Case Study

Initial Observation:

The given data is about the Job vacancies in various provinces in Canada. It has total of 16 column, out of which 2 are empty. One of the column 'VALUE' has around 55,000 blank values. There are many transitive functional dependencies and needs to be normalized.

Data Cleaning:

I have performed the following data cleaning operations:

1. Removed empty columns 'SYMBOL' and 'TERMINATED'.
2. Inserted 'Nan' into blank cells in column 'VALUE'.
3. Removed the duplicate rows.
4. Separated meta data file into individual CSV files.

Initial Data Model:

I have created the initial data model without establishing any relationships. The tables are not yet normalized.

The initial data model has the following design issues:

1. Transitive Functional Dependencies.
2. Multivalued columns.

Transitive Functional Dependencies:

1. In the table 'JOB_STATISTICS', column 'DGUID' depends on 'GEO' other than the primary key of the table.
2. Column 'UOM' depends on 'UOM_ID'.
3. Column 'SCALAR_FACTOR' depends on 'SCALAR_ID'.

Multivalued column: The Column ‘CUBE NOTES’ in the table ‘CUBE’ has multiple values in each cell.

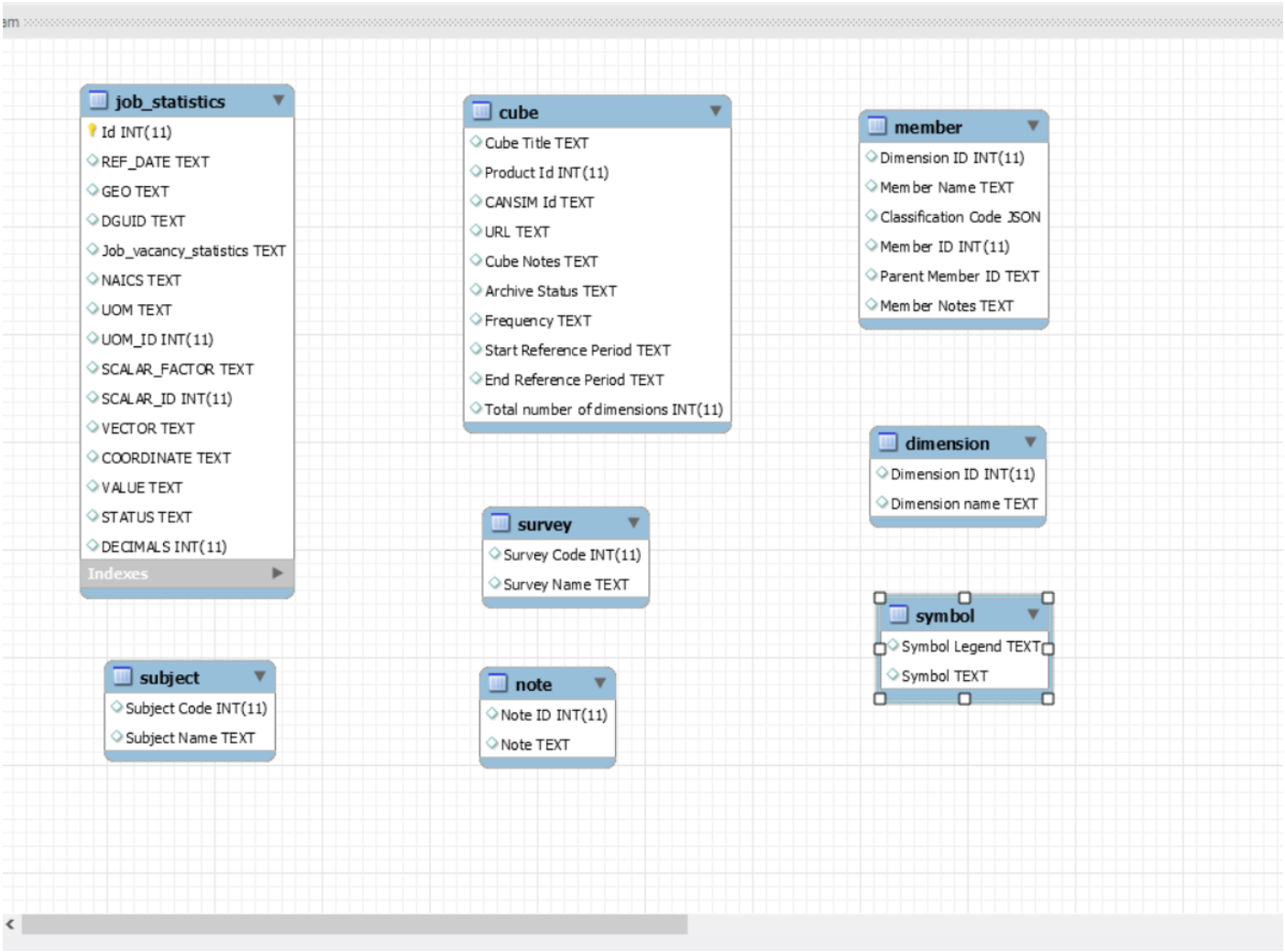


Figure:1 Initial data model without relationships

Final Data Model:

Below is the final data model with normalized tables and established relationships among them.

