Cours 01

Un tour d'horizon sur l'évolution des systèmes de base de données

Abdelkader OUARED

a_ouared@esi.dz

Objectifs du cours

- Présenter la notion de cycle de vie de base de données
- Découvrir l'évolution des systèmes de bases de données
- Donner un aperçu des principaux modèles de donneés

Turing Awards in Data Management

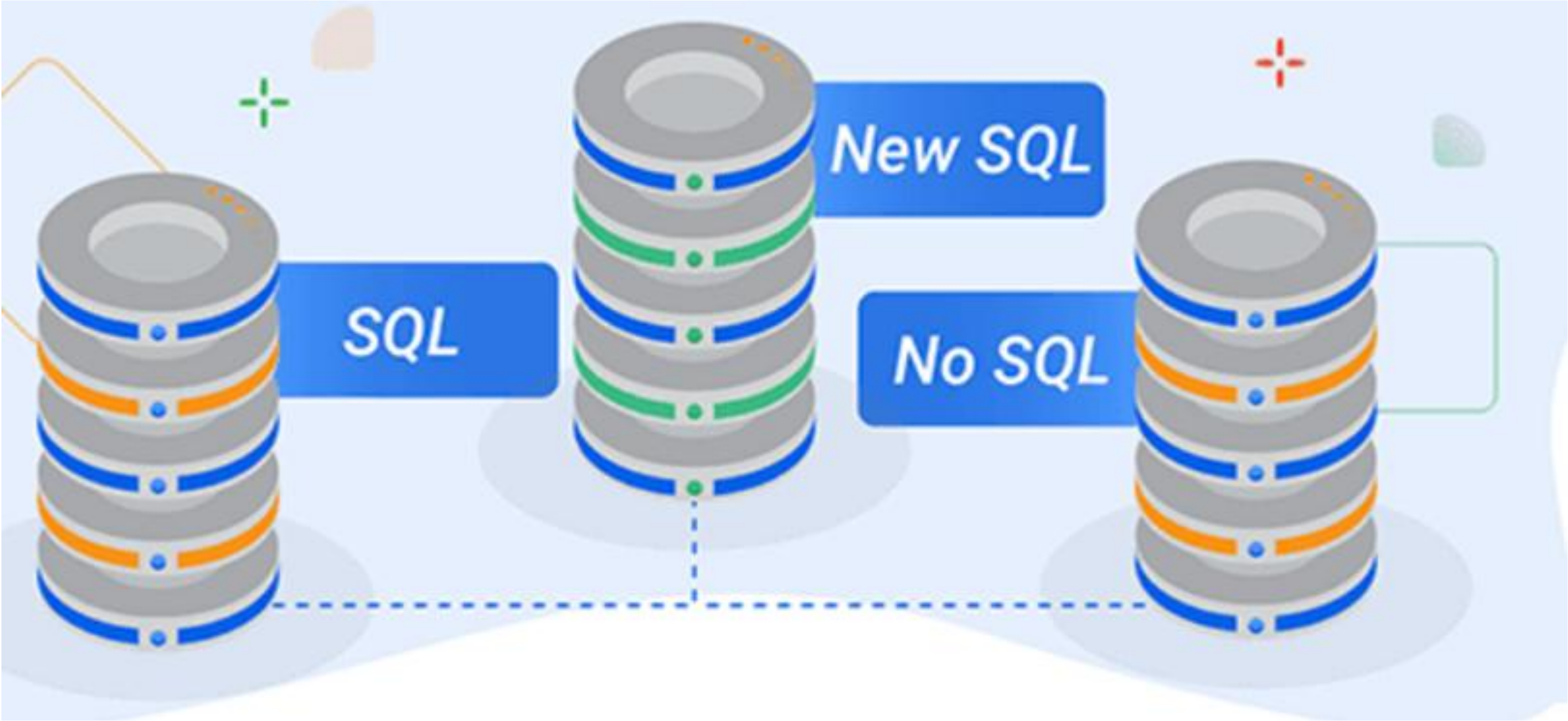
Charles Bachman, 1973
IDS and CODASYL

Ted Codd, 1981
Relational model

Jim Gray, 1998
Transaction processing

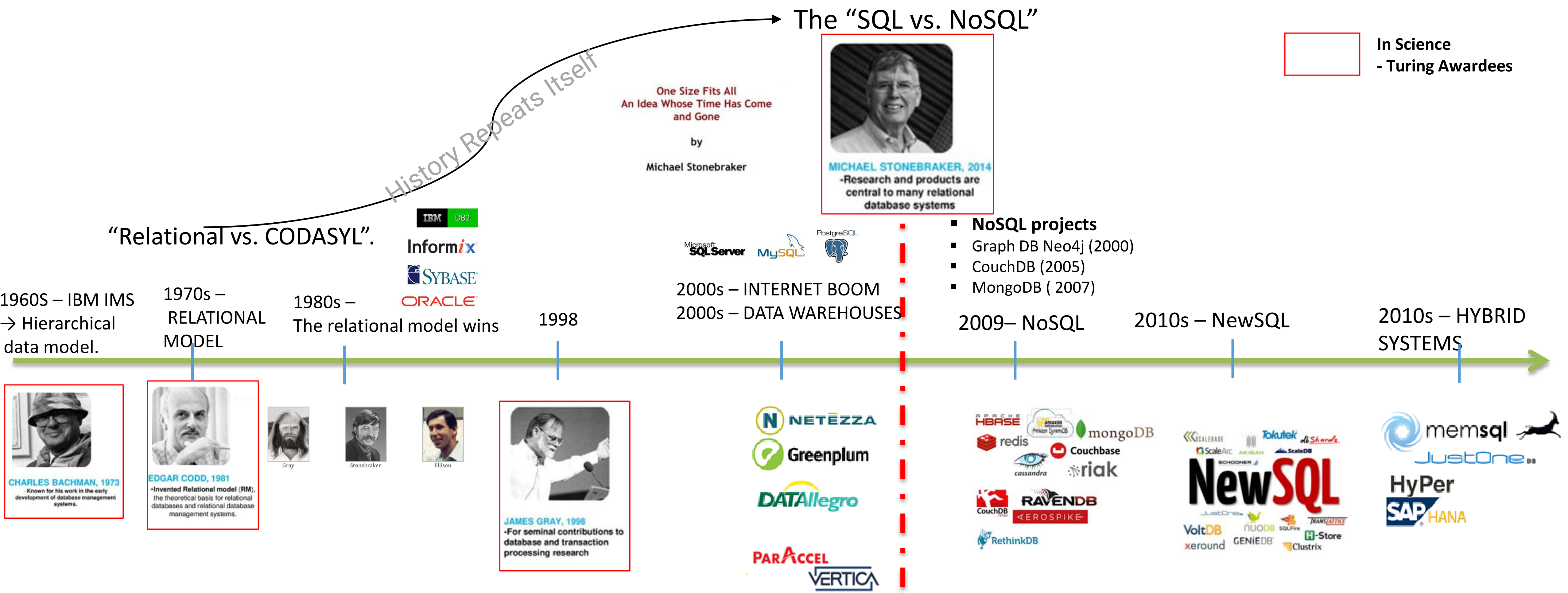
Michael Stonebraker, 2014
INGRES and Postgres

Evolution des systèmes de bases de données (SQL, NoSQL, NewSQL)



Database System Evolution: ~ A Long Story

In Science & In Industry



Database System Evolution: In Industry

331 systems in ranking, August 2017

Rank Aug 2017	Rank Jul 2017	Rank Aug 2016	DBMS	Database Model	Score Aug 2017	Score Jul 2017	Score Aug 2016
