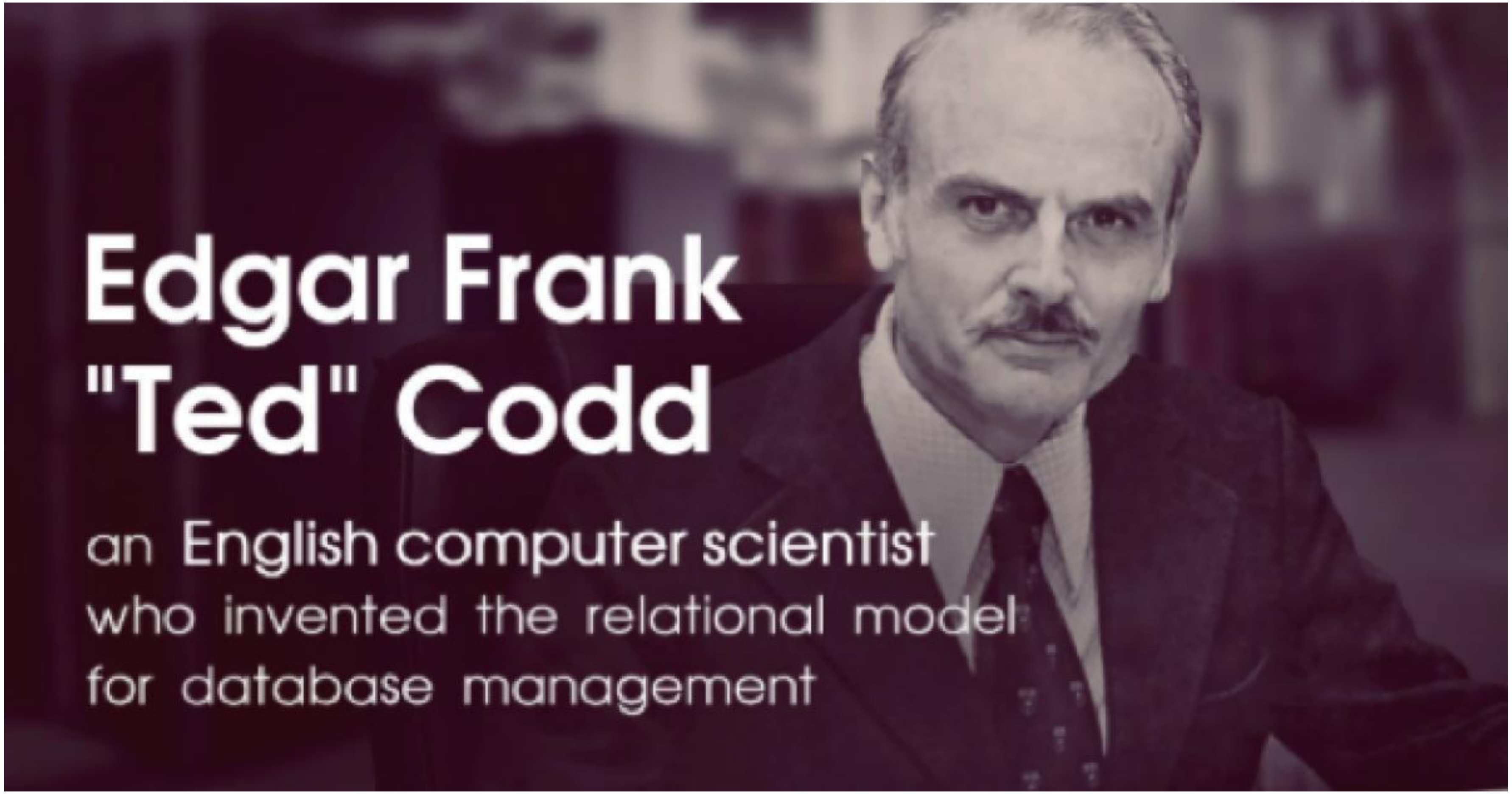
Normalization

<Let’s explore the 1NF, 2NF and 3NF>

CODEUS_

About me:

- Senior Software Engineer
- Co-Organizer of Codeus community

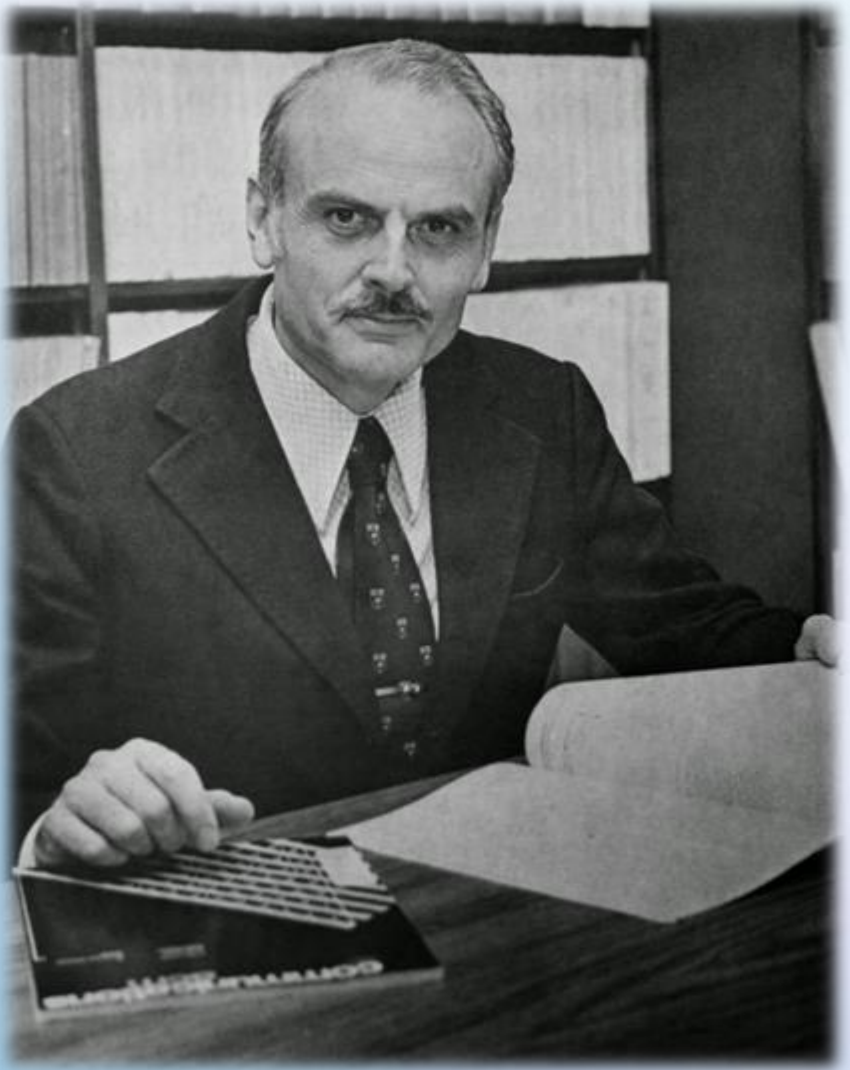


Edgar Frank "Ted" Codd

an English computer scientist who invented the relational model for database management

History

<Know your history to not repeat mistakes>



Edgar F. Codd, a British computer scientist working at IBM, invented the concept of normalization in 1970.

He introduced the relational database model in his paper "A Relational Model of Data for Large Shared Data Banks" and subsequently developed the theory of normalization.

Codd's work revolutionized database design and led to the development of the relational database management systems (RDBMS) that dominate the industry today.

His contributions to database theory earned him the Turing Award in 1981.

What is Normalization?

<Let's explore what is normalization?>

CODEUS_

Normalization is a structured process that breaks down large, complex tables into smaller, more manageable ones.

Each table focuses on a specific entity or concept, with relationships between tables maintained through primary and foreign keys.

The process follows a series of rules or "normal forms" (1NF, 2NF, 3NF, BCNF, 4NF, 5NF), each addressing specific types of data redundancy and dependency issues.

Reason?

<For what reason was it invented?>

Eliminate Data Redundancy: Prevents storing the same data in multiple places.

Prevent Update Anomalies: Ensures changes to data need only be made in one place.

Avoid Insertion Anomalies: Makes it possible to add new records without requiring complete information for unrelated entities.

Prevent Deletion Anomalies: Ensures deleting data about one entity doesn't accidentally remove data about other entities.

Improve Query Performance: While joins between normalized tables can be expensive for some operations, properly normalized databases often perform better for updates and offer more flexible querying options.

Enhance Data Integrity: Makes it easier to enforce constraints and maintain accurate data.

Eliminate Data Redundancy: Prevents storing the same data in multiple places.

Prevent Update Anomalies: Ensures changes to data need only be made in one place.

Avoid Insertion Anomalies: Makes it possible to add new records without requiring complete information for unrelated entities.

Prevent Deletion Anomalies: Ensures deleting data about one entity doesn't accidentally remove data about other entities.

Improve Query Performance: While joins between normalized tables can be expensive for some operations, properly normalized databases often perform better for updates and offer more flexible querying options.

Enhance Data Integrity: Makes it easier to enforce constraints and maintain accurate data.

