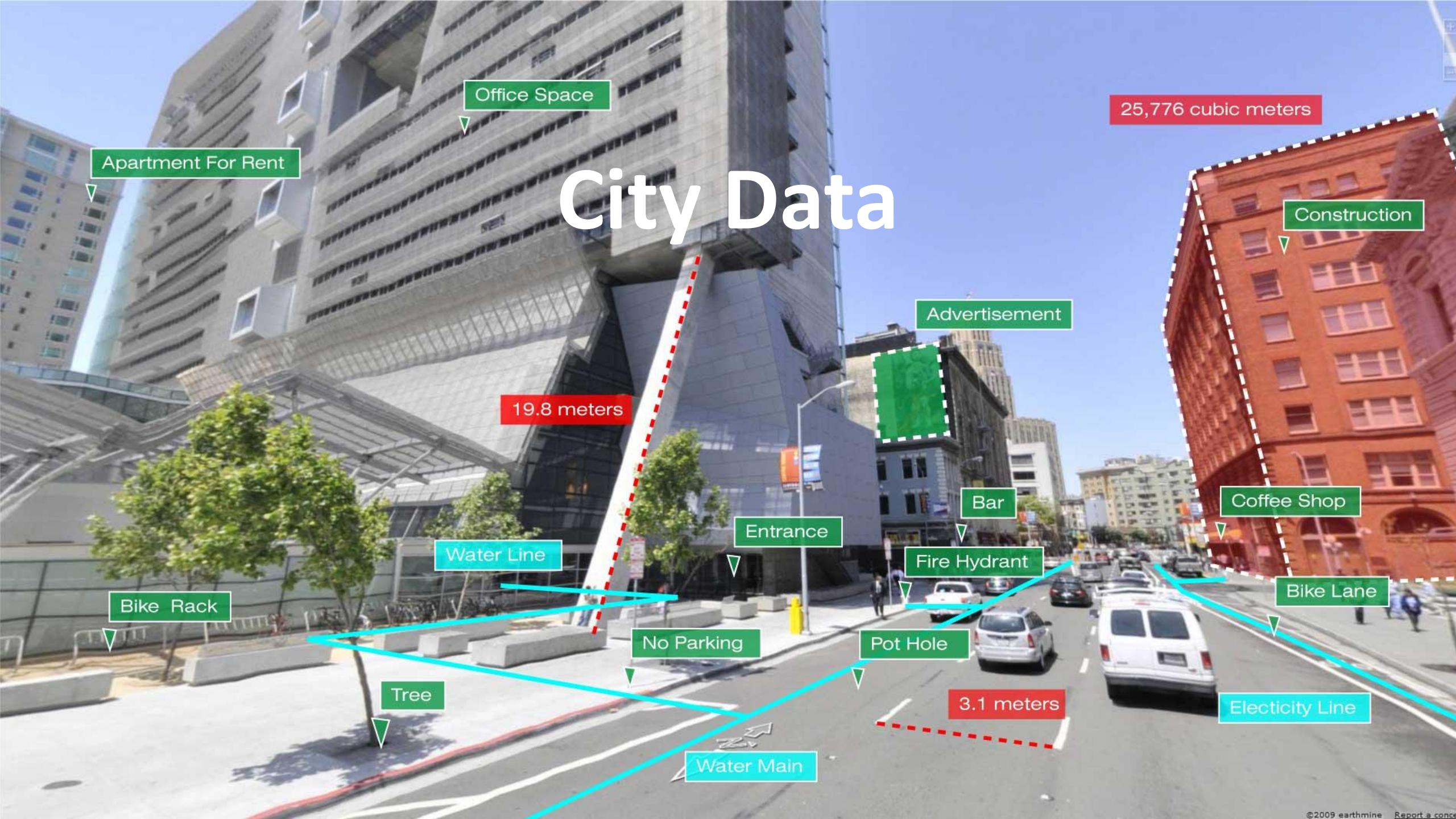


Data Driven World

Abel Sanchez, John R Williams



City Data

THE COMING FLOOD OF DATA IN AUTONOMOUS VEHICLES

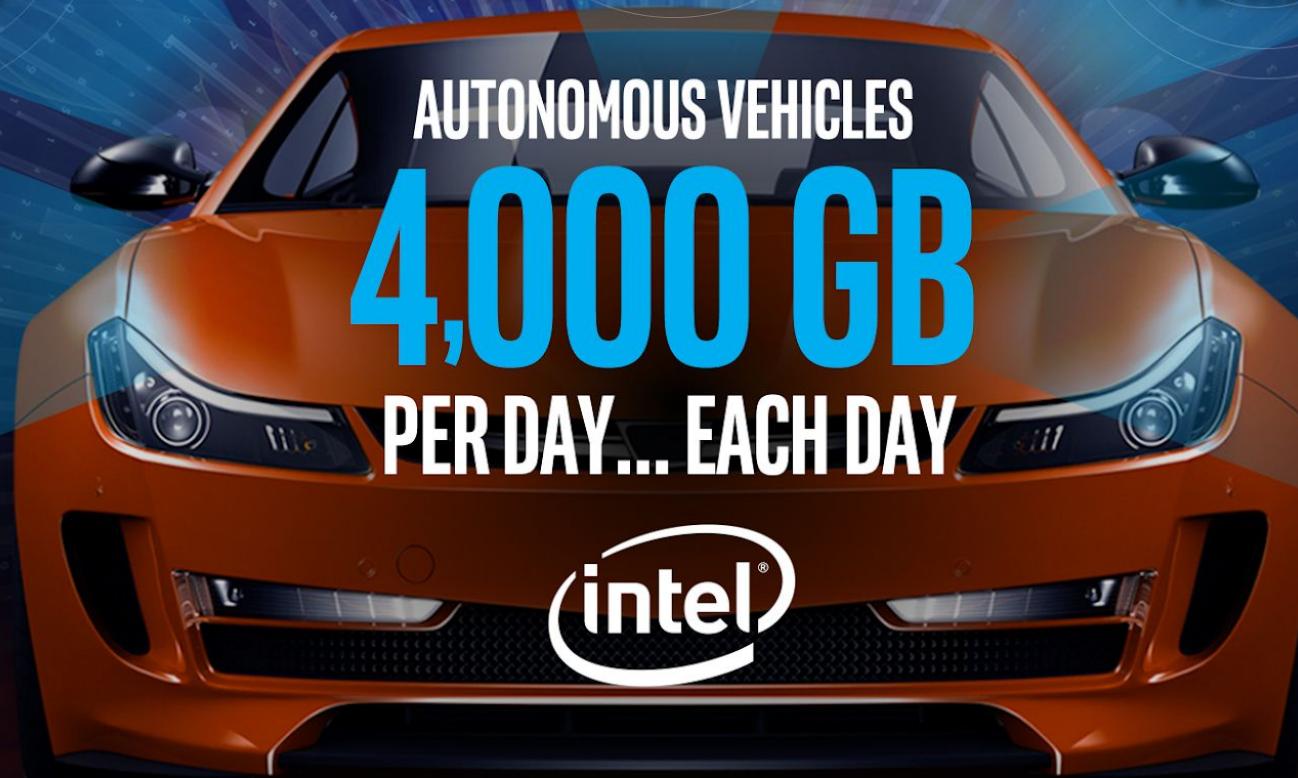
RADAR
~10-100 KB
PER SECOND

SONAR
~10-100 KB
PER SECOND

GPS
~50KB
PER SECOND

CAMERAS
~20-40 MB
PER SECOND

LIDAR
~10-70 MB
PER SECOND



AUTONOMOUS VEHICLES
4,000 GB
PER DAY... EACH DAY



transit



police



energy



health



DATA

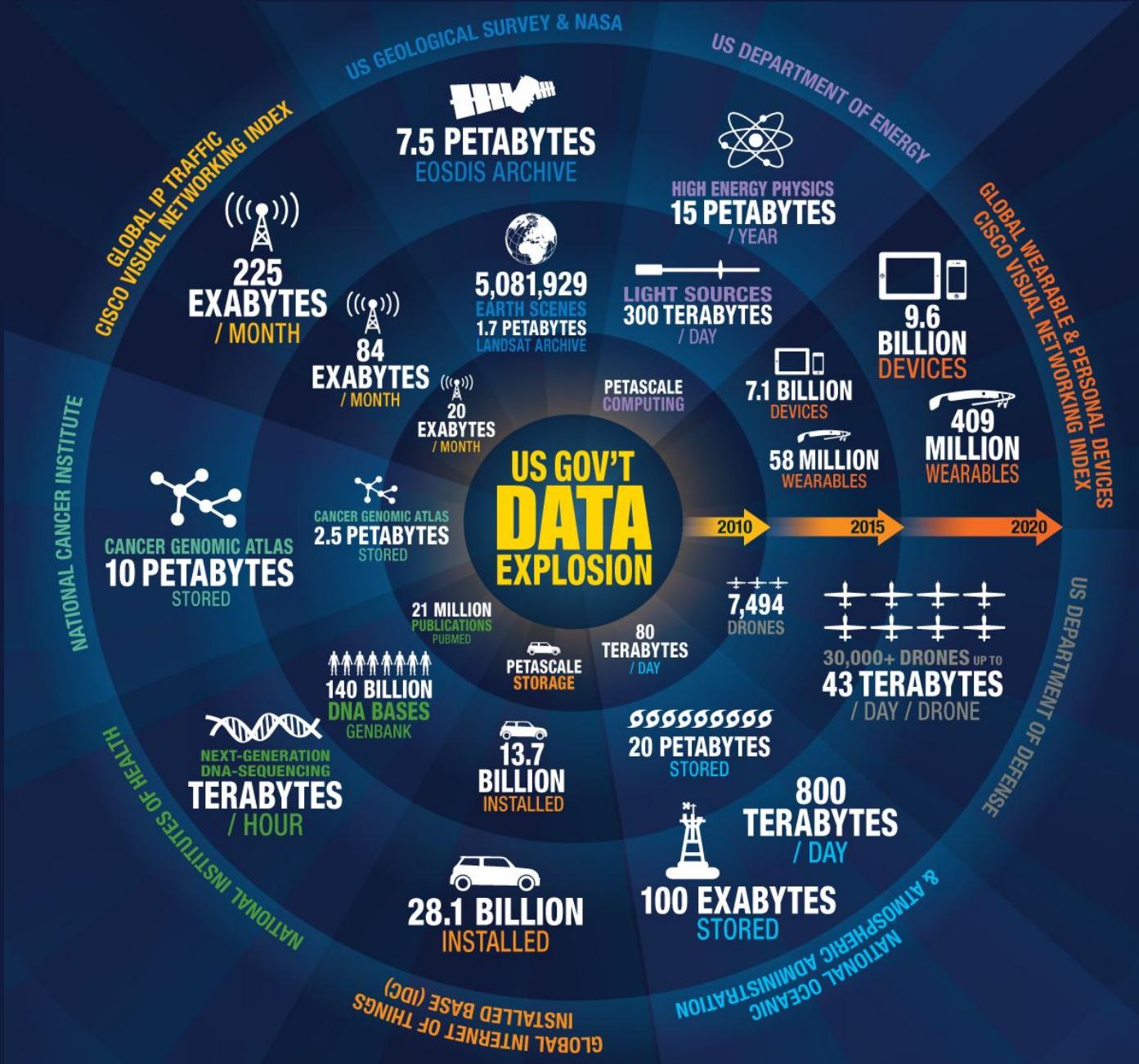


A photograph of a lush green field with rows of young, healthy plants growing in dark brown soil under a clear blue sky.

DATA



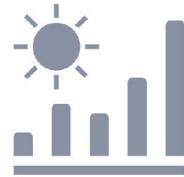
Low tech industry that will be data rich in the future? Can you think of one that will not?



