

|  |
| --- |
| Canadian Trade Database |
| Image result for canada iconRelated image |
| June 12, 2019  **The Containeers**  Ghislain Beaulac  Imelda Yumul  Roberta Rios Foltran |



Table of Contents

[1. Overview iii](#_30j0zll)

[2. Assumptions/Constraints/Risks iv](#_3znysh7)

[2.1 Assumptions iv](#_2et92p0)

[2.2 Constraints iv](#_tyjcwt)

[2.3 Risks iv](#_1t3h5sf)

[3. Design Decisions v](#_4d34og8)

[3.1 Key Factors Influencing Design v](#_2s8eyo1)

[3.2 Functional Design Decisions v](#_17dp8vu)

[3.3 Database Management System Decisions v](#_3rdcrjn)

[3.4 Performance and Maintenance Design Decisions v](#_26in1rg)

[4. Detailed Database Design vi](#_35nkun2)

[4.1 Performance Monitoring and Database Efficiency vi](#_1ksv4uv)

[4.1.1 Operational Implications vi](#_44sinio)

[4.1.2 Data Transfer Requirements vi](#_2jxsxqh)

[4.1.3 Data Formats vi](#_z337ya)

[Appendix A: Acronyms vii](#_1y810tw)

[Appendix B: Scripts viii](#_1ci93xb)

