Database Systems I

Radim Bača

Department of Computer Science, FEECS

radim.baca@vsb.cz

dbedu.cs.vsb.cz

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

- Subject organization

- Why database systems?
- History of database systems
- Key koncepts
- Relational data model
- Starting SQL commands

Subject

- The page dbedu.cs.vsb.cz contains all needed materials in the bookmark 'Introduction to Database Systems'
- Use your school login/password for login.

- When needed, send me an e-mail to radim.baca@vsb.cz

Credit

- Two tests held directly on exercises
- The first test will be concerned with SELECT queries (40 points)
- The second test will be concerned with SQL DDL and DML (20 points)
- There will be three attempts for each test, however; *you need to solve at least one problem correctly* in order to try again!
- All materials can be used when writing each test
- Home project focused on information system design (30 points)

Software

- MS SQL Server 2016
- You will recieve emails with login details for `dbsys.cs.vsb.cz\Student` server
- Use `Sql Server Management Studio`, or `DataGrip`

Literature and Other Sources

- H.Garcia-Molina, J.D.Ullman, J.D.Widom. Database Systems: The Complete Book, Prentice Hall, ISBN 0-13-031995-3, 2002.
- R. Elmasri, S. Navathe. Fundamentals of Database Systems, Addison Wesley, ISBN 0-321-36957-2, 2010.
- J.D.Widom. `https://www.coursera.org/course/db` - e-learning course for a introduction to databases, Stanford University

Who is who

Who is who

Who is who

Friendster

Facebook

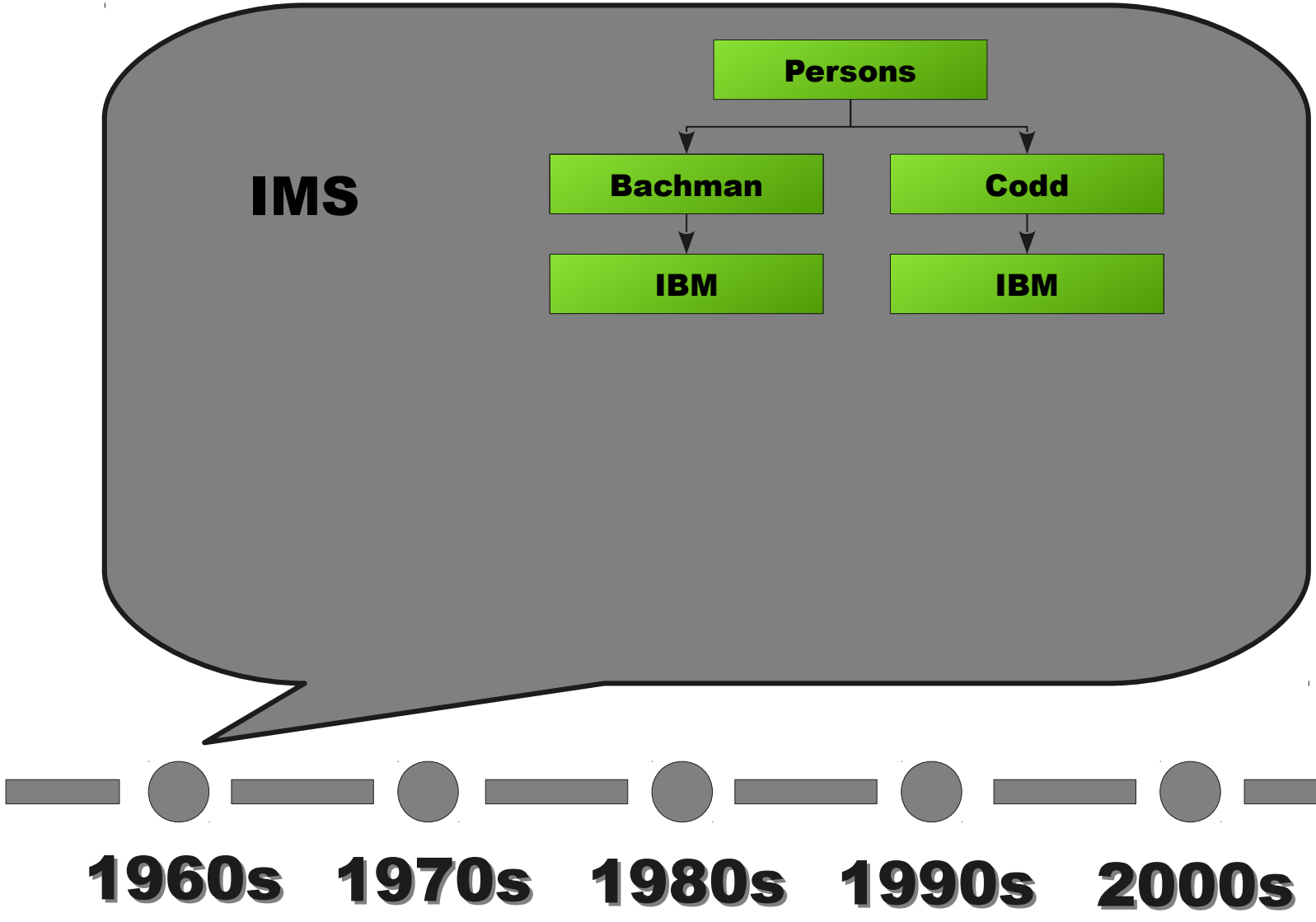
Why database systems?

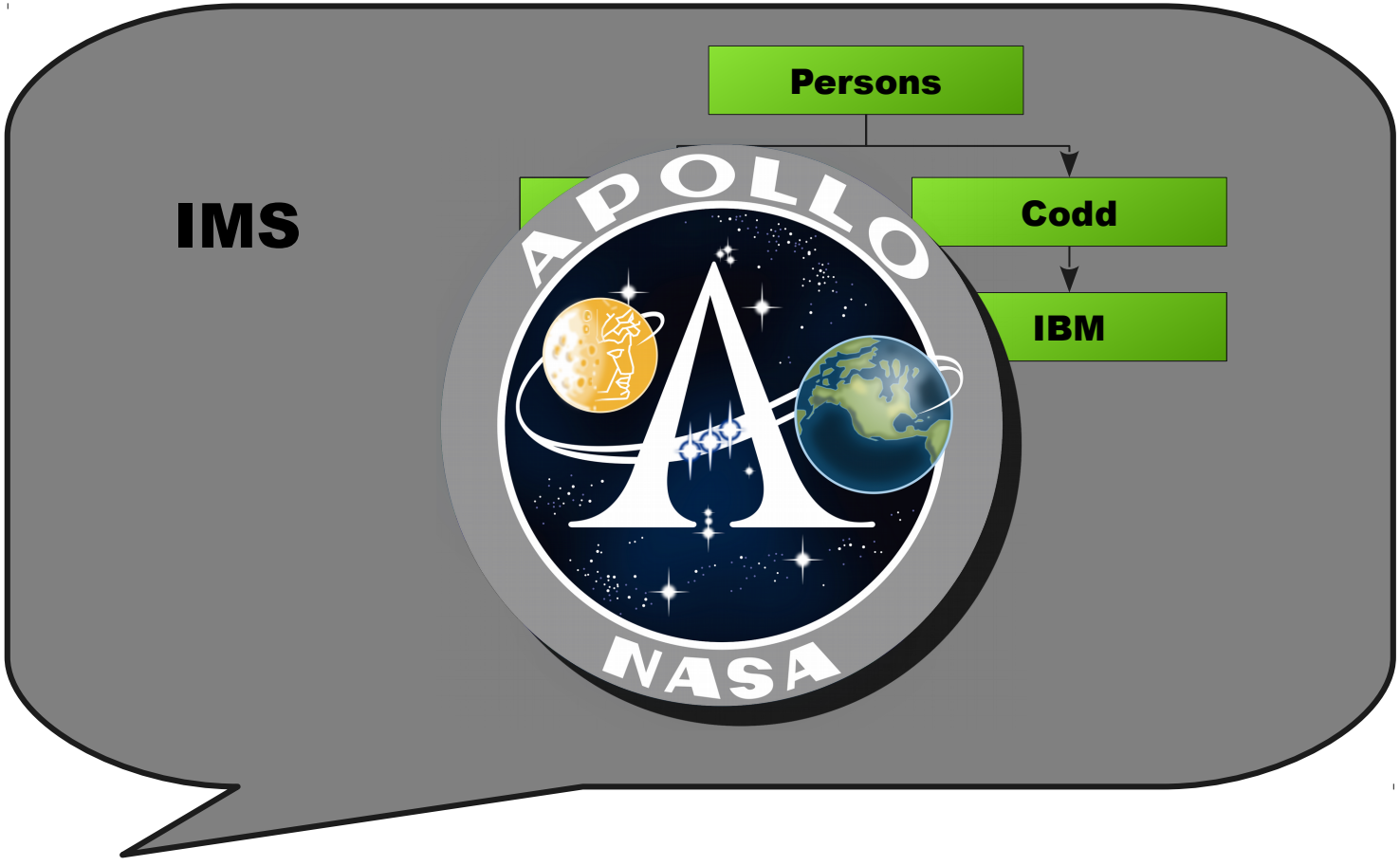
- They are literally everywhere
- Banking systems, games, on-line applications, desktop applications, communicators, social networks, search engines, etc.
- Database systems are:
  - massive
  - persistent
  - safe
  - multi-user
  - convenient
  - efficient
  - reliable

