблицу, если она существует
DROP TABLE IF EXISTS temp_counter;
cols := string_agg( -- Объединяем строки в одну строку с разделителем (запятая)
    /* Получаем разницу между общим количеством записей и ненулевыми значениями */
    'COUNT(*) - COUNT(' || column_name::TEXT || ') AS ' || column_name::TEXT, ','
)
FROM (
    /* Подзапрос для получения всех имен колонок из таблицы sales_import */
    SELECT column_name
    FROM information_schema.columns
    WHERE
        table_name = 'sales_import'
    ) AS c;
-- Создаем временную таблицу
EXECUTE format(
    'CREATE TEMP TABLE temp_counter AS SELECT %s FROM sales_import;',
    cols
);
END $$;
SELECT * FROM temp_counter; -- Вывод данных из временной таблицы
```

	na_sales	jp_sales	pal_sales	other_sales	critic_score	total_sales	release_date	img	last_update	title	console	genre	published	developer
1	51379	57290	51192	48888	57338	45094	7051	0	46137	0	0	0	0	17

```
/* Очистка null-значений */
UPDATE sales_import
SET
    total_sales = COALESCE(total_sales, 0),
    na_sales = COALESCE(na_sales, 0),
    jp_sales = COALESCE(jp_sales, 0),
    pal_sales = COALESCE(pal_sales, 0),
    other_sales = COALESCE(other_sales, 0),
    critic_score = COALESCE(critic_score, 0),
    release_date = COALESCE(release_date, '1900-01-01'),
    last_update = COALESCE(last_update, '1900-01-01'),
    developer = COALESCE(developer, '--UNKNOWN--');
```

	na_sales	jp_sales	pal_sales	other_sales	critic_score	total_sales	release_date
1	0	0	0	0	0	0	0

	img	last_update	title	console	genre	published	developer
	0	0	0	0	0	0	0



PostgreSQL

Нормализация базы данных



