7

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

Практикум №1

з курсу «Аналіз даних в інформаційнних системах»

на тему: «Створення процедур завантаження даних»

|  |  |
| --- | --- |
| Викладач:  Олійник Ю.О. | Виконав:  студент 2 курсу  групи ІП-11 ФІОТ  Рябов Ю. І. |

Київ-2023

**1. ПОСТАНОВКА ЗАДАЧІ**

**1.1 Опис предметної області**

Для виконання роботи у якості предметної області було результати змагань з швидкубінгу(складання кубику Рубіка та інший головоломок на швидкість) з ресурсу всесвітньої асоціації швидкубінгу <https://www.worldcubeassociation.org/results/misc/export.html>

Тему роботи було обрано згідно моїх власних інтересів. В датасеті присутні результати більш ніж 180 тисяч людей по всьому світу, в тому числі [мої](https://www.worldcubeassociation.org/persons/2018RIAB01).

Датасет містить результати з 21 дисципліни, змагання з 17 з яких проходять до нині.

**1.2 Опис джерел даних**

В якості джерела даних виступив офіційний .sql файл для створення структури бази даних та заповнення таблиць даними.

Нижче наведені поля для кожної з таблиць, які безпосередньо використовувались у подальшій побудові бізнес-процесів:

8

Таблиця 1.2.1 - поля вхідних таблиць

|  |  |  |
| --- | --- | --- |
| Results | competitionId | Ідентифікатор змагань, де був поставлений результат |
| eventId | Ідентифікатор дисципліни |
| roundTypeId | Ідентифікатор типу раунду |
| Pos, best, average | Зайняте учасником місце, кращий та середній результат |
| personName, personId, personCountryId | Ім’я, айді та країна учасника |
| formatId | Формат результату |
| Value1, value2, value3, value4, value5 | Результати спроб учасника |
| regionalSingleRecord, regionalAverageRecord | Ідентифікатори рекордів, якщо вони встановлені(одинична спроба і середній результат) |
| RoundTypes | id | Ідентифікатор типу раунду |
| name | Назва типу раунду |
| rank | Ранг раунду |
| final | Флаг що вказує чи є раунд фіналом |
| Events | id | Ідентифікатор дисципліни |
| name | Назва дисципліни |
| format | Формат дисципліни |
| Formats | id | Айді формату |
| Name | Назва формату |
| Sort\_by | Параметр сортування |
| Sort\_by\_second | Другий параметр сортування |
| Expected\_solve\_count | Очікувана кількість збірок в результаті дисципліни |
| Trim fastest n | Кількість швидших збірок що відкидаються |
| Trim slowest n | Кількість гірших збірок що відкидаються |
| Countries | Id | Айді країни |
| Name | Назва країни |
| Continent id | Айді континенту |
| Iso2 | Ісо2 країни |
| Continents | Id | Айді континенту |
| Name | Назва континенту |
| Record name | Назва континентального рекорду |
| Latitude, longitude | Координати континенту |
| Zoom | Зум континенту |
| Persons | id | Айді учасника |
| Sub id | Вторичне айді (коли змінюється ім’я чи країна створюється новий запис) |
| name | Ім’я |
| Country id | Айді країни |
| Gender | Гендер учасника |
| Competitions | Id | Айді змагань |
| Name | Назва змагань |
| City name, country id | Місто та країна змагань |
| Information | Інформація про змагання |
| Year, month, day | Рік, місяць та день початку змагань |
| End month, end day | Місяць та день кінця змагань |
| Cancelled | Чи змагання відмінені |
| Event specs | Набір дисциплін на змаганнях |
| Wca delegate, organizer | Делегат та організатор змагань |
| Venue, venue address, venue details | Назва, адреса да деталі місця проведення |
| External website | Зовнішній вебсайт змагань |
| Latitude, longitude | Координати змагань |
| Scrambles | Scramble id | Айді скрамбла (скрабл – алгоритм заплутування головоломки) |
| Competition id | Айді змагань |
| Event id | Айді дисципліни |
| Round type id | Айді типу раунду |
| Group id | Айді групи |
| Is extra | Чи є скрамбл екстрою (додатковою спробою) |
| Scramble num | Номер скрамблу |
| Scramble | Текст скрамблу |
| Championships | Id | Айді чемпіонату |
| Competition id | Айді змагань |
| Championship type | Тип чемпіонату |