|  |
| --- |
| **Overview** The world of nations is one giant trading marketplace. Trade is an exchange of goods between two parties, which can also be referred to as commerce. It is generally conducted for the purpose of providing a nation with commodities it lacks in exchange for those that it produces in abundance by other nations. The exchange of commercial goods or commodities between two nations can be in form of tangible products or of services. As trade originated throughout much of recorded human history, the amount of data that can be gathered is vast and complex.  Trade data are important for obvious reasons, to understand global economic growth and trends, business analysis and strategies, and social awareness. There are dozens of official sources of data on international trade. However, if you compare these different sources, you will find that they do not agree with one another. Even if you focus on what seems to be the same indicator for the same year in the same country, discrepancies are large. And why data doesn’t add up? It is due to difference in guidelines used by countries to record and report trade data.  In this project, we will focus mainly on Canadian Trade, its imports and exports transactions in maritime settings. In short, all goods coming in and out of Canada by sea, shipped in containers loaded in big vessels of international steamship lines.  Importers, exporters, and carriers are identified by assigned number to record all transactions. Commodities are processed for clearance prior to release by using HS Codes. Vessel sailings and commodities are recorded & monitored in every ports in all routes involved.  All data pertaining to imported commodities and exporting goods to & from Canada are processed and managed by The Canada Border Services Agency (CBSA). It has a mandate to ensure that all goods entering Canada do not pose a risk to the health, safety, and security of Canadians, while facilitating the free flow of legitimate goods.  Canadian Trade Database System Dependencies  *Electronic Data Interchange*  Electronic Data Interchange (EDI) is a standardized way of electronically exchanging information between and within businesses, organizations, government entities and other groups. These standards specify the formats, character sets, and data elements used in the exchange of business documents and forms.  Portal  A Portal is a secure data submission option developed by the CBSA that will allow the trade community to electronically submit their information through the Internet.  The eManifest Portal was developed primarily for small- to medium-sized enterprises to facilitate their compliance and ease the transition from paper reporting to pre-arrival electronic data submission. The eManifest Portal is designed for highway carriers to submit pre-arrival cargo and conveyance data and to receive House Bill Manifest Forwards sent by other Trade Chain Partners (TCPs).  The Trusted Trader Portal (TTP) is a secure online tool that allows members of the trade community to complete and submit applications for membership in the Partners in Protection (PIP) program, as well as maintain their membership documentation online. The PIP program only accepts applications and updated Security Profiles through the TTP.  Differences between Electronic Data Interchange and Portal  Terminology, data requirements and business rules vary somewhat between EDI and the Portal. The differences between these two submission options are largely because EDI and the Portal are based on different technologies and interfaces and face different technical constraints when interfacing with the CBSA system.  With respect to terminology, for example, EDI documentation often refers to transmission of trade data while Portal documentation typically refers to submission of Trade Documents. EDI refers to Statuses and Notices while Portal refers to Statuses.  *Source:* [*https://www.cbsa-asfc.gc.ca/eservices/menu-eng.html*](https://www.cbsa-asfc.gc.ca/eservices/menu-eng.html)  Keeping in mind the processes involved in Canadian imports and exports, the Canadian Trade Database proposed relies on Canadian Government Plan of “Open Government” portals. These portals aim to provide citizens with access to resources like open data, open information, and open dialogue. The data that will be used will be extracted from the electronic data interchange between business companies, customs brokers, carriers submitted to CBSA.  Having a Canadian-centric trade data available online in real or near real time, gathered, stored and processed using Canadian trade guidelines, it will help Canadian businesses for the following;   * Find new import or export markets; * Determine competition for your products; * Identify domestic opportunities for import replacement; * Discover a country's trade balance; * Find out what products countries are importing, exporting or re-exporting. |
| **Assumptions/Constraints/Risks**Assumptions Given the dependency of the proposed Canadian Trade Database on the government’s process of data collection for the country’s commercial trade, the following assumptions are described in designing the trade database.  On Data Collection   * Trade data are submitted by importers/exporters, licensed customs brokers, & carriers. * All data collected relating to Canadian trade refers to imports & exports of commercial goods are transmitted thru CBSA’s Electronic Data Interchange (EDI). * Manual data entry is expected for some trade transactions. * Due to manual entry, data accuracy is somehow compromised due to human error. * Change in data entry, policy, and access are subject to government’s regulations. * Delay in data validation is expected.   On Data Management System   * Relational Database on SQL Server * Tableau for Data Visualization   Group brainstormOn End Users  Classroom Companies, business owners and entrepreneurs are expected to be heavy users of the database for business strategies, analysis and market trends.  Professor Students, researchers and other people of the academe are most likely to use the database as well for educational purposes.  Other government bodies doing research and analysis. Constraints The Canadian Trade Database will be based on relational model as it is mainly collection of relations of data. