CCT College Dublin

Assessment Cover Page

To be provided separately as a word doc for students to include with every submission

Module Title:	Analytical Methods for Big Data
Assessment Title:	Repeat CA summer 2015
Lecturer Name:	Kislay Raj
Student Full Name:	Angel Thompson
Student Number:	SBA23353
Assessment Due Date:	15 / AUG / 2025
Date of Submission:	15 / AUG / 2025

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Table of Contents

Database creation in MySQL

The first thing I did was create a new database called environment, from studying the two CSV files given, I then created more tables that represented key columns such as Countries, pollutants, wastes and sectors.

The original two csv's provided called total wastes transferred and total pollutions transferred acted as 'fact tables'. In the fact tables there were a number of different columns, so the data types were hard to define

I added Primary keys to all tables, including the original fact tables, which I also added foreign keys to, for relational models. The new tables I created acted as 'dimension tables'. (Anon., 2019)

Creating the dimension tables:

There were mostly two columns. Primary keys for each dimension table, and the other was a VARCHAR(variable length), with UNIQUE constraints on them to prevent duplicate entries. The variable length columns were the only columns of raw data transported from the fact table, the original dataset. There was only one dimension table out of the 5 created, with an extra column, the waste table.

The waste table had an extra column because it had two VARCHAR's, one was a classification for HW or NONHW, the other was a description for the previous code, like Hazardous or non-hazardous waste. I contemplated putting a waste Treatment column in there, which would indicate if the waste was disposed of or recovered, I decided against it for the following reasons. The waste, just like the other dimension tables, were created to represent a distinct set of values for that type.

The column Waste Treatment would show how an individual case by a country or industry, dealt with that waste, it varies and its context matters for that specific event.
The one to many relationship is important to mention, waste classification can be linked with many different treatments, for example hazardous waste can either be recovered or disposed of, depending on who handed it. So it was best not to store it with a single waste class like HW/NONHW, because it would potentially misrepresent how each waste was dealt with.

Populating the dimension tables:

was done by extracting unique values from the original datasets 'fact tables'. I used INSERT IGNORE to tell the databse to only insert a row if it doesn't violate a constraint. I used SELECT DISTINCT to avoid duplicate rows, adding only unique values

When investigating the wastage of the factories, I came to a dilemma when extracting information from the waste transfers table.
At first I was going to extract the raw values country, industry/sector type and reporting year to another table, directly from the Waste Transfers table, using the simple 'GROUP BY' query, but with research that approach did not seem appropriate. Waste transfers table had repetitive data, it was better for the analysis of the data to create dimension tables for Country, Sectors, Pollution and Waste classes. These dimension tables were joined to the original transfer tables using foreign and primary keys, it makes it easier to work with large and complex datasets

Creating structured relationships to facilitate 'JOINS', the dimension tables came in handy. While extracting data directly from the raw waste transfers table may have been simpler, using the 'JOINS' schema ensures the data quality and flexibility. (Wako, 2022)

Creation of Two New datasets:

I was tasked to create and extract information from the two given datasets, pollutant transfers and waste transfers. I created two new datasets, which were great to get an overall idea of each country's role every year in pollution and wastage.

The wastage and the type of industry, country as a common value:

The first dataset created was based on each sectors total waste each year, both grouped by year and country in ascending order, starting in 2007 til 2012, country started by Austria and ended with France, with a thousand rows. Below is the first five rows of that dataset.

	country_name	eprtrSectorName	reportingYear	total_waste
0	Austria	Energy sector	2007	55554.29
1	Austria	Chemical industry	2007	124939.19
2	Austria	Production and processing of metals	2007	226393.45
3	Austria	Waste and wastewater management	2007	1852365.40
4	Austria	Mineral industry	2007	9963.24

The Pollution and the Wastage per Year by Country:

