Додаток 1

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»

на тему: «Створення сховища даних»

Виконав студент ІП-13, Бабашев Олексій Дмитрович
(шифр, прізвище, ім'я, по батькові)

Перевірив Олійник Юрій Олександрович
(прізвище, ім'я, по батькові)

Київ 2023

Комп’ютерний практикум 1

Тема – створення сховища даних.

Мета – ознайомитись з підходами до створення сховищ даних.

Для виконання даної лабораторної роботи було вибрано даний набір даних:
https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016
https://www.kaggle.com/datasets/fernandol/countries-of-the-world
https://www.kaggle.com/datasets/alcidesoxa/world-happiness-report-2005-2018
https://www.kaggle.com/datasets/madhurpant/world-deaths-and-causes-1990-2019

Даний набір містить інформацію про випадки смертей в країнах світу за певний період часу. Також цей набір даних містить статистичну інформацію про різні аспекти здоров'я, економіки та соціальних показників різних країн світу.

Таблиця annual_deaths_by_causes містить дані про випадки та причини смертей в країнах за певний період часу.

Таблиця suicide rates1985-2016 містить інформацію про випадки суїцидів в країнах світу за різний період часу.

Таблиця countries of the world.csv містить інформацію про країни світу, їх населення, територію і тд.

Таблиця world-happiness-report-2005-2018 містить дані про умови проживання в країнах світу, індекс демократії, тривалість життя та інше.

