Contents

1. SPECIFICATIONS
   - OS: Ubuntu 18.04.06 LTS Bionic
   - Datastore: Google Spreadsheet
   - Automation tool: Notion2Sheets
     - https://workspace.google.com/marketplace/app/notion2sheets/887187948180
   - Makefile: GNU Make 4.1
   - Go: 1.16.5
2. SETUPS AND CONFIGURATIONS

   A. NOTION ENVIRONMENT
   - Update Notion secret key in the script file located in "scripts" folder
   - Execute the script file with command ". setup_notion.sh"

   B. BUILD APP FOR NOTION API

   The app is created with Golang that is able to create database and pages, insert pages, and update pages using http request.
   - The executable can be build in the "scripts" folder using "make build_app"
3. GENERATING DATA

   A. HOW IS DATA GENERATED?
   - The data is imported into Notion directly with the given csv
   - The data can also be populated into Notion with the built app located in "bin" folder
     - Command: ./app -g <page_id>
       - E.g. ./app -g "2697606ce40348569b8a690c16d6f6ef"
4. INSERTING DATA
   - The data can be inserted into Notion with command with app located in "bin" folder
     - ./app -i <database_id> <table_name val> <column_name val> <ordinal_position val> <column_default val> <is_nullable val> <data_type val> <character_maximum_length val>
       - E.g. ./app -i "f9cd42b93c434beba00f97bcbdd20dbe" "my_table" "tmp1" "18" "" "NO" "integer"
       - E.g. ./app -i "f9cd42b93c434beba00f97bcbdd20dbe" "my_table" "tmp1" "18" "" "NO" "character varying" "66"

- The data can also be inserted into Notion with command with app located in “bin” folder using a csv file
  - ./app -i <database_id> <csv_path>
    - E.g. ./app -i “f9cd42b93c434beba00f97bcbdd20dbe” “notion-sync/data/table-definition.csv”
  - Supported features
    - Insert a row
    - Insert multiple rows

5. UPDATING DATA

- The data can be updated in Notion with command with app located in “bin” folder using a csv file
  - csv format
    - csv file to contain the names of the column for editing as first row
    - Page ID to be included as one of the column names as “page_id”
    - Examples of the csv format is stored in “data” folder as “sampleCSV.csv” and “sampleCSV.csv”
  - Command
    - ./app -u <page_id> <csv_path>
      - E.g. ./app -u “notion-sync/data/sampleCSV.csv”

