*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 лабораторної роботи № 1 з дисципліни

«Аналіз даних в інформаційних системах»

«СТВОРЕННЯ СХОВИЩА ДАНИХ»

Варіант 15

Виконав студент ІП-13, Лисенко Анастасія Олегівна

(шифр, прізвище, ім'я, по батькові)

Перевірив Олійник Юрій Олександрович

( прізвище, ім'я, по батькові)

Київ 202 3

Лабораторна робота 1

**СТВОРЕННЯ СХОВИЩА ДАНИХ**

Мета – ознайомитись з підходами до створення сховищ даних.

**Варіант 15**

Вибраний набір даних складається з трьох таблиць, пов’язаних із футболом: сезони, команди та матчі.

Таблиця «Сезони» містить інформацію про різні сезони, зокрема ідентифікатор сезону, рік, рівень, дивізіон, підрозділ, переможця та кількість команд.

Таблиця «Команди» містить інформацію про різні команди, включно з ідентифікатором команди, назвою команди, колишніми назвами команд, а також відомостями про поточну, колишню чи неіснуючу команду.

Таблиця «Матчі» містить інформацію про кожен матч, включаючи ідентифікатор матчу, ідентифікатор і назву господарської команди, ідентифікатор і назву команди-гостя, рахунок, рахунок домашньої команди, рахунок команди-гостя та результат.

Посилання на датасет, з якого я використала 3 зазначених файли: <https://www.kaggle.com/datasets/sujaykapadnis/english-football>

Stage zone

Моя stage зона має наступний вигляд:

Graphical user interface

Description automatically generated

Для створення stage зони я використовувала наступні sql scripts:

DROP DATABASE IF EXISTS football\_stage;  
CREATE DATABASE football\_stage;

use football\_stage;  
DROP TABLE IF EXISTS Seasons;  
DROP TABLE IF EXISTS Standings;  
DROP TABLE IF EXISTS Teams;  
DROP TABLE IF EXISTS Matches;  
DROP TABLE IF EXISTS Appearences;  
  
CREATE TABLE Seasons  
(  
 key\_id INT PRIMARY KEY,  
 season\_id VARCHAR(10),  
 season INT,  
 tier INT,  
 division VARCHAR(50),  
 subdivision VARCHAR(50),  
 winner VARCHAR(50),  
 count\_teams INT  
);  
  
CREATE TABLE Teams  
(  
 key\_id INT PRIMARY KEY,  
 team\_id VARCHAR(10),  
 team\_name VARCHAR(50),  
 former\_team\_names VARCHAR(500),  
 current tinyint(1),  
 former tinyint(1),  
 defunct tinyint(1),  
 first\_appearance INT  
);  
  
CREATE TABLE Matches  
(  
 key\_id INT PRIMARY KEY,  
 season\_id VARCHAR(10),  
 season INT,  
 tier INT,  
 division VARCHAR(50),  
 subdivision VARCHAR(50),  
 match\_id VARCHAR(20),  
 match\_name VARCHAR(100),  
 home\_team\_id VARCHAR(10),  
 home\_team\_name VARCHAR(50),  
 away\_team\_id VARCHAR(10),  
 away\_team\_name VARCHAR(50),  
 score VARCHAR(10),  
 home\_team\_score INT,  
 away\_team\_score INT,  
 home\_team\_score\_margin INT,  
 away\_team\_score\_margin INT,  
 result VARCHAR(20),  
 home\_team\_win tinyint(1),  
 away\_team\_win tinyint(1),  
 draw tinyint(1)  
);

Завантаження даних(в моєму випадку чистих) мало такий вигляд:

use football\_stage;  
  
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/matches.csv' INTO TABLE matches  
 FIELDS TERMINATED BY ','  
 LINES TERMINATED BY '\n'  
 IGNORE 1 ROWS  
 (key\_id, season\_id, season, tier, division, subdivision, match\_id, match\_name, home\_team\_id, home\_team\_name,  
 away\_team\_id, away\_team\_name, score, home\_team\_score, away\_team\_score, home\_team\_score\_margin,  
 away\_team\_score\_margin, result, home\_team\_win, away\_team\_win, draw);  
  
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Seasons.csv' INTO TABLE seasons  
 FIELDS TERMINATED BY ','  
 LINES TERMINATED BY '\n'  
 IGNORE 1 ROWS  
 (key\_id, season\_id, season, tier, division, subdivision, winner, count\_teams);  
  
