CS306 Spring 23 Group Project Step 2

Group Name: Hasta La Vista Baby!

Group Members: Ahmet Emre Eser, Barkın Var, Beste Bayhan, Sadig Qara, Ecem Akın

Github Link: https://github.com/barkinvar/CS306-Hasta-La-Vista-Baby

Creation of the Database

Country and Continents Table

The continent table contains two different attributes which are continent_code and continent_name. We first created a continent entity with the create table command. The “continent_code” attribute is used as a primary key for the continent table which uniquely identifies each continent and we keep the continent name data in the continent_name attribute.

The country data was taken from pop data (taken from covid dataset).csv which can be found in our GitHub project. The country table is created by using the create table command. It has two attributes which are country_code and country_name. “country_code” is the primary key of the country entity that uniquely identifies each of them. There is a many to one relationship between country and continent because multiple countries can be located on a country but inverse of it is not possible. Each country has at most one location, according to the key constraint on the “Located” relationship table, so key constraint is used. Also, since each continent must have at least one country and each country must have at least one continent, participation constraint is used in both directions. If an attribute marked as NOT NULL is not provided when an instance is being inserted, the database will refuse the attempt.

```
CREATE TABLE country(

country_code VARCHAR(10) NOT NULL,
country_name VARCHAR(50) NOT NULL,

PRIMARY KEY (country_code)
)


CREATE TABLE continent(
```

```
continent_code VARCHAR(10) NOT NULL,
continent_name VARCHAR(50) NOT NULL,

PRIMARY KEY (continent_code)
)
```

———————

The “s_deaths″ table consists of a csv file named "share-deaths-smoking.csv" which is uploaded in our github project under the filtered data folder.The s_deaths table has three attributes: country_code, death_stat_year, and percentage_among_all_deaths. The country_code attribute is a foreign key that references the country table, and it is not null. The death_stat_year attribute is an integer that represents the year for which the death statistics are recorded, and it is also marked as not null. The percentage_among_all_deaths attribute is a real number that represents the percentage of deaths in a country among all deaths recorded in the world for the given year.

The primary key for the s_deaths table is a combination of country_code and death_stat_year, which ensures that each row in the table represents a unique combination of country and year. The foreign key constraint on the country_code attribute ensures that only valid country codes can be inserted into the table which purports that the s_death entity is a weak entity .

The ON DELETE CASCADE option is specified in the foreign key constraint, which means that if a row in the country table is deleted, all the corresponding rows in the s_deaths table will also be deleted automatically. This ensures referential integrity between the two tables.

———————

```
CREATE TABLE s_deaths(

country_code VARCHAR(10) NOT NULL,
death_stat_year INT NOT NULL,
percentage_among_all_deaths REAL,

PRIMARY KEY (country_code, death_stat_year),
FOREIGN KEY (country_code) REFERENCES
country(country_code) ON DELETE CASCADE
)
```

———————

Smoker Table:

We organized the smoker data into one csv file: prevalence-of-tobacco-use-sdgs.csv. The entity has a superkey pair containing the country_code and `percentage_among_all_adults` attributes. The country_code is a PRIMARY FOREIGN KEY that is obtained from a country object and has the property ON DELETE CASCADE, meaning the smoker element is set to be deleted when the country element is deleted, which makes it a weak entity.

```
percentage_among_all_adults REAL,
Smoker_stat_year bigint,
PRIMARY KEY (country_code, percentage_among_all_adults),
FOREIGN KEY (country_code) REFERENCES
country(country_code) ON DELETE CASCADE
)
```

The actual data insertion was done with the mySql Workbench csv import feature however the log file contains the commands processed to achieve such effect. The command below inserts a three element tuple into the population entity of the cs306 database (our database's name).

```
PREPARE stmt FROM 'INSERT INTO `CS306`.`smoker`
(`smoker_stat_year`,`percentage_among_all_adults`,`country_code`)
VALUES(?,?,?)

CREATE TABLE smoker(

country_code VARCHAR(10) NOT NULL,
```

Located Table:

The binary relation located links the country data to the continent data, with each country represented by a single continent, including countries like Russia and Turkey which do occupy parts of multiple continents, whereas a continent can contain multiple countries, therefore creating a one-to-many relation. The relation contains a PRIMARY FOREIGN KEY from a country object, country_code with the ON DELETE CASCADE statement, meaning the relation is to be removed upon the removal of the country object. Similarly we have a FOREIGN KEY from a continent object, continent_code, also with the ON DELETE CASCADE statement, meaning the relation will be removed when the continent object is removed.

```
CREATE TABLE located(

continent_code VARCHAR(5) NOT NULL,
country_code VARCHAR(5) NOT NULL,
```

```
PRIMARY KEY (country_code), # each country can at most one continent,
whereas a continent can have many countries

FOREIGN KEY (continent_code) REFERENCES
continent(continent_code) ON DELETE CASCADE,

FOREIGN KEY (country_code) REFERENCES
country(country_code) ON DELETE CASCADE
)
```

We then used mySQL workbench to merge the country_code and continent_code into the located table in a similar fashion to the entities discussed above.