6. STEPS

1. POPULATING DATA WITH TABLE-DEFINITION.CSV

    1. Importing the table-definition.csv file in Notion

    2. Populating the table-definition.csv file by command

        i. Requires

            1. Page_id of the page you populating the data into

        ii. Command

            1. ./app -g <page_id>

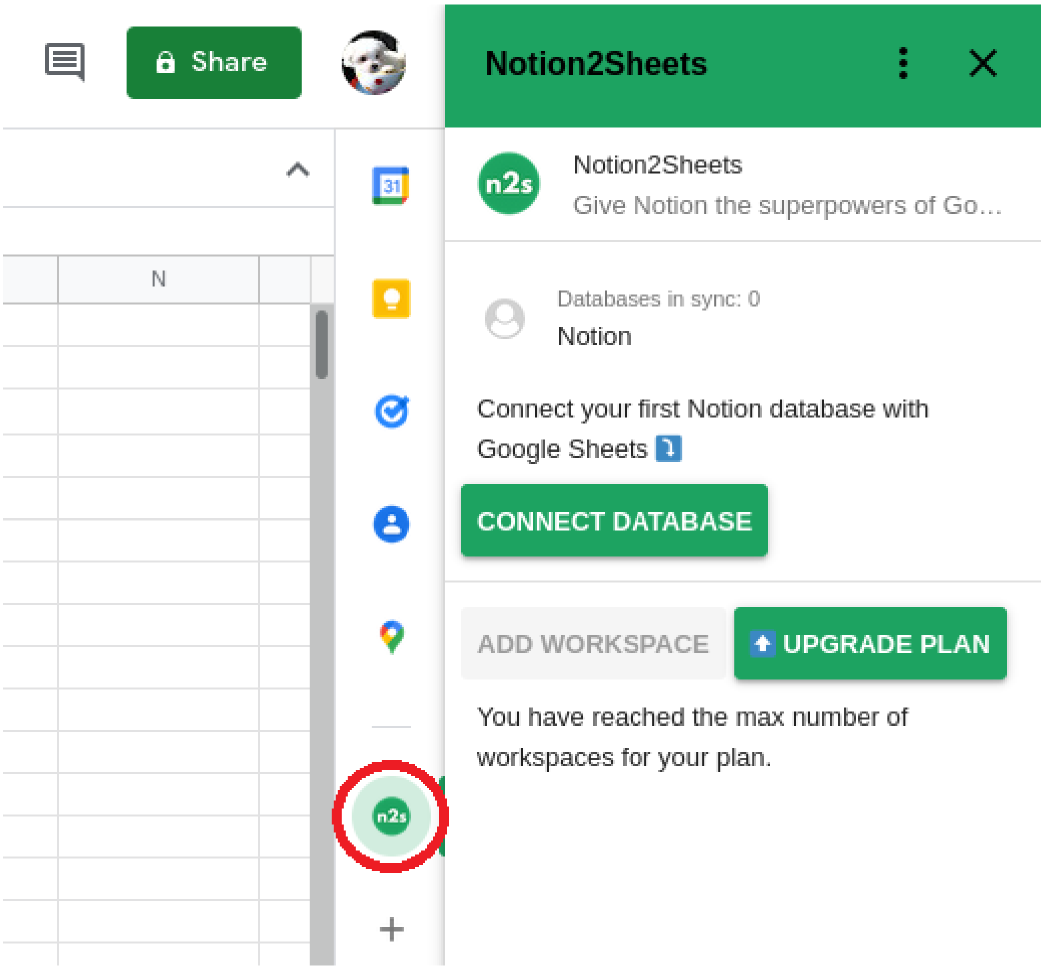
2. INSTALLING APP FOR SYNC PROCESS

- Install “Notion2Sheets” from Google Workspace Marketplace

3. SYNC NOTION TO SHEET

- First create a google spreadsheet

- Click on the Notion2Sheet app on the right sidebar to open the app

○ 

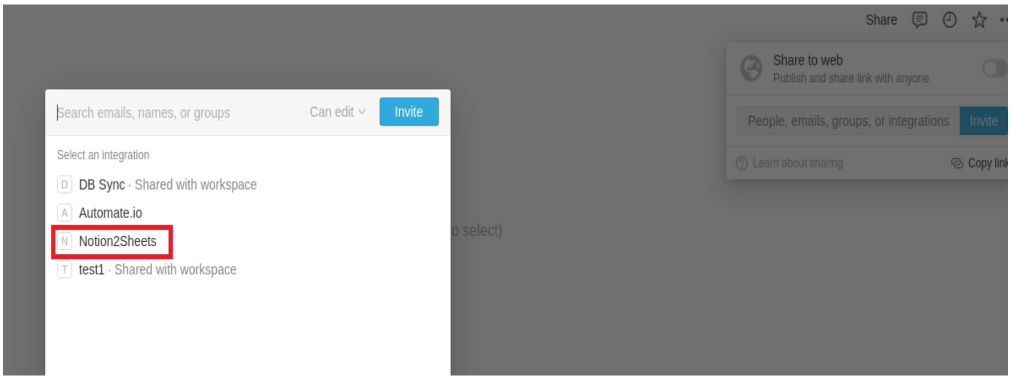
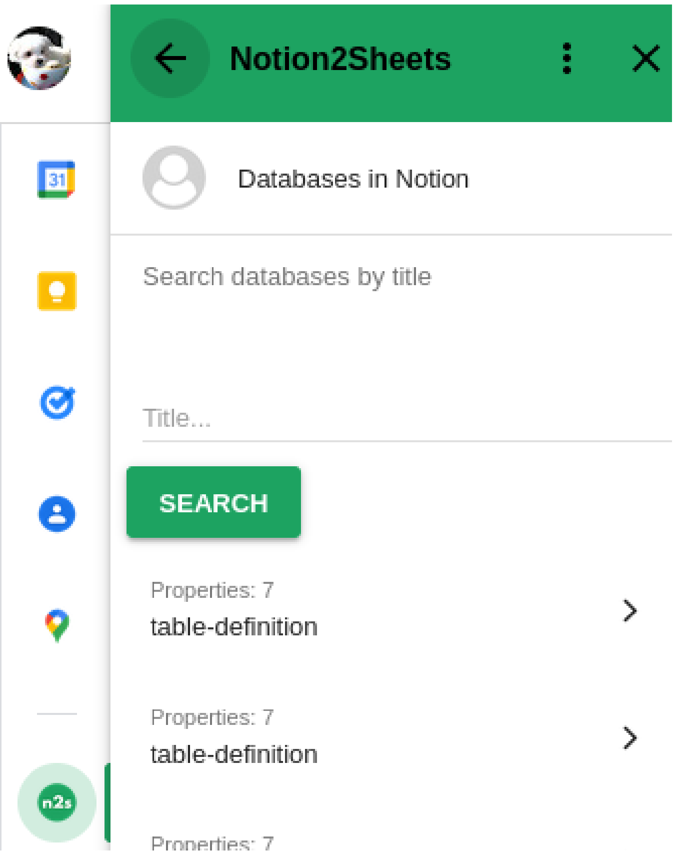
Notion2Sheets
Give Notion the superpowers of Go…

