



DATABASE NORMALIZATION?



imgflip.com



Why normalize database?

- Remove data redundancy...
- ...and use less storage
- More extensibility without changing existing schema
- Easier to deploy database changes without downtime
- Backwards compatibility!
- Tables are smaller and more readable
- More data integrity and consistency



CHUCK NORRIS APPROVES



DATABASE NORMALIZATION

B

A man in a dark blue suit, white shirt, and red tie is speaking at a podium. He has his eyes closed and a slight smile. The background is a light blue wall. The word "BUT" is written in large, white, bold, sans-serif capital letters with a black outline in the upper right corner of the image.

BUT



B

We don't normalize databases because...

- It's too much work / I don't have time for this
- "I'm not DBA"
- I'll need shitload of JOINS, querying is harder and queries are more difficult to read
- Storage is cheap
- It's harder to maintain performance





B



B

LET'S CREATE



A PERSON

Person Entity

- Id
- Name
- Last name



Person Entity

- **Id - unique**
- **Name – can repeat itself**
- **Last name – can repeat itself**



Person Entity

- Id
- Name – **separate lookup table**
- Last name – **separate lookup table**



```
CREATE TABLE [dbo].[Name] (  
    [Id] int NOT NULL PRIMARY KEY IDENTITY(1,1),  
    [Value] nvarchar(100) NOT NULL  
);
```

```
CREATE TABLE [dbo].[LastName] (  
    [Id] int NOT NULL PRIMARY KEY IDENTITY(1,1),  
    [Value] nvarchar(100) NOT NULL  
);
```



```
CREATE TABLE [dbo].[Person] (  
    [Id] int NOT NULL PRIMARY KEY IDENTITY(1,1),  
    [NameId] int NOT NULL,  
    [LastNameId] int NOT NULL,  
  
    CONSTRAINT FK_Person_Name FOREIGN KEY ([NameId])  
        REFERENCES [dbo].[Name] ([Id])  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    CONSTRAINT FK_Person_LastName FOREIGN KEY ([LastNameId])  
        REFERENCES [dbo].[LastName] ([Id])  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION  
  
);
```

```
INSERT INTO [dbo].[Name] ([Value])  
VALUES ('John'), ('Mark'), ('Jim'), ('Gary'), ('Bill');
```

```
INSERT INTO [dbo].[LastName] ([Value])  
VALUES ('Sample'), ('Smith'), ('Test'), ('Doe'), ('Dupa');
```

```
INSERT INTO [dbo].[Person] ([NameId], [LastNameId])  
SELECT n.[Id], ln.[Id]  
FROM [dbo].[Name] n  
CROSS JOIN [dbo].[LastName] ln;
```

```
SELECT *  
FROM [dbo].[Person]
```

Id	NameId	LastNameId
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	1	2
7	2	2
8	3	2
9	4	2
10	5	2
11	1	3
12	2	3
13	3	3
14	4	3
15	5	3
16	1	4

**Something went wrong with the
summoning!**

**Are you sure you weren't holding the
spell upside down?**



```
SELECT p.[Id] Id,  
       n.[Value] Name,  
       ln.[Value] LastName  
FROM [dbo].[Person] p  
JOIN [dbo].[Name] n  
     ON p.[NameId] = n.[Id]  
JOIN [dbo].[LastName] ln  
     ON p.[LastNameId] = ln.[Id]
```

Id	Name	LastName
1	John	Sample
2	Mark	Sample
3	Jim	Sample
4	Gary	Sample
5	Bill	Sample
6	John	Smith
7	Mark	Smith
8	Jim	Smith
9	Gary	Smith
10	Bill	Smith
11	John	Test
12	Mark	Test
13	Jim	Test
14	Gary	Test
15	Bill	Test
16	John	Doe
17	Mark	Doe




```
CREATE VIEW [dbo].[Vw_Person]
AS
SELECT p.[Id] Id,
       n.[Value] Name,
       ln.[Value] LastName
FROM [dbo].[Person] p
JOIN [dbo].[Name] n
     ON p.[NameId] = n.[Id]
JOIN [dbo].[LastName] ln
     ON p.[LastNameId] = ln.[Id]
```



```
SELECT *  
FROM [dbo].[Vw_Person]
```

Id	Name	LastName
1	John	Sample
2	Mark	Sample
3	Jim	Sample
4	Gary	Sample
5	Bill	Sample
6	John	Smith
7	Mark	Smith
8	Jim	Smith
9	Gary	Smith
10	Bill	Smith
11	John	Test
12	Mark	Test
13	Jim	Test
14	Gary	Test
15	Bill	Test
16	John	Doe
17	Mark	Doe

EVERYTHING ALRIGHT,



DAWG?



OH CRAP



INSERTS AND UPDATES

```
INSERT INTO [dbo].[Vw_Person]
    ([Name], [LastName])
VALUES
    (N'Rafał', 'Hryniewski')
```

View or function 'dbo.Vw_Person' is not updatable because the modification affects multiple base tables.




```
UPDATE [dbo].[Vw_Person]
SET [LastName] = 'Grey'
WHERE [Id] = 1
```

(1 row affected)



**IF TOO GOOD TO BE TRUE,
SOMETHING SOUNDS...**



**TOO GOOD TO BE TRUE,
SOMETHING IS...**



```
SELECT *  
FROM [dbo].[Vw_Person]
```

Id	Name	LastName
1	John	Sample
2	Mark	Sample
3	Jim	Sample
4	Gary	Sample
5	Bill	Sample
6	John	Smith
7	Mark	Smith
8	Jim	Smith
9	Gary	Smith
10	Bill	Smith
11	John	Test
12	Mark	Test
13	Jim	Test
14	Gary	Test
15	Bill	Test
16	John	Doe
17	Mark	Doe

	Id	Name	LastName
1	1	John	Grey
2	2	Mark	Grey
3	3	Jim	Grey
4	4	Gary	Grey
5	5	Bill	Grey
6	6	John	Smith
7	7	Mark	Smith
8	8	Jim	Smith
9	9	Gary	Smith
10	10	Bill	Smith
11	11	John	Test
12	12	Mark	Test
13	13	Jim	Test
14	14	Gary	Test
15	15	Bill	Test
16	16	John	Doe
17	17	Mark	Doe

I CAN FIX IT



```
CREATE OR ALTER TRIGGER [TR_Vw_Person_InsteadOfInsert]
ON [dbo].[Vw_Person]
INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;
    MERGE [dbo].[Name] n
    USING inserted i
        ON n.[Value] = i.[Name]
    WHEN NOT MATCHED BY TARGET
    THEN
        INSERT ([Value]) VALUES (i.[Name]);

    MERGE [dbo].[LastName] n
    USING inserted i
        ON n.[Value] = i.[LastName]
    WHEN NOT MATCHED BY TARGET
    THEN
        INSERT ([Value]) VALUES (i.[LastName]);

    INSERT INTO [dbo].[Person] ([NameId],[LastNameId])
    SELECT n.[Id], ln.[Id]
    FROM inserted i
    JOIN [dbo].[Name] n
        ON n.[Value] = i.[Name]
    JOIN [dbo].[LastName] ln
        ON ln.[Value] = i.[LastName]
END
```




```
CREATE OR ALTER TRIGGER [TR_Vw_Person_InsteadOfUpdate]
ON [dbo].[Vw_Person]
INSTEAD OF UPDATE
AS
BEGIN
    --BLAH BLAH BLAH
    --HATE NORMALIZATION
    --THIS IS FUCKED UP
END
```

THERE I FIXED IT



**NOW IF YOU WOULD
ADD DATE OF BIRTH**

THAT WOULD BE GREAT



FUCK THIS



5 MAN-DAYS



A close-up, high-angle shot of Morpheus from the movie The Matrix. He is bald, has a serious expression, and is wearing his signature black sunglasses. The background is a blurred, dimly lit interior. The text "WHAT IF I TOLD YOU" is superimposed at the top in a bold, white, sans-serif font with a black outline.

WHAT IF I TOLD YOU

IT DOESN'T HAVE TO BE LIKE THIS

Anchor Modeling

maintain highly normalized data model that
can be changed anytime



Rafał Hryniewski

Senior .NET Dev

Blogger

Speaker

Community leader

<https://hryniewski.net>

rafal@hryniewski.net



@r_hryniewski



fb.me/hryniewskinet



devstyle.pl

BLOG DLA KAŻDEGO PROGRAMISTY



BIĄŁOSTOCKA GRUPA .NET



Microsoft Azure
User Group Poland



programistok

A MARVEL COMICS EVENT IN SEVEN PARTS

CIVIL WAR



WHOSE SIDE ARE YOU ON?



Agenda

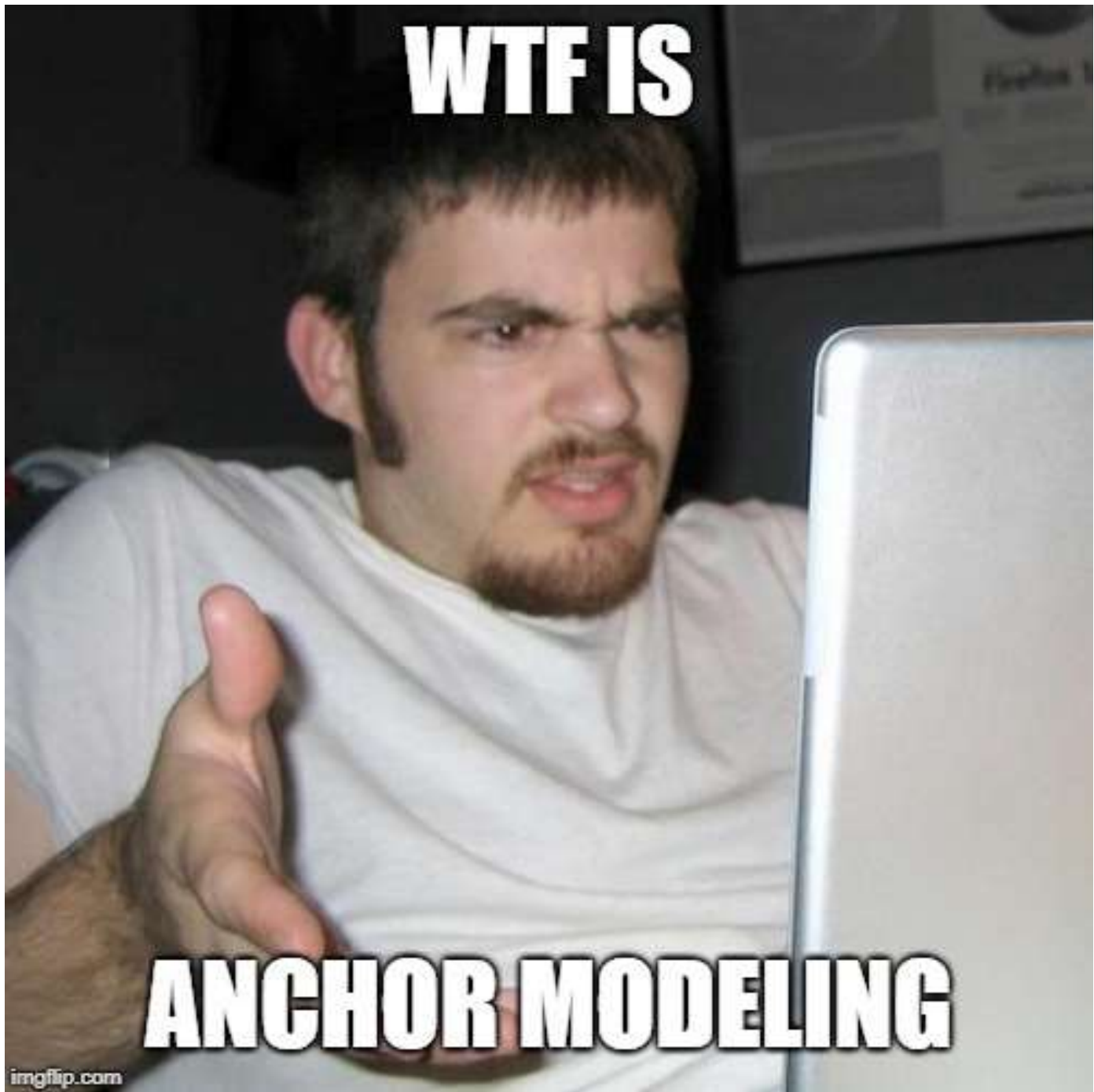
- Evolving highly normalized schema in traditional way.
- Glossary.
- How does it look?
- Anchors, ties and other modeling constructs.
- Features



Glossary

- Normalization – structuring relational database to minimize data redundancy and improve data integrity
- Temporal Data – data with relation to time in terms of validity time, transaction time and/or decision time