DATA.GOV



Agriculture



Climate



Energy



Local
Government



Maritime

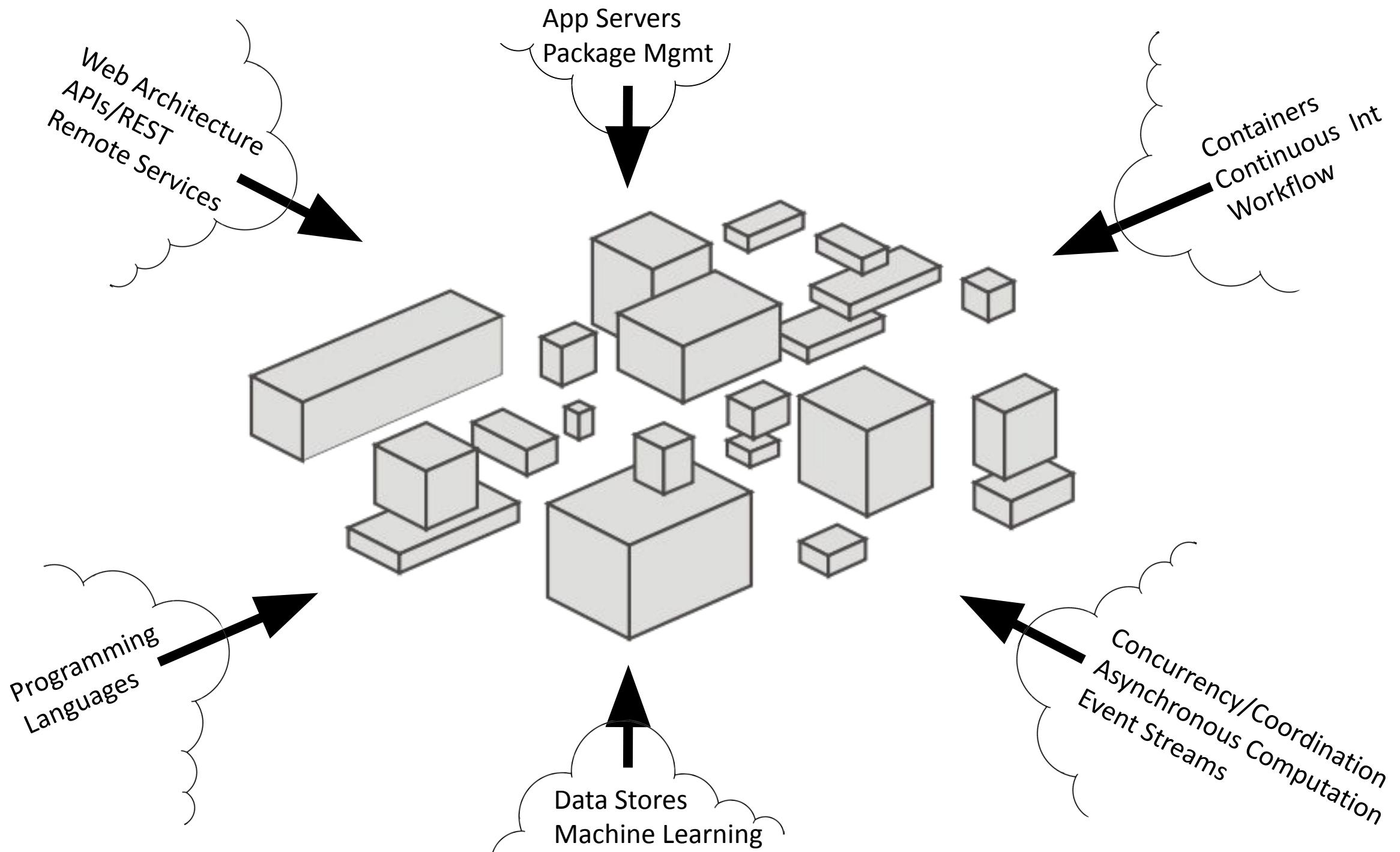


Ocean

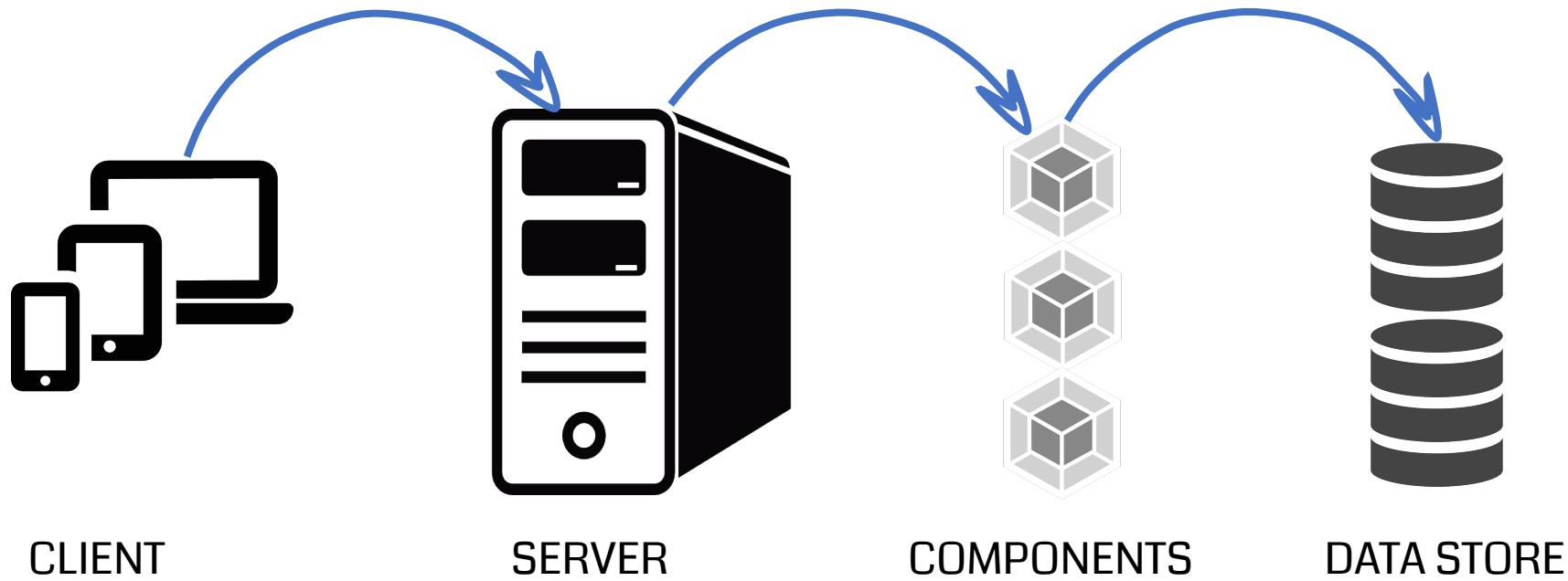
Active learning: Select something – e.g. agriculture.
Identify the data producing sub-components.

Building Systems

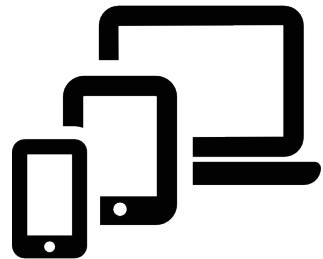
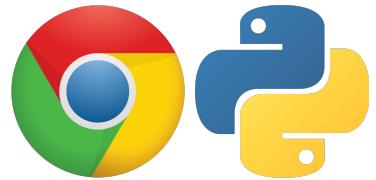
Data lives within a system



