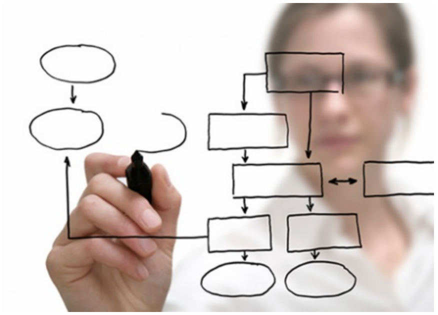
CORK INSTITUTE OF TECHNOLOGY

INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Computing@CIT



Distributed Data Management

Lecture 3: RDBMS and its Limitations

Outline

1. Traditional Relational Model: RDBMS.
2. Drawbacks and Limitations of RDBMS.

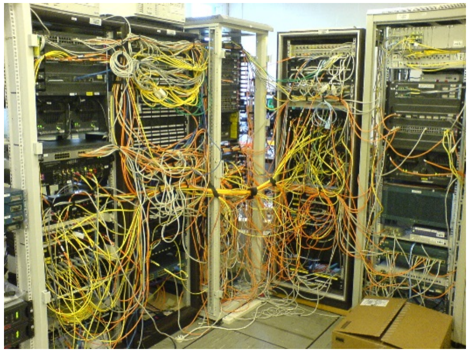
Outline

1. Traditional Relational Model: RDBMS.
2. Drawbacks and Limitations of RDBMS.

Traditional Relational Model: RDBMS

Data Storage. Context during the last decades…

1. Data in companies was mainly entered by their own workers (rather than from users/customers or sensors).
2. IT infrastructure was local, with few servers located in place (rather than leveraging the entire IT infrastructure to an external DC).
3. Relational Database Management Systems (RDBMS) was used as the predominant choice for storing, updating and retrieving information.



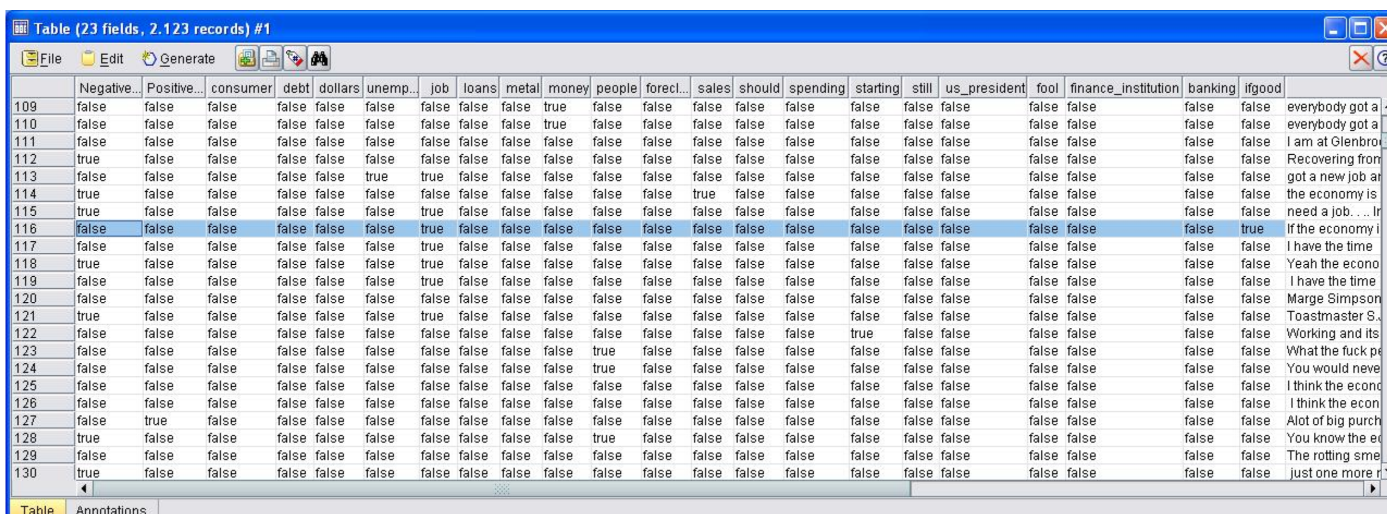
Traditional Relational Model: RDBMS

- RDBMS has been during this time the *de facto* standard, and due to its wide adoption and familiarity, very mature solutions and an abundance of tools have been developed:
  - MySQL, MS SQL Server, Oracle PostgreSQL.

