NOVA IMS Information Management School

Storing and Retrieving Data
Lecture 1
Introduction and Normalization

Lecturer: Mijail Naranjo-Zolotov
Email: mijail.naranjo@novaims.unl.pt

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Overview

Lecturer (Theoretical):
Mijail Naranjo-Zolotov
Email: mijail.naranjo@novaims.unl.pt

Lecturer (Labs):
Yuri Binev
Email: ybinev@novaims.unl.pt

Lecturer (Labs):
Américo Rio
Email: americo.rio@novaims.unl.pt

Overview – Learning units

1. Introduction and Normal forms in relational database.
2. Architecture of a DBMS.
3. Intoduction to SQL. CRUD operations.
4. SQL queries (aggregation, sorting)
5. SQL Joins. SQL views. SQL Triggers
6. Advanced topics: MySQL query optimization.
7. SQL vs NoSQL databases. CAP theorem. Wrap-up;

Overview – Grading policy

Regular examination period (1st epoch)

- Group project (35%).
- Final exam (65%).

Resit/improvement examination period (2nd epoch)

- Group project (35%)
- Final exam (65%).

Overview – Grading policy

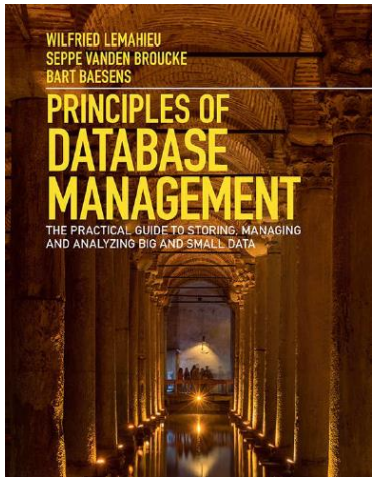
Rules:

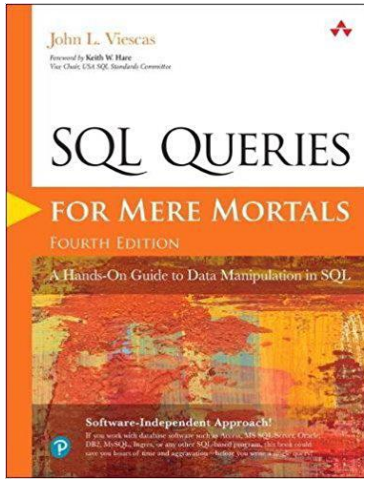
- The score in the exam should at least 9.5 (out of 20).
- The teams are made of 5 students.
- Late deliveries for the project will be penalized with 1 point for each late day up to 5 points.

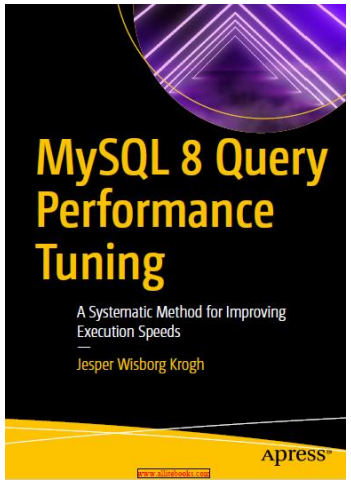
You can choose your team in moodle

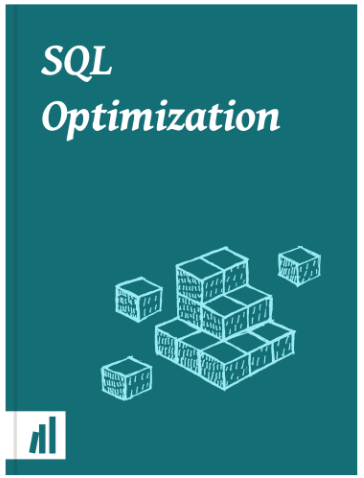
Image sources: https://medium.com/magenta-lifestyle/why-two-large-pizza-team-is-the-best-team-ever-4f19b0f5f719

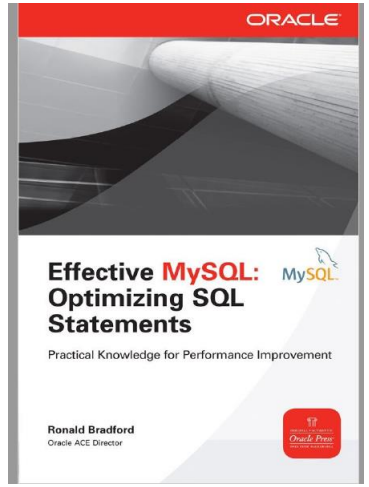
Overview - Bibliography











https://dataschool.com/sql-optimization/

Lecture 1: Introduction to Databases

What is database?

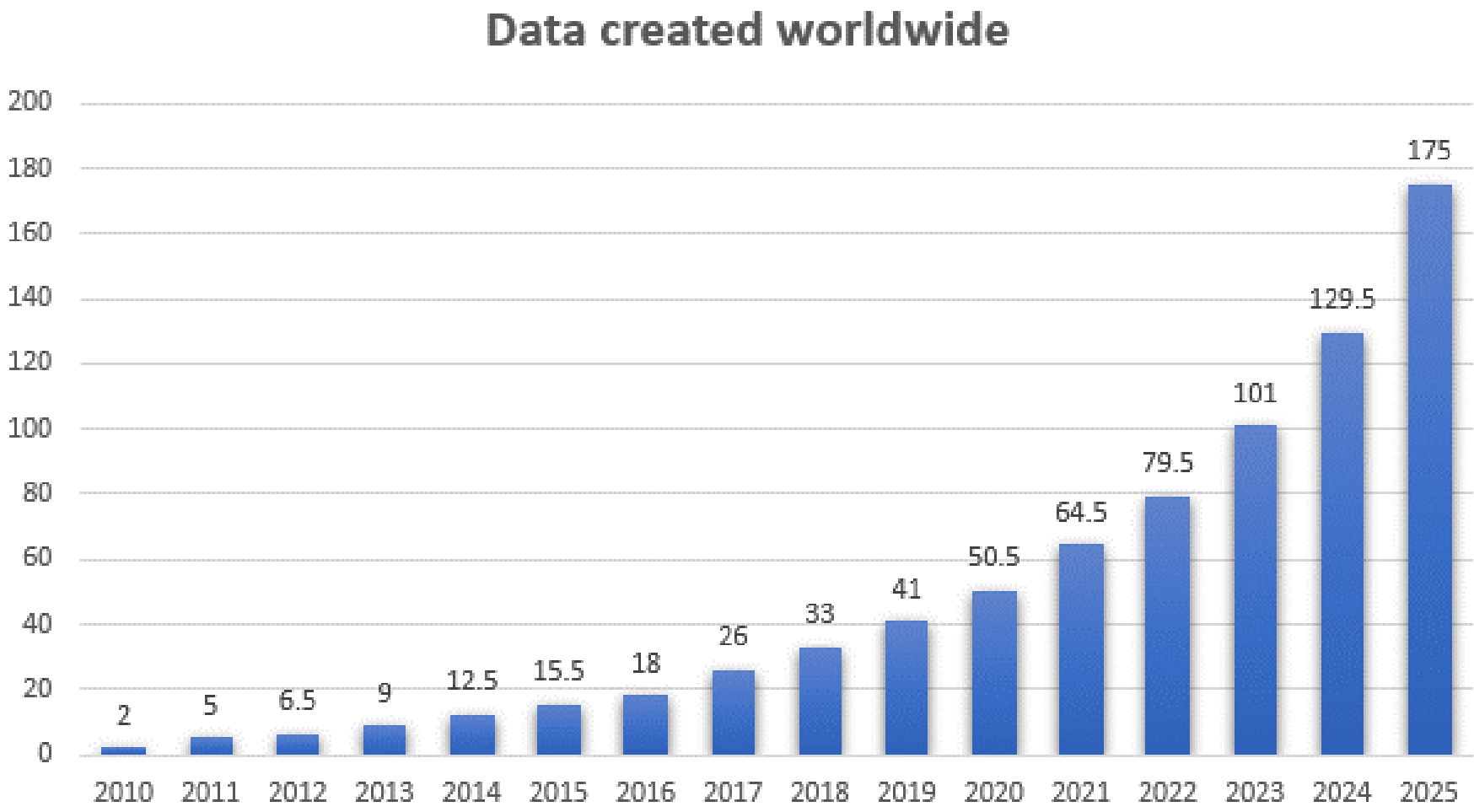
What is database?

- A **database** can be defined as a collection of related data items within a specific business process or problem setting.
- A **database management system (DBMS)** is the software package used to define, create, use, and maintain a database.
- The combination of a DBMS and a database is then often called a **database system**.

