BC2402
Project Presentation
Recommendation to WHO

Considerations

1. Our understanding of the 2 types of data modelling approaches
2. Assumed needs and requirements of WHO
3. Current implementation of the relational and non-relational dataset
4. Nature of the data

Comparison of data modelling approaches (1)

Flexibility

- Rigidness of schemas
- Dealing with a change in data types requirements

Comparison of data modelling approaches (1)

Flexibility

- Rigidness of schemas
- Dealing with a change in data types requirements

Scalability

- Vertical v.s. Horizontal scaling (and the cost involved)

Comparison of data modelling approaches (2)

Data storage footprint

- Amount of data duplication

Comparison of data modelling approaches (2)

Data storage footprint

- Amount of data duplication

Data Integrity

- ACID v.s. BASE
- Data integrity v.s. Data availability

Comparison of data modelling approaches (2)

Data storage footprint

- Amount of data duplication

Data Integrity

- ACID v.s. BASE
- Data integrity v.s. Data availability

Suitability for Data Analysis

- Type of data (Well structured v.s. Semi-structured)

Assumption about requirements of WHO

- Decision affects lives (millions of it!)
- Access to **accurate** and **consistent** data will be of utmost importance.

Recommendation and Justification (1)

Data model of choice: **Relational table (SQL)**

Cost-savings v.s. Human lives

- Priority for human lives EXCEEDS cost-savings

Recommendation and Justification (1)

Data model of choice: **Relational table (SQL)**

Cost-savings v.s. Human lives

- Priority for human lives EXCEEDS cost-savings
- Do not require high data availability (unlike banking services)
  - More READ (a few times a day) than WRITE (once a day)
  - Concurrency not an issue → A simple DB instance should works too

Recommendation and Justification (1)

Data model of choice: **Relational table (SQL)**

Cost-savings v.s. Human lives

- Priority for human lives EXCEEDS cost-savings
- Do not require high data availability (unlike banking services)
  - More READ (a few times a day) than WRITE (once a day)
  - Concurrency not an issue → A simple DB instance should works too
- Savings are negligible since current dataset not in TBs
  - SQL might even be cheaper due to memory footprint!

Recommendation and Justification (2)

Data model of choice: **Relational table (SQL)**

Unlikely for data structure and requirements to change

- COVID been around since Dec 2019 (2 years!)

Recommendation and Justification (2)

Data model of choice: **Relational table (SQL)**

Unlikely for data structure and requirements to change

- COVID been around since Dec 2019 (2 years!)

Type of data

- Data are mostly text and integers

Limitations

Table: covid19data

Columns:

iso_code	text
continent	text
location	text
date	text
total_cases	text
new_cases	text
new_cases_smoothed	text
total_deaths	text
new_deaths	text
new_deaths_smoothed	text
total_cases_per_million	text
new_cases_per_million	text
new_cases_smoothed_per_million	text
total_deaths_per_million	text
new_deaths_per_million	text
new_deaths_smoothed_per_million	text

"*Covid19data*" are all in TEXT

Limitations

Table: covid19data

Columns:

iso_code	text
continent	text
location	text
date	text
total_cases	text
new_cases	text
new_cases_smoothed	text
total_deaths	text
new_deaths	text
new_deaths_smoothed	text
total_cases_per_million	text
new_cases_per_million	text
new_cases_smoothed_per_million	text
total_deaths_per_million	text
new_deaths_per_million	text
new_deaths_smoothed_per_million	text

"*Covid19data*" are all in TEXT

Solution to ensure accuracy:

- Datacasting to correct type before performing queries OR
- Normalization and enforce data type