1.	1.	1.	Oracle	Relational DBMS	1367.88	-7.00	-59.85
2.	2.	2.	MySQL	Relational DBMS	1340.30	-8.81	-16.73
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1225.47	-0.52	+20.43
4.	4.	↑ 5.	PostgreSQL	Relational DBMS	369.76	+0.32	+54.51
5.	5.	↓ 4.	MongoDB	Document store	330.50	-2.27	+12.01
6.	6.	6.	DB2	Relational DBMS	197.47	+6.22	+11.58
7.	7.	↑ 8.	Microsoft Access	Relational DBMS	127.03	+0.90	+2.98
8.	8.	↓ 7.	Cassandra	Wide column store	126.72	+2.60	-3.52
9.	9.	↑ 10.	Redis	Key-value store	121.90	+0.38	+14.57
10.	10.	↑ 11.	Elasticsearch	Search engine	117.65	+1.67	+25.16

Diversité de SGBD: DB-Engine*: **316** DBMS en 2016

***http://db-engines.com/en/ranking**

Cycle de vie de base de données

Today

➔ As data management became more complex ☹

real world systems
and processes

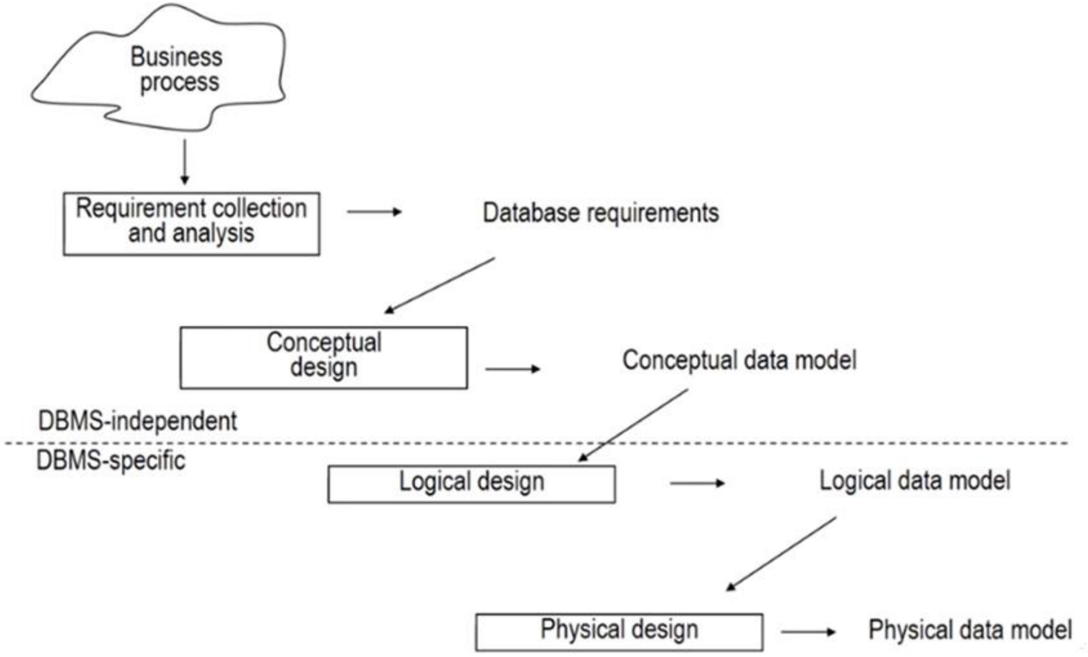
Transformation

a framework
was need

Structures: be implemented in a computer ☺

ANSI/X3/SPARC, 1978:

- External model
- Conceptual data model
- Logical data model
- Internal data model



⇨Les **modèles** sont au **centre** du processus de conception de BD

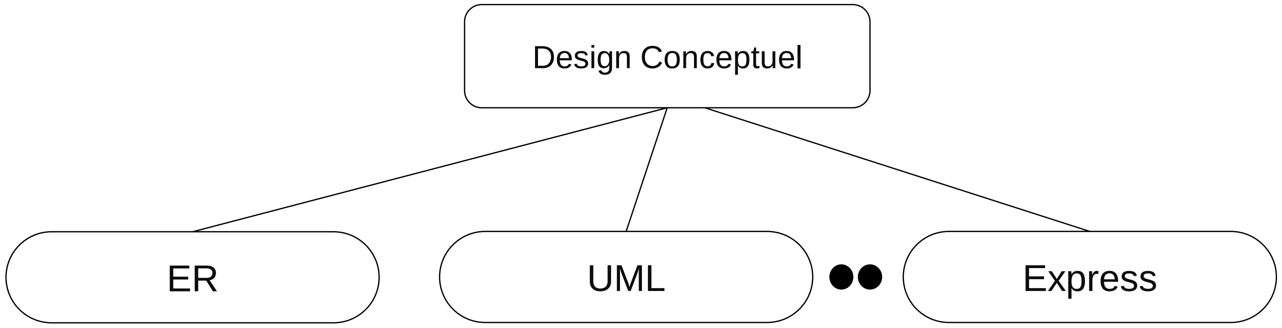
Modèle de données (Data Model)

- It defines the structure of database:
- A database model shows the **conceptual/logical and physical** structure of database
  - How databse is modeled ?
  - How data is connected to each other?
  - Describes the **data relationships**, the **semantics** and the **data integrity constraints**
  - Show how data can be organized, stored and processed in system?
- **Categories of Data Models**
  - **Object based logical Model** (e.g. ER Model, Object Oriented Model)
  - **Record Based logical Model** (e.g. Relational Model)

Modèles conceptuel de données

❑ **Phase conceptuelle:**

- Représentation structurée et abstraite de données
- N'inclut pas de détails sur la manière dont les données sont stockées ou les opérations sont implémentées.
- **Langages de modélisation** : *Entité/Association (EA), UML, Express, etc.*
- L'expressivité et la qualité du modèle conceptuel dépend fortement des compétences du concepteur, plutôt que du formalisme utilisé
- Cette phase ne peut s'automatiser totalement



Le Modèle Entité – Association (Entity Relationship (ER) Model)

Entity Relationship (ER) Model by Peter Chen

- Born in Taiwan
- Ph.D from Harvard University in 1973
- Professor at Louisiana State University

A Relational Model of Data for Large Shared Data Banks

1970

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

1976

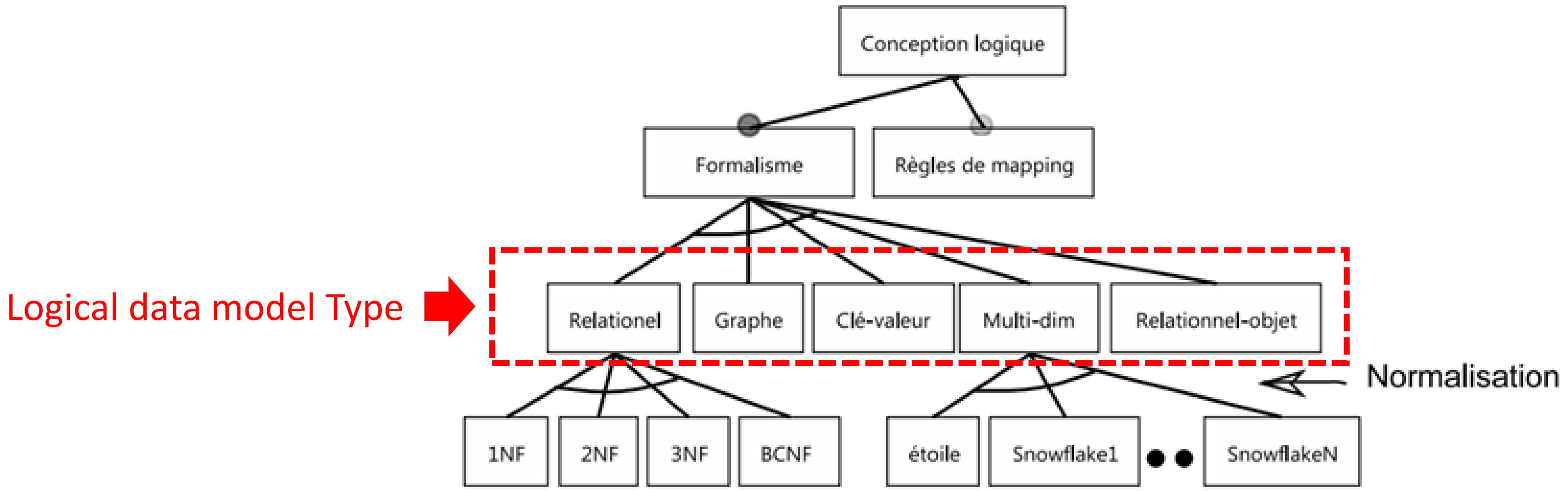
- Formalisé en **1976** par **P. Chen**
- Ensemble de concepts pour modéliser les données d'une application d'une entreprise (les liens, la sémantique et les contraintes.)
- Etendu vers E/R généralisé puis vers l'objet
  - L'approche parait plus naturelle → obtention direct de résultat
  - Outil de communication efficace
  - Représentation graphique accessible pour les utilisateur finaux

