Normalization and Denormalization

CSci 150 – Fundamentals of Database Systems

Rodney M. Maniego Jr.

Instructor I / DCST / VSU

Version: 20231030

COURSE OUTCOME

Normalization of databases in DBMS

LEARNING OUTCOME

- Identify and explain the concepts about database normalization.
- Apply and demonstrate database normalization.

TOPICS

1. Anomalies
2. Functional Dependency
3. Normalization Theory
4. Denormalization

Anomalies

Bad Database Design

- Waste of storage
- Waste of other resources
- Anomalies

What are Anomalies?

It refer to unexpected and undesired behaviors that occurs during database operations.

Poorly designed databases lead to data processing anomalies, especially when relationships and dependencies are not strictly considered.

Data Quality Issues

- Require a series of additional operations to ensure that data aligns with the irregular database structure.
- Consistency and integrity are compromised when one or more values are either left inaccurate or accidentally excluded during these operations.

Insertion Anomaly

- An anomaly in which an unintended information is included during the insertion operation.

Employee_ID	Full_Name	Department	Location	Date_Hired
1	Juan dela Cruz	IT	22nd Floor	2014-09-11
2	Maria Sta Ana	Engineering	Basement	2015-06-29
3	Pedro Malaya	Engineering	Basement	2015-09-11
4	Juan Rosario	HR	14th Floor	2016-12-08
5	Mateo Abante	Web	7th Floor	2018-03-22
6	Juan dela Cruz	Logistics	17th Floor	2019-01-03
7	Elias Pinaglabanan	HR	9th Floor	2019-08-17

Update Anomaly

- An anomaly in which the some content should be updated repeatedly when modifying information.

Employee_ID	Full_Name	Department	Location	Date_Hired
1	Juan dela Cruz	IT	22nd Floor	2014-09-11
2	Maria Sta Ana	Engineering	Basement	2015-06-29
3	Pedro Malaya	Engineering	Basement	2015-09-11
4	Juan Rosario	HR	14th Floor	2016-12-08
5	Mateo Abante	Web	7th Floor	2018-03-22
6	Juan dela Cruz	Logistics	17th Floor	2019-01-03
7	Elias Pinaglabanan	HR	9th Floor	2019-08-17

Deletion Anomaly

❑ An anomaly in which the necessary information is lost when deleting some information.

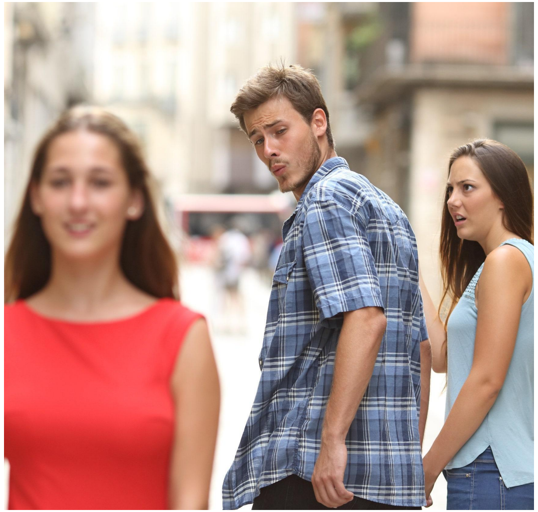
Employee_ID	Full_Name	Department	Location	Date_Hired
1	Juan dela Cruz	IT	22nd Floor	2014-09-11
2	Maria Sta Ana	Engineering	Basement	2015-06-29
3	Pedro Malaya	Engineering	Basement	2015-09-11
4	Juan Rosario	HR	14th Floor	2016-12-08
5	Mateo Abante	Web	7th Floor	2018-03-22
6	Juan dela Cruz	Logistics	17th Floor	2019-01-03
7	Elias Pinaglabanan	HR	9th Floor	2019-08-17

Example

Employee(Employee_ID, Project_ID, Employee_Name, Project_Name, Hours_of_Work)

- Insert Anomalies:
  - Cannot insert an Employee without assigning to a Project.
  - Cannot insert a Project without an assigned Employee.
- Update Anomaly:
  - Updating a Project Name, requires updating all other employees working in the same project.
- Delete Anomalies:
  - Deleting a Project, requires the deletion of all employees working on it.
  - Deleting an employee working solely on a project, will also delete the project information.