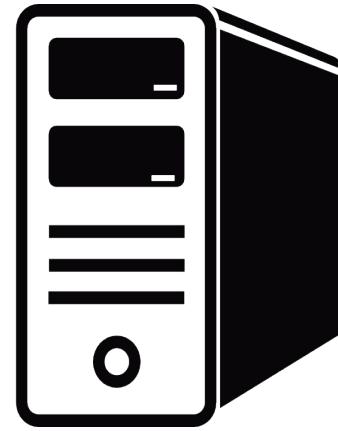
N Tiers Systems



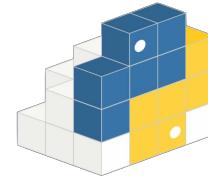
Stack: Open Source, Large Adoption, Fastest Growing



CLIENT



SERVER



COMPONENTS

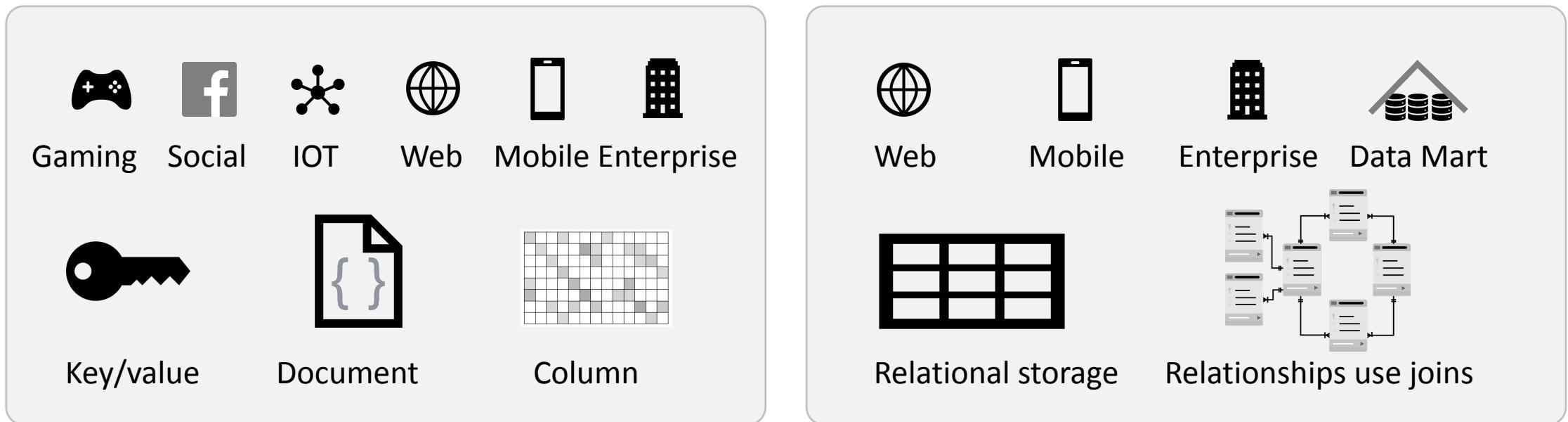
DATA
STORE



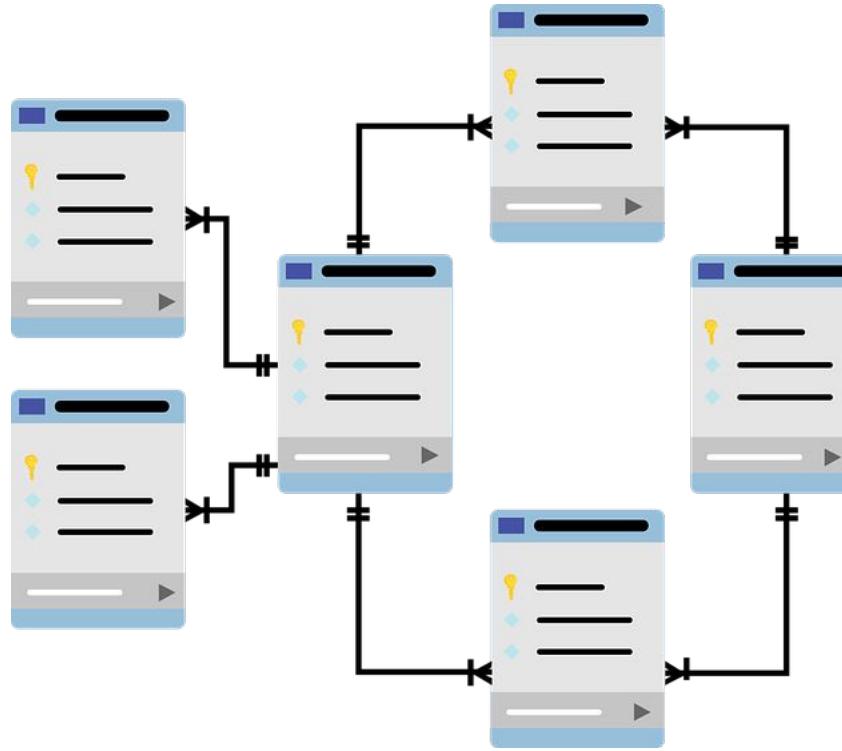
DATA STORE

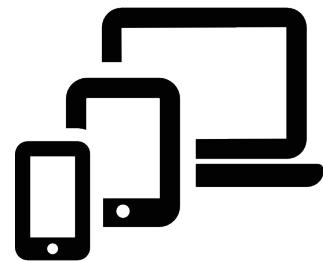
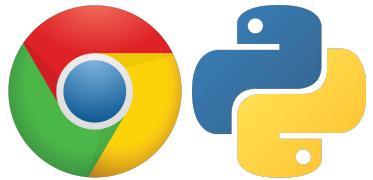