Peter P. Chen: **The Entity-Relationship Model - Toward a Unified View of Data. ACM Trans. Database Syst. 1(1): 9-36 (1976)

Types des Modèles de données: **Modèle logique**

❑ Phase logique

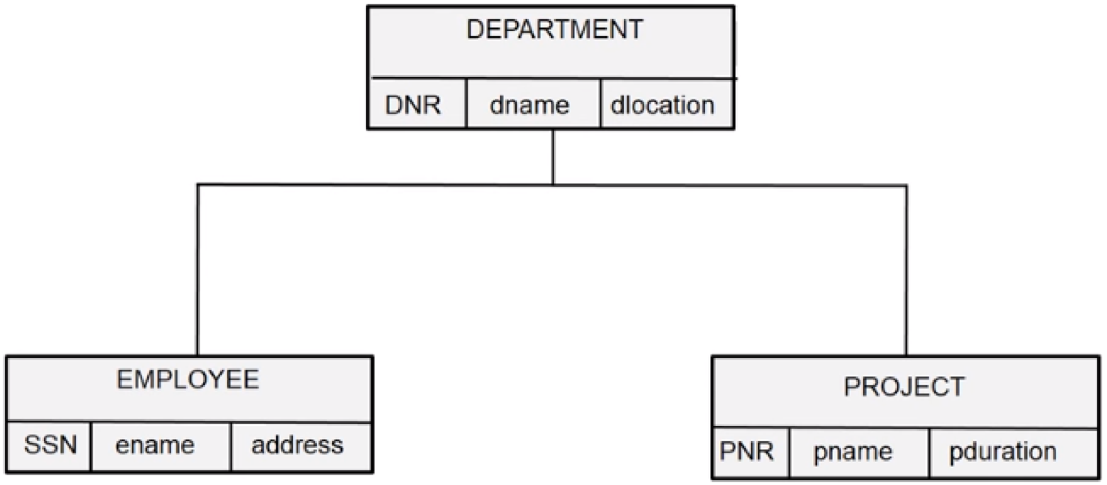
- transformation (**mapping**) du modèle conceptuel en un modèle logique lié à l'implémentation de BD dans un SGBD,
- Logical data model : provide the explicit forms **(data structure)** that the conceptual models can take and is the first step in computing (e.g. hierarchical, network, relational; etc.).
- Cette phase peut ainsi être automatisée par les outils de conception de BD (outils **CASE**)
- formalismes existent pour représenter logiquement les données:
  - **i.** **modèles orientés enregistrements** : le relationnel, multidimensionnel et réseau,
  - **ii.** **orientés-objet :** l'objet et le relationnel-objet, et
  - **iii.** **orientés NoSQL** : les modèles clé-valeur, graphes, documents ou colonnes .



Modèle logique (Hierarchical data model)

Hierarchical Data Model:

- In hierarchical model, records are organized as trees
- Each entity has only one parent but can have several children



HIERARCHICAL DATA MODEL

- ❑ Duplicate Data
- ❑ No Independence

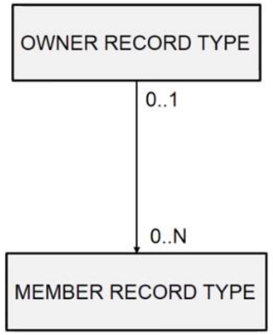
Modèle logique (Network data model)

CHARLES BACHMAN , 1973

1960S – IBM IMS
First database system developed to keep track of purchase orders <u>for Apollo moon mission</u>.
→ Hierarchical data model.
→ Programmer-defined physical storage format.
→ Tuple-at-a-time queries

1970s – CODASYL
COBOL people got together and proposed a standard for how programs will access a database. **Lead by Charles Bachman.**
→ Network data model.
→ Tuple-at-a-time queries

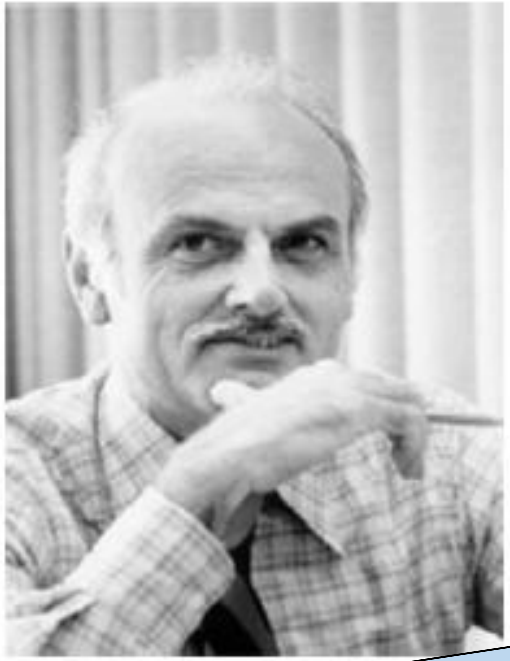
- <u>**Network Data model**</u> - many to many relationships.
  - In this data is represented by collection of records and relationships among data are represented by links
  - Record linked together like a family tree
    - Each record type can have **more that one** owner
  - <u>Record</u> it is a call of attributes, each of which contains only one data value
  - **Link** it is an association **between two records**



\+ reduces redundancy.
\+ more flexible paths to data.
\+ very fast
\- Complex Queries: pointers expensive and difficult to update when inserting and deleting