Functional Dependency



Distracted Boyfriend Meme

Functional Dependency

What is Dependency?

It is describes the general concept of relationships between the attributes of a relation.

R(a, b, c, …, z)

Relation **R** having the attributes <a, b, c, …, z>

What is Dependency?

$$\underset{\text{Determinant}}{X} \longrightarrow \underset{\text{Dependent}}{Y}$$

Definition: There exist subsets X and Y of the attributes of relation R such that, for any two tuples in R with the same values for the attributes in X, the corresponding values for the attributes in Y are always the same.

Huh?

What is Dependency?

$$\underset{\text{Determinant}}{X} \rightarrow \underset{\text{Dependent}}{Y}$$

- Loosely, if you have two tuples sharing the same values for the attributes in X, then those two tuples will also have the same values for the attributes in Y.

What is Dependency?

$$X \rightarrow Y$$

Determinant Dependent

Can be read as:

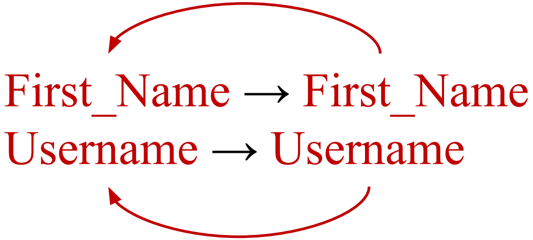
- **X** determines **Y**.
- Determinant **X** determines dependent **Y**.
- Dependent **Y** is dependent on determinant **X**.

What is Dependency?

Full_Name	Birth_Date
Juan dela Cruz	2000-09-11
Maria Sta Ana	1999-06-29
Pedro Malaya	2000-09-11
Juan Rosario	2001-12-08
Mateo Abante	2000-03-22
Juan dela Cruz	2000-09-11
Elias Pinaglabanan	2002-08-17

Trivial Dependency

- It is a self-evident relationship where the presence or absence of one attribute implies the same for another, making it uninteresting.



First_Name → First_Name

Username → Username

* Basically, the first thing you shouldn’t do when considering dependencies.

Non-Trivial Dependency

- ❑ Implies meaningful information without specifying a particular type of dependency.

Username → {First_Name, Last_Name}
Product_ID → {Name, Description}

What is Functional Dependency?

It is describes the focused relationships between the attributes of a relation, serving as a constraint that specify how one attribute is determined by another attribute.

What is Functional Dependency?

$$\underset{\text{Determinant}}{X} \rightarrow \underset{\text{Dependent}}{Y}$$

Definition: There are subsets X and Y of the attributes of relation R. If the values of the arbitrary attributes of X are the same, then the values of the attributes in Y are always the same.

Huh?

What is Functional Dependency?

$$X \rightarrow Y$$

Determinant Dependent

❑ Simply, if you know X, you can uniquely determine Y.

What is Functional Dependency?

First_Name	Last_Name	Username	Birth_Date	Role
Juan	Dela Cruz	OneDC	2000-09-11	0
Maria	Sta. Ana	lilLamb	1999-06-29	0
Pedro	Malaya	peter20	2000-09-11	1
Juan	Rosario	JanRoss	2001-12-08	0
Mateo	Abante	10math01	2000-03-22	1
Juan	Dela Cruz	WanAndOnly	2000-09-11	0
Elias	Pinaglabanan	elyuzzz	2002-08-17	0

Birth_Date → {First_Name, Last_Name}
(Invalid Functional Dependency)

What is Functional Dependency?