Data stores: NoSQL vs SQL

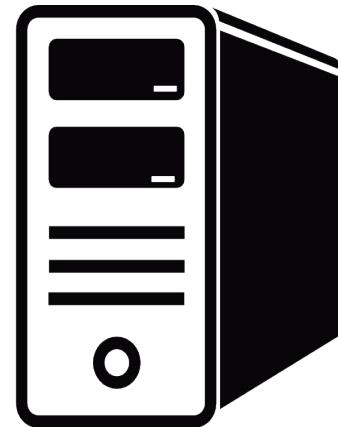


Relational Database Management System (RDBMS)

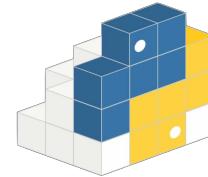




CLIENT



SERVER



COMPONENTS



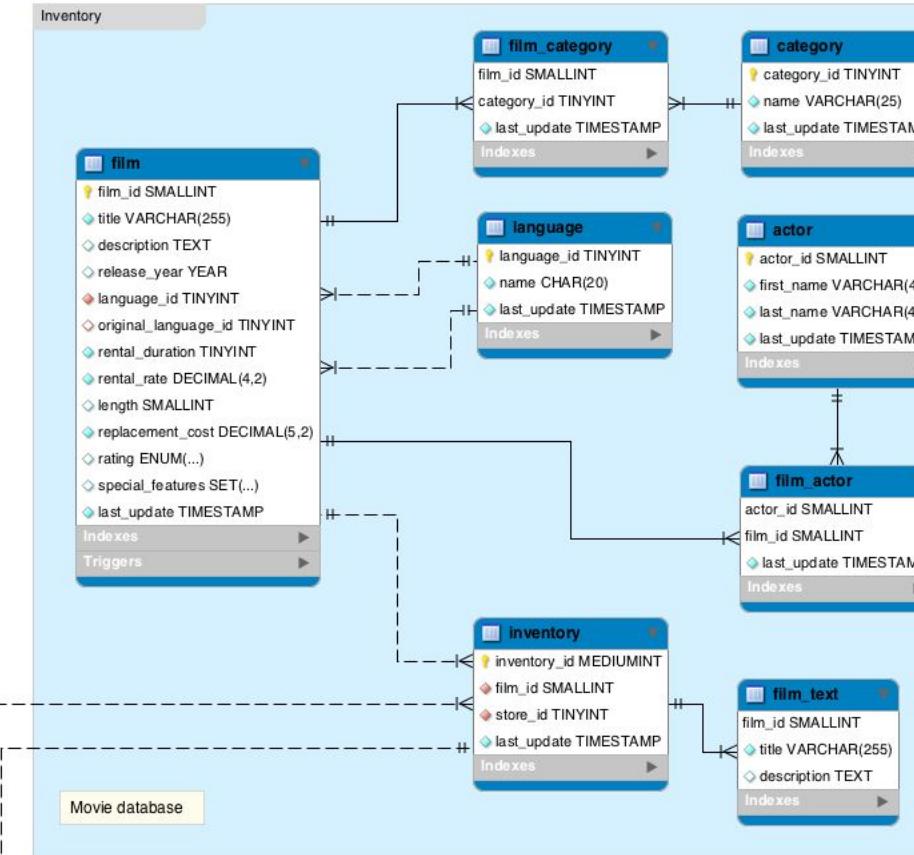
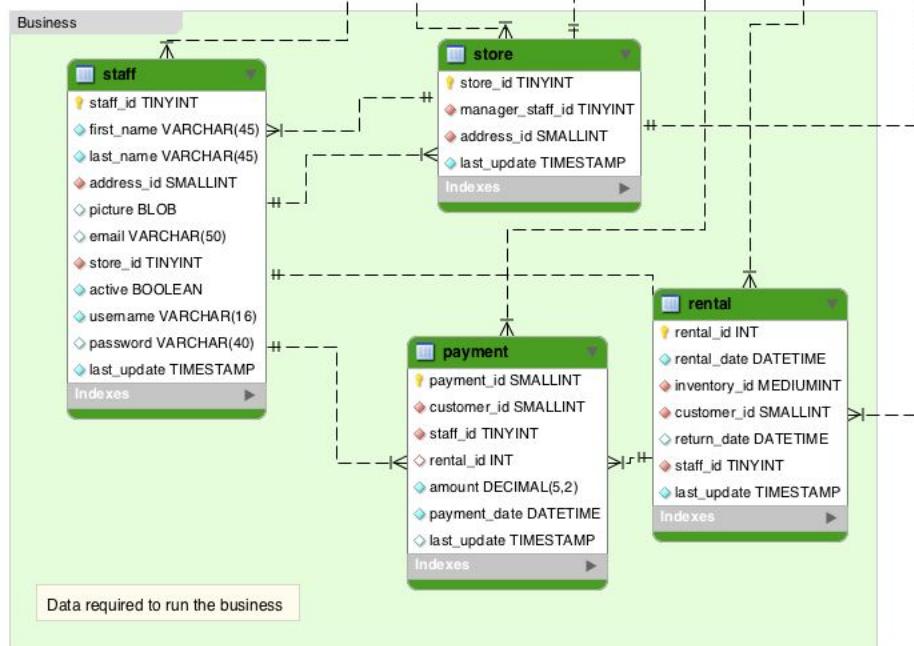
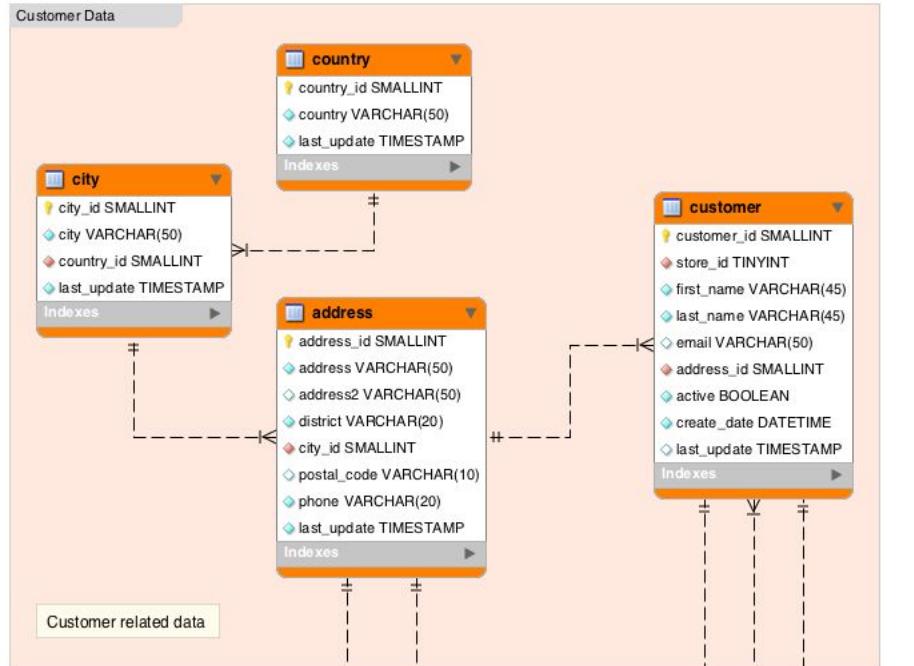
DATA STORE