9

**2. РОЗРОБКА МОДЕЛІ СХОВИЩА ДАНИХ**

У процесі розробки моделі сховища даних було декомпозовано існуючі таблиці та виділені наступні:

results\_fact – фактова таблиця результатів

championships\_dim – вимір чемпіонатів

championshipTypes\_dim – вимір типів чемпіонатів

scrambles\_dim – вимір скрамблів

groups\_dim – вимір груп учасників раунду

competitions\_dim – вимір змагань

venues\_dim – вимір місця проведення змагань

eventSpecs\_dim – вимір набору дисциплін на змаганнях

organisers\_dim – вимір організаторів змагань

delegates\_dim – вимір делегатів

cities\_dim – вимір міст

dates\_dim – вимір дат

persons\_dim – вимір учасників змагань

genders\_dim – вимір гендерів

countries\_dim – вимір країн

continents\_dim – вимір континентів

coordinates\_dim – вимір координат

recordTypes\_dim – вимір типів рекордів

resultFormats\_dim – вимір форматів результатів

events\_dim – вимір дисциплін

eventFormats\_dim – вимір форматів дисциплін

roundTypes\_dim – вимір типів раундів

11

В результаті була спроектована схема сховища даних, яка зображена на рисунках 2.1-2.2. Дана модель дозволяє описувати відповідні бізнес-процеси згідно предметній області. Для представлення даних була вибрана багатовимірна модель зі схемою “сніжинка”.