Modèle logique (Relational Model)

- Les recherches de Codd 1970 : (IBM Research).
- Structure d'une BD relationnelle (ensembles de n-uplets/tables)
- Simple et bien formalisé
- Basé sur l'algèbre relationnelle et le et langage SQL,
- Utilisé par la majorité des systèmes actuels (70% de SI)
- Théorie de la normalisation: 1NF, 2NF, 3NF, BCNF, 4NF, et 5NF



Codd Edgar

HISTORY REPEATS ITSELF

Old database issues are still relevant today.
The "**SQL vs. NoSQL**" debate is reminiscent of "**Relational vs. CODASYL**" debate.
Many of the ideas in today's database systems are not new

1970s – RELATIONAL MODEL

- Early implementations of relational DBMS:
  → **System R** – IBM Research
  → **INGRES** – U.C. Berkeley
  → **Oracle** – Larry Ellison

Gray

Stonebraker

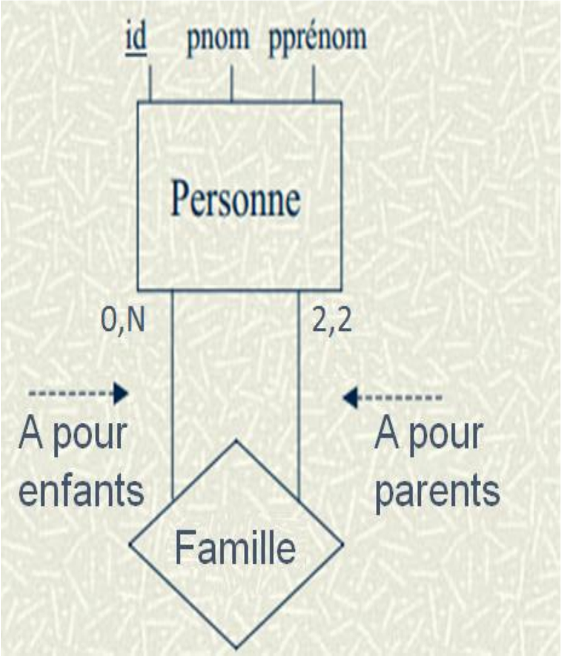
Ellison

1980s – RELATIONAL MODEL

- The relational model wins.
  → IBM comes out with DB2 in 1983.
  → SQL becomes the standard.
  Many new “enterprise” DBMSs
  but Oracle wins marketplace.
  Stonebraker creates Postgres.

Exercice1: Passage de modèle conceptuel vers le modèle Relationnel

Donnez le modèle Relationnel qui correspond au modèle conceptuel suivant :



Entity Relationship (ER) Model

Personne (<u>id</u>, nom, prénom)

Parent (<u>idPère</u>, IdMère, #id)

Enfant (<u>idE</u>, #id)

Relationnel Model

1980s – OBJECT- ORIENTED DATA BASES

- Avoid “relational-object impedance mismatch” by tightly **coupling objects and database**.

- Few of these original DBMSs from the 1980s still exist today but many of the technologies exist in other forms (JSON, XML)

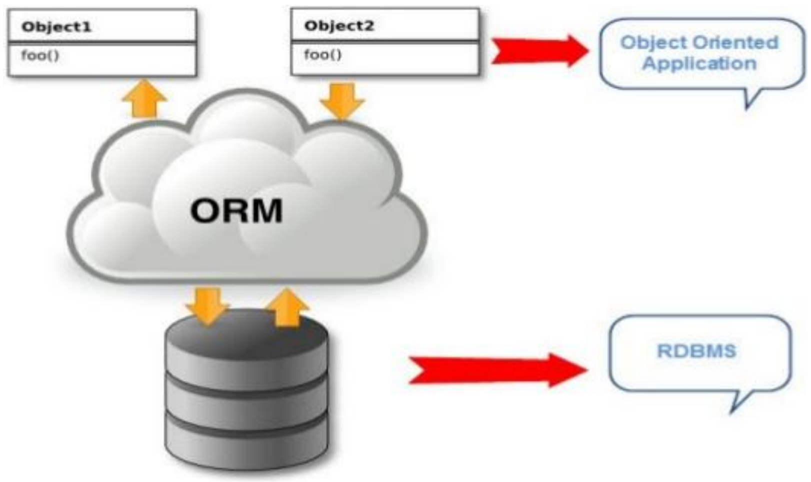
Exercice2: Des diagrammes objets vers le modèle Relationnel

Exercice 2 (6pts) : On veut définir en SQL3 le schéma d'une base de données touristique. On veut définir un type Ville, qui a un nom, des restaurants et des musées. Un restaurant a un nom, une adresse et une liste de 3 menus. Les types Musée et Menu sont définis de la manière suivante :

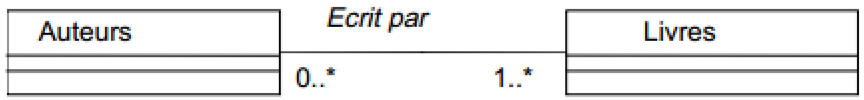
```
Create type Musee as object(
Nom varchar(20),
JourFermeture varchar(15) ) ;

Create type Menu as object (
Nom varchar (20),
Prix Number (2) ) ;
```

1. Définir le type Restaurant
2. Définir le type Ville
3. Définir la table permettant de stocker toutes ces données.
4. Donnez l'expression en SQL3 des requêtes suivantes :
   - Nom des villes ayant un musée ouvert le lundi.
   - Nom et adresse d'un restaurant à Oran où on peut manger pour moins de 3000 DA

