



# CHAPTER

## DATA NORMALIZATION

# DATABASE NORMALIZATION

## Structured data

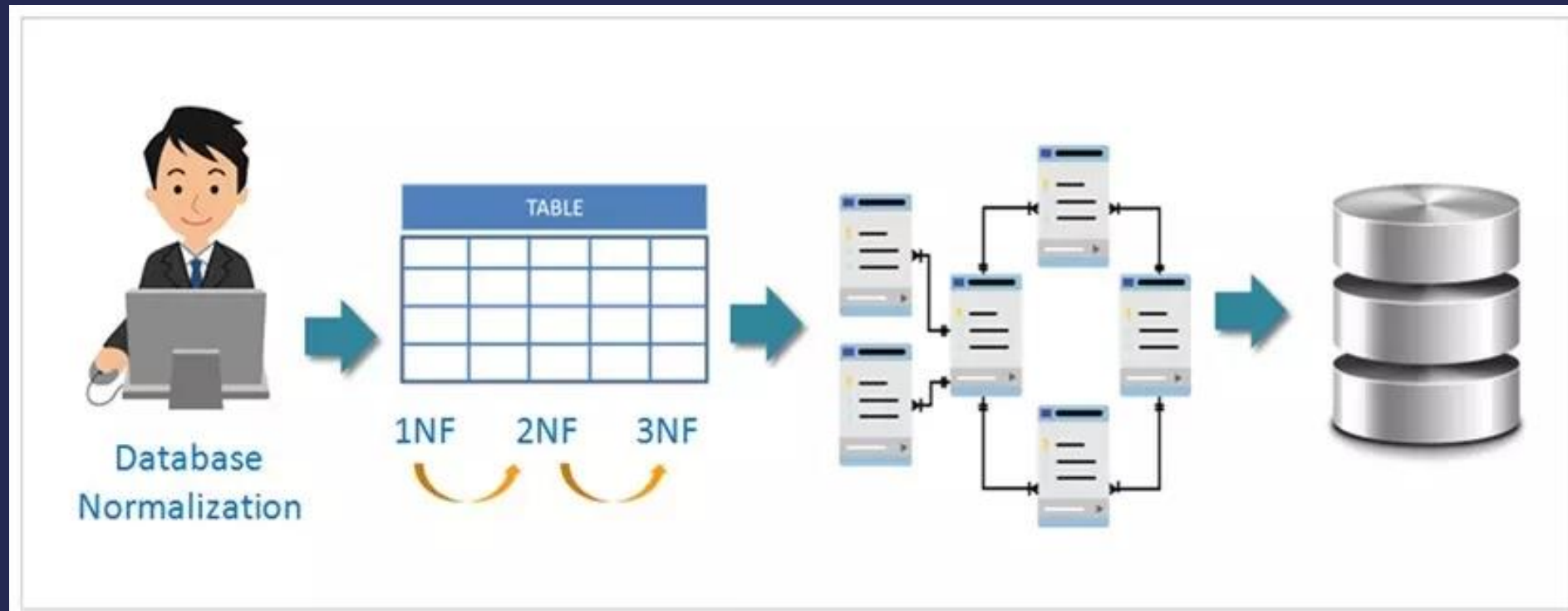
Data that is collected from a known method and can be neatly arranged

## Unstructured data

Data that has no pre-defined format or organization, making analysis more difficult