Source: Lemahieu, W., vanden Broucke, S., & Baesens, B. (2018). Principles of Database Management

Data created

(in zettabytes)



Source: https://www.statista.com/statistics/871513/worldwide-data-created/

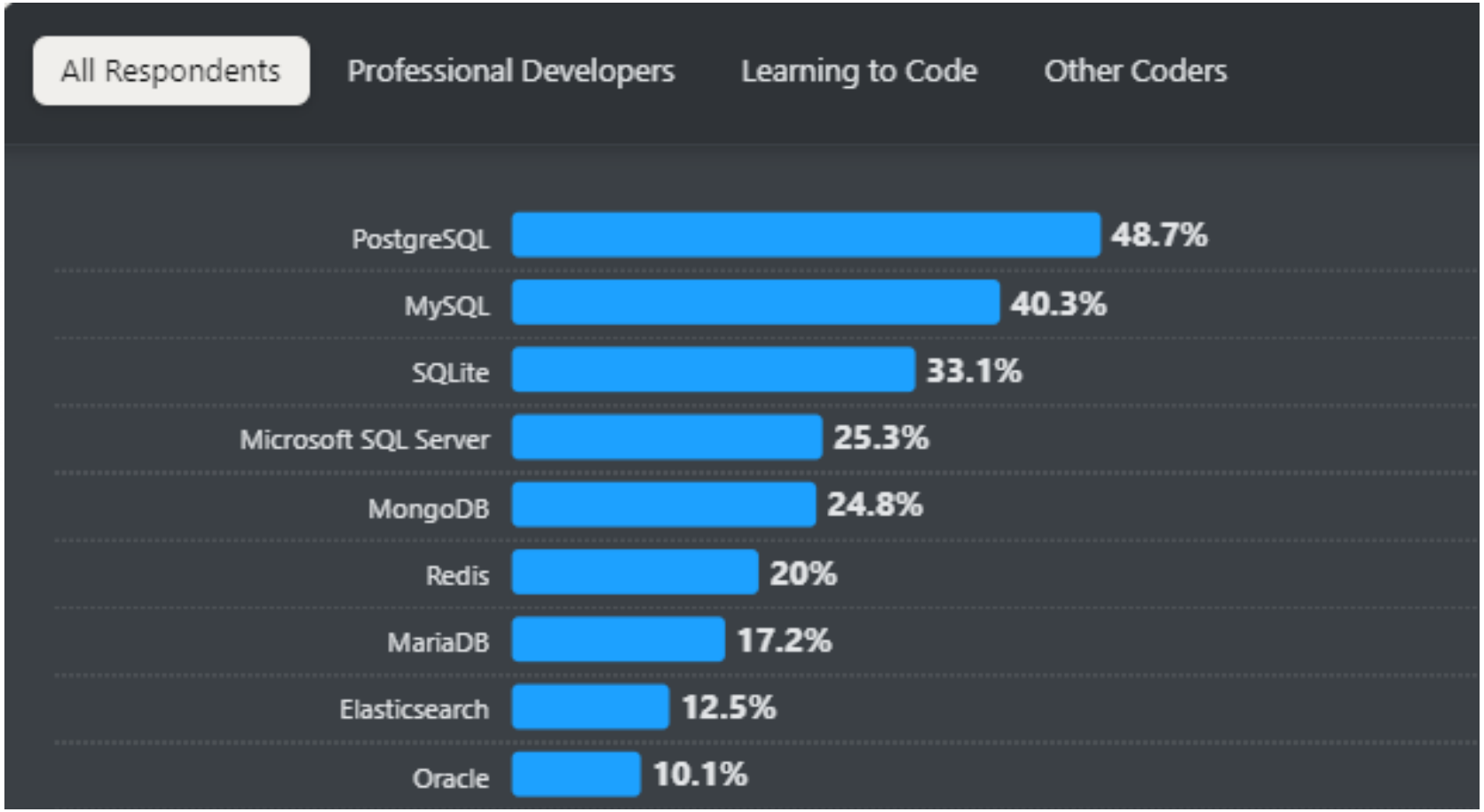
DBMS ranking – top 10

423 systems in ranking, September 2024

Rank			DBMS	Database Model	Score		
Sep 2024	Aug 2024	Sep 2023			Sep 2024	Aug 2024	Sep 2023
1.	1.	1.	Oracle	Relational, Multi-model	1286.59	+28.11	+45.72
2.	2.	2.	MySQL	Relational, Multi-model	1029.49	+2.63	-82.00
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	807.76	-7.41	-94.45
4.	4.	4.	PostgreSQL	Relational, Multi-model	644.36	+6.97	+23.61
5.	5.	5.	MongoDB	Document, Multi-model	410.24	-10.74	-29.18
6.	6.	6.	Redis	Key-value, Multi-model	149.43	-3.28	-14.26
7.	7.	↑ 11.	Snowflake	Relational	133.72	-2.25	+12.83
8.	8.	↓ 7.	Elasticsearch	Search engine, Multi-model	128.79	-1.04	-10.20
9.	9.	↓ 8.	IBM Db2	Relational, Multi-model	123.05	+0.04	-13.67
10.	10.	↓ 9.	SQLite	Relational	103.35	-1.44	-25.85

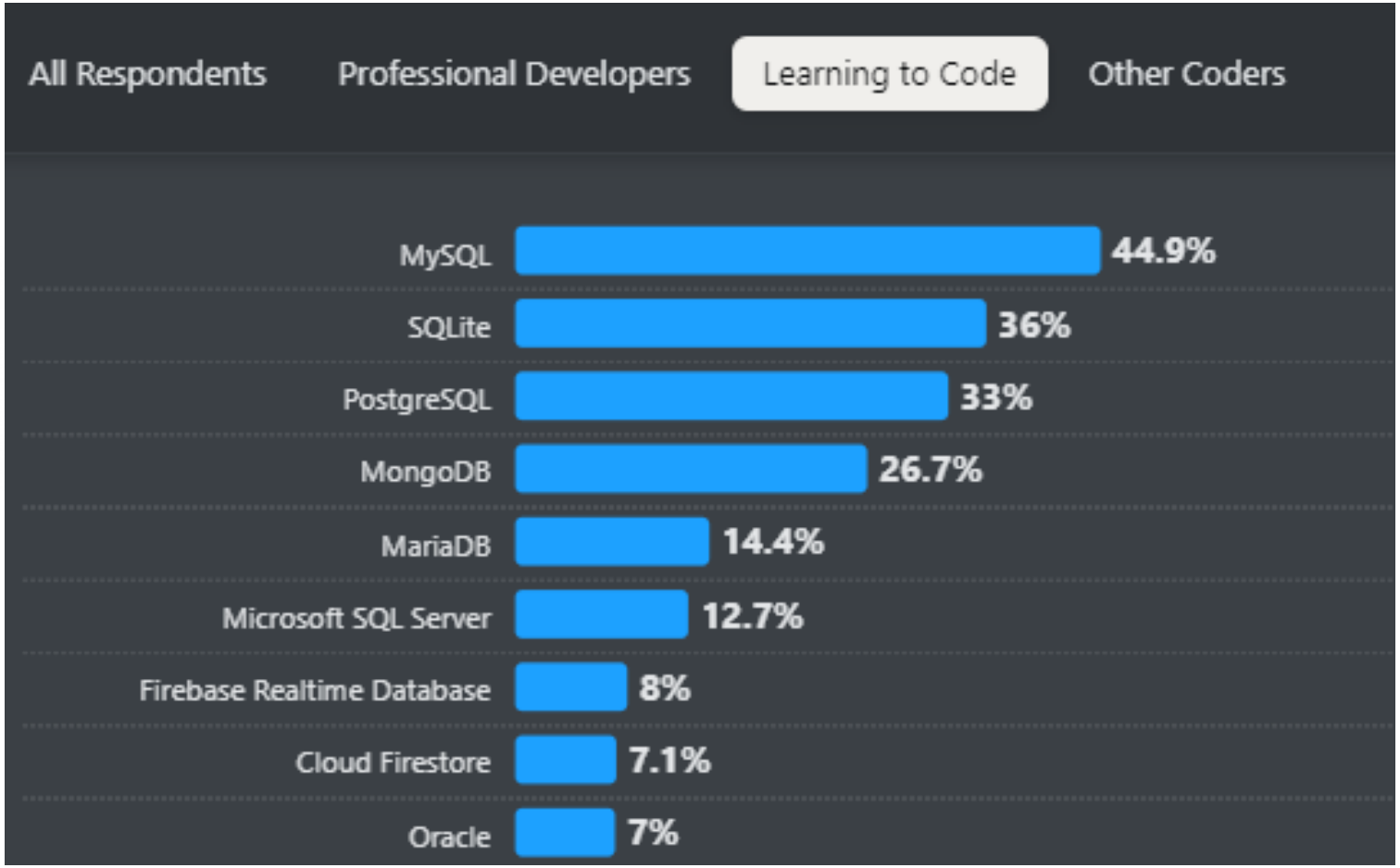
Source: https://db-engines.com/en/ranking
Method of calculating the scores: https://db-engines.com/en/ranking_definition