Database Design

conceptual, logical, physical

Today



Conceptual

→ Entities

Logical

→ Entities, properties

Physical

→ Entities, properties, types

Relational advantage: Independence of physical and logical design

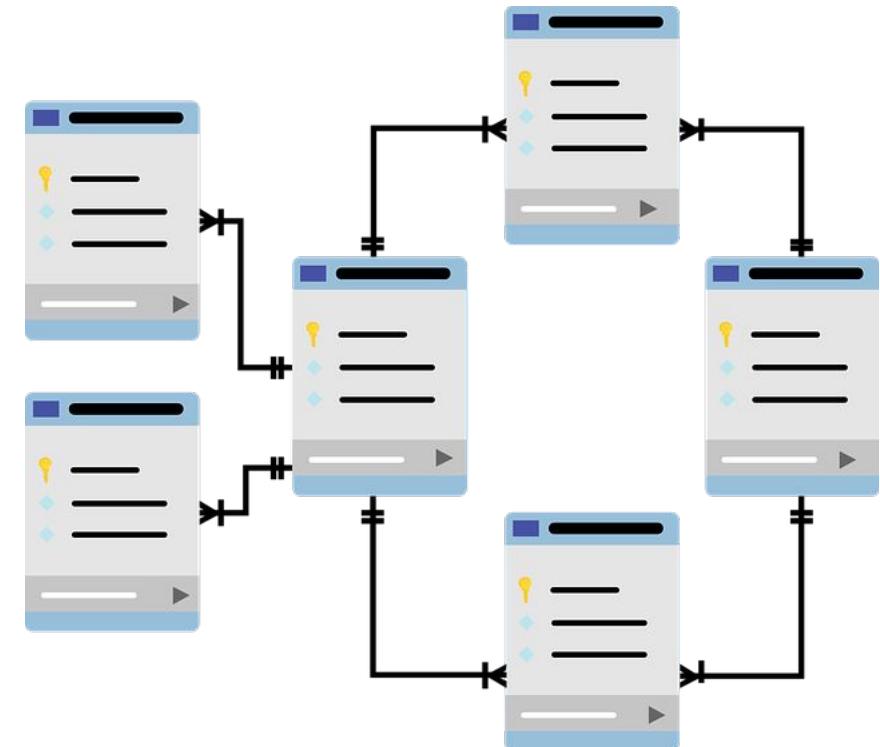
- You do not define physical storage
- You do not define the logical access paths to data
- You can answer questions on the data - others than the original plan

Downside

- No built protections against bad decisions

Database design is deciding:

- The tables
- The columns
- The relationships



Step 1: You need a deep understanding of the world you are trying to model

- Investigate
- Think about the information environment
- Where will information come from? In what form?
- How will input be done? By whom? By what?
- How frequently? How often will it change?
- You need a deep understanding of what you are trying to model!

Step 1: Understand what you model (contd)

- Review files and forms
- Track data through the organization
- Departments/functions
- Talk to data consumers, producers
- Consider what output is needed
 - Reports, purchase orders, statistical information
- Getting the data model wrong is painful

Step 2: identify entities and attributes

- An entity is a thing that has properties. For example:
 - Entity: person
 - Properties(attributes): name, DOB, email, address
- Make a list of entities and corresponding properties
- Entities are potential tables
- Properties are potential columns
- Questions to ask yourself:
 - Did I miss any entities, properties
 - Are your properties attached to the right entity – easy to get wrong

Active learning: Select something – e.g. agriculture.
Identify the data producing sub-components.

1. Understand what you are going to model
2. Identify entities and attributes

Step 3: Unique identifiers – Primary Keys

- Identify unique identifier per entity
- You will use the unique identifier to uniquely identify any row
- The unique identifier is the primary key
- If it does not exist, add column

Active learning: add unique identifiers

Step 4: Relationships

- What are your table relationships?
- Formally, cardinality is how one table relates to another.
- Cardinality can be 1-to-1, 1-to-N, N-to-N

Active learning: add relationships

Summary: Database design is Entity Relationship Modeling

- The things, the entities
- The properties of entities
- The relationships between tables

Avoid

- Forcing redundant data entry
- Increasing the risk of introducing data inconsistencies
- Increasing misunderstanding of your design
- Rigid designs that make it difficult to change the structure of tables

The Entity-Relationship Model—Toward a Unified View of Data

PETER PIN-SHAN CHEN

Massachusetts Institute of Technology

A data model, called the entity-relationship model, is proposed. This model incorporates some of the important semantic information about the real world. A special diagrammatic technique is introduced as a tool for database design. An example of database design and description using the model and the diagrammatic technique is given. Some implications for data integrity, information retrieval, and data manipulation are discussed.

The entity-relationship model can be used as a basis for unification of different views of data: the network model, the relational model, and the entity set model. Semantic ambiguities in these models are analyzed. Possible ways to derive their views of data from the entity-relationship model are presented.

Let's design a
Sample database

Book business database

- A good practice is to ask yourself sample questions:
 - Which authors live in Massachusetts
 - Which books cost more than \$15
 - Which author has written the most books
 - Which author has sold the most books
 - Which publisher has sold the most books
 - Which store has sold the most books?
 - Who has edited the most books?

Consider what policies you are hard coding

- Can author can have more than one book?
- Can a book have more than one author?
- Can a sales order request many titles?
- Can a book have more than one editor?

Entity and Relationship Modeling

- The things, the entities
- The properties of entities
- The relationships

Active Learning:
Identify entities – things that have properties

Active learning: Identify entity properties

Columns being considered

Naming Conventions

Naming considerations

- Clarity – perceived meaning
- Brevity – meaningful but not long

Naming convention

- Plural tables names
- Singular columns names
- Pascal casing

Naming convention (contd)

- Primary key → "[SingularOfTableName]ID"
 - Table name: Customers
 - Primary key: CustomerID
- Foreign key → Same name of PK
- Do not very naming
 - DateOfBirth and DOB
- Pascal naming example – DateOfBirth
- If possible, do not abbreviate

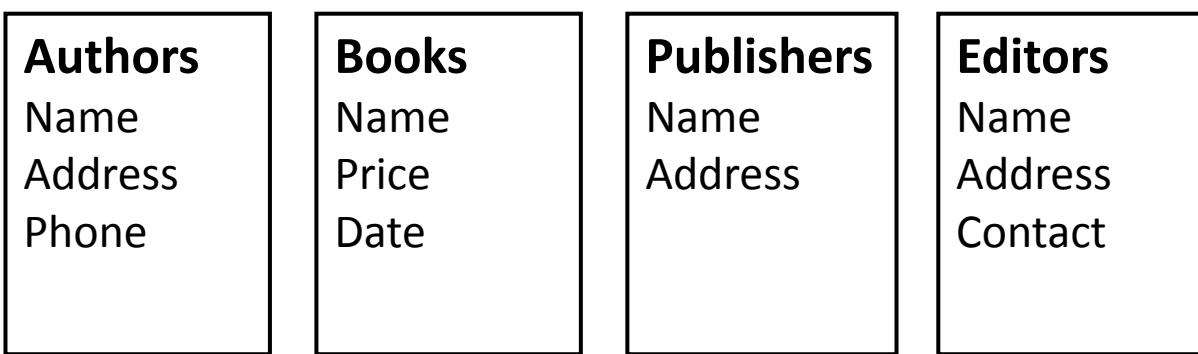
Consistency most important

- The naming convention is not just for yourself
- Like all code, it must communicate intent – you'll thank your self later
- Strive for transparency
- Like UI design, consider user expectation and knowledge
- Take into account unofficial conventions
- Above all be consistent

Sample Database

contd

Preliminary entity-relationship diagram



Primary keys, unique identifiers, for tables?

Authors
Name
Address
Phone

Books
Name
Price
Date

Publishers
Name
Address

Editors
Name
Address
Contact

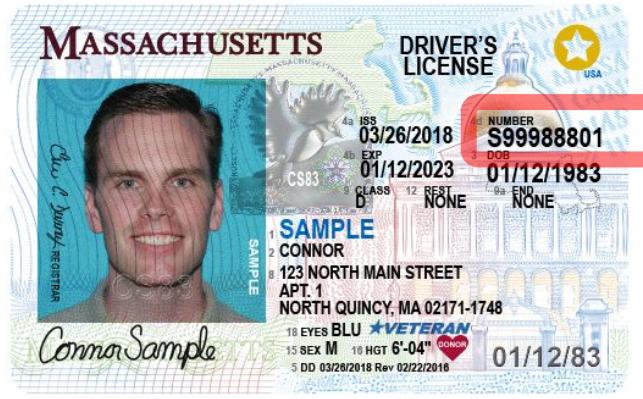
Are there problems with “name” as a primary key?