1 Normal Form

<Let’s explore what the 1NF>

1NF

CODEUS_

1NF

ROWS ORDER DOESN'T MATTER

I need some cars

AUDI, PORSHE, VOLVO, TOYOTA

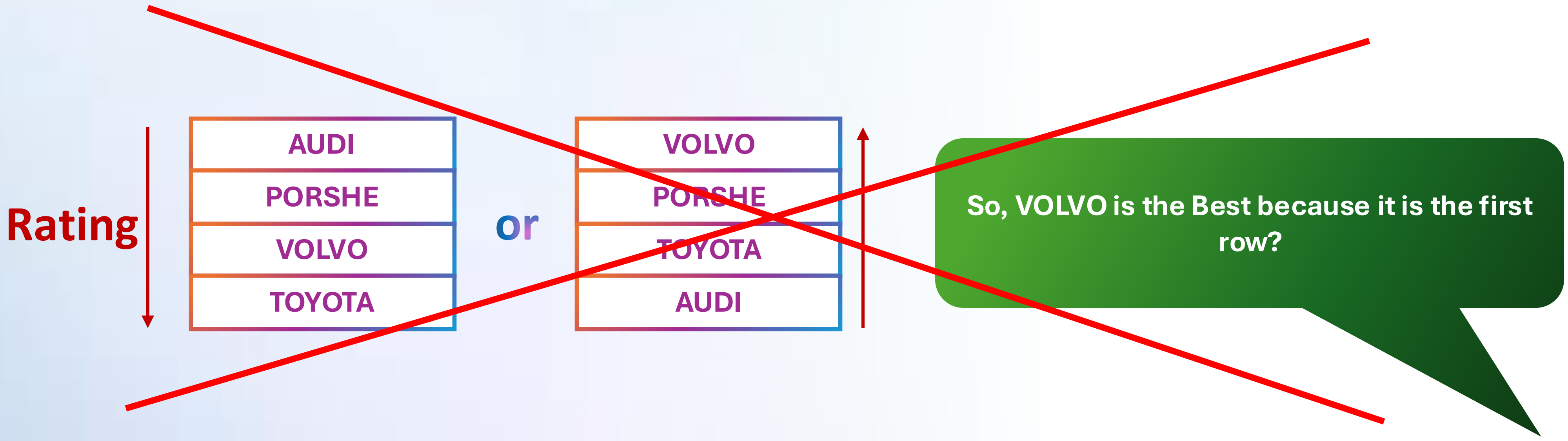
equivalent

VOLVO, PORSHE, TOYOTA, AUDI

1NF

ROWS ORDER DOESN'T MATTER

SELECT best_cars FROM cars;



Rating

AUDI
PORSHE
VOLVO
TOYOTA

or

VOLVO
PORSHE
TOYOTA
AUDI

So, VOLVO is the Best because it is the first row?

Using ROW order to convey information violates 1NF!

1NF

ROWS ORDER DOESN'T MATTER

```
SELECT car_name, rating FROM cars;
```

SALON	RATING
PORSHE	best
SKODA	good
FIAT	normal
CHERY	bad

✓

CODEUS_

1NF

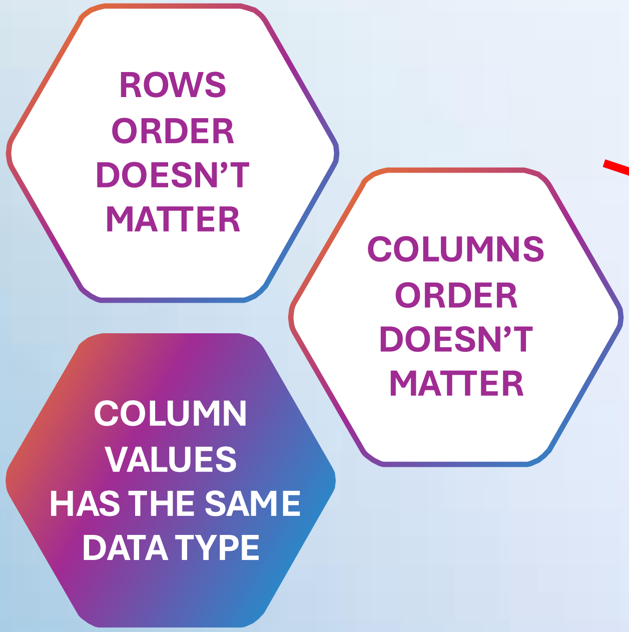
ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

The same story as with ROWS.

Using COLUMN order to convey information violates 1NF!

1NF



ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

SALON	RATING (int)
PORSHE	9
SKODA	7
FIAT	5-6
CHERY	2

Mixing the datatypes within the same column violates 1NF!