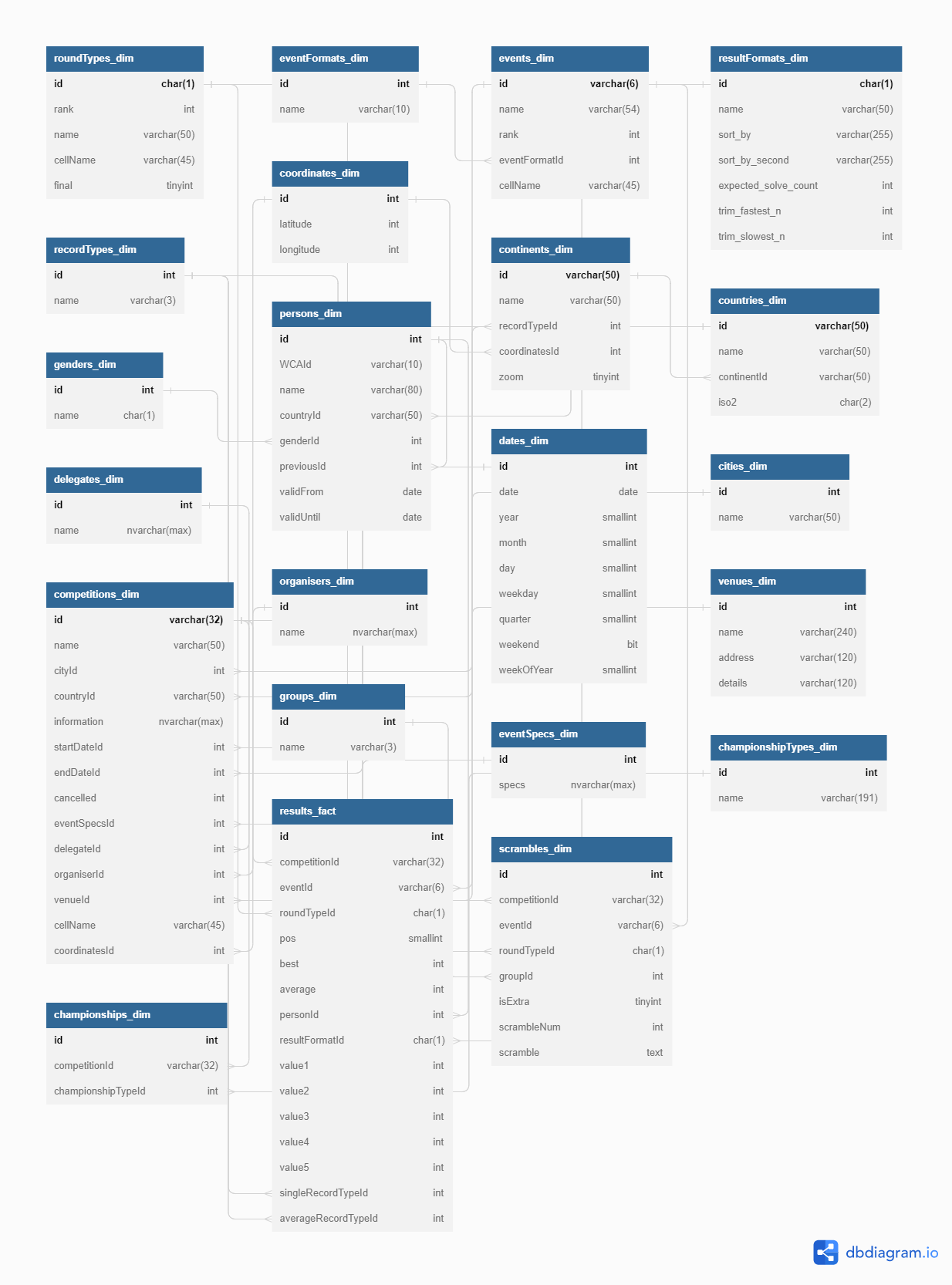
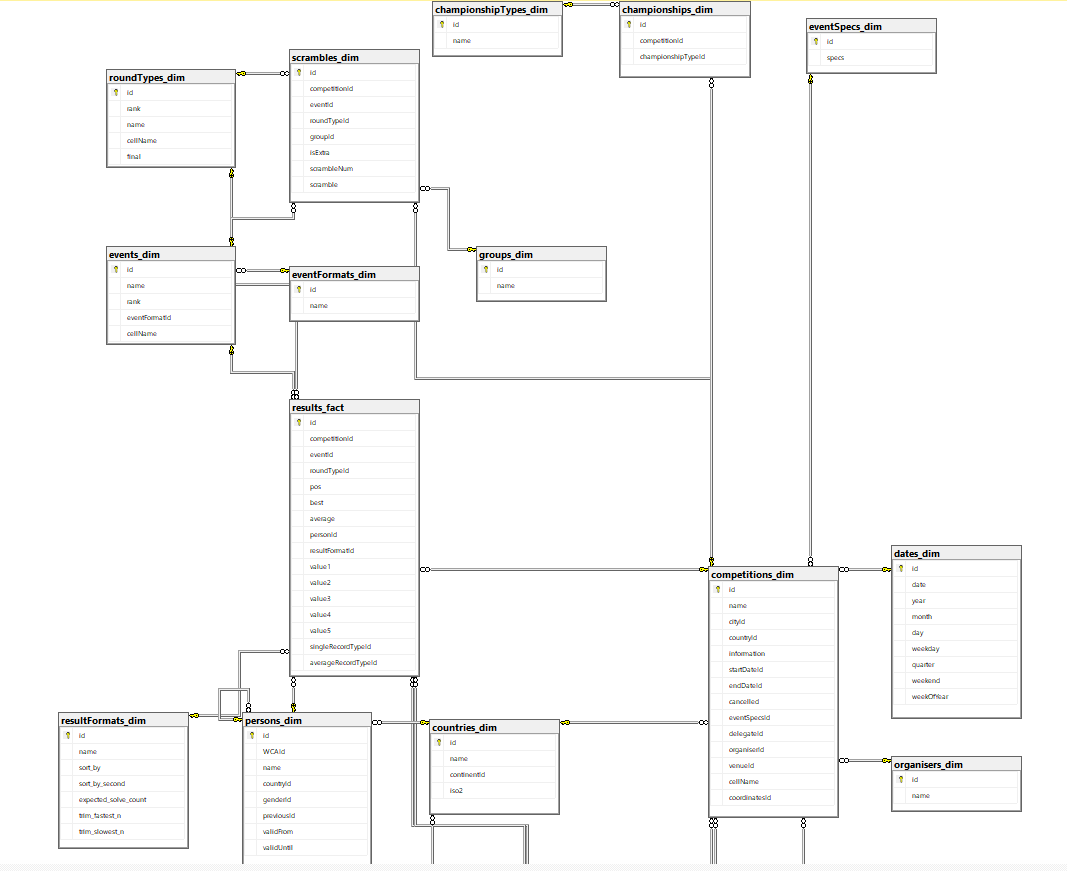
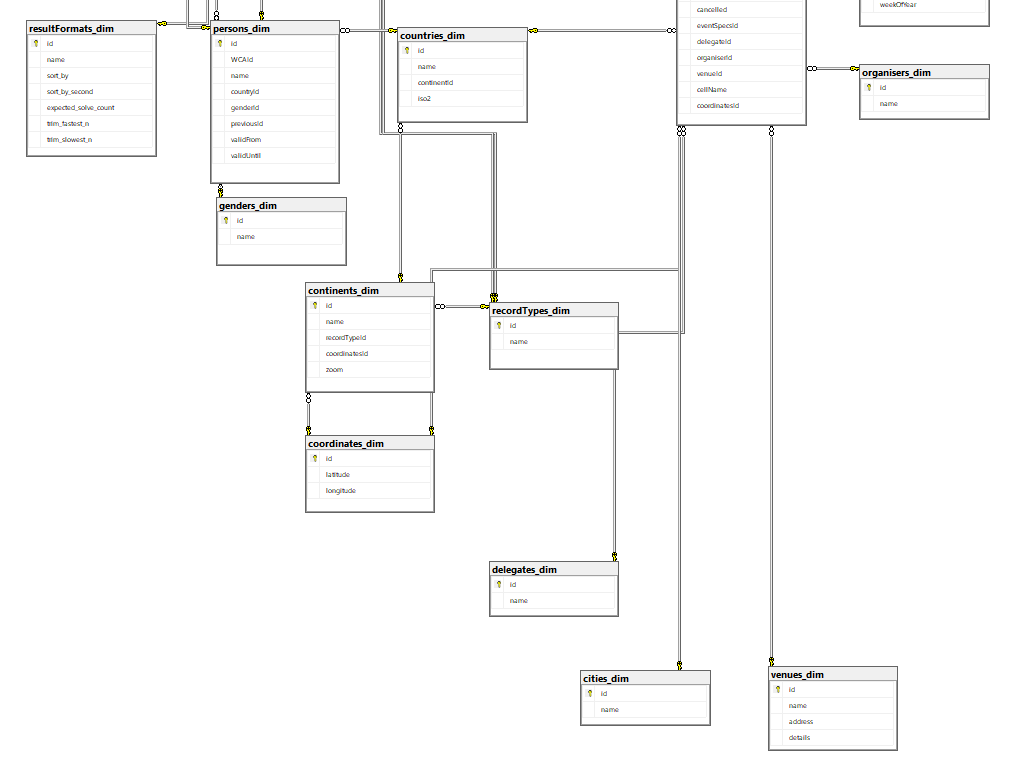