Is it a good idea to combine first and last name?

How long with that work? What else could we combine?

What could be a problem with combinations?

Numbers as PKs – a world full of numbers



Disadvantages of numbers

- Readability
- Speaking codes
- Advantage: speed, index simplicity

Active learning:

Choose primary keys

Authors

Name
Address
Phone

Books

Name
Price
Date

Publishers

Name
Address

Editors

Name
Address
Contact

Choosing PKs is one the crucial steps in database design

Cardinality

Sample Database

contd

Cardinality - how one table relates to another

- 1-to-1, one row in table A relates to one row in table B
- 1-to-N, one row in table A relates to many rows in table B
- N-to-N, Many rows in table A relate to many rows in table B

Table A
Column
Column
Column
Column

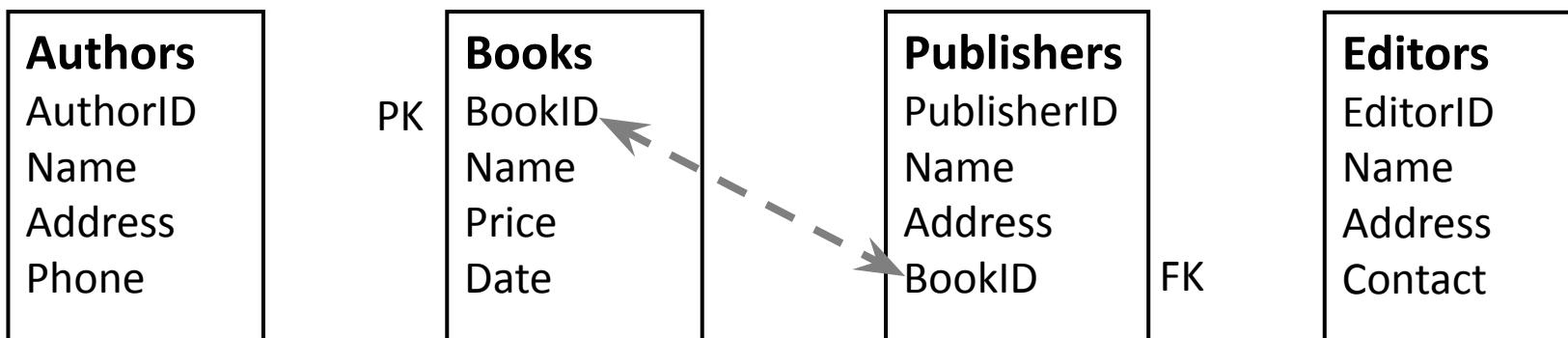
Table B
Column

So far our samples database has ...

- 4 tables: authors, books, publishers, editors
- Entity properties
- Primary keys
- But we are missing relationships between tables
 - Example: what books has a publisher published?

One-to-many relationships, 1-to-N, or 1:N

- The relationship between publishers and books is **one-to-many**.
- Would the following work?



How would you enter multiple books for the same publisher?

A better approach?

Much better

Foreign keys

- Informally, a column that matches the PK in another table.

Referential integrity

- A complete database design should include planning for PK and FK consistency
- Example
 - What happens if someone enters book with an invalid publisher ID?
 - What happens if a publisher PK is updated or deleted?
- We will return to these points

Find the many-to-many relationships, N-to-N, or N:N. Add to diagram.

- How are authors and books related?
 - Some books have more than one author
 - Some authors have written more than one book

Book authors is an N-to-N relationship

One-to-one, 1:1

- 1:1 relationships can be collapsed into one table
- Justification can be speed
- Think of a list of definitions

Entity-Relationship Summary

Entity-Relationship Summary

- Represent each independent entity as a table
- Represent each property of an entity as a column in a table
- Make sure each table has a primary key
- Identify one-to-many relationships between tables, PK & FK
- Represent each many-to-many relationship as a mapping table between two independent entities

Is your intended data model logic captured?

- Can you capture author royalty split?
- How about sales?
- Book editors?

Normalization

Normalization goals

- Reduction of data redundancy within tables
- Simpler data integrity by avoiding duplicate data
- Splitting up tables into two or more related tables that can be put back together with joins.
- Non-loss decomposition
 - Splitting up a table into smaller tables without losing information.

First normal form

- Each row-and-column cell has only one value
- Value must be atomic (one entry)
- How would handle one order with multiple books?

Second normal form

- Every non-key must depend on the entire primary key
- Every entity property must depend on the PK
- Irrelevant when the PK is only one column

Third normal form

- Non-key column must not depend on another non-key column
- Each non-key column must be a fact about the PK column

ER Diagram Symbols

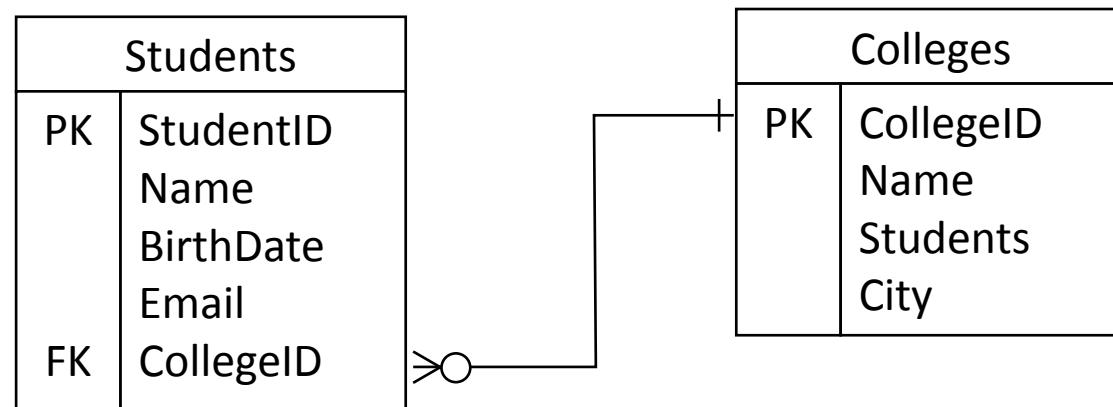
Cardinality – how one table relates to another

	One
	Many
	One (and only one)
	Zero or one
	One or many
	Zero or many

Entities, properties, PK, FK

Entity	
PK	Property Property Property Property Property
FK	Property

Example



Tools



jgraph / drawio

[Notifications](#)[Star](#)

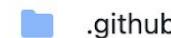
22k

[Fork](#)

4.5k

[Code](#)[Issues 567](#)[Pull requests 3](#)[Actions](#)[Projects 5](#)[Wiki](#)[Security](#)[Insights](#)[master](#)[2 branches](#)[659 tags](#)[Go to file](#)[Code](#)**davidjgraph** 14.4.4 release

cd8fb3a 4 days ago 1,544 commits



.github

14.2.6 release

2 months ago



etc

14.3.0 release

28 days ago



src/main

14.4.4 release

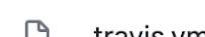
4 days ago



.gitignore

ignoing vscode config

7 months ago



.travis.yml

Adds ant install

2 years ago



CODE_OF_CONDUCT.md

Update CODE_OF_CONDUCT.md

14 months ago



ChangeLog

14.4.4 release

4 days ago



LICENSE

Changed to Apache 2.0 license

4 years ago



README.md

14.2.8 release

last month



VERSION

14.4.4 release

4 days ago

About

Source to app.diagrams.netwww.diagrams.net[Readme](#)[Apache-2.0 License](#)

Releases 659

[v14.4.4](#) Latest

4 days ago

[+ 658 releases](#)

Packages 1

[com.github.librepdf.openpdf 1.3.23-jgraph](#)