```
...
/* Основные сущности базы данных */
- 'games' (Игры)
- 'consoles' (Консоли)
- 'genres' (Жанры)
- 'publishers' (Издатели)
- 'developers' (Разработчики)
...
...
```

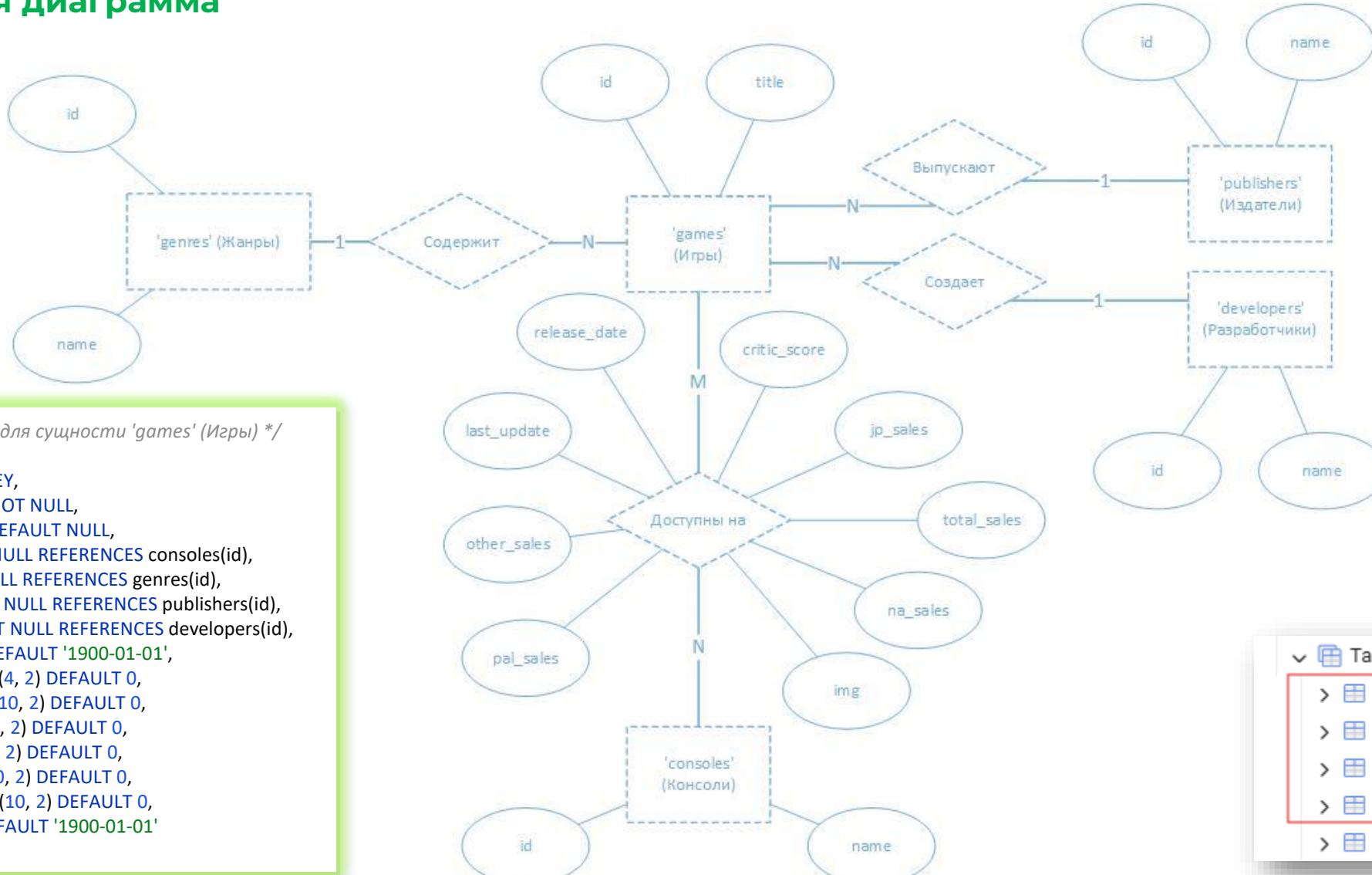
```
...
/* Основные связи базы данных */
* Игры доступны на нескольких Консолях (M:N)
* Жанр содержит множество Игр (1:N)
* Издатель выпускает Игры (1:N)
* Разработчик создает Игры (1:N)
...
...
```

```
/*
Определение длины полей для текстовых значений:
Максимальная длина поля * 1.5
*/
SELECT
    ROUND(MAX(LENGTH(title)) * 1.5) AS max_title_length,
    ROUND(MAX(LENGTH(img)) * 1.5) AS max_img_length,
    ROUND(MAX(LENGTH(console)) * 1.5) AS max_console_length,
    ROUND(MAX(LENGTH(genre)) * 1.5) AS max_genre_length,
    ROUND(MAX(LENGTH(publisher)) * 1.5) AS max_publisher_length,
    ROUND(MAX(LENGTH(developer)) * 1.5) AS max_developer_length
FROM sales_import;
```

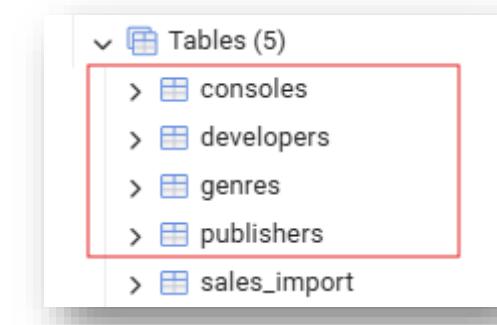
	max_title_length	max_img_length	max_console_length	max_genre_length	max_publisher_length	max_developer_length
	numeric	numeric	numeric	numeric	numeric	numeric
1	221	213	9	24	74	102



Логическая диаграмма