First_Name	Last_Name	Username	Birth_Date	Role
Juan	Dela Cruz	OneDC	2000-09-11	0
Maria	Sta. Ana	lilLamb	1999-06-29	0
Pedro	Malaya	peter20	2000-09-11	1
Juan	Rosario	JanRoss	2001-12-08	0
Mateo	Abante	10math01	2000-03-22	1
Juan	Dela Cruz	WanAndOnly	2000-09-11	0
Elias	Pinaglabanan	elyuzzz	2002-08-17	0

First_Name → Last_Name
(Invalid Functional Dependency)

What is Functional Dependency?

First_Name	Last_Name	Username	Birth_Date	Role
Juan	Dela Cruz	OneDC	2000-09-11	0
Maria	Sta. Ana	lilLamb	1999-06-29	0
Pedro	Malaya	peter20	2000-09-11	1
Juan	Rosario	JanRoss	2001-12-08	0
Mateo	Abante	10math01	2000-03-22	1
Juan	Dela Cruz	WanAndOnly	2000-09-11	0
Elias	Pinaglabanan	elyuzzz	2002-08-17	0

{First_Name, Last_Name} → {Username, Birth_Date, Role}
(Invalid Functional Dependency)

What is Functional Dependency?

First_Name	Last_Name	Username	Birth_Date	Role
Juan	Dela Cruz	OneDC	2000-09-11	0
Maria	Sta. Ana	lilLamb	1999-06-29	0
Pedro	Malaya	peter20	2000-09-11	1
Juan	Rosario	JanRoss	2001-12-08	0
Mateo	Abante	10math01	2000-03-22	1
Juan	Dela Cruz	WanAndOnly	2000-09-11	0
Elias	Pinaglabanan	elyuzzz	2002-08-17	0

Username → {First_Name, Last_Name, Birth_Date, Role}
(Valid Functional Dependency)

* Choose the minimal set of attributes that can uniquely determine all the other attributes of the relation.

What is Functional Dependency?

Account_ID	First_Name	Last_Name	Username	Birth_Date	Role
1	Juan	Dela Cruz	OneDC	2000-09-11	0
2	Maria	Sta. Ana	lilLamb	1999-06-29	0
3	Pedro	Malaya	peter20	2000-09-11	1
4	Juan	Rosario	JanRoss	2001-12-08	0
5	Mateo	Abante	10math01	2000-03-22	1
6	Juan	Dela Cruz	WanAndOnly	2000-09-11	0
7	Elias	Pinaglabanan	elyuzzz	2002-08-17	0

Account_ID → {First_Name, Last_Name, Username, Birth_Date, Role}
(Valid Functional Dependency)

* On modern systems, a surrogate key (alphanumeric, autoincremented) which has no intrinsic meaning outside the database system is typically used as the primary key instead of natural keys (TIN, Government IDs) for simplicity, maintainability, and to prevent potential issues with real-world identifiers.

Multi-Valued Dependency

- ❑ A specific case of a non-trivial functional dependency where a set of attributes uniquely determines a possibly multiple set of values.

A ↠ {B,C}, but not B→C.

Employee_ID ↠ {Skills, Projects}

{Author, Title} ↠ {Edition, Publisher, Date_Published} wuuut?

Multi-Valued Dependency

Employee_ID ↠ {Skills, Projects}, but not Skills→Projects.

Employee_ID	Skills	Projects
1	Python, Java	AAA-001, AAA-002, PPP-001
2	C, C++, Java	BBB-001, AAA-002
3	C#, VB.NET	VVV-001, VVV-0002
4	PHP, CSS, JavaScript	PPP-001, PPP-002, PPP-003, PPP-004
5	Python, R	AAA-003, RRR-001
6	Java, C, C++	AAA-002, BBB-001
7	MS Office	MMM-001

Multi-Valued Dependency

Employee_ID ↠ {Skills, Projects}, but not Skills→Projects.