You probably won't meet such issue, unless somebody decided to use TEXT and put in such column different stuff.

1NF

ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES

(no repeating groups)

SALON	WHERE_TO_BUY
AUDI	Official distributor: Kyiv, Panov str. 10; Reseller: Kyiv, Demyana 56/2
PORSHE	Authorized Dealer: Kyiv, Vynna 156; Certified Auto Broker: Lviv, Lopatina 90A
VOLVO	Official distributor: Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker: Lviv, Lopatina 90A

Repeating groups!

If you try to find the Authorized Dealer in Kyiv that can sell Porshe, you need 'play' with query to do so…
Another point that such query will be 'heavy'!

1NF

ROWS ORDER DOESN’T MATTER

COLUMNS ORDER DOESN’T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES (no repeating groups)

SALON	WHERE_TO_BUY
AUDI	Official distributor: Kyiv, Panov str. 10; Reseller: Kyiv, Demyana 56/2
PORSHE	Authorized Dealer: Kyiv, Vynna 156; Certified Auto Broker: Lviv, Lopatina 90A
VOLVO	Official distributor: Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker: Lviv, Lopatina 90A

solution

SALON	offical_ditributor	reseller	auth_diller	cert_broker
AUDI	Kyiv, Panov str. 10	Kyiv, Demyana 56/2	NULL	NULL
PORSHE	NULL	NULL	Kyiv, Vynna 156;	NULL
VOLVO	Kyiv, Panov str. 10	NULL	NULL	Kyiv, Panov str. 10
TOYOTA	NULL	NULL	NULL	Lviv, Lopatina 90A

👍 👎

1NF

ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES
(no repeating groups)

SALON	WHERE_TO_BUY
AUDI	Official distributor: Kyiv, Panov str. 10; Reseller: Kyiv, Demyana 56/2
PORSHE	Authorized Dealer: Kyiv, Vynna 156; Certified Auto Broker: Lviv, Lopatina 90A
VOLVO	Official distributor: Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker: Lviv, Lopatina 90A

solution

SALON	offical_ditributor	reseller	auth_diller	cert_broker
AUDI	Kyiv, Panov str. 10	Kyiv, Demyana 56/2	NULL	NULL
PORSHE	NULL	NULL	Kyiv, Vynna 156;	NULL
VOLVO	Kyiv, Panov str. 10	NULL	NULL	Kyiv, Panov str. 10
TOYOTA	NULL	NULL	NULL	Lviv, Lopatina 90A

What if we have a new seller type?

We need to create a new column -_-

1NF

ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES (no repeating groups)

SALON	WHERE_TO_BUY
AUDI	Official distributor: Kyiv, Panov str. 10; Reseller: Kyiv, Demyana 56/2
PORSHE	Authorized Dealer: Kyiv, Vynna 156; Certified Auto Broker: Lviv, Lopatina 90A
VOLVO	Official distributor: Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker: Lviv, Lopatina 90A

solution

SALON	offical_ditributor	reseller	auth_diller	cert_broker
AUDI	Kyiv, Panov str. 10	Kyiv, Demyana 56/2	NULL	NULL
PORSHE	NULL	NULL	Kyiv, Vynna 156;	NULL
VOLVO	Kyiv, Panov str. 10	NULL	NULL	Kyiv, Panov str. 10
TOYOTA	NULL	NULL	NULL	Lviv, Lopatina 90A

It's not okay to use DDL (Data Definition Language) to just support a new type, or delete some seller type!

1NF

ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES
(no repeating groups)

SALON	WHERE_TO_BUY
AUDI	Official distributor: Kyiv, Panov str. 10; Reseller: Kyiv, Demyana 56/2
PORSHE	Authorized Dealer: Kyiv, Vynna 156; Certified Auto Broker: Lviv, Lopatina 90A
VOLVO	Official distributor: Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker: Lviv, Lopatina 90A

solution

SALON	SELLER_TYPE	ADDRESS
AUDI	Official distributor	Kyiv, Demyana 56/2
AUDI	Reseller	Kyiv, Demyana 56/2
PORSHE	Authorized Dealer	Kyiv, Vynna 156
PORSHE	Certified Auto Broker	Lviv, Lopatina 90A
VOLVO	Official distributor	Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A

✓

1NF

ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES (no repeating groups)

DATA INTEGRITY (?)

SALON	SELLER_TYPE	ADDRESS
AUDI	Official distributor	Kyiv, Demyana 56/2
AUDI	Reseller	Kyiv, Demyana 56/2
PORSHE	Authorized Dealer	Kyiv, Vynna 156
PORSHE	Certified Auto Broker	Lviv, Lopatina 90A
VOLVO	Official distributor	Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A

Does it follow 1NF?