Databases in sync: 0
Notion

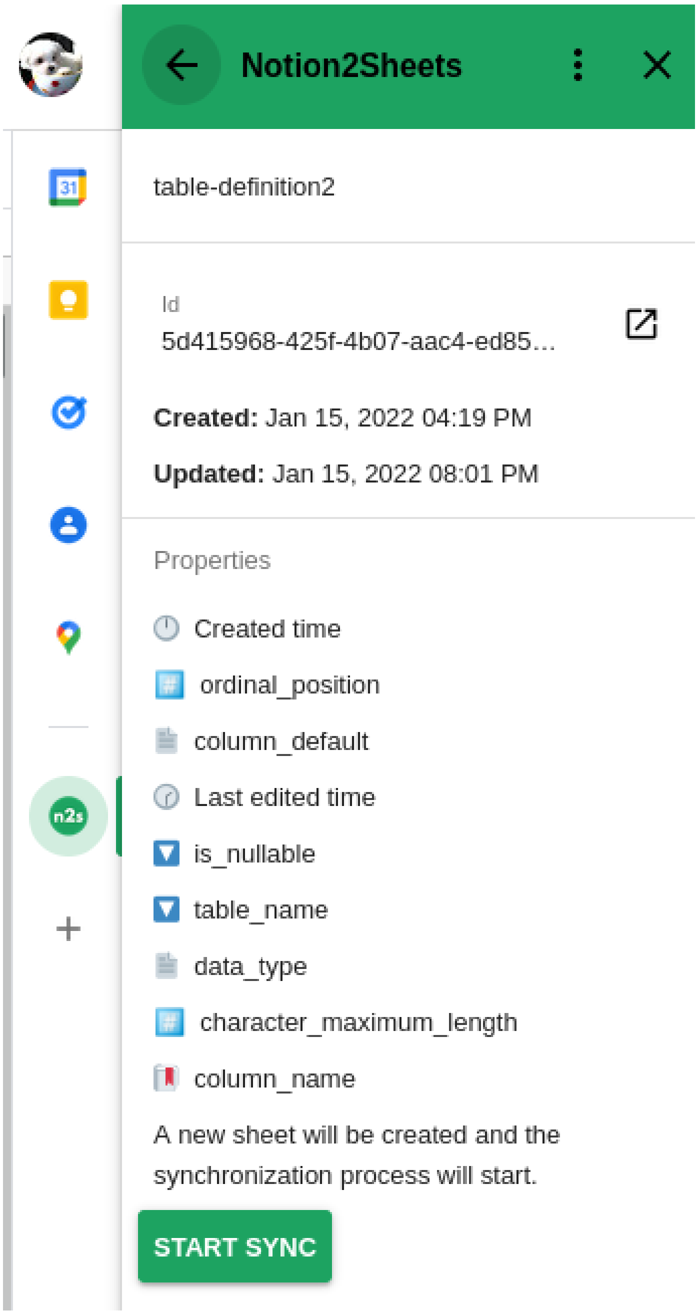
Connect your first Notion database with Google Sheets ⤵

You have reached the max number of workspaces for your plan.