Форматування даних

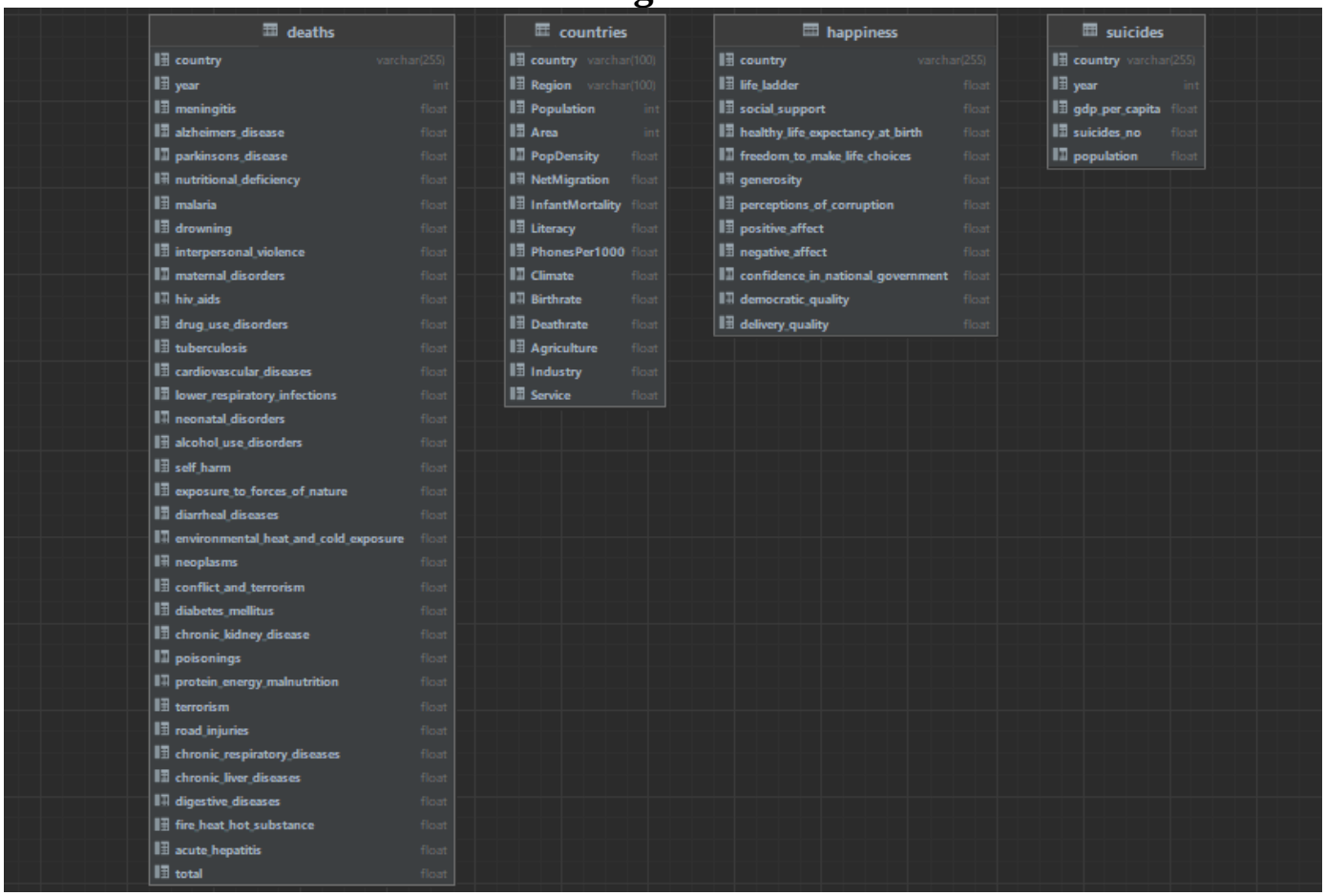
Для форматування даних використаємо python скрипт наведений нижче:

```
import pandas as pd
#DEATHES
df = pd.read_csv('../dataset/annual_deaths_by_causes.csv')

#Cleaning from extra spaces in columns and data Replace NaN with 0
df.columns = df.columns.str.strip()
df = df.apply(lambda x:x.str.strip() if x.dtype == "object" else x)
df.fillna(0, inplace=True)

#delete extra columns
df.drop('code',axis=1, inplace=True)

#add column sum of deaths
df['total'] = df.iloc[:,2:].sum(axis=1)

# Write the modified DataFrame to a new CSV file
df.to_csv('C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cleaned-dataset/deaths1990-2019.csv', index=False)
#%%
#SUICIDES
df = pd.read_csv('../dataset/suicide rates1985-2016.csv')

#Cleaning from extra spaces in columns and data Replace NaN with 0
df.columns = df.columns.str.strip()
df = df.apply(lambda x:x.str.strip() if x.dtype == "object" else x)
df.fillna(0, inplace=True)

# group by 'country', 'year', 'gdp_per_capita' columns and aggregate 'suicides_no' and 'population' columns
```

```
df = df.groupby(['country','year','gdp_per_capita ($)'], as_index=False).agg({'suicides_no': 'sum', 'population': 'sum'})
#after this you do not have to delete extra columns!

df.to_csv('C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cleaned-dataset/suicides1985-2016GROUPED.csv', index=False)
#%%
#HAPPINESS
# Read the CSV file with ';' delimiter
df = pd.read_csv('../dataset/world-happiness-report-2005-2018.csv', sep=';')

#Cleaning from extra spaces in columns and data Replace NaN with 0
df.columns = df.columns.str.strip()
df = df.apply(lambda x:x.str.strip() if x.dtype == "object" else x)
df.fillna(0, inplace=True)

#delete extra column
df.drop('Year',axis=1, inplace=True)

#making the same name of the same countries
stand_name = {
    'Congo (Brazzaville)':'Congo',
    'Congo (Kinshasa)':'Congo'
}
df['Country name'] = df['Country name'].replace(stand_name)

#group by country name and find average of other data of different years
df = df.groupby(['Country name'],as_index=False).mean()

# Save the file with ',' delimiter
df.to_csv('C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cleaned-
```

```
dataset/happiness.csv', sep=',', index=False)
#%%
#COUNTRIES
df = pd.read_csv('../dataset/countries of the world.csv')

#Cleaning from extra spaces in columns and data Replace NaN with 0
df.columns = df.columns.str.strip()
df = df.apply(lambda x:x.str.strip() if x.dtype == "object" else x)
df.fillna(0, inplace=True)



# replace commas with periods in the numeric columns
df = df.replace(',', '.', regex=True)

mapping = {
    'ASIA (EX. NEAR EAST)': 'ASIA',
    'EASTERN EUROPE': 'EUROPE',
    'NORTHERN AFRICA': 'AFRICA',
    'WESTERN EUROPE': 'EUROPE',
    'SUB-SAHARAN AFRICA': 'AFRICA',
    'LATIN AMER. & CARIB': 'LATIN AMERICA',
    'C.W. OF IND. STATES': 'ASIA',
    'NEAR EAST': 'ASIA',
    'BALTICS': 'EUROPE'
}
df['Region'] = df['Region'].replace(mapping)

df.to_csv('C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cleaned-
dataset/countries.csv', index=False)
```