Quote

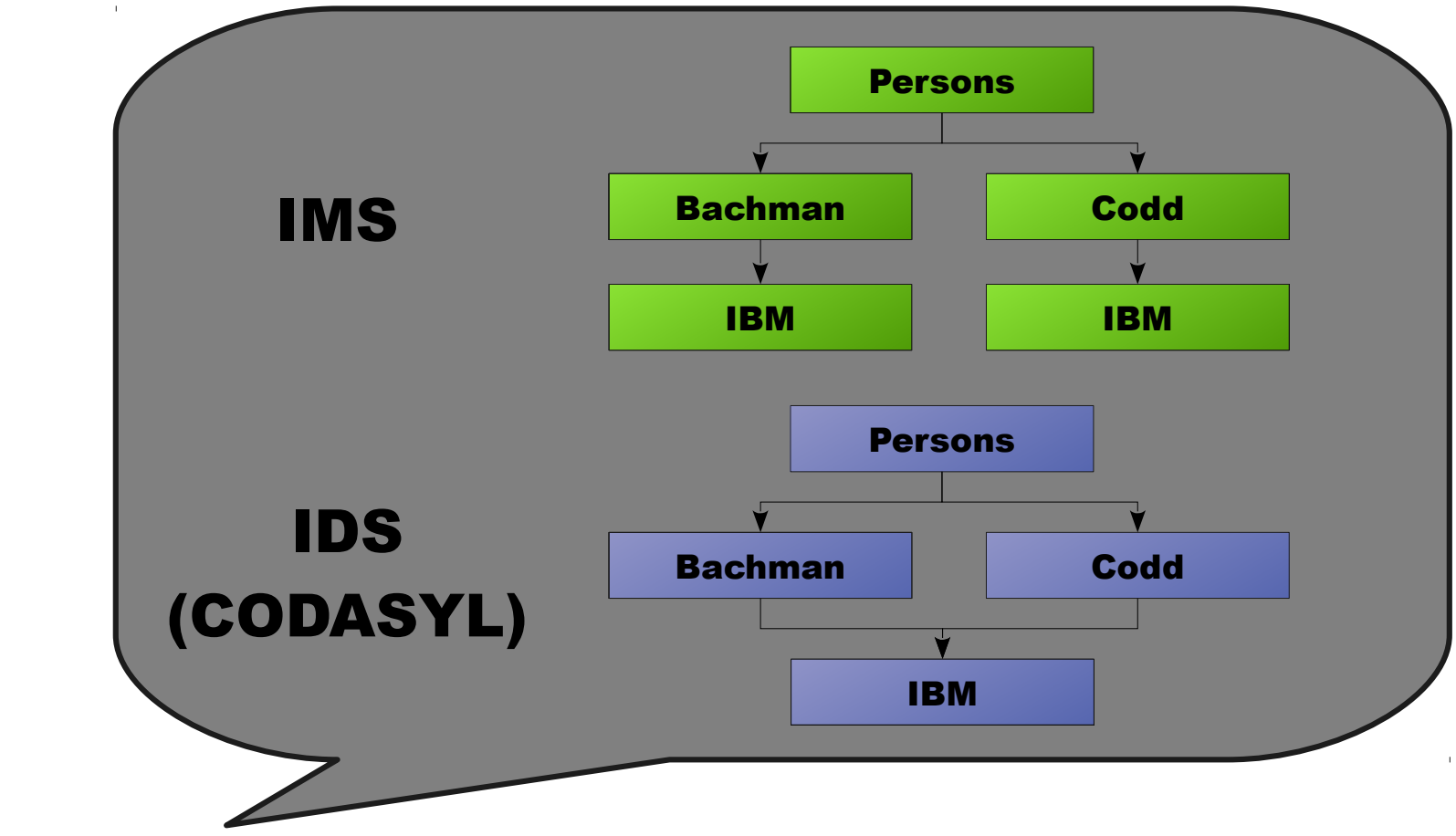
"Those who cannot learn from history are doomed to repeat it."
George Santayana

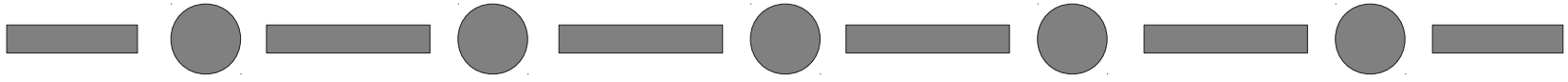
1960s 1970s 1980s 1990s 2000s


IMS
Persons
Bachman
Codd
IBM
IBM
1960s
1970s
1980s
1990s
2000s



1960s 1970s 1980s 1990s 2000s





1960s 1970s 1980s 1990s 2000s

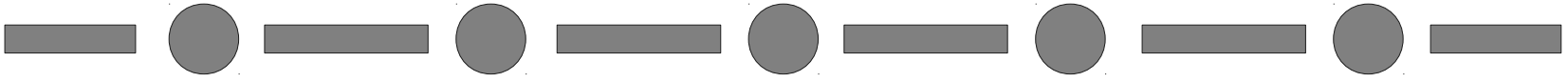
Relational data model

Affiliation

id	name
1	IBM
2	Berkeley
3	Oracle
4	Microsoft

Person

id	name	aff.
1	Bachman	1
2	Codd	1
3	Stonebreaker	2
4	Chamberlin	1
5	Ellison	3
6	Lomet	4