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.  The table name and column names are helpful to understand Canadian Trade as a whole. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.  The limitations that can be foreseen in the proposed Canadian Trade Database are as follows:   * The database can have limits on field lengths which can't be exceeded. * Relational database can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated. * Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.   As known constraints for relational databases, the Canadian Trade Database is mostly like to have the following constraints;   * Domain constraints – This can be violated if an attribute value of data is not appearing in the corresponding domain or it is not of the appropriate data type. * Key constraints - The value of the attribute for different tuples in the relation must be unique. * Referential integrity constraints - happens where relation refers to a key attribute of a different or same relation. However, that key element must exist in the table.  Risks Attempting to maintain big data in a relational database would be an almost impossible feat from both a management and a cost standpoint.  Basic steps to mitigate risks:   * Create restrictions regarding who has access to certain data. It is important that people only have access to documents and information that are necessary to tasks. * Maintain records of user activity to audit who is accessing what data and whether there have been any breaches. * Use additional security products built specifically to protect big data in order to bolster data’s safety.  **Design Decisions**Key Factors Influencing Design The following factors influenced the choice of a relational database for this solution:  DatabaseStructure data only  Well defined fields and limited number of relations  Main purpose is to enable analytics  Data volume to be handled is suitable for relational database  *https://lh4.googleusercontent.com/qPssdr3nBx_a-eWwqXrvDfDGLfWIGGHqKfeVLiK_wFGCMikRrbpef1bSGIbv_Q53pwKUcLQLnPBttE2irBGbNRpZcyzvpQ-NS4RsTq5dkbZD1bCzEXz9dtaGBNw4vA* |
| *https://munvo.com/2017/07/18/hadoop-big-data-vs-relational-databases/* Functional Design Decisions Import Data Type  We expect our data import to be using CSV format that will be coming from the outside service. Another way of importing the data will be through a website upload form that user can choose to upload a CSV as well. Through an ETL (Extract Transform Load) process, we will skip any duplicate and continue with the next entry.  Upload Process  The upload process will be executed in a batch mode with a Python script to read the CSV file and insert the data (if not already exists) into SQL Server. The reading of the file is sequential since our backend server will read the CSV file sequentially line by line, but we will use a batch insert after the file is read. We will also skip any insertion errors and instead simply return the number of insertions and skipped entries, this count of insert & skipped, will be sent back as confirmation to the REST API and/or to a mailing list when the process runs overnight. The skipped entries will have their UIDs, with the reason of why it was not inserted, showing in the logs or notification so that user can manually re-insert them if needed  How Data Appear to the User  There will be 2 ways of showing the data to the users.  First option is through a data grid, we will be reading the data through a REST API and then display it into the data grid. Since we also expect few thousand lines of rows of data to be shown in the data grid, we will use a grid with pagination. The use of pagination will limit the amount of data sent over the wire and will provide a better user experience. For the pagination in the database, it will simply require a limit of entries to display (top) and the start/end cursors (offset). User will have access to change the pagination size if he/she wishes to see more results displayed in the data grid.  A screenshot of a social media post  Description automatically generated  Second option will be to use [Tableau](https://www.tableau.com/) for analytics. Tableau allows to connect directly to an SQL Server connection, and that allows anyone with the [Tableau for Desktop](https://www.tableau.com/products/desktop) to use it directly. Since not everyone will have Tableau installed (because it’s not free), we will also have Tableau dashboard to allow user to use built-in charts with some minimal data drilldown available, that in terms will require the need of a [Tableau Server](https://www.tableau.com/products/server). Database Management System Decisions Database Choice  We will be using Microsoft SQL Server (abbreviated to MS-SQL) because it has the capacity to support a huge variety of transaction processing applications as well as business intelligence and analytics in corporate IT environments. Many other Database Management Systems (DBMS) are less secure, while MS-SQL enables us to maintain the security, integrity and consistency of the data we are working with. There are also other constraints related to integrity that help you maintain data accuracy, collectively known as the ACID properties (atomicity, consistency, isolation, and durability), in short consistencies is our biggest concern.  Database Transactions  MS-SQL also provides Transact- SQL (T-SQL) which enables us to program queries and commands. We will be programming a few commands and methods to better access the data; we will also use VIEWs as well.  A screenshot of a cell phone  Description automatically generated  Simple Installation and Automatic Updates  A screenshot of a social media post  Description automatically generatedSince it is a Microsoft product, it has an easy installation and automatic updates when installed on a Windows (typically Windows Server) platform. The advantage this implies is that you don’t require a special toolkit to run the installation. The system updates are detected automatically and downloaded without any input from the operator. Optimized Security Feature The Microsoft SQL server’s data compression feature is built in, and encryption enables you to modify programs for data encryption. The server also pairs access control with efficient and secure permission management. SQL Servers use Policy-Based Management to keep your security policies compliant and updated. This means that only authorized personnel will be able to access the database.  