Stage зона



Таблиця "deaths":
"country" - назва країни, де сталися смерті
"year" - рік, коли сталися смерті
Колонки, що містять назви різних хвороб і причин смерті, такі як "meningitis", "alzheimers_disease", "malaria" і т.д. Кожна з цих колонок містить кількість смертей, пов'язаних з відповідною хворобою або причиною.
"total" - сумарний показник усіх смертей за цей рік.

Таблиця "suicides":
"country" - назва країни, де сталися суїциди
"year" - рік, коли сталися суїциди
"gdp_per_capita" - ВВП на душу населення
"suicides_no" - кількість суїцидів
"population" - загальна кількість населення

Таблиця "happiness":
"country" - назва країни
"life_ladder" - загальний показник щастя за допомогою рейтингу
"social_support" - рівень соціальної підтримки, яку люди отримують від своїх родин, друзів і інших людей
"healthy_life_expectancy_at_birth" - очікувана тривалість здорового життя при народженні
"freedom_to_make_life_choices" - рівень свободи людини в прийнятті власних життєвих рішень

"generosity" - рівень щедрості відповідної країни
"perceptions_of_corruption" - сприйняття корупції відповідної країни
"positive_affect" - рівень позитивних емоцій у громадян країни
"negative_affect" - рівень негативних емоцій у громадян країни
"confidence_in_national_government" - рівень довіри національному уряду
"democratic_quality" - якість демократії відповідної країни
"delivery_quality" - якість державних послуг відповідної країни

Таблиця "countries":
"country" - назва країни
"Region" - регіон, до якого належить країна
"Population

MySQL скрипти для створення stage:

```
drop database if exists stage;

create database stage;

use stage;

drop table if exists suicides;
drop table if exists deaths;
drop table if exists happiness;


CREATE TABLE deaths
(
    country                             VARCHAR(255),
    `year`                              INT,
    meningitis                          FLOAT,
    alzheimers_disease                  FLOAT,
    parkinsons_disease                  FLOAT,
    nutritional_deficiency              FLOAT,
    malaria                             FLOAT,
    drowning                            FLOAT,
    interpersonal_violence              FLOAT,
    maternal_disorders                  FLOAT,
    hiv_aids                            FLOAT,
    drug_use_disorders                  FLOAT,
    tuberculosis                        FLOAT,
    cardiovascular_diseases             FLOAT,
    lower_respiratory_infections        FLOAT,
    neonatal_disorders                  FLOAT,
    alcohol_use_disorders               FLOAT,
    self_harm                           FLOAT,
    exposure_to_forces_of_nature        FLOAT,
    diarrheal_diseases                  FLOAT,
    environmental_heat_and_cold_exposure FLOAT,
    neoplasms                           FLOAT,
    conflict_and_terrorism              FLOAT,
    diabetes_mellitus                   FLOAT,
    chronic_kidney_disease              FLOAT,
    poisonings                          FLOAT,
    protein_energy_malnutrition         FLOAT,
    terrorism                           FLOAT,
    road_injuries                       FLOAT,
    chronic_respiratory_diseases        FLOAT,
    chronic_liver_diseases              FLOAT,
    digestive_diseases                  FLOAT,
    fire_heat_hot_substance             FLOAT,
    acute_hepatitis                     FLOAT,
```