Traditional Relational Model: RDBMS

- RDBMS are based in a <u>relational model</u>:
  - o Focus is set on *what the data stored consists of.*
  - o The model tries to build relations among the data.

- This relations are logically represented by 2 dimensional tables, where the rows represent the entries and the columns the attributes of each entry.



	Negative...	Positive...	consumer	debt	dollars	unemp...	job	loans	metal	money	people	forecl...	sales	should	spending	starting	still	us_president	fool	finance_institution	banking	ifgood	
109	false	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	everybody got a
110	false	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	everybody got a
111	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	I am at Glenbro
112	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	Recovering from
113	false	false	false	false	false	true	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	got a new job ar
114	true	false	false	false	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	the economy is
115	true	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	need a job. . .. Ir
116	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	true	If the economy i
117	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	I have the time
118	true	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	Yeah the econo
119	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	I have the time
120	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	Marge Simpson
121	true	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	Toastmaster S.J
122	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	Working and its
123	false	false	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	What the fuck pe
124	false	false	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	You would neve
125	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	I think the econo
126	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	I think the econ
127	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	Alot of big purch
128	true	false	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	You know the ec
129	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	The rotting sme
130	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	just one more r

Traditional Relational Model: RDBMS

- These relational tables lead to an <u>enforced schema</u>:
  - Each entry has the very same amount of attributes (horizontal homogeneity) and the datatype of each attribute has to be preserved (vertical homogeneity)
  - Moreover, integrity constraints can be posted to restrict the attribute domain.
- Another major feature of the relational model is <u>normalisation</u>, which aims to store the data in a higher number of smaller tables:
  - Each table → "Atomic" piece of information.
    Efficient link between tables → Full vision of the information.
  - Normalisation minimises the redundancy of information and eases the database maintenance.

Traditional Relational Model: RDBMS

- ❑ Normalisation example: CIT database of programs and students. Three “pieces” or tables:
  - o Table Programs → Contains each program and year.
  - o Table Students → Contains each student info.
  - o Table Registrations → Contains the registration of students per program.

- ❑ Maintaining these three tables helps to make the system consistent (<u>integrity of the database</u>).

Traditional Relational Model: RDBMS

- The relational model is supported by the Structured Query Language (SQL), which has been the standard for the last 30 years.
- SQL → Declarative language (focus on *declaring* the properties our query solution must hold, rather than the *imperative* way of enumerating the steps for obtaining our query solution).
- Based on an algebra calculus, SQL is very flexible and supports multiple operations over a RDBMS, all of them highly optimised for an efficient management of the database. This operations include:
  - Primitive operations: Insert, consult, update and delete.
  - Access to information among several tables: Joins.
  - Data aggregation: Count, sum (for purposes such as statistical analysis).

