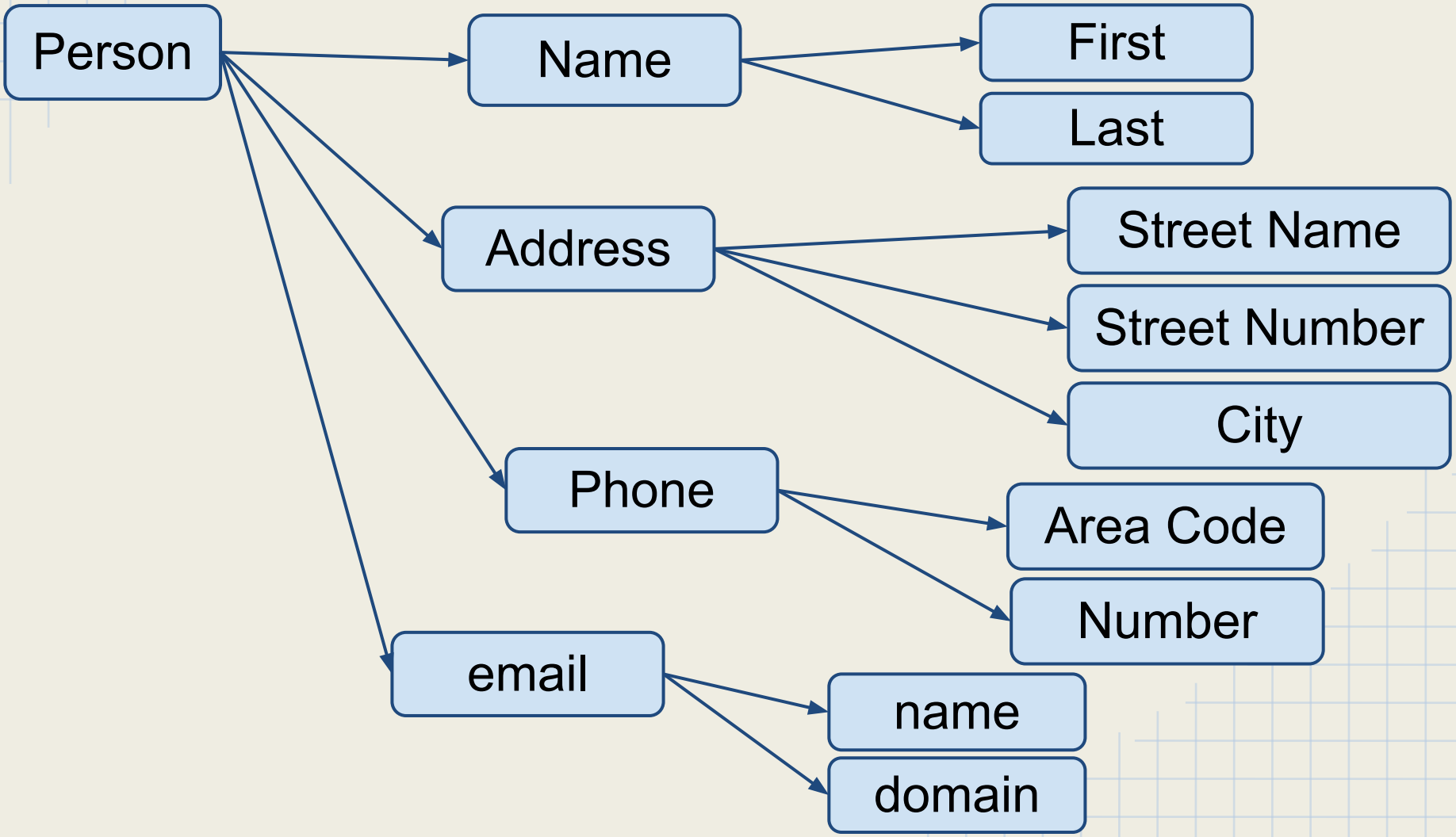
M Database

Dima Kassab & Luis Ibanez

SUNY-Albany 2013

Trees are Everything !

People are Trees Too !