DBMS ranking



Source https://survey.stackoverflow.co/2024/technology#most-popular-technologies

DBMS ranking



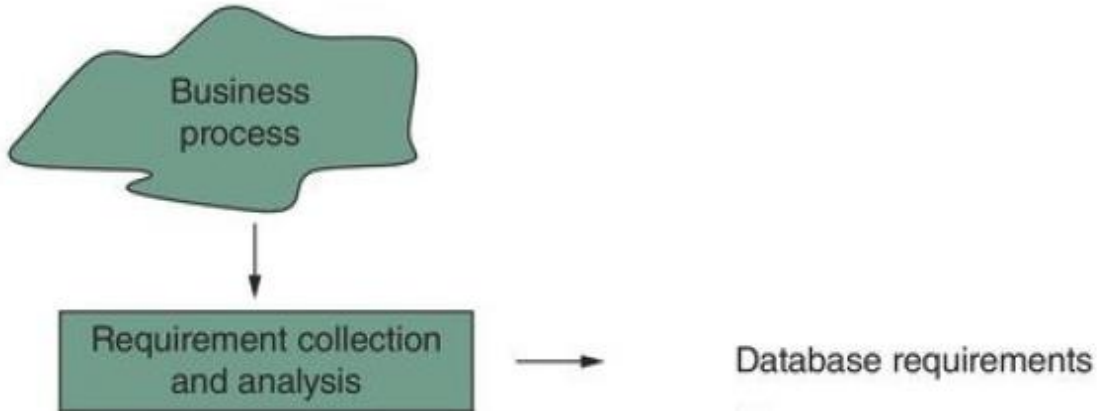
Source https://survey.stackoverflow.co/2024/technology#most-popular-technologies

Database conceptual modelling

What is a model?

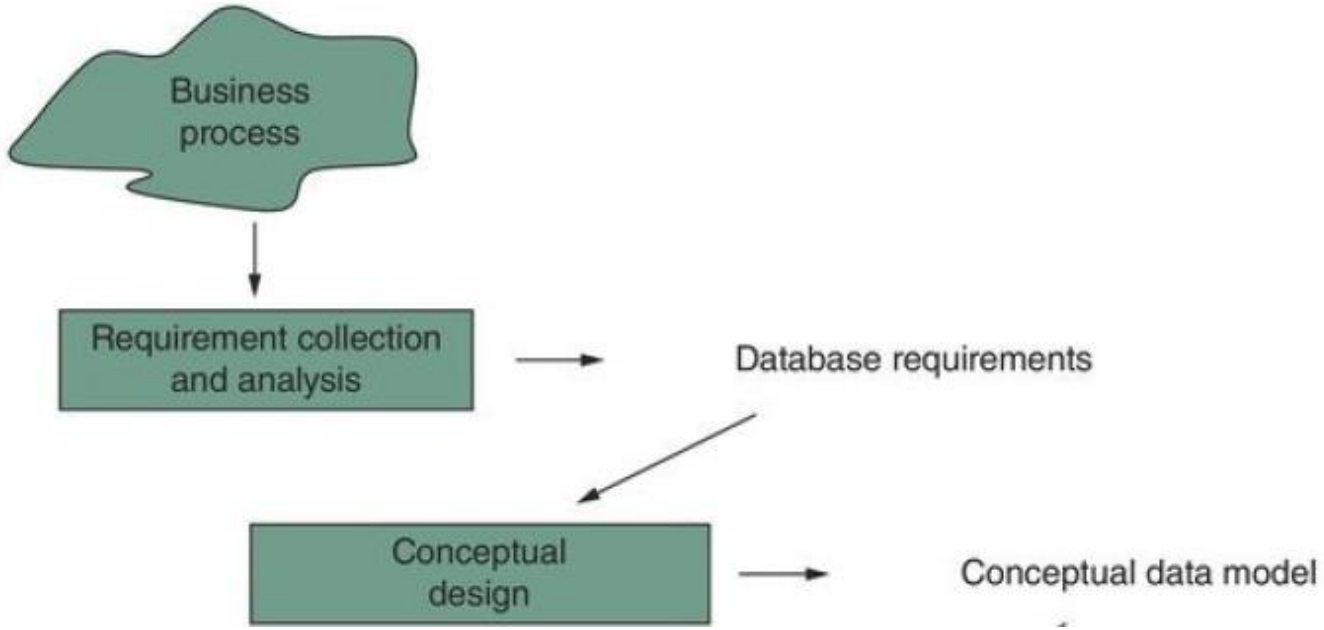
Image Source: https://www.stpetertravel.com/en/viaggi_tour/7072/coliseum-architecture-coliseum-roman-e-imperial-forum-rome-tours.html

The database design process



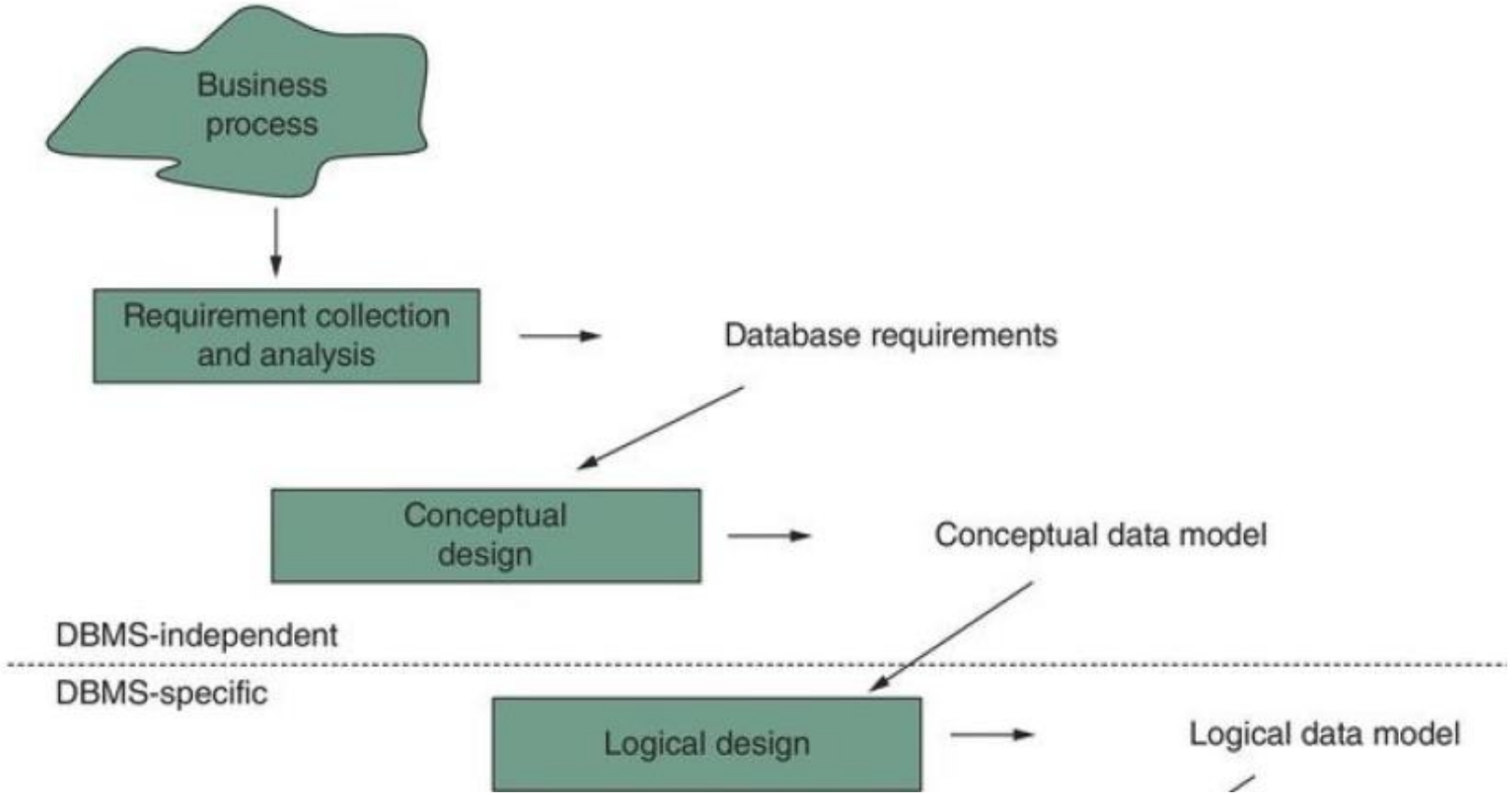
The aim is to understand the diferente steps and data needs of the process. Techniques: interviews, surveys, inspections of documents, etc