B

Anchor Modeling is

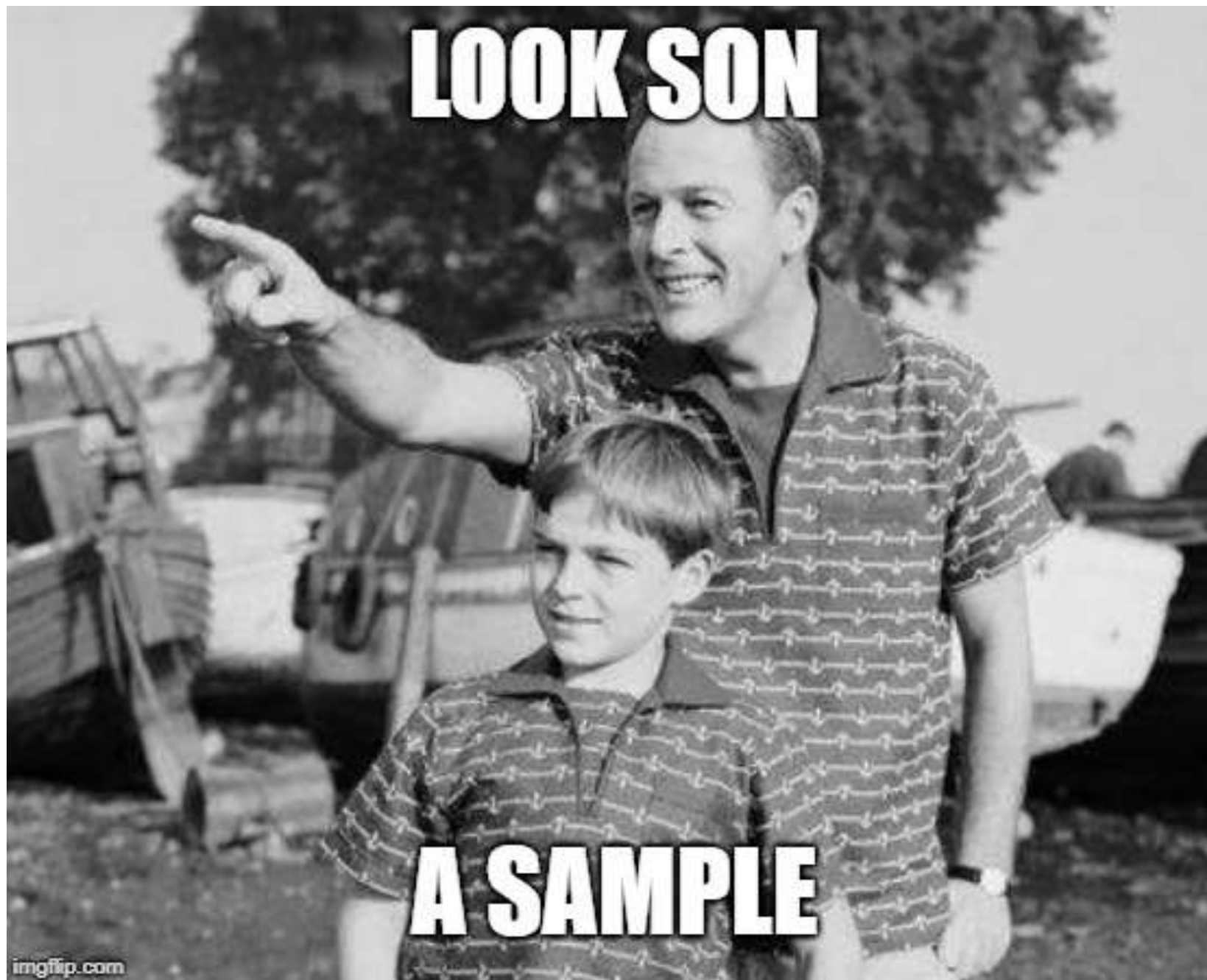
- Database modeling technique
- Agile
- Can work with temporal data
- Easy
- Has graphic UI modeling tool
- Uses four modeling constructs: anchors, attributes, ties and knots



Databases created with Anchor Modeling are

- Mostly in 6th normal form
- Easy to expand
- Temporal (if you want it to be)
- Ready to work with in no time
- Can be deployed with zero downtime





LOOK SON

A SAMPLE

imgflip.com

B

Requirements

- 3 entities – Person, Company, Post
- Person should have **Name**, **Last Name** and **Date of Birth**
- Company should have **Name**
- Post should have **Title**, **Content** and **Creation Date**



Requirements

- 3 entities – Person, Company, Post
- **Person** should be able to be **friend** with other **Person**
- **Person** should be able to be **employed** in **Company**
- **Person** and **Company** are able to create **Posts**



IN 6TH NORMAL FORM



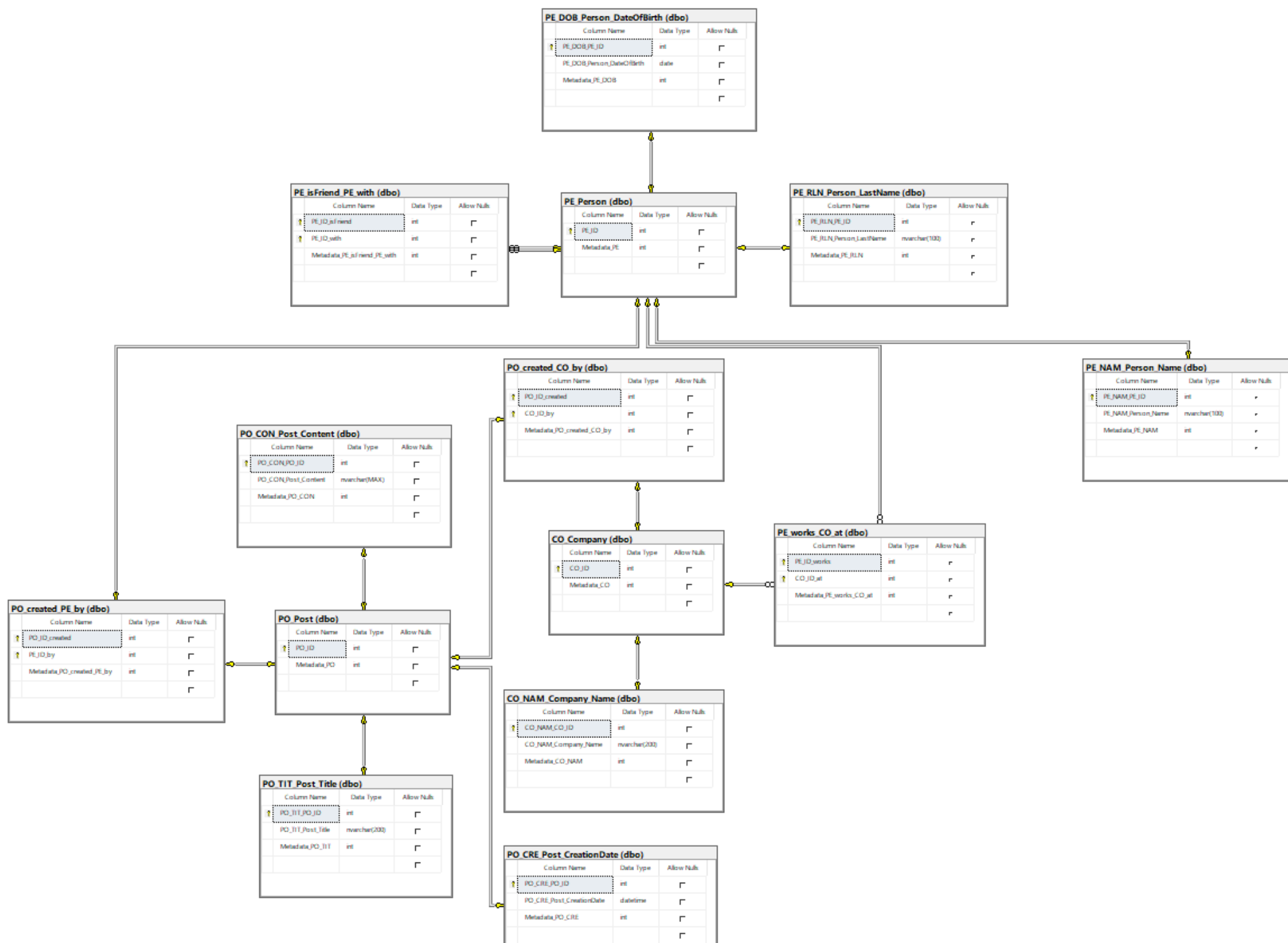
imgflip.com





imgflip.com





PE_DOB_Person_DateOfBirth (dbo)

Column Name	Data Type	Allow Nulls
PE_DOB_PE_ID	int	<input type="checkbox"/>
PE_DOB_Person_DateOfBirth	date	<input type="checkbox"/>
Metadata_PE_DOB	int	<input type="checkbox"/>

PE_NAM_Person_Name (dbo)

Column Name	Data Type	Allow Nulls
PE_NAM_PE_ID	int	<input type="checkbox"/>
PE_NAM_Person_Name	nvarchar(100)	<input type="checkbox"/>
Metadata_PE_NAM	int	<input type="checkbox"/>

PE_isFriend_PE_with (dbo)

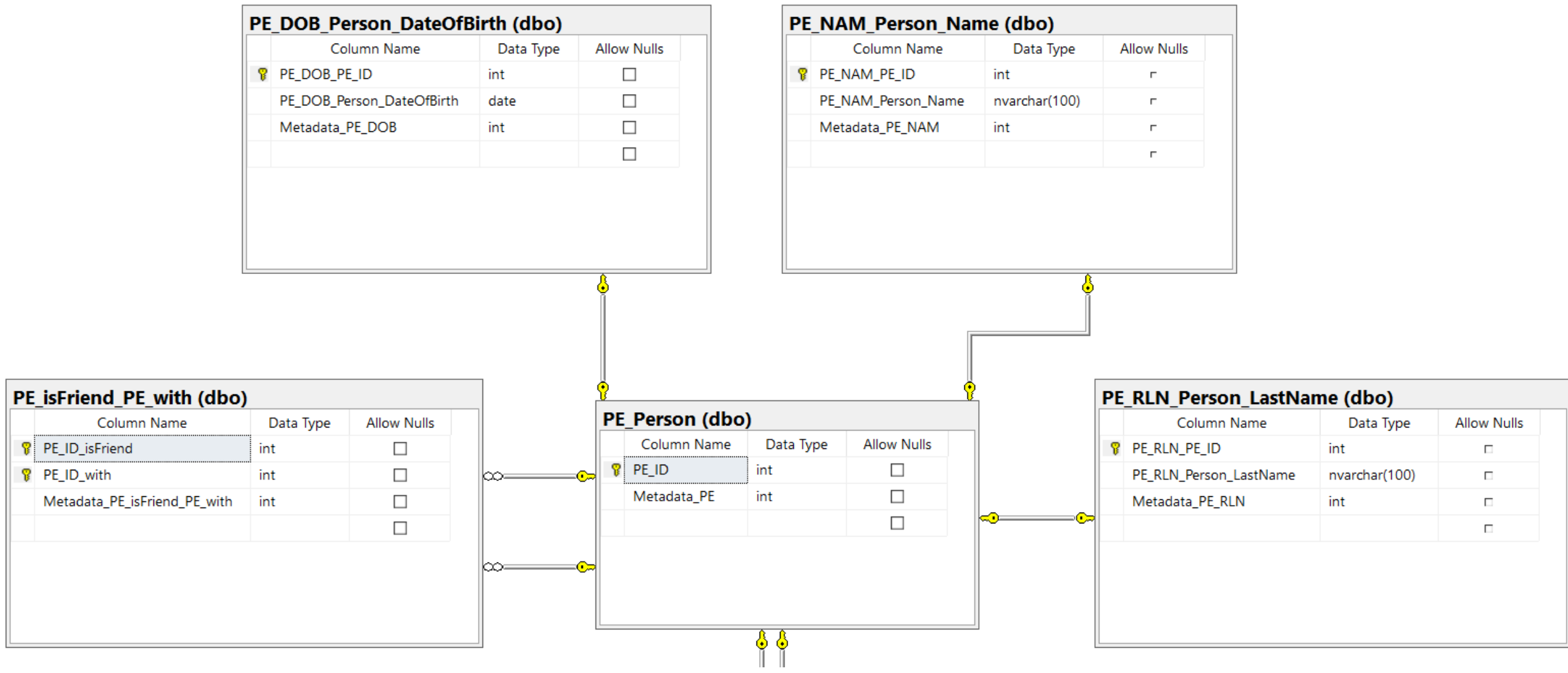
Column Name	Data Type	Allow Nulls
PE_ID_isFriend	int	<input type="checkbox"/>
PE_ID_with	int	<input type="checkbox"/>
Metadata_PE_isFriend_PE_with	int	<input type="checkbox"/>

PE_Person (dbo)

Column Name	Data Type	Allow Nulls
PE_ID	int	<input type="checkbox"/>
Metadata_PE	int	<input type="checkbox"/>

PE_RLN_Person_LastName (dbo)

Column Name	Data Type	Allow Nulls
PE_RLN_PE_ID	int	<input type="checkbox"/>
PE_RLN_Person_LastName	nvarchar(100)	<input type="checkbox"/>
Metadata_PE_RLN	int	<input type="checkbox"/>



PE_DOB_Person_DateOfBirth (dbo)

Column Name	Data Type	Allow Nulls
PE_DOB_PE_ID	int	<input type="checkbox"/>
PE_DOB_Person_DateOfBirth	date	<input type="checkbox"/>
Metadata_PE_DOB	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_NAM_Person_Name (dbo)

Column Name	Data Type	Allow Nulls
PE_NAM_PE_ID	int	<input type="checkbox"/>
PE_NAM_Person_Name	nvarchar(100)	<input type="checkbox"/>
Metadata_PE_NAM	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_isFriend_PE_with (dbo)