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/teams.csv' INTO TABLE teams  
 FIELDS TERMINATED BY ','  
 LINES TERMINATED BY '\n'  
 IGNORE 1 ROWS  
 (key\_id, team\_id, team\_name, former\_team\_names, current, former, defunct, first\_appearance);

Warehouse

Мій warehouse має наступний вигляд:

A screenshot of a computer

Description automatically generated with medium confidence

Для його створення я використовувала наступні sql scripts:

DROP DATABASE IF EXISTS football\_wr;  
CREATE DATABASE football\_wr;

use football\_wr;  
DROP TABLE IF EXISTS DimYear;  
DROP TABLE IF EXISTS DimTeams;  
DROP TABLE IF EXISTS DimSeasons;  
DROP TABLE IF EXISTS DimResult;  
DROP TABLE IF EXISTS DimTierDivision;  
DROP TABLE IF EXISTS FactMatches;  
  
CREATE TABLE DimYear  
(  
 year\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 year INT  
);  
  
  
CREATE TABLE DimTeams  
(  
 team\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 team\_code VARCHAR(10),  
 team\_name VARCHAR(50),  
 former\_team\_names VARCHAR(500),  
 current tinyint(1),  
 former tinyint(1),  
 defunct tinyint(1),  
 Load\_Date date not null DEFAULT '1000-01-01',  
 End\_Load\_Date date not null DEFAULT '3000-01-01',  
 previous INT DEFAULT NULL,  
 first\_appearance INT  
);  
  
  
CREATE TABLE DimSeasons  
(  
 season\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 winner\_id INT,  
 season\_code VARCHAR(20),  
 season\_year\_id INT,  
 winner\_name VARCHAR(50),  
 count\_teams INT,  
 FOREIGN KEY (season\_year\_id) REFERENCES DimYear (year\_id),  
 FOREIGN KEY (winner\_id) REFERENCES DimTeams (team\_id)  
);  
  
CREATE TABLE DimResult  
(  
 result\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 result\_name VARCHAR(20)  
);  
  
CREATE TABLE DimTierDivision  
(  
 tier\_div\_sub\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 tier INT,  
 division VARCHAR(50),  
 subdivision VARCHAR(50)  
);  
  
CREATE TABLE FactMatches  
(  
 match\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 season\_id INT,  
 match\_name VARCHAR(100),  
 team\_home\_id int,  
 team\_away\_id int,  
 result\_id INT,  
 tier\_div\_sub\_id INT,  
 home\_team\_score INT,  
 away\_team\_score INT,  
 home\_team\_score\_margin INT,  
 away\_team\_score\_margin INT,  
 FOREIGN KEY (season\_id) REFERENCES DimSeasons (season\_id),  
 FOREIGN KEY (team\_home\_id) REFERENCES DimTeams (team\_id),  
 FOREIGN KEY (team\_away\_id) REFERENCES DimTeams (team\_id),  
 FOREIGN KEY (result\_id) REFERENCES DimResult (result\_id),  
 FOREIGN KEY (tier\_div\_sub\_id) REFERENCES DimTierDivision (tier\_div\_sub\_id)  
);

Завантаження даних мало такий вигляд:

use football\_stage;  
  
INSERT INTO football\_wr.DimYear(year)  
SELECT DISTINCT season  
FROM football\_stage.Seasons s  
WHERE s.season IS NOT NULL  
 AND NOT *EXISTS*(  
 SELECT year\_id  
 FROM football\_wr.DimYear dy  
 WHERE dy.year = s.season);  
  
INSERT INTO football\_wr.DimTeams(team\_code, team\_name, former\_team\_names, current, former, defunct, first\_appearance)  
SELECT DISTINCT  
 team\_id,  
 team\_name,  
 former\_team\_names,  
 current,  
 former,  
 defunct,  
 first\_appearance  
FROM football\_stage.Teams;  
  
INSERT INTO football\_wr.DimSeasons(winner\_id, season\_code, season\_year\_id, winner\_name, count\_teams)  
SELECT DISTINCT dt.team\_id, fss.season\_id, dy.year\_id, fss.winner, fss.count\_teams  
FROM football\_stage.Seasons fss  
 JOIN football\_wr.DimTeams dt ON dt.team\_name = fss.winner  
 JOIN football\_wr.DimYear dy ON dy.year = fss.season;  
  