Рисунок 2.1.1 - діаграма таблиць сховища даних з типами





Рисунки 2.1.2-2.1.3 - діаграма бази даних

**3. РОЗРОБКА ETL ПРОЦЕСІВ**

**3.1 Методи вирішення проблеми**

Існує декілька підходів до розробки ETL процесів, а саме:

- SQL скрипти

- скрипти/програми для обробки даних на різноманітних мовах програмування

- програмні застосунки з UI

У процесі виконання роботи для реалізації ETL процесів було обрано SQL скрипти, оскільки вихідні дані подані саме в такому вигляді

**3.2 Аналіз вхідних даних**

Всі необхідні вхідні джерела представляються файлами у форматі sql.

Після аналізу вхідних джерел був зроблений висновок, що інформації достатньо для заповнення усіх таблиць сховища даних.

13

**3.3 Підготовка таблиць в stage-зоні та сховищі даних**

В результаті розробки була спроектована схема stage-зони, яка зображена на рисунку 3.3.1. Дана модель відображає таблиці для даних із вхідних джерел.



Рисунок 3.3.1 – схема stage-зони

**3.4 Екстракт даних**

Вихідний файл було розбито на 34 файли по 200 тисяч строк та відформатовано для роботи з ms sql server. Після цього дані було завантажено до stage-зони