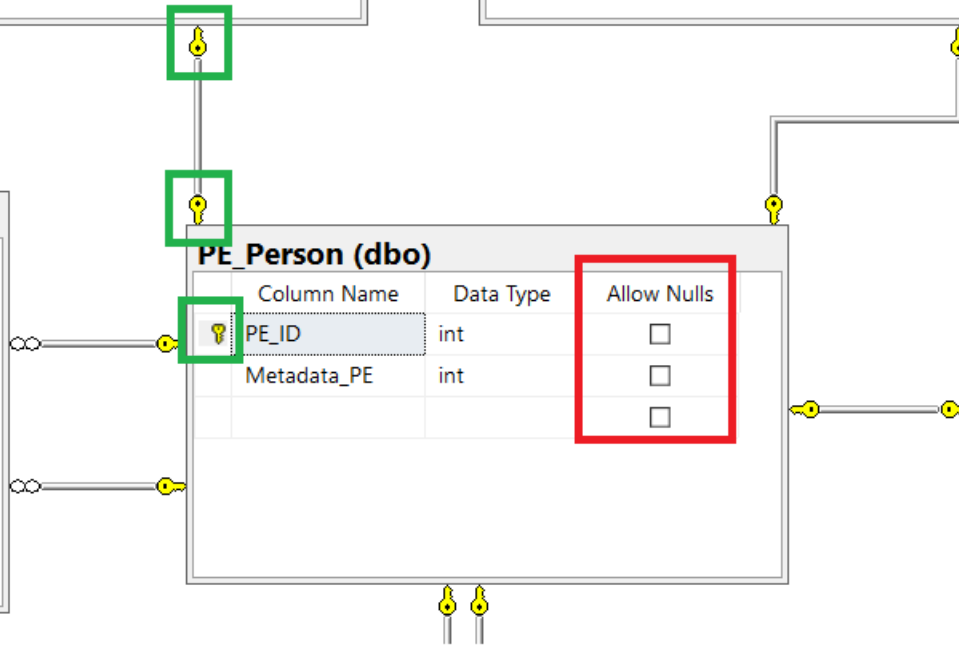
Column Name	Data Type	Allow Nulls
PE_ID_isFriend	int	<input type="checkbox"/>
PE_ID_with	int	<input type="checkbox"/>
Metadata_PE_isFriend_PE_with	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_Person (dbo)

Column Name	Data Type	Allow Nulls
PE_ID	int	<input type="checkbox"/>
Metadata_PE	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_RLN_Person_LastName (dbo)

Column Name	Data Type	Allow Nulls
PE_RLN_PE_ID	int	<input type="checkbox"/>
PE_RLN_Person_LastName	nvarchar(100)	<input type="checkbox"/>
Metadata_PE_RLN	int	<input type="checkbox"/>
		<input type="checkbox"/>



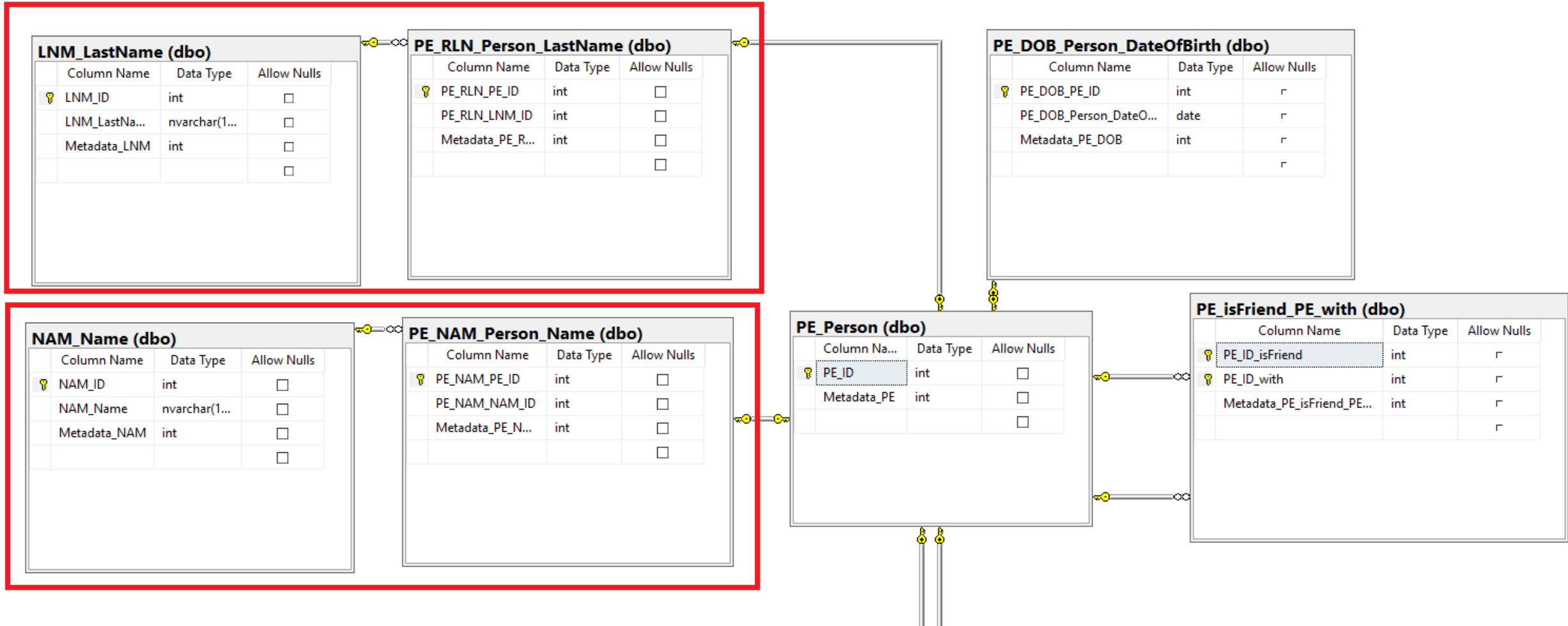
PEOPLE HAVE SAME NAMES AND LAST NAMES



DATA WILL BE DUPLICATED!

imgflip.com





IT LOOKS TIME-CONSUMING



AND NOT SO EASY



MODELING CONSTRUCTS



Anchor

- Represents business entity



LNМ_LastName (dbo)

Column Name	Data Type	Allow Nulls
LNМ_ID	int	<input type="checkbox"/>
LNМ_LastNa...	nvarchar(1...	<input type="checkbox"/>
Metadata_LNM	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_RLN_Person_LastName (dbo)

Column Name	Data Type	Allow Nulls
PE_RLN_PE_ID	int	<input type="checkbox"/>
PE_RLN_LNM_ID	int	<input type="checkbox"/>
Metadata_PE_R...	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_DOB_Person_DateOfBirth (dbo)

Column Name	Data Type	Allow Nulls
PE_DOB_PE_ID	int	<input type="checkbox"/>
PE_DOB_Person_DateO...	date	<input type="checkbox"/>
Metadata_PE_DOB	int	<input type="checkbox"/>
		<input type="checkbox"/>

NAM_Name (dbo)

Column Name	Data Type	Allow Nulls
NAM_ID	int	<input type="checkbox"/>
NAM_Name	nvarchar(1...	<input type="checkbox"/>
Metadata_NAM	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_NAM_Person_Name (dbo)

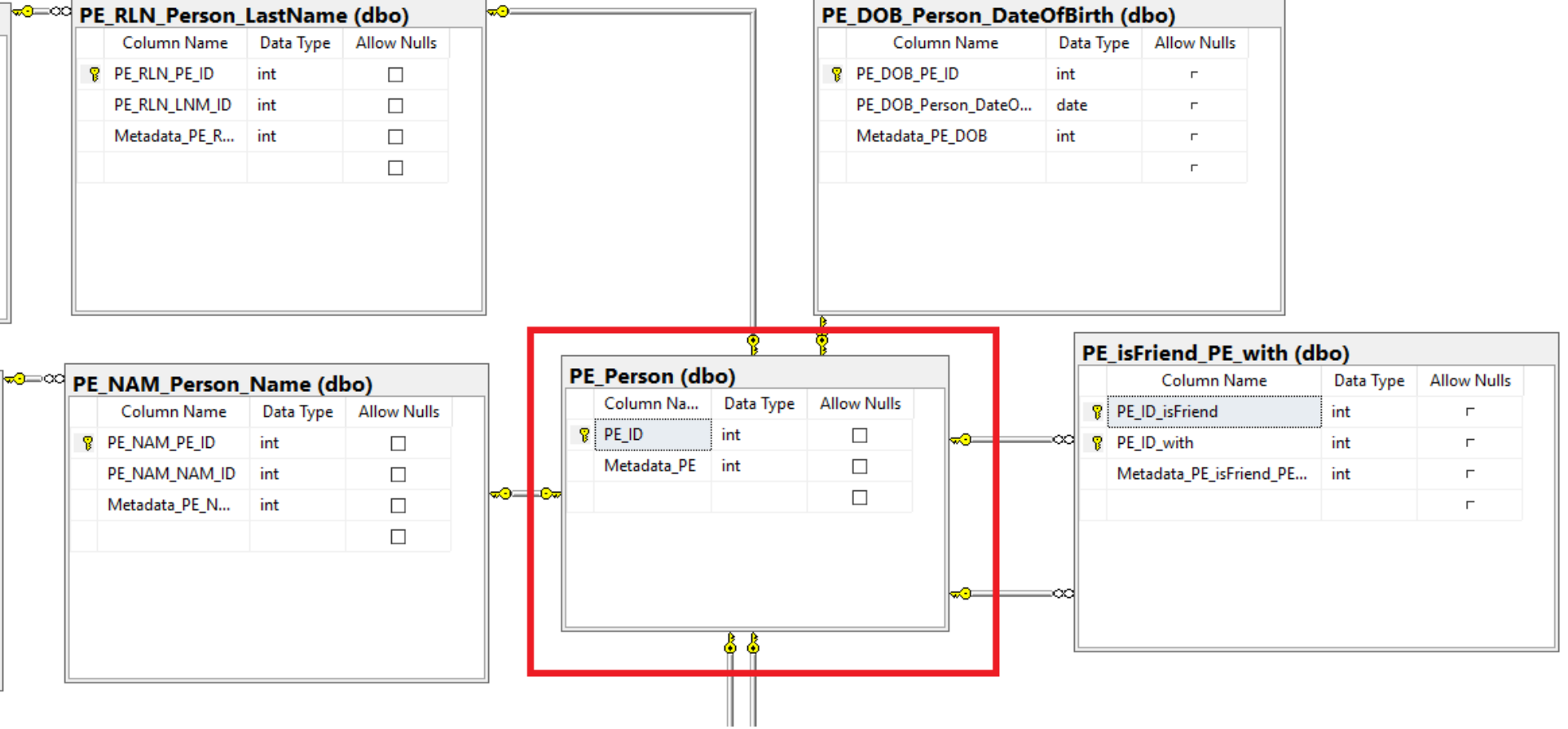
Column Name	Data Type	Allow Nulls
PE_NAM_PE_ID	int	<input type="checkbox"/>
PE_NAM_NAM_ID	int	<input type="checkbox"/>
Metadata_PE_N...	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_Person (dbo)

Column Na...	Data Type	Allow Nulls
PE_ID	int	<input type="checkbox"/>
Metadata_PE	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_isFriend_PE_with (dbo)

Column Name	Data Type	Allow Nulls
PE_ID_isFriend	int	<input type="checkbox"/>
PE_ID_with	int	<input type="checkbox"/>
Metadata_PE_isFriend_PE...	int	<input type="checkbox"/>
		<input type="checkbox"/>



Attribute

- Is property of an anchor
- Mnemonic must be unique in scope of an anchor
- Can be knotted



Column Name	Data Type	Allow Nulls
LNM_ID	int	<input type="checkbox"/>
LNM_LastNa...	nvarchar(1...	<input type="checkbox"/>
Metadata_LNM	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
PE_RLN_PE_ID	int	<input type="checkbox"/>
PE_RLN_LNM_ID	int	<input type="checkbox"/>
Metadata_PE_R...	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
PE_DOB_PE_ID	int	<input type="checkbox"/>
PE_DOB_Person_DateO...	date	<input type="checkbox"/>
Metadata_PE_DOB	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
NAM_ID	int	<input type="checkbox"/>
NAM_Name	nvarchar(1...	<input type="checkbox"/>
Metadata_NAM	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
PE_NAM_PE_ID	int	<input type="checkbox"/>
PE_NAM_NAM_ID	int	<input type="checkbox"/>
Metadata_PE_N...	int	<input type="checkbox"/>

Column Na...	Data Type	Allow Nulls
PE_ID	int	<input type="checkbox"/>
Metadata_PE	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
PE_ID_isFriend	int	<input type="checkbox"/>
PE_ID_with	int	<input type="checkbox"/>
Metadata_PE_isFriend_PE...	int	<input type="checkbox"/>



Knot

- Used when attribute is repetitive for many entities ie. Category
- Has it's own identity key (ie. CategoryId)
- Think of it as dictionary table
- Linked with anchor through attribute



Column Name	Data Type	Allow Nulls
LNM_ID	int	<input type="checkbox"/>
LNM_LastNa...	nvarchar(1...	<input type="checkbox"/>
Metadata_LNM	int	<input type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
PE_RLN_PE_ID	int	<input type="checkbox"/>
PE_RLN_LNM_ID	int	<input type="checkbox"/>
Metadata_PE_R...	int	<input type="checkbox"/>
		<input type="checkbox"/>