1960s 1970s 1980s 1990s 2000s

Relational data model

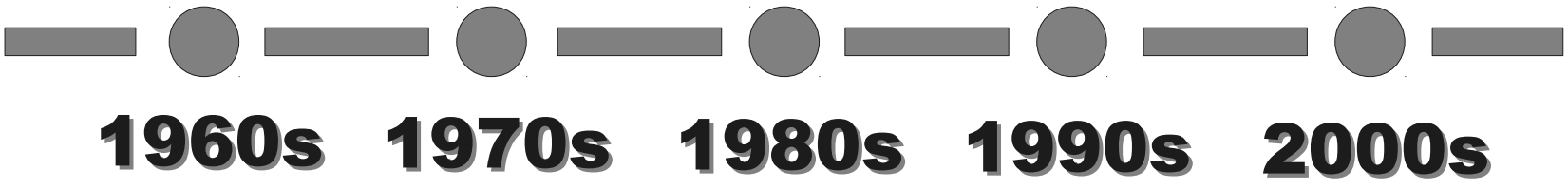
- Ingres
- System R

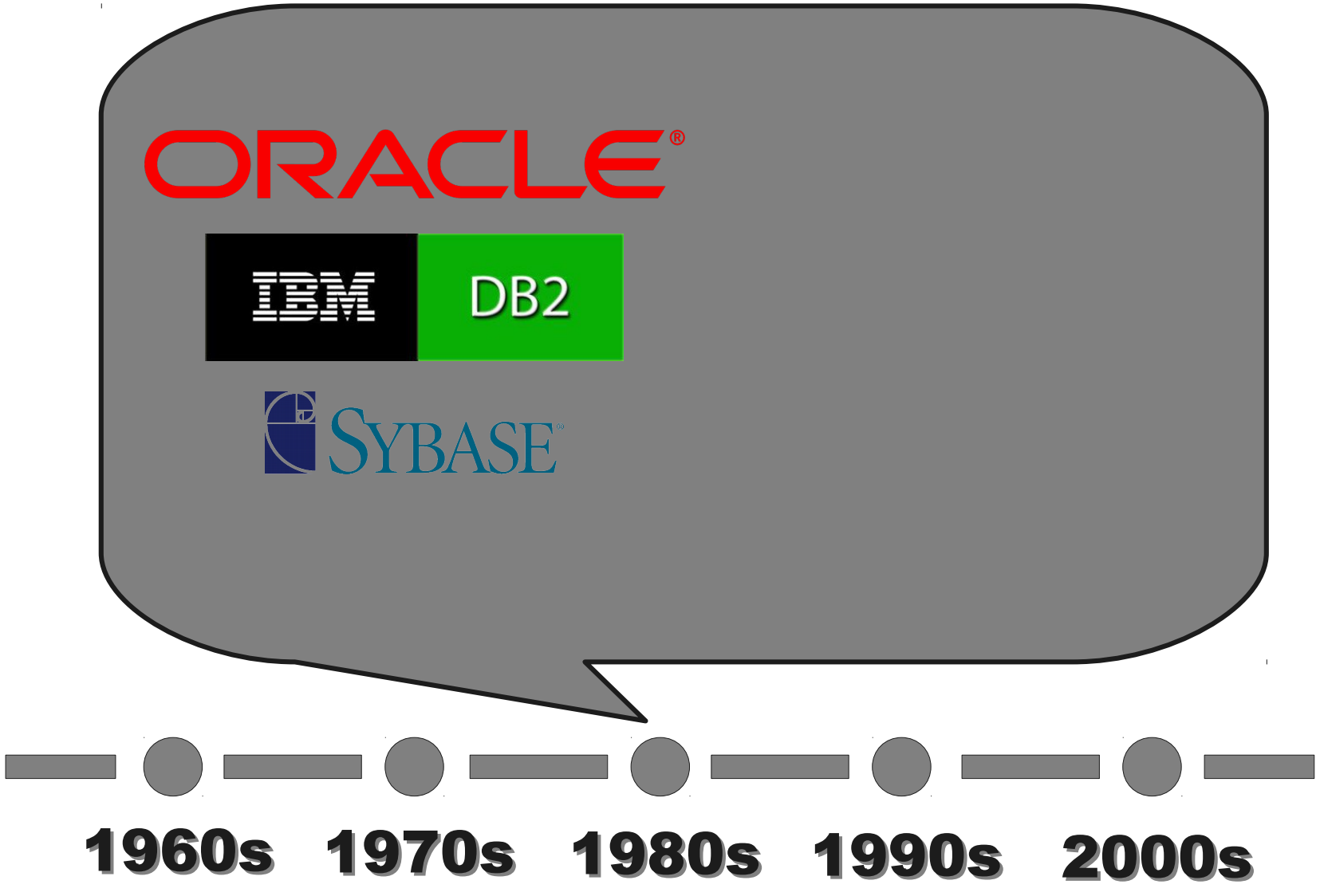
Person

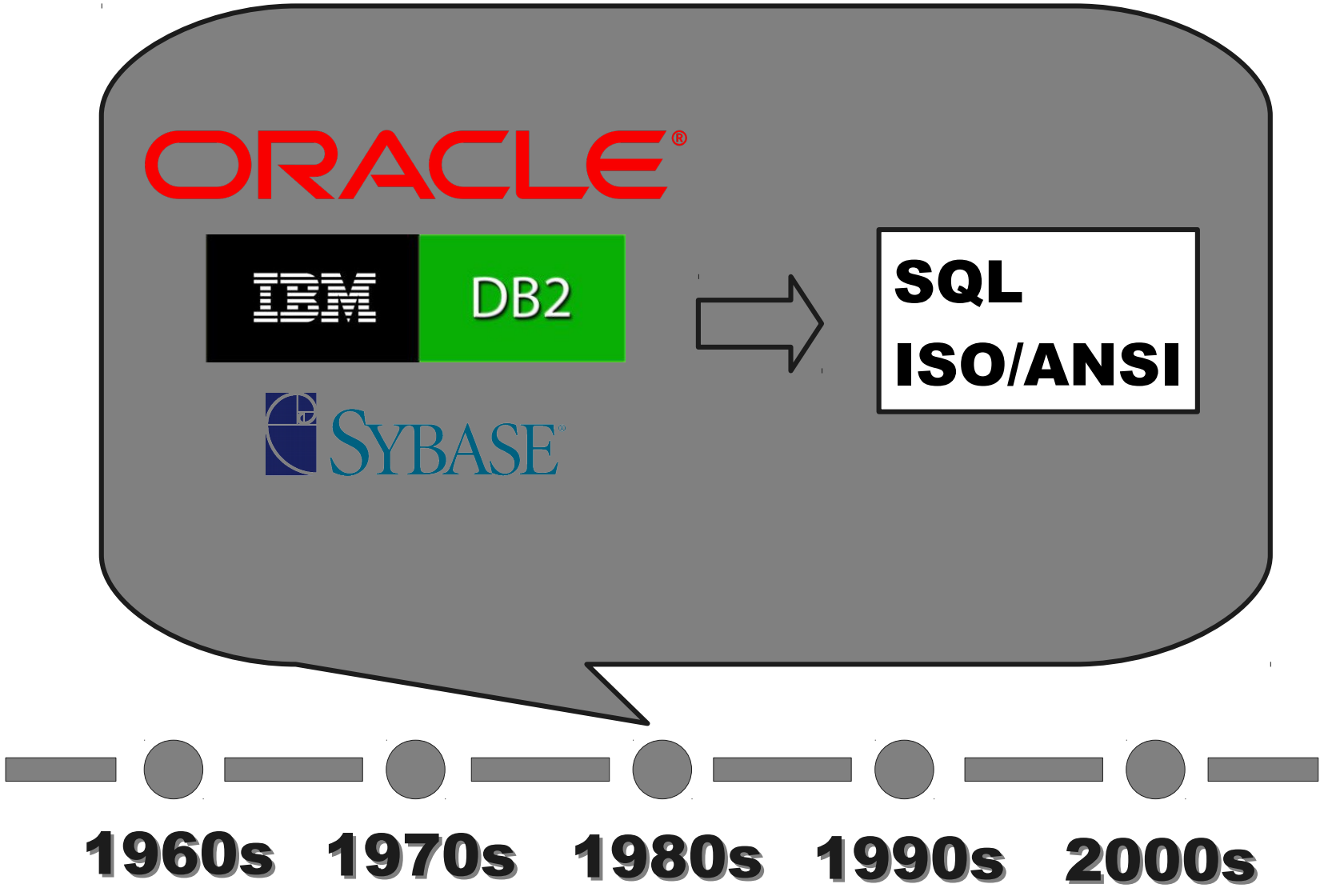
id	name	aff.
1	Bachman	1
2	Codd	1
3	Stonebreaker	2
4	Chamberlin	1
5	Ellison	3
6	Lomet	4