- Link Notion workspace with the app. Notion2Sheets would be automatically integrated to Notion workspace

- Add Notion2Sheet integration to pages
    - 
Share
Share to web
Publish and share link with anyone
People, emails, groups, or integrations
Invite
Learn about sharing
Copy link
Search emails, names, or groups
Can edit
Invite
Select an integration
DB Sync · Shared with workspace
Automate.io
Notion2Sheets
test1 · Shared with workspace
o select)
- Click on "CONNECT DATABASE" on Notion2Sheets app in Google sheet
- Choose any linked pages to sync
    - 

- Press "START SYNC" to start syncing the Notion database into google spreadsheet



Notion2Sheets

table-definition2

Id
5d415968-425f-4b07-aac4-ed85…

Created: Jan 15, 2022 04:19 PM

Updated: Jan 15, 2022 08:01 PM

Properties

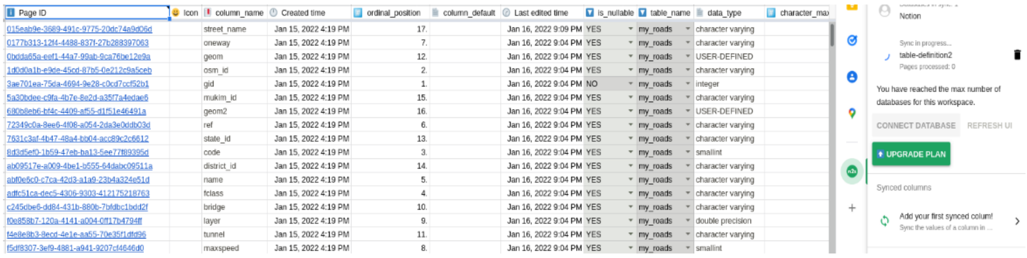
- Created time
- ordinal_position
- column_default
- Last edited time
- is_nullable
- table_name
- data_type
- character_maximum_length
- column_name

A new sheet will be created and the synchronization process will start.

START SYNC

○

- A new sheet would be created



Page ID	Icon	column_name	Created time	ordinal_position	column_default	Last edited time	is_nullable	table_name	data_type	character_max
015eab9e-3689-491c-9775-20dc74a9d06d		street_name	Jan 15, 2022 4:19 PM	17.		Jan 16, 2022 9:09 PM	YES	my_roads	character varying	
0177b313-12f4-4488-837f-27b288397063		oneway	Jan 15, 2022 4:19 PM	7.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
0bdda65a-eef1-44a7-99ab-9ca76be12e9a		geom	Jan 15, 2022 4:19 PM	12.		Jan 16, 2022 9:04 PM	YES	my_roads	USER-DEFINED	
1d0d0a1b-e9de-45cd-87b5-0e212c9a5ceb		osm_id	Jan 15, 2022 4:19 PM	2.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
3ae701ea-75da-4694-9e28-c0cd7ccf52b1		gid	Jan 15, 2022 4:19 PM	1.		Jan 16, 2022 9:04 PM	NO	my_roads	integer	
5a30bdee-c9fa-4b7e-8e2d-a35f7a4edae6		mukim_id	Jan 15, 2022 4:19 PM	15.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
680b8eb6-bf4c-4409-af55-d1f51e46491a		geom2	Jan 15, 2022 4:19 PM	16.		Jan 16, 2022 9:04 PM	YES	my_roads	USER-DEFINED	
72349c0a-8ee6-4f08-a054-2da3e0ddb03d		ref	Jan 15, 2022 4:19 PM	6.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
7631c3af-4b47-48a4-bb04-acc89c2c6612		state_id	Jan 15, 2022 4:19 PM	13.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
8d3d5ef0-1b59-47eb-ba13-5ee77f89395d		code	Jan 15, 2022 4:19 PM	3.		Jan 16, 2022 9:04 PM	YES	my_roads	smallint	
ab09517e-a009-4be1-b555-64dabc09511a		district_id	Jan 15, 2022 4:19 PM	14.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
abf0e6c0-c7ca-42d3-a1a9-23b4a324e51d		name	Jan 15, 2022 4:19 PM	5.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
adfc51ca-dec5-4306-9303-412175218763		fclass	Jan 15, 2022 4:19 PM	4.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
c245dbe6-dd84-431b-880b-7bfdbc1bdd2f		bridge	Jan 15, 2022 4:19 PM	10.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
f0e858b7-120a-4141-a004-0ff17b4794ff		layer	Jan 15, 2022 4:19 PM	9.		Jan 16, 2022 9:04 PM	YES	my_roads	double precision	
f4e8e8b3-8ecd-4e1e-aa55-70e35f1dfd96		tunnel	Jan 15, 2022 4:19 PM	11.		Jan 16, 2022 9:04 PM	YES	my_roads	character varying	
f5df8307-3ef9-4881-a941-9207cf4646d0		maxspeed	Jan 15, 2022 4:19 PM	8.		Jan 16, 2022 9:04 PM	YES	my_roads	smallint	