Employee_ID	Skill	Project_ID
2	C	BBB-001
2	C	AAA-002
2	C++	BBB-001
2	Java	AAA-002

Multi-Valued Dependency

$\{\text{Author, Title}\} \twoheadrightarrow \{\text{ISBN10, ISBN13, Edition, Date\_Published}\}$

Author	Title	Edition	Publisher_ID	Date_Published
Juan dela Cruz	An Autobiography	1	32	2013-09-11
Maria Sta Ana	The Raven and Cat	1	512	2017-06-29
Pedro Malaya	Seven Lucky Sisters	1	111	2020-09-11
Juan Rosario	Ako, Ikaw, at Sila	1	76	2021-12-08
Mateo Abante	Bayanihan	1	15	2022-03-22
Juan dela Cruz	An Autobiography	2	156	2023-01-03
Elias Pinaglabanan	Purple Stars	1	44	2023-08-17

Partial Dependency

- ❑ Occurs when an attribute in a relation is functionally dependent on only a subset of a candidate key rather than the entire key.

Order(Order_ID, Customer_ID, Product_ID, Shipping_Address)

- ○ {Order_ID, Customer_ID, Product_ID} → Shipping_Address
- ○ Order_ID → Shipping_Address

Transitive Dependency

- ❑ Occurs when an attribute is indirectly dependent on another attribute through a third attribute.
- ❑ If X → Y and Y → Z, then, by transitivity, you can infer X → Z.

Example

- ○ Employee_ID → Department_ID and Department_ID → Manager_ID
- ○ Then there is a transitive functional dependency that implies Employee_ID → Manager_ID

Transitive FD

Employee_ID → Department_ID and Department_ID → Manager_ID

Employee_ID	Full_Name	Department_ID	Manager_ID	Date_Hired
1	Juan dela Cruz	2	4	2014-09-11
2	Maria Sta Ana	5	7	2015-06-29
3	Pedro Malaya	7	4	2015-09-11
4	Juan Rosario	3	5	2016-12-08
5	Mateo Abante	4	6	2018-03-22
6	Juan dela Cruz	1	3	2019-01-03
7	Elias Pinaglabanan	4	6	2019-08-17

Normalization

What is Normalization?

❑ It refers to the process of organizing and structuring the data in a relational database to reduce data redundancy and improve data integrity.