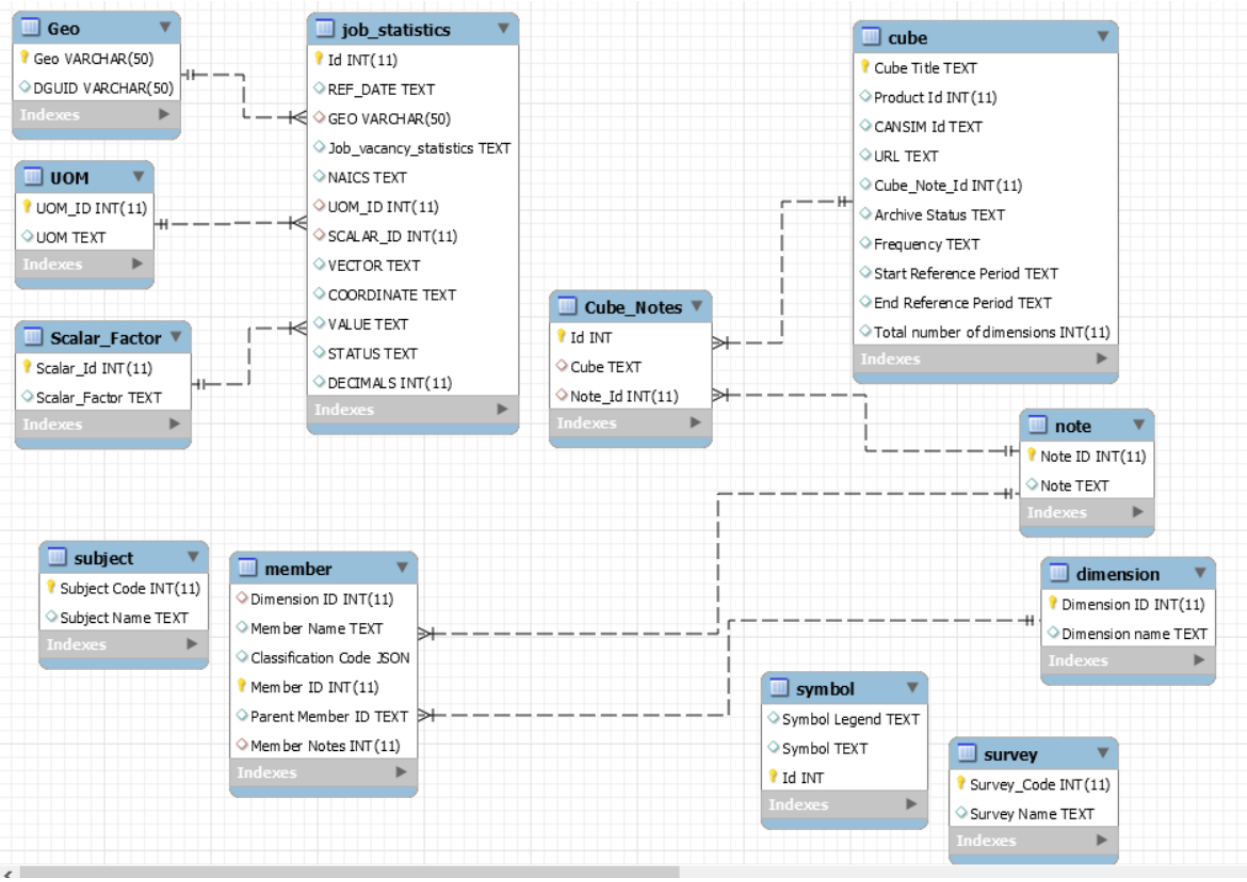


Figure:2 Final Data model without any design issues

I have removed the above-mentioned design errors to get the final data model.