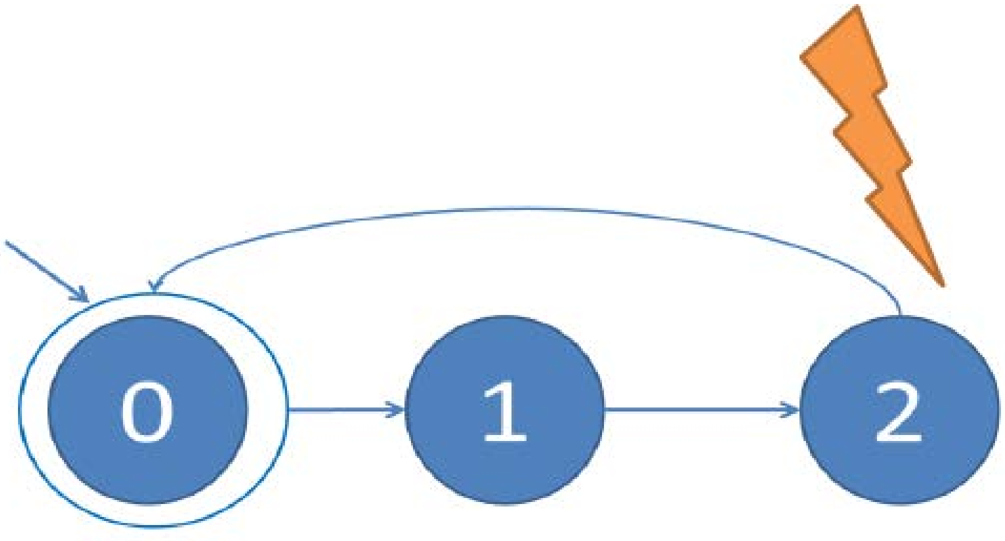
Traditional Relational Model: RDBMS

- The relational model is transactional, which implies:
  - Provide fault tolerance and predictable (deterministic) behaviour.

- It is said that RDBMS implement ACID transactions, as a result of enforcing them to hold the following 4 properties:
  - Atomic.
  - Consistent.
  - Isolated.
  - Durable.

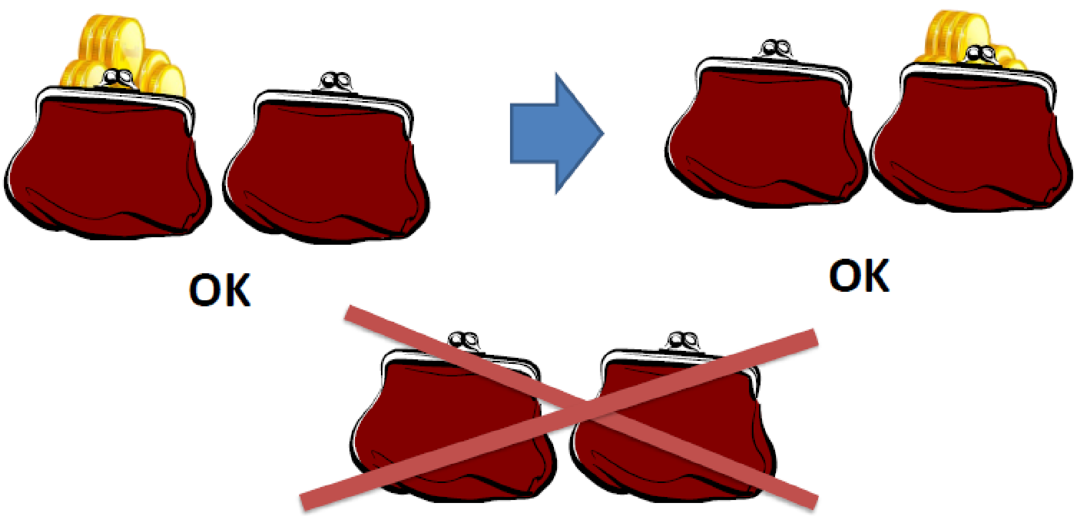
Traditional Relational Model: RDBMS

- Atomicity:
  - Provide a basic operation that is indivisible.
  - Gather several basic operations in one single step.
  - If all the basic operations succeed, then the step is performed. Otherwise, the subset of successful operations are rollback and the entire step is not done (as in a state machine).