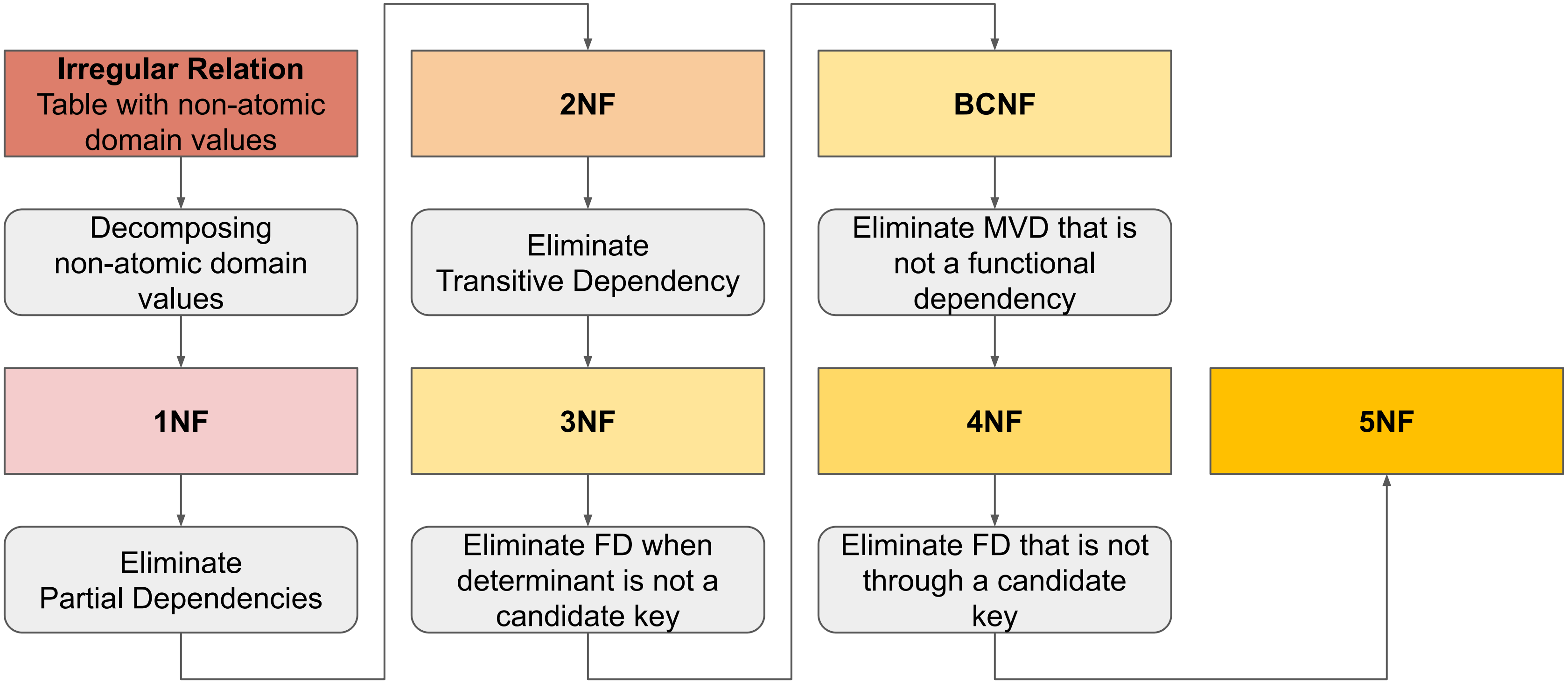
Involves dividing large relations into smaller, more manageable relations and establishing relationships between them. It can benefit the long-term performance and cost of maintaining large databases.

Note: Surrogate keys are not the main focus in normalization as long as the natural keys and the candidate keys can still determine the non-prime keys, but introduction of surrogate keys as the primary key (PK) can simplify relationships and enhance performance, maintainability, and adaptability.

Levels of Normalization

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

Levels of Normalization



Irregular Relation
Table with non-atomic domain values

Decomposing non-atomic domain values

1NF

Eliminate Partial Dependencies

2NF

Eliminate Transitive Dependency

3NF

Eliminate FD when determinant is not a candidate key

BCNF

Eliminate MVD that is not a functional dependency

4NF

Eliminate FD that is not through a candidate key

5NF

First Normal Form – 1NF

- Ensures that each column in a table only contains atomic domain values.

The goal is to decompose columns with related values but are not directly accessible.

Employee_ID	Full_Name	Skills
1	Juan dela Cruz	Python, Java
2	Maria Sta Ana	C, C++, Java
3	Pedro Malaya	C#, VB.NET
4	Juan Rosario	PHP, CSS, JavaScript
5	Mateo Abante	Python, R
6	Juan dela Cruz	Java, C, C++
7	Elias Pinaglabanan	MS Office

First Normal Form – 1NF

Employee_ID	Full_Name	Skills
1	Juan dela Cruz	Python, Java
2	Maria Sta Ana	C, C++, Java
3	Pedro Malaya	C#, VB.NET
4	Juan Rosario	PHP, CSS, JavaScript
5	Mateo Abante	Python, R
6	Juan dela Cruz	Java, C, C++
7	Elias Pinaglabanan	MS Office

