

Data Engineering 1: Term 1 Project Report

Naida Dzigal

November 18, 2022

1 Task description

The goal of this project is to tie together different pieces learnt during the course. This was an individual exercise, where students were invited to go beyond the SQL statements and techniques covered in the course. A particular emphasis was put on the naming, packaging, versioning, documenting and testing of this project.

Higher level requirements included:

1. **Operational Layer:** Create an operational data layer in MySQL. Import a relational data set of your choosing into your local instance. Find a data set that makes sense to be transformed into an analytical data layer for further analysis. In the ideal case, you may use the outcome of HW1.
2. **Analytics:** Create a short plan of what kind of analytics can be potentially executed on this data set. Plan how the analytical data layer, ETL, Data Mart would look like to support these analytics. (Remember the ProductSales example during class).
3. **Analytical Layer:** Design a normalized data structure using the operational layer. Create a table in MySQL for this structure.
4. **ETL Pipeline:** Create an ETL pipeline using Triggers, Stored procedures. Make sure to demonstrate every element of ETL (Extract, Transform, Load).
5. **Data Mart:** Create Views as data marts.

Each section below contains information for each of the above requirements and these are listed in the same order.

2 Data Set Importing

The data was downloaded using the World Bank APIs, accessed through the Postman software. Seven datasets for 266 countries (data from years 1960 - 2021) were acquired and these are:

1. Population, indicator SP.POP.TOTL linked [here](#).
2. CO2 emissions, indicator EN.ATM.CO2E.PC linked [here](#).
3. GDP, PPP (current international \$), indicator NY.GDP.MKTP.PP.CD linked [here](#).
4. Electricity production from oil gas and coal sources, indicator EG.ELC.FOSL.ZS linked [here](#).
5. Electricity production from nuclear sources, indicator EG.ELC.NUCL.ZS linked [here](#).
6. Electricity production from renewable sources excluding hydroelectric, indicator EG.ELC.RNWX.ZS linked [here](#).
7. Electricity production from hydroelectric sources, with the indicator EG.ELC.HYRO.ZS linked [here](#).

Once the data was downloaded, it was imported into the MySQL Workbench, first by creating a schema named wdi, and then a separate table for each of the datasets above. The table names are: wdi_pop, wdi_co2_emissions, wdi_gdp, wdi_oilgascoal, wdi_nuclear, wdi_renewables and wdi_hydroelec. These tables were populated using csv files and it may be necessary to edit the file paths when reproducing this project. Each table was tested for import failures with the SELECT * FROM (table name) statement after importing, along with DESCRIBE (table name) to review the structure of imported tables. These tables made up the operational data layer (see Figures 1. and 2.)

3 Analytical Layer

From the operational data layer, two tables were created using procedures and together made the analytical layer: wdi_2000 and wdi_2010. Both of these tables contained the following variables for the year 2000 (wdi_2000) or 2010 (wdi_2010):

1. Country Name,
2. Country Code,
3. oilgascoal (a value for percentage of electricity production from oil, gas and coal sources),
4. nuclear (a value for percentage of nuclear electricity production),
5. hydroel (a value for percentage of hydroelectric electricity production),
6. renewables (a value for percentage of electricity production from renewables),
7. GDP (a value for GDP),
8. population (a value for population) and
9. co2_emissions (a value for CO_2 emissions).

See Figures 1. and 2. for a depiction of how the analytical layer fits in the ETL pipeline and also the overall EER diagram.