👍 👎

1NF

ROWS ORDER DOESN’T MATTER

COLUMNS ORDER DOESN’T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES (no repeating groups)

DATA INTEGRITY (?)

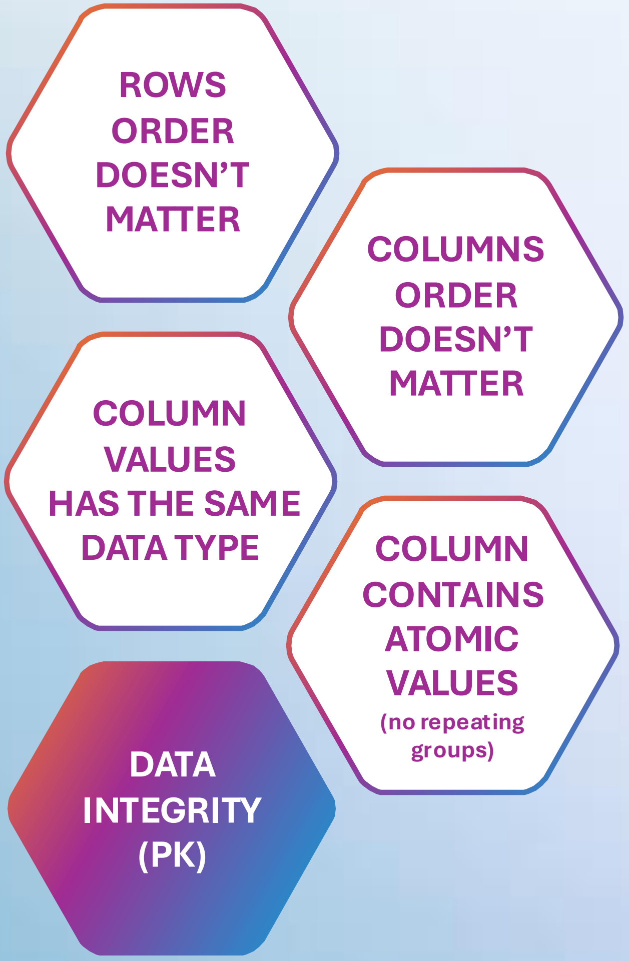
SALON	SELLER_TYPE	ADDRESS
AUDI	Official distributor	Kyiv, Demyana 56/2
AUDI	Reseller	Kyiv, Demyana 56/2
PORSHE	Authorized Dealer	Kyiv, Vynna 156
PORSHE	Certified Auto Broker	Lviv, Lopatina 90A
VOLVO	Official distributor	Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A
TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A

Does it follow 1NF?

Not really...

What would happen if we add duplicates, will the system allow it now?

1NF



ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES (no repeating groups)

DATA INTEGRITY (PK)

SALON	SELLER_TYPE	ADDRESS
AUDI	Official distributor	Kyiv, Demyana 56/2
AUDI	Reseller	Kyiv, Demyana 56/2
PORSHE	Authorized Dealer	Kyiv, Vynna 156
PORSHE	Certified Auto Broker	Lviv, Lopatina 90A
VOLVO	Official distributor	Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A
TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A

As I mentioned before, the Normalization should resolve the 'data redundancy' problem.

We need to prevent data duplications!

To do so, we have Constrains!

Additionally, we need to identify somehow each table as an entity, that will let us build relationships later…

1NF

ROWS ORDER DOESN'T MATTER

COLUMNS ORDER DOESN'T MATTER

COLUMN VALUES HAS THE SAME DATA TYPE

COLUMN CONTAINS ATOMIC VALUES (no repeating groups)

DATA INTEGRITY (PK)

composite PK

SALON	SELLER_TYPE	ADDRESS
AUDI	Official distributor	Kyiv, Demyana 56/2
AUDI	Reseller	Kyiv, Demyana 56/2
PORSHE	Authorized Dealer	Kyiv, Vynna 156
PORSHE	Certified Auto Broker	Lviv, Lopatina 90A
VOLVO	Official distributor	Kyiv, Panov str. 10
TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A
TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A

✓

What approach would definitely prevent duplications?

In this particular case!

ID PK	SALON	SELLER_TYPE	ADDRESS
1	AUDI	Official distributor	Kyiv, Demyana 56/2
2	AUDI	Reseller	Kyiv, Demyana 56/2
3	PORSHE	Authorized Dealer	Kyiv, Vynna 156
4	PORSHE	Certified Auto Broker	Lviv, Lopatina 90A
5	VOLVO	Official distributor	Kyiv, Panov str. 10
6	TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A
7	TOYOTA	Certified Auto Broker	Lviv, Lopatina 90A

First Normal Form (1NF):

Using row, column order to convey information is not permitted.

