Human Freedom Index

Data

The database consists of data of the Human Freedom Index over the years 2008 to 2018, collected from Kaggle based on the report from the CATO Institute. The data was primarily chosen because of it's interesting content about freedom in different countries based on different aspects; we thought that some interesting things could be shown using the data. The data also seemed to fit the project well in terms of size and complexity. In figure 1 there is an extract of the data, in total there are 113 columns and 1620 rows.

year character varying (40)	iso_code character varying (40)	countries character varying (40)	region character varying (40)	hf_score numeric	hf_rank real
2018	ALB	Albania	Eastern Europe	7.81	43
2017	ALB	Albania	Eastern Europe	7.78	44
2016	ALB	Albania	Eastern Europe	7.63	50
2015	ALB	Albania	Eastern Europe	7.55	52
2014	ALB	Albania	Eastern Europe	7.65	48
2013	ALB	Albania	Eastern Europe	7.52	54
2012	ALB	Albania	Eastern Europe	7.54	53
2011	ALB	Albania	Eastern Europe	7.63	50
2010	ALB	Albania	Eastern Europe	7.75	45
2009	ALB	Albania	Eastern Europe	7.72	45
2008	ALB	Albania	Eastern Europe	7.7	45
2018	DZA	Algeria	Middle East & NA	5.2	154
2017	DZA	Algeria	Middle East & NA	5.03	154

Figure 1. Example of data

Purpose

The database was made with the purpose of answering the following questions and questions similar to them.

1. Which are the 5 countries with the highest and lowest score regarding religion in 2018?
2. Rank the regions according to the average freedom index
3. List the countries that have a higher score on freedom of expression than economic freedom in 2017
4. Which country has made the biggest jump in freedom index over the years, when it comes to ranking and score respectively?

5. For each year, which countries have the highest respectively lowest human freedom index scores on all three measures, hf_score, pf_score and ef_score?

Design process

The next step in designing the database was to create a schema to visualise how the database could be built. We began by creating the entities region, country and data, which are shown in the figure 2.

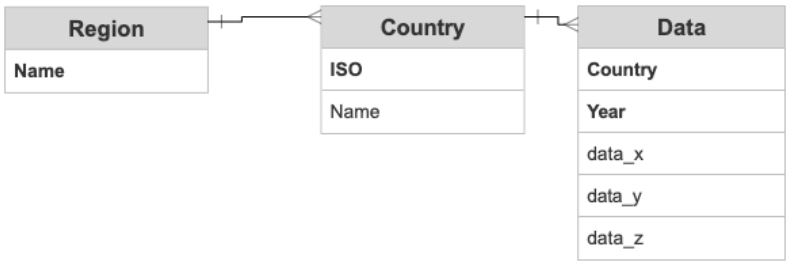

Figure 2. Initial ER model

The following relations between the different entities are shown with arrows; one-to-many relation between region and country because every country only exists in one region, but every region consists of many countries. The countries also have a one-to-many relation to data which contains data about the human freedom index according to country and year. Every tuple in the data table has a relation to a country, but only one country.

To reduce the size of the tables we decided to divide the data entity into smaller tables based on what kind of data it consists of, therefore the tables religion, expression, economy and summary were created. This is due to the fact that when information about religion is required there is no need to go through the economy data.

To differentiate between which data refers to which country and which year, every data-relation needed to contain those attributes, and also to make every tuple unique. Therefore we made country and year the primary key. Though there is a problem with redundancy, which is to be avoided in a well designed database, when almost every table contains country and year. However we could not find a better solution that did not involve a considerably greater amount of tables or the same amount of redundancy.

The final ER model is shown in figure 3 which illustrates all the entities and relations used in the database.