Simple Management and Maintenance  Maintenance is easy to perform, including backups and recovery. Effective data management practices will ensure that your data is available and recoverable in the event of data loss elsewhere. We can schedule differential backup that will run every 15 minutes and a full backup that will run every night when the system is less in use. Performance and Maintenance Design Decisions Database Distribution  *https://cdn-images-1.medium.com/max/1600/1*WPnv_6sG9k4oG3S1A09MDA.jpeg*In the CAP theorem, we are looking for Consistency and Availability (CA) and since our focus is analytics, we expect our users to be consistently connected on the network. Our SQL Server installation fits nicely in the CA side.  [*https://towardsdatascience.com/cap-theorem-and-distributed-database-management-systems-5c2be977950e*](https://towardsdatascience.com/cap-theorem-and-distributed-database-management-systems-5c2be977950e)  Database File Updates  Since our main interest is mainly analytics, we are comfortable uploading CSV files only once a day. This might increase in the future but for the beginning, it is more than enough to stick with daily upload. This upload can be done outside of working hours, and we will be using SQL transactions in batch mode to send all data in 1 big insert.  Database Backup and Restoration  Since we plan to only upload data once a day, we can also plan a backup scheduled after the upload. Our backup plan will be a full backup that runs every night outside of working hours. With that in mind, there is no need to use incremental backup, however if the business grows and we ever decide to upload more frequently we might want to review this schedule and possibly enable incremental updates. For example, if we ever plan to upload data every hour, then we would enable incremental backup every hour as well. The tool that we will use is Microsoft SQL Server Management Studio (SSMS) is free and available from Microsoft at this [link](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017)  Data Purging and Archiving  We mentioned earlier that all records pertaining to the import are kept for 10 years, with that in mind we will keep all data for six years and purge it after that period. We will archive any other after 6 years to keep the database performant. **Detailed Database Design** The data will be structured in tables, following a relational database model. The csv to be ingested has the fields listed below:   |  |  |  |  | | --- | --- | --- | --- | | Direction | US Origin/Destination City | Container Size | Foreign Company State | | Year | US Origin/Destination State | Ship Line Name | Foreign Company Country | | Quarter | Commodity Group | Voyage | Foreign Company Address | | Month | HS Code 2 Digit | Vessel Name | Quantity and Units | | WeekOfYear | HS Code 2 Description | Canada Company Name | UOM | | Date | HS Code 4 Digit | Canada Company City | Value | | Port of Origin | HS Code 4 Description | Canada Company Province | TEUS | | Port of Destination | Commodity Description | Canada Company Address | Shipment count | | Country of Origin | Reefer | Foreign Company Name |  | | Country of Destination | Hazmat | Foreign Company City |  |   The diagram below shows the conceptual model, listing the tables, fields and their relationships:  The logical model and ERD is illustrated below:     * *Physical Data Model (PDM) with a description of the DBMS schemas, sub-schemas, records, sets, tables.*   Physical Data model  4.1.1 Table Company   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | id | int | PK | Unique ID, assigned sequentially as companies are registered in the database | 4 | | name | varchar(64) |  | Company name | 64 | | address | varchar(255) |  | Company address (street and number) | 255 | | city\_id | int |  | Unique city ID | 4 | | country\_code | char(2) | subject to validation | Unique country code | 2 |   Bytes per entry: 329 4.1.2 Table shipment  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | uuid | int | PK | Unique identifier (https://en.wikipedia.org/wiki/Universally\_unique\_identifier) | 16 | | Direction | int | subject to validation | Possible values:   * I (to designate import) * E (to designate export) | 1 | | time\_created | timestamp | Automatically populated by the system | Timestamp of when the transaction is recorded | 19 | | Date | date | Calculated | Date is a calculated field, derived from timestamp | 10 | | Year | int | Calculated | Year is a calculated field, derived from timestamp | 4 | | Quarter | int | Calculated | Quarter is a calculated field, derived from timestamp | 1 | | Month | int | Calculated | Month is a calculated field, derived from timestamp | 2 | | Week | int | Calculated | Week is a calculated field, derived from timestamp | 2 | | Origin\_Port\_ID | int | Not null | ID of the port of loading | 4 | | Destination\_Port\_ID | int | Not null | ID of the port of discharging | 4 | | Reefer | bool | subject to validation | Possible values:   * Y * N | 1 | | Hazmat | bool | subject to validation | Possible values:   * Y   N | 1 | | Voyage | varchar(6) |  | Voyage number of the vessel | 6 | | Quantity\_and\_Units | int |  | Commodity quantity | 3 | | UOM | char(3) | subject to validation | Unit of measure | 3 | | Value | int |  | Value in CAD | 10 | | TEUS | decimal(2,2) |  | Twenty-foot equivalent unit | 5 | | Container\_Size | char(2) |  | Container size | 2 | | SHIPMENTCOUNT | int |  |  | 2 | | Canada\_Company\_id | int |  | Canada company unique ID (who is receiving import or sending exported goods) | 4 | | Vessel\_id | int |  | Vessel unique ID | 4 | | commodity\_code | int |  | Commodity unique code | 6 | | HS\_Codes\_2\_digit\_id | int |  | HS 2 digits unique code | 2 | | HS\_Codes\_4\_digit\_id | int |  | HS 4 digits unique code | 4 | | Foreign\_Company\_id | int |  | Foreign company (who is receiving export or