INSERT INTO football\_wr.DimResult (result\_id, result\_name)  
VALUES (1, 'home team win'),  
 (2, 'away team win'),  
 (3, 'draw');  
  
INSERT INTO football\_wr.DimTierDivision(tier, division, subdivision)  
SELECT DISTINCT tier, division, subdivision  
FROM football\_stage.Matches m  
WHERE m.tier IS NOT NULL  
 AND NOT *EXISTS*(  
 SELECT tier, division, subdivision  
 FROM football\_wr.DimTierDivision td  
 WHERE td.tier = m.tier  
 AND td.division = m.division  
 AND td.subdivision = m.subdivision);  
  
  
INSERT INTO football\_wr.FactMatches (season\_id, match\_name, team\_home\_id, team\_away\_id, home\_team\_score,  
 away\_team\_score, home\_team\_score\_margin, away\_team\_score\_margin, result\_id,  
 tier\_div\_sub\_id)  
SELECT s.season\_id,  
 m.match\_name,  
 th.team\_id,  
 ta.team\_id,  
 m.home\_team\_score,  
 m.away\_team\_score,  
 m.home\_team\_score\_margin,  
 m.away\_team\_score\_margin,  
 r.result\_id,  
 td.tier\_div\_sub\_id  
FROM football\_stage.Matches m  
 JOIN football\_wr.DimSeasons s ON m.season\_id = s.season\_code  
 JOIN football\_wr.DimTeams ta ON m.away\_team\_id = ta.team\_code  
 JOIN football\_wr.DimTeams th ON m.home\_team\_id = th.team\_code  
 JOIN football\_wr.DimResult r ON m.result = r.result\_name  
 JOIN football\_wr.DimTierDivision td  
 ON m.tier = td.tier and m.division = td.division and m.subdivision = td.subdivision;

Виконання scd має наступний вигляд:

DROP PROCEDURE IF EXISTS football\_wr.*scd\_teams*;  
  
DELIMITER $$  
CREATE PROCEDURE football\_wr.*scd\_teams*(old\_name varchar(70), new\_name varchar(70))  
BEGIN  
 DECLARE old\_id int DEFAULT null;  
 DECLARE old\_team\_code VARCHAR(10);  
 DECLARE old\_former\_team\_names VARCHAR(500);  
 DECLARE old\_current TINYINT(1);  
 DECLARE old\_former TINYINT(1);  
 DECLARE old\_defunct TINYINT(1);  
 DECLARE old\_first\_appearance INT;  
 DECLARE new\_id int;  
  
 SELECT team\_id, team\_code, former\_team\_names, current, former, defunct, first\_appearance  
 into old\_id, old\_team\_code, old\_former\_team\_names, old\_current, old\_former, old\_defunct, old\_first\_appearance  
 FROM dimteams  
 WHERE team\_name = old\_name;  
 IF old\_id IS NULL THEN  
 SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Old team name has not been found!';  
 ELSE  
 INSERT INTO dimteams (team\_code, team\_name, former\_team\_names, current, former, defunct, Load\_Date, previous,  
 first\_appearance)  
 VALUE (old\_team\_code, new\_name,  
 *IF*(old\_former\_team\_names = "None", old\_name, *CONCAT*(old\_former\_team\_names, "|", old\_name)),  
 old\_current, old\_former, old\_defunct, *CURRENT\_DATE*, old\_id, old\_first\_appearance);  
 UPDATE dimteams  
 SET End\_Load\_Date = *CURRENT\_DATE* WHERE team\_id = old\_id;  
  
 SELECT team\_id  
 into new\_id  
 FROM dimteams  
 WHERE team\_name LIKE new\_name;  
 end if;  
end $$  
  
DELIMITER ;

Виконання incremental load має наступний вигляд:

use football\_stage;  
  
INSERT INTO football\_wr.DimYear(year)  
SELECT DISTINCT season  
FROM football\_stage.Seasons s  
WHERE s.season IS NOT NULL  
 AND NOT *EXISTS*(  
 SELECT year\_id  
 FROM football\_wr.DimYear dy  
 WHERE dy.year = s.season);  
  