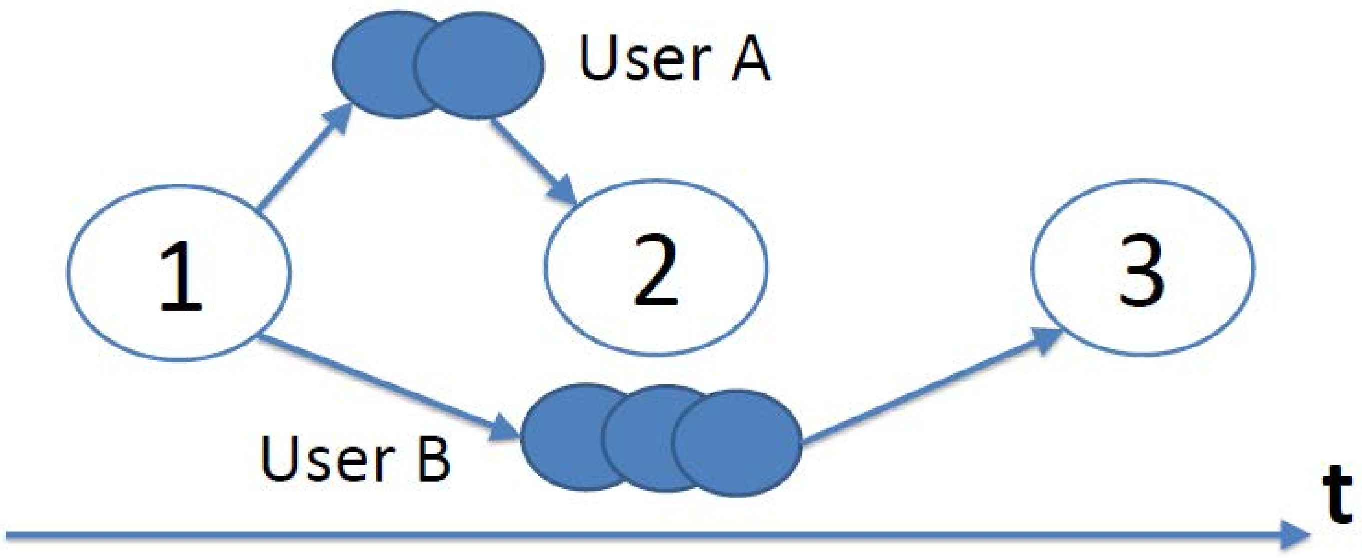
Traditional Relational Model: RDBMS

- Consistency:
  - Tied with the A property.
  - Each transaction at the end should leave the database in a consistent state (with the view of the state machine, every transaction should make the database to end up in a valid state).



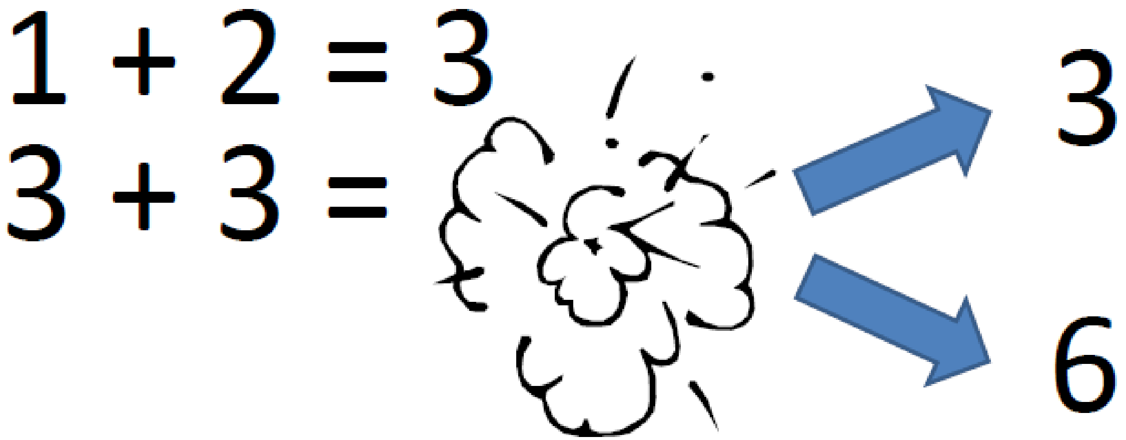
Traditional Relational Model: RDBMS

- Isolation:
  - Current transactions taking place are independent (so they do not influence themselves).
  - This allows multiple users to do things at the same time.