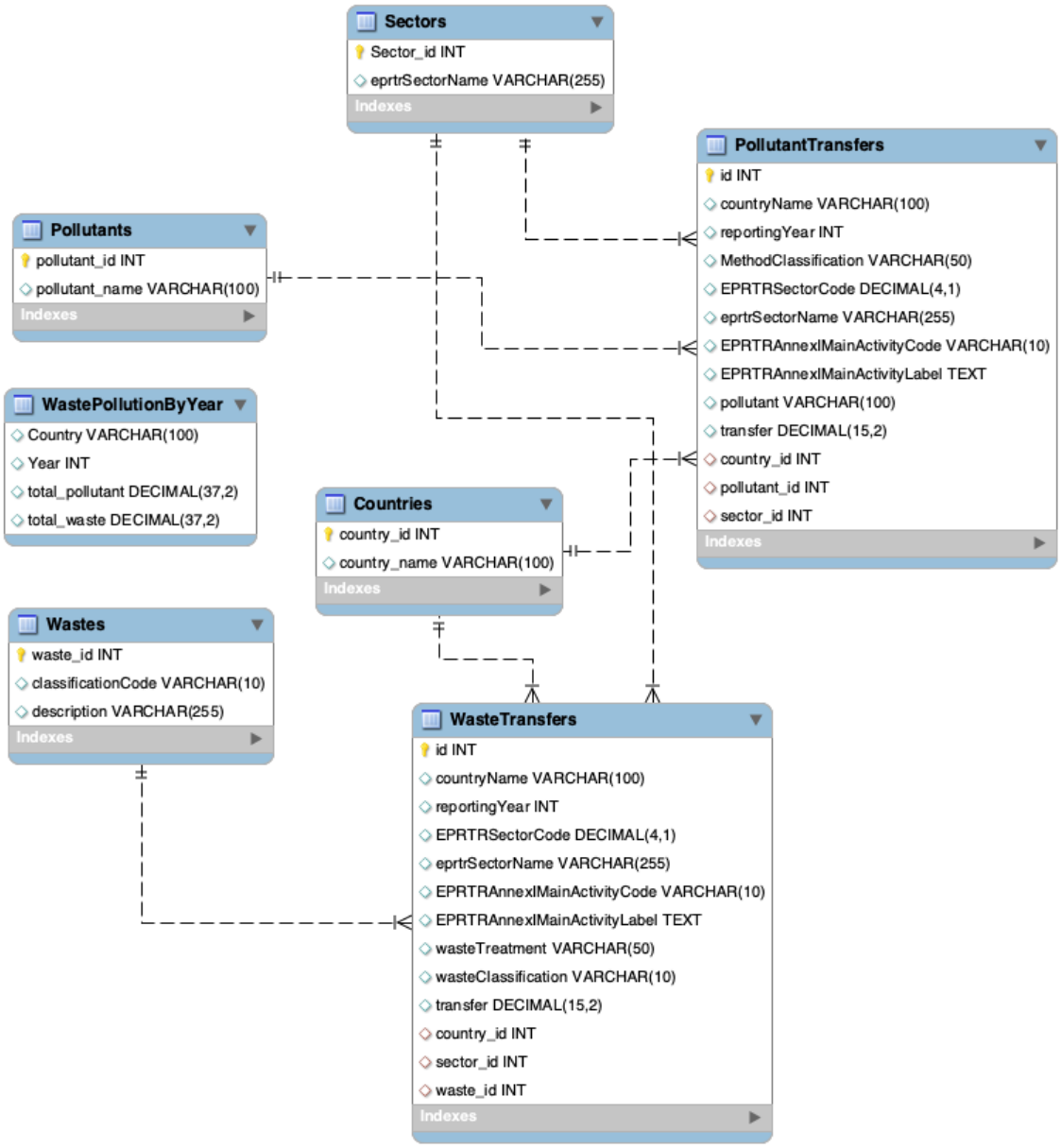
The second task was to study the pollution in the air and relate it with the wastage per year, I also added country so the whole dataset was not integers.
The dataset was also grouped by country and year, with 326 rows, it showed each countries total waste and total pollution per year.

Country	Year	total_pollutant	total_waste
Austria	2007	3.748543e+09	2.093215e+08
Austria	2008	4.336794e+09	2.407896e+08
Austria	2009	1.755751e+10	2.846215e+08
Austria	2010	3.442818e+09	4.023696e+08
Austria	2011	3.700248e+09	3.926058e+08

Creating this dataset involved joining the two original waste and pollution transfers, which was a difficult task in MySQL.
I created the waste and pollution table directly in my SQL, as the program was having difficulty displaying the selected columns, it dropped the connection, probably because I was fetching a lot of results from two different tables.