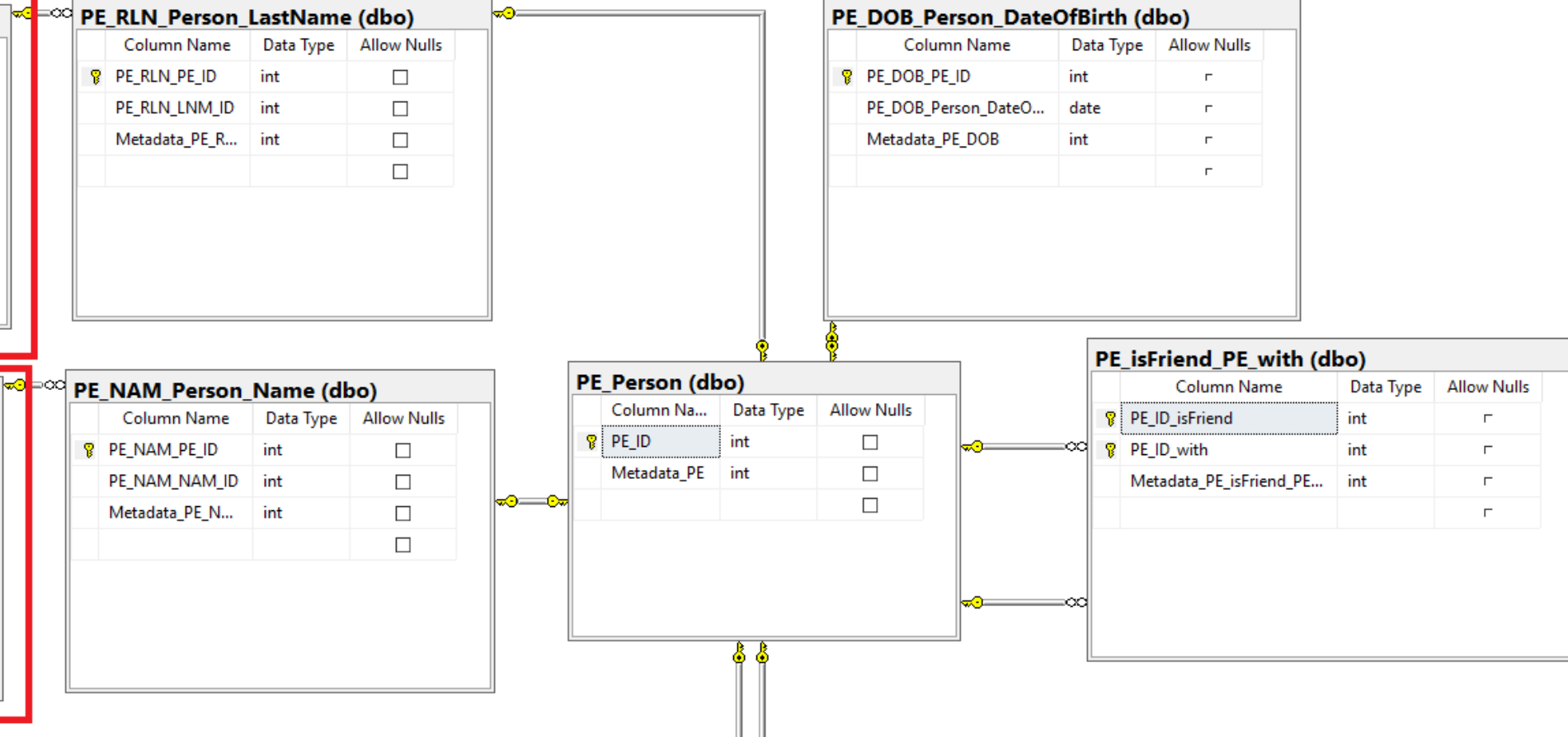
Column Name	Data Type	Allow Nulls
PE_DOB_PE_ID	int	<input type="checkbox"/>
PE_DOB_Person_DateO...	date	<input type="checkbox"/>
Metadata_PE_DOB	int	<input type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
NAM_ID	int	<input type="checkbox"/>
NAM_Name	nvarchar(1...	<input type="checkbox"/>
Metadata_NAM	int	<input type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
PE_NAM_PE_ID	int	<input type="checkbox"/>
PE_NAM_NAM_ID	int	<input type="checkbox"/>
Metadata_PE_N...	int	<input type="checkbox"/>
		<input type="checkbox"/>

Column Na...	Data Type	Allow Nulls
PE_ID	int	<input type="checkbox"/>
Metadata_PE	int	<input type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
PE_ID_isFriend	int	<input type="checkbox"/>
PE_ID_with	int	<input type="checkbox"/>
Metadata_PE_isFriend_PE...	int	<input type="checkbox"/>
		<input type="checkbox"/>



Tie

- Models relationships between anchors
- Has roles (ie. “knows”, “is owned by”)



LNМ_LastName (dbo)

Column Name	Data Type	Allow Nulls
LNМ_ID	int	<input type="checkbox"/>
LNМ_LastNa...	nvarchar(1...	<input type="checkbox"/>
Metadata_LNM	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_RLN_Person_LastName (dbo)

Column Name	Data Type	Allow Nulls
PE_RLN_PE_ID	int	<input type="checkbox"/>
PE_RLN_LNM_ID	int	<input type="checkbox"/>
Metadata_PE_R...	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_DOB_Person_DateOfBirth (dbo)

Column Name	Data Type	Allow Nulls
PE_DOB_PE_ID	int	<input type="checkbox"/>
PE_DOB_Person_DateO...	date	<input type="checkbox"/>
Metadata_PE_DOB	int	<input type="checkbox"/>
		<input type="checkbox"/>

NAM_Name (dbo)

Column Name	Data Type	Allow Nulls
NAM_ID	int	<input type="checkbox"/>
NAM_Name	nvarchar(1...	<input type="checkbox"/>
Metadata_NAM	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_NAM_Person_Name (dbo)

Column Name	Data Type	Allow Nulls
PE_NAM_PE_ID	int	<input type="checkbox"/>
PE_NAM_NAM_ID	int	<input type="checkbox"/>
Metadata_PE_N...	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_Person (dbo)

Column Na...	Data Type	Allow Nulls
PE_ID	int	<input type="checkbox"/>
Metadata_PE	int	<input type="checkbox"/>
		<input type="checkbox"/>

PE_isFriend_PE_with (dbo)

Column Name	Data Type	Allow Nulls
PE_ID_isFriend	int	<input type="checkbox"/>
PE_ID_with	int	<input type="checkbox"/>
Metadata_PE_isFriend_PE...	int	<input type="checkbox"/>
		<input type="checkbox"/>



IT'S EASY

RIGHT?

imgflip.com



**THERE IS
ONE QUESTION**



YOU SHOULD ASK

imgflip.com

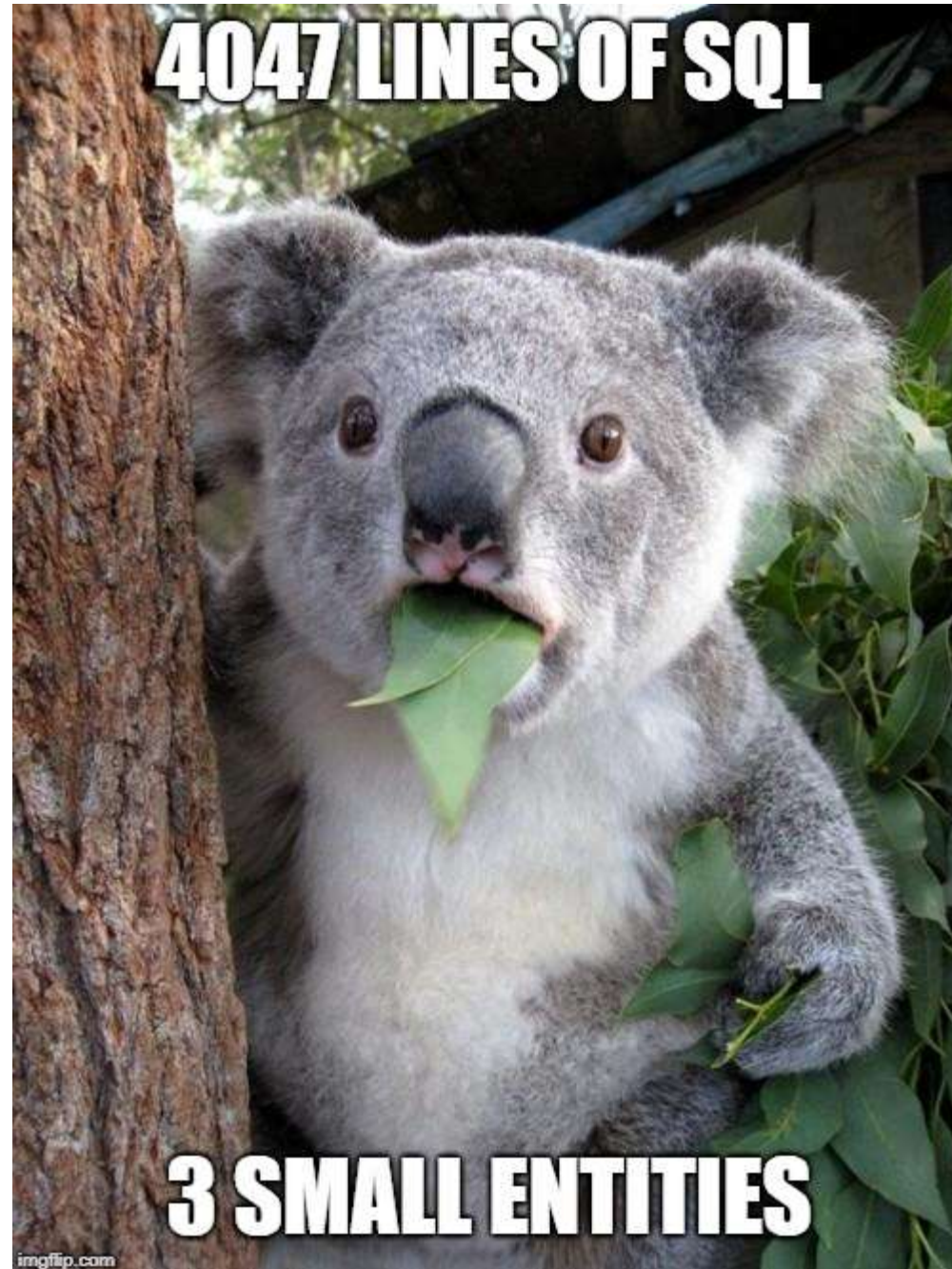


How many lines was required to build
above sample model?