```
/* Создание таблицы для сущности 'games' (Игры) */
CREATE TABLE games (
    id SERIAL PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    img VARCHAR(255) DEFAULT NULL,
    console_id INT NOT NULL REFERENCES consoles(id),
    genre_id INT NOT NULL REFERENCES genres(id),
    publisher_id INT NOT NULL REFERENCES publishers(id),
    developer_id INT NOT NULL REFERENCES developers(id),
    release_date DATE DEFAULT '1900-01-01',
    critic_score DECIMAL(4, 2) DEFAULT 0,
    total_sales DECIMAL(10, 2) DEFAULT 0,
    na_sales DECIMAL(10, 2) DEFAULT 0,
    jp_sales DECIMAL(10, 2) DEFAULT 0,
    pal_sales DECIMAL(10, 2) DEFAULT 0,
    other_sales DECIMAL(10, 2) DEFAULT 0,
    last_update DATE DEFAULT '1900-01-01'
);
```



Анализ ключевых факторов, влияющих на успех разработки компьютерной игры (1)



```
/*
Удаление малозначимых игр,
у которых total_sales < 0.5 и critic_score <= 0
*/
DELETE FROM games
WHERE
total_sales < 0.5 OR
critic_score <= 0;
```



```
/*
Вычисление качества набора данных
*/
SELECT
ROUND((COUNT(title)::DECIMAL / 64016) * 100, 2) || '%' AS
quality_db
FROM
games;
```

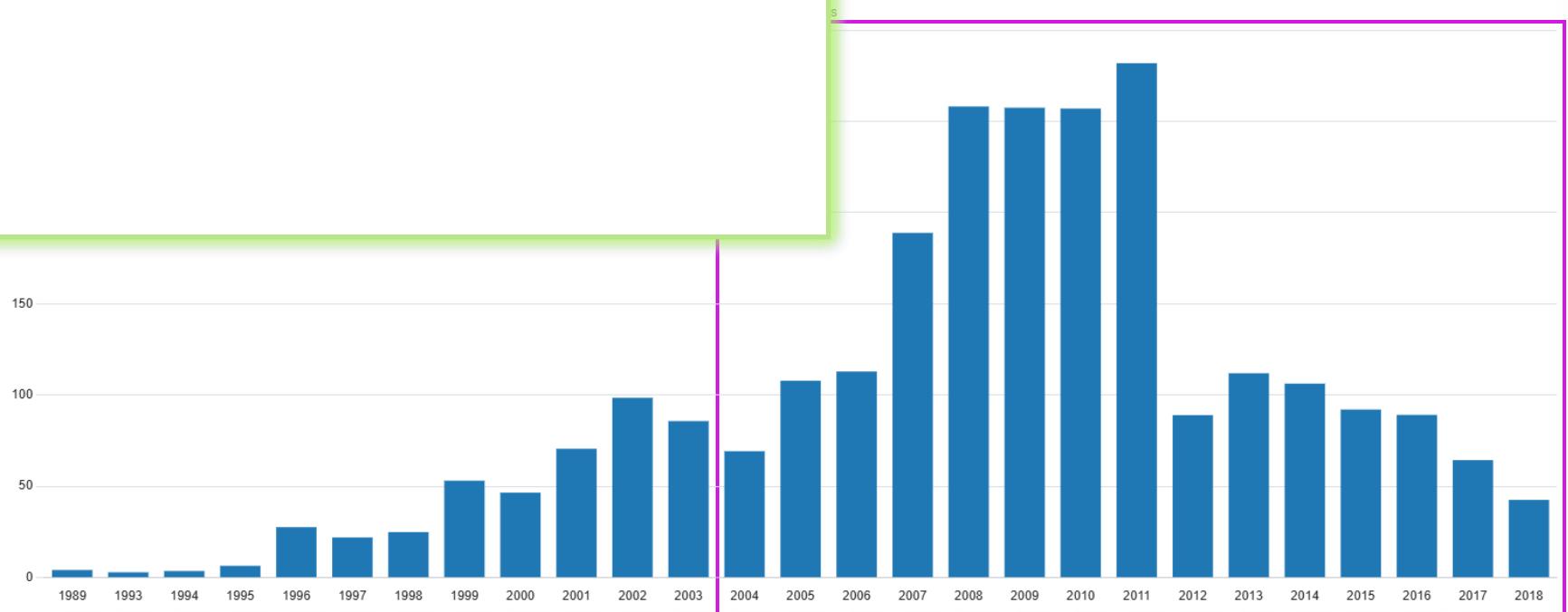
	quality_db	🔒
1	2.37%	

	id [PK] integer	title character varying (255)		total_sales numeric (10,2)	critic_score numeric (4,2)
1	3409	Hometown Story		0.50	5.50
2	3397	Avatar: The Last Airbender		0.50	5.50
3	3360	Steel Diver		0.50	5.60
4	3422	Super Monkey Ball 3D		0.50	5.70
5	3423	Spider-Man 2		0.50	6.10
6	3400	Ratatouille		0.50	6.50
7	3403	Epic Mickey: Power of Illusion		0.50	6.60
8	3355	Wet		0.50	6.90
9	3399	Alice: Madness Returns		0.50	6.90
10	3353	Goldeneye 007: Reloaded		0.50	7.40
11	3406	Pro Evolution Soccer 2012		0.50	7.70
12	3371	Pokémon Art Academy		0.50	7.80
13	3413	Transformers: War for Cybertron		0.50	7.80
14	3415	WWE SmackDown vs. RAW 2007		0.50	8.00
15	3420	R.U.S.E.		0.50	8.00
16	3362	Rock Band Unplugged		0.50	8.00

Total rows: 1000 of 1516 Query complete 00:00:00.358 Ln 13, C

Анализ ключевых факторов, влияющих на успех разработки компьютерной игры (2)