sending imported goods) | 4 |   Bytes per entry: 120 4.1.3. Table Port  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | id | int | PK | Unique port id | 4 | | Name | varchar(30) |  | Port Name | 30 | | city\_id | int |  | Unique city ID where the port is located | 4 | | country\_code | char(2) |  | Unique country code where the port is located | 2 |   Bytes per entry: 40 4.1.4. Table city  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | id | int | PK | Unique city ID | 4 | | city\_name | varchar(128) |  | City name | 128 | | country\_code | char(2) | subject to validation | Unique country code | 2 | | province\_state\_id | int |  | Unique province or state id | 2 |   Bytes per entry: 136 4.1.5. Table country  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | code | char(2) | PK, subject to validation | Unique 2 letter country code, following ISO 3166-1 alpha-2 [[1]](#footnote-1) | 2 | | country\_name | varchar(128) |  | Country name | 128 |   Bytes per entry: 130 **4.1.6 Table HS\_Codes\_4\_digit**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | code | int | PK | 4 digits harmonized system code [[2]](#footnote-2) | 4 | | description | varchar(64) |  | Item description | 64 |   Bytes per entry: 68 4.1.7 Table commodity  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | code | int | PK | Commodity code following HS code [[3]](#footnote-3) | 6 | | Type | varchar(255) |  | Commodity type | 255 | | Description | varchar(255) |  | Commodity description | 255 |   Byte per entry: 506 4.1.8. Table HS\_Codes\_2\_digit  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | code | int | PK | 2 digits harmonized system code [[4]](#footnote-4) | 2 | | description | varchar(255) |  | Item description | 255 |   Bytes per entry: 257 4.1.9. Table Vessel  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | id | int | PK | Unique vessel id | 4 | | name | varchar(60) |  | Vessel name | 60 | | Shipping\_Line\_id | int |  | Shipping line id | 4 |   Bytes per entry: 68 4.1.10. Table Shipping\_Line  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | id | int | PK | Unique shipping line id | 4 | | name | varchar(60) |  | Shipping line name | 60 |   Bytes per entry: 64 4.1.11 Table province\_state  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Column name | Type | Properties | Description | Bytes | | id | int | PK | State or province ID | 4 | | code | char(2) |  | State or province 2 letter code | 2 | | name | varchar(30) |  | Province or state name | 30 | | country\_code | char(2) |  | Unique country code | 2 |   Bytes per entry: 38 4.2 Data volume The expected data volume for 2019 is 6.5 million transactions, with a growth of 200 thousand transactions per year. In the first 6 years, we expect to store 42 million entries.  The database has 11 tables. In order to estimate the storage needed for 6 years of data, the estimated number of bytes for each entry on each table was calculated. The number of bytes was then multiplied by the number of entries expected for each table. Most of the tables are expected to be relatively static. For those, the number volume was considered static over the years. The table shipments are the one recording the transactions and is estimated to grow over the years. For this table, growth was considered in the dimensioning.  The final figure ins 45Gbytes of storage. A margin of 5% should be added to account for any estimation error, resulting in nearly 50GB needed for storage.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Year | Cumulative Entries (M) from shipment table | Bytes from shipment table | Data volume from other tables (Mbps) | Total data volume [Mbytes] | | 2019 | 6.5 | 120 | 19,909 | 20,689 | | 2020 | 13.2 | 120 | 19,909 | 21,493 | | 2021 | 26.6 | 120 | 19,909 | 23,101 | | 2022 | 53.4 | 120 | 19,909 | 26,317 | | 2023 | 107 | 120 | 19,909 | 32,749 | | 2024 | 214.2 | 120 | 19,909 | 45,613 |  **Performance Monitoring and Database Efficiency**Operational Implications High level, the system will consist of an ODS that provides data to a BI interface. Data transfer and refresh will happen during the night, in the window from 12:00 am to 5:00 am. Maintenance will be scheduled in the same time window, unless it is estimated to take longer than the 5 hours available during the night. In this case, work will be scheduled over the weekend. In case of planned system downtime all users should be informed. Data Transfer Requirements Data will be loaded from csv files daily, during maintenance window. Flat file is estimated to be 2.5Mbps per day, which does not impose a challenge in terms of data transfer. Data Formats Input raw file is in csv format. This data will be loaded to the SQL database as per model previously described.  The user interface will be provided by Tableau. Tableau can be connected directly to the SQL database, so no format conversion is needed.  Appendix A: Acronyms  *Table 1 – Acronyms*   |  |  | | --- | --- | | Acronym | Literal Translation | | ACID | Atomicity, Consistency, Isolation, And Durability | | API | Application Programming Interface | | CAP | Consistency, Availability, Partition Tolerance | | CBSA | Canada Border Services Agency | | CSV | Comma Separated Values | | DBMS | Database Management Systems | | EDI | Electronic Data Interchange | | ERD | Entity-Relationship Diagram | | ETL | Extract Transform Load | | HS Codes | Harmonized System Code | | MS-SQL | Microsoft SQL Server | | ODS | Operational Data Store | | SQL | Structured Query Language | | TCP | Trade Chain Partners | | TEU | Twenty Foot Equivalent Unit | | T-SQL | Transact Structured Query Language | | TTP | Trusted Trader Portal | | UID | Unique Identifier | | UOM | Unit of Measurement |   Appendix B: Scripts |

1. <https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2> [↑](#footnote-ref-1)
2. <https://unstats.un.org/unsd/tradekb/Knowledgebase/50018/Harmonized-Commodity-Description-and-Coding-Systems-HS> [↑](#footnote-ref-2)
3. <https://unstats.un.org/unsd/tradekb/Knowledgebase/50018/Harmonized-Commodity-Description-and-Coding-Systems-HS> [↑](#footnote-ref-3)
4. <https://unstats.un.org/unsd/tradekb/Knowledgebase/50018/Harmonized-Commodity-Description-and-Coding-Systems-HS> [↑](#footnote-ref-4)