- 4047



4047 LINES OF SQL



3 SMALL ENTITIES

imgflip.com

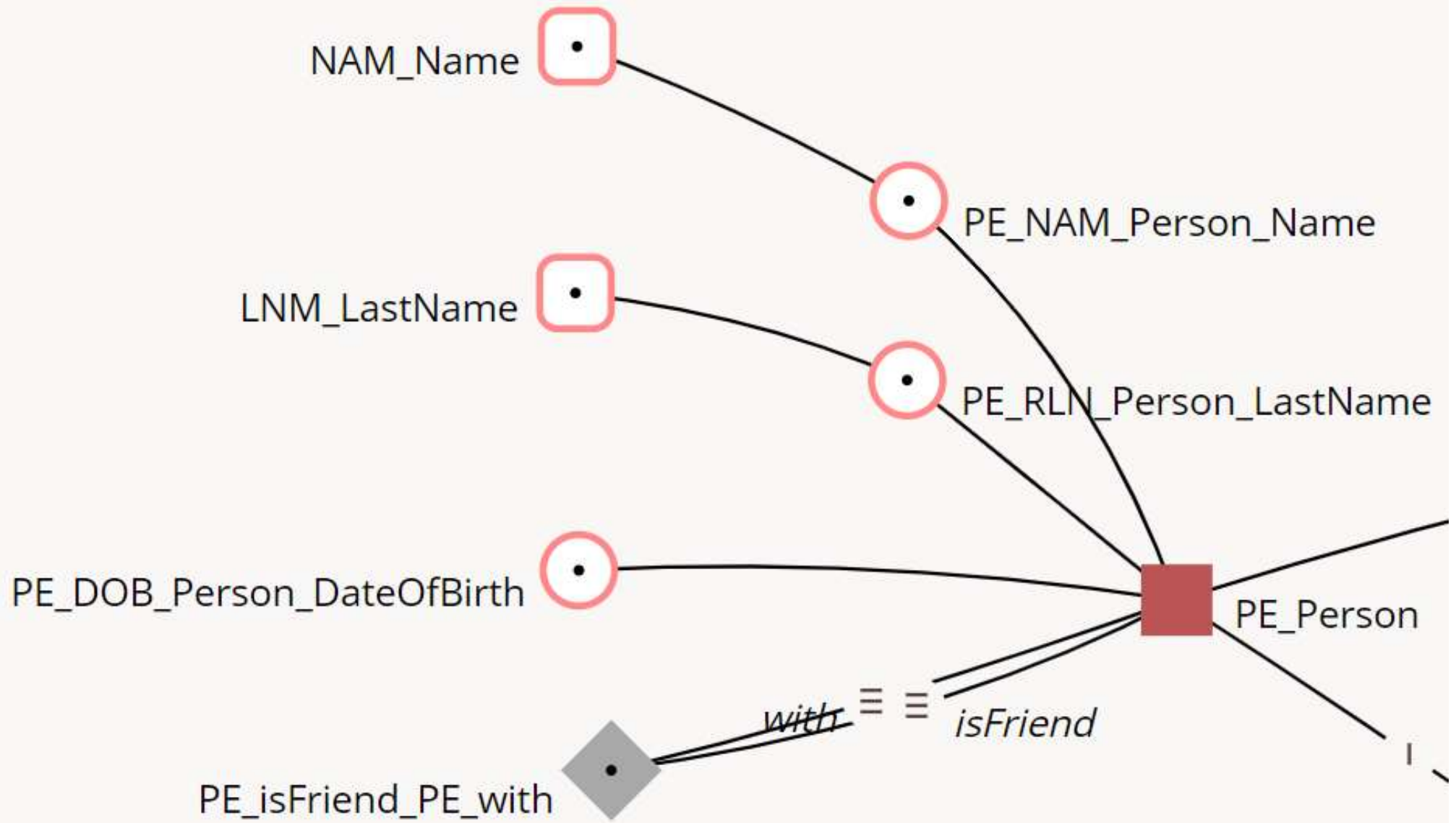


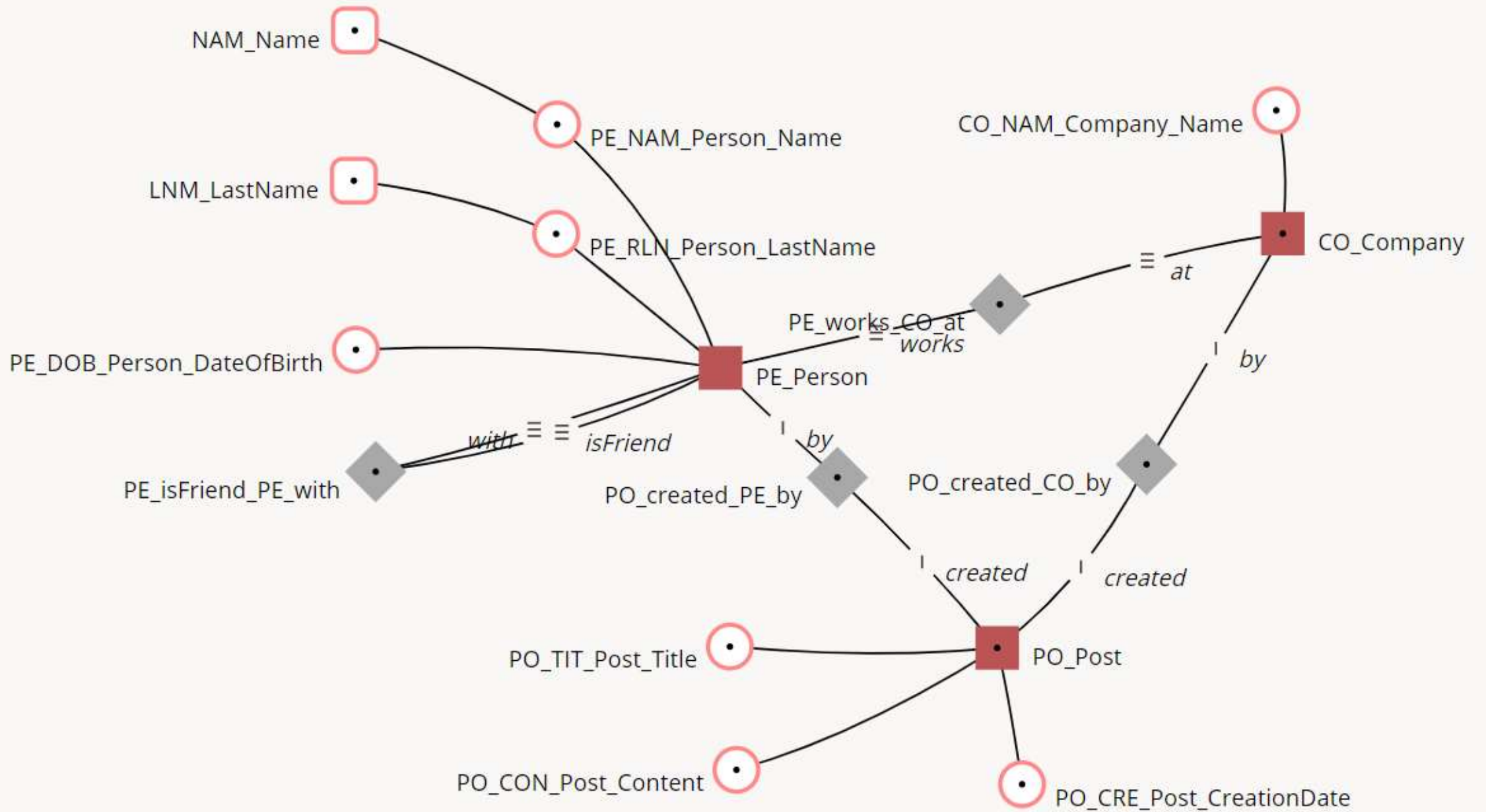
A close-up image of Morpheus from the movie The Matrix, wearing his signature sunglasses. The text "WHAT IF I TOLD YOU" is overlaid in large, white, bold, sans-serif font at the top of the image.