```
/*
  Распределение глобальных продаж игр по времени
*/
SELECT
  EXTRACT(YEAR FROM release_date) AS year,
  SUM(total_sales) AS total_sales
FROM
  games
GROUP BY
  year
ORDER BY
  year ASC;
```



Анализ ключевых факторов, влияющих на успех разработки компьютерной игры (3)



```
/*
    Создание представления (view) "ТОП-100 игр" по количеству продаж
*/
DROP VIEW IF EXISTS top_100_games_by_sales; -- Удаление представления (если существует)

CREATE VIEW top_100_games_by_sales AS -- Создание представления
SELECT
    g.title,
    c.name AS console,
    j.name AS genre,
    p.name AS publisher,
    d.name AS developer,
    SUM(g.total_sales) AS total_sales,
    EXTRACT(YEAR FROM g.release_date) AS release_year
FROM games g
JOIN consoles c ON g.console_id = c.id
JOIN genres j ON g.genre_id = j.id
JOIN publishers p ON g.publisher_id = p.id
JOIN developers d ON g.developer_id = d.id
WHERE g.release_date >= NOW() - INTERVAL '20 years'
GROUP BY g.title, c.name, j.name, p.name, d.name, g.release_date
ORDER BY total_sales DESC
LIMIT 100;

SELECT * FROM top_100_games_by_sales; -- Отображение представления
```

	title character varying (255)	console character	genre character varying (2)	critic_score numeric (4,2)	release_date date	total_sales numeric	
1	Grand Theft Auto V	PS3	Action	9.40	2013-09-17	20.32	
2	Grand Theft Auto V	PS4	Action	9.70	2014-11-18	19.39	
3	Call of Duty: Black Ops 3	PS4	Shooter	8.10	2015-11-06	15.09	
4	Call of Duty: Modern Warfare 3	X360	Shooter	8.70	2011-11-08	14.82	
5	Call of Duty: Black Ops	X360	Shooter	8.80	2010-11-09	14.74	
6	Red Dead Redemption 2	PS4	Action-Adventure	9.80	2018-10-26	13.94	
7	Call of Duty: Black Ops II	X360	Shooter	8.40	2012-11-13	13.86	
8	Call of Duty: Black Ops II	PS3	Shooter	8.00	2012-11-13	13.80	
9	Call of Duty: Warzone	title character varying (255)	console character	genre character va	critic_score numeric (4,2)	release_date date	total_sales numeric
10	Call of Duty: Warzone	X360	Action	10.00	2008-04-29	11.09	
11	Call of Duty: Warzone	PS3	Action	10.00	2008-04-29	10.57	
12	Call of Duty: Warzone	PS4	Action-A...	9.80	2018-10-26	13.94	
13	FIFA 18	X360	Action	9.70	2014-11-18	19.39	
Total rows: 100							
Query complete 00:00:00.238							
Ln 1, Col 1							

Анализ ключевых факторов, влияющих на успех разработки компьютерной игры (4)