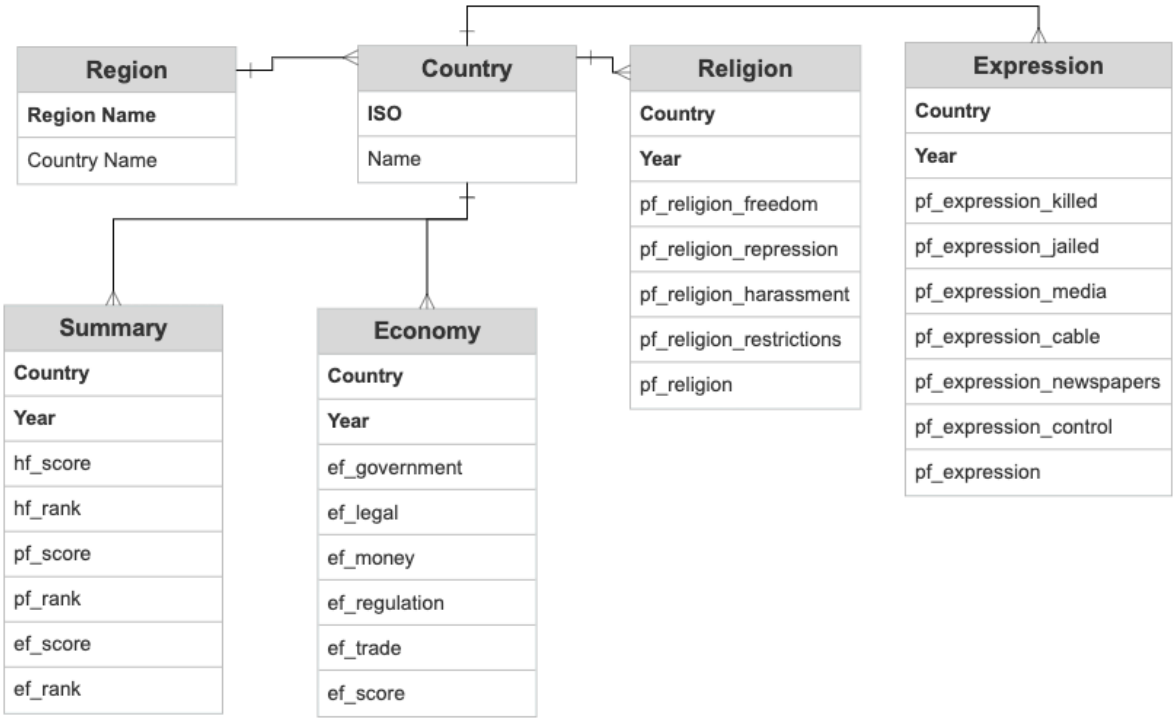

Figure 3. Final ER model

SQL queries and relational algebra

To answer the questions stated above queries were written. The first question "Which are the 5 countries with the lowest and highest score regarding religion in 2018?" were answered by the SQL code

```
SELECT name, pf_religion
FROM religion
WHERE pf_religion IS NOT NULL AND year ='2018'
ORDER BY pf_religion ASC
LIMIT 5
```

where the *ASC* can be switched to *DESC* to get the countries with the highest score. Translated to relational algebra the operation select is used to only choose the tuples from 2018 that have pf_religion data. The project operator is used to only show the name of the country and the religion score in the result. The whole segment is composed into an expression as it uses multiple relational algebra operators.

To answer the second question "Rank the regions according to the average freedom index" the following SQL sequence was used

```
SELECT AVG(summary.hf_score) AS hf_score, region.region_name
FROM region
```

```
INNER JOIN summary ON region.country_name = summary.name
GROUP BY region_name
ORDER BY hf_score DESC
```

Here the relational algebra operation project is used to only show region names and the hf_score, composition is used as in the first question and the assignment operation is used to give the column with average human freedom score a name. An inner join is used to create the result containing both the hf_score and the region names which comes from two different tables. As all countries should exist in both tables it should not matter which type of join is used, but inner join is relatively fast and if a country only exists in one of the tables it will not affect the result and should not be included in the join.

Question number three “List the countries that have a lower score on freedom of expression than economic freedom in 2017” used the sequence

```
SELECT economy.name, economy.year, pf_expression, ef_score
FROM expression
NATURAL JOIN economy
where ef_score>pf_expression and economy.year='2017'
```

The operations project, compose and select are used as above, but in this case the select operator includes a comparison. A natural join is used to make the result contain both ef_score and pf_expression, and every country only appear once as we only want to compare the pf_expression and ef_score for each country.

The fourth question “Which country has made the biggest jump in the Freedom Index over the years, when it comes to ranking and score respectively?” is answered by

```
SELECT MAX(hf_rank) - MIN(hf_rank) AS rank_change, name
FROM summary
WHERE year ='2008' OR year = '2018'
GROUP BY name
ORDER BY rank_change DESC
LIMIT 1
```

for ranking and

```
SELECT MAX(hf_score) - MIN(hf_score) AS rank_change, name
FROM summary
WHERE year ='2008' OR year = '2018'
```

```
GROUP BY name
ORDER BY rank_change DESC
LIMIT 1
```

for score. Here a new combination of project, select and rename is used to show the needed columns and choose the data needed to make the subtraction of the highest and lowest rank/score for 2008 and 2018.

To answer the last question “For each year, which countries have the highest respectively lowest human freedom index scores on all three measures, hf_score, pf_score and ef_score?” the following code were used for the lowest score

```
SELECT DISTINCT ON (year) name, year, hf_score, pf_score,
ef_score
FROM summary
WHERE hf_score IS NOT NULL
AND pf_score IS NOT NULL
AND ef_score IS NOT NULL
ORDER BY year, (hf_score, pf_score, ef_score) ASC
```

where the *ASC* can be switched to *DESC* to get the countries with the highest score. Also here project and select are used to show the right result, though the projection is distinct for the year column because we are only interested in one result for every year.

Discussion

During the design process it was difficult to decide how to divide the data into tables when every country has data for ten different years and redundancy is something to avoid. One option that we considered was to make one table for every country containing all the data concerning that country, but that would be too many tables to be effective and easy to work with. Another one was to make one table for every year containing all the data regarding that year, which would lower the redundancy a little. That could have worked well, but we felt that the divide we did according to the kind of data was more effective because less tables would be involved in the queries. Of course a combination of the options could have been made, but we did not think it was necessary on account of the size of the data and the scope of the project.

The first query is checking if the pf_religion column is null, because if it is, those countries will be the ones with the lowest score regarding religion. This was realised when executing the query and therefore the null check was implemented. The problem did not occur in all the other queries and the check was therefore not

added, though it should because the data could vary. For example, a null value would not give a null result in query number two because an average is calculated, but the null check would give a different result if a null was found. More error handling in general would improve the queries a lot because now they are designed only for the data input we used.

When writing the queries we realised that we had to make the questions more specific compared to what we initially had written because otherwise the result did not say anything. A good example of this is the years specified in questions one and four, without that limitation the same country could appear more than once and scores from different years would show. This made it difficult to interpret the result; the score from Yemen in 2018 is not easily compared to China in 2010.

When looking at the finished queries we did realise that the table country is never used because the data already exists in the other tables; a clear example of unnecessary redundancy. Therefore that table should be removed from the database, though the entity is good for visualising in the ER-diagram.