Taxes table:

The taxes data was organized under the share-of-tobacco-retail-price-that-is-tax.csv file, which can be found on our GitHub page under the "filtered-data" folder. To turn the data into a table, the create table command was used. The PRIMARY KEY is a superkey that contains tobacco_stat_year and country_code, which is also a FOREIGN KEY taken from a country object, its also set with the property ON DELETE CASCADE, so upon the deletion of the country object the taxes object corresponding to the country_code will also be deleted.

```
CREATE TABLE Taxes(


country_code VARCHAR(10) NOT NULL,

tobacco_stat_year INT NOT NULL,

percentage_increase_tobacco INT,


PRIMARY KEY (country_code, tobacco_stat_year),

FOREIGN KEY (country_code) REFERENCES

country(country_code) ON DELETE CASCADE

):
```

The data insertion was done through the mySql Workbench csv import feature however the log file contains the commands processed to achieve such effect. The command below inserts a three element tuple into the population entity of the cs306 database (our database's name).

```
PREPARE stmt FROM 'INSERT INTO `CS306`.`taxes`
(`tobacco_stat_year`,`percentage_increase_tobacco`,`country_code`)
VALUES(?,?,?)
```

M_Disorder Table:

The M_Disorder data was organized into two separate csv files, share-with-mental-and-substance-disorders.csv and share-with-depression.csv, both of which can be found on the "filtered-data" folder in our GitHub. The contents of the two csv datasets were joined with the create table command. The PRIMARY KEY consists of a superkey pair with country_code and mental_health_stat_year. The country_code is also a FOREIGN KEY taken from a country object with the ON DELETE CASCADE property, meaning the M_Disorder object is set to be removed when the corresponding country object is removed.

```
CREATE TABLE M_Disorder(

country_code VARCHAR(10) NOT NULL,

mental_health_stat_year INT NOT NULL,

depression_rate REAL,

PRIMARY KEY (country_code, mental_health_stat_year),

FOREIGN KEY (country_code) REFERENCES

country(country_code) ON DELETE CASCADE

):
```

The insertion of the data was done by using mySql Workbench csv import feature but the log file contains the commands processed to achieve such effect. The command below inserts a three element tuple into the population entity of the cs306 database (our database's name).

```
PREPARE stmt FROM 'INSERT INTO `CS306`.`population`
(`mental_health_stat_year`,`depression_rate`,`country_code`)
VALUES(?,?,?)
```

Inserting the data:

Continents Table:

To make insertion from this csv file, we used the upload from csv option, which is processed as an "INSERT INTO" command in the background and is committed to the logs of the database.

```
PREPARE stmt FROM 'INSERT INTO `CS306`.`continent`
(`continent_name`,`continent_code`) VALUES(?,?)
```

The uploads to the Countries Table was done similarly:

```
PREPARE stmt FROM 'INSERT INTO `CS306`.`located`
(`country_code`,`continent_code`) VALUES(?,?)
```

Population Table:

Population data was organized into three separate csv files: population-demography.csv, not null life-expectancy.csv, population-density.csv (all three could be found in our github project under filtered data). To join the content of all three files under the same database table we first created a population entity with the create table command. It should be noted that the entity has a superkey pair containing the country_code and pop_year attributes. The country_code attrireferences an external entity, namely the country entity, and that the population entity is programmed to be removed from the database upon the deletion of the country object as specified with the ON DELETE CASCADE statement. The primary key being a foreign key and

the deletion of the entity being tied to the deletion of the country entity makes the population entity a weak entity as per our ER model.

```
CREATE TABLE population(

country_code VARCHAR(10) NOT NULL,
pop_year INT NOT NULL,
life_exp INT,
num_pop INT,
pop_density REAL,

PRIMARY KEY (country_code, pop_year),
Foreign Key (country_code) References
country(country_code) ON DELETE CASCADE
)
```

The actual data insertion was done with the mySql Workbench csv import feature however the log file contains the commands processed to achieve such effect. The command below inserts a three element tuple into the population entity of the cs306 database (our database's name).

```
PREPARE stmt FROM 'INSERT INTO `CS306`.`population`
(`pop_year`,`num_pop`,`country_code`) VALUES(?,?,?)
```

To merge the data we had to create a temporary table for the other csv files pertaining to the population entity and join them with the entity table. Then we used an update statement with an inner join statement to place the correct life expectancy and population density values based on the equality of the country_code and population_year attributes of the merged entities' rows as denoted by `ON population.country_code = lifeExpectancy.Code AND population.pop_year = lifeExpectancy.Year in the life expectancy table` and similarly with `ON population.country_code = populationdensity.Code AND population.pop_year = populationdensity.Year` in the population density entity. The rest of the commands specify which new field is to be set to which attribute of the population entity.

Creation of a temporary life expectancy entity:

```
CREATE TABLE `CS306`.`life-expectancy` (`Code` text, `Year` bigint,
`Life expectancy at birth (historical)` double)
```

Joining the life expectancy entity with the population entity:

```
UPDATE population INNER JOIN lifeExpectancy ON
population.country_code = lifeExpectancy.Code AND population.pop_year
= lifeExpectancy.Year SET population.country_code =
```

```
LifeExpectancy.Code, population.pop_year = lifeExpectancy.Year,
population.life_exp = lifeExpectancy.life_exp
```

Creation of a temporary population density entity:

```
CREATE TABLE `CS306`.`populationdensity` (`Code` text, `Year` int,
`Population density` double)
```

Joining the population density entity with the population entity:

```
UPDATE population INNER JOIN populationdensity ON
population.country_code = populationdensity.Code AND
population.pop_year = populationdensity.Year SET
population.country_code = populationdensity.Code, population.pop_year
= populationdensity.Year, population.pop_density =
populationdensity.popd
```

Then we imported data from the life-expectancy and population-density csv file into mySql workbench:

```
PREPARE stmt FROM 'INSERT INTO `CS306`.`life-expectancy`
(`Code`,`Year`,`Life expectancy at birth (historical)`) VALUES(?,?,?)
```

```
PREPARE stmt FROM 'INSERT INTO `CS306`.`populationdensity`
(`Code`,`Year`,`Population density`) VALUES(?,?,?)
```