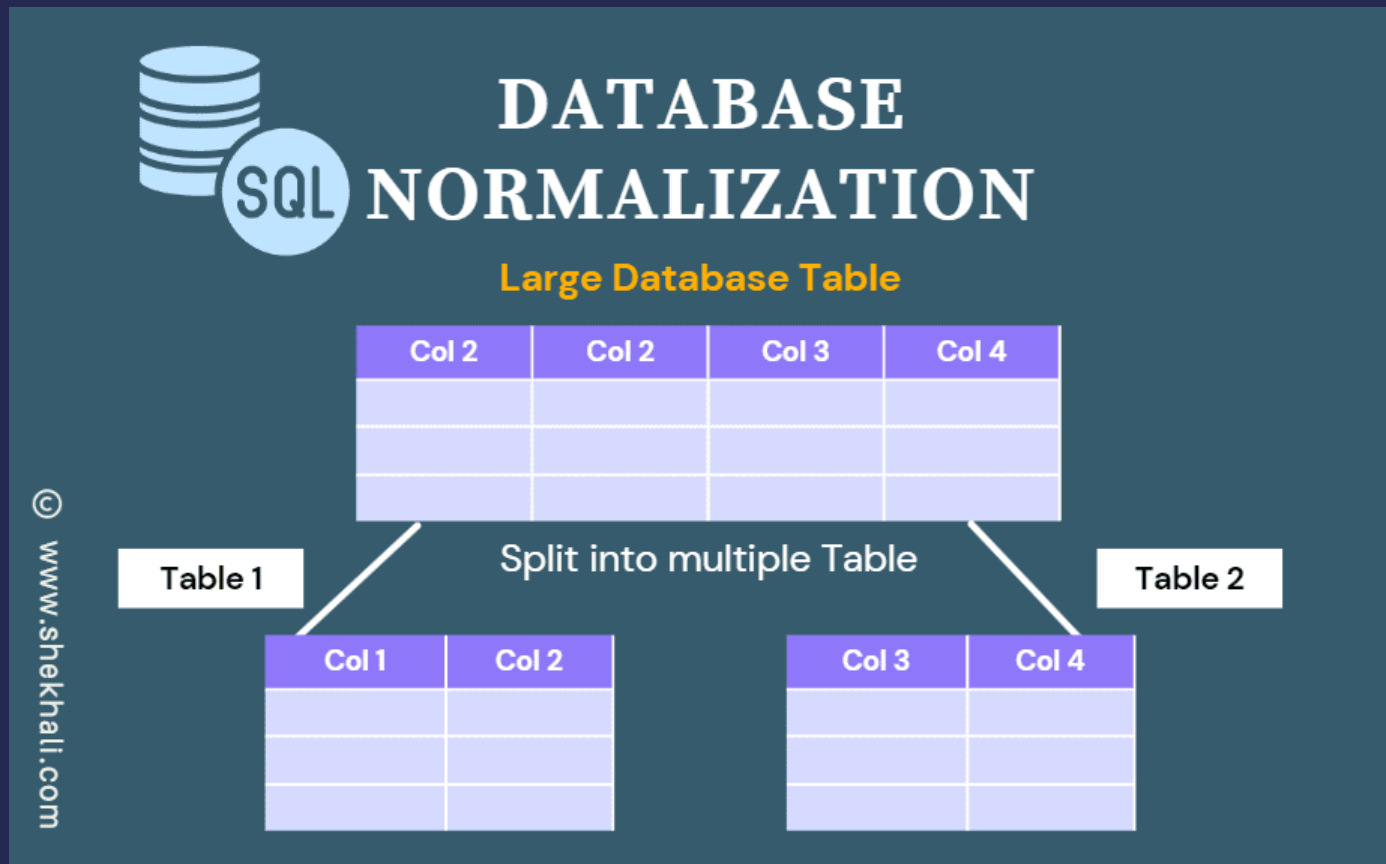
# WHAT IS NORMALIZATION IN DBMS (SQL)?

**Normalization** គឺជាបច្ចេកទេសរចនាមូលដ្ឋានទិន្នន័យដែលកាត់បន្ថយភាពស្មុកស្មាញ របស់ទិន្នន័យ និងលុបបំបាត់លក្ខណៈដែលមិនចង់បានដូចជា INSERTION UPDATE និង DELETE ដែលមិនប្រក្រតី។



# WHAT IS NORMALIZATION IN DBMS (SQL)?

**Normalization** បែងចែកតារាងធំទៅជាតារាងតូច ហើយភ្ជាប់ពួកវាដោយប្រើទំនាក់ទំនង។ គោលបំណងនៃ Normalization នៅក្នុង SQL គឺដើម្បីលុបបំបាត់ទិន្នន័យដែលលែងត្រូវការគ្នា (ដដែលៗ) និងធានាថាទិន្នន័យត្រូវបានរក្សាទុកក្នុងទម្រង់មួយត្រឹមត្រូវ។