Employee_ID	Full_name	Skill
1	Juan dela Cruz	Python
1	Juan dela Cruz	Java
2	Maria Sta Ana	C
2	Maria Sta Ana	C++
2	Maria Sta Ana	Java
3	Pedro Malaya	C#
3	Pedro Malaya	VB.NET
4	Juan Rosario	PHP
4	Juan Rosario	CSS
4	Juan Rosario	JavaScript
5	Mateo Abante	Python
5	Mateo Abante	R
6	Juan dela Cruz	Java
6	Juan dela Cruz	C
6	Juan dela Cruz	C++
7	Elias Pinaglabanan	MS Office

$\{\text{Employee\_ID}, \text{Skill}\} \rightarrow \text{Full\_Name}$

Second Normal Form – 2NF

- From 1NF, all non-prime attributes are fully functionally dependent on the primary key.
- 2NF requires:
  - Identify the primary key.
  - Identify the relationships of the primary key and the non-prime attributes.

Eliminate partial dependencies: A → {B,C,D}, but not B→C, B→D, or C→D.

Employee_ID	Full_Name	Department_ID	Project_ID
1	Juan dela Cruz	2	AAA-001
2	Maria Sta Ana	5	BBB-001
3	Pedro Malaya	2	VVV-001
4	Juan Rosario	3	PPP-001

Second Normal Form – 2NF

Employee_ID → {Department_ID, Project_ID}, but not Department_ID → Project_ID

Employee_ID	Full_Name	Department_ID	Project_ID
1	Juan dela Cruz	2	AAA-001
2	Maria Sta Ana	5	BBB-001
3	Pedro Malaya	7	VVV-001
4	Juan Rosario	3	PPP-001

↓

Employee_ID	Full_Name
1	Juan dela Cruz
2	Maria Sta Ana
3	Pedro Malaya
4	Juan Rosario

+

Employee_ID	Department_ID
1	2
2	5
3	7
4	3

+

Employee_ID	Project_ID
1	AAA-001
2	BBB-001
3	VVV-001
4	PPP-001

Third Normal Form – 3NF

- From 2NF, ensures no prime attribute depends on other non-prime attribute.

Eliminate transitive dependencies: A→B and B→C, implies A→C.

Employee_ID	Full_Name	Department_ID	Department_Name
1	Juan dela Cruz	2	IT
2	Maria Sta Ana	5	Engineering
3	Pedro Malaya	7	Logistics
4	Juan Rosario	3	HR

Third Normal Form – 3NF

Employee_ID→Department_ID and Department_ID→Department_Name, implies Employee_ID→Department_Name

Employee_ID	Full_Name	Department_ID	Department_Name
1	Juan dela Cruz	2	IT
2	Maria Sta Ana	5	Engineering
3	Pedro Malaya	7	Logistics
4	Juan Rosario	3	HR

⇩

Employee_ID	Full_Name
1	Juan dela Cruz
2	Maria Sta Ana
3	Pedro Malaya
4	Juan Rosario

+

Department_ID	Department_Name
2	IT
5	Engineering
7	Logistics
3	HR

+

Employee_ID	Department_ID
1	2
2	5
3	2
4	3

Boyce-Codd Normal Form – BCNF

- From 3NF, ensures that for every non-trivial functional dependency involves a superkey.

Eliminate FD if determinant is not a candidate key: $\{A,B\} \rightarrow C$, but not $B \rightarrow C$.

Employee_ID	Full_Name	Project_ID	Hours_Worked
1	Juan dela Cruz	AAA-001	5
2	Maria Sta Ana	BBB-001	3
3	Pedro Malaya	VVV-001	4
4	Juan Rosario	PPP-001	5
3	Mateo Abante	BBB-001	1

Boyce-Codd Normal Form – BCNF

{Employee_ID, Project_ID} → Hours_Worked, but not Project_ID → Hours_Worked.