Traditional Relational Model: RDBMS

- Durability:
  - Steps are committed forever (making them persistent on disk, not in the database memory).
  - This makes that, if there is a system error at any time, the database can be brought back to a valid state.



Traditional Relational Model: RDBMS

In summary, the relational model of RDBMS was originally designed with the assumption that:

- The end user will interact directly with the database.
  - Thus, it makes really sense for a RDBMS to manage concurrency and integrity.
  - Access patterns are unkown (user can query anything). Thus, the query language is close to English, and the data structures have no bias to a particular pattern of query.
- Finally, the database will run on a single machine:
  - This is the only way to promise true ACID transactions.

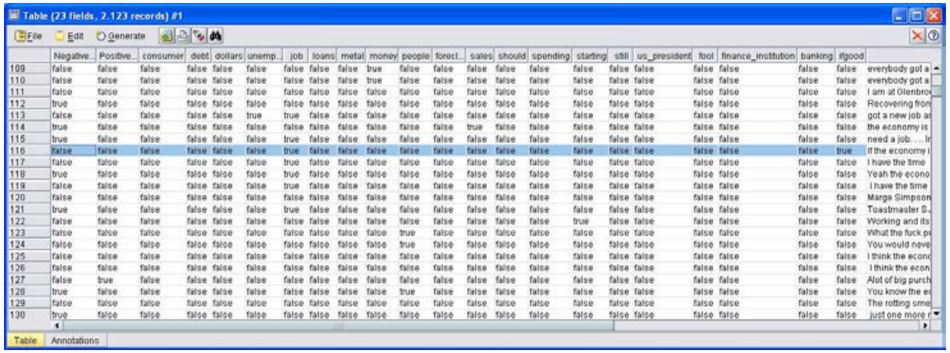
Outline

1. Traditional Relational Model: RDBMS.
2. Drawbacks and Limitations of RDBMS.

Drawbacks and Limitations of RDBMS

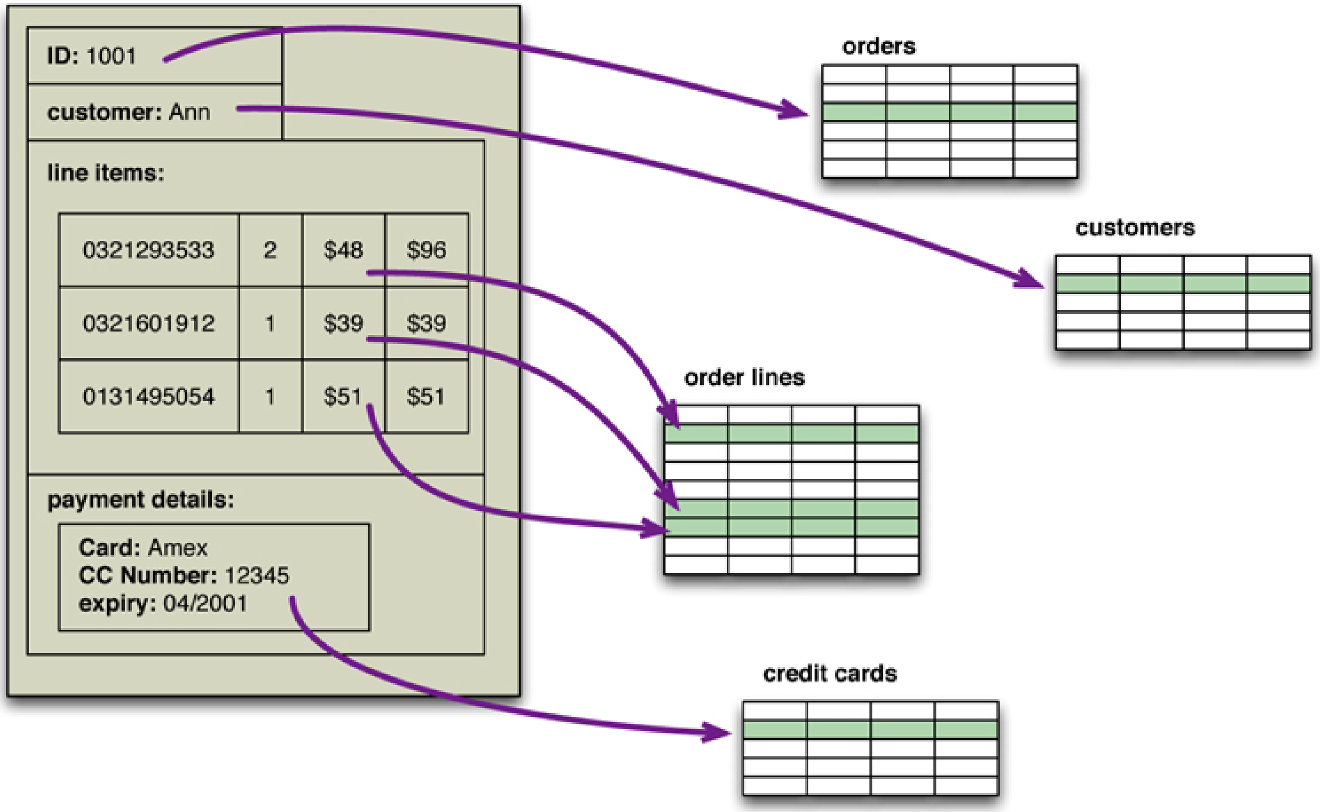
RDBMS worked well during many years:

- ❑ First source of problems → Object Oriented Programming.
  - o Applications become more complex.
  - o Impedance mismatch: How we process data, completely different of how to write it and store it in a database.
    - ▪ Dealing with the database becomes complex → Objects to be sliced and diced into tables.



Drawbacks and Limitations of RDBMS

- Impedance mismatch example – we read an invoice (single object), but it is split into multiple tables



Drawbacks and Limitations of RDBMS

- Impedance mismatch – we then need complex data mapping techniques in our application code, e.g. object-relational mapping (ORM) [i.e. transform objects in Java, Python, etc. to the relational representation in the data… and vice versa!]

Drawbacks and Limitations of RDBMS

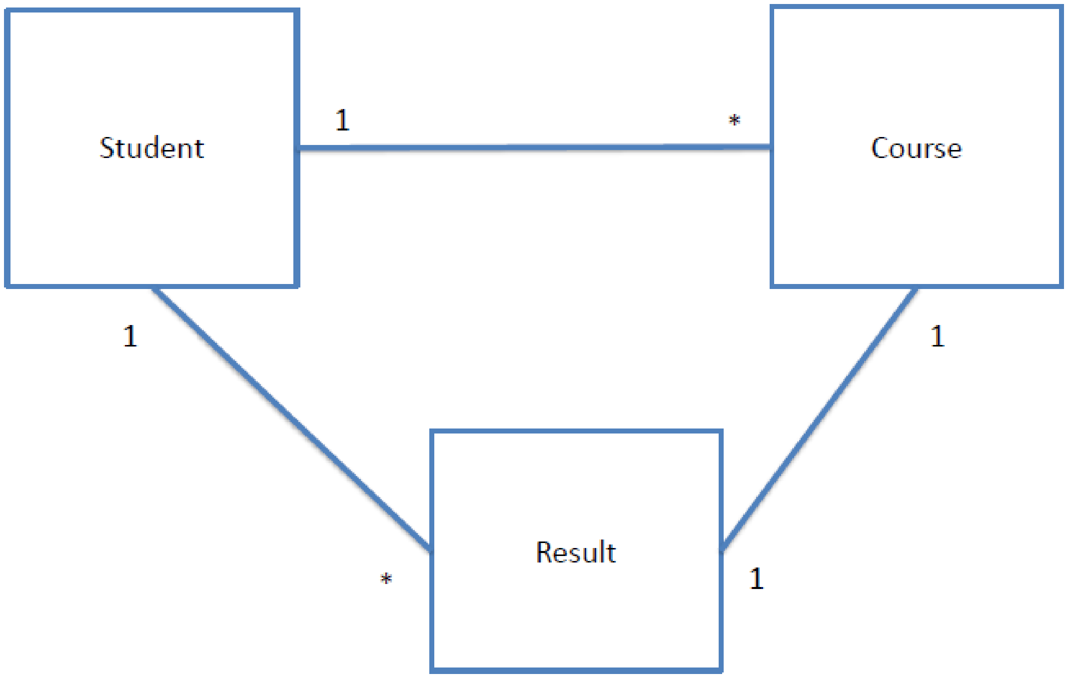
RDBMS worked well for many years (and still does for certain applications):