The EER diagram:

The EER diagram below, created in MySQL, shows the tables and the one to many relationships between dimension to fact tables.
After looking over the diagram, I have realised I never joined the Waste Pollution by year Table.


Sectors
Sector_id INT
eprtrSectorName VARCHAR(255)
Indexes
PollutantTransfers
id INT
countryName VARCHAR(100)
reportingYear INT
MethodClassification VARCHAR(50)
EPRTRSectorCode DECIMAL(4,1)
eprtrSectorName VARCHAR(255)
EPRTRAnnexIMainActivityCode VARCHAR(10)
EPRTRAnnexIMainActivityLabel TEXT
pollutant VARCHAR(100)
transfer DECIMAL(15,2)
country_id INT
pollutant_id INT
sector_id INT
Indexes
Pollutants
pollutant_id INT
pollutant_name VARCHAR(100)
Indexes
WastePollutionByYear
Country VARCHAR(100)
Year INT
total_pollutant DECIMAL(37,2)
total_waste DECIMAL(37,2)
Countries
country_id INT
country_name VARCHAR(100)
Indexes
Wastes
waste_id INT
classificationCode VARCHAR(10)
description VARCHAR(255)
Indexes
WasteTransfers
id INT
countryName VARCHAR(100)
reportingYear INT
EPRTRSectorCode DECIMAL(4,1)
eprtrSectorName VARCHAR(255)
EPRTRAnnexIMainActivityCode VARCHAR(10)
EPRTRAnnexIMainActivityLabel TEXT
wasteTreatment VARCHAR(50)
wasteClassification VARCHAR(10)
transfer DECIMAL(15,2)
country_id INT
sector_id INT
waste_id INT
Indexes

Conclusion:

There were two fact tables consisting of total wastes transferred and total pollutants transferred, from those fact tables there were important columns I created dimension tables for, those were Countries, Pollutants, Wastes and Sectors. These fact tables had a lot of different variations of data types and primary keys linking them to fact tables through their foreign keys.
Each of the dimension tables were links between the fact tables, not an information storer like the fact tables. They generally had two columns with a unique primary key and a VARCHAR column taken from the fact tables, but the waste table had an extra column for Hazardous and Non-Hazardous waste. I decided against adding 'Treatment' columns for Waste and Pollutants as there was a one-to-many relationship between the Class and treatment, like different waste and pollutants have different treatments.

By using 'JOINS'I populated the dimension tables by adding a primary key that linked to the foreign key to the information rich Fact table. I used the unique values and avoided using duplicates by 'SELECT DISTINCT' and 'INSERT IGNORE'.

I created an EER diagram which showed the relationships between the tables, the dimension and fact tables held a one-to-many relationship. This visualisation shows if tables were properly joined or not, which I realised I did not join the waste pollution by year table properly.

The datasets I created were:
Waste by sector, it was grouped by Year and Country and showed the total waste per sector from the years 2007-2012 with 1000 rows.
Pollution and waste, it was grouped by country and year, it showed the total pollution and waste per country with 326 rows. This was a challenging dataset as it was very large by combining all the values from the two raw fact tables.

Reflections:

In MySQL I spent a lot of time creating the dimension tables, with research creating them seemed to be the right practice, but in the end they seemed pointless for data analysis in python, only good for linking the raw data in Waste and Pollution tables.
after completing the Data exploration and Prepartion course, I wish I spent more time on Data cleansing techniques. I could have rounded up all values by the second decimal place, renamed columns for a better format and imputed the null values instead of dropping valuable rows.

Bibliography

Anon., 2019. *Difference between Fact Table and Dimension Table.* [Online]
Available at: https://www.geeksforgeeks.org/computer-networks/difference-between-fact-table-and-dimension-table/
[Accessed Aug 2025].

Wako, K., 2022. *Advanced SQL Techniques: Subqueries, Joins, and Aggregate Functions.* [Online]
Available at: https://medium.com/learning-sql/advanced-sql-techniques-subqueries-joins-and-aggregate-functions-5a44d4705000
[Accessed July 2025].