

Міністерство освіти і науки України
**Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»**
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з комп'ютерного практикуму № 1 з дисципліни
«Аналіз даних в інформаційних системах»
„ Створення процедур завантаження даних”

Виконала

IT-01 Дмитрієва Ірина
(шифр, прізвище, ім'я, по батькові)

Перевірив

Олійник Ю.О.
(прізвище, ім'я, по батькові)

Київ 2022

Тема комп'ютерного практикуму: створення процедур завантаження даних.

Мета комп'ютерного практикуму: ознайомлення зі сховищем даних і вимірами, ETL-засобами, схемами «зірка» та «сніжинка», здобуття практичних навичок створення багатовимірних моделей.

Завдання:

- 1) Самостійно обрати не менше 3-х джерел відкритих даних.
- 2) Спроектувати модель Stage зони для ETL процесів.
- 3) Спроектувати модель основного сховища за типом за типом «зірка» або «сніжинка».
- 4) Створити ETL засоби:
 - завантажити дані до Stage зони
 - створити набір процедур/функцій для перетворення та завантаження даних до основного сховища. (Або створити засобами програмних ETL засобів). Передбачити можливість завантаження змінених та додаткових даних.
- 5) Завантажити дані до основного сховища даних.

Відомості про виконавця

Тема дослідження даних у даному комп'ютерному практикумі – продажі комп'ютерних ігор, їхня оцінка популярності по різних регіонах, а також визначення рейтингу ігор за стандартами ERSB.

[1. 1-vgsales.csv](#)

Цей набір даних містить список відеоігор із оцінкою продажів, критиків та користувачів. Станом на 12 квітня 2019 року в наборі даних є 55 792 записи. Він був згенерований за допомогою [vgchartz.com](#).

Це продовження роботи над цим вихідним набором даних:

<https://www.kaggle.com/gregorut/videogamesales>

Деякі зі включених полів включають:

- Rank – рейтинг загальних продажів
- Name – назва гри
- basename – коротка назва
- Platform – платформа гри (наприклад, ПК, PS4, XOne тощо)
- Genre - жанр гри
- ESRB_Rating - рейтинг гри ESRB
- Publisher – видавець гри
- Developer – розробник гри
- Critic_Score – оцінка критиків у грі від 10
- User_Score – користувачі оцінюють гру з 10
- Total_Shipped – загальна кількість відвантажених копій гри
- Global_Sales – загальний обсяг продажів у всьому світі (у мільйонах)
- NA_Sales – продажі в Північній Америці (у мільйонах)
- PAL_Sales - продажі в Європі (у мільйонах)
- JP_Sales – продажі в Японії (у мільйонах)
- Other_Sales - Продажі в решті світу (у мільйонах)
- Year - Рік випуску гри

2. [2-Video_games_esrb_rating.csv](#)

Ці дані містять 1895 ігор із 34 видами оцінювання за рейтингом ESRB з назвою та можливістю консолі для кожної гри.

- дані представлені у вигляді двійкового значення 0-1 для консолі та двійкового вектора для ознак рейтингу ESRB.
- RP, EC, A, рейтинг не надається в поточній версії даних, він може бути включений в наступні оновлення.

Перелік полів датасету:

- title
- console
- alcohol_reference
- animated_blood
- blood
- blood_and_gore
- cartoon_violence
- crude_humor
- drug_reference
- fantasy_violence
- intense_violence
- language
- lyrics
- mature_humor
- mild_blood
- mild_cartoon_violence
- mild_fantasy_violence
- mild_language
- mild_lyrics
- mild_suggestive_themes
- mild_violence
- no_descriptors
- nudity
- partial_nudity

- sexual_content
- sexual_themes
- simulated_gambling
- strong_language
- strong_sexual_content
- suggestive_themes
- use_of_alcohol
- use_of_drugs_and_alcohol
- violence
- esrb_rating

3. [3-vgsales.csv](#)

Скорочений варіант першого датасету. Його перевагами є те, що він більш детально заповнений, проте має меншу вибірку та менше полів.