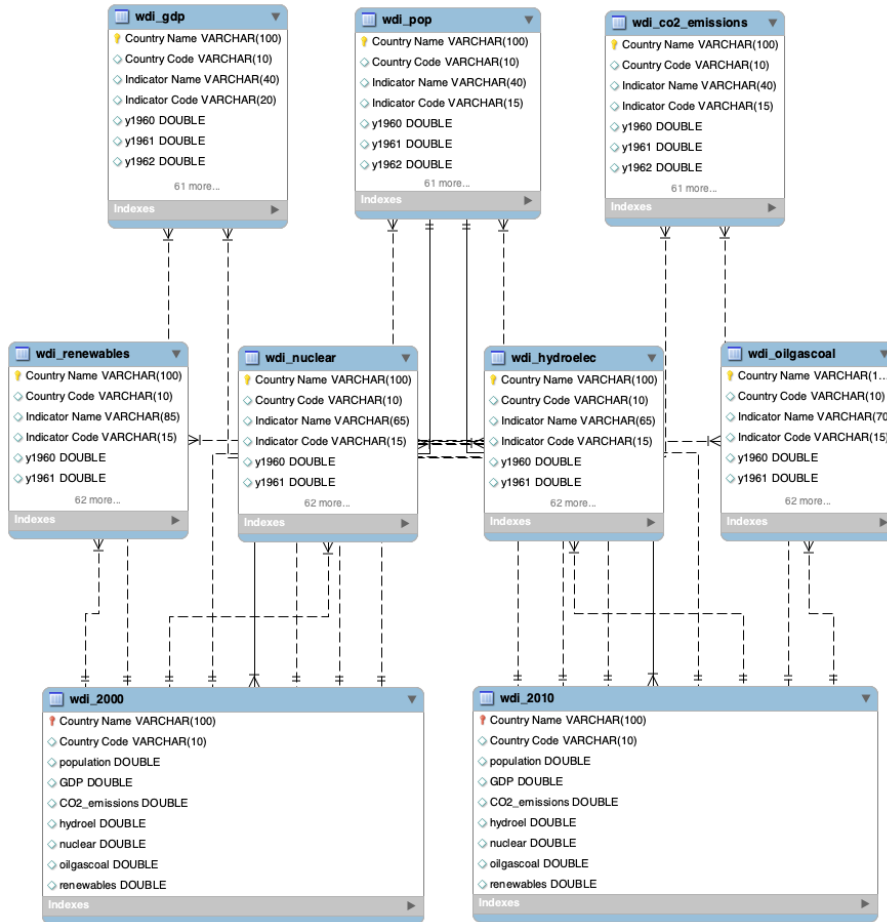


Figure 1: The EER diagram of operational and analytical layer database tables in MySQL. Note that not all fields are shown (e.g. there are hidden year fields that start at y1960 and go to y2021).

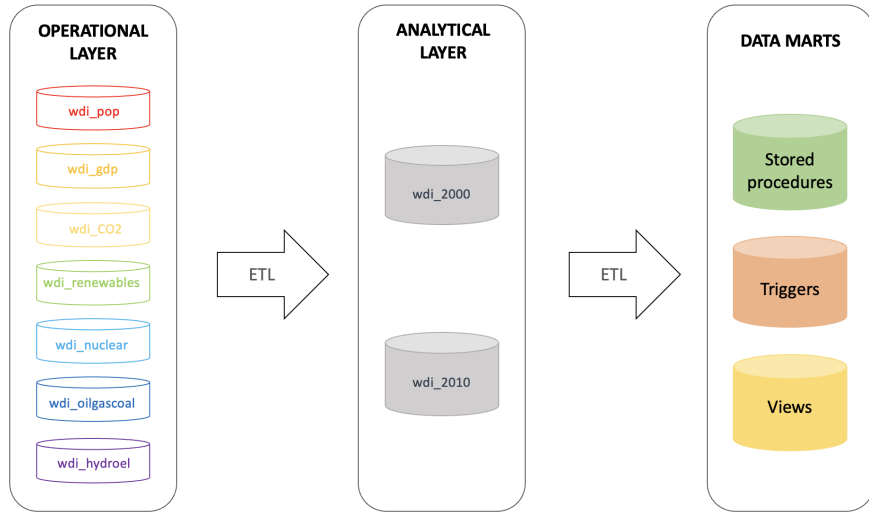


Figure 2: The ETL pipeline depicting the operational, analytical and data marts layers.

4 Data Analytics

4.1 Analytics Short Plan

A short plan was devised to investigate certain common knowledge statements and see if the data supports these:

1. What is the GDP pP for Austria versus its neighbours?
2. Do we have nuclear electricity production in Austria versus its neighbours in 2010?
3. What is the percentage of oil, gas and coal electricity production in Austria vs. Hungary in 2000?
4. Whether renewable (including hydroelectric) electricity production has been changing in Austria (increasing or decreasing) in 2010 versus 2000?
5. Whether renewable (including hydroelectric) electricity production has been changing in Austria (increasing or decreasing) in the last 5 years?
6. Which country (Austria or Hungary) has the higher CO2 emission rate pP? Did this change between 2000 and 2010?

5 ETL Pipelines

The ETL pipeline in this project was twofold (see Figure 2.): one was used to create snapshots of the operational layer data for the years 2000 (and a second table for the year 2010) and the second consisted of an event, a trigger and several views as data marts. The first one was described in detail in the previous section.