Databases in sync: 1
Notion

Sync in progress...
table-definition2
Pages processed: 0

You have reached the max number of databases for this workspace.

CONNECT DATABASE REFRESH UI

UPGRADE PLAN

Synced columns

Add your first synced colum!
Sync the values of a column in ...

○

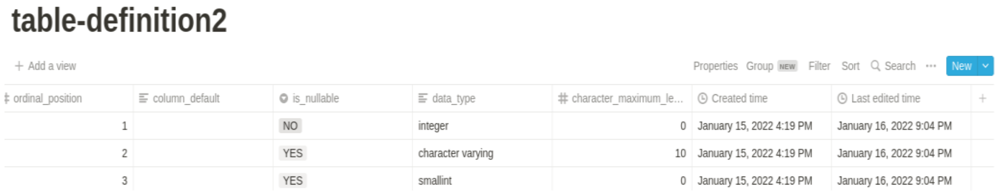
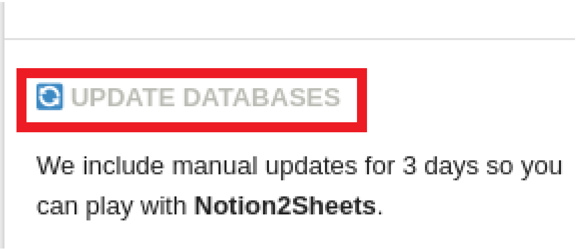
- There would be new columns created that are based on the sync Notion database such as
    - Page ID
    - Created time
    - Last edited time
- The hidden values, created and last edited time, would be added in Notion and shown explicitly, which are not editable
    - 

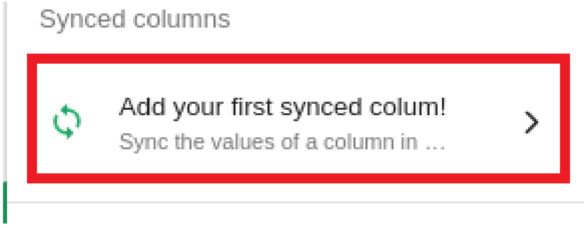
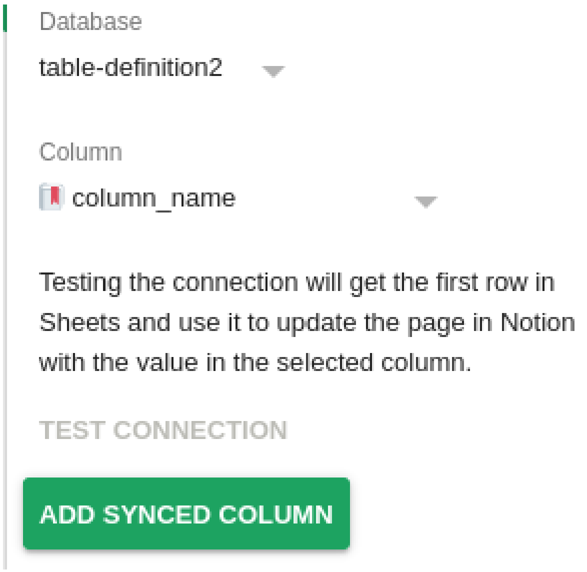
table-definition2