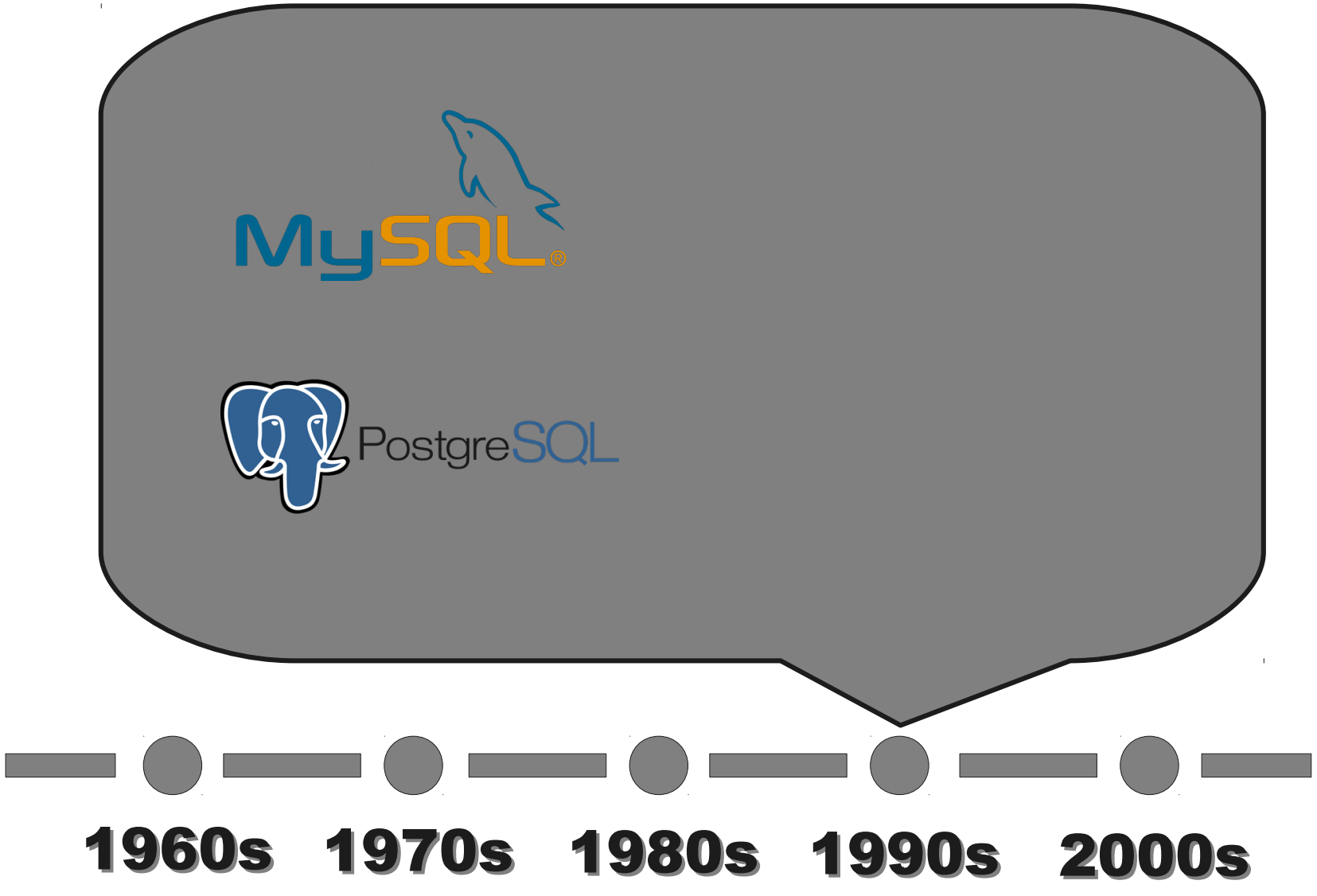
Affiliation

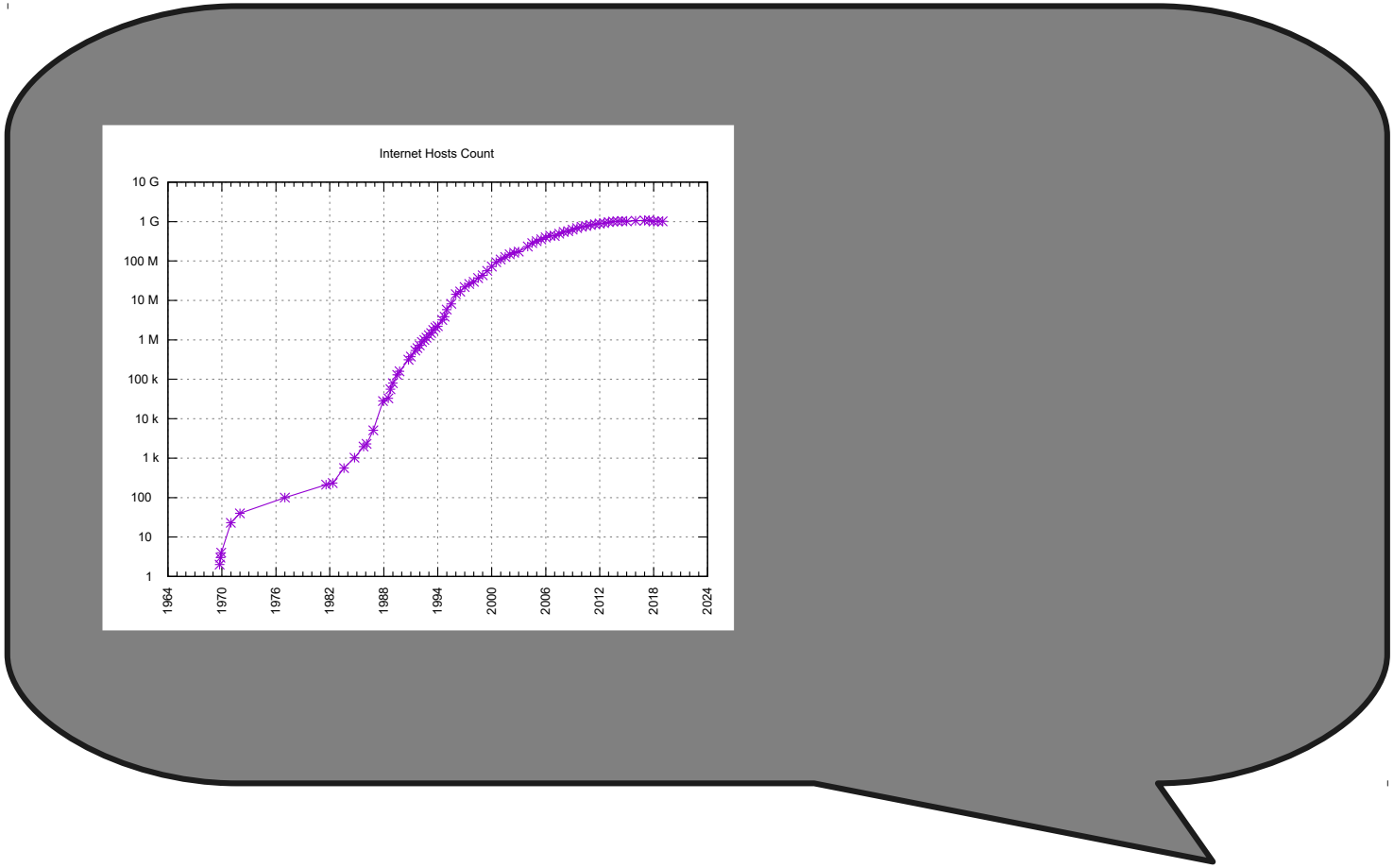
id	name
1	IBM
2	Berkeley
3	Oracle
4	Microsoft