Trees are Flexible

Sparse Data

Relational Limits

Given Enough Time,

all Fields are **Optional**

Relational Limits

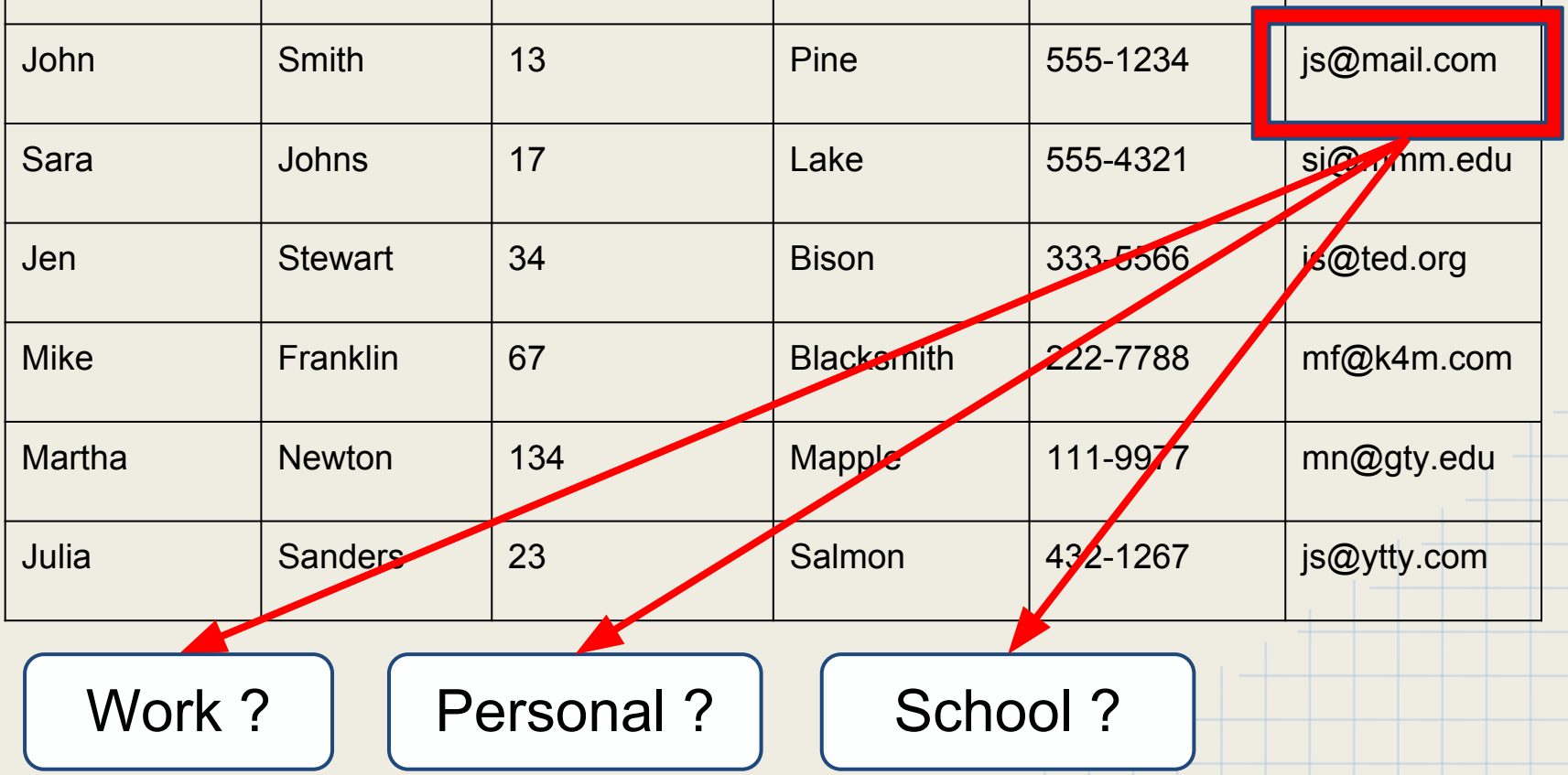
First	Last	Street Number	Street Name	Phone	email
John	Smith	13	Pine	555-1234	js@mail.com
Sara	Johns	17	Lake	555-4321	sj@mmm.edu
Jen	Stewart	34	Bison	333-5566	js@ted.org
Mike	Franklin	67	Blacksmith	222-7788	mf@k4m.com
Martha	Newton	134	Mapple	111-9977	mn@gty.edu
Julia	Sanders	23	Salmon	432-1267	js@ytty.com