The database design process



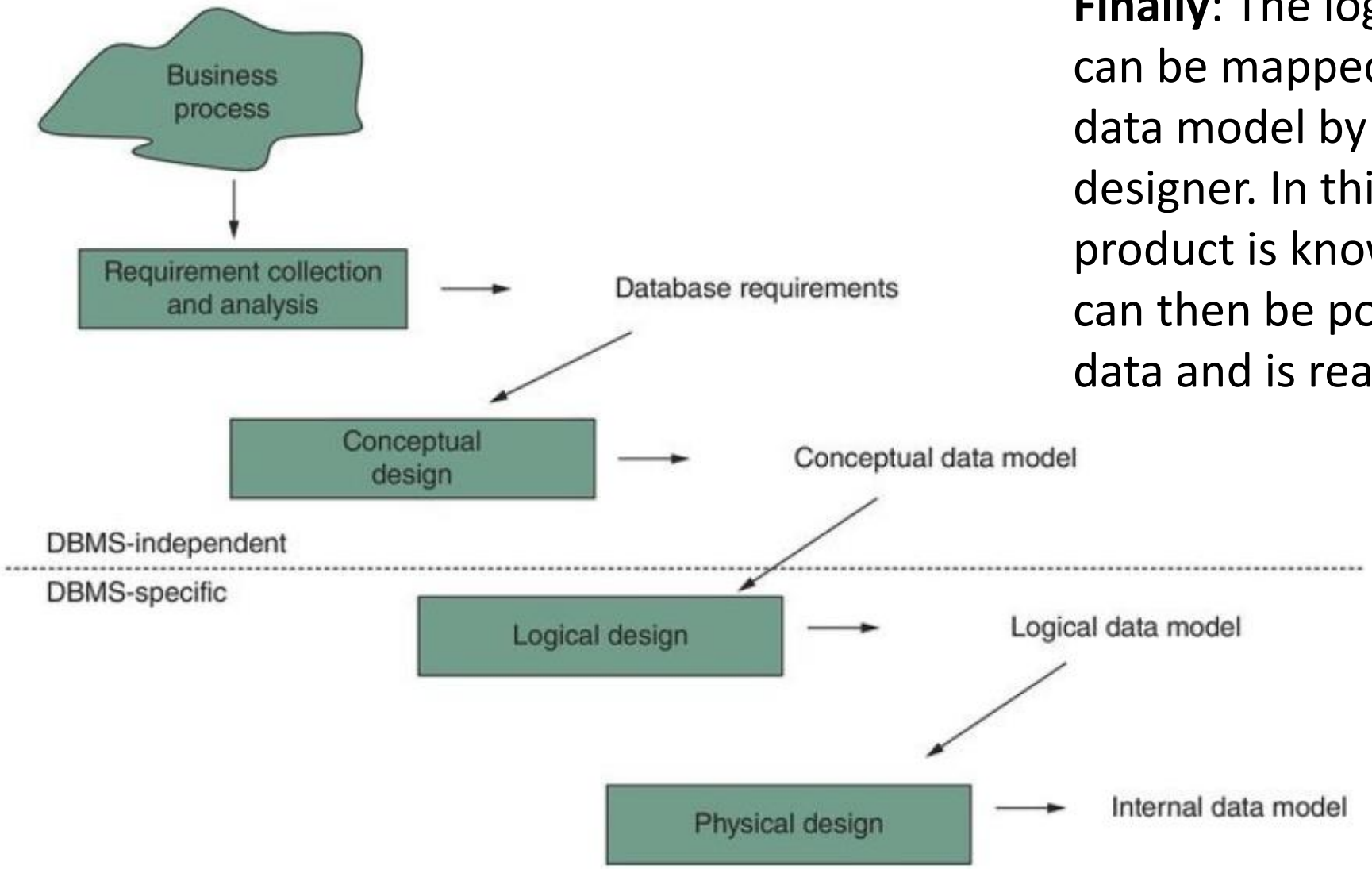
The information architect and the business user formalize the requirements in a **conceptual data model**. This is a high-level model, easy to understand for the business user and formal enough for the database designer who will use it in the next step.

The database design process



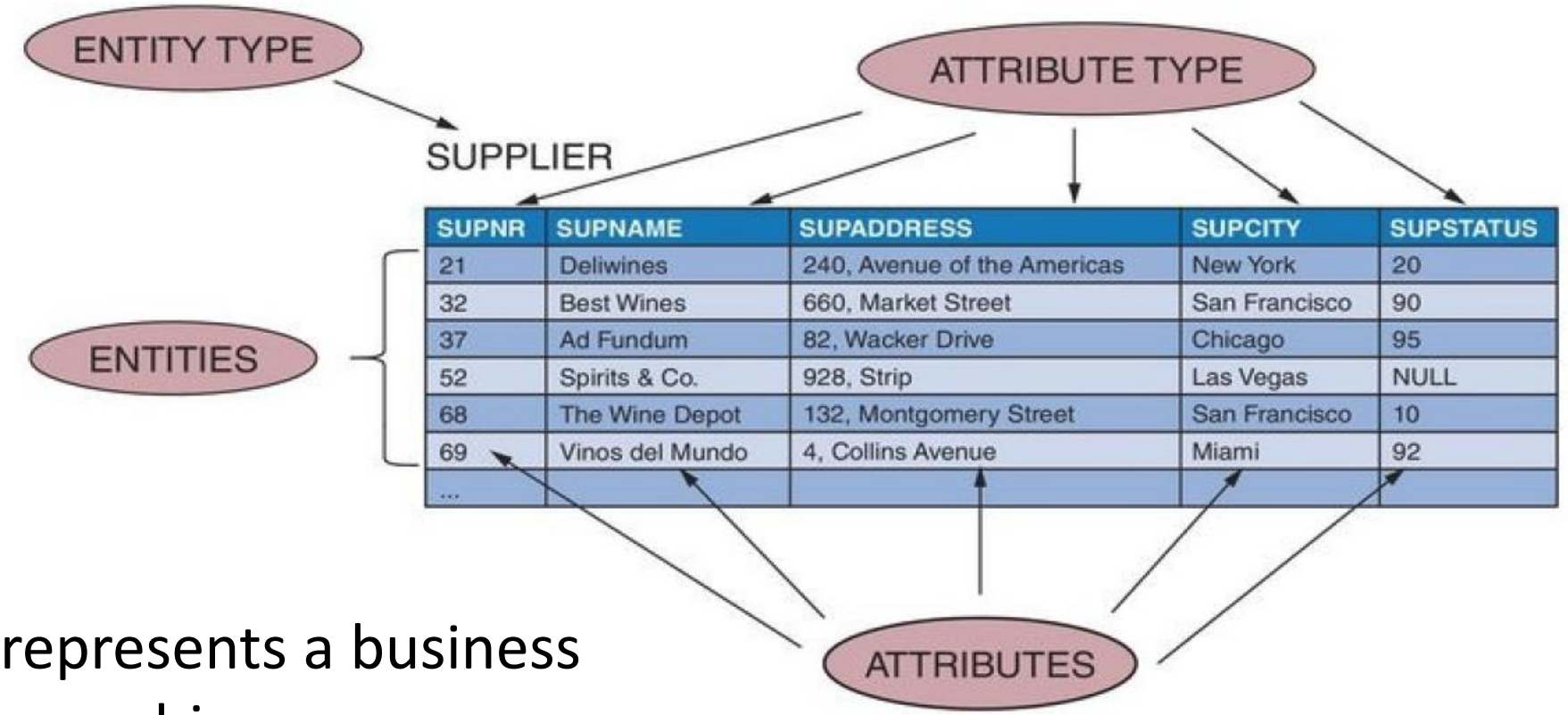
The logical data model is based upon the implementation environment. At this stage it is already known what type of DBMS (e.g., RDBMS, OODBMS, etc.) will be used, the product itself (e.g., Microsoft, IBM, Oracle) has not been decided yet.

The database design process



Finally: The logical data model can be mapped to an internal data model by the database designer. In this step, the DBMS product is known. The database can then be populated with data and is ready for use.