S

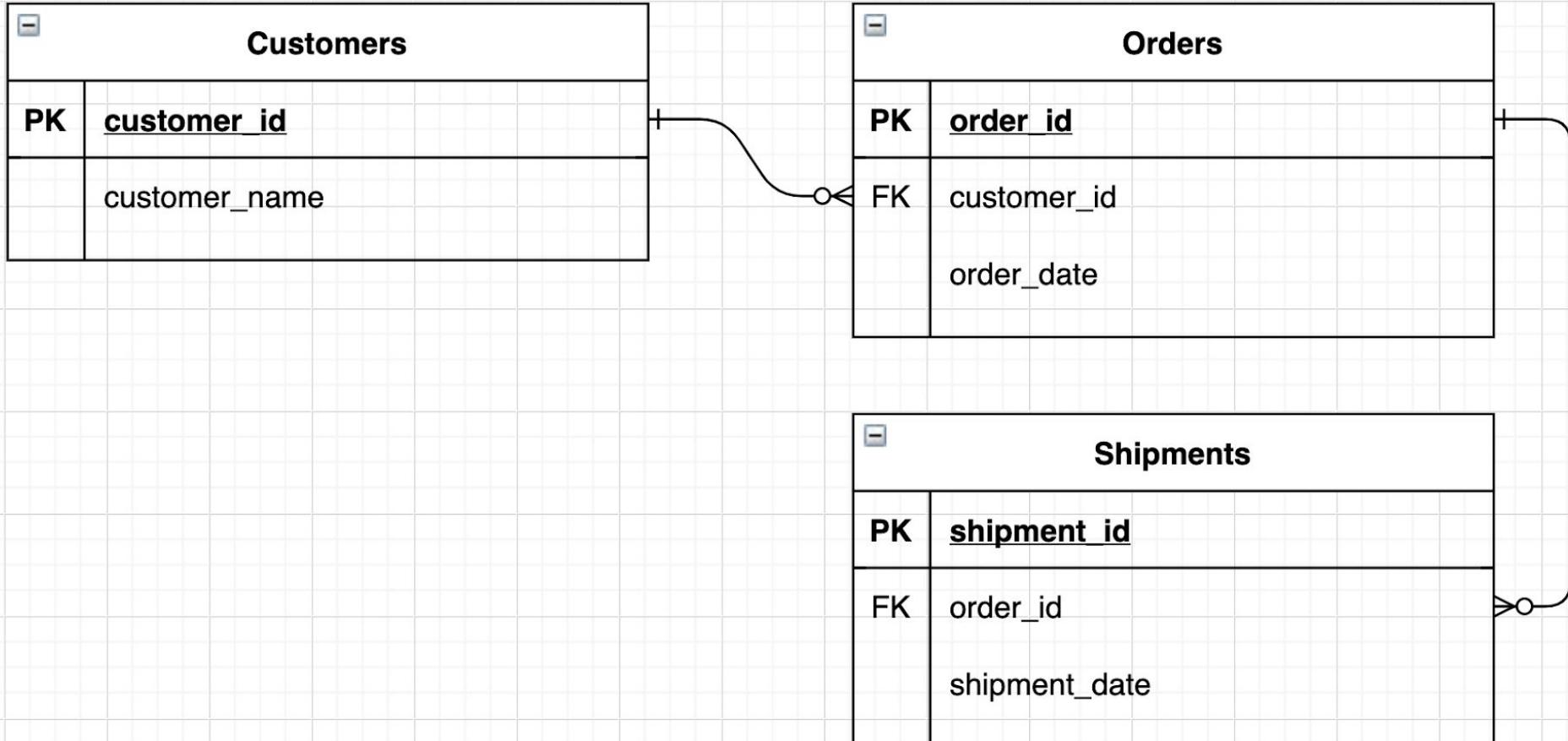
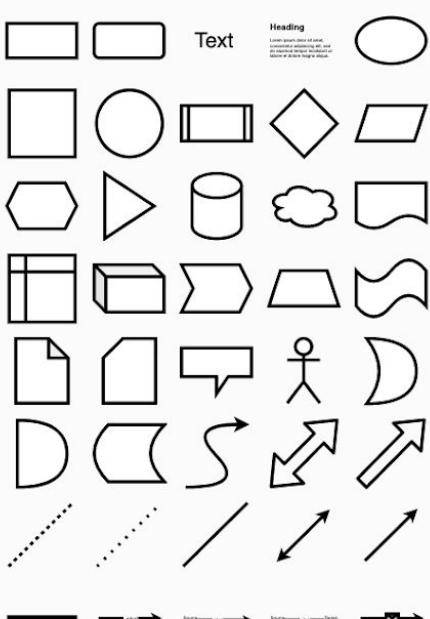
Search Shapes