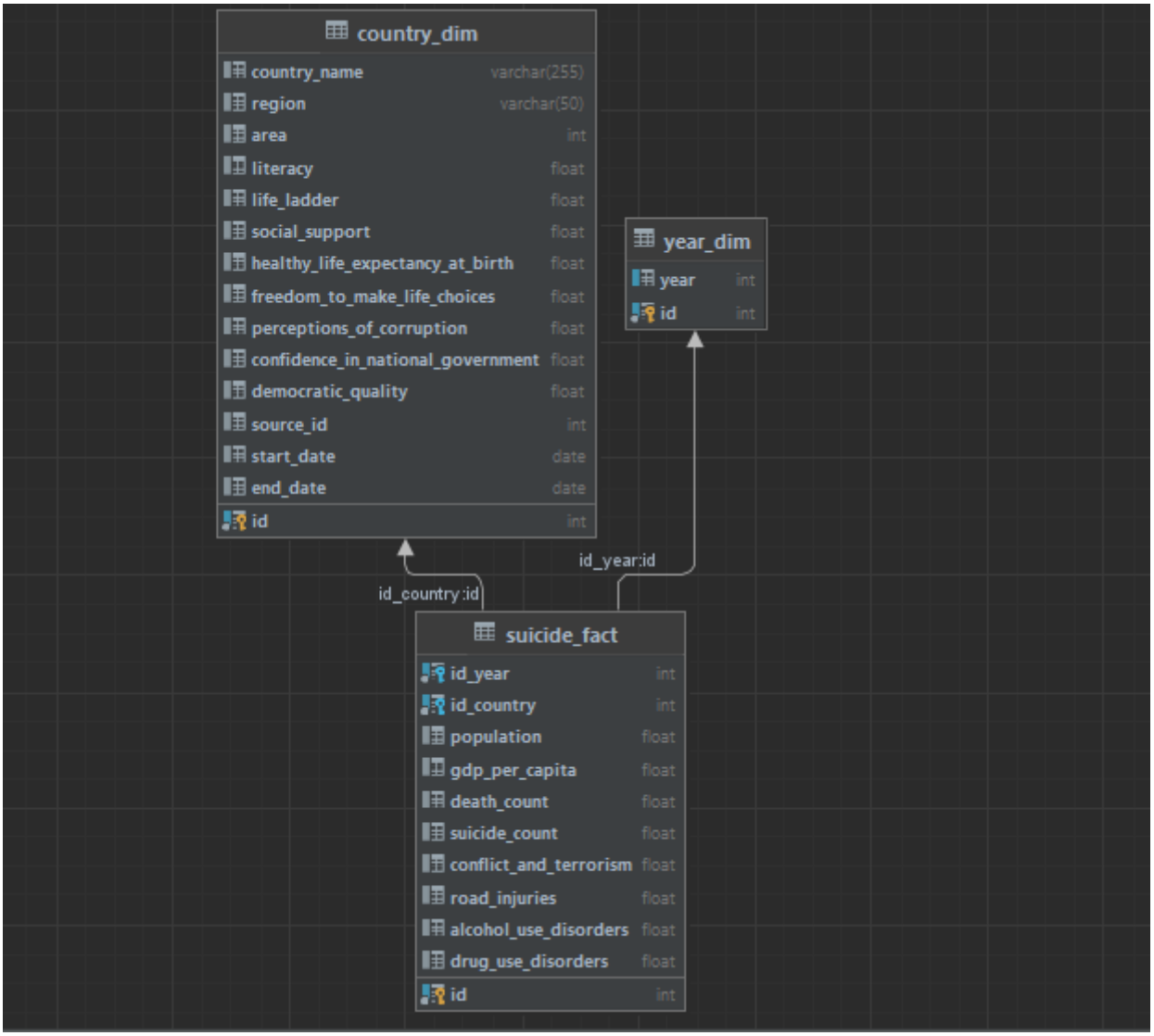
```
    total FLOAT
);


CREATE TABLE suicides
(
    country         VARCHAR(255),
    `year`          INT,
    gdp_per_capita FLOAT,
    suicides_no     FLOAT,
    population      FLOAT
);


CREATE TABLE happiness
(
    country                                     VARCHAR(255),
    life_ladder                                 FLOAT,
    social_support                              FLOAT,
    healthy_life_expectancy_at_birth            FLOAT,
    freedom_to_make_life_choices                FLOAT,
    generosity                                  FLOAT,
    perceptions_of_corruption                   FLOAT,
    positive_affect                             FLOAT,
    negative_affect                             FLOAT,
    confidence_in_national_government           FLOAT,
    democratic_quality                          FLOAT,
    delivery_quality                            FLOAT
);

CREATE TABLE countries (
  country VARCHAR(100),
  Region VARCHAR(100),
  Population INT,
  Area INT,
  PopDensity FLOAT,
  NetMigration FLOAT,
  InfantMortality FLOAT,
  Literacy FLOAT,
  PhonesPer1000 FLOAT,
  Climate FLOAT,
  Birthrate FLOAT,
  Deathrate FLOAT,
  Agriculture FLOAT,
  Industry FLOAT,
  Service FLOAT
);
```

