Homes.co.nz Technical Test

Project

We have a property rates file, that looks like:

ID,Street address,Town,Valuation date,Value

1,1 Northburn RD,WANAKA,1/01/15,280000

2,1 Mount Ida PL,WANAKA,1/01/15,280000

3,1 Mount Linton AVE,WANAKA,1/01/15,780000

4,1 Kamahi ST,WANAKA,1/01/15,155000

1,1 Northburn RD,WANAKA,1/01/16,380000

2,1 Mount Ida PL,WANAKA,1/01/16,285000

3,1 Mount Linton AVE,WANAKA,1/01/16,880000

4,1 Kamahi ST,WANAKA,1/01/16,185000

…

i.e., multiple rating valuations for a property for different years.

Unfortunately, the file is not super clean, so every now and again, we get duplicates:

ID,Street address,Town,Valuation date,Value

1,1 Northburn RD,WANAKA,1/01/15,280000

2,1 Mount Ida PL,WANAKA,1/01/15,280000

3,1 Mount Linton AVE,WANAKA,1/01/15,780000

4,1 Kamahi ST,WANAKA,1/01/15,155000

**1,1 Northburn RD,WANAKA,1/01/16,380000**

**2,1 Mount Ida PL,WANAKA,1/01/16,285000**

3,1 Mount Linton AVE,WANAKA,1/01/16,880000

4,1 Kamahi ST,WANAKA,1/01/16,185000

**1,1 Northburn RD,WANAKA,1/01/16,386000**

**2,1 Mount Ida PL,WANAKA,1/01/16,282000**

…

In some cases, we can make a decision and say that because the amounts are the same, we can use the row, and insert one record and ignore the other (eg: duplicate with propertyID =1)

Please complete this project in Go ([https://golang.org](https://golang.org/)), even if you’ve never used it before.

**Test #1**

Write a routine that implements the logic described above. Use the attached properties.txt file as input.

In the case of duplicates, use the last encountered record.

NOTE: A duplicate is a row that has the same address and same date. The ID is irrelevant.

Print the list.

**Test #2**

Modify the code in case of duplicates to use the first encountered record.

**Test #3**

Instead of inserting the last record, make sure that no duplicates are entered at all. i.e., if there are duplicate records, do not insert any of the duplicate records. Continue to send all non-duplicates as per normal.

Print the list.

**Test #4**

Modify the codebase to run the following filters:

* Filter out cheap properties (anything under 400k)
* Filter out properties that are avenues, crescents, or places (AVE, CRES, PL) cos those guys are just pretentious...
* Filter out every 10th property (to keep our users on their toes!)

Print the list.

**Test #4 (extra credit – highly recommended)**

Split the validated (non-duplicate) records into chunks, and 'process' each chunk via its own goroutine. That is, run all filters on each chunk, then combine the filtered results of each chunk back into a single list.