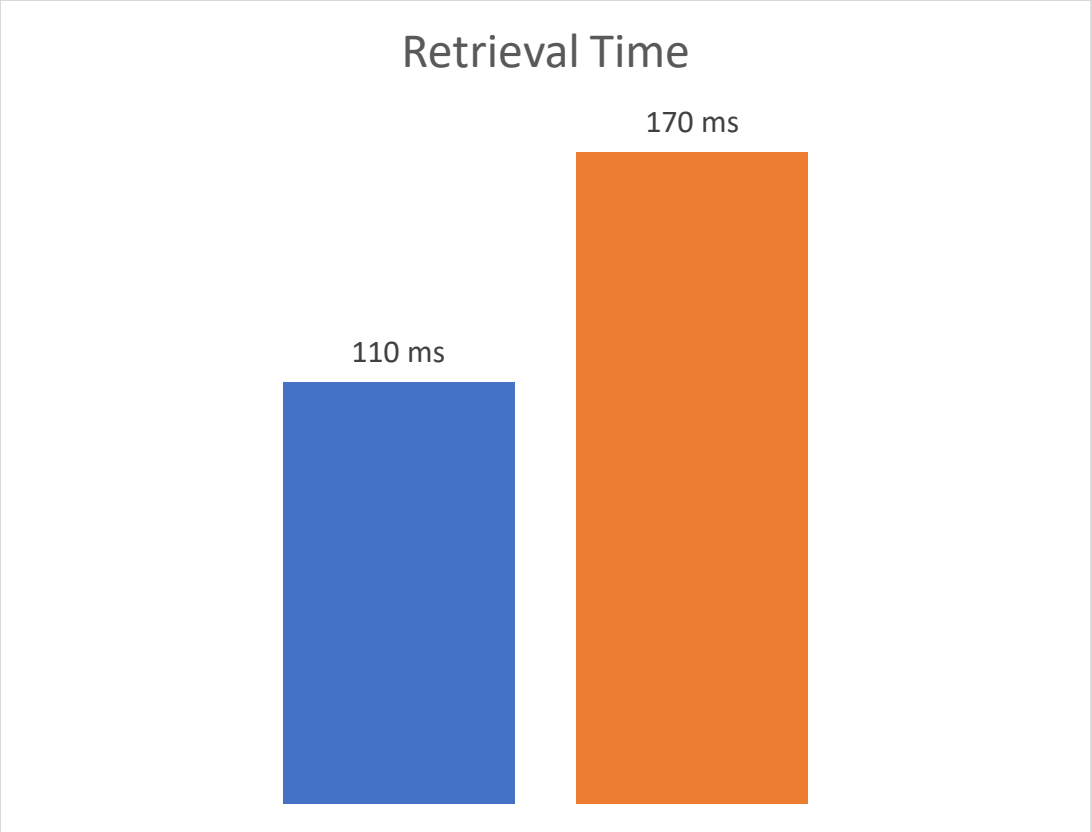
- ➔ Created a new table ‘GEO’ with columns ‘GEO’ and ‘DGUID’ and removed ‘DGUID’ as column in ‘JOB_STATISTICS’ table. Added a foreign key from ‘GEO’ table. This removed the transitive functional dependency between ‘GEO’ and ‘DGUID’ in ‘JOB_STATISTICS’.
- ➔ Created a new table ‘UOM with columns ‘UOM and ‘UOM_ID and removed ‘UOM as column in ‘JOB_STATISTICS’ table. Added a foreign key from ‘UOM

table. This removed the transitive functional dependency between ‘UOM’ and ‘UOM_ID’ in ‘JOB_STATISTICS’.

- ➔ Created a new table ‘SCALAR_FACTOR’ with columns ‘SCALAR_ID’ and ‘SCALAR_FACTOR ‘and removed ‘SCALAR_FACTOR’ as column in ‘JOB_STATISTICS’ table. Added a foreign key from ‘SCALAR_FACTOR’ table. This removed the transitive functional dependency between ‘SCALAR_FACTOR’ and ‘SCALAR_ID’ in ‘JOB_STATISTICS’.
- ➔ Created a new table ‘CUBE_NOTES’ with columns ‘CUBE’ and ‘NOTE_ID’ and removed ‘CUBE NOTES’ as column in ‘JOB_STATISTICS’ table to remove the multivalued column.

Response time measures:

I have used Express Js [1], Node Js for my backend [2] and HTML, JQUERY for frontend. After taking the input of Database and the Location from the form, upon the form submission, a POST request has been sent to the server where it receives the data from client and uses that data to query the data from the Database. The retrieved data is then sent as a response to the POST request to the client [1].



The average retrieval time taken by MySQL is 110 milliseconds while Mongo Db took 170 milliseconds on average. The insertion time taken by MySQL is 2 seconds which is slightly more than Mongo Db's 1.3 seconds. As the data is more structured in MySQL, the retrieval time is far better even after using complex queries using joins whereas Mongo Db's speed in inserting data from an unstructured data source is way too good.

References:

Professional Internet Site:
[1] Express JS Documentation on 'Routing' [Online]
Available: https://expressjs.com/en/guide/routing.html
[Accessed on Oct. 13, 2019]
[2] W3School's Tutorial on 'Node.js' [Online]
Available: https://www.w3schools.com/nodejs/nodejs_mysql.asp
[Accessed on Oct. 12, 2019]