Цей набір даних містить список відеоігор із продажами понад 100 000 копій. Він був створений за допомогою [vgchartz.com](#).

Перелік полів:

- Rank – рейтинг загальних продажів
- Name – назва гри
- Platform – платформа гри (наприклад, ПК, PS4, XOne тощо)
- Genre - жанр гри
- Publisher – видавець гри
- Global_Sales – загальний обсяг продажів у всьому світі (у мільйонах)
- NA_Sales – продажі в Північній Америці (у мільйонах)
- EU_Sales – продажі в Європі (у мільйонах)
- JP_Sales – продажі в Японії (у мільйонах)
- Other_Sales - Продажі в решті світу (у мільйонах)
- Year - Рік випуску гри

Він заснований на BeautifulSoup з використанням Python. Існує 16 598 записів.

Текст функцій / процедур або конфігурування ETL засобів

Створення бази даних,

```
CREATE SCHEMA IF NOT EXISTS videogames;  
USE videogames;
```

Створення стейджу для першого датасету. Мовою Python створюється таблиця та завантажуються дані. На жаль, таким способом не можливо підгрузити дані зі значенням NULL, тому на самому початку вони переформатуються, а потім у самій СКБД по введеному значенню повертається оригінал датасету.

```
import pandas as pd  
import mysql.connector as mysql  
from mysql.connector import Error  
  
data = pd.read_csv('datasets/1-vgsales.csv')  
  
# замінити пропущені строкові дані на empty  
data.fillna({'Name': 'N/A', 'Basename': 'N/A', 'Genre': 'N/A', 'ESRB_Rating': 'N/A', 'Platform': 'N/A',  
            'Publisher': 'N/A', 'Developer': 'N/A', 'url': 'N/A', 'img_url': 'N/A'}, inplace=True)  
  
# замінити пропущені числові дані на -555, дати на 2055 рік для подальшої заміни на NULL СКБД засобами  
data.fillna({'Rank': -555, 'VGChartz_Score': -555, 'Critic_Score': -555, 'User_Score': -555,  
            'Total_Shipped': -555, 'Global_Sales': -555, 'NA_Sales': -555, 'PAL_Sales': -555,  
            'JP_Sales': -555, 'Other_Sales': -555, 'Year': 1901.0, 'Last_Update': '01 Jan 55', 'status':  
            -555,  
            'Vgchartzscore': -555}, inplace=True)  
  
# форматування 2020.0 -> 2020 та '24th Jan 2018' -> 24 Jan 2018 -> 2018-06-24  
data['Year'] = data['Year'].round(0).astype(int)  
data['Last_Update'].replace(r'(\d)(st|nd|rd|th)', r'\1', regex=True, inplace=True)  
data['Last_Update'] = pd.to_datetime(data['Last_Update'], format="%d %b %y")  
  
def go(f, seed, acc):  
    seed, acc => {  
        const res = f(seed)  
        return res ? go(f, res[1], acc.concat([res[0]])) : acc  
    }  
    return go(f, seed, [])  
}
```

```

try:
    conn = mysql.connect(host="localhost", user="root", password="21ranwan11", database="videogames")
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You're connected to database: ", record)
        cursor.execute('DROP TABLE IF EXISTS stage_vgsales;')
        print('Creating table stage_vgsales....')
        cursor.execute("CREATE TABLE stage_vgsales (`rank` INT NOT NULL, "
            "`name` CHAR(160) DEFAULT NULL, `basename` CHAR(160) DEFAULT NULL, "
            "`genre` CHAR(50) DEFAULT NULL, `esrb_rating` CHAR(10) DEFAULT NULL, "
            "`platform` CHAR(10) DEFAULT NULL, `publisher` CHAR(50) DEFAULT NULL, "
            "`developer` CHAR(160) DEFAULT NULL, `vgchartz_score` FLOAT(10, 2) DEFAULT NULL, "
            "`critic_score` FLOAT(10, 2) DEFAULT NULL, `user_score` FLOAT(10, 2) DEFAULT NULL, "
            "`total_shipped` FLOAT(10, 2) DEFAULT NULL, `global_sales` FLOAT(10, 2) DEFAULT NULL, "
            "`na_sales` FLOAT(10, 2) DEFAULT NULL, `pal_sales` FLOAT(10, 2) DEFAULT NULL, "
            "`jp_sales` FLOAT(10, 2) DEFAULT NULL, `other_sales` FLOAT(10, 2) DEFAULT NULL, "
            "`year` YEAR DEFAULT NULL, `last_update` DATE DEFAULT NULL, "
            "`url` CHAR(255) DEFAULT NULL, `status` INT DEFAULT NULL, "
            "`vgchartzscore` FLOAT(10, 2) DEFAULT NULL, `img_url` CHAR(255) DEFAULT NULL)")
        print("Table stage_vgsales is created....")

    for i, row in data.iterrows():
        sql = "INSERT INTO videogames.stage_vgsales " \
            "VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
        tuplerow = tuple(row)
        cursor.execute(sql, tuplerow)
        print("Record inserted " + str(i))
        # the connection is not autocommited by default, so we must commit to save our changes
        conn.commit()
except Error as e:
    print("Error while connecting to MySQL", e)