Employee_ID	Full_Name	Project_ID	Hours_Worked
1	Juan dela Cruz	AAA-001	5
2	Maria Sta Ana	BBB-001	3
3	Pedro Malaya	VVV-001	4
4	Juan Rosario	PPP-001	5

↓

Employee_ID	Full_Name
1	Juan dela Cruz
2	Maria Sta Ana
3	Pedro Malaya
4	Juan Rosario

+

Employee_ID	Project_ID	Hours_Worked
1	AAA-001	5
2	BBB-001	3
3	VVV-001	4
4	PPP-001	5
3	BBB-001	1

Fourth Normal Form – 4NF

- ❑ From 3NF, ensures that there are no non-trivial multivalued dependencies from candidate keys.

Eliminate MVD: A↠B.

Fourth Normal Form – 4NF

Employee_ID	Full_name	Skill
1	Juan dela Cruz	Python
1	Juan dela Cruz	Java
2	Maria Sta Ana	C
2	Maria Sta Ana	C++
2	Maria Sta Ana	Java
3	Pedro Malaya	C#
3	Pedro Malaya	VB.NET
4	Juan Rosario	PHP
4	Juan Rosario	CSS
4	Juan Rosario	JavaScript
5	Mateo Abante	Python
5	Mateo Abante	R
6	Juan dela Cruz	Java
6	Juan dela Cruz	C
6	Juan dela Cruz	C++
7	Elias Pinaglabanan	MS Office

Employee_ID	Full_Name
1	Juan dela Cruz
2	Maria Sta Ana
3	Pedro Malaya
4	Juan Rosario
5	Mateo Abante
6	Juan dela Cruz
7	Elias Pinaglabanan

Employee_ID	Skill
1	Python
1	Java
2	C
2	C++
2	Java
3	C#
3	VB.NET
4	PHP
4	CSS
4	JavaScript
5	Python
5	R
6	Java
6	C
6	C++
7	MS Office

Fifth Normal Form – 5NF

- From 4NF, ensures that there are no overlapping non-trivial multivalued dependencies.

Eliminate MVD: $A \twoheadrightarrow \{B,C\}$

Employee_ID	Skill	Project_ID
2	C	BBB-001
2	C	AAA-002
2	C++	BBB-001
2	C++	AAA-002
2	Java	BBB-001
2	Java	AAA-002

Fifth Normal Form – 5NF

Employee_ID	Skill	Project_ID
2	C	BBB-001
2	C	AAA-002
2	C++	BBB-001
2	C++	AAA-002
2	Java	BBB-001
2	Java	AAA-002

⇩

Employee_ID	Skill
2	C
2	C++
2	Java

+

Employee_ID	Project_ID
2	BBB-001
2	AAA-002

+

Project_ID	Skill
BBB-001	C
AAA-002	C
BBB-001	C++
AAA-002	C++
BBB-001	Java
AAA-002	Java

Star Schema

- ❑ Alternative to Normal Forms
- ❑ Defines entities in a way that supports the view of the decision maker in a business and reflects the important operational aspects of a business itself.

Involves dividing large relations into smaller, more manageable relations and establishing relationships between them. It can benefit the long-term performance and cost of maintaining large databases.

Key Differences

Criteria	Normal Forms	Star Schema
Data Integrity	Ensures data integrity.	Introduces data redundancy.
Consistency	Promotes logical organization.	Less strict.
Transaction Optimization	Optimized for Online Transaction Processing (OLTP).	Optimized for Online Analytical Processing (OLAP), such as analytics, aggregation, and reporting.
Complex Querying	Retrieval involves complex joins especially for higher normal forms.	Simplified querying.
User-friendliness	Not intuitive for end-users.	Aligned structure to client’s perspective.

Denormalization

What is Denormalization?

- ❑ It refers to the intentional introduction of data redundancy into a normalized database to improve the query performance.

Often implemented when there is a strong need for faster read access and the overhead of maintaining the highly normalized tables outweighs the benefits.