Relational Limits

First	Last	Street Number	Street Name	Phone	email
John	Smith	13	Pine	555-1234	js@mail.com
Sara	Johns	17	Lake	555-4321	sj@mmm.edu
Jen	Stewart	34	Bison	333-5566	js@ted.org
Mike	Franklin	67	Blacksmith	222-7788	mf@k4m.com
Martha	Newton	134	Mapple	111-9977	mn@gty.edu
Julia	Sanders	23	Salmon	432-1267	js@ytty.com

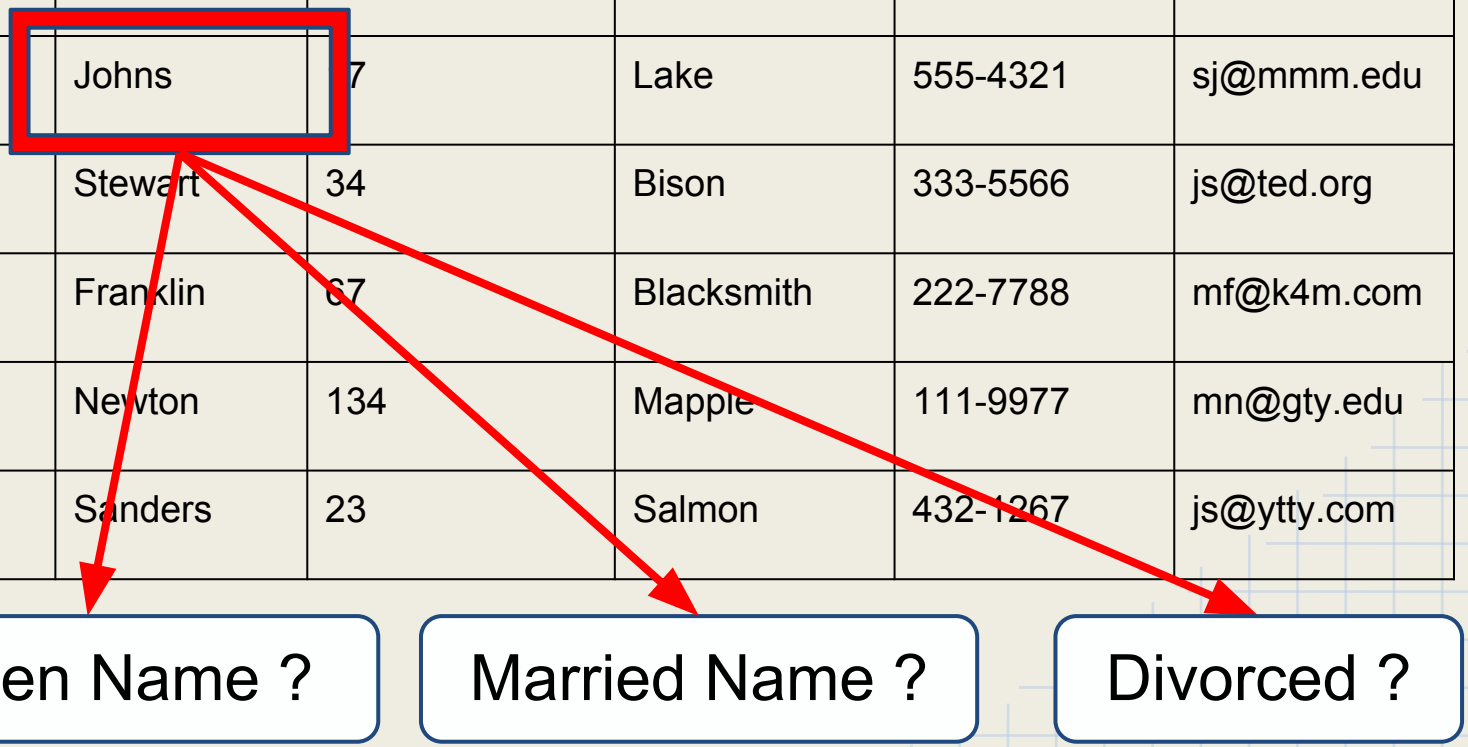
Relational Limits

First	Last	Street Number	Street Name	Phone	email
John	Smith	13	Pine	555-1234	js@mail.com
Sara	Johns	17	Lake	555-4321	si@[illegible]mm.edu
Jen	Stewart	34	Bison	333-5566	js@ted.org
Mike	Franklin	67	Blacksmith	222-7788	mf@k4m.com
Martha	Newton	134	Mapple	111-9977	mn@gty.edu
Julia	Sanders	23	Salmon	432-1267	js@ytty.com