```

Так само зчитуються дані з другого та третього датасетів.

```

import pandas as pd
import mysql.connector as mysql
from mysql.connector import Error

data = pd.read_csv('datasets/2-Video_games_esrb_rating.csv')

try:
    conn = mysql.connect(host="localhost", user="root", password="21ranwan11", database="videogames")
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You're connected to database: ", record)
        cursor.execute('DROP TABLE IF EXISTS stage_esrb_rating;')
        print('Creating table stage_esrb_rating....')
        cursor.execute("CREATE TABLE stage_esrb_rating ( "
            "title CHAR(160) , console TINYINT(1), alcohol_reference TINYINT(1), animated_blood TINYINT(1), "
            "blood TINYINT(1), blood_and_gore TINYINT(1), cartoon_violence TINYINT(1), "
            "crude_humor TINYINT(1), drug_reference TINYINT(1), fantasy_violence TINYINT(1),"
            "intense_violence TINYINT(1), language TINYINT(1), lyrics TINYINT(1), mature_humor TINYINT(1),"
            "mild_blood TINYINT(1), mild_cartoon_violence TINYINT(1), mild_fantasy_violence TINYINT(1),"
            "mild_language TINYINT(1), mild_lyrics TINYINT(1), mild_suggestive_themes TINYINT(1), "
            "mild_violence TINYINT(1), no_descriptors TINYINT(1), nudity TINYINT(1), "
            "partial_nudity TINYINT(1), sexual_content TINYINT(1), sexual_themes TINYINT(1), "
            "simulated_gambling TINYINT(1), strong_language TINYINT(1), strong_sexual_content TINYINT(1),"
            "suggestive_themes TINYINT(1), use_of_alcohol TINYINT(1), use_of_drugs_and_alcohol TINYINT(1),"
            "violence TINYINT(1), esrb_rating CHAR(10) DEFAULT NULL);")
        print("Table stage_esrb_rating is created....")