Mixing data types within the same column is not permitted.

Having Primary Key recommended.
It is not a strict requirement for 1NF.

Repeating groups are not permitted.

2NF

2 Normal Form

< Every non-prime attribute is fully functionally dependent on the entire primary key.>

CODEUS_

Insertion anomaly

composite PK	SALON	ADDRESS	RATING	BRAND_COUNTRY
	PORSHE	Kyiv, Demyana 56/2	best	Germany
	SKODA	Kyiv, Demyana 56/2	good	CR
	FIAT	Kyiv, Vynna 156	normal	Italy
	CHERY	Kyiv, Panov str. 10	bad	China

~~AUDI~~	~~NULL~~	~~NULL~~	~~Germany~~

We want to add AUDI but we don’t know its address

It is ‘Insertion anomaly’.
An insertion anomaly occurs when we cannot insert data into the database without providing other unrelated data.

Deletion anomaly

composite PK	SALON	ADDRESS	RATING	BRAND_COUNTRY
	PORSHE	Kyiv, Demyana 56/2	best	Germany
	SKODA	Kyiv, Demyana 56/2	good	CR
	FIAT	Kyiv, Vynna 156	normal	Italy
	CHERY	NULL	NULL	China

We want to delete CHERY’s address and rating but want to preserve salon and brand country

It is ‘deletion anomaly’.
An deletion anomaly occurs when we cannot remove specific customer-related data without deleting the entire customer record.

Update anomaly

composite PK	SALON	ADDRESS	RATING	BRAND_COUNTRY
	PORSHE	Kyiv, Demyana 56/2	best	Germany
	SKODA	Kyiv, Demyana 56/2	good	CR
	FIAT	Kyiv, Vynna 156	normal	Italy
	CHERY	Kyiv, Panov str. 10	bad	Germany
	CHERY	Lviv, Vasylya str. 9	bad	China

CHERY was bought by VW and now we need to change in system.

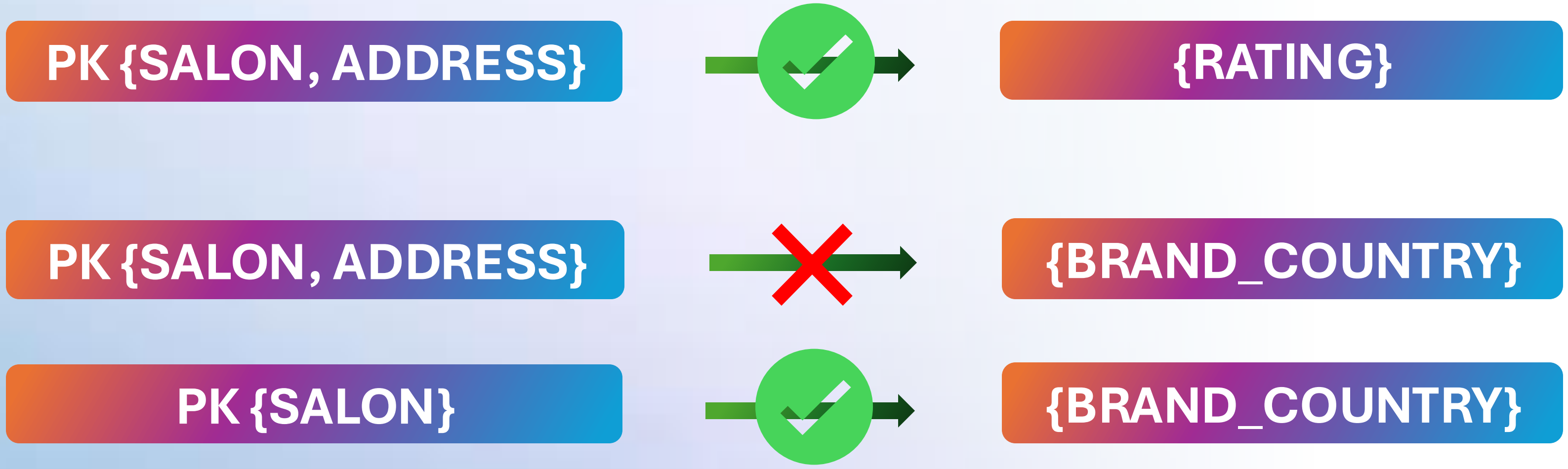
While update we chose only Kyiv and now have in system two different countries

It is ‘update anomaly’.
An update anomaly occurs when inconsistent data appears in the database because the same piece of information is duplicated across rows or columns, and not all copies are updated properly.