MySQL скрипт для заповнення stage зони даними:

```
use stage;

truncate table suicides;
truncate table deaths;
truncate table happiness;
truncate table countries;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cleaned-
dataset/deaths1990-2019.csv' INTO TABLE deaths
    FIELDS TERMINATED BY ','
    ENCLOSED BY '"'
    LINES TERMINATED BY '\n'
    IGNORE 1 ROWS;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cleaned-
dataset/suicides1985-2016GROUPED.csv' INTO TABLE suicides
    FIELDS TERMINATED BY ','
```

```
    ENCLOSED BY '"'
    LINES TERMINATED BY '\n'
    IGNORE 1 ROWS;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cleaned-
dataset/happiness.csv' INTO TABLE happiness
    FIELDS TERMINATED BY ','
    ENCLOSED BY '"'
    LINES TERMINATED BY '\n'
    IGNORE 1 ROWS
    (country, life_ladder, @skip, social_support, healthy_life_expectancy_at_birth,
freedom_to_make_life_choices,
     generosity, perceptions_of_corruption, positive_affect, negative_affect,
confidence_in_national_government,
     democratic_quality, delivery_quality, @skip, @skip, @skip, @skip, @skip, @skip,
@skip, @skip, @skip, @skip,
     @skip, @skip);


LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cleaned-
dataset/countries.csv' INTO TABLE countries
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(country, Region, Population, Area, PopDensity, @skip, NetMigration, InfantMortality,
@skip, Literacy, PhonesPer1000,
 @skip, @skip, @skip, Climate, Birthrate, Deathrate, Agriculture, Industry, Service);
```

Main сховище



Фактова таблиця містить зовнішні ключі на таблиці виміри, що містять дані про країни та роки. Також фактова таблиця містить інформацію про кількість смертей, суїцидів, смертей на дорогах, смертей від алкоголю та наркотиків, смертей вніслідок конфліктів та тероризму за конкретний рік в конкретній країні.