```

```
for i, row in data.iterrows():
    sql = "INSERT INTO videogames.stage_esrb_rating " \
          "VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s) \" \
          \"%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s\""
    tuplerow = tuple(row)
    cursor.execute(sql, tuplerow)
    print("Record inserted " + str(i))
    # the connection is not autocommited by default, so we must commit to save our changes
    conn.commit()
except Error as e:
    print("Error while connecting to MySQL", e)
```

```
import pandas as pd
import mysql.connector as mysql
from mysql.connector import Error

data = pd.read_csv('datasets/3-vgsales.csv')

data.fillna({'Year': 1901, 'Publisher': ''}, inplace=True)

try:
    conn = mysql.connect(host="localhost", user="root", password="21ranwan11", database="videogames")
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You're connected to database: ", record)
        cursor.execute('DROP TABLE IF EXISTS stage_little_vgsales;')
        print('Creating table stage_little_vgsales....')
        cursor.execute("CREATE TABLE stage_little_vgsales (`rank` INT NOT NULL,`name` CHAR(160) DEFAULT NULL, "
            "`platform` CHAR(10) DEFAULT NULL, `year` YEAR DEFAULT NULL, `genre` CHAR(50) DEFAULT NULL, "
            "`publisher` CHAR(50) DEFAULT NULL, `na_sales` FLOAT(10, 2) DEFAULT NULL, "
            "`eu_sales` FLOAT(10, 2) DEFAULT NULL, `jp_sales` FLOAT(10, 2) DEFAULT NULL, "
            "`other_sales` FLOAT(10, 2) DEFAULT NULL, `global_sales` FLOAT(10, 2) DEFAULT NULL);")
        print("Table stage_little_vgsales is created....")

    for i, row in data.iterrows():
        sql = "INSERT INTO videogames.stage_little_vgsales " \
            "VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
        tuplerow = tuple(row)
        cursor.execute(sql, tuplerow)
        print("Record inserted " + str(i))
        # the connection is not autocommited by default, so we must commit to save our changes
        conn.commit()
except Error as e:
    print("Error while connecting to MySQL", e)
```

На даному етапі створилися три стейджу разом з даними трьох датасетів.

```
SELECT * FROM videogames.stage_vgsales;
```

```
SELECT * FROM videogames.stage esrb rating;
```

```
SELECT * FROM videogames.stage little vgsales;
```

SQL-запити для реформатування даних після завантаження даних до БД засобами мови Python.


```
UPDATE videogames.stage_vgsales SET vgchartz_score = NULL WHERE  
vgchartz_score = -555;
```

```
UPDATE videogames.stage_vgsales SET critic_score = NULL WHERE  
critic_score = -555;
```

```
UPDATE videogames.stage_vgsales SET user_score = NULL WHERE  
user_score = -555;
```

```
UPDATE videogames.stage_vgsales SET total_shipped = NULL WHERE  
total_shipped = -555;
```

```
UPDATE videogames.stage_vgsales SET global_sales = NULL WHERE  
global_sales = -555;
```

```
UPDATE videogames.stage_vgsales SET na_sales = NULL WHERE  
na_sales = -555;
```

```
UPDATE videogames.stage_vgsales SET pal_sales = NULL WHERE  
pal_sales = -555;
```

```
UPDATE videogames.stage_vgsales SET jp_sales = NULL WHERE  
jp_sales = -555;
```

```
UPDATE videogames.stage_vgsales SET other_sales = NULL WHERE  
other_sales = -555;
```

```
UPDATE videogames.stage_vgsales SET vgchartzscore = NULL WHERE  
vgchartzscore = -555;
```

```
UPDATE videogames.stage_vgsales SET `year` = NULL WHERE `year` =  
2055;
```

```
UPDATE videogames.stage_vgsales SET last_update = NULL WHERE  
last_update = '2055-1-1';
```

**Команда MySQL для отримання дозволу апдейтити таблиці не через
первинні ключі.**

```
SET SQL_SAFE_UPDATES = 0;
```

```
USE videogames;
```