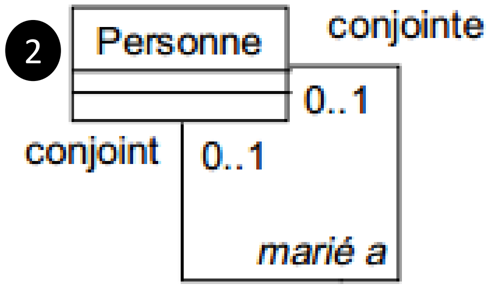
Exercice3: des diagrammes objets vers le modèle Relationnel



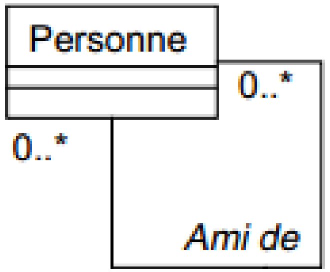
1



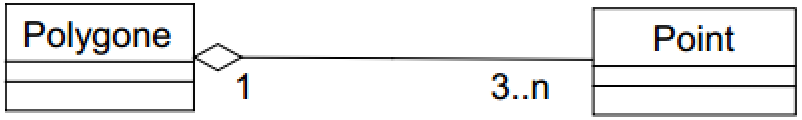
2



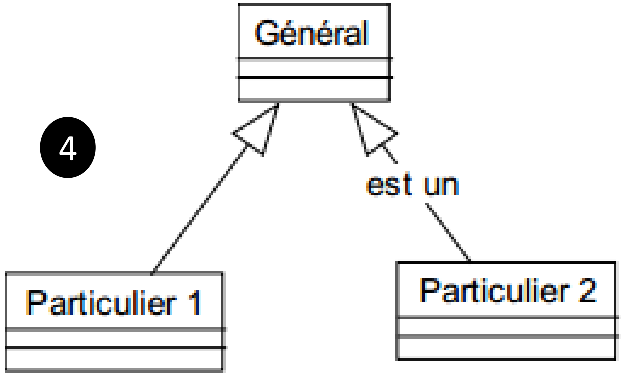
3



4



4



1990s – BORING DAYS

- **No major advancements** in database systems or application workloads.
  → Microsoft forks Sybase and creates SQL Server.
  → MySQL is written as a replacement for mSQL.
  → Postgres gets SQL support.

2000s – INTERNET BOOM

- All the **big players** were heavyweight and expensive. Open-source databases were missing important features.

- Many companies wrote their own custom middleware to scale out database across single node DBMS instances.

2000s – DATA WAREHOUSES

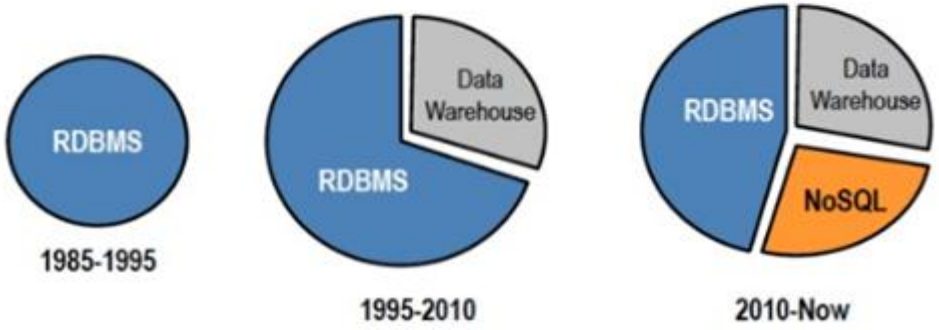
Rise of the special purpose OLAP DBMSs.
→ Distributed / Shared-Nothing
→ Relational / SQL
→ Usually closed-source.
Significant performance benefits from using Decomposition Storage Model (i.e., **columnar**)

2000s – NoSQL SYSTEMS

Eras of Database

Why NoSQL?

RDBMS one-size-fits-all needs



- Focus on high-availability & high-scalability:
  → Schemaless (i.e., "Schema Last")
  → Non-relational data models (document, key/value, etc)
  → No ACID transactions
  → Custom APIs instead of SQL
  → Usually open-source

ICDE 2005 conference

"One Size Fits All": An Idea Whose Time Has Come and Gone

Michael Stonebraker
Computer Science and Artificial Intelligence Laboratory, M.I.T., and StreamBase Systems, Inc.
stonebraker@csail.mit.edu

Uğur Çetintemel
Department of Computer Science Brown University, and StreamBase Systems, Inc.
ugur@cs.brown.edu

The last 25 years of commercial DBMS development can be summed up in a single phrase: "one size fits all". This phrase refers to the fact that **the traditional DBMS architecture (originally designed and optimized for business data processing) has been used to support many data-centric applications** with widely varying characteristics and requirements. In this paper, we argue that this concept is no longer applicable to the database market, and that the commercial world will fracture into a collection of independent database engines, some of which may be unified by a common front-end parser. We use examples from the stream-processing market and the data-warehouse market to bolster our claims. We also briefly discuss other markets for which the traditional architecture is a poor fit and argue for a critical rethinking of the current factoring of systems services into products.