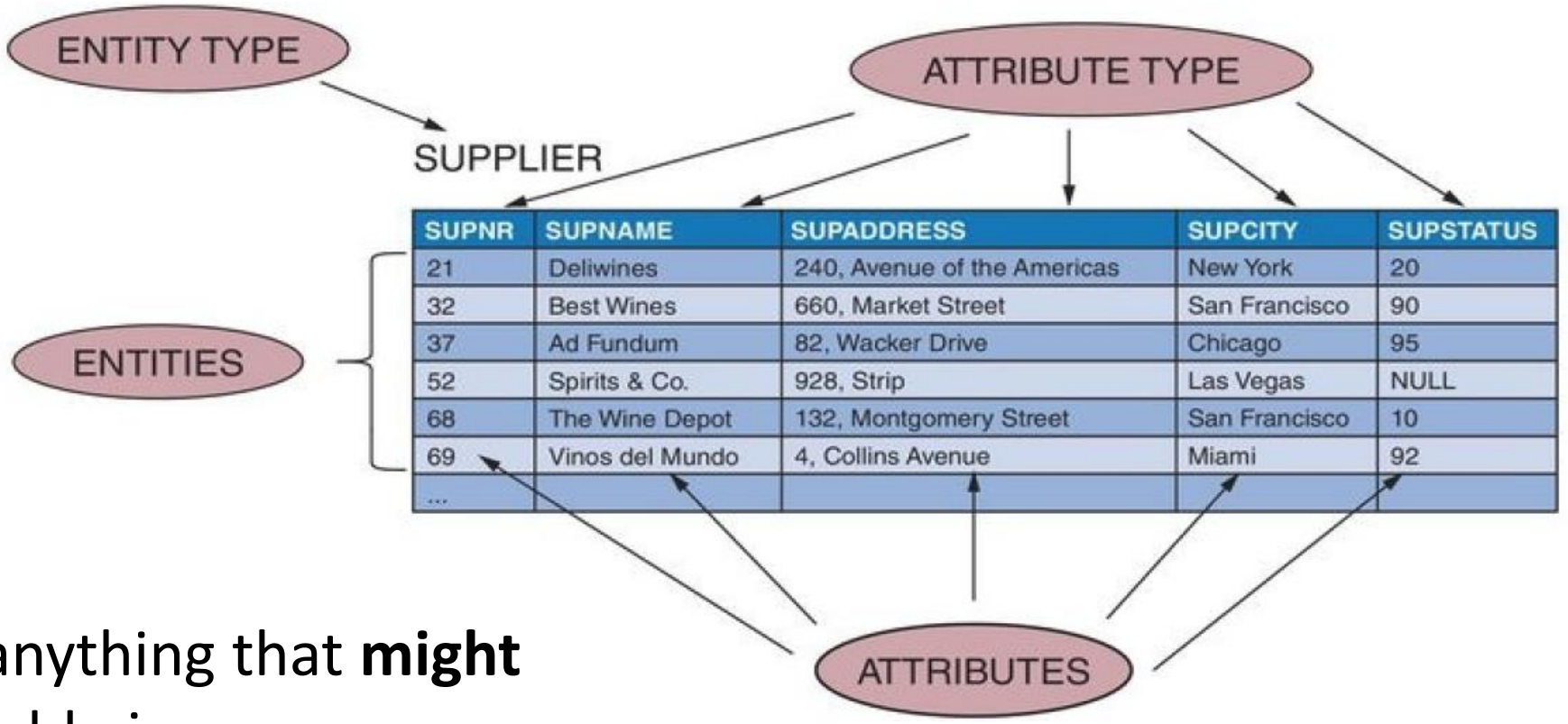
The entity relationship model



SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & Co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92
...				

An **ENTITY TYPE** represents a business concept with an unambiguous meaning to a particular set of users.
Examples?

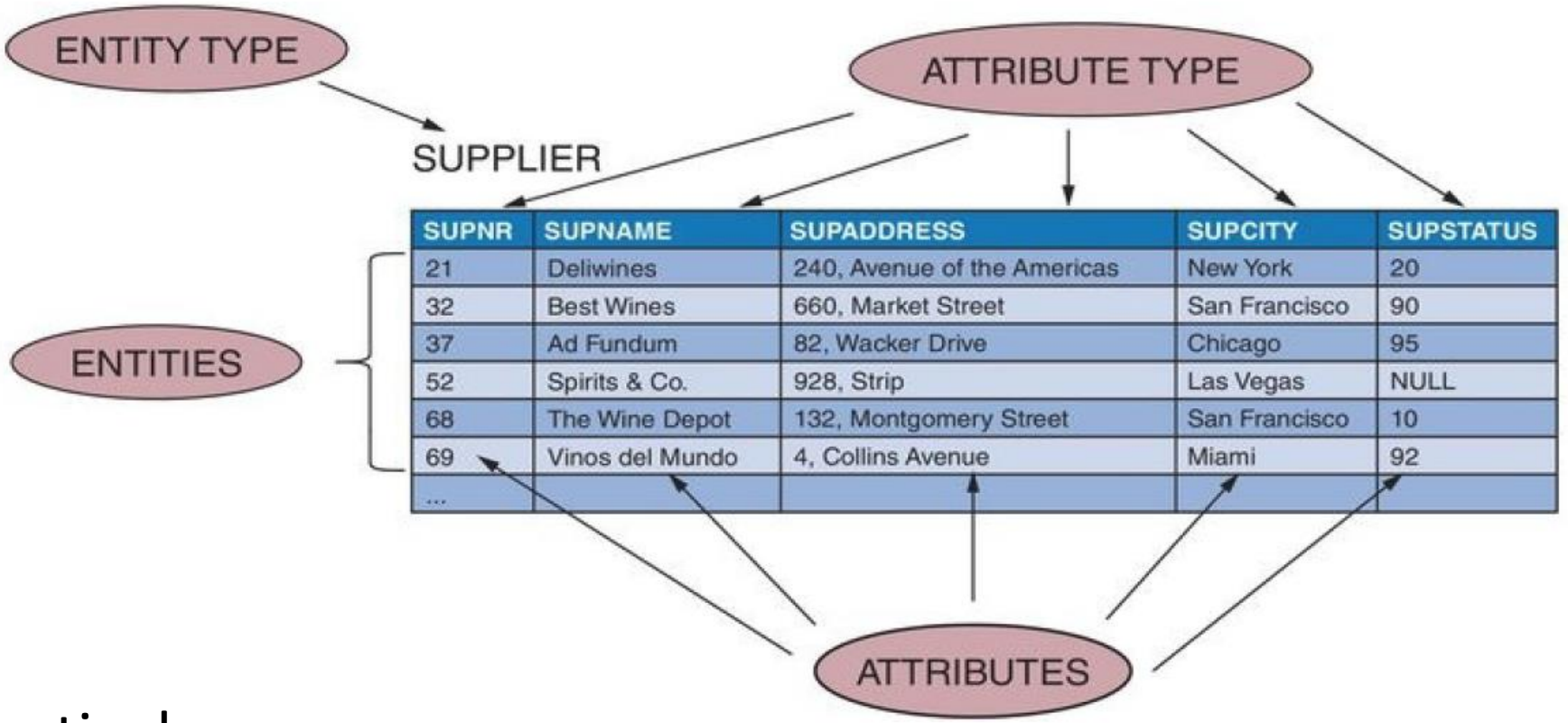
The entity relationship model



SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & Co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92
...				

"ENTITY TYPE is anything that **might** deserve its own table in your database model". (Tekstenuitleg.net)