# DATABASE NORMAL FORMS

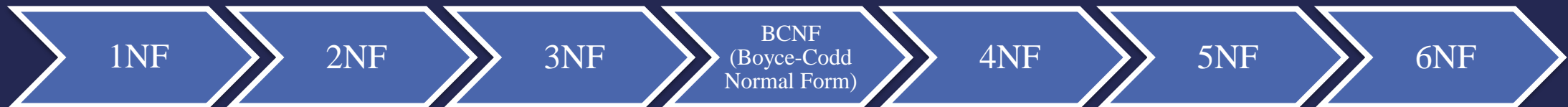
ទំរង់ Normal Forms:

- ✓ 1NF (First Normal Form)
- ✓ 2NF (Second Normal Form)
- ✓ 3NF (Third Normal Form)
- ✓ BCNF (Boyce-Codd Normal Form)
- ✓ 4NF (Fourth Normal Form)
- ✓ 5NF (Fifth Normal Form)
- ✓ 6NF (Sixth Normal Form)

# DATA NORMALIZATION

ទ្រឹស្តីនៃ Data Normalization (ការកំណត់ទិន្នន័យ) ធម្មតានៅក្នុង Database Server ត្រូវបានអភិវឌ្ឍបន្ថែមទៀត។

ឧទាហរណ៍ មានការពិភាក្សានៅលើ ទម្រង់ទី 6 (6<sup>th</sup> Normal Form) ក៏ដោយ ប៉ុន្តែនៅក្នុងការអនុវត្តជាក់ស្តែងភាគច្រើន ការធ្វើ Normalize ត្រឹមទម្រង់ទី ៣ (3<sup>th</sup> Normal Form) នៅទិន្នន័យត្រូវបានរៀបចំបានយ៉ាងល្អហើយ។





# DATABASE NORMALIZATION WITH EXAMPLES

ដើម្បីឲ្យកាន់តែងាយយល់ពី Database Normalization យើងមកមើលឧទាហរណ៍តាម Normalization Form នីមួយៗ។

ប្រសិនបើគេមាន Movies Database សម្រាប់ជួលដូចខាងក្រោម ហើយតារាងមិនទាន់បាន ធ្វើ Normalize Form នូវឡើយទេ យើងនឹងចាប់ផ្តើមជាមួយ 1NF

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3 <sup>rd</sup> Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5 <sup>th</sup> Avenue	Clash of the Titans	Mr.

Multi Value

# 1NF (FIRST NORMAL FORM)

## លក្ខខណ្ឌ 1NF:

- ✓ Each table cell should contain a single value.
- ✓ Each record needs to be unique.

## 1NF Example

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3 <sup>rd</sup> Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3 <sup>rd</sup> Street 34	Daddy's Little Girls	Mr.
Robert Phil	5 <sup>th</sup> Avenue	Clash of the Titans	Mr.



# WHAT IS A KEY IN SQL?

—មុននឹងយើងបន្តទៀតត្រូវដឹងគន្លឹះមួយចំនួន—

**KEY** ក្នុង SQL ជាតម្លៃមួយត្រូវបានប្រើសម្រាប់បញ្ជាក់ថាតម្លៃក្នុងតារាងគឺ **Unique** ។



**KEY** មួយអាចជា តម្លៃមួយក្នុងជួរឈរមួយ ឬ បង្គុំជួរឈរច្រើន បង្កើតបានជាតម្លៃ **Unique** មួយជួរដេកក្នុងតារាង។

**KEY** ត្រូវបានប្រើដើម្បីបញ្ជាក់ព័ត៌មានស្ទួនដែលមានទំនាក់ទំនងរវាងតារាងផ្សេងៗក្នុង **Database**។

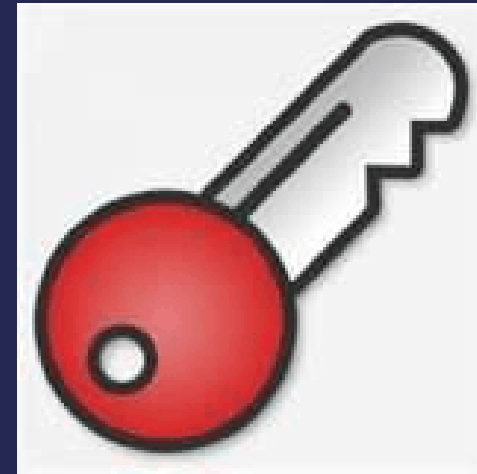
**ចំណាំ៖** ជួរឈរ (Columns) ក្នុងតារាងដែលមិនត្រូវបានប្រើដើម្បីកំណត់អត្តសញ្ញាណ ទិន្នន័យមិនស្ទួន (Unique) ត្រូវបានគេហៅថា **non-key columns**