2NF

The 2NF says that:
- Be in 1NF
- All NON-KEY attributes must be fully functionally dependent on the ENTIRE primary key (not just part of it).

composite PK	SALON	ADDRESS	RATING	BRAND_COUNTRY
	PORSHE	Kyiv, Demyana 56/2	best	Germany
	SKODA	Kyiv, Demyana 56/2	good	CR
	FIAT	Kyiv, Vynna 156	normal	Italy
	CHERY	Kyiv, Panov str. 10	bad	China



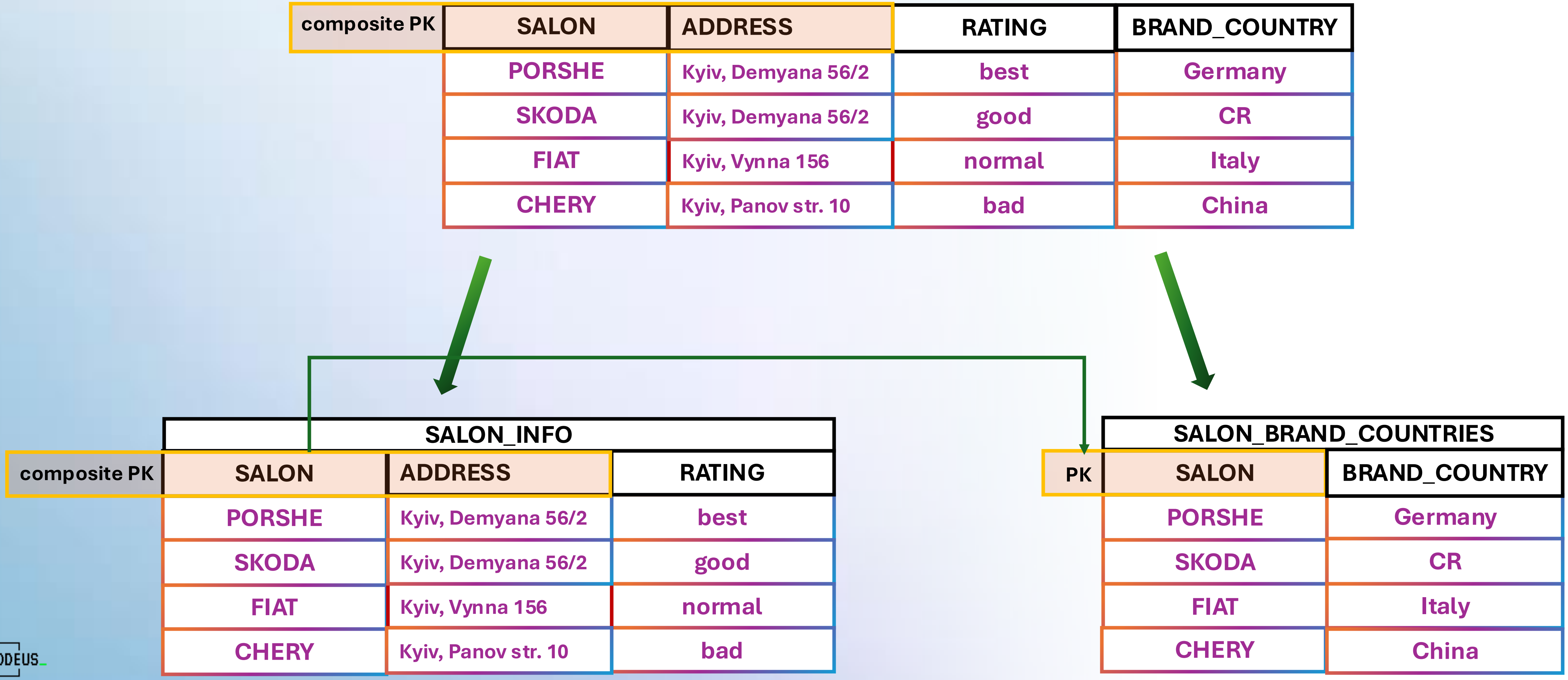
PK {SALON, ADDRESS} ✓ → {RATING}

PK {SALON, ADDRESS} ✗ → {BRAND_COUNTRY}

PK {SALON} ✓ → {BRAND_COUNTRY}

2NF

The 2NF says that:
- Be in 1NF
- All NON-KEY attributes must be fully functionally dependent on the ENTIRE primary key (not just part of it).



composite PK	SALON	ADDRESS	RATING	BRAND_COUNTRY
	PORSHE	Kyiv, Demyana 56/2	best	Germany
	SKODA	Kyiv, Demyana 56/2	good	CR
	FIAT	Kyiv, Vynna 156	normal	Italy
	CHERY	Kyiv, Panov str. 10	bad	China

SALON_INFO

composite PK	SALON	ADDRESS	RATING
	PORSHE	Kyiv, Demyana 56/2	best
	SKODA	Kyiv, Demyana 56/2	good
	FIAT	Kyiv, Vynna 156	normal
	CHERY	Kyiv, Panov str. 10	bad

SALON_BRAND_COUNTRIES

PK	SALON	BRAND_COUNTRY
	PORSHE	Germany
	SKODA	CR
	FIAT	Italy
	CHERY	China

3NF

3 Normal Form

< Every NON-KEY attribute in a table should depend on the key, the WHOLE key, and NOTHING but the key!