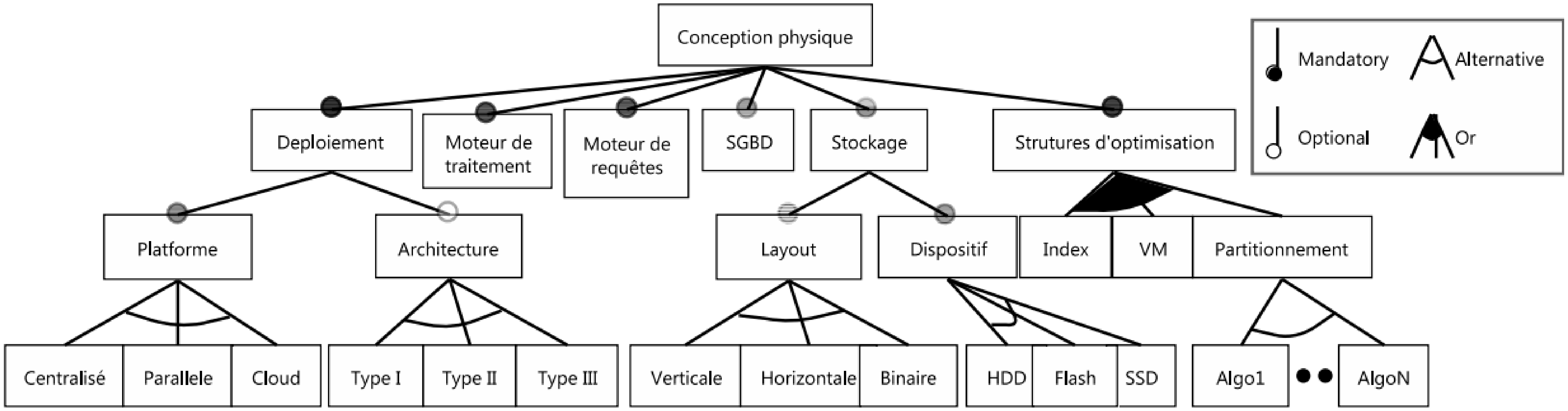
NoSQL practitioners focus on physical data model design

Physical Data model **represents** the data at data layer or internal layer (low level data structures, records, pointers, Index etc.)

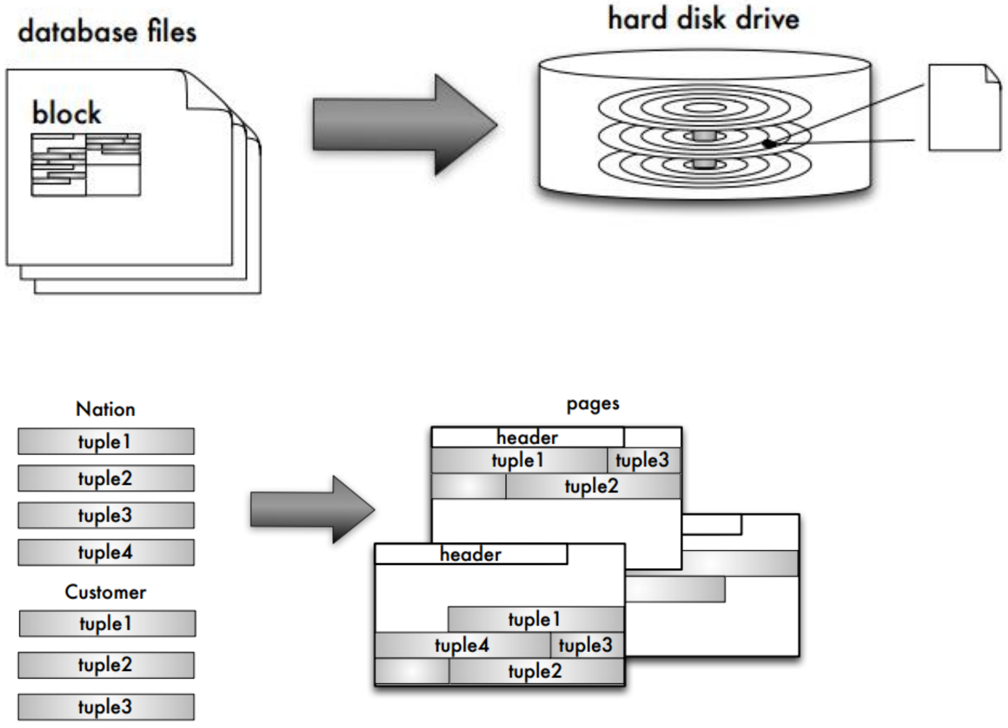
Physical Data model **describes:**

- How data are stored
- How data are scattered and ordered
- How data would be retrieved from memory
- spécifiés des structures internes de stockage, des index, des chemins d’accès, des paramètres physiques et l’organisation des fichiers de la BD.

Goal of physical design (l’efficacité du traitement des données?)

