

# Лекція 04: Нормалізація бази даних

Lecture 04: Database normalization

# Нормалізація / Normalization

Якщо база даних ненормалізована, вона може містити **аномалії**, які є нічним кошмаром для кожного адміністратора баз даних. Управління базою даних, що містить аномалії, майже неможливо.

If a database design is not perfect, it may contain **anomalies**, which are like a bad dream for any database administrator. Managing a database with anomalies is next to impossible.

# Аномалії оновлення / Modification anomalies

Якщо дані розкидані і не пов'язані між собою належним чином, це може призвести до дивних ситуацій. Наприклад, коли ми намагаємося оновити один елемент даних, який має копії, розкидані по декількох місцях, декілька екземплярів оновлюються правильно, тоді як декілька інших залишаються зі старими значеннями. Такі випадки залишають базу даних у неузгодженому стані.

If data items are scattered and are not linked to each other properly, then it could lead to strange situations. For example, when we try to update one data item having its copies scattered over several places, a few instances get updated properly while a few others are left with old values. Such instances leave the database in an inconsistent state.

Student ID	Student Name	Student Phone	Assigned Subject
1	A	1234	Programming
2	A	1234	Databases
3	B	2345	Biology
4	C	3456	Math
5	D	4567	Physics

Якщо телефон студента А зміниться, його необхідно оновити у всіх відповідних записах таблиці.

If student's A phone changes, must change it in each table record assigned to this student.

# Аномалії видалення / Deletion anomalies

- В результаті спроби видалити запис, деякі його частини не були видалені через незнання того, що дані також зберігаються десь в іншому місці.
- We tried to delete a record, but parts of it was left undeleted because of unawareness, the data is also saved somewhere else.

Student ID	Student Name	Student Phone	Assigned Subject
1	A	1234	Programming
2	A	1234	Databases
3	B	2345	Biology
4	C	3456	Math
5	D	4567	Physics

Якщо видалити записи про біологію з бази даних, будуть також втрачені усі дані про студента B.

If we delete Biology entries from the database, we also loose all data about student B.


# Аномалії вставки / Insert anomalies

Спроба вставити дані в запис, який взагалі не існує.

We tried to insert data in a record that does not exist at all.

Student ID	Student Name	Student Phone	Assigned Subject
1	A	1234	Programming
2	A	1234	Databases
3	B	2345	Biology
4	C	3456	Math
5	D	4567	Physics

English



В базі даних неможливо зберегти дані про новий предмет, тому що він не буде пов'язаний з жодним студентом.

It is impossible to store data about the new subject, since it won't be associated to any student.



# Перша нормальна форма / First normal form

Це правило визначає, що всі атрибути у відношенні повинні мати атомарні домени. Значення в атомарному домені є неподільними одиницями. Кожен атрибут повинен містити лише одне значення з його попередньо визначеного домену.

This rule defines that all the attributes in a relation must have atomic domains. The values in an atomic domain are indivisible units. Each attribute must contain only a single value from its pre-defined domain.

Course	Content
Programming	Java, c++
Web	HTML, PHP, ASP

Non-atomic  
attributes



Course	Content
Programming	Java
Programming	c++
Web	HTML
Web	PHP
Web	ASP

# Друга нормальна форма / Second normal form

- Для дотримання другої нормальної форми, кожен не первинний атрибут повинен повністю функціонально залежати від первинного ключа. Тобто, якщо  $X \rightarrow A$  справедливо, то не повинно бути жодної підмножини  $Y$  множини  $X$ , для якої  $Y \rightarrow A$  також справедливо.
- If we follow second normal form, then every non-prime attribute should be fully functionally dependent on prime key attribute. That is, if  $X \rightarrow A$  holds, then there should not be any proper subset  $Y$  of  $X$ , for which  $Y \rightarrow A$  also holds true.

# Функціональна залежність / Functional Dependency

Функціональна залежність (FD) – це сукупність обмежень між двома атрибутами у відношенні. Функціональна залежність говорить, що якщо два кортежі мають однакові значення для атрибутів  $A_1, A_2, \dots, A_n$ , то ці два кортежі повинні мати однакові значення для атрибутів  $B_1, B_2, \dots, B_n$ .

Functional dependency (FD) is a set of constraints between two attributes in a relation. Functional dependency says that if two tuples have same values for attributes  $A_1, A_2, \dots, A_n$ , then those two tuples must have to have same values for attributes  $B_1, B_2, \dots, B_n$ .

Функціональна залежність представлена знаком стрілки ( $\rightarrow$ ), тобто  $X \rightarrow Y$ , де  $X$  функціонально визначає  $Y$ . Атрибути лівої частини визначають значення атрибутів у правій частині.

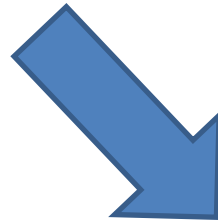
Functional dependency is represented by an arrow sign ( $\rightarrow$ ) that is,  $X \rightarrow Y$ , where  $X$  functionally determines  $Y$ . The left-hand side attributes determine the values of attributes on the right-hand side.