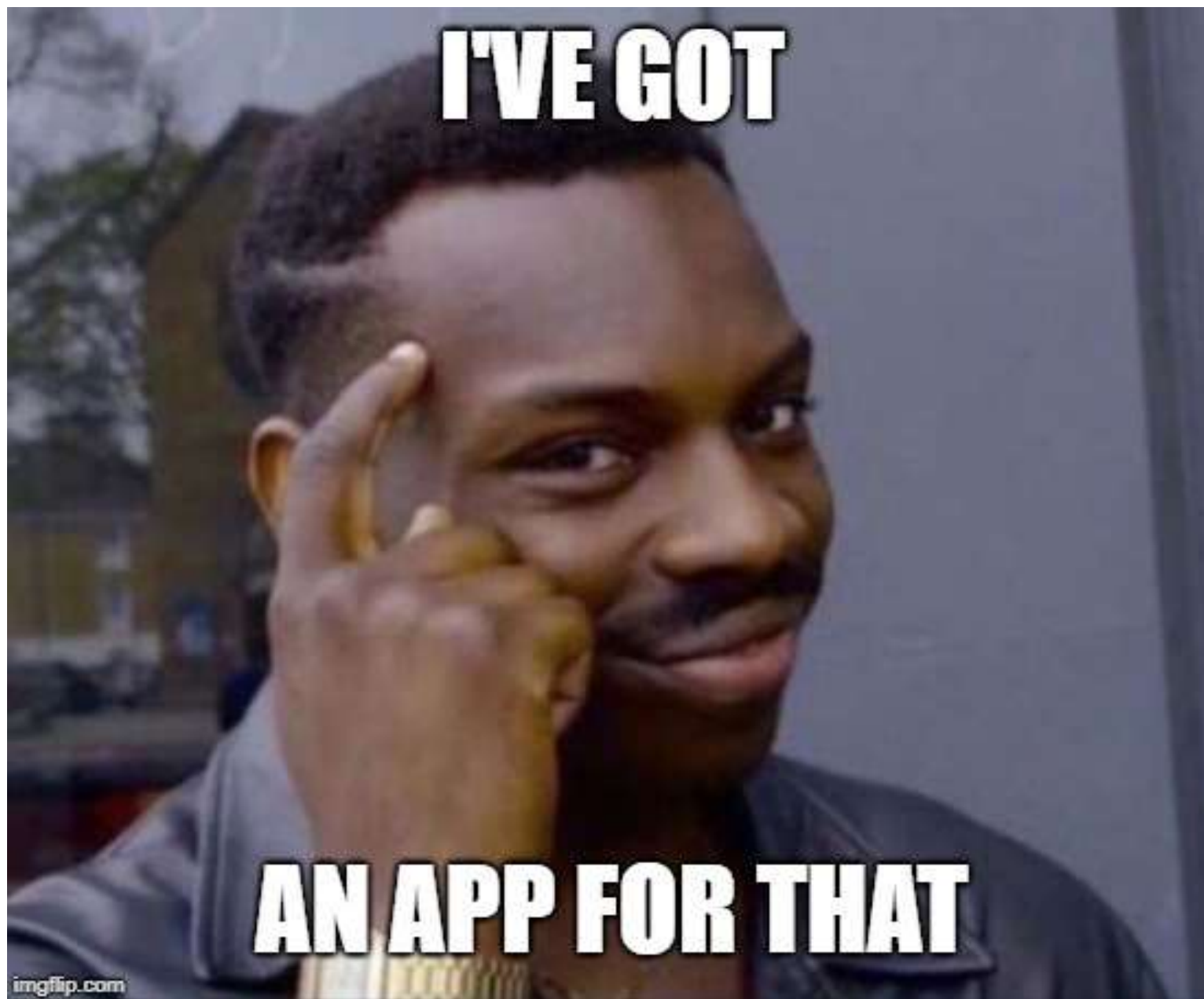
WHAT IF I TOLD YOU

THERE'S AN EASIER WAY





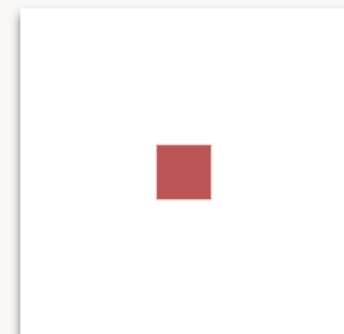




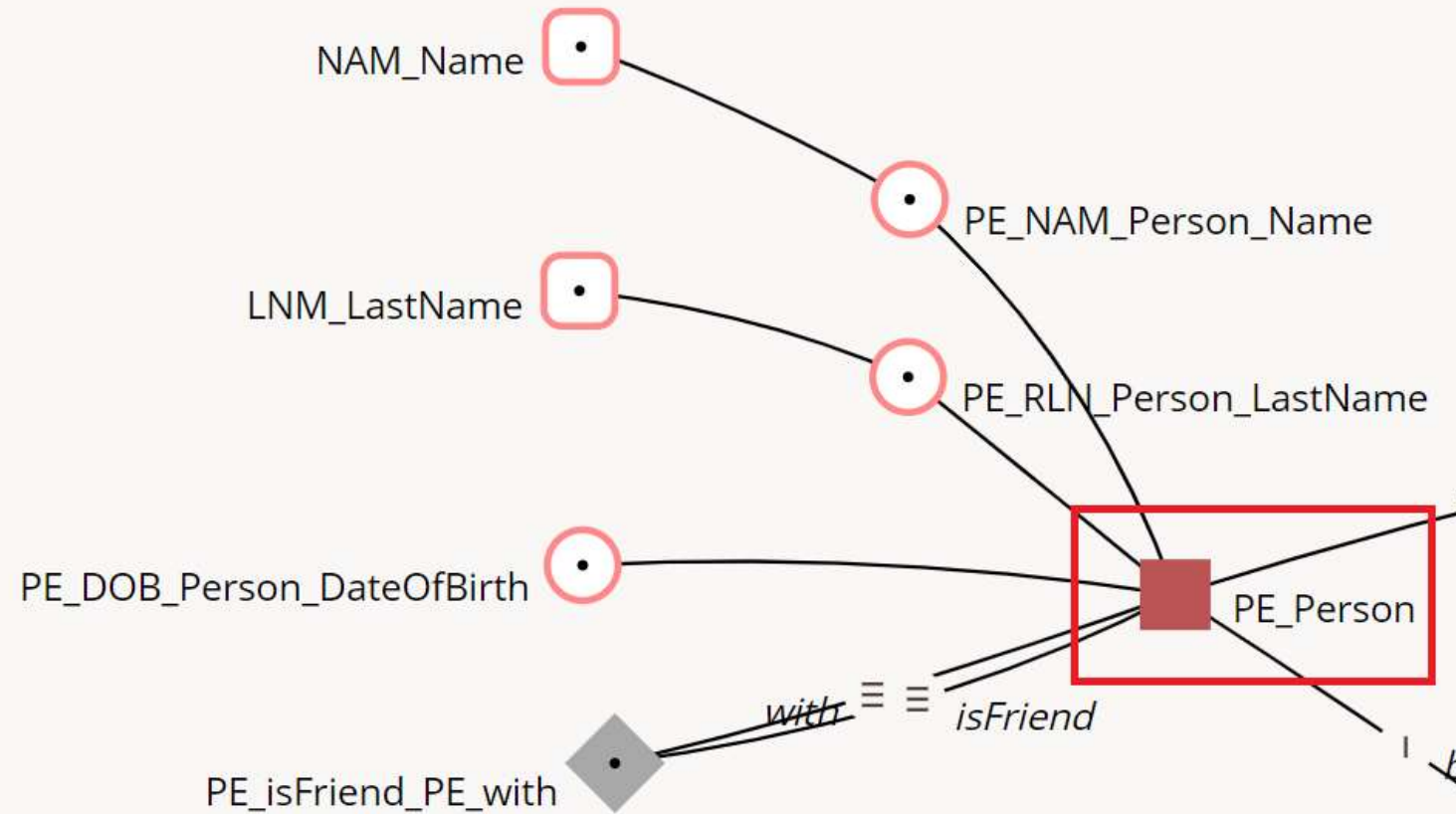
B



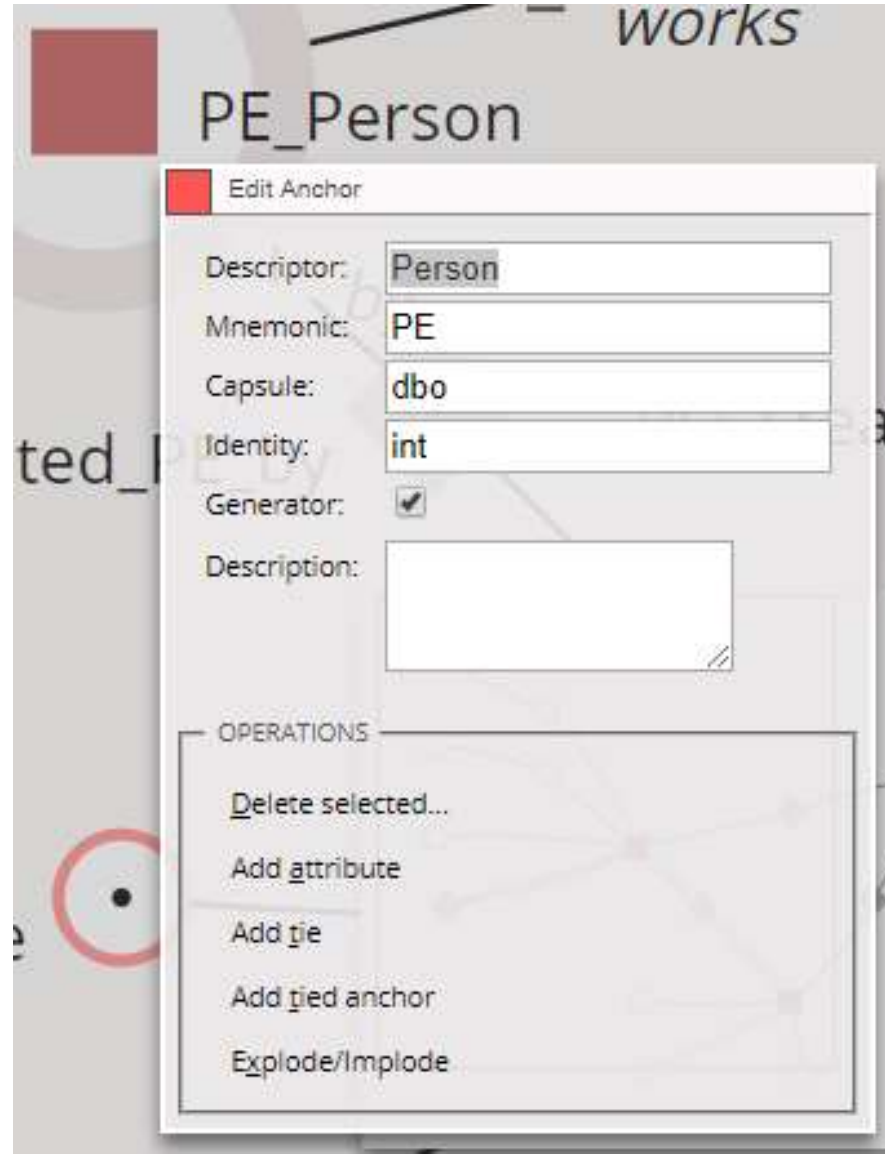
-



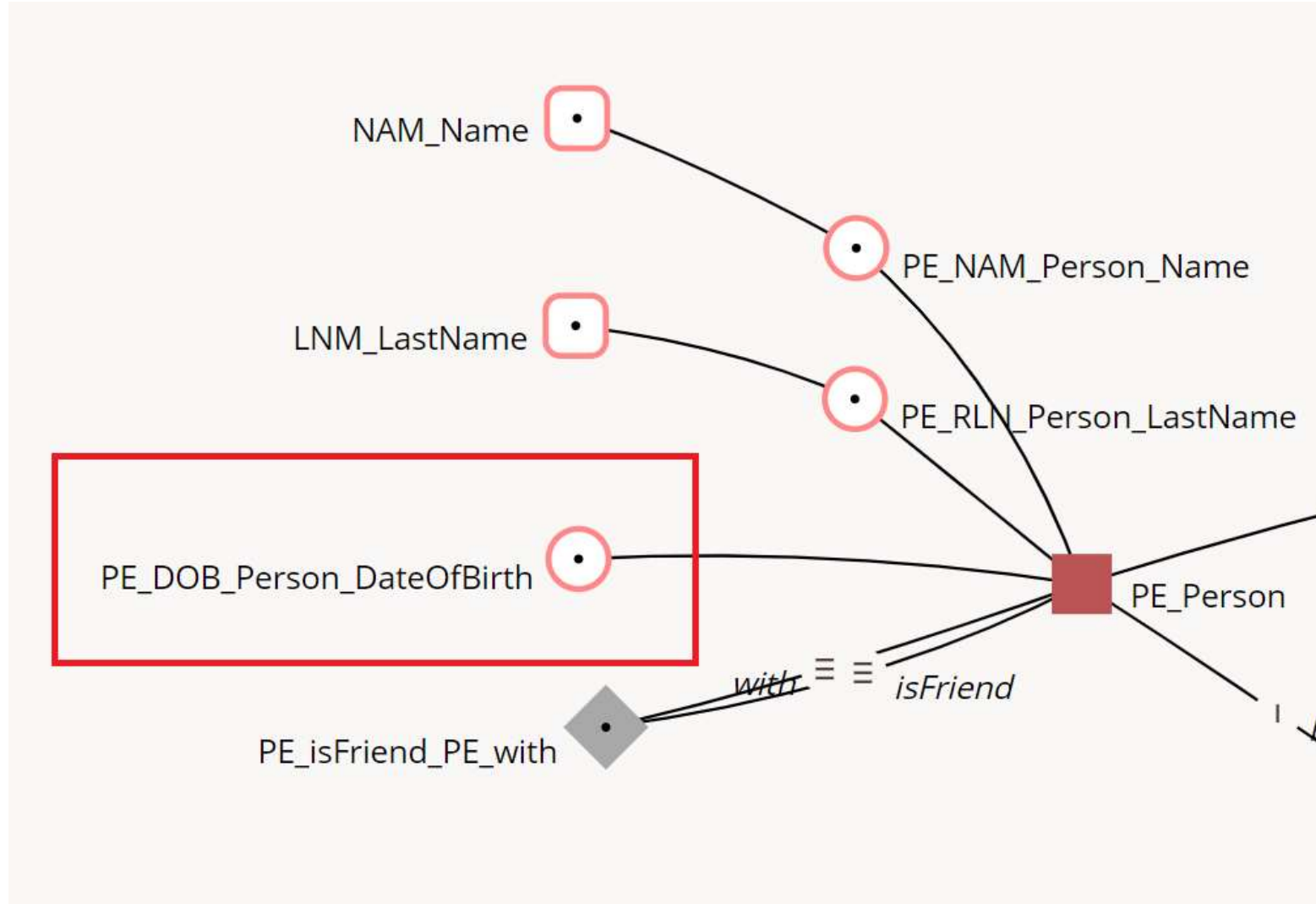
Anchor



Anchor



Attribute



Attribute



The image shows a screenshot of a software interface with a dialog box titled "Edit Attribute". The dialog box has a red square icon in its top-left corner. It contains several input fields and a section for operations. The fields are: "Descriptor" with the value "DateOfBirth", "Mnemonic" with "DOB", "Capsule" with "dbo", and "Data range" with "date". There is an unchecked checkbox for "Checksum" and an empty text area for "Description". At the bottom, there is a section titled "OPERATIONS" containing three buttons: "Delete selected...", "Toggle knotted", and "Toggle historized".

h

Fr

PO

Edit Attribute

Descriptor:

Mnemonic:

Capsule:

Data range:

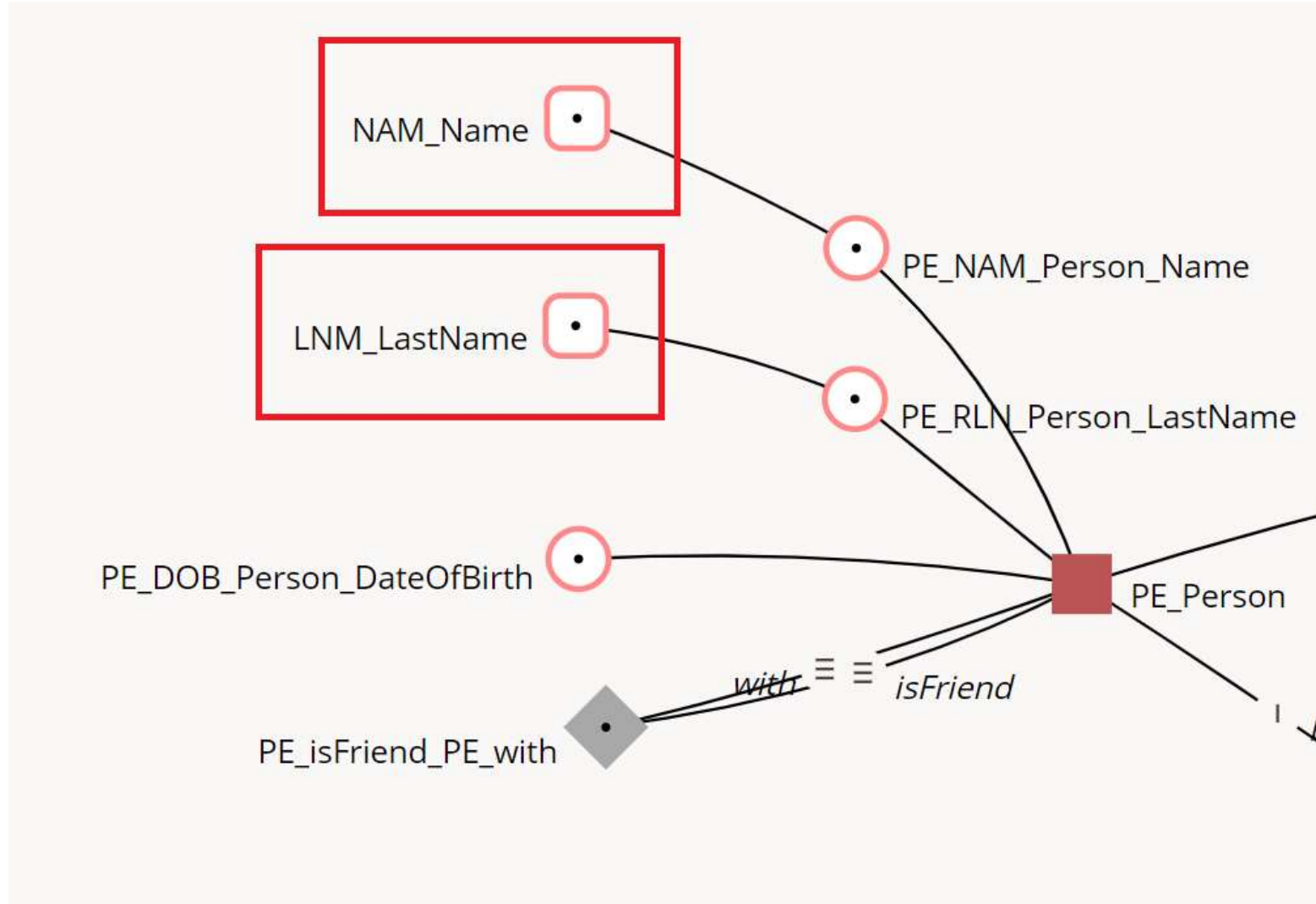
Checksum:

Description:

OPERATIONS



Knot



Knot

Edit Knot

Descriptor:

Mnemonic:

Capsule:

Data range:

Identity:

Generator:

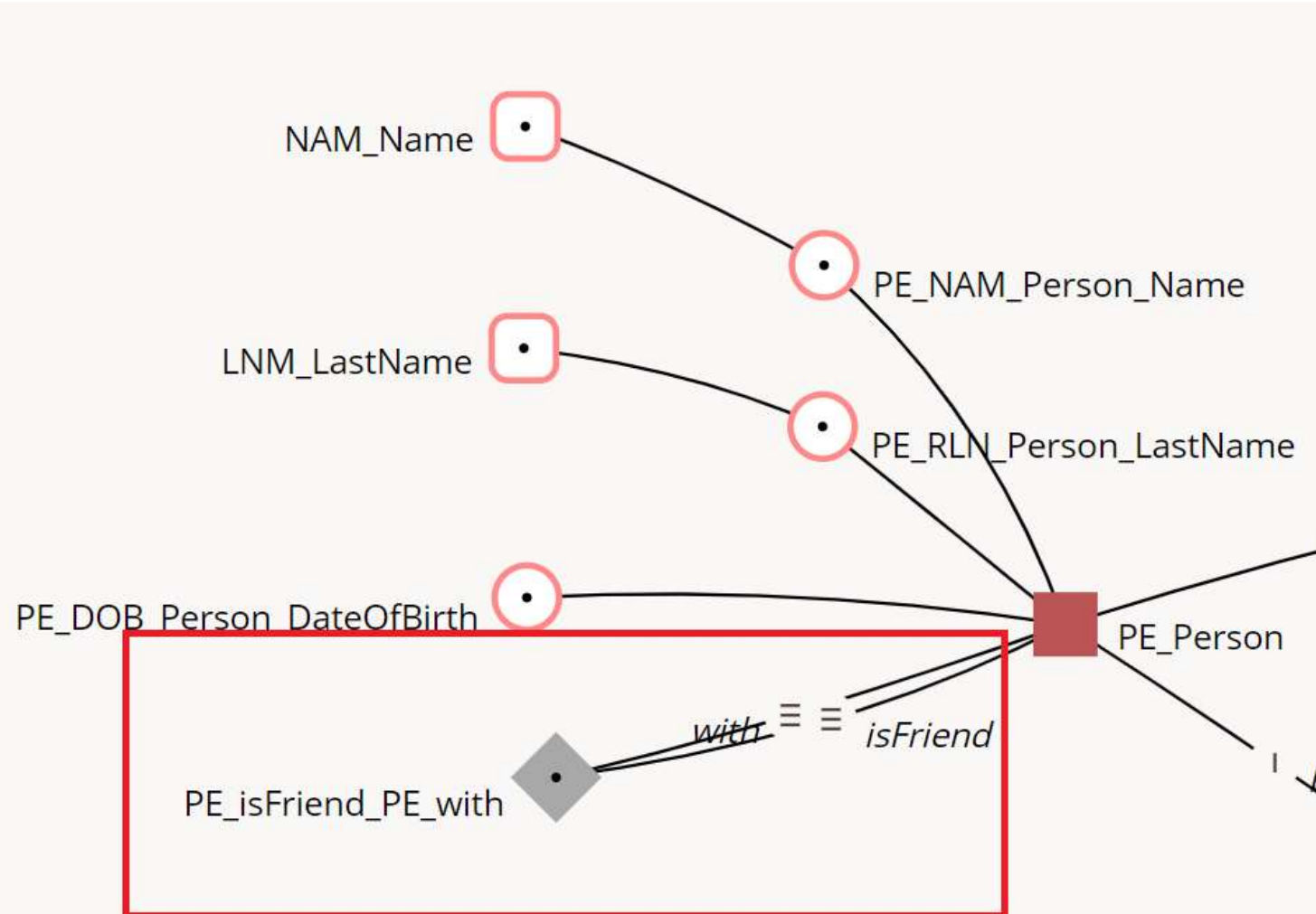
Checksum:

Description:

OPERATIONS

B

Tie



Tie

Edit Tie

Capsule:

Description:

OPERATIONS

- [Delete selected...](#)
- [Toggle historized](#)
- [Add anchor](#)
- [Add knot](#)

Edit Edge

Role:

Coloring:

Description:

OPERATIONS

- [Delete selected...](#)
- [Toggle identifier](#)
- [Make first role](#)



TIMETRAVEL!