- ❑ Second and most important source of problem → Big Data.
- ❑ We will classify the problems into 3 categories:
  1. Rigid schema.
  2. Scale up.
  3. Scale out.

Drawbacks and Limitations of RDBMS

1. Rigid Schema.

- Let’s suppose we have to represent the CIT results of the year.
  - We have the following Entity Relationship Diagram (ERD):



Drawbacks and Limitations of RDBMS

1. Rigid Schema.

- We can represent it with 4 tables:
  - Student.
  - Course.
  - Result.
  - Link among them.

STUDENT

ID	NAME, ..

COURSE

ID	NAME, ..

STID	COID	REID

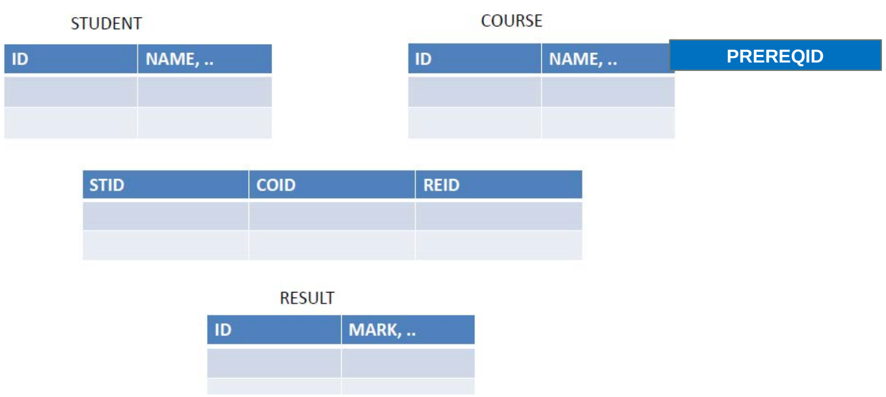
RESULT

ID	MARK, ..

Drawbacks and Limitations of RDBMS

1. Rigid Schema.

- Imagine that, with the database implemented and running a few years, we realise that *some* courses should have an extra field `pre-requisite-id`.
- For some courses, the new field will be null, for others it will be a recursive foreign key (i.e. linking back to another course).
- Then you realise later there might be **multiple** pre-requisites and a new link table is required
- Very disruptive with the rigid schema as table is dropped and recreated each time