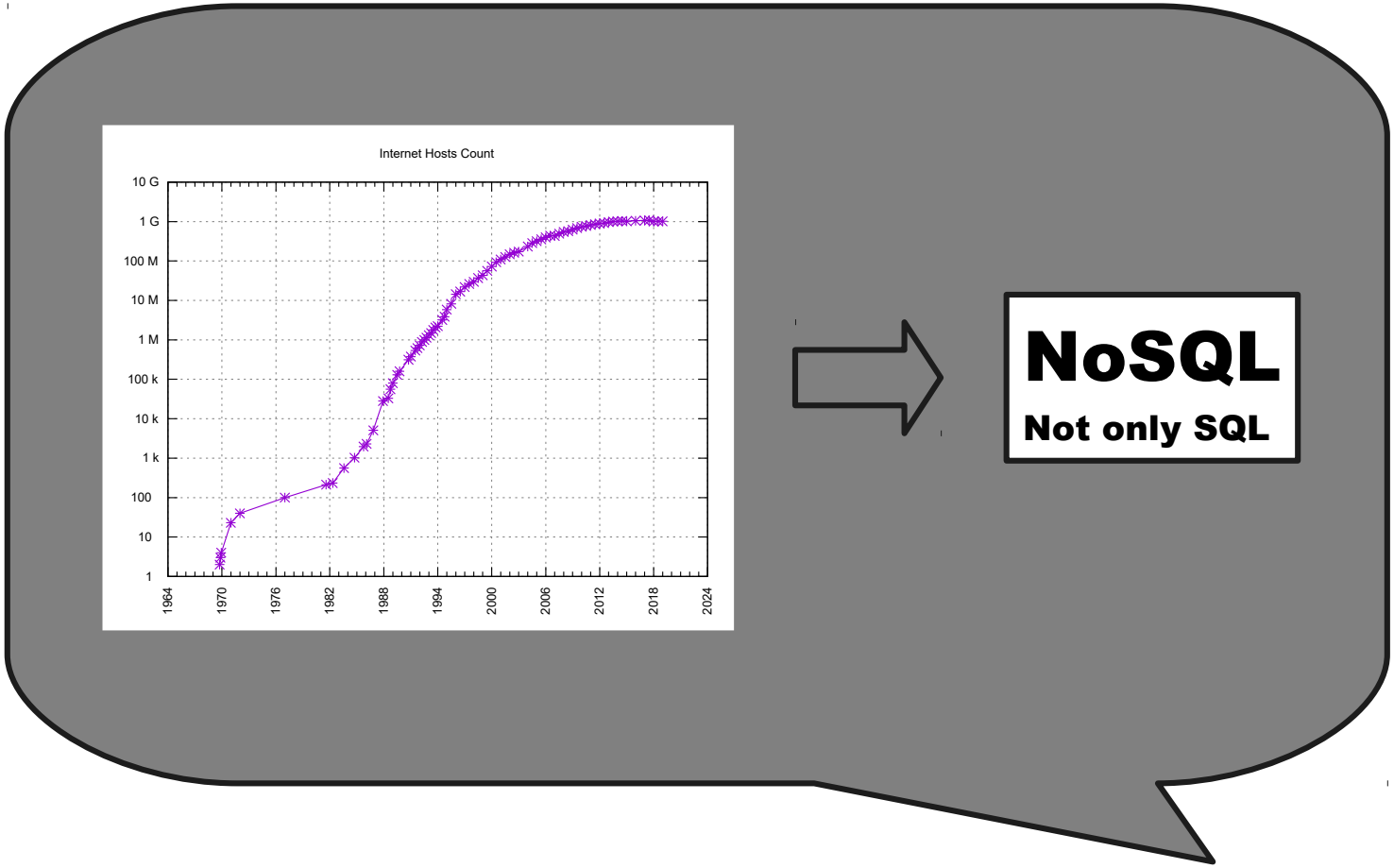

ORACLE®
IBM
DB2
SYBASE
1960s
1970s
1980s
1990s
2000s


ORACLE
IBM DB2
SYBASE
SQL
ISO/ANSI
1960s
1970s
1980s
1990s
2000s

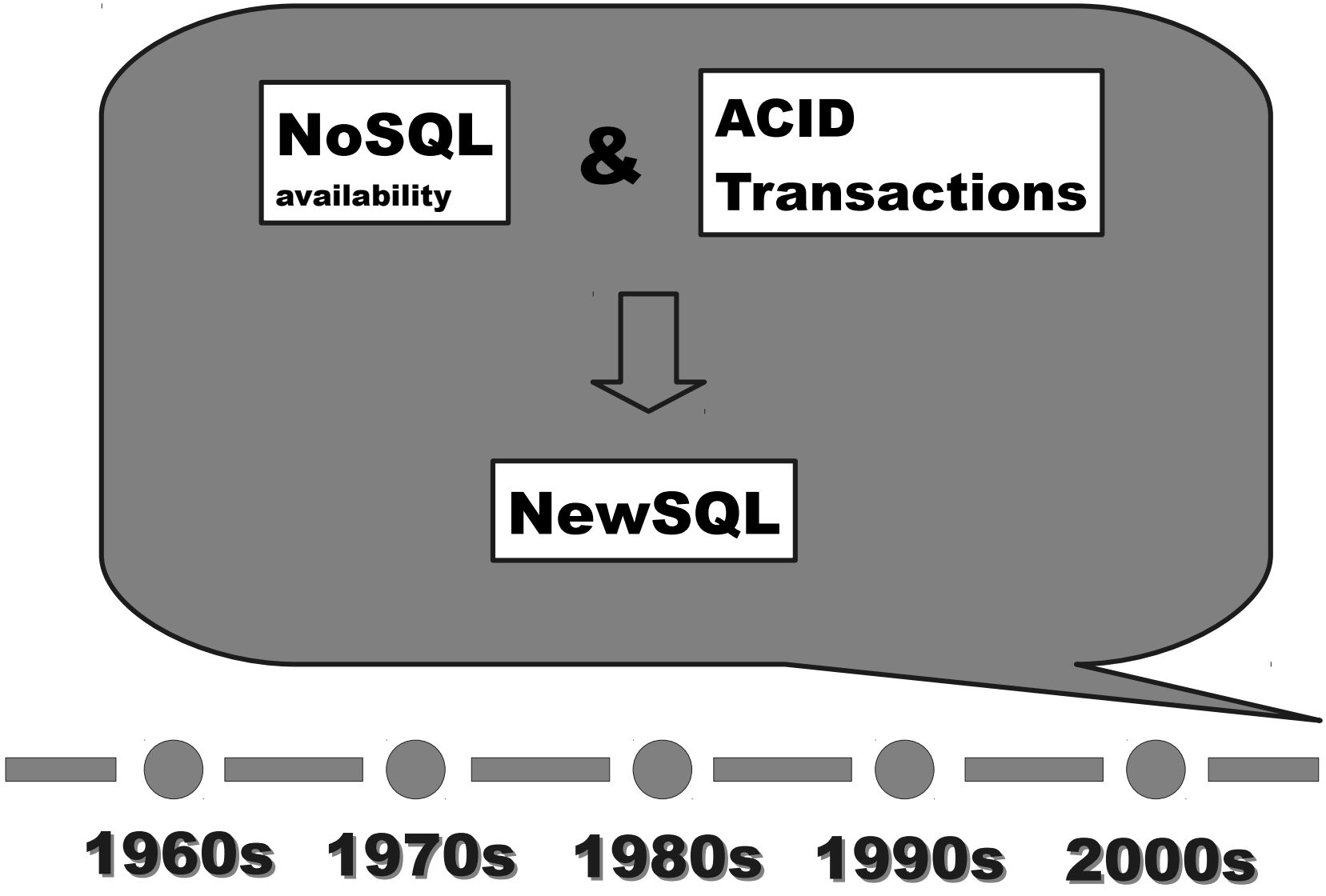

MySQL
PostgreSQL
1960s
1970s
1980s
1990s
2000s



1960s 1970s 1980s 1990s 2000s

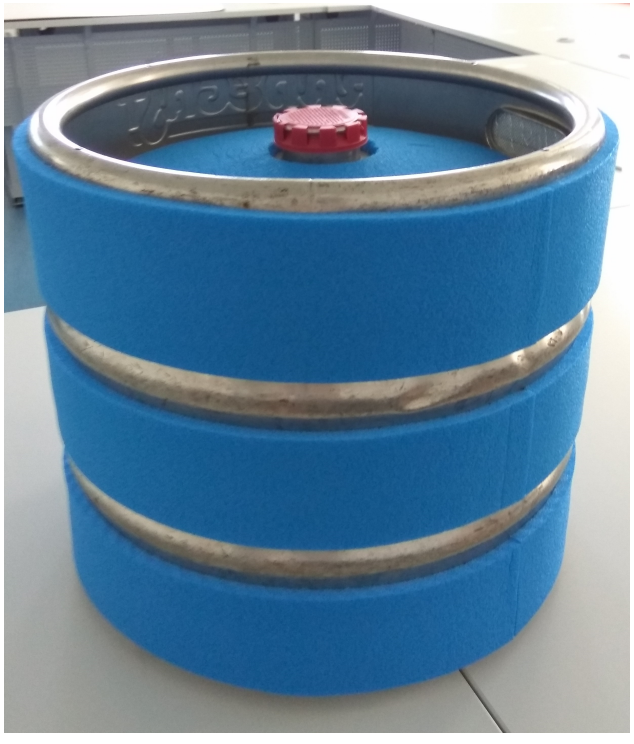


1960s 1970s 1980s 1990s 2000s


NoSQL
availability
&
ACID
Transactions
NewSQL
1960s
1970s
1980s
1990s
2000s

Database Systems in the World

New Types of Database Systems

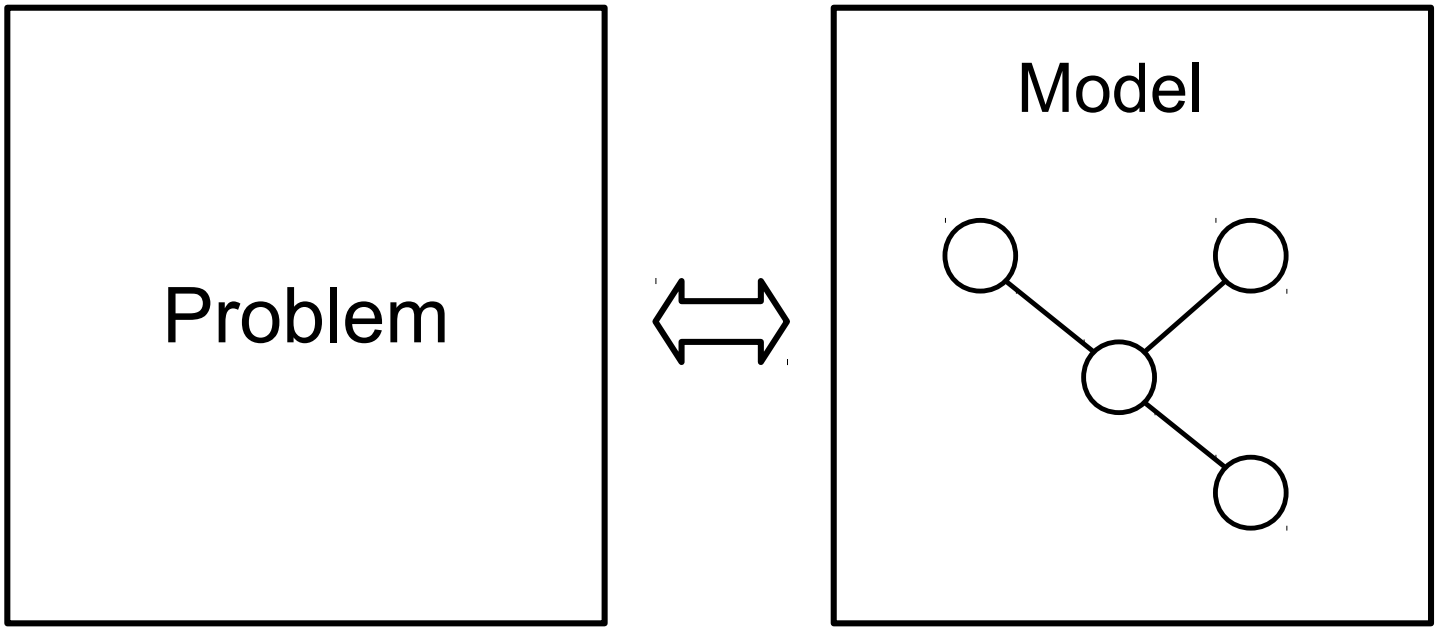


Database of database systems `https://dbdb.io`

Database vs. Database System

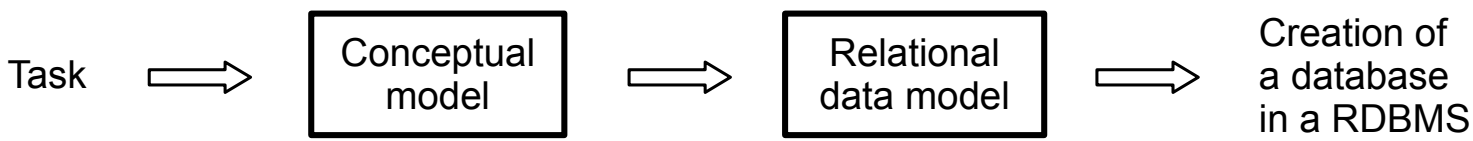
- **Database system** - it is an application that have an interface allowing to create and communicate with a database. It has previously specified attributes.
- **Database** - it is a set of mutually interconnected data that are stored in a database system.

Model



Model

- Typical models created during the analysis:
  - **Conceptual model/Logical model** - logical description of a database, which does not consider the database system
  - **Data model** (i.e., relational data model) - decription of a database with respect to a specific database system and data model
- Steps during the database creation can look like this:



Data model

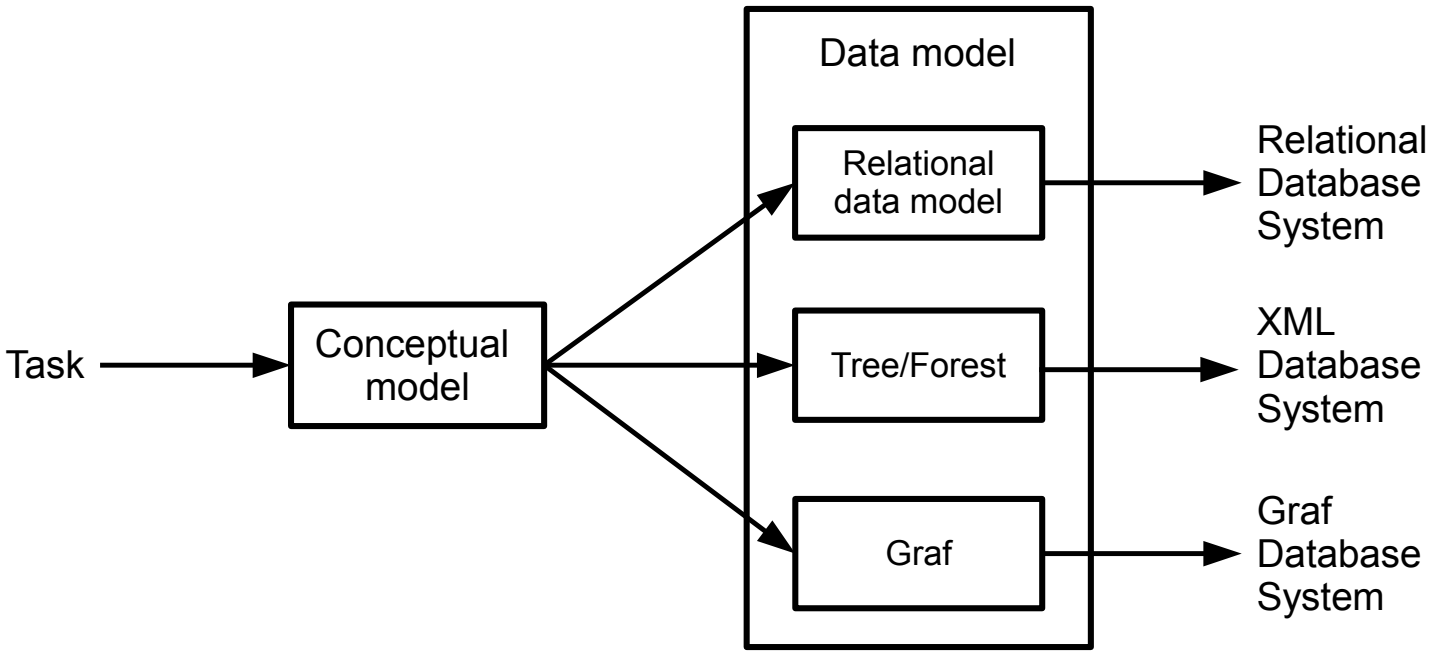
- Data can be organized (structured) in various ways:
  - Relational model (set of records)
  - Semistructured model (XML, JSON, BSON)
  - Graph (RDF)
  - Unstructured data

Data model

- Data can be organized (structured) in various ways:
  - Relational model (set of records)
  - Semistructured model (XML, JSON, BSON)
  - Graph (RDF)
  - Unstructured data

Model

- More generally steps during a database creation can look like this:



Scheme vs. data

- Scheme describes the database structure
- Data are concrete records corresponding to this structure
- It is similar to:
  - class vs. object
  - data type vs. variable
  - ...

Roles of people occupied with database systems

- Application programmer - programs applications above data
- Database administrator - decides which data (and how it) will be stored in database, modifies data structures, assigns access rights to data, reconstructs database when it is damaged, etc.
- Database designer - designs a database scheme
- Database application developer - often large teams

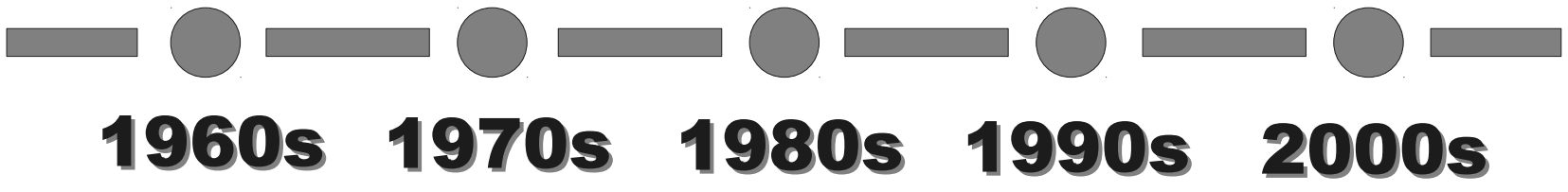
Relational data model

Person

id	name	aff.
1	Bachman	1
2	Codd	1
3	Stonebreaker	2
4	Chamberlin	1
5	Ellison	3
6	Lomet	4

Affiliation

id	name
1	IBM
2	Berkeley
3	Oracle
4	Microsoft



Parts of relational data model

- **Structure** - definition of relations and their content
- **Integrity** - definition of conditions that has to be satisfied
- **Operations** - how we can work with the data

Relation

- **Relation** = table

Relation

Employee

ID	name	passport no.	position
223	Newman	7905051111	programmer
124	Carter	6901112233	manager
154	Trier	7105029876	programmer

- **Relation**
- **Relation scheme** = describe a structure of a relation

Relation

Employee

ID	name	passport no.	position
223	Newman	7905051111	programmer
124	Carter	6901112233	manager
154	Trier	7105029876	programmer

- **Relation**
- **Relation scheme**
- **Tuple** = row of a table
- **Atribute and it domain** = column and his possible values

Database

- (Relational) database = set of relations
- Scheme of a (relational) database = set of relational schemes

Database

- (Relational) database = set of relations
- Scheme of a (relational) database = set of relational schemes

Attribute constraints (AC)

- AC are conditions further defining properties of attributes
- Examples:
  - *Can an attribute has a NULL value?*
  - *Does the value corresponds to a data type?*
- Consistent database = all relations meet all AC

Primary key

Employee

ID	name	passport no.	position
223	Newman	7905051111	programmer
124	Carter	6901112233	manager
154	Trier	7105029876	programmer

- **Primary key** - identifies uniquely row of a table
- It can be a single attribute

Primary key

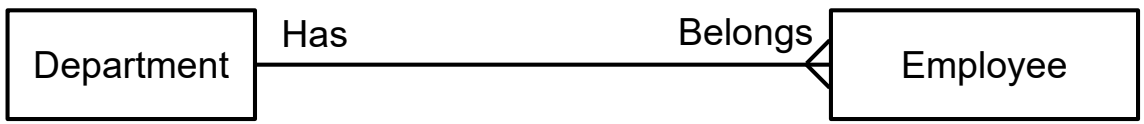
Department

branch	name	street	sales
22	export	St. James street	15
22	import	St. James street	16
26	export	Queen street	2

- **Primary key** - identifies uniquely row of a table
- It can be a single attribute, or it can be formed by more attributes

Associations

- Now we have two relations `Employee` and `Department`
- We want to represent a relationship where employee belongs to exactly one department
- In other words we want to model the following conceptual model:



Associations

Employee

ID	name	passport no.	position	branch of dpt.	name of dpt.
223	Newman	7905051111	programmer	22	export
124	Carter	6901112233	manager	22	export
154	Trier	7105029876	programmer	26	export

- **Foreign key** - represents an association between relations
- On the example we can see that foreign key corresponds to the primary key of the table to which it refers
- Therefore, in this case, it will be more convenient to create an artificial primary key ID for departments

Associations

Employee

ID	name	passport no.	position	dID
223	Newman	7905051111	programmer	1
124	Carter	6901112233	manager	1
154	Trier	7105029876	programmer	3

Department

dID	branch	name	street	sales
1	22	export	St. James street	15
2	22	import	St. James street	16
3	26	export	Queen street	2

Referential Integrity

- Referential Integrity - the foreign key value must exists in a attribute where the foreign key reference.
- Scalability problem
- NoSQL, NewSQL

Querying relational databases

- Queries consist of elementary operations above relations, whose result is again a relation
- Example of a query:

Find name of those employees working at departments having sales at least 10 millions

Querying relational databases

- Queries consist of elementary operations above relations, whose result is again a relation
- Example of a query:

Find name of those employees working at departments having sales at least 10 millions

- SQL:

```
SELECT e.name FROM Employee e, Department d WHERE e.dID = d.dID AND d.sales > 10
```

Querying relational databases

- Queries consist of elementary operations above relations, whose result is again a relation
- Example of a query:

Find name of those employees working at departments having sales at least 10 millions

- SQL:

```
SELECT e.name FROM Employee e, Department d WHERE e.dID = d.dID AND d.sales > 10
```

- Relational algebra:

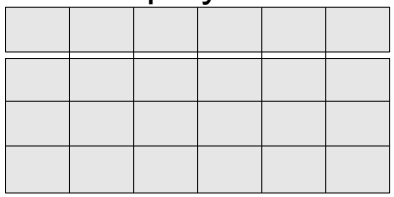
$$\pi_{\text{employee.name}}(\sigma_{\text{sales}>10}(\text{Department} \bowtie \text{Employee}))$$