Temporal data

- Any data can be historized
- You can record changes at specified point of time (ie. Price year ago)



BRACE YOURSELVES



A CHANGE REQUEST IS COMING

makeameme.org



Change request

- Post Title should be versioned
- Post Content should be versioned



Edit Attribute

Descriptor:

Mnemonic:

Capsule:

Data range:

Checksum:

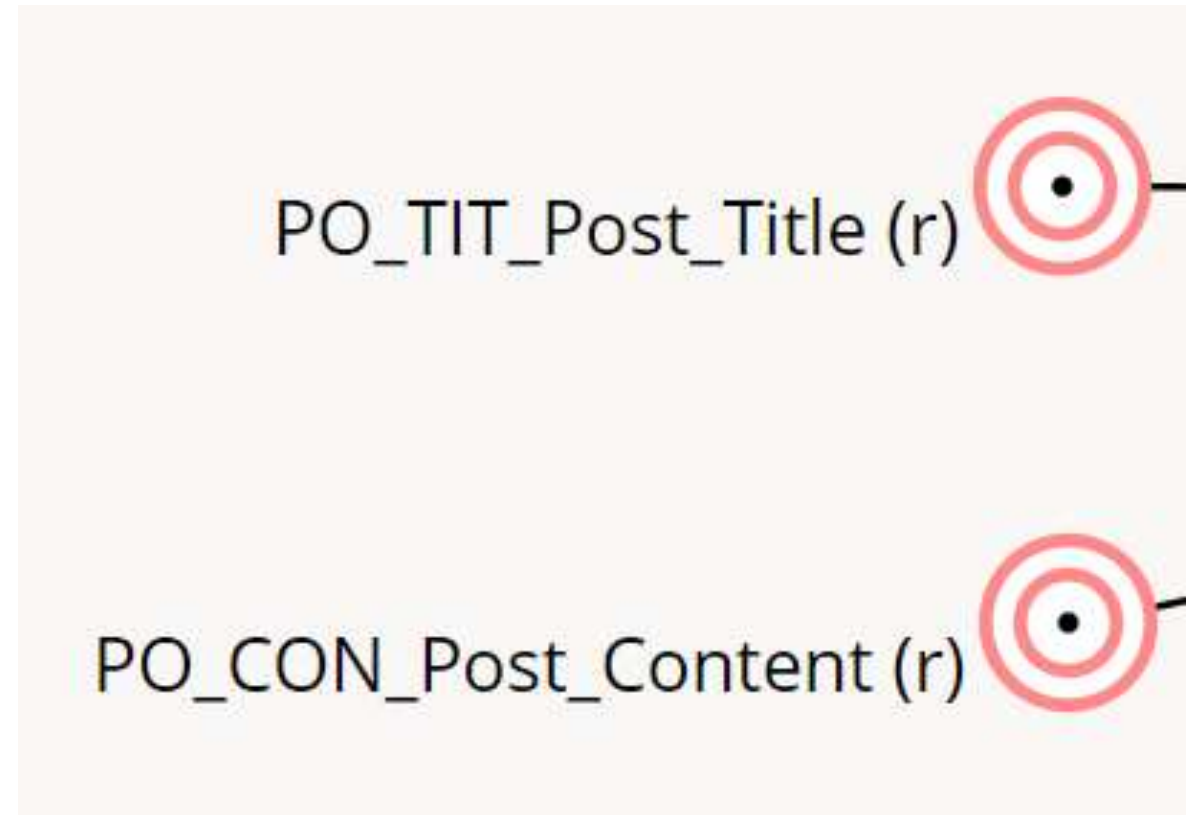
Description:

OPERATIONS

[Delete selected...](#)

[Toggle knotted](#)

[Toggle historized](#)



PO_Post (dbo)

	Column Name	Data Type	Allow Nulls
🔑	PO_ID	int	<input type="checkbox"/>
	Metadata_PO	int	<input type="checkbox"/>
			<input type="checkbox"/>

PO_TIT_Post_Title (dbo)

	Column Name	Data Type	Allow Nulls
🔑	PO_TIT_PO_ID	int	<input type="checkbox"/>
	PO_TIT_Post_Title	nvarchar(200)	<input type="checkbox"/>
🔑	PO_TIT_ChangedAt	datetime	<input type="checkbox"/>
	Metadata_PO_TIT	int	<input type="checkbox"/>
			<input type="checkbox"/>

PO_CON_Post_Content (dbo)

	Column Name	Data Type	Allow Nulls
🔑	PO_CON_PO_ID	int	<input type="checkbox"/>
	PO_CON_Post_Content	nvarchar(MAX)	<input type="checkbox"/>
🔑	PO_CON_ChangedAt	datetime	<input type="checkbox"/>
	Metadata_PO_CON	int	<input type="checkbox"/>
			<input type="checkbox"/>

PO_Post (dbo)

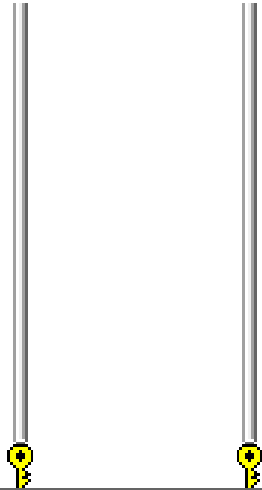
	Column Name	Data Type	Allow Nulls
🔑	PO_ID	int	<input type="checkbox"/>
	Metadata_PO	int	<input type="checkbox"/>
			<input type="checkbox"/>

PO_TIT_Post_Title (dbo)

	Column Name	Data Type	Allow Nulls
🔑	PO_TIT_PO_ID	int	<input type="checkbox"/>
	PO_TIT_Post_Title	nvarchar(200)	<input type="checkbox"/>
🔑	PO_TIT_ChangedAt	datetime	<input type="checkbox"/>
	Metadata_PO_TIT	int	<input type="checkbox"/>
			<input type="checkbox"/>

PO_CON_Post_Content (dbo)

	Column Name	Data Type	Allow Nulls
🔑	PO_CON_PO_ID	int	<input type="checkbox"/>
	PO_CON_Post_Content	nvarchar(MAX)	<input type="checkbox"/>
🔑	PO_CON_ChangedAt	datetime	<input type="checkbox"/>
	Metadata_PO_CON	int	<input type="checkbox"/>
			<input type="checkbox"/>



**AND HOW I'M
SUPPOSED TO READ**

**FROM SHITLOAD OF
TABLES WITH FUCKED UP NAMES?**



- [-] Views
 - + System Views
 - + dbo._Anchor
 - + dbo._Attribute
 - + dbo._Knot
 - + dbo._Schema_Expanded
 - + dbo._Tie
 - + dbo.Current_Company
 - + dbo.Current_Person
 - + dbo.Current_Person_with_Person_isFriend
 - + dbo.Current_Person_works_Company_at
 - + dbo.Current_Post
 - + dbo.Current_Post_created_Company_by
 - + dbo.Current_Post_created_Person_by
 - + dbo.Latest_Company
 - + dbo.Latest_Person
 - + dbo.Latest_Person_with_Person_isFriend
 - + dbo.Latest_Person_works_Company_at
 - + dbo.Latest_Post
 - + dbo.Latest_Post_created_Company_by
 - + dbo.Latest_Post_created_Person_by

- [-] Functions
 - [-] Table-valued Functions
 - + dbo._Evolution
 - + dbo.Difference_Post
 - + dbo.dPO_Post
 - + dbo.pCO_Company
 - + dbo.Point_Company
 - + dbo.Point_Person
 - + dbo.Point_Person_with_Person_isFriend
 - + dbo.Point_Person_works_Company_at
 - + dbo.Point_Post
 - + dbo.Point_Post_created_Company_by
 - + dbo.Point_Post_created_Person_by



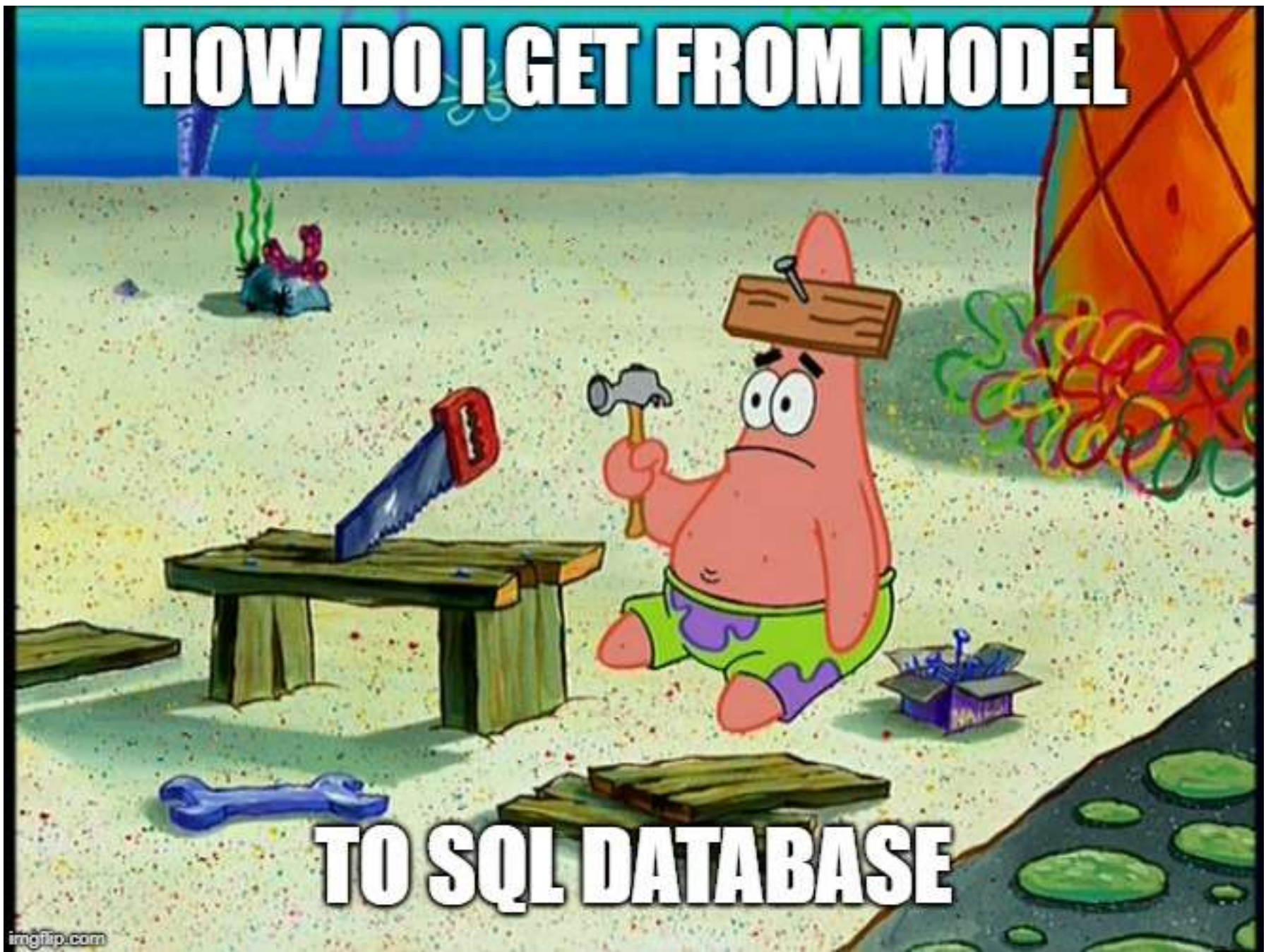
Temporal perspectives

Based on views/functions with prefixes:

- n / Current –now
- l / Latest – latest (can include future data)
- p / Point – data in specified point in time
- d / Difference – data changes in two points in time