ordinal_position	column_default	is_nullable	data_type	character_maximum_le...	Created time	Last edited time
1		NO	integer	0	January 15, 2022 4:19 PM	January 16, 2022 9:04 PM
2		YES	character varying	10	January 15, 2022 4:19 PM	January 16, 2022 9:04 PM
3		YES	smallint	0	January 15, 2022 4:19 PM	January 16, 2022 9:04 PM

- To manually sync the data from Notion -> Google sheet click on “UPDATE DATABASES” on the Notion2Sheets app on Google spreadsheet
    - 

We include manual updates for 3 days so you can play with **Notion2Sheets**.

4. SYNC GOOGLE SHEET TO NOTION

- To sync the data from Google sheet -> Notion click on “Add your first synced column!” on the Notion2Sheets app on Google spreadsheet
    - 
Synced columns
Add your first synced colum!
Sync the values of a column in …
    - Click on “ADD SYNCED COLUMN”
    - Select the database to sync
    - Select the column to sync
    - Click on “ADD SYNCED COLUMN”
    - 

      Database: table-definition2

      Column: column_name

      Testing the connection will get the first row in Sheets and use it to update the page in Notion with the value in the selected column.

    - To sync other columns, delete the created sync and repeat the steps

5. INSERT ROWS IN NOTION
    a. Run the app in “bin” folder
        - Requires
            - Database ID
            - Csv file or values to insert
        - Command
            - ./app -i <database_id> <csv_path>
            - ./app -i <database_id> <table_name val> <column_name val> <ordinal_position val> <column_default val> <is_nullable val> <data_type val> <character_maximum_length val>
6. UPDATING ROWS IN NOTION

- Run the app in “bin” folder
    - Requires
        - Page ID
        - Formatted csv file
    - Command
        - ./app -u <csv_path>
    - Format of csv in “data” folder, with two sample created, sampleCSV.csv and sampleCSV2.csv

7. UPDATING ROWS IN GOOGLE SHEET

- Can always edit on the spot manually
- Use macros written with Google Apps Script
    - Activated from top tabs “Extentions” -> “Macros” -> update
    - First, enter the name of the sheet to update
    - Second, you can choose to edit the data by row and col
    - Else, when press “cancel”, would be able to search the row to update by page_id
    - Next, would be able to indicate the column name to get the column
        - After pressing ok, the value to edit would be required to input
        - Press “Ok” to continue indicate another column to edit for that row
        - Press “Cancel” to not indicate another column to edit would stop adding field to edit
- Scripts are located in top tabs “Extentions” -> “Apps Script” -> “Code.gs” and also in project folder “src/gs/app.gs”
- Macros are imported with top tab “Extentions” -> “Macros” -> “Import macro” -> “Add Function” for update

7. RESTRICTION

    1. Properties of the database is fixed to the default created properties when importing csv file into Notion
    2. When inserting data into Notion with the app, the order of how the values are given are fixed based on the given table-definition.csv file with the properties fixed from point 1. This is done so as to know which to be inserted with the least arguments needed. For consistency, the order remains even if created with csv file.
        i. E.g. <table_name val> <column_name val> <ordinal_position val> <column_default val> <is_nullable val> <data_type val> <character_maximum_length val>
    3. Delete/archive is not available in the app. Would need to manually delete in Notion page
    4. Due to the restriction of Notion2Sheets app
        i. Only one database is allowed to be sync
        ii. Only can choose to sync Notion -> Sheet or Sheet -> Notion
            1. E.g. If "column_name" is selected to sync in Notion2Sheets app, that column of data would be synced Sheet -> Notion, while the rest is sync Notion -> Sheet
        iii. Only one column is able to sync from Sheet -> Notion due to free version
        iv. Unable to sync new row creation in sheet to notion with Notion2Sheets app
            1. The app depends on page_id, created time, and last edited time to sync the data. Hence, new rows is yet to be supported.