- **Рефлексивність.** Якщо  $A$  – це набір атрибутів і  $B \in$  підмножиною  $A$ , то  $A$  містить  $B$ .
- **Reflexive rule.** If  $A$  is a set of attributes and  $B$  is subset of  $A$ , then  $A$  holds  $B$ .

- **Доповнення.** Якщо  $A \rightarrow B$  виконується, та  $Y$  є множиною атрибутів, то  $AY \rightarrow BY$  також виконується. Це додавання атрибутів до залежностей, не змінює основних залежностей.
- **Augmentation rule.** If  $A \rightarrow B$  holds and  $Y$  is attribute set, then  $AY \rightarrow BY$  also holds. That is adding attributes in dependencies, does not change the basic dependencies.
- **Транзитивність.** Те ж, що і транзитивне правило в алгебрі, якщо виконується  $A \rightarrow B$ , і  $B \rightarrow C$ , то  $A \rightarrow C$  також має місце.  $A \rightarrow B$  функціонально визначає  $B$ .
- **Transitivity rule.** Same as transitive rule in algebra, if  $A \rightarrow B$  holds and  $B \rightarrow C$  holds, then  $A \rightarrow C$  also holds.  $A \rightarrow B$  is called as a functionally that determines  $b$ .

## Student\_Project

Stu_ID	Proj_ID	Stu_Name	Proj_Name
--------	---------	----------	-----------



## Часткова залежність Partial dependency

$A = \{Stu\_Name, Proj\_Name\}$   
 $X = \{Stu\_ID, Proj\_ID\}$

$Stu\_ID \rightarrow Stu\_Name$   
 $Proj\_ID \rightarrow Proj\_Name$

$A1 = \{Stu\_Name, Proj\_ID\}$   
 $X1 = \{Stu\_ID\}$   
 $X1 \rightarrow A1$

$A2 = \{Proj\_Name\}$   
 $X2 = \{Proj\_ID\}$   
 $X2 \rightarrow A2$

## Student

Stu_ID	Stu_Name	Proj_ID
--------	----------	---------

## Project

Proj_ID	Proj_Name
---------	-----------

# Третя нормальна форма / Third normal form

Відношення, яке має бути у третій нормальній формі, має бути у другій нормальній формі, а також має виконуватися:

- немає нетривіальних атрибутів, які транзитивно залежать від первинного ключа;
- для будь-якої нетривіальної функціональної залежності  $X \rightarrow A$ ,  $X$  має бути супер-ключем або  $A$  має бути первинним ключем.

For a relation to be in Third Normal Form, it must be in Second Normal form and the following must satisfy:

- No non-prime attribute is transitively dependent on prime key attribute.
- For any non-trivial functional dependency,  $X \rightarrow A$ , then either  $X$  is a super key or,  $A$  is prime attribute.



# Тривіальна функціональна залежність / Trivial Functional Dependency

- Якщо виконується функціональна залежність  $X \rightarrow Y$ , де  $Y$  – підмножина  $X$ , то вона називається **тривіальною**.
- If a functional dependency  $X \rightarrow Y$  holds, where  $Y$  is a subset of  $X$ , then it is called a **trivial**.
- Якщо виконується  $X \rightarrow Y$ , де  $Y$  не є підмножиною  $X$ , то функціональну залежність називають **нетривіальною**.
- If an functional dependency  $X \rightarrow Y$  holds, where  $Y$  is not a subset of  $X$ , then it is called a **non-trivial**.
- Якщо виконується  $X \rightarrow Y$ , де перетин  $X$  та  $Y = \emptyset$ , то функціональна залежність вважається **абсолютно нетривіальною**.
- If an functional dependency  $X \rightarrow Y$  holds, where  $x$  intersect  $Y = \emptyset$ , it is said to be a **completely non-trivial**.

## Student\_Detail

Stu_ID	Stu_Name	City	Zip
--------	----------	------	-----

Транзитивна залежність  
Transitive dependency

Zip -> City ???

Zip IS NOT A SUPER KEY

City IS NOT A PRIMARY KEY

Stu\_ID -> Zip -> City



## Student\_Detail

Stu_ID	Stu_Name	Zip
--------	----------	-----

Stu\_ID -> Stu\_Name, Zip

## ZipCodes

Zip	City
-----	------

Zip -> City

Super Key

# Нормальна форма Бойса-Кодда / Boyce-Codd Normal Form

Нормальна форма Бойса-Кодда (BCNF) є продовженням третьої нормальної форми на більш строгих умовах. BCNF стверджує, що для будь-якої нетривіальної функціональної залежності,  $X \rightarrow A$ ,  $X$  має бути супер-ключем.

Boyce-Codd Normal Form (BCNF) is an extension of Third Normal Form on strict terms. BCNF states that for any non-trivial functional dependency,  $X \rightarrow A$ ,  $X$  must be a super-key.