```
/*
  Создание материализованного представления по 2 представлениям
*/
-- Удаляем мат. представление, если существует
DROP MATERIALIZED VIEW IF EXISTS top_100_games_join;

-- Создаём мат. представление
CREATE MATERIALIZED VIEW top_100_games_join AS
SELECT
    v1.title,
    v1.console,
    v1.genre,
    v1.publisher,
    v1.developer
FROM
    top_100_games_by_sales v1 -- Представление по количеству продаж
INTERSECT
SELECT
    v2.title,
    v2.console,
    v2.genre,
    v2.publisher,
    v2.developer
FROM
    top_100_games_by_critics v2; -- Представление по оценкам критиков

-- Вывод мат. представления
SELECT * FROM top_100_games_join;
```

	title character varying (255)	console character varying (255)	genre character varying (255)	publisher character varying (255)	developer character varying (255)
1	Call of Duty 4: Modern Warfare	PS3	Shooter	Activision	Infinity Ward
2	The Elder Scrolls V: Skyrim	PC	Role-Playing	Bethesda Softworks	Bethesda Game Studios
3	Call of Duty: Modern Warfare 2	X360	Shooter	Activision	Infinity Ward
4	Grand Theft Auto V	PS4	Action	Rockstar Games	Rockstar North
5	Guitar Hero II	PS2	Misc	RedOctane	Harmonix Music Systems
6	Halo: Reach	X360	Shooter	Microsoft Game Studios	Bungie
7	Batman: Arkham City	X360	Action	Warner Bros. Interactive	Rocksteady Studios
8	Grand Theft Auto V	PS3	Action	Rockstar Games	Rockstar North
9	Call of Duty: Advanced Warfare	X360	Shooter	Activision	High Moon Studios
10	FIFA Soccer 12	PS3	Sports	EA Sports	EA Canada
11	Red Dead Redemption	X360	Action	Rockstar Games	Rockstar San Diego
12	Uncharted 2: Among Thieves	PS3	Action	Sony Computer Entertainment	Naughty Dog
13	Minecraft	PS4	Misc	Sony Computer Entertainment	Mojang

Total rows: 33 of 33 Query complete 00:00:00.295 Ln 25, Col 34



Анализ ключевых факторов, влияющих на успех разработки компьютерной игры (5)



/* Поиск популярной консоли */

```
SELECT
    console,
    COUNT(console) AS console_count
FROM
    top_100_games_join
GROUP BY
    console
ORDER BY console_count DESC;
```

	console character varying (255)	console_count bigint
1	X360	14
2	PS3	11
3	PS4	4
4	PS2	2
5	XOne	1
6	PC	1

/* Поиск самого успешного издателя */

```
SELECT
    publisher,
    COUNT(publisher) AS publisher_count
FROM
    top_100_games_join
GROUP BY
    publisher
ORDER BY publisher_count DESC;
```

	publisher character varying (255)	publisher_count bigint
1	Rockstar Games	7
2	Activision	6
3	EA Sports	3
4	Microsoft Game Studios	3
5	Warner Bros. Interactive	2
6	Electronic Arts	2
7	Sony Computer Entertainment	2
8	Bethesda Softworks	2

/*
Поиск популярного издателя по коэффициенту:
количество продаж / игра