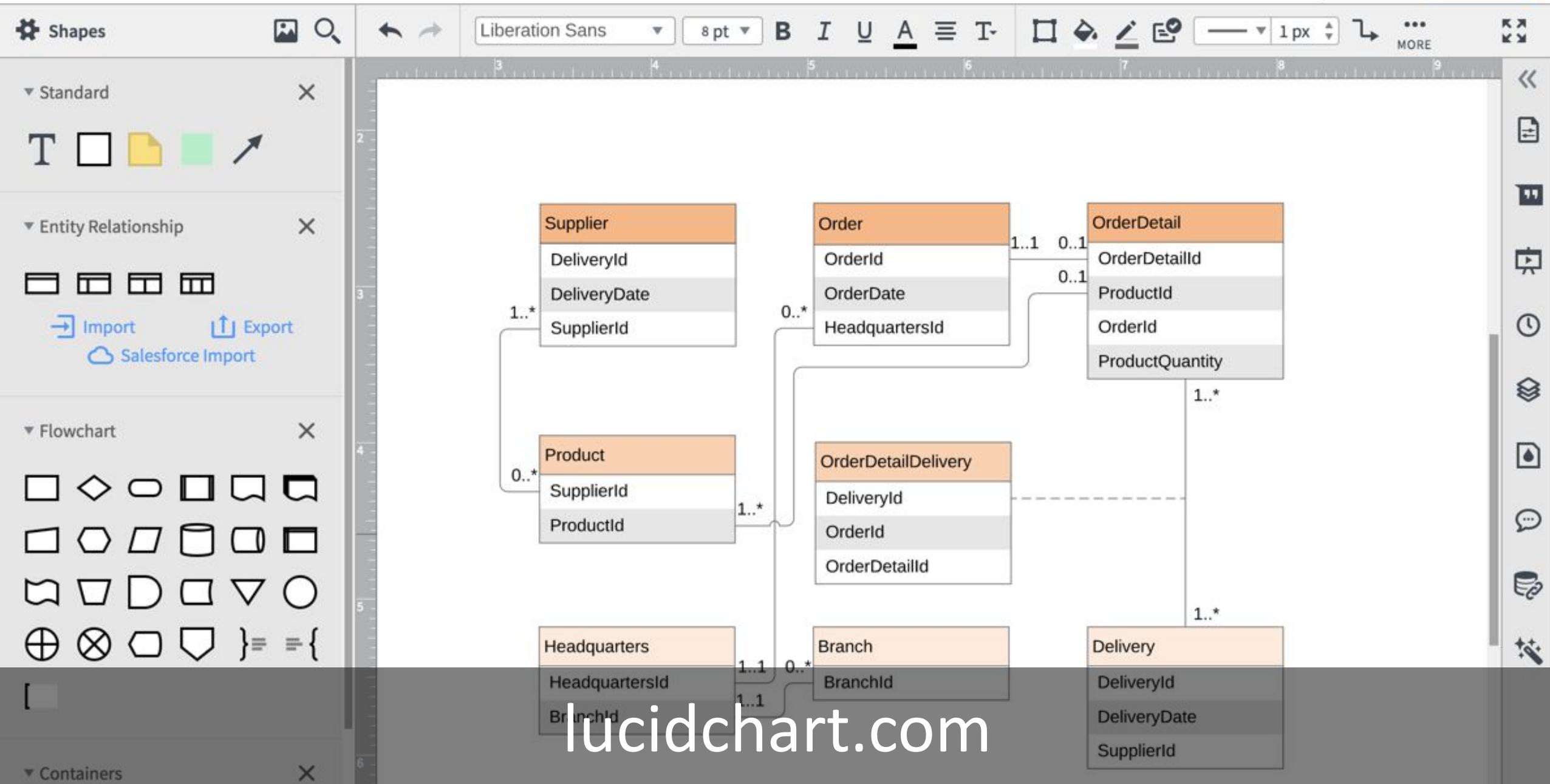


Scratchpad



General

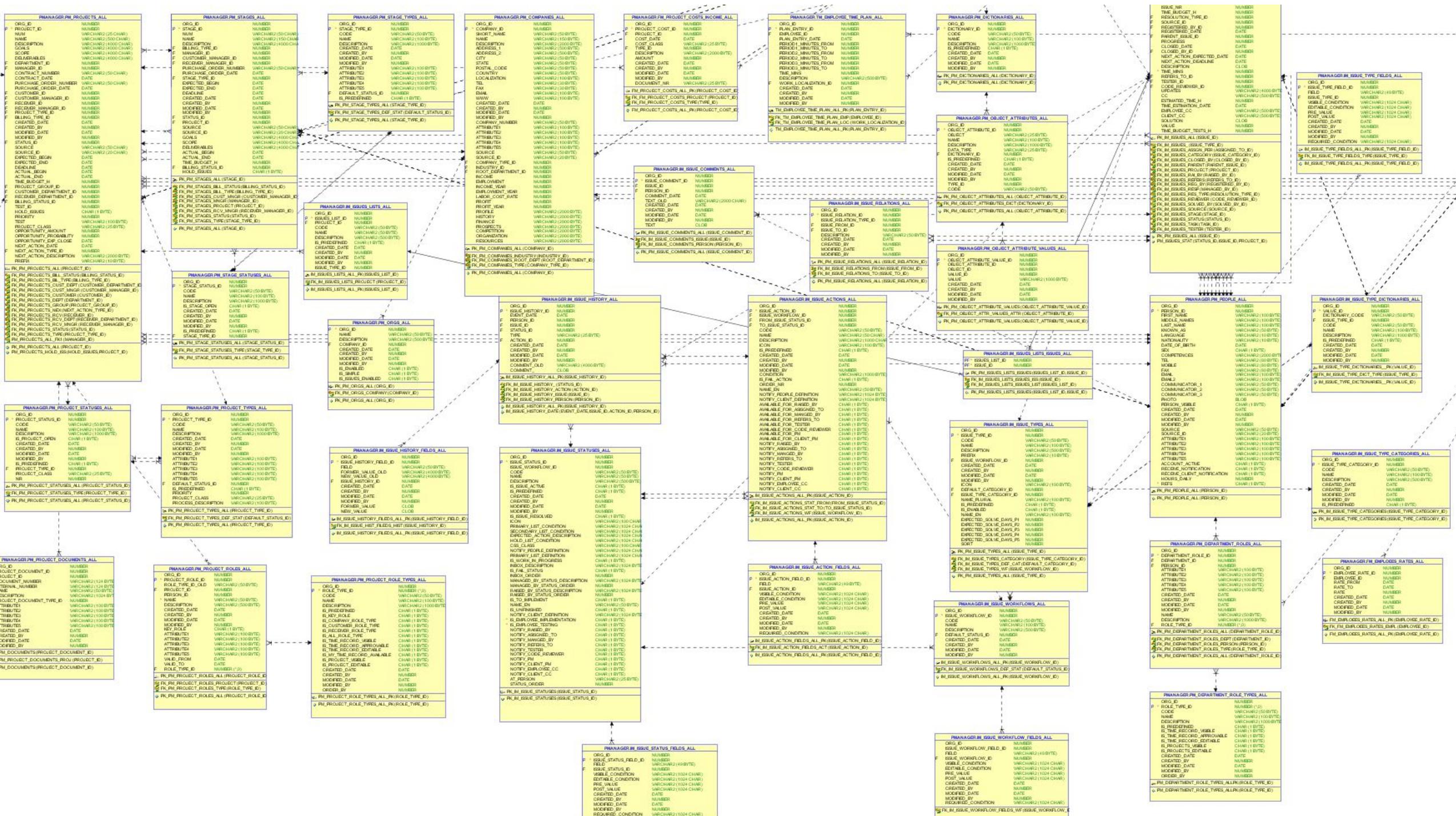




Active learning

- Finish Entity-Relationship diagram
- Submit diagram
- <http://bit.ly/3bnw3WG>

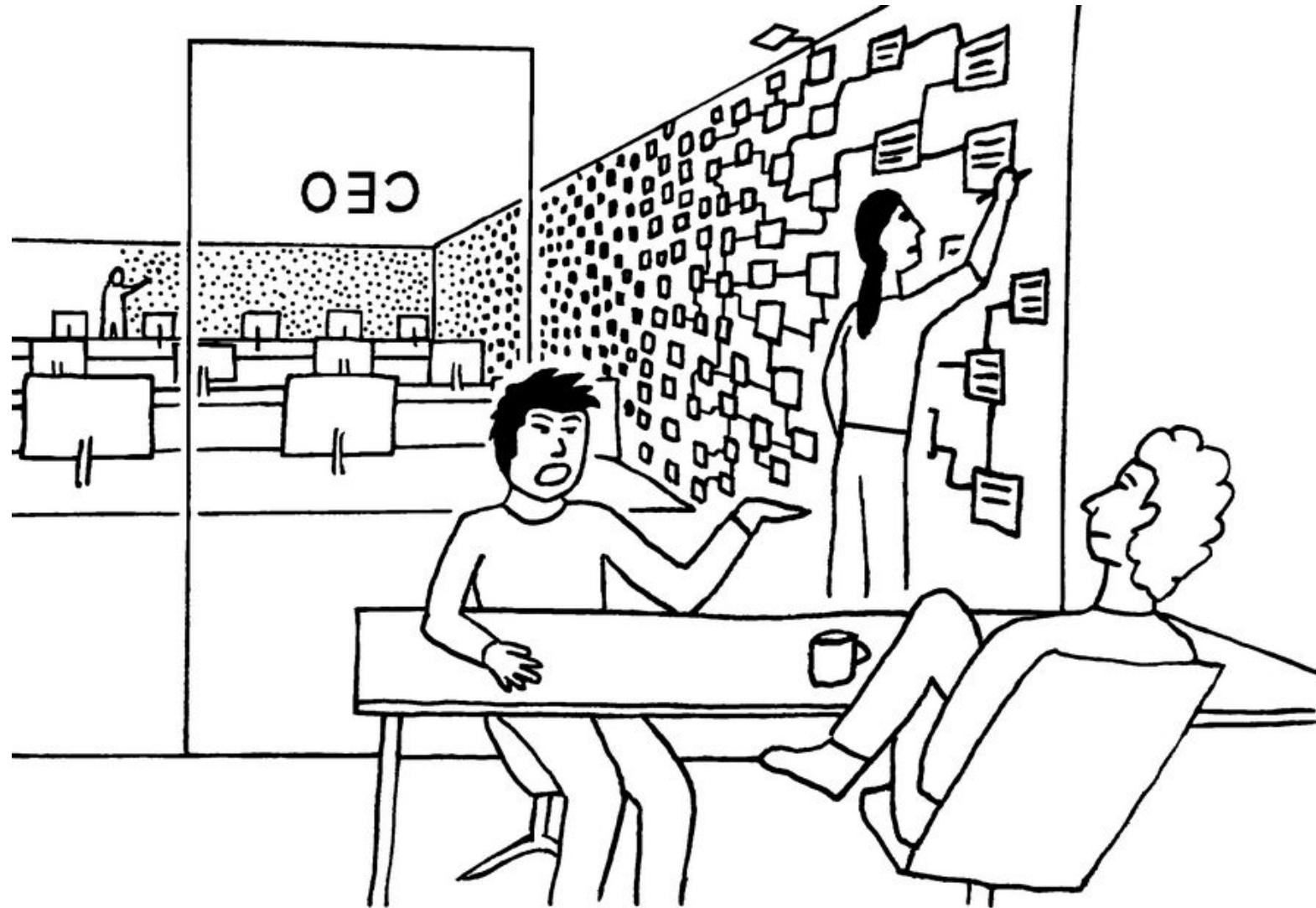
Unified Vision



PORTAL 7.3 ENTITY RELATIONSHIP DIAGRAM



INDEX



Oracle EBS
24,000 tables

Remember when you asked us to create an entity-relationship diagram for the whole company? Well, we're going to need a bigger office ...

Privacy

Data trusts

Any entity wishing to place a sensor would have to request a license to both collect and use data. A review board, made of community members, would monitor and enforce that collection and use

TechReview, <https://bit.ly/3kXI7C0>