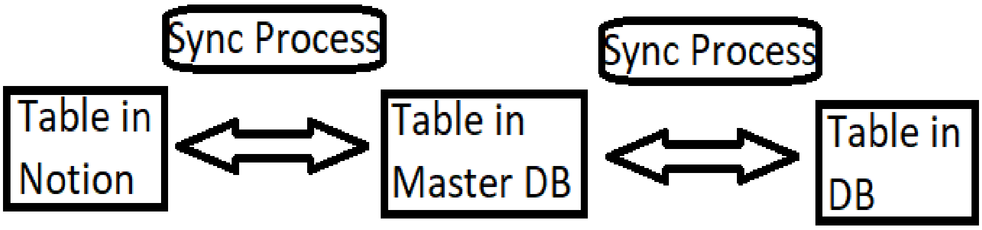
8. CHALLENGES & ISSUES FACED & SOLUTIONS DONE OR POSSIBLE SOLUTIONS

    1. A suitable automation tool to sync Notion <-> DB. More research is required to find the best tool to automate the process using free version, as there are limitations to using free version. For example, unsupported DB for free version.
        i. Possible solution
            1. Personally, I feel that automation tools available for integrating the DB to notion for sync process had a lot of restriction. One possible solution is to build an automation process to sync the data from Notion to DB and vice versa. It would also enable a more complicated and secure sync process to be created to prevent events such as, data loss, unauthorised changes, etc.
    2. The automation tool available are unable to link to a locally deployed datastore.
        i. Possible solution
            1. This would be solved by deploying the datastore on a node with public IP address.
        ii. Solution done
            1. However, for simplicity, google spreadsheet was chosen.
    3. Deploying the datastore on a node with a public IP address or on cloud. There's a limit to how long I could use them for free.

9. IMPROVEMENTS

There are a few possible improvements that could be made in the project.

1. Create a python app to generate json files to create, insert, update data in Notion.
    i. This is harder to perform on typed language Go.
2. Get the properties of the columns and data from the given json file instead on hardcoding the properties and fixing it to one in the app.
3. Develop the automation process from scratch so as to allow more features to be added. In addition, to allow more complicated automation process.

    i. 

    ii. Depending on the sync frequency, a more complicated process could be introduced to prevent data loss or unauthorised changes.
        1. Data could be overwritten with such rule/s
            a. Last edited
            b. User priority
                i. User with higher authority or power could overwrite the data written by a lower authority or power if last edited is around the same timing
            c. Requires authorisation by “admin”
                i. User that has admin rights would receive a report of the changes to be sync into the master DB from both tables. Which could include a comparison of the data. It could also include validation of data to check whether the data is valid and readable to prevent incorrect or corrupted data to be synced.
                ii. The user with admin rights would then have to authorise the syncing to be done.

d. Sync from master DB to two other DB does not require authorisation

4. Add delete function in app.

10. LIST OF COMMANDS

1. SCRIPTS
    - Located in “scripts” folder
        - Building app
            - make build_app
        - Export Notion env
            - . setup_notion.sh
        - Running all test for app
            - make test_all
        - Running test for populating data into Notion
            - make test_generate
        - Running test for inserting data into Notion
            - make test_insert
        - Running test for updating data in Notion
            - make test_update
        - Cleaning up the project
            - make clean
2. APP
    - Located in “bin” folder
        - Create database (create the default table-definition.csv)
            - ./app -g <page_id>
        - Insert pages
            - ./app -i <database_id> <csv_path>
            - ./app -i <table_name val> <column_name val> <ordinal_position val> <column_default val> <is_nullable val> <data_type val> <character_maximum_length val>
        - Update pages
            - ./app -u <csv_path>