Querying relational databases

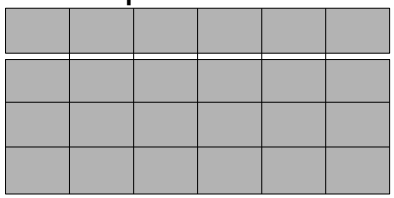
- Queries consist of elementary operations above relations, whose result is again a relation
- Example of a query:

Find name of those employees working at departments having sales at least 10 millions

Employee



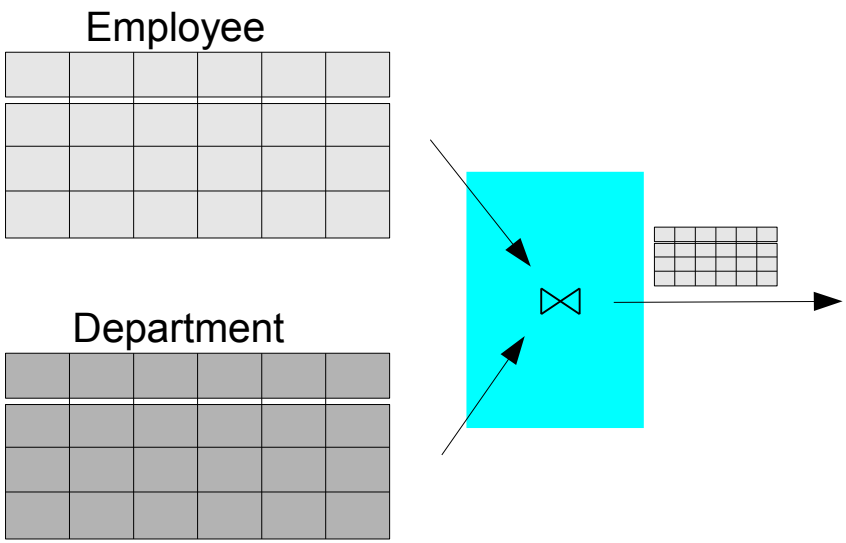
Department



Querying relational databases

- Queries consist of elementary operations above relations, whose result is again a relation
- Example of a query:

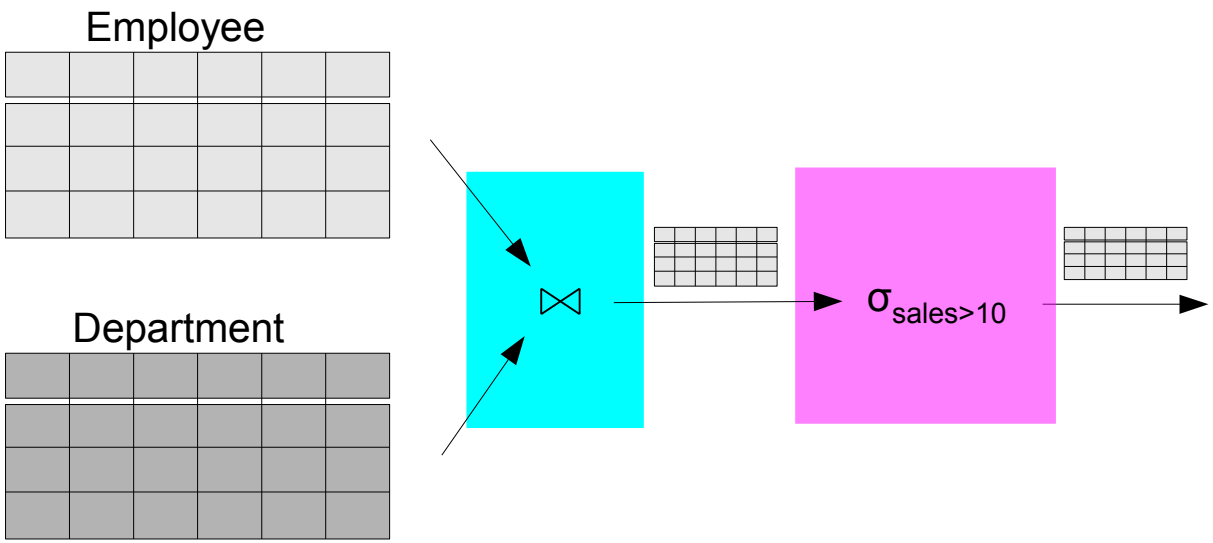
Find name of those employees working at departments having sales at least 10 millions



Querying relational databases

- Queries consist of elementary operations above relations, whose result is again a relation
- Example of a query:

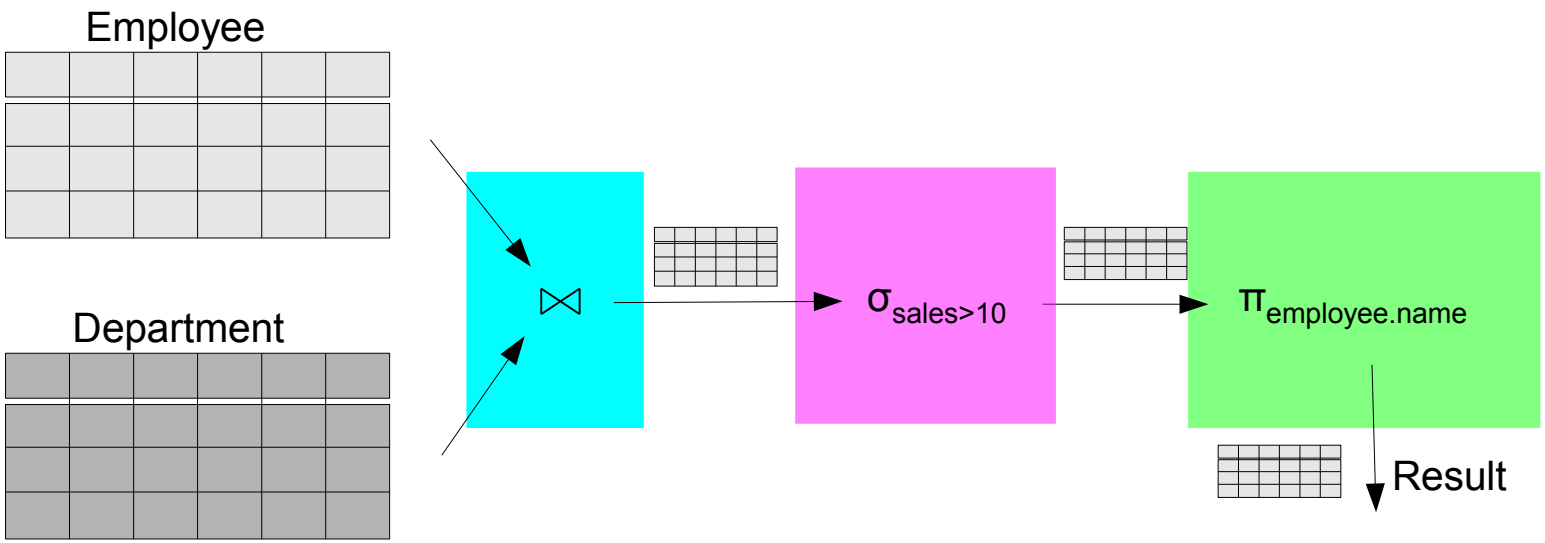
Find name of those employees working at departments having sales at least 10 millions



Querying relational databases

- Queries consist of elementary operations above relations, whose result is again a relation
- Example of a query:

Find name of those employees working at departments having sales at least 10 millions



References

- Jennifer Widom. Introduction to Databases. `https://www.coursera.org/course/db`
- Course home pages `http://dbedu.cs.vsb.cz`
- History of databases `https://www.youtube.com/watch?v=KG-mqHoXOXY`