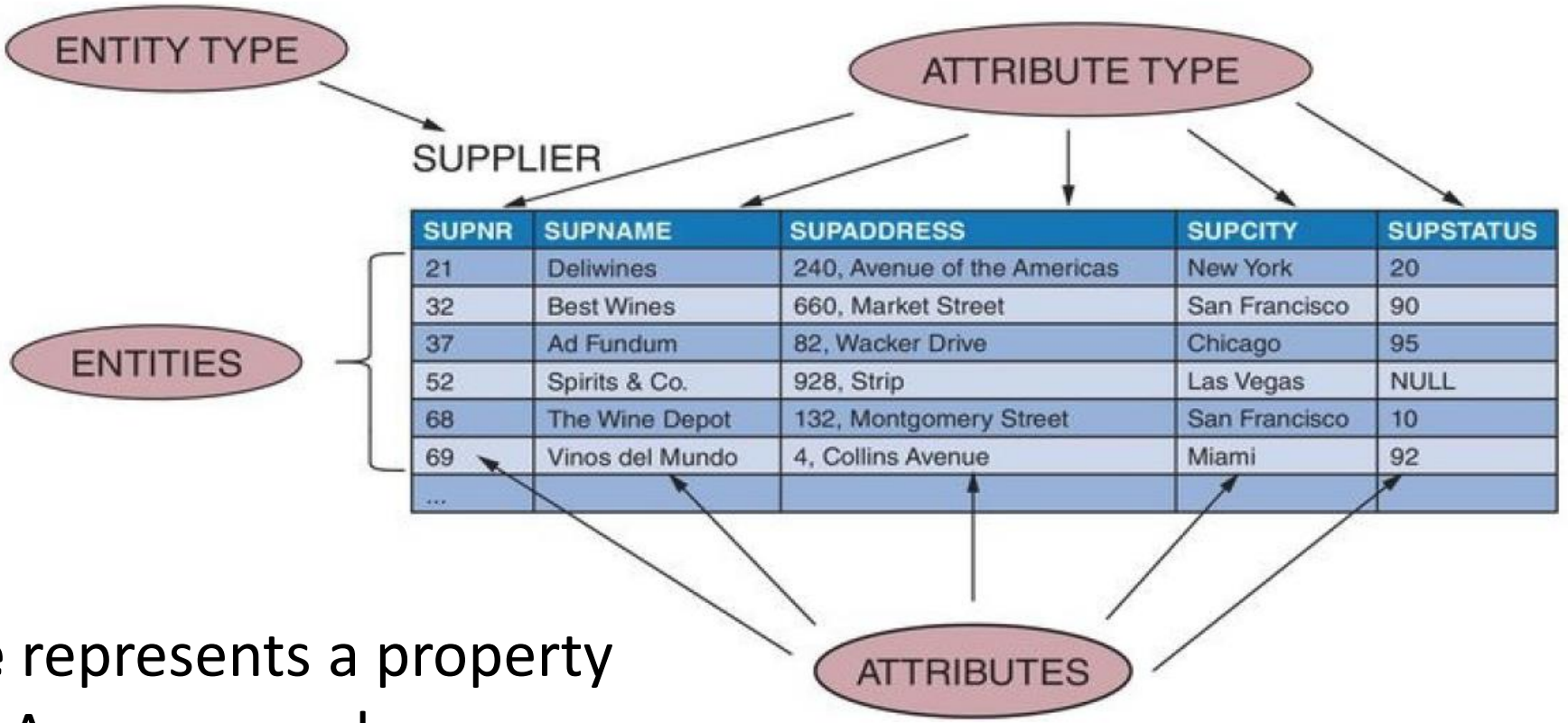
The entity relationship model



SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & Co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92
...				

An **entity** is one particular occurrence or instance of an entity type

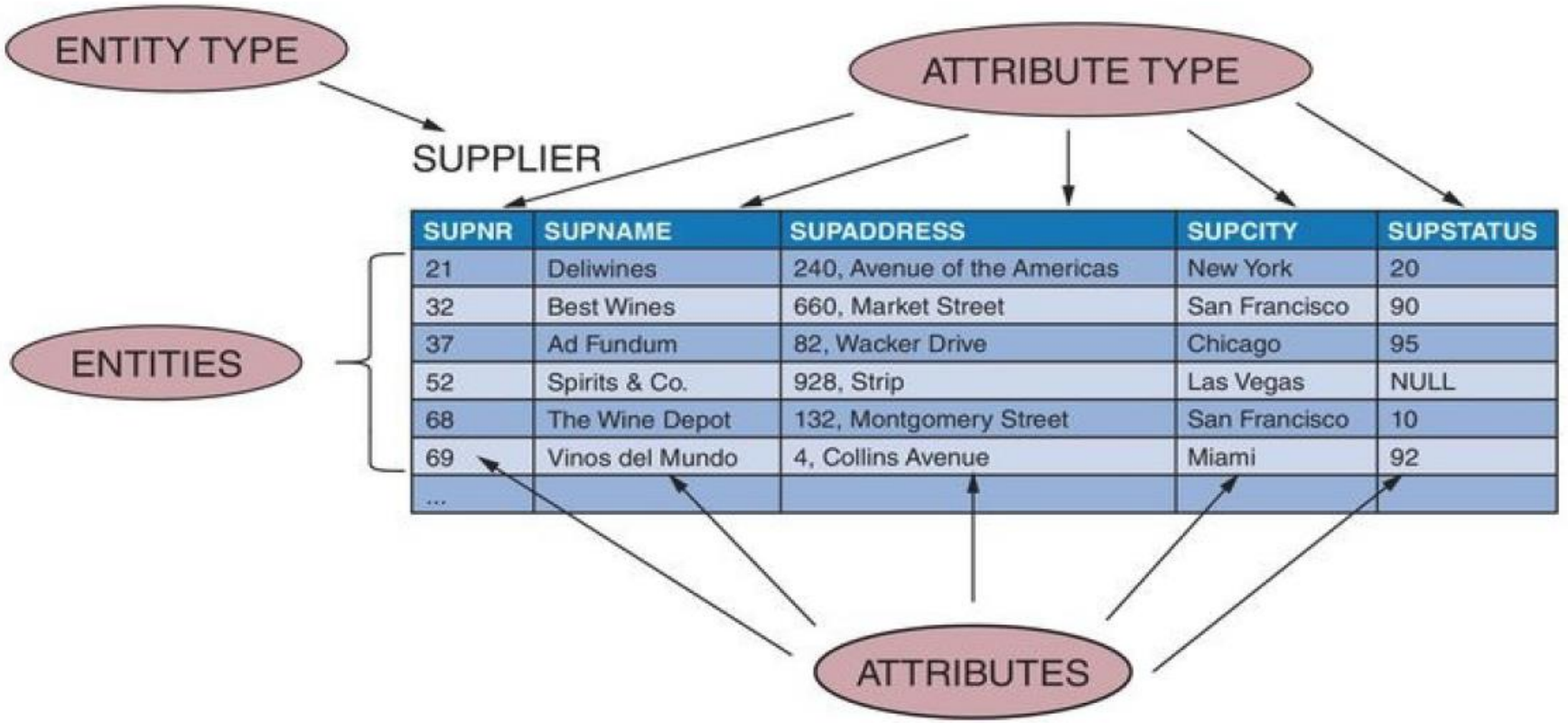
The entity relationship model



SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & Co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92
...				

An **attribute type** represents a property of an entity type. As an example, name and address are attribute types of the entity type supplier.

The entity relationship model

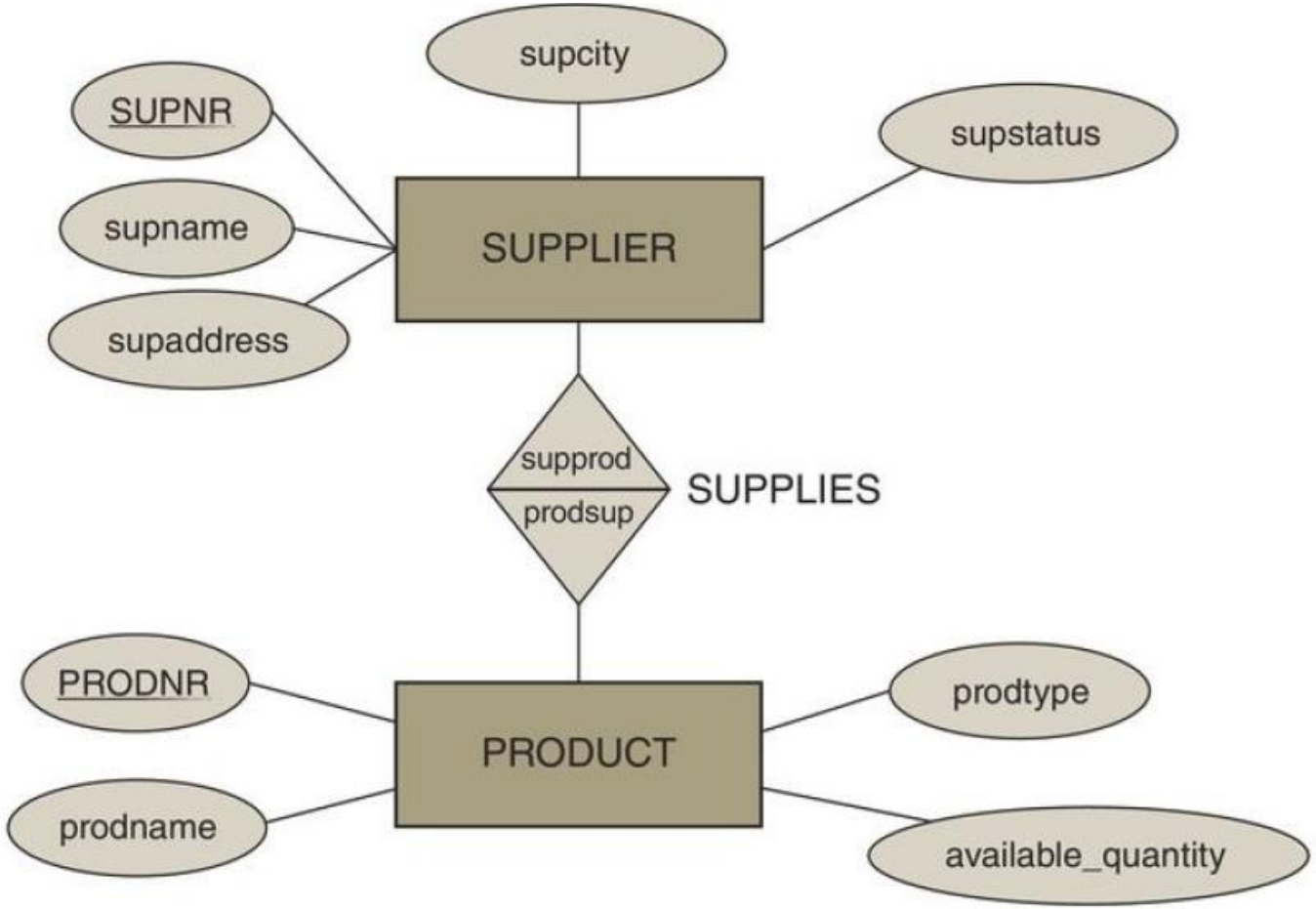


SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & Co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92
...				