На основі даних з «великого» стейджу продаж ігор (перший датасет), були створені шість вимірів: видавці гри, жанри гри, платформи, розробники, тип оцінювання за рейтингом ESRB, рік видання.

```
-- ----- publishers -----
```

```
DROP TABLE IF EXISTS publishers;
```

```
CREATE TABLE IF NOT EXISTS publishers (
```

```
    `id` INT PRIMARY KEY AUTO_INCREMENT,
```

```
    `name` CHAR(50) UNIQUE);
```

```
INSERT INTO publishers (name) (
```

```
SELECT DISTINCT publisher FROM (SELECT publisher FROM  
stage_vgsales
```

```
    UNION
```

```
    SELECT publisher FROM stage_little_vgsales) as s3) ;
```

```
SELECT * FROM videogames.publishers ORDER BY id ASC;
```

```
-- ----- genres -----
```

```
DROP TABLE IF EXISTS genres;
```

```
CREATE TABLE IF NOT EXISTS genres (
```

```
    id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    `name` VARCHAR(50) UNIQUE);
```

```
INSERT INTO genres (name) (
```

```
SELECT DISTINCT genre FROM (SELECT genre FROM stage_vgsales
```

```
    UNION
```

```

        SELECT genre FROM stage_little_vgsales) as s3);

SELECT * FROM videogames.genres ORDER BY id ASC;

-- ----- platform -----

DROP TABLE IF EXISTS platforms;

CREATE TABLE IF NOT EXISTS platforms (
    id INT PRIMARY KEY AUTO_INCREMENT,
    `name` VARCHAR(10) UNIQUE);

INSERT INTO platforms (name) (
SELECT DISTINCT platform FROM (SELECT platform FROM
stage_vgsales

    UNION

    SELECT platform FROM stage_little_vgsales) as s3);

SELECT * FROM videogames.platforms ORDER BY id ASC;

-- ----- developer -----

DROP TABLE IF EXISTS developers;

CREATE TABLE IF NOT EXISTS developers (
    id INT PRIMARY KEY AUTO_INCREMENT,
    `name` VARCHAR(160) UNIQUE);

INSERT INTO developers (name) (
SELECT DISTINCT developer FROM stage_vgsales);

```

```
SELECT * FROM videogames.developers ORDER BY id ASC;
```

```
-- ----- esrb_rating type -----
```

```
DROP TABLE IF EXISTS esrb_ratings;
```

```
CREATE TABLE IF NOT EXISTS esrb_ratings (
```

```
    id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    `name` VARCHAR(10) UNIQUE,
```

```
    ful_name VARCHAR(50) UNIQUE);
```

```
INSERT INTO esrb_ratings (name) (
```

```
SELECT DISTINCT esrb_rating FROM (SELECT esrb_rating FROM  
stage_vgsales
```

```
    UNION
```

```
    SELECT esrb_rating FROM stage_esrb_rating) as s3);
```

```
SELECT * FROM videogames.esrb_ratings ORDER BY id ASC;
```

```
-- ----- release_year -----
```

```
DROP TABLE IF EXISTS release_year;
```

```
CREATE TABLE IF NOT EXISTS release_year (
```

```
    id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    `year` YEAR UNIQUE);
```

```
INSERT INTO release_year (`year`) (
```

```
SELECT DISTINCT `year` FROM (SELECT `year` FROM stage_vgsales
```

```
    UNION
```

```
    SELECT `year` FROM stage_little_vgsales) as s3);
```

```
SELECT * FROM videogames.release_year ORDER BY id ASC;
```