# WHAT IS A PRIMARY KEY?

Primary Key គឺជាតម្លៃជួរឈរតែមួយ(Unique) ដែលប្រើដើម្បីកំណត់អត្តសញ្ញាណទិន្នន័យ មានមួយគត់ក្នុង Table ។

Primary Key មានលក្ខណៈដូចជា

- ✓ A primary key cannot be NULL
- ✓ A primary key value must be unique
- ✓ The primary key values should rarely be changed
- ✓ The primary key must be given a value when a new record is inserted.



Primary Key

# WHAT IS A PRIMARY KEY?

**Composite key** ជា Primary key មួយដែលកើតឡើងពីជួរឈរពីរប្រើសម្រាប់បញ្ជាក់លក្ខណៈ **Unique**.

ឧទាហរណ៍ ក្នុង Database យើងមានមនុស្សពីរនាក់មានឈ្មោះដូចគ្នា ប៉ុន្តែរស់នៅកន្លែងផ្សេងគ្នា

**Composite Key**

Robert Phil	3 <sup>rd</sup> Street 34	Daddy's Little Girls	Mr.
Robert Phil	5 <sup>th</sup> Avenue	Clash of the Titans	Mr.

*Names are common. Hence you need name as well Address to uniquely identify a record.*

ដូចនេះ យើងអាចប្រើ Full Name និង Address ដើម្បីបញ្ជាក់ថាជួរដេកនោះ Unique តើហៅថា **Composite Key**.

# 2NF (SECOND NORMAL FORM)

## លក្ខខណ្ឌ 2NF:

- Rule 1- Be in 1NF
- Rule 2- Single Column Primary Key that does not functionally dependant on any subset of candidate key relation

វាច្បាស់ណាស់ថាឃើងមិនអាចបង្កើតមូលដ្ឋានទិន្នន័យ (Database) របស់យើងក្នុងទម្រង់  
2nd Normalization លុះត្រាតែឃើងបែងចែកតារាងខាងលើ

## 2NF (SECOND NORMAL FORM)

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3 <sup>rd</sup> Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3 <sup>rd</sup> Street 34	Daddy's Little Girls	Mr.
Robert Phil	5 <sup>th</sup> Avenue	Clash of the Titans	Mr.

តារាងខាងលើក្នុងទម្រង់ 1NF យើងនឹងបំបែកជាតារាងពីរផ្សេងគ្នា ដើម្បីឱ្យតារាងការជួល  
Movies ក្នុងទម្រង់ 2NF

## 2NF (SECOND NORMAL FORM)

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 <sup>rd</sup> Street 34	Mr.
3	Robert Phil	5 <sup>th</sup> Avenue	Mr.

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

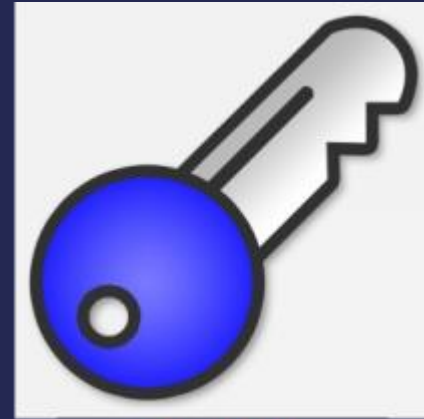
យើងបានបង្កើត Column ថ្មីមួយ Membership\_id ដែលជា Primary Key សម្រាប់តារាងទី១ ដើម្បីជៀសវាងការងាយស្រួលក្នុងការកំណត់ត្រាដូចគ្នា។ មានលក្ខណៈ Unique ដោយប្រើ Membership\_ID