An **attribute** is an instance of an attribute type

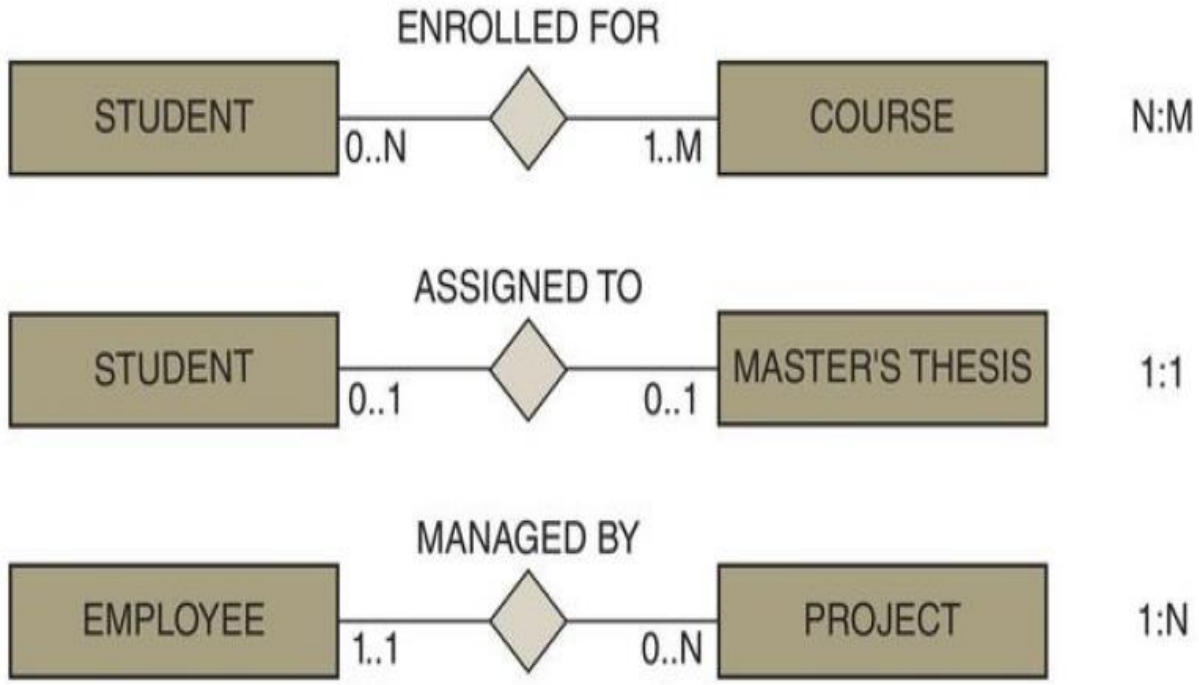
Relationship

A **relationship** represents an association between two or more entities. A **relationship type** then defines a set of relationships among instances of one, two, or more entity types.



Cardinalities

Every relationship type can be characterized in terms of its cardinalities, which specify the minimum or maximum number of relationship instances that an individual entity can participate in.



Example: A student is **enrolled for** 1 or M courses

Normalization

Data Normalization (3-Normal Forms)

Normalization of a relational model is a process of analyzing the given relations to ensure they do not contain any redundant data. The goal of normalization is to ensure that no anomalies can occur during data insertion, deletion, or update. A step-by-step procedure needs to be followed to transform an unnormalized relational model to a normalized relational model.

Data Normalization
(3-Normal Forms)

Think about the normalization forms as filters. The more filters you apply, the better your DB.

BUT….. The more normal forms, the more complex your database structure.

Relational Databases - Data Normalization

3 Normal Forms (NF):

1. The 1NF states that every attribute type must be atomic and single-valued. Hence, no composite or multi-valued attribute types are tolerated.
2. An entity type is in **2NF** when it is in **1NF** and when all of its non-key attributes are fully functionally dependent on its primary key.
3. An entity type is in **3NF** when it is in **2NF** and no non-key attribute is **transitively dependent** on the primary key.

Relational Databases - First Normal Form (1NF)

An entity type is in 1NF when it contains no repeating groups of data

"each cell in the table can have only one value, never a list of values"

Product ID	Color	Price
1	brown, yellow	$15
2	red, green	$13
3	blue, orange	$11

Relational Databases - First Normal Form

An entity type is in 1NF when it contains no repeating groups of data

"each cell in the table can have only one value, never a list of values"

Product ID	Color	Price
1	brown, yellow	$15
2	red, green	$13
3	blue, orange	$11

Does this table comply with 1NF?

Relational Databases - First Normal Form

An entity type is in 1NF when it contains no repeating groups of data

"each cell in the table can have only one value, never a list of values"

Product ID	Color	Price
1	brown, yellow	$15
2	red, green	$13
3	blue, orange	$11

Does this table comply with 1NF?

NO

What is the solution?

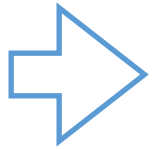
Relational Databases - First Normal Form

An entity type is in 1NF when it contains no repeating groups of data

"each cell in the table can have only one value, never a list of values"

Product ID	Color	Price
1	brown, yellow	$15
2	red, green	$13
3	blue, orange	$11

Is this solution enough?



Product ID	Color1	Color2	Price
1	brown	yellow	$15
2	red	green	$13
3	blue	orange	$11

Relational Databases - First Normal Form

An entity type is in 1NF when it contains no repeating groups of data

"each cell in the table can have only one value, never a list of values"

Product ID	Color	Price
1	brown, yellow	$15
2	red, green	$13
3	blue, orange	$11

Is this solution enough?

NO

Product ID	Color1	Color2	Price
1	brown	yellow	$15
2	red	green	$13
3	blue	orange	$11

Relational Databases - First Normal Form

An entity type is in 1NF when it contains no repeating groups of data

"each cell in the table can have only one value, never a list of values"

Product ID	Color	Price
1	brown, yellow	$15
2	red, green	$13
3	blue, orange	$11

1NF

Product ID	Price
1	$15
2	$13
3	$11

ColorID	ProductID	Color
9001	1	brown
9002	2	red
9003	3	blue
9004	1	yellow
9005	2	green
9006	3	orange

Example – 1NF

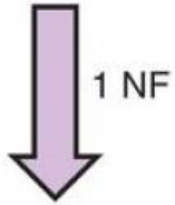
Normalize the following table that contains information about department location:

DNUMBER	DLOCATION	DMGRSSN
15	{New York, San Francisco}	110
20	Chicago	150
30	{Chicago, Boston}	100

Example – 1NF

Solution:

DNUMBER	DLOCATION	DMGRSSN
15	{New York, San Francisco}	110
20	Chicago	150
30	{Chicago, Boston}	100



DEPARTMENT

DNUMBER	DMGRSSN
15	110
20	150
30	100

DEP-LOCATION

DNUMBER	DLOCATION
15	New York
15	San Francisco
20	Chicago
30	Chicago
30	Boston

Relational Databases - Second Normal Form (2NF)

An entity type is in 2NF when it is in 1NF and when all of its non-key attributes are functionally dependent on its primary key.

But what is **functional dependency?** Let´s explain it with one example:

Table Person

SSN	First name	Last name	Date of birth	Address	Phone number
123-98-1234	Cindy	Cry	15-05-1983	Los Angeles	123-456-7891
121-45-6145	John	O'Neill	30-01-1980	Paris	568-974-2562
658-78-2369	John	Lannoy	30-01-1980	Dallas	963-258-7413

Source: https://vertabelo.com/blog/normalization-1nf-2nf-3nf/

Relational Databases - Second Normal Form (2NF)

Table **Person**

SSN	First name	Last name	Date of birth	Address	Phone number
123-98-1234	Cindy	Cry	15-05-1983	Los Angeles	123-456-7891
121-45-6145	John	O'Neill	30-01-1980	Paris	568-974-2562
658-78-2369	John	Lannoy	30-01-1980	Dallas	963-258-7413

The column SSN (Social Security Number) determines the values in columns first_name, last_name, date_of_birth, address, and phone_number.

This means that if we had two rows with the same value in the SSN column, then values in columns first_name, last_name, date_of_birth, address, and phone_number would be equal.

A situation like this is called **functional dependency.**

Source: https://vertabelo.com/blog/normalization-1nf-2nf-3nf/

Relational Databases - Second Normal Form (2NF)

An entity type is in 2NF when it is in 1NF and when all of its non-key attributes are fully dependent on its primary key.

"features should be fully dependent on the entire primary key"

Primary Keys

OrderNumber	ProductID	ProductName
1	232	Pespsi
2	234	Coca-Cola
3	241	Polar

Does it comply with the 2NF?