18

**3.5 Завантаження даних**

Останнім кроком необхідно виконати завантаження даних із stage зони у сховище даних. Для цього у проекті було створено скрипти для створення талиць та завантаження в них даних з стейдж зони:

drop table if exists results\_fact

drop table if exists championships\_dim

drop table if exists championshipTypes\_dim

drop table if exists scrambles\_dim

drop table if exists groups\_dim

drop table if exists competitions\_dim

drop table if exists venues\_dim

drop table if exists eventSpecs\_dim

drop table if exists organisers\_dim

drop table if exists delegates\_dim

drop table if exists cities\_dim

drop table if exists dates\_dim

drop table if exists persons\_dim

drop table if exists genders\_dim

drop table if exists countries\_dim

drop table if exists continents\_dim

drop table if exists coordinates\_dim

drop table if exists recordTypes\_dim

drop table if exists resultFormats\_dim

drop table if exists events\_dim

drop table if exists eventFormats\_dim

drop table if exists roundTypes\_dim

CREATE TABLE roundTypes\_dim (

id char(1) primary key,

rank int NOT NULL DEFAULT '0',

name varchar(50) NOT NULL DEFAULT '',

cellName varchar(45) NOT NULL DEFAULT '',

final tinyint NOT NULL

)

create table eventFormats\_dim (

id int primary key identity(1, 1),

name varchar(10)

)

CREATE TABLE events\_dim (

id varchar(6) primary key,

name varchar(54) NOT NULL DEFAULT '',

rank int NOT NULL DEFAULT '0',

eventFormatId int,

cellName varchar(45) NOT NULL DEFAULT '',

foreign key (eventFormatId) references eventFormats\_dim(id)

)

CREATE TABLE resultFormats\_dim (

id char(1) primary key,

name varchar(50) NOT NULL DEFAULT '',

sort\_by varchar(255) NOT NULL,

sort\_by\_second varchar(255) NOT NULL,

expected\_solve\_count int NOT NULL,

trim\_fastest\_n int NOT NULL,

trim\_slowest\_n int NOT NULL

)

create table recordTypes\_dim (

id int primary key identity(1, 1),

name varchar(3) not null

)

create table coordinates\_dim (

id int primary key identity(1, 1),

latitude int NOT NULL DEFAULT '0',

longitude int NOT NULL DEFAULT '0'

)

CREATE TABLE continents\_dim (

id varchar(50) primary key,

name varchar(50) NOT NULL DEFAULT '',

recordTypeId int default null,

coordinatesId int,

zoom tinyint NOT NULL DEFAULT '0',

foreign key (recordTypeId) references recordTypes\_dim(id),

foreign key (coordinatesId) references coordinates\_dim(id)

)

CREATE TABLE countries\_dim (

id varchar(50) primary key,

name varchar(50) NOT NULL DEFAULT '',

continentId varchar(50) NOT NULL DEFAULT '',

iso2 char(2) DEFAULT NULL,

foreign key (continentId) references continents\_dim(id)

)

CREATE TABLE genders\_dim (

id int primary key identity(1, 1),

name char(1)

)

CREATE TABLE persons\_dim (

id int primary key identity(1, 1),

WCAId varchar(10) NOT NULL DEFAULT '',

name varchar(80) DEFAULT NULL,

countryId varchar(50) NOT NULL DEFAULT '',

genderId int,

previousId int null,

validFrom date null,

validUntil date null,

foreign key (countryId) references countries\_dim(id),

foreign key (genderId) references genders\_dim(id),

foreign key (previousId) references persons\_dim(id)

)

create table dates\_dim (

id int primary key identity(1, 1),

[date] date,

year smallint NOT NULL DEFAULT '0',

month smallint NOT NULL DEFAULT '0',

day smallint NOT NULL DEFAULT '0',

weekday smallint,

quarter smallint,

weekend bit,

weekOfYear smallint

)

create table cities\_dim (

id int primary key identity(1, 1),

name varchar(50)

)

create table delegates\_dim (

id int primary key identity(1, 1),

name nvarchar(max)

)

create table organisers\_dim (

id int primary key identity(1, 1),

name nvarchar(max)

)

create table eventSpecs\_dim (

id int primary key identity(1, 1),

specs nvarchar(max)

)

create table venues\_dim (

id int primary key identity(1, 1),

name varchar(240),

address varchar(120),

details varchar(120)

)