Наступним кроком є створення фактової таблиці. До складу її полів увійдуть усі поля першої та другої таблиць, а також зовнішні ключі на виміри.

```
SET SQL_SAFE_UPDATES = 0;
```

```
CREATE TABLE IF NOT EXISTS fact_vgsales (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  `rank` INT NOT NULL,  
  `name` CHAR(160) DEFAULT NULL,  
  platform CHAR(10) DEFAULT NULL,  
  basename CHAR(160) DEFAULT NULL,  
  genre_id INT NOT NULL,  
  esrb_rating_id INT NOT NULL,  
  platform_id INT NOT NULL,  
  publisher_id INT NOT NULL,  
  developer_id INT NOT NULL,  
  year_id INT NOT NULL,  
  critic_score FLOAT(10,2) DEFAULT NULL,  
  user_score FLOAT(10,2) DEFAULT NULL,  
  total_shipped FLOAT(10,2) DEFAULT NULL,  
  global_sales FLOAT(10,2) DEFAULT NULL,  
  na_sales FLOAT(10,2) DEFAULT NULL,  
  pal_sales FLOAT(10,2) DEFAULT NULL,  
  eu_sales FLOAT(10,2) DEFAULT NULL,  
  jp_sales FLOAT(10,2) DEFAULT NULL,  
  other_sales FLOAT(10,2) DEFAULT NULL,  
  last_update DATE DEFAULT NULL,  
  url CHAR(255) DEFAULT NULL,  
  `status` INT DEFAULT NULL,  
  vgchartzscore FLOAT(10,2) DEFAULT NULL,  
  img_url CHAR(255) DEFAULT NULL,  
  console TINYINT(1) DEFAULT NULL,  
  alcohol_reference TINYINT(1) DEFAULT NULL,  
  animated_blood TINYINT(1) DEFAULT NULL,  
  blood TINYINT(1) DEFAULT NULL,  
  blood_and_gore TINYINT(1) DEFAULT NULL,  
  cartoon_violence TINYINT(1) DEFAULT NULL,  
  crude_humor TINYINT(1) DEFAULT NULL,  
  drug_reference TINYINT(1) DEFAULT NULL,  
  fantasy_violence TINYINT(1) DEFAULT NULL,  
  intense_violence TINYINT(1) DEFAULT NULL,  
  `language` TINYINT(1) DEFAULT NULL,  
  lyrics TINYINT(1) DEFAULT NULL,  
  mature_humor TINYINT(1) DEFAULT NULL,  
  mild_blood TINYINT(1) DEFAULT NULL,  
  mild_cartoon_violence TINYINT(1) DEFAULT NULL,  
  mild_fantasy_violence TINYINT(1) DEFAULT NULL,  
  mild_language TINYINT(1) DEFAULT NULL,
```

```

mild_lyrics TINYINT(1) DEFAULT NULL,
mild_suggestive_themes TINYINT(1) DEFAULT NULL,
mild_violence TINYINT(1) DEFAULT NULL,
no_descriptors TINYINT(1) DEFAULT NULL,
nudity TINYINT(1) DEFAULT NULL,
partial_nudity TINYINT(1) DEFAULT NULL,
sexual_content TINYINT(1) DEFAULT NULL,
sexual_themes TINYINT(1) DEFAULT NULL,
simulated_gambling TINYINT(1) DEFAULT NULL,
strong_janguage TINYINT(1) DEFAULT NULL,
strong_sexual_content TINYINT(1) DEFAULT NULL,
suggestive_themes TINYINT(1) DEFAULT NULL,
use_of_alcohol TINYINT(1) DEFAULT NULL,
use_of_drugs_and_alcohol TINYINT(1) DEFAULT NULL,
violence TINYINT(1) DEFAULT NULL,
KEY `nm_platform` (`name`, `platform`),
CONSTRAINT vgsales_publishers FOREIGN KEY (publisher_id)
REFERENCES publishers (id) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT vgsales_genres FOREIGN KEY (genre_id) REFERENCES
genres (id) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT vgsales_esrb_ratings FOREIGN KEY (esrb_rating_id)
REFERENCES esrb_ratings (id) ON DELETE CASCADE ON UPDATE
CASCADE,
CONSTRAINT vgsales_platforms FOREIGN KEY (platform_id)
REFERENCES platforms (id) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT vgsales_developers FOREIGN KEY (developer_id)
REFERENCES developers (id) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT vgsales_release_year FOREIGN KEY (year_id) REFERENCES
release_year (id) ON DELETE CASCADE ON UPDATE CASCADE
);

SELECT * FROM fact_vgsales;
DROP TABLE IF EXISTS fact_vgsales;

```

У фактову таблицю за допомогою JOIN та вкладених запитів заповнюються дані з першого та другого стейджів, кожне поле належить окремому стейджу.

```

INSERT INTO fact_vgsales
(`rank`, `name`, platform, basename, critic_score, user_score,
total_shipped, global_sales, na_sales,
pal_sales, jp_sales, other_sales, last_update, url, `status`,
vgchartzscore, img_url, genre_id, esrb_rating_id, platform_id,
publisher_id, developer_id, year_id, animated_blood, console,
alcohol_reference, blood, blood_and_gore, cartoon_violence,
crude_humor, drug_reference, fantasy_violence, intense_violence,
`language`, lyrics, mature_humor, mild_blood,
mild_cartoon_violence, mild_fantasy_violence, mild_language,
mild_lyrics, mild_suggestive_themes, mild_violence,
no_descriptors, nudity, partial_nudity, sexual_content,
sexual_themes, simulated_gambling, strong_janguage,
strong_sexual_content, suggestive_themes, use_of_alcohol,
use_of_drugs_and_alcohol, violence)