Relational Databases - Second Normal Form

An entity type is in 2NF when it is in 1NF and when all of its non-key attributes are fully dependent on its primary key.

"features should be fully dependent on the entire primary key"

Primary Keys

OrderNumber	ProductID	ProductName
1	232	Pespsi
2	234	Coca-Cola
3	241	Polar

Fails because Product Name Depends on Product ID but not on Order Number.
Solution?

Relational Databases - Second Normal Form

An entity type is in 2NF when it is in 1NF and when all of its non-key attributes are fully dependent on its primary key.

"features should be fully dependent on the entire primary key"

OrderNumber	ProductID	ProductName
1	232	Pespsi
2	234	Coca-Cola
3	241	Polar

OrderNumber	ProductID
1	232
2	
3	

ProductID	ProductName
232	Pespsi
234	Coca-Cola
241	Polar

Example – 2NF

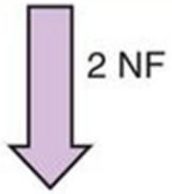
Normalize to the 2NF. The following table contains information about the project and the hours each employee worked on the project:

SSN	PNUMBER	PNAME	HOURS
100	1000	Hadoop	50
220	1200	CRM	200
280	1000	Hadoop	40
300	1500	Java	100
120	1000	Hadoop	120

Example – 2NF

Solution:

SSN	PNUMBER	PNAME	HOURS
100	1000	Hadoop	50
220	1200	CRM	200
280	1000	Hadoop	40
300	1500	Java	100
120	1000	Hadoop	120

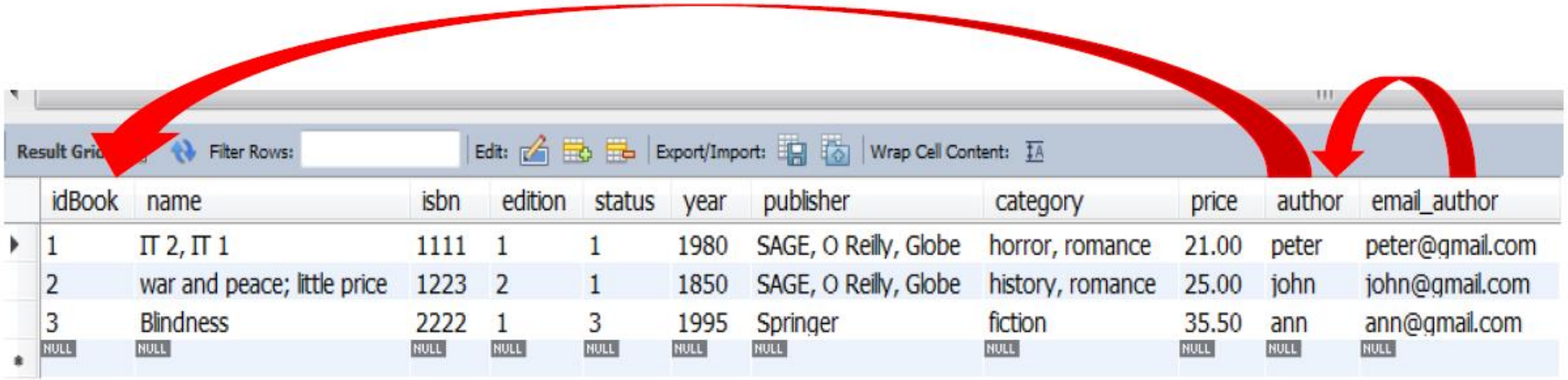


PNUMBER	PNAME
1000	Hadoop
1200	CRM
1500	Java

SSN	PNUMBER	HOURS
100	1000	50
220	1200	200
280	1000	40
300	1500	100
120	1000	120

Relational Databases - Third Normal Form

Transitive dependency:



idBook	name	isbn	edition	status	year	publisher	category	price	author	email_author
1	IT 2, IT 1	1111	1	1	1980	SAGE, O Reilly, Globe	horror, romance	21.00	peter	peter@gmail.com
2	war and peace; little price	1223	2	1	1850	SAGE, O Reilly, Globe	history, romance	25.00	john	john@gmail.com
3	Blindness	2222	1	3	1995	Springer	fiction	35.50	ann	ann@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Author email depends on author, author depends on book_id

Relational Databases - Third Normal Form

An entity type is in 3NF when it is in 2NF, and non-key attributes are directly (non-transitively) dependent on the primary key.

In other words: non-prime attributes must be functionally dependent on the key(s), but they **must not** depend on another non-prime attribute.

Example: What are the functional dependencies on the following table:

order_id	date	customer	customer email
1/2020	2020-01-15	Jason White	white@example.com
2/2020	2020-01-16	Mary Smith	msmith@mailinator.com
3/3030	2020-01-17	Jacob Albertson	jasobal@example.com
4/2020	2020-01-18	Bob Dickinson	bob@fakemail.com

Source: https://vertabelo.com/blog/normalization-1nf-2nf-3nf/

Relational Databases - Third Normal Form

Example: What are the functional dependencies on the following table:

order_id	date	customer	customer email
1/2020	2020-01-15	Jason White	white@example.com
2/2020	2020-01-16	Mary Smith	msmith@mailinator.com
3/3030	2020-01-17	Jacob Albertson	jasobal@example.com
4/2020	2020-01-18	Bob Dickinson	bob@fakemail.com

Functional dependencies:

- date depends on order_id
- customer depends on order_id
- **customer email depends on customer (transitively on order_id)**

Source: https://vertabelo.com/blog/normalization-1nf-2nf-3nf/

Relational Databases - Third Normal Form

Solution:
We split it in two tables:
Orders and Customers

Orders

order_id	date	customer
1/2020	2020-01-15	Jason White
2/2020	2020-01-16	Mary Smith
3/3030	2020-01-17	Jacob Albertson
4/2020	2020-01-18	Bob Dickinson

Customers

customer	customer email
Jason White	white@example.com
Mary Smith	msmith@mailinator.com
Jacob Albertson	jasobal@example.com
Bob Dickinson	bob@fakemail.com

Source: https://vertabelo.com/blog/normalization-1nf-2nf-3nf/

Relational Databases - Third Normal Form

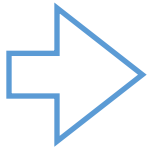
An entity type is in 3NF when it is in 2NF and no non-key attribute is transitively dependent on the primary key.

Tournament	Year	Winner	Winner DoB
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Indiana Invitational	1999	Chip Masterson	14 March 1977

Relational Databases - Third Normal Form

An entity type is in 3NF when it is in 2NF and no non-key attribute is transitively dependent on the primary key.

Tournament	Year	Winner	Winner DoB
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Indiana Invitational	1999	Chip Masterson	14 March 1977



Tournament	Year	Winner
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Indiana Invitational	1999	Chip Masterson

Winner	Winner DoB
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968
Chip Masterson	14 March 1977

Example – 3NF

Normalize the following table that contains information about employees and departments:

SSN	NAME	DNUMBER	DNAME	*DMGRSSN*
10	O´Reilly	10	Marketing	210
22	Donovan	30	Logistics	150
28	Bush	10	Marketing	210
30	Jackson	20	Finance	180
12	Thompson	10	Marketing	210

Example – 3NF

Solution:

SSN	NAME	DNUMBER	DNAME	DMGRSSN
10	O´Reilly	10	Marketing	210
22	Donovan	30	Logistics	150
28	Bush	10	Marketing	210
30	Jackson	20	Finance	180
12	Thompson	10	Marketing	210

3 NF

SSNR	NAME	DNUMBER
10	O´Reilly	10
22	Donovan	30
28	Bush	10
30	Jackson	20
12	Thompson	10

DNUMBER	DNAME	DMGRSSN
10	Marketing	210
30	Logistics	150
20	Finance	180

How to install MySQL

- **MySQL installation will NOT be done in labs.**
- You can download MySQL community edition from here (choose **Windows (x86, 64-bit), MSI Installer**): https://dev.mysql.com/downloads/mysql/
- **If requested** the C++ redistributable package, you can download from here: https://aka.ms/vs/17/release/vc_redist.x64.exe
- You can download MySQL workbench from here: https://dev.mysql.com/downloads/workbench/

Installation videos:

- Windows: https://www.youtube.com/watch?v=u96rVINbAUI

https://www.youtube.com/watch?v=v8OYcHgnshY

https://www.youtube.com/watch?v=s0YoPLbox40 (in Portuguese)

- Mac: https://www.youtube.com/watch?v=-BDbOOY9jsc

https://www.youtube.com/watch?v=5tjKVkbWglY

- Linux: https://www.youtube.com/watch?v=zRfI79BHf3k

More info about installation:

- The Workbench: https://dev.mysql.com/doc/workbench/en/wb-installing.html
- The MySQL server: https://dev.mysql.com/doc/refman/8.0/en/installing.html

END OF LECTURE 1