CREATE TABLE competitions\_dim (

id varchar(32) primary key,

name varchar(50) NOT NULL DEFAULT '',

cityId int,

countryId varchar(50) NOT NULL DEFAULT '',

information nvarchar(max) ,

startDateId int,

endDateId int,

cancelled int NOT NULL DEFAULT '0',

eventSpecsId int,

delegateId int,

organiserId int,

venueId int,

cellName varchar(45) NOT NULL DEFAULT '',

coordinatesId int,

foreign key (startDateId) references dates\_dim(id),

foreign key (endDateId) references dates\_dim(id),

foreign key (cityId) references cities\_dim(id),

foreign key (countryId) references countries\_dim(id),

foreign key (delegateId) references delegates\_dim(id),

foreign key (eventSpecsId) references eventSpecs\_dim(id),

foreign key (organiserId) references organisers\_dim(id),

foreign key (coordinatesId) references coordinates\_dim(id),

foreign key (venueId) references venues\_dim(id)

)

create table groups\_dim (

id int primary key identity(1, 1),

name varchar(3)

)

CREATE TABLE scrambles\_dim (

id int primary key identity(1, 1),

competitionId varchar(32) NOT NULL,

eventId varchar(6) NOT NULL,

roundTypeId char(1) NOT NULL,

groupId int,

isExtra tinyint NOT NULL,

scrambleNum int NOT NULL,

scramble text NOT NULL,

foreign key (competitionId) references competitions\_dim(id),

foreign key (eventId) references events\_dim(id),

foreign key (roundTypeId) references roundTypes\_dim(id),

foreign key (groupId) references groups\_dim(id)

)

create table championshipTypes\_dim (

id int primary key identity(1, 1),

name varchar(191) not null

)

CREATE TABLE championships\_dim (

id int primary key identity(1, 1),

competitionId varchar(32) NOT NULL,

championshipTypeId int,

foreign key (championshipTypeId) references championshipTypes\_dim(id),

foreign key (competitionId) references competitions\_dim(id)

)

CREATE TABLE results\_fact (

id int primary key identity(1, 1),

competitionId varchar(32) NOT NULL DEFAULT '',

eventId varchar(6) NOT NULL DEFAULT '',

roundTypeId char(1) NOT NULL DEFAULT '',

pos smallint NOT NULL DEFAULT '0',

best int NOT NULL DEFAULT '0',

average int NOT NULL DEFAULT '0',

personId int,

resultFormatId char(1) NOT NULL DEFAULT '',

value1 int NOT NULL DEFAULT '0',

value2 int NOT NULL DEFAULT '0',

value3 int NOT NULL DEFAULT '0',

value4 int NOT NULL DEFAULT '0',

value5 int NOT NULL DEFAULT '0',

singleRecordTypeId int null,

averageRecordTypeId int null,

foreign key (competitionId) references competitions\_dim(id),

foreign key (eventId) references events\_dim(id),

foreign key (roundTypeId) references roundTypes\_dim(id),

foreign key (personId) references persons\_dim(id),

foreign key (resultFormatId) references resultFormats\_dim(id),

foreign key (singleRecordTypeId) references recordTypes\_dim(id),

foreign key (averageRecordTypeId) references recordTypes\_dim(id),

)

insert into roundTypes\_dim select \* from RoundTypes

insert into eventFormats\_dim (name)

select distinct format from Events

insert into events\_dim (id, name, rank, eventFormatId, cellName)

select e.id, e.name, e.rank, f.id, e.cellName

from events as e

outer apply (

select top(1) id

from eventFormats\_dim as ef

where e.format = ef.name

) as f

insert into resultFormats\_dim select \* from Formats

insert into recordTypes\_dim (name)

select distinct regionalSingleRecord from Results where regionalSingleRecord != ''

insert into coordinates\_dim (latitude, longitude)

(

select latitude, longitude from Competitions

union

select latitude, longitude from Continents

)

insert into continents\_dim (id, name, recordTypeId, coordinatesId, zoom)

select con.id, con.name, r.id, cor.id, con.zoom

from Continents as con

outer apply (

select id

from recordTypes\_dim

where con.recordName = name

) as r

outer apply (

select id

from competitions\_dim

where con.latitude = latitude and con.longitude = longitude

) as cor

insert into countries\_dim select \* from Countries

insert into genders\_dim (name)

select distinct gender from Persons

insert into persons\_dim (WCAId, name, countryId, genderId)

select p.id, p.name, p.countryId, g.id

from Persons as p

outer apply (

select id

from genders\_dim

where p.gender = name

) as g

insert into dates\_dim(date, year, month, day, weekday, quarter, weekend, weekOfYear)