HOW DO I GET FROM MODEL



TO SQL DATABASE



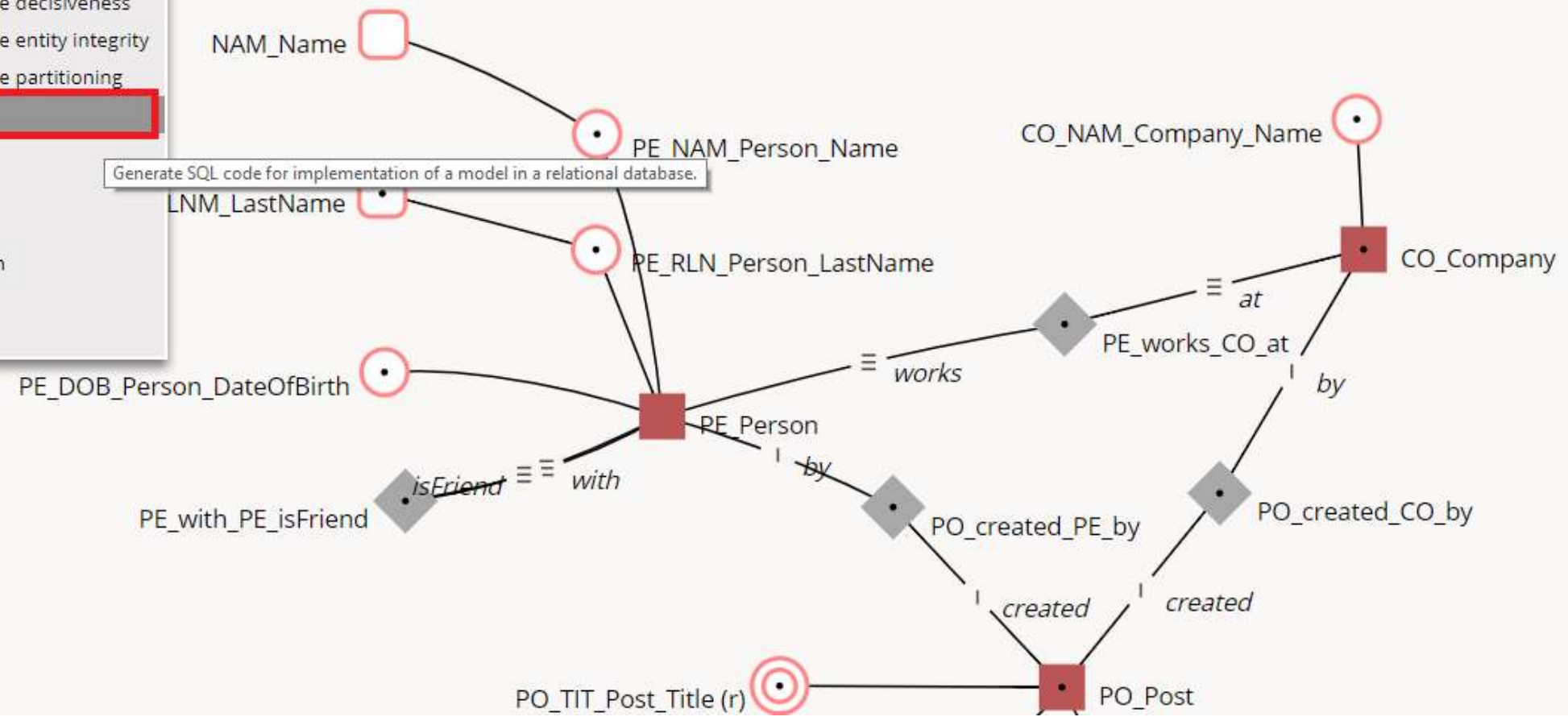
Microsoft SQL ▾ Target database
Improved ▾ Naming convention
Unitemporal ▾ Temporalization

- Use metadata
- Use business views
- Use equivalence
- Use decisiveness
- Use entity integrity
- Use partitioning

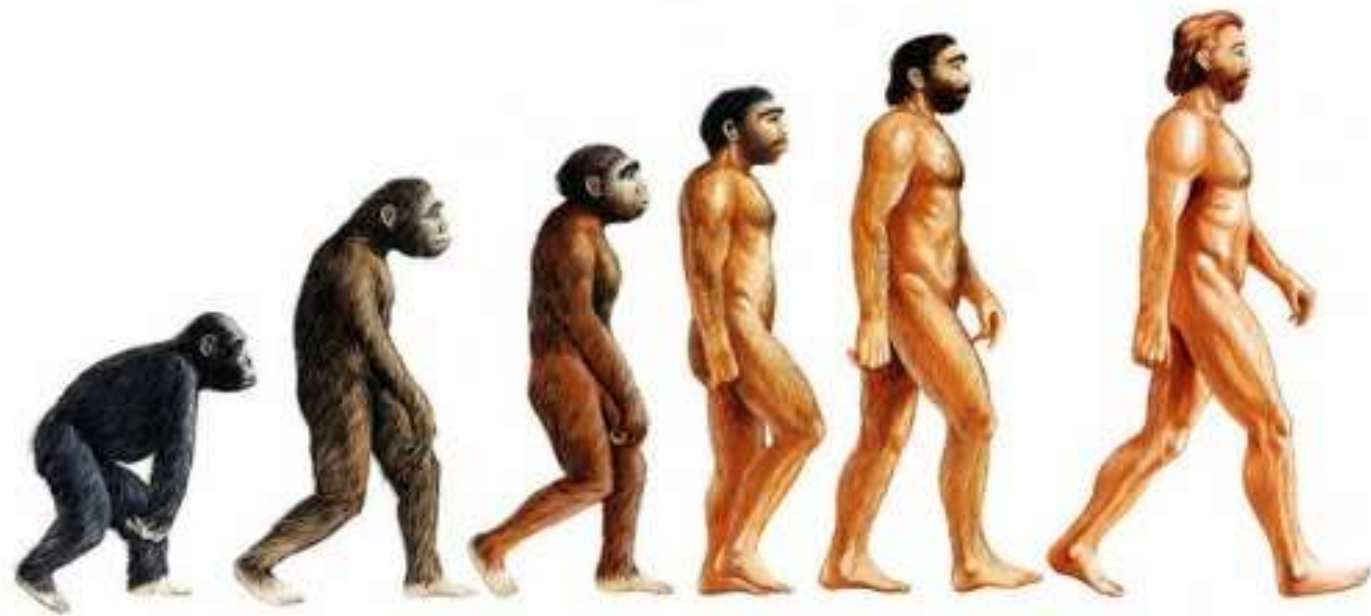
SQL code

- XML code
- JSON string
- URL for sharing
- HTML documentation
- PNG screenshot
- SVG screenshot

search filter None selected ▾



EVOLVING



YOUR SCHEMA



Evolving your schema

- Highly iterative (Modifying already deployed tables can cause errors)
- Does not touch existing data
- Previous schema is always subset of new one
- Deployments can be made without any downtime
- It's easy to maintain backwards compatibility within an app



DB TUNING ADVISOR



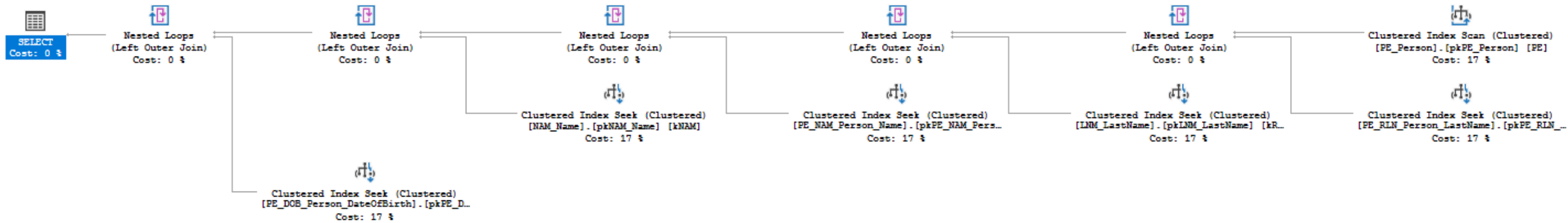
Table elimination

- It's possible because of highly normalization and relational structure
- In short “if you don't need it – don't touch it”
- Requires explicit select statements

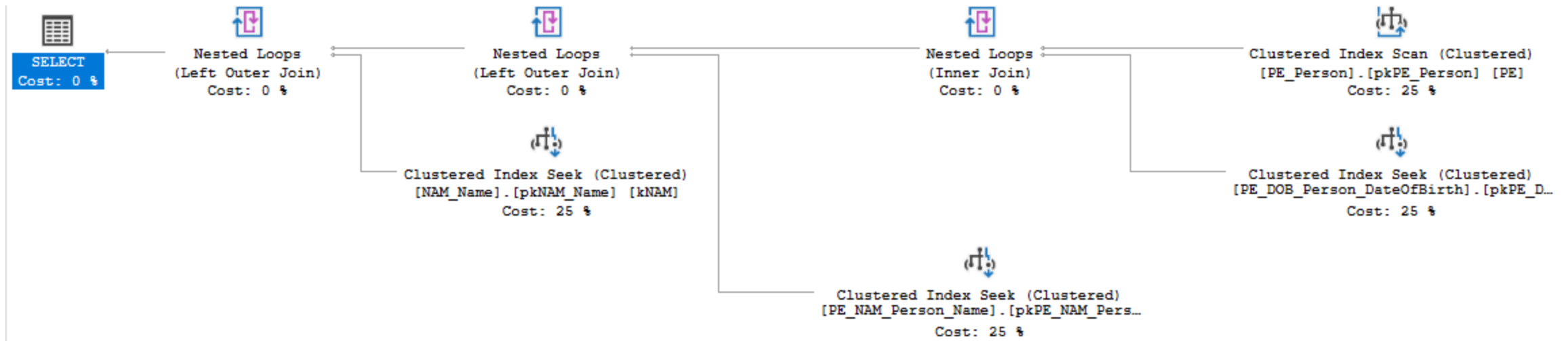


SELECT *

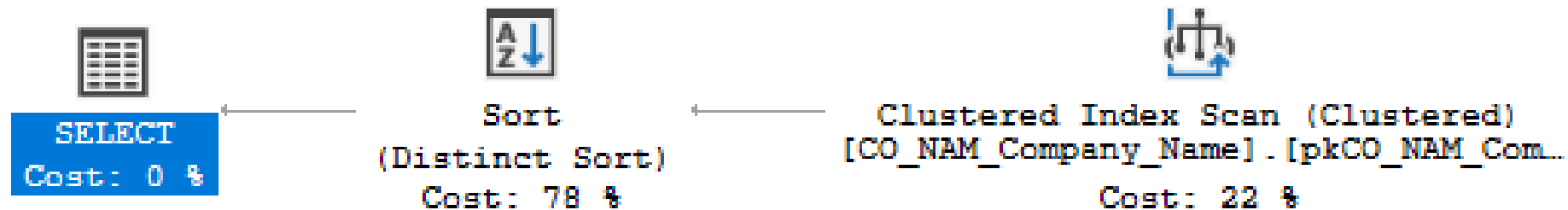
FROM [dbo].[Current_Person]



```
] SELECT [Name_Name]
FROM [dbo].[Current_Person]
WHERE [DateOfBirth] >= '1988-01-01'
```




```
SELECT DISTINCT [Name]
FROM [dbo].[Current_Company]
WHERE [Name] LIKE 'Mic%'
```



ITS ONLY JUST THE BEGINNING



imgflip.com





BUT

**NOT EVERYTHING
IS PERFECT**



Cons

- Using it in your apps will require some skill
- It's niche
- Not everything works in Azure SQL
- Modifications by hand are not easy



Resources

- www.anchormodeling.com

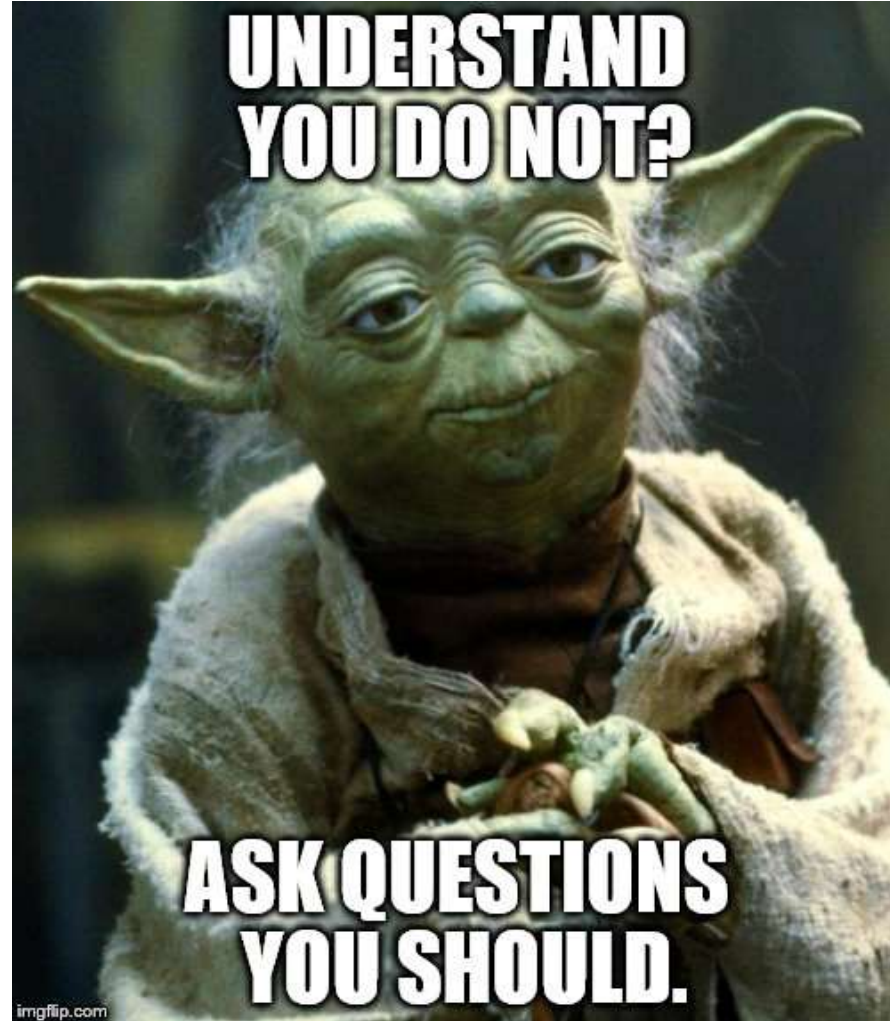


Slides

- **bit.ly/rh-anchor**



Questions?



Try it – at home



fb.me/hryniewskinet

HRYNIEWSKI.NET



[@r_hryniewski](https://twitter.com/r_hryniewski)