```

```

(SELECT vgs.rank, vgs.name, vgs.platform, vgs.basename,
vgs.critic_score, vgs.user_score, vgs.total_shipped,
vgs.global_sales, vgs.na_sales,
vgs.pal_sales, vgs.jp_sales, vgs.other_sales, vgs.last_update,
vgs.url, vgs.`status`, vgs.vgchartzscore, vgs.img_url,
g.id, esrb.id, pl.id, p.id, d.id, y.id, st.animated_blood,
st.console, st.alcohol_reference, st.blood, st.blood_and_gore,
st.cartoon_violence, st.crude_humor, st.drug_reference,
st.fantasy_violence, st.intense_violence, st.language,
st.lyrics, st.mature_humor, st.mild_blood,
st.mild_cartoon_violence, st.mild_fantasy_violence,
st.mild_language, st.mild_lyrics,
st.mild_suggestive_themes, st.mild_violence, st.no_descriptors,
st.nudity, st.partial_nudity, st.sexual_content,
st.sexual_themes, st.simulated_gambling, st.strong_language,
st.strong_sexual_content, st.suggestive_themes,
st.use_of_alcohol, st.use_of_drugs_and_alcohol,
st.violence
FROM stage_vgsales AS vgs
LEFT JOIN genres AS g ON (vgs.genre = g.name)
LEFT JOIN esrb_ratings AS esrb ON (vgs.esrb_rating =
esrb.name)
LEFT JOIN platforms AS pl ON (vgs.platform = pl.name)
LEFT JOIN publishers AS p ON (vgs.publisher = p.name)
LEFT JOIN developers AS d ON (vgs.developer = d.name)
LEFT JOIN release_year AS y ON (vgs.year = y.year)
LEFT JOIN stage_esrb_rating AS st ON (vgs.name =
st.title)
);

```

Окремим апдейтом за зовнішніми ключами імені та платформи гри у деякі поля з першого стейджу дописуються дані з третього. Це зроблено задля доповнення інформації, адже перший та третій датасети мають спільні поля, але перший датасет може не мати інформацію, що є в третьому.

```

UPDATE fact_vgsales AS f
LEFT JOIN stage_little_vgsales as b
ON (f.name = b.name and f.platform = b.platform)
SET f.other_sales = b.other_sales, f.na_sales = b.na_sales,
f.jp_sales = b.jp_sales, f.global_sales = b.global_sales,
f.eu_sales = b.eu_sales;

```

```

SELECT * FROM fact_vgsales;

```

Моделі Stage зони та основного сховища

stage_vgsales	
rank	INT
name	CHAR(160)
basenam e	CHAR(160)
genre	CHAR(50)
esrb_rating	CHAR(10)
platform	CHAR(10)
publisher	CHAR(50)
developer	CHAR(160)
vgchartz_score	FLOAT(10,2)
critic_score	FLOAT(10,2)
user_score	FLOAT(10,2)
total_shipped	FLOAT(10,2)
global_sales	FLOAT(10,2)
na_sales	FLOAT(10,2)
pal_sales	FLOAT(10,2)
jp_sales	FLOAT(10,2)
other_sales	FLOAT(10,2)
year	YEAR
last_update	DATE
url	CHAR(255)
status	INT
vgchartzscore	FLOAT(10,2)
img_url	CHAR(255)