# DATABASE – FOREIGN KEY

តារាងទី២ Membership\_ID គឺជា **Foreign Key**


MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans



Foreign Key


- **Foreign Key** references ទៅកាន់ Primary Key នៃតារាងមួយទៀត វាជួយធ្វើការភ្ជាប់រវាងតារាងផ្សេងៗគ្នា
- **Foreign Key** អាចមានឈ្មោះផ្សេងពី **Primary Key** របស់វា។ វាធានាថាជួរដេកក្នុងតារាងមួយមានជួរដែលត្រូវគ្នាក្នុងតារាងមួយទៀត មិនដូច **Primary Key** ទេ ពួកវាមិនចាំបាច់មានតែមួយ

# DATABASE – FOREIGN KEY

 **Foreign Key**

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

**Foreign Key references Primary Key**  
**Foreign Key can only have values present in primary key**  
**It could have a name other than that of Primary Key**

 **Primary Key**

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 <sup>rd</sup> Street 34	Mr.
3	Robert Phil	5 <sup>th</sup> Avenue	Mr.

# DATABASE – FOREIGN KEY

ហេតុអ្វីចាំបាច់ត្រូវការ Foreign Key ?

ប្រសិន គេបញ្ចូលទិន្នន័យថ្មីក្នុងតារាង Movie

Insert a record in Table 2 where Member ID =101

MEMBERSHIP ID	MOVIES RENTED
101	Mission Impossible

But Membership ID 101 is not present in Table 1

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 <sup>rd</sup> Street 34	Mr.
3	Robert Phil	5 <sup>th</sup> Avenue	Mr.

Database will throw an **ERROR**. This helps in referential integrity

## DATABASE — FOREIGN KEY

អ្នកនឹងអាចបញ្ចូលតម្លៃទៅតារាងមាន **Foreign Key** របស់អ្នក ប្រសិនបើមានតម្លៃក្នុងតារាង មាន **Primary Key** ។ នេះជួយក្នុងភាពត្រឹមត្រូវក្នុងតារាង។

បញ្ហាខាងលើអាចដោះស្រាយបានដោយប្រកាស `membership_id` នៃតារាងទី២ ជា **Foreign Key** នៃ `membership_id` ពីតារាងទី១

ឥឡូវនេះ ប្រសិនបើនរណាម្នាក់ព្យាយាមបញ្ចូលតម្លៃនៅក្នុង Membership ID ដែលមិនមាន នៅក្នុងតារាងមេ នោះកំហុសនឹងត្រូវបានបង្ហាញ!

## TRANSITIVE FUNCTIONAL DEPENDENCIES

**Transitive functional dependency** គឺនៅពេលផ្លាស់ប្តូរជួរឈរដែលមិនមែនជា KEY (non-key column) អាចបណ្តាលឱ្យមានការផ្លាស់ប្តូរជួរឈរដែលមិនមែនជា Key (non-key column)ផ្សេងទៀត

ពិចារណាតារាង 1. ការផ្លាស់ប្តូរជួរឈរដែលមិនមែនជាKey ឈ្មោះពេញ(Full Names)អាចផ្លាស់ប្តូរ Satutation

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 <sup>rd</sup> Street 34	Mr.
3	Robert Phil	5 <sup>th</sup> Avenue	Mr.

*Change in Name* → *May Change Salutation*

# 3NF (THIRD NORMAL FORM)

## 3NF Rules:

- Rule 1- Be in 2NF
- Rule 2- Has no transitive functional dependencies

ដើម្បីបម្លែងតារាង 2NF ទៅជា 3NF យើងត្រូវបែងចែកតារាងរបស់យើងម្តងទៀត។



# 3NF (THIRD NORMAL FORM)

## Example

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3 <sup>rd</sup> Street 34	1
3	Robert Phil	5 <sup>th</sup> Avenue	1

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

SALUTATION ID	SALUTATION
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

# 3NF (THIRD NORMAL FORM)

យើងបានបំបែកតារាង និងបង្កើតតារាងថ្មីដើម្បីផ្ដុំ Salutations.