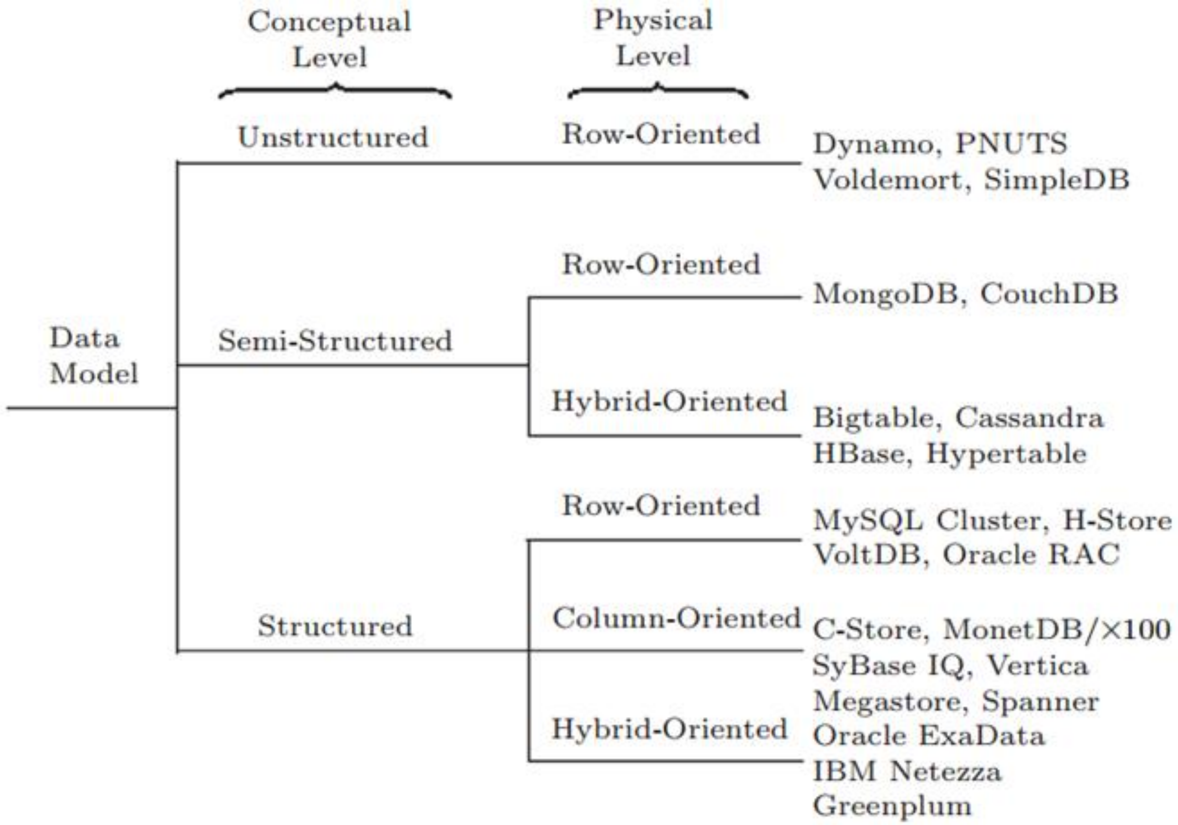


Taxonomy of physical data model



column-oriented database system

OID	C_NationKey	OID	C_MktSegment	OID	C_Phone	OID	C_Name	OID	...
1	24	1	Automobile	1	21314	1	Smith	1	...
2	23	2	Building	2	33421	2	Reilly	2	...
3	24	3	Automobile	3	09832	3	Miller	3	...
4	7	4	Furniture	4	32455	4	Schmidt	4	...



Landscape of representative data management systems

- **NoSQL databases** have been emerged as a revolutionary technology for modern web-scale and cloud-based applications.
- NoSQL practitioners focus on **physical data model design** rather than **the traditional conceptual / logical data model** process
  - A variety of NoSQL databases are industrialized which have different types of **physical level data models**
  - lack of common standardization among NoSQL databases
  - lack of commonly accepted conceptual model for NoSQL databases
    - Difficult to choose the right physical data model for specific application.
    - **A strong need for common conceptual model and logical level data model for those databases**

UMLtoGraphDB: Mapping Conceptual Schemas to Graph Databases

Gwendal Daniel[1], Gerson Sunyé[1], and Jordi Cabot[2]

[1] AtlanMod Team
Inria, Mines Nantes & Lina
{gwendal.daniel,gerson.sunye}@inria.fr
[2] ICREA - UOC
jordi.cabot@icrea.cat

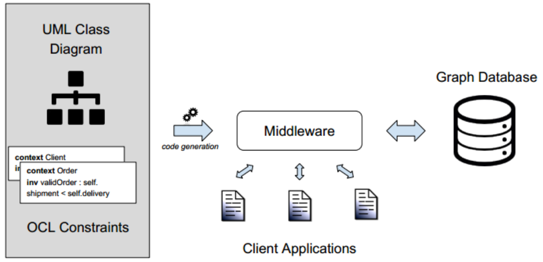
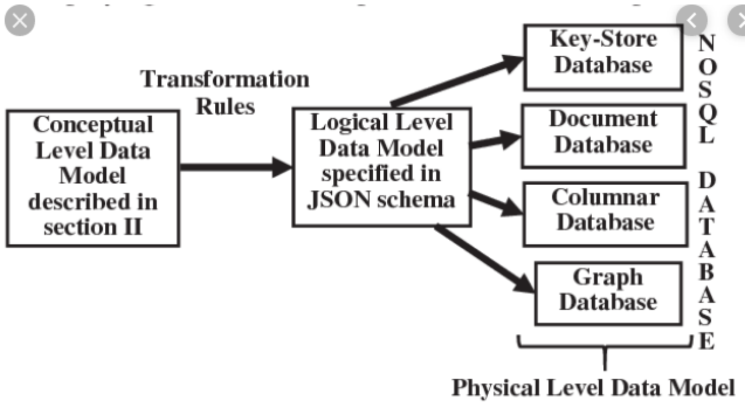


Fig. 1. Conceptual Model to Graph database

Modeling NoSQL Databases: From Conceptual to Logical Level Design



2010s – NewSQL

- Provide same performance for OLTP workloads as NoSQL DBMSs without giving up ACID:
  → Relational / SQL
  → Distributed
  → Usually closed-source

2010s – HYBRID SYSTEMS

- **H**ybrid **T**ransactional-**A**nalytical **P**rocessing.
  Execute fast OLTP like a NewSQL system while also executing complex OLAP queries like a data warehouse system.
  → Distributed / Shared-Nothing
  → Relational / SQL
  → All closed-source (as of 2016).

Questions