```
/*
Поиск популярного издателя по коэффициенту:
количество продаж / игра
*/
SELECT
    p.name AS publisher,
    SUM(g.total_sales) AS total_sales,
    COUNT(g.title) AS games_count,
    ROUND(AVG(g.critic_score), 2) AS critic_score_avg, -- средняя оценка
    ROUND(SUM(g.total_sales) / COUNT(g.title), 2) AS rate -- коэффициент
FROM games g
JOIN publishers p ON g.publisher_id = p.id
WHERE
    release_date >= NOW() - INTERVAL '20 years' -- интервал: 20 лет
GROUP BY
    p.name
HAVING
    COUNT(g.title) >= 20 -- не менее 20 игр
ORDER BY
    rate DESC -- сортируем по коэффициенту
LIMIT 10; -- ТОП-10 издателей
```

	publisher character varying (255)	total_sales numeric	games_count bigint	critic_score_avg numeric	rate numeric
1	Rockstar Games	146.25	31	8.82	4.72
2	Activision	333.66	119	7.53	2.80
3	Bethesda Softworks	72.07	26	7.97	2.77
4	Microsoft Game Studios	43.60	21	8.10	2.08

Анализ ключевых факторов, влияющих на успех разработки компьютерной игры (6)



```
/* Поиск популярного жанра игры: ТОП-10 */

SELECT
    j.name AS genre,
    SUM(g.total_sales) AS total_sales
FROM games g
JOIN genres j ON g.genre_id = j.id
WHERE
    release_date >= NOW() - INTERVAL '20 years'
GROUP BY
    genre
ORDER BY
    total_sales DESC
LIMIT 10;
```

	genre character varying (255)	total_sales numeric
1	Shooter	504.18
2	Action	372.22
3	Sports	358.26
4	Role-Playing	159.03
5	Racing	147.58
6	Misc	128.60
7	Adventure	74.57
8	Fighting	74.20
9	Action-Adventure	69.48
10	Simulation	63.24

Анализ ключевых факторов, влияющих на успех разработки компьютерной игры (7)



```
/* Поиск часто встречающихся слов в играх */

WITH word_list AS (SELECT unnest(string_to_array(lower(title), ' ')) AS word
    FROM games) -- строим массив слов из всех названий игр
SELECT word,
    COUNT(*) AS count
FROM word_list
WHERE word <> '' -- Исключаем пустые слова
    AND LENGTH(word) > 3 -- длина слова: более 3 символов
GROUP BY word
ORDER BY count DESC
LIMIT 10; -- Ограничиваем выборку (10)
```

	word text	count bigint
1	lego	49
2	call	47
3	soccer	43
4	madden	43
5	fifa	42
6	star	42
7	need	38
8	duty:	34
9	game	31
10	wars	30

```
/* Выбор рынка игры */
```

```
SELECT
    SUM(na_sales) AS "North America",
    SUM(jp_sales) AS Japan,
    SUM(pal_sales) AS Europe,
    SUM(other_sales) AS Other
FROM games;
```

	North America numeric	japan numeric	europe numeric	other numeric
1	1290.51	106.48	882.11	295.80



Анализ ключевых факторов, влияющих на успех разработки компьютерной игры (8)



```
/*
Определение наиболее перспективной серии игр:
извлекаем первые слова из названия и считаем их встречаемость в названии
*/
SELECT
    SUBSTRING(title FROM '^[^ ]+') AS first_word, -- Извлекаем первое слово из названия
    COUNT(*) AS game_count, -- Считаем количество игр
    ARRAY_AGG(title) AS titles -- Собираем названия игр в массив
FROM
    top_100_games_join
GROUP BY
    first_word -- Группируем по первому слову
HAVING
    COUNT(*) > 1 -- Отбираем группы, где более одной строки
ORDER BY
    game_count DESC, first_word; -- Сортируем по встречаемости игр и первому слову
```

	first_word text	game_count bigint	titles character varying[]
1	Call	6	{"Call of Duty 4: Modern Warfare","C...
2	Grand	4	{"Grand Theft Auto V","Grand Theft ...
3	Forza	3	{"Forza Motorsport 3","Forza Motor...
4	Red	3	{"Red Dead Redemption","Red Dead ...
5	The	3	{"The Elder Scrolls V: Skyrim","The E...
6	Assassin's	2	{"Assassin's Creed II","Assassin's Cr...
7	Batman:	2	{"Batman: Arkham City","Batman: Ar...
8	Battlefield	2	{"Battlefield 1","Battlefield 1"}
9	FIFA	2	{"FIFA Soccer 12","FIFA Soccer 12"}

Создание триггера для отслеживания изменения данных