INSERT INTO football\_wr.DimTeams(team\_code, team\_name, former\_team\_names, current, former, defunct, first\_appearance)  
SELECT DISTINCT team\_id,  
 team\_name,  
 former\_team\_names,  
 current,  
 former,  
 defunct,  
 first\_appearance  
FROM football\_stage.Teams t  
WHERE NOT *EXISTS*(  
 SELECT 1  
 FROM football\_wr.DimTeams dt  
 WHERE dt.team\_code = t.team\_id  
 and dt.team\_name = t.team\_name  
 and dt.former\_team\_names = t.former\_team\_names  
 and dt.current = t.current  
 and dt.former = t.former  
 and dt.defunct = t.defunct  
 and dt.first\_appearance = t.first\_appearance);  
  
INSERT INTO football\_wr.DimSeasons(winner\_id, season\_code, season\_year\_id, winner\_name, count\_teams)  
SELECT DISTINCT dt.team\_id, fss.season\_id, dy.year\_id, fss.winner, fss.count\_teams  
FROM football\_stage.Seasons fss  
 JOIN football\_wr.DimTeams dt ON dt.team\_name = fss.winner  
 JOIN football\_wr.DimYear dy ON dy.year = fss.season  
WHERE NOT *EXISTS*(  
 SELECT 1  
 FROM football\_wr.DimSeasons ds  
 WHERE ds.winner\_id = (select team\_id from football\_wr.DimTeams dt where dt.team\_name = fss.winner)  
 and ds.season\_code = fss.season\_id  
 and ds.season\_year\_id = (select dy.year\_id from football\_wr.DimYear dy where dy.year = fss.season)  
 and ds.winner\_name = fss.winner  
 and ds.count\_teams = fss.count\_teams);  
  
  
INSERT INTO football\_wr.FactMatches (season\_id, match\_name, team\_home\_id, team\_away\_id, home\_team\_score,  
 away\_team\_score, home\_team\_score\_margin, away\_team\_score\_margin, result\_id,  
 tier\_div\_sub\_id)  
SELECT DISTINCT s.season\_id,  
 m.match\_name,  
 th.team\_id,  
 ta.team\_id,  
 m.home\_team\_score,  
 m.away\_team\_score,  
 m.home\_team\_score\_margin,  
 m.away\_team\_score\_margin,  
 r.result\_id,  
 td.tier\_div\_sub\_id  
FROM football\_stage.Matches m  
 JOIN football\_wr.DimSeasons s ON m.season\_id = s.season\_code  
 JOIN football\_wr.DimTeams ta ON m.away\_team\_id = ta.team\_code  
 JOIN football\_wr.DimTeams th ON m.home\_team\_id = th.team\_code  
 JOIN football\_wr.DimResult r ON m.result = r.result\_name  
 JOIN football\_wr.DimTierDivision td  
 ON m.tier = td.tier and m.division = td.division and m.subdivision = td.subdivision  
WHERE NOT *EXISTS*(  
 SELECT 1  
 FROM football\_wr.FactMatches fm  
 WHERE fm.season\_id = (select season\_id from football\_wr.DimSeasons s where m.season\_id = s.season\_code)  
 and fm.match\_name = m.match\_name  
 and fm.team\_home\_id = (select ta.team\_id from football\_wr.DimTeams ta where m.away\_team\_id = ta.team\_code)  
 and fm.team\_away\_id = (select th.team\_id from football\_wr.DimTeams th where m.home\_team\_id = th.team\_code)  
 and fm.home\_team\_score = m.home\_team\_score  
 and fm.away\_team\_score = m.away\_team\_score  
 and fm.home\_team\_score\_margin = m.home\_team\_score\_margin  
 and fm.away\_team\_score\_margin = m.away\_team\_score\_margin  
 and fm.result\_id = (select result\_id from football\_wr.DimResult r where m.result = r.result\_name)  
 and fm.tier\_div\_sub\_id = (select tier\_div\_sub\_id  
 from football\_wr.DimTierDivision td  
 where (m.tier = td.tier and m.division = td.division and  
 m.subdivision = td.subdivision)));

Всі надані вище scripts(окрім scd та incremental load) можуть бути виконані за допомогою коду на мові Python, який я прикріплю до завдання.

1. **Висновки**

В цій лабораторній роботі ми ознайомилися з підходами до створення сховищ даних. Та попрацювали з ними на практиці. Створили сховище даних англійського футболу.