fact_vgsales	
id	INT
rank	INT
name	CHAR(160)
platform	CHAR(10)
basenam e	CHAR(160)
genre_id	INT
esrb_rating_id	INT
platform_id	INT
publisher_id	INT
developer_id	INT
year_id	INT
critic_score	FLOAT(10,2)
user_score	FLOAT(10,2)
total_shipped	FLOAT(10,2)
global_sales	FLOAT(10,2)
na_sales	FLOAT(10,2)
pal_sales	FLOAT(10,2)
eu_sales	FLOAT(10,2)
jp_sales	FLOAT(10,2)
other_sales	FLOAT(10,2)
last_update	DATE
url	CHAR(255)
status	INT
vgchartzscore	FLOAT(10,2)
img_url	CHAR(255)
console	TINYINT(1)
alcohol_reference	TINYINT(1)
animated_blood	TINYINT(1)
blood	TINYINT(1)
blood_and_gore	TINYINT(1)
carbon_violence	TINYINT(1)
crude_hum or	TINYINT(1)
drug_reference	TINYINT(1)
fantasy_violence	TINYINT(1)
intense_violence	TINYINT(1)
language	TINYINT(1)
lyrics	TINYINT(1)
mature_humor	TINYINT(1)
mild_blood	TINYINT(1)
mild_cartoon_violence	TINYINT(1)
mild_fantasy_violence	TINYINT(1)
mild_language	TINYINT(1)
mild_lyrics	TINYINT(1)
mild_suggestive_themes	TINYINT(1)
mild_violence	TINYINT(1)
no_descriptors	TINYINT(1)
nudity	TINYINT(1)
partial_nudity	TINYINT(1)
sexual_content	TINYINT(1)
sexual_themes	TINYINT(1)
sim ulated_gam bling	TINYINT(1)
strong_language	TINYINT(1)
strong_sexual_content	TINYINT(1)
suggestive_themes	TINYINT(1)
use_of_alcohol	TINYINT(1)
use_of_drugs_and_alcohol	TINYINT(...)
violence	TINYINT(1)

genres	
id	INT
name	VARCHAR(50)
Indexes	

developers	
id	INT
name	VARCHAR(160)
Indexes	

platforms	
id	INT
name	VARCHAR(10)
Indexes	

esrb_ratings	
id	INT
name	VARCHAR(10)
ful_name	VARCHAR(50)
Indexes	

publishers	
id	INT
name	CHAR(50)
Indexes	

release_year	
id	INT
year	YEAR
Indexes	

stage_esrb_rating	
title	CHAR(160)
console	TINYINT(1)
alcohol_reference	TINYINT(1)
anim ated_blood	TINYINT(1)
blood	TINYINT(1)
blood_and_gore	TINYINT(1)
cartoon_violence	TINYINT(1)
crude_hum or	TINYINT(1)
drug_reference	TINYINT(1)
fantasy_violence	TINYINT(1)
intense_violence	TINYINT(1)
language	TINYINT(1)
lyrics	TINYINT(1)
mature_humor	TINYINT(1)
mild_blood	TINYINT(1)
mild_cartoon_violence	TINYINT(1)
mild_fantasy_violence	TINYINT(1)
mild_language	TINYINT(1)
mild_lyrics	TINYINT(1)
mild_suggestive_themes	TINYINT(1)
mild_violence	TINYINT(1)
no_descriptors	TINYINT(1)
nudity	TINYINT(1)
partial_nudity	TINYINT(1)
sexual_content	TINYINT(1)
sexual_themes	TINYINT(1)
sim ulated_gam bling	TINYINT(1)
strong_language	TINYINT(1)
strong_sexual_content	TINYINT(1)
suggestive_themes	TINYINT(1)
use_of_alcohol	TINYINT(1)
use_of_drugs_and_alcohol	TINYINT(1)
violence	TINYINT(1)
esrb_rating	CHAR(10)

stage_little_vgsales	
rank	INT
name	CHAR(160)
platform	CHAR(10)
year	YEAR
genre	CHAR(50)
publisher	CHAR(50)
na_sales	FLOAT(10,2)
eu_sales	FLOAT(10,2)
jp_sales	FLOAT(10,2)
other_sales	FLOAT(10,2)
global_sales	FLOAT(10,2)
Indexes	

Висновок: у ході даного комп'ютерного практикуму було ознайомлено зі сховищем даних і вимірами, ETL-засобами, схемами «зірка» та «сніжинка»; були здобуті практичні навички створення багатовимірних моделей.