Relational Limits

First	Last	Street Number	Street Name	Phone	email
John	Smith	13	Pine	555-1234	js@mail.com
Sara	Johns	7	Lake	555-4321	sj@mmm.edu
Jen	Stewart	34	Bison	333-5566	js@ted.org
Mike	Franklin	67	Blacksmith	222-7788	mf@k4m.com
Martha	Newton	134	Mapple	111-9977	mn@gty.edu
Julia	Sanders	23	Salmon	432-1267	js@ytty.com



Relational Limits

Given Enough Time,

all Fields are **Optional**

Hierarchical Databases

Things are Different in a world of Trees

Hierarchical Structures

Only those fields
with data
are created

Multi-Dimensional Arrays

```
set ^person("name","first")="John"
set ^person("name","last")="Smith"
set ^person("address","street","name")="Pine"
set ^person("address","street","number")="13"
set ^person("address","city")="Boston"
set ^person("phone","area code")="555"
set ^person("phone","number")="1234"
set ^person("email","name")="johnsmith"
set ^person("email","domain")="gmail.com"
```

Hands On.

Let's do This !

Join the Party

Log into
the Server

Join the Party

Click Here if INF-202

Click Here if INF-362

Join the Party - INF 202

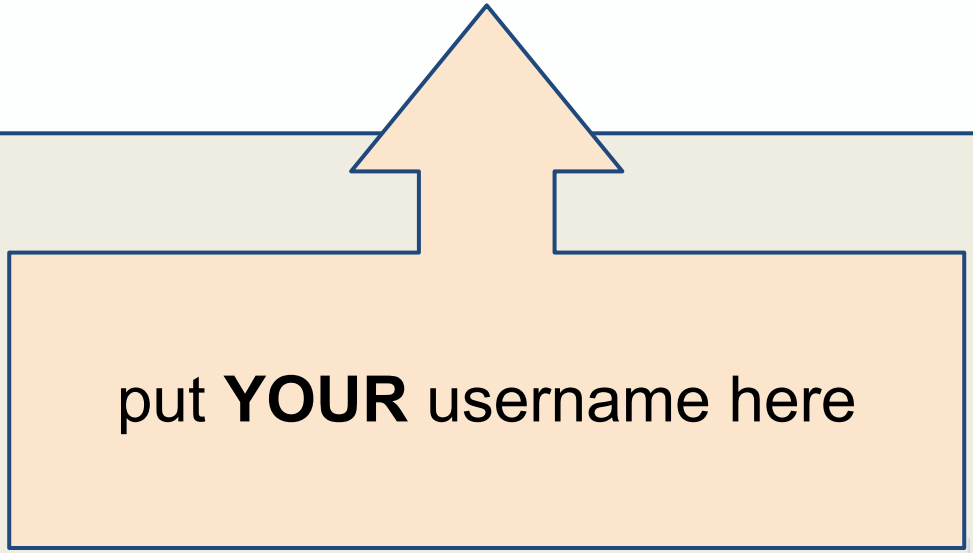
```
ssh-add -K ~/.ssh/inf_202_spring_2013_student_key
```

Open your SSH Key

chocolate

Join the Party - INF 202

```
ssh  -p 9744  yourusername@54.225.78.7
```



Once in the Server...

Continue here

Once you log
into the Server

Set up your Environment

Using favorite text editor (Vim or Nano), open your file:

```
vim    ~/.bashrc
```

Go to the end of the file, and add the line:

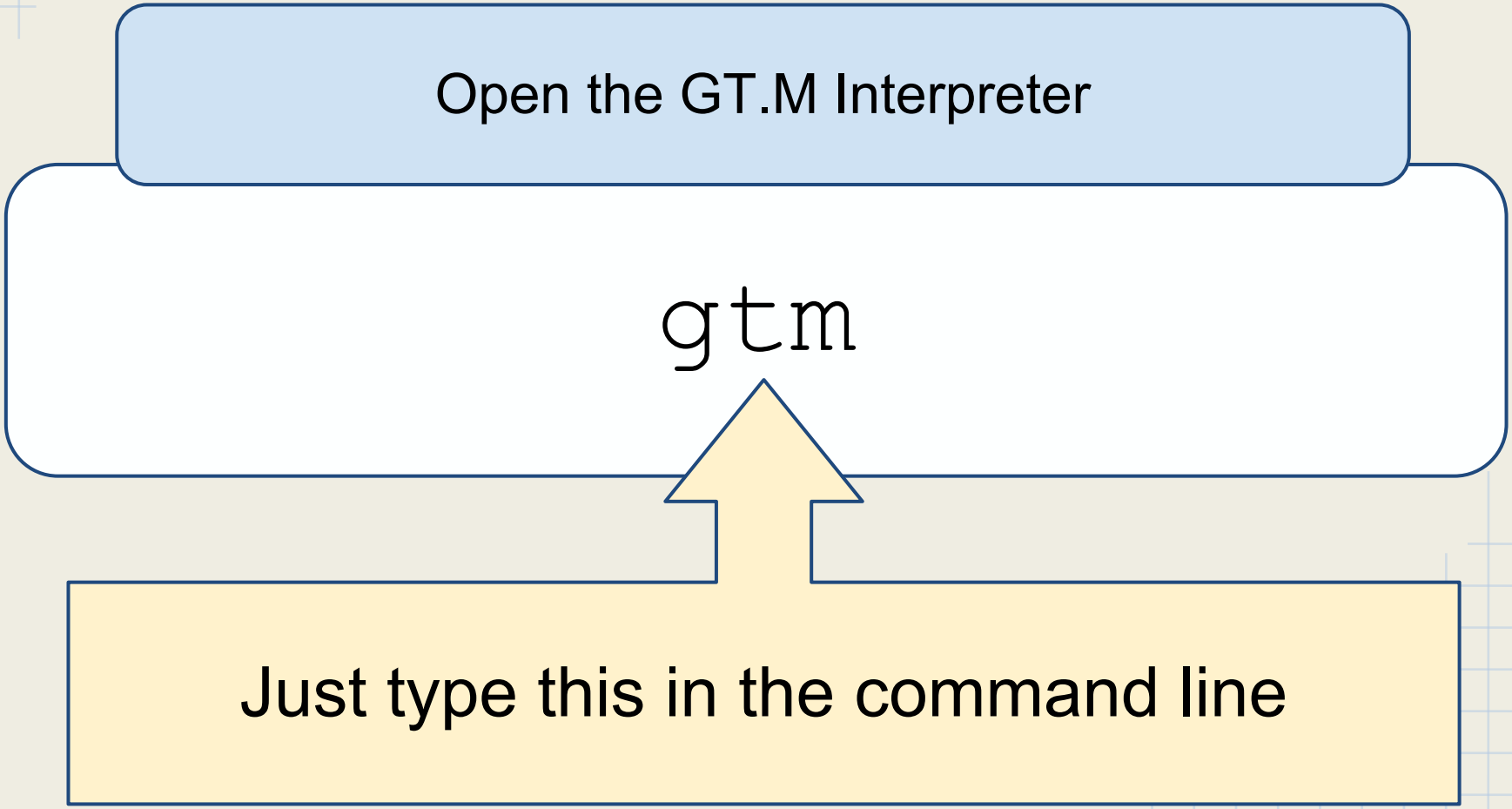
```
source /INF362-EWD/gtm/setup/add_to_bashrc.txt
```

save the file, quit the text editor

and from the command line do:

```
source ~/.bashrc
```

GT.M is Open Source



Look at the Data in a Tree

Look at the "^person" tree

```
GTM>zwrite ^person(*)
```

This is the GT. M Prompt

One and Only **one** Space

Insert Data

Add yourself to the "^person" tree

```
GTM>set ^person("username")=""
```

This is the GT.M Prompt

One and only **ONE** space

YOUR username

Insert Data

Add yourself to the "^person" tree

```
GTM>set ^person("username")=""
```

By Assigning an **Empty** string
we create a node in the tree

Insert Data

Add your Name : First, Last

```
GTM>set ^person("username","name","first")="Luis"
GTM>set ^person("username","name","last")="Ibanez"
```

YOUR username

User **YOUR** name here

Watch the Tree Grow !

Look at the "^person" tree

```
GTM>zwrite ^person(*)
```

Watch the Tree Grow !

Look at your portion of the "^person" tree

```
GTM>zwrite ^person("username",*)
```

YOUR username

Must Have more Data !

Add your Favorite Food Items

```
GTM>set ^person("username","food","apple")=""
GTM>set ^person("username","food","broccoli")=""
GTM>set ^person("username","food","chocolate")=""
```

YOUR username

Must Have more Data !