ដូននេះមិនមាន Transitive functional dependencies ហើយតារាងស្ថិតក្នុងទម្រង់ 3NF

ក្នុងតារាងទី៣ មាន Salutation\_ID ជា Primary Key ហើយតារាងទី១ មាន Salutation\_ID គឺជា Foreign Key និងជា Primary key ក្នុងតារាងទី៣

ឥឡូវនេះឧទាហរណ៍របស់យើងគឺនៅកម្រិតមួយដែលមិនអាចបំបែកបានបន្ថែមទៀតដើម្បីទទួលបានប្រភេទ Normal Form ខ្ពស់ជាងធម្មតានៅក្នុង DBMS ។ តាមពិតទៅ វាមានទម្រង់ធម្មតាខ្ពស់ជាងហើយ។ កិច្ចខិតខំប្រឹងប្រែងដាច់ដោយឡែកសម្រាប់ការផ្លាស់ប្តូរទៅកម្រិតបន្ទាប់នៃការធ្វើឱ្យទិន្នន័យមានលក្ខណៈធម្មតាជាធម្មតាត្រូវការនៅក្នុងមូលដ្ឋានទិន្នន័យសុគតស្មាញ។

## MORE NORMALIZATION FORMS

### BCNF (Boyce-Codd Normal Form)

ទោះបីតារាងស្ថិតនៅក្នុងទម្រង់ទី 3 (3NF) ក៏នៅតែមានលទ្ធផលមិនប្រក្រតី ប្រសិនបើវាមានកូនសោបេក្ខជនច្រើនជាងមួយ។

ពេលខ្លះ BCNF ត្រូវបានគេសំដៅផងដែរថាជា 3.5 Normal Form ។

### 4NF (Fourth Normal Form) Rules:

ប្រសិនបើតារាងនៃ Database មិនមានទិន្នន័យដែលមានមិនឯករាជ្យ (independent) និងតម្លៃច្រើន (multivalued) នោះវាស្ថិតនៅក្នុង 4th Normal Form ។

# MORE NORMALIZATION FORMS

## 5NF (Fifth Normal Form) Rules

តារាងមួយក្នុង 5th Normal Form ប្រសិនបើវាស្ថិតក្នុង 4NF ហើយវាមិនអាចបំបែកជាតារាងតូចបានដោយមិនមានការបាត់បង់ទិន្នន័យ។

## 6NF (Sixth Normal Form) Proposed

6th Normal Form មិនទាន់ជាស្តង់ដារឡើយទេ ទោះបីយ៉ាងណាដោយ មានការជំរុញគ្នាជាច្រើនដោយអ្នកជំនាញ Database ។ សង្ឃឹមថាឃើងនឹងមាននិយមន័យនិងស្តង់ដារមួយសម្រាប់ 6th Normal Form ក្នុងពេលអនាគតជំនួញខាងមុខនេះ

# EXAMPLE

## Problems Without Normalization

rollno	name	branch	hod	office_tel
401	Akon	CSE	Mr. X	53337
402	Bkon	CSE	Mr. X	53337
403	Ckon	CSE	Mr. X	53337
404	Dkon	CSE	Mr. X	53337

# EXAMPLE

## Time for an Example

Although all the rules are self explanatory still let's take an example where we will create a table to store student data which will have student's roll no., their name and the name of subjects they have opted for.

Here is our table, with some sample data added to it.

roll_no	name	subject
101	Akon	OS, CN
103	Ckon	Java
102	Bkon	C, C++



# EXAMPLE

## How to solve this Problem?

It's very simple, because all we have to do is break the values into atomic values.

Here is our updated table and it now satisfies the First Normal Form.

<b>roll_no</b>	<b>name</b>	<b>subject</b>
101	Akon	OS
101	Akon	CN
103	Ckon	Java
102	Bkon	C
102	Bkon	C++

# EXAMPLE

So let's use the same example, where we have 3 tables, **Student**, **Subject** and **Score**.

## Student Table

<b>student_id</b>	<b>name</b>	<b>reg_no</b>	<b>branch</b>	<b>address</b>
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat
12	Bkon	09-WY	IT	Rajasthan

## Subject Table

<b>subject_id</b>	<b>subject_name</b>	<b>teacher</b>
1	Java	Java Teacher
2	C++	C++ Teacher
3	Php	Php Teacher

# EXAMPLE

Score Table

score_id	student_id	subject_id	marks
1	10	1	70
2	10	2	75
3	11	1	80