```
/* Создание таблицы для отслеживания изменений в таблице GAMES */

CREATE TABLE change_log_tbl_games (
    id SERIAL PRIMARY KEY,
    game_id INT REFERENCES games(id) ON DELETE CASCADE,
    old_title VARCHAR(255),
    new_title VARCHAR(255),
    <...>
    old_last_update DATE,
    new_last_update DATE,
    change_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
/* Создание триггера */

-- Удаляем триггер (если он существует)
DROP TRIGGER IF EXISTS trigger_changes_log_tbl_games ON games;

-- Создаем триггер
CREATE TRIGGER trigger_changes_log_tbl_games
AFTER UPDATE ON games
FOR EACH ROW
EXECUTE PROCEDURE log_changes_tbl_games();
```

```
/* Функция триггера: добавление записи в таблицу CHANGE_LOG_TBL_GAMES */

CREATE OR REPLACE FUNCTION log_changes_tbl_games()
RETURNS TRIGGER AS $$

BEGIN
    -- Проверяем, изменились ли какие-либо из полей:
    IF (NEW.title IS DISTINCT FROM OLD.title) THEN
        INSERT INTO change_log_tbl_games (game_id, old_title, new_title)
        VALUES (OLD.id, OLD.title, NEW.title);
    END IF;

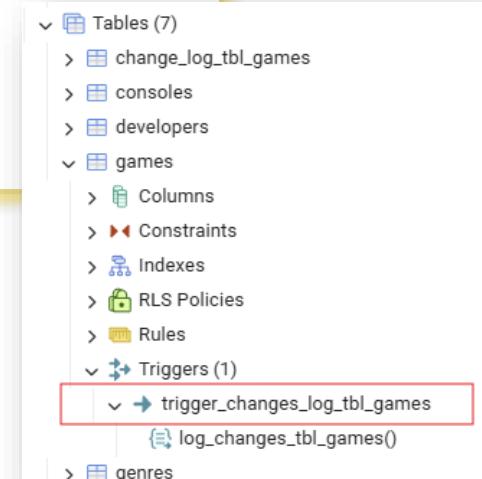
    <...>

    RETURN NEW; -- Возвращаем новые значения
END;
$$ LANGUAGE plpgsql;
```

```
/* Проверяем таблицу изменений */

SELECT *
FROM change_log_tbl_games
WHERE game_id IN (
    SELECT id
    FROM games
    WHERE title IN ('Updated: This is Test Game 1', 'Updated: This is Test Game 2')
)
ORDER BY old_title;
```

	id [PK] integer	game_id integer	old_title character varying (255)	new_title character varying (255)	old_console_id integer
1	1	64017	This is Test Game 1	Updated: This is Test Game 1	[null]
2	14	64018	This is Test Game 2	Updated: This is Test Game 2	[null]
3	11	64017	[null]	[null]	[null]



Основные выводы и рекомендации:

20

Исследование проводилось в рамках
двадцатилетнего временного интервала.



Наиболее привлекательные жанры:
Shooter, Action или **Sport**.



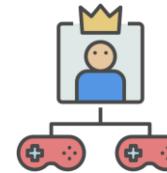
Наиболее перспективные серии игр:
Call of Duty и **Grand Theft Auto**.



Сотрудничать с признанными и
успешными издателями:
Rockstar Games и **Activision**.



Наиболее привлекательные регионы:
North America и **PAL**.



Вести разработку для консольных
платформ: **X360, PS3, PS4**
и их современных аналогов.



Избегать слов-паразитов в названиях игр:
lego, call, soccer, madden, fifa, star, need, duty, game, wars.



ИНСТИТУТ
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС



Спасибо за внимание!

Контакты:



Боталов Константин Владимирович

✉ konstantin.botalov@rt.ru

👉 @konstantin_botalov



Итоговый
проект



SQL-файл