Add your Favorite Food Items

```
GTM>set ^person("username","food","apple")=""
GTM>set ^person("username","food","broccoli")=""
GTM>set ^person("username","food","chocolate")=""
```

Multiple Values !!

Terminal Nodes

When the Value is **Unique**: use the End

```
GTM>set ^person("username","name","first")="Luis"
```

When the Value is **Multiple**: use the Index

```
GTM>set ^person("username","food","apple")=""
```

Terminal Nodes

Seriously !

This is important !

When the Value is **Unique**: use the End

When the Value is **Multiple**: use the Index

Watch the Tree Grow !

Look at the "^person" tree

```
GTM>zwrite ^person(*)
```

Your Turn !

Practice

Your Turn !

You like various
types of Apples
(red, yellow, green)

How would you enter them
in the database ?

Taking over the World ! (202)

Assigning Continents

```
GTM>set ^place("asia")="Informative Revolver"
GTM>set ^place("europe")="Starving Students"
GTM>set ^place("africa")="Stacked like Pancakes"
GTM>set ^place("america")="Mammoth Number 9"
```

Each TBL Team : Create your Continent

Taking over the World ! (362)

Assigning Continents

```
GTM>set ^place("asia")="Amazing As"
GTM>set ^place("europe")="Mellow United Nations"
GTM>set ^place("africa")="PRO 2013"
GTM>set ^place("america")="Yummy Tacos"
GTM>set ^place("australia")=""
```

Each TBL Team : Create your Continent

Taking over the World !

Creating and Assigning Countries

```
GTM>set ^place("asia","japan")="Steve"
GTM>set ^place("asia","korea")="Mary"
GTM>set ^place("asia","vietnam")="John"
```

Each Student in the TBL Team:
Create your Country in that Continent

Taking over the World !

Creating Cities

```
GTM>set ^place("asia","japan","tokyo")=""
GTM>set ^place("asia","japan","kyoto")=""
GTM>set ^place("asia","japan","osaka")=""
```

Each Student in the TBL Team:
Create **FIVE** Cities in that Country

Taking over the World !

Tracking Zombies

```
GTM>set ^place("asia","japan","tokyo","human")=5000
GTM>set ^place("asia","japan","tokyo","zombie")=400
```

Each Student in the TBL Team:
Assign an estimated **Human** and **Zombie**
Population to her/his Cities

Watch the Zoombies !

Look at the "^place" tree

```
GTM>zwrite ^place(*)
```

Watch the Zombies !

Look at the "^place" sub-trees

```
GTM>zwrite ^place("asia",*)
```

```
GTM>zwrite ^place("asia","china",*)
```

Look at the sub-tree of **YOUR** country entry

Walking in a Tree

Visiting
the Data

Taking over the World !

Start with one step

```
GTM>set i=$order(^place(""))
GTM>write i
GTM>set i=$order(^place(i)) write i
GTM>set i=$order(^place(i)) write i
GTM>set i=$order(^place(i)) write i
```

The **$order()** function returns
the next value of a given index

Your Turn

Write the expressions that will walk down the **Countries** of your **Continent**

Write the expressions that will walk down the **Cities** of your **Country**

Repetition is Boring

Tired of Typing ?

It's time to master the:

for loop

Taking over the World !

Start with initializing the index: **i**

```
GTM>set i=""
GTM>for  set i=$order(^place(i)) write i,! quit:i=""
```

TWO Spaces

ONE Space

ONE Space

ONE Space

Taking over the World !

What have we done !?

```
GTM>set i=""
GTM>for  set i=$order(^place(i)) write i,! quit:i=""
```

This means: **Forever !**

This gets the next index

This means: Quit if `(i="")`

LOL BRB

Why to type words when we can just do letters...

```
GTM>s i=""
GTM>f  s i=$o(^place(i)) w i,! q:i=""
```

This means:
Forever !

This gets the next index

This means:
Quit if `(i="")`

Be the One !

Why to use two lines, when we can do ONE !

```
GTM>s i="" f  s i=$o(^place(i)) w i,! q:i=""
```

Your Turn

Write a **FOR** loop that lists
all the **Countries** in your **Continent**

```
HINT:     write $order(^place("asia",""))
```

Empty
Index

Your Turn

Write a **FOR** loop that lists
all the **Cities** in your **Country**

```
HINT:   write $order(^place("asia","china",""))
```

Empty
Index

Welcome to M !

The END

for now...