Types of Denormalization

- Merging frequently used tables
- Vertical splitting when some attributes are rarely used
- Horizontal splitting for large tables (by year, etc.)
- Adding a table when multiple attributes in different tables are frequently used.
- Adding a history attribute.
- Adding a duplicate attribute to a different table.
- Adding a derived attribute for frequently used derived values.
- Adding a “previous value” column for error handling.

Performance Design Concepts

- ❑ **Concurrency:** Multiple concurrent users and transactions.
- ❑ **Scalability:** Handling increased load by adding new resources.
- ❑ **Resource Utilization:** Efficiency of resource usage.
- ❑ **Fault Tolerance:** Ability to continue functioning after failures.

Performance Design Considerations

- **Throughput:** Number of transactions in a given period.
- **Throughput Time:** Total time to fully complete a process.
- **Response Time:** Time taken to respond to a request.
- **Load Time:** Duration for a system until ready to use.

Summary

- ❑ Dependency is the general concept of the relationship between the attributes in a relation.
- ❑ Normalization involves the different levels of organizing and structuring of relation to remove data redundancy and improve data integrity.
- ❑ Denormalization the intentional introduction of data redundancy into a normalized database.

QUESTIONS?

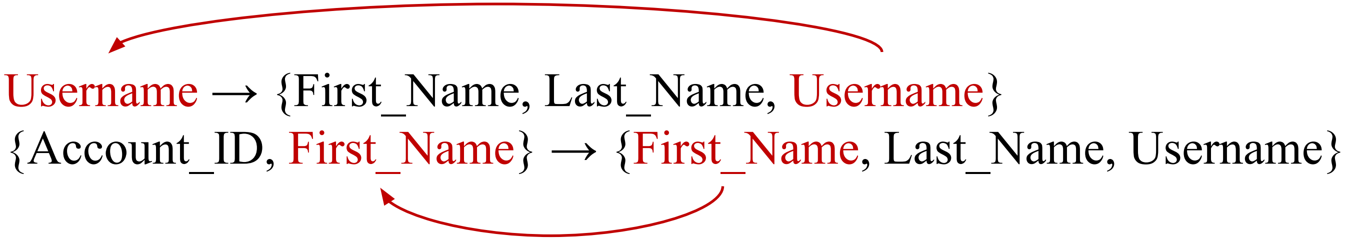
THANK YOU!

References

- Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387.
- Elmasri, R. et al. (2016). Fundamentals of Database Systems, 7th ed. Pearson Higher Education, 221 River Street, Hoboken, NJ 07030
- Kahl, W., Winter, M., & Oliveira, J. (2015). Relational and algebraic methods in computer science: 15th International Conference, RAMiCS 2015, Braga, Portugal, September 28 - October 1, 2015, Proceedings. Springer.

Trivial FD

- Occurs when an attribute is functionally dependent on itself or a superkey.
- It is trivial because it doesn’t add any new information.



Username → {First_Name, Last_Name, Username}

{Account_ID, First_Name} → {First_Name, Last_Name, Username}

* Basically, the first thing you shouldn’t do when designing databases.

Non-Trivial FD

❑ Occurs when an attribute is functionally dependent on a proper subset of the candidate keys.

Important Terms

❑ **Proper Subset:** A subset not equal to the original set.

❑ **Candidate Keys:** Set of keys to uniquely determine the tuples in a relation.

Username → {First_Name, Last_Name}
Account_ID → {First_Name, Last_Name}
Employee_ID → {Department_ID, Manager_ID}

Non-Trivial FD

{Employee_ID → {Department_ID, Manager_ID}

Employee_ID	Name	Department_ID	Manager_ID
1	Juan dela Cruz	2	521
2	Maria Sta Ana	5	345
3	Pedro Malaya	2	243
4	Juan Rosario	3	987
5	Mateo Abante	4	1235
6	Juan dela Cruz	1	231
7	Elias Pinaglabanan	4	902