MySQL скрипти для створення main warehouse:

```
drop database if exists warehouse;
create database warehouse;

use warehouse;

drop table if exists country_dim;
drop table if exists year_dim;
drop table if exists suicide_fact;

CREATE TABLE country_dim
(
    id                                  INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    country_name                        VARCHAR(255),
    region                              VARCHAR(50),
    area                                INT,
    literacy                            FLOAT,
    life_ladder                         FLOAT,
```

```
    social_support                    FLOAT,
    healthy_life_expectancy_at_birth  FLOAT,
    freedom_to_make_life_choices      FLOAT,
    perceptions_of_corruption         FLOAT,
    confidence_in_national_government FLOAT,
    democratic_quality                FLOAT,
    source_id                         int  default null,
    start_date                        date default null,
    end_date                          date default null
);

CREATE TABLE year_dim
(
    id     INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    `year` INT UNIQUE
);

CREATE TABLE suicide_fact
(
    id                    INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    id_year               INT NOT NULL,
    id_country            INT NOT NULL,
    population            FLOAT,
    gdp_per_capita        FLOAT,
    death_count           FLOAT,
    suicide_count         FLOAT,
    conflict_and_terrorism FLOAT,
    road_injuries         FLOAT,
    alcohol_use_disorders FLOAT,
    drug_use_disorders    FLOAT,

    FOREIGN KEY (id_year) references year_dim (id),
    FOREIGN KEY (id_country) references country_dim (id)
);
```

MySQL скрипти для заповнення даними main warehouse:

```
use stage;

insert into warehouse.year_dim(`year`)
select distinct deaths.year
from deaths
         join suicides on deaths.year = suicides.year;

insert into warehouse.country_dim(country_name, region, area, literacy, life_ladder,
social_support,
                                  healthy_life_expectancy_at_birth,
freedom_to_make_life_choices,
                                  perceptions_of_corruption,
confidence_in_national_government, democratic_quality)
select distinct deaths.country,
                countries.Region,
                countries.Area,
                countries.Literacy,
                happiness.life_ladder,
                happiness.social_support,
                happiness.healthy_life_expectancy_at_birth,
                happiness.freedom_to_make_life_choices,
                happiness.perceptions_of_corruption,
                happiness.confidence_in_national_government,
                happiness.democratic_quality
from deaths
         join suicides on suicides.country = deaths.country
         join happiness on happiness.country = deaths.country
         join countries on countries.country = deaths.country;
```

```
insert into warehouse.suicide_fact(id_year, id_country, population, gdp_per_capita,
death_count,
                                   suicide_count, conflict_and_terrorism,
road_injuries,
                                   alcohol_use_disorders, drug_use_disorders)
select warehouse.year_dim.id,
       warehouse.country_dim.id,
       suicides.population,
       suicides.gdp_per_capita,
       deaths.total,
       suicides.suicides_no,
       deaths.conflict_and_terrorism,
       deaths.road_injuries,
       deaths.alcohol_use_disorders,
       deaths.drug_use_disorders
from suicides
         join warehouse.country_dim on suicides.country =
warehouse.country_dim.country_name
         join warehouse.year_dim on suicides.year = warehouse.year_dim.year
         join deaths on deaths.country = suicides.country and deaths.year =
suicides.year;
```