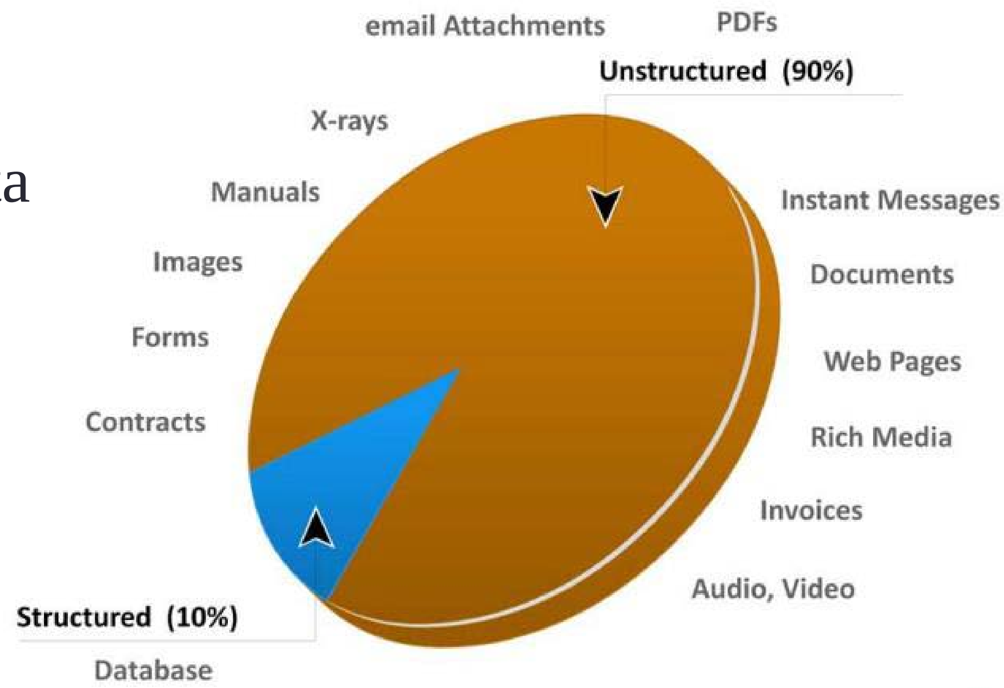
Drawbacks and Limitations of RDBMS

1. Rigid Schema.

- ❑ Moreover, big data is introducing vast amount of unstructured data in the scene.

Data is no longer just about text and numbers. The information is now linked, and consists of multiple data types.

- ❑ **How does this fit into the rigid schema of RDBMS?**



Drawbacks and Limitations of RDBMS

2. Scale up (or vertical scaling).

- ❑ Even if the data introduced is restricted to be structured data, the vast amount of data to be stored represents a problem itself for the RDBMS database.

Drawbacks and Limitations of RDBMS

2. Scale up.

- With our datasets growing massively, the performance of the database decreases exponentially.
  - The first approach was to buy bigger machines. But this had a penalisation in the IT infrastructure budget (a single highly performant and fault tolerant server can be more expensive than the cost of many less performant / fault tolerant servers added together.
  - The second approach was to invest more money/time/expertise on optimising the data and its representation in the database:
    - Secondary Indexes (these take up lots of space)
    - Denormalizing data (i.e. bigger, flatter tables).
    - Caching (requiring more memory or ultrafast disks)

Drawbacks and Limitations of RDBMS

2. Scale up.

- But even doing like this there is a limit for scaling up.
  - Even if you have all money, there is a ceiling where relying on a single powerful machine becomes unsustainable.

Drawbacks and Limitations of RDBMS

3. Scale out (or horizontal scaling).

RDBMS is not designed for scaling out (there was no concept of the massive amounts of today's data back in the 60's and 70's)

Reason for lack of suitability → **The CAP theorem**.

In a distributed system, it is impossible to provide all 3 guarantees:

- ❑ ***Consistency***: *All nodes see the same data at the same time.*
- ❑ ***Availability***: *Every request receives a response about whether it was successful or failed.*
- ❑ ***Partition tolerance***: *The system continues to operate despite arbitrary message loss or failure of part of the system.*

https://www.youtube.com/watch?v=Jw1iFr4v58M

Drawbacks and Limitations of RDBMS

3. Scale out.

<u>Conclusion:</u>

The ACID transactional model of RDBMS does not work well in a distributed system!

- ❑ It cannot be consistent if there is to be partition tolerance → AP.
- ❑ If we force nodes to wait until the synchronization is done, it cannot be always available → CP.
- ❑ In any case, join operations cannot be done if the data is distributed among nodes.

Outline

1. Traditional Relational Model: RDBMS.
2. Drawbacks and Limitations of RDBMS.

Thank you for your attention!