select CONVERT(date, date\_str),

DATEPART(year, date\_str),

DATEPART(month, date\_str),

DATEPART(dayOfYear, date\_str),

DATEPART(weekday, date\_str),

DATEPART(quarter, date\_str),

case when DATEPART(weekday, date\_str) IN (1,7) then 1 else 0 end,

DATEPART(wk, date\_str)

from (

select distinct CONCAT(year, '-', month, '-', day) AS date\_str

from (

select year, month, day from Competitions

union

select

(case when endMonth < month then year + 1 else year end) as year,

endMonth as month,

endDay as day

from Competitions

) as dates

) as subq

insert into cities\_dim (name)

select distinct cityName from Competitions

insert into delegates\_dim(name)

select distinct wcaDelegate from Competitions

insert into organisers\_dim(name)

select distinct organiser from Competitions

insert into eventSpecs\_dim (specs)

select distinct eventSpecs from Competitions

insert into venues\_dim (name, address, details)

select distinct venue, venueAddress, venueDetails from Competitions

insert into competitions\_dim (

id,

name,

cityId,

countryId,

information,

startDateId,

endDateId,

cancelled,

eventSpecsId,

delegateId,

organiserId,

venueId,

cellName,

coordinatesId

)

select

comp.id,

comp.name,

city.id,

comp.countryId,

comp.information,

startDate.id,

endDate.id,

comp.cancelled,

eventSpecs.id,

delegate.id,

organiser.id,

venue.id,

comp.cellName,

coordinates.id

from Competitions as comp

outer apply (

select id

from cities\_dim

where comp.cityName = name

) as city

outer apply (

select id

from dates\_dim

where year = comp.year and month = comp.month and day = comp.day

) as startDate

outer apply (

select id

from dates\_dim

where

year = (

case when comp.endMonth < comp.month

then comp.year + 1

else comp.year

end

) and

month = comp.month and

day = comp.day

) as endDate

outer apply (

select id

from eventSpecs\_dim

where comp.eventSpecs = specs

) as eventSpecs

outer apply (

select id

from delegates\_dim

where comp.wcaDelegate = name

) as delegate

outer apply (

select id

from organisers\_dim

where comp.organiser = name

) as organiser

outer apply (

select id

from venues\_dim

where comp.venue = name and comp.venueAddress = address and comp.venueDetails = details

) as venue

outer apply (

select id

from coordinates\_dim

where comp.latitude = latitude and comp.longitude = longitude

) as coordinates

insert into groups\_dim (name)

select distinct groupId from Scrambles

insert into scrambles\_dim (

competitionId,

eventId,

roundTypeId,

groupId,

isExtra,

scrambleNum,

scramble

)

select

s.competitionId,

s.eventId,

s.roundTypeId,

g.id,

s.isExtra,

s.scrambleNum,

s.scramble

from Scrambles as s

outer apply (

select top(1) id

from groups\_dim

where s.groupId = name

) as g

insert into championshipTypes\_dim (name)

select distinct championship\_type from championships

insert into championships\_dim (competitionId, championshipTypeId)

select c.competition\_id, ct.id

from championships as c

outer apply (

select top 1 id

from championshipTypes\_dim

where name = c.championship\_type

) as ct

insert into results\_fact (

competitionId,

eventId,

roundTypeId,

pos,

best,

average,

personId,

resultFormatId,

value1,

value2,

value3,

value4,

value5,

singleRecordTypeId,

averageRecordTypeId

)

select

r.competitionId,

r.eventId,

r.roundTypeId,

r.pos,

r.best,

r.average,

p.id,

r.formatId,

r.value1,

r.value2,

r.value3,

r.value4,

r.value5,

s.id,

a.id

from Results as r

outer apply (

select top(1) id

from persons\_dim

where WCAId = r.personId and name = r.personName and countryId = r.personCountryId

) as p

outer apply (

select id

from recordTypes\_dim

where name = r.regionalSingleRecord

) as s

outer apply (

select id

from recordTypes\_dim

where name = r.regionalAverageRecord

) as a

В результаті пророблених операцій дані з вхідних джерел були витягнуті, перетворені згідно із зазначеними правилами та завантажені у сховище даних для подальшого аналізу.

28

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Набір даних

<https://www.worldcubeassociation.org/results/misc/export.html>