As described above, the analytical layer consists of two snapshots of the operational layer data for the years 2000 and for the year 2010. These snapshots contain the same variables which are percentages of a country's electricity production per source (nuclear, hydroelectric, renewables, oil, gas and coal), as well as information such as the population of a country, its GDP (GDP, PPP current international \$) and overall CO2 emissions (in metric tons per capita). The tables were made as part of an automated procedure that joined data from the operational level tables. We note that only one country was inconsistently named (Czechia vs. Czechoslovakia) in the datasets (out of 266 countries).

The second ETL pipeline was built by using MySQL Triggers and Events, and finally data marts with MySQL View. One stored procedure was written to extract from the analytical layer the data by country for the year 2000 (Get2000DataByCountry()).

5.1 Events and Triggers

An event was created for checking if there is new data for Austria in the 2000 table: CreateDataCheckAustria() (see Figure 3). This event was set to call Get2000DataByCountry('Austria') every 1 minute in the next 1 hour. The calls can be seen in the messages table (see Figure 5).

```
CREATE EVENT CreateDataCheckAustria
ON SCHEDULE EVERY 1 MINUTE
STARTS CURRENT_TIMESTAMP
ENDS CURRENT_TIMESTAMP + INTERVAL 1 HOUR
DO
    BEGIN
        INSERT INTO messages SELECT CONCAT('event:', NOW());
        CALL Get2000DataByCountry('Austria');
    END$$
```

Figure 3: An event CreateDataCheckAustria() calling Get2000DataByCountry('Austria') every 1 minute for 1 hour.

Next, a trigger as part of the ETL pipeline was written for monitoring the source table wdi_pop (see Figure 4). The trigger was set up so that if there is a change in the table wdi_pop, this will propagate to the analytical layer tables (such as wdi_2000). This was tested on 3 fictional countries inserted into this table, which were then printed by a select statement afterwards. A count(*) was also run to verify that 3 more countries were added to the tables. These changes could be viewed as well in the message table (see Figure 5).

```
CREATE TRIGGER after_insert
AFTER INSERT ON wdi_pop
FOR EACH ROW
BEGIN
    -- log the order number of the newly inserted order
    INSERT INTO messages
    SELECT CONCAT('new Country row: ', NEW.`Country Name`);
    -- archive the order and associated table entries to wdi_2000
    INSERT INTO wdi_2000 (`Country Name`)
    SELECT `Country Name`
    FROM wdi_pop
    WHERE `Country Name` = NEW.`Country Name`;
```

Figure 4: A trigger created for insertions in the wdi_pop table.

	message
▶	event:2022-11-18 22:19:05
	event:2022-11-18 22:20:05
	event:2022-11-18 22:21:05
	new Country row: Fantasia
	new Country row: Narnia
	new Country row: Middle Earth
	event:2022-11-18 22:22:05

Figure 5: A list of results in the message table: events of the CreateDataCheckAustria() as well as the messages triggered by adding the countries Fantasia, Narnia and Middle Earth to wdi_pop.

6 Views as Data Marts

The project contains four views as data marts: one that lists nuclear electricity production for West Balkan countries in year 2010 (West_Balkans_nuclear_2010), one that checks the same for hydroelectric production (West_Balkans_hydro_2010), one that checks nuclear electricity production in 2010 but this time for Central European countries (Central_Europe_nuclear_2010) and a

final one that checks the hydroelectric production for Central European countries in 2010 (Central_Europe.hydroel.2010).

```
CREATE VIEW `West_Balkans_hydro_2010` AS
SELECT
    wdi_2010.`Country Name`, wdi_2010.`Country Code`,
    wdi_2010.hydroel AS `Hydroelectric production (% of total)`
FROM
    wdi_2010 WHERE wdi_2010.hydroel > 0
AND (`Country Name` = 'Bosnia and Herzegovina'
OR `Country Name` = 'Croatia'
OR `Country Name` = 'North Macedonia'
OR `Country Name` = 'Slovenia'
OR `Country Name` = 'Serbia' );
```

Figure 6: The SQL code for creating the view of the West Balkans Hydroelectricity production for year 2010. The project contains three more views.




Result Grid   Filter Rows: <input type="text" value="Search"/> Export: 			
	Country Name	Country Code	Hydroelectric production (% of total)
▶	Bosnia and Herzegovina	BIH	46.8698902125672
▶	Croatia	HRV	61.6788321167883
▶	North Macedonia	MKD	33.4848484848485
▶	Serbia	SRB	31.7745771317104
▶	Slovenia	SVN	27.7945247616118
West_Balkans_hydro_2010 18			

Figure 7: A view created of the West Balkans Hydroelectricity production for year 2010. The project contains three further views as data marts.