CODEUS_

3NF

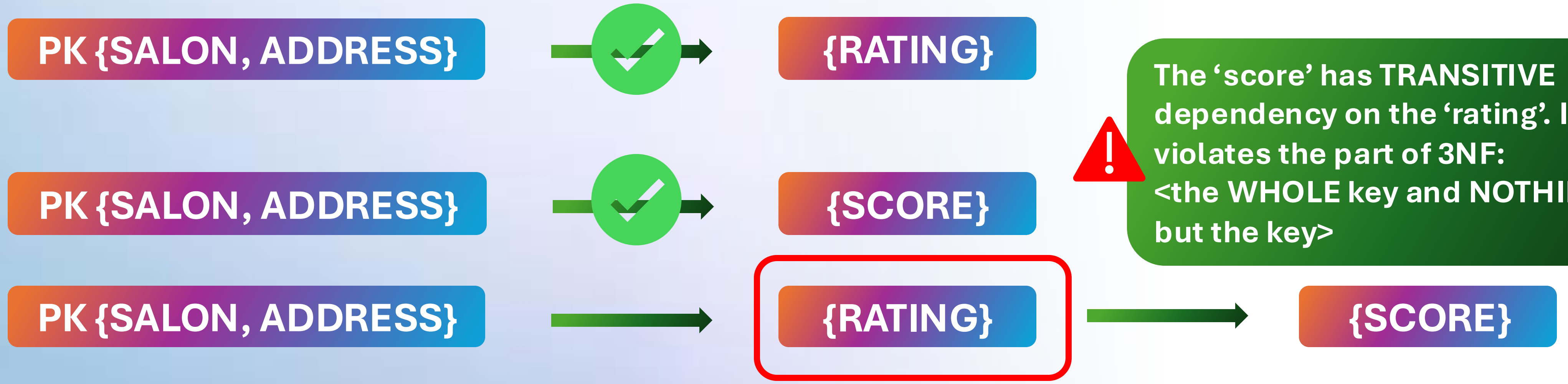
The 3NF says that:
- Be in 2NF
- Every NON-KEY attribute in a table should depend on the key, the WHOLE key, and NOTHING but the key!

SALON_INFO

composite PK

SALON	ADDRESS	RATING	SCORE
PORSHE	Kyiv, Demyana 56/2	best	9
SKODA	Kyiv, Demyana 56/2	good	7
FIAT	Kyiv, Vynna 156	normal	5
CHERY	Kyiv, Panov str. 10	bad	2

1-2: bad
3-5: normal
6-7: good
8-9: best



PK {SALON, ADDRESS} → ✓ → {RATING}

PK {SALON, ADDRESS} → ✓ → {SCORE}

PK {SALON, ADDRESS} → {RATING} → {SCORE}

The ‘score’ has TRANSITIVE dependency on the ‘rating’. It violates the part of 3NF: <the WHOLE key and NOTHING but the key>

3NF

The 3NF says that:
- Be in 2NF
- Every NON-KEY attribute in a table should depend on the key, the WHOLE key, and NOTHING but the key!

SALON_INFO

composite PK: SALON	composite PK: ADDRESS	RATING	SCORE
PORSHE	Kyiv, Demyana 56/2	best	9
SKODA	Kyiv, Demyana 56/2	good	7
FIAT	Kyiv, Vynna 156	normal	5
CHERY	Kyiv, Panov str. 10	bad	2

1-2: bad
3-5: normal
6-7: good
8-9: best

SALON_INFO

composite PK: SALON	composite PK: ADDRESS	SCORE
PORSHE	Kyiv, Demyana 56/2	9
SKODA	Kyiv, Demyana 56/2	7
FIAT	Kyiv, Vynna 156	5
CHERY	Kyiv, Panov str. 10	2

RATING

PK: SCORE	RATING
1	bad
2	bad
3	normal
...	...
6	good
7	good
8	best
9	best

Thank you

- Author: Pavlo Khshanovskyi
- My LinkedIn: https://www.linkedin.com/in/khshanovskyi/
- Date: April 2025
- Join Codeus community in Discord
- Join Codeus community in LinkedIn