MySQL скрипти реалізації процедури, прикладу scd.

```
use warehouse;

drop procedure if exists slow_change_country;

delimiter //
create procedure slow_change_country(old_name varchar(50), new_name varchar(50))
begin
    declare old_id int default null;
    declare old_region VARCHAR(50);
    declare old_area INT;
    declare old_literacy FLOAT;
    declare old_life_ladder FLOAT;
    declare old_social_support FLOAT;
    declare old_healthy_life_expectancy_at_birth FLOAT;
    declare old_freedom_to_make_life_choices FLOAT;
    declare old_perceptions_of_corruption FLOAT;
    declare old_confidence_in_national_government FLOAT;
    declare old_democratic_quality FLOAT;


    select id,
           region,
           area,
           literacy,
           life_ladder,
           social_support,
           healthy_life_expectancy_at_birth,
           freedom_to_make_life_choices,
           perceptions_of_corruption,
           confidence_in_national_government,
           democratic_quality
    into
old_id,old_region,old_area,old_literacy,old_life_ladder,old_social_support,old_healthy
_life_expectancy_at_birth,

old_freedom_to_make_life_choices,old_perceptions_of_corruption,old_confidence_in_natio
nal_government,old_democratic_quality
    from country_dim
    where country_name = old_name;
```

```
    if old_name is null then
        signal sqlstate '45000' set message_text = 'Check country name. This country
is not in database.';
    else
        insert into country_dim (country_name, region, area, literacy, life_ladder,
social_support,
                                 healthy_life_expectancy_at_birth,
freedom_to_make_life_choices,
                                 perceptions_of_corruption,
confidence_in_national_government, democratic_quality,
                                 source_id, start_date)
            value (new_name, old_region, old_area, old_literacy, old_life_ladder,
old_social_support,
                   old_healthy_life_expectancy_at_birth,
                   old_freedom_to_make_life_choices, old_perceptions_of_corruption,
                   old_confidence_in_national_government,
                   old_democratic_quality, old_id, CURRENT_DATE);

        update country_dim
        set end_date = CURRENT_DATE
        where id = old_id;

    end if;
end //
delimiter ;

call slow_change_country('Albania', 'CHECK');
call slow_change_country('CHECK', 'Albania');
```

MySQL скрипт для завантаження нових даних до існуючих, incremental load:

```
use stage;

INSERT INTO warehouse.year_dim(`year`)
SELECT DISTINCT deaths.year
FROM deaths
JOIN suicides ON deaths.year = suicides.year
WHERE NOT EXISTS (
    SELECT 1 FROM warehouse.year_dim y
    WHERE y.year = deaths.year
);


INSERT INTO warehouse.country_dim(country_name, region, area, literacy, life_ladder,
social_support,
                                  healthy_life_expectancy_at_birth,
freedom_to_make_life_choices,
                                  perceptions_of_corruption,
confidence_in_national_government, democratic_quality)
SELECT DISTINCT deaths.country,
                countries.Region,
                countries.Area,
                countries.Literacy,
                happiness.life_ladder,
                happiness.social_support,
                happiness.healthy_life_expectancy_at_birth,
                happiness.freedom_to_make_life_choices,
                happiness.perceptions_of_corruption,
                happiness.confidence_in_national_government,
                happiness.democratic_quality
FROM deaths
JOIN suicides ON suicides.country = deaths.country
JOIN happiness ON happiness.country = deaths.country
JOIN countries ON countries.country = deaths.country
WHERE NOT EXISTS (
    SELECT 1
```

```
    FROM warehouse.country_dim c
    WHERE c.country_name = deaths.country
);

INSERT INTO warehouse.suicide_fact(id_year, id_country, population, gdp_per_capita,
death_count,
                                   suicide_count, conflict_and_terrorism,
road_injuries,
                                   alcohol_use_disorders, drug_use_disorders)
SELECT warehouse.year_dim.id,
       warehouse.country_dim.id,
       suicides.population,
       suicides.gdp_per_capita,
       deaths.total,
       suicides.suicides_no,
       deaths.conflict_and_terrorism,
       deaths.road_injuries,
       deaths.alcohol_use_disorders,
       deaths.drug_use_disorders
FROM suicides
JOIN warehouse.country_dim ON suicides.country = warehouse.country_dim.country_name
JOIN warehouse.year_dim ON suicides.year = warehouse.year_dim.year
JOIN deaths ON deaths.country = suicides.country AND deaths.year = suicides.year
WHERE NOT EXISTS (
    SELECT 1
    FROM warehouse.suicide_fact s
    WHERE s.id_year = warehouse.year_dim.id
        AND s.id_country = warehouse.country_dim.id
);
```

Висновок

Ознайомився з можливістю проектування сховищ даних, проходячи етапи створення stage зони для завантаження даних та створення основного сховища для розподілення даних зі зв’язками між ними. Було реалізовано можливості slowly changing dimension та incremental load. Перед завантаження у stage дані були оброблені python скриптами за допомогою бібліотеки pandas.