



॥ त्वं ज्ञानमयो विद्वानमयोऽसि ॥

# Normalization and Denormalization of Relations

Saptarshi Pyne

Assistant Professor

Department of Computer Science and Engineering  
Indian Institute of Technology Jodhpur, Rajasthan, India 342030

**CSL4030 Data Engineering Lecture 8**  
**August 18<sup>th</sup>, 2023**

# What we discussed in the last class

## An Introduction to Relational Data Model

- Relations/tables, attributes, records/tuples
- Superkey, Candidate key, Primary key and foreign key
- Entity-Relational model, ER diagrams, UML
- Functional dependency, **Prime** and non-prime attributes
  - An attribute is called a 'prime attribute' of a relation if the attribute belongs to at least one **candidate key**.

# Things we will discuss today

- Why do we need to normalize relations?
- How can we normalize relations?
- When do we need to denormalize relations?

# Reference

- R. ELMASRI, S.B. NAVATHE (2017), Fundamentals of Database Systems, Pearson Education, 7<sup>th</sup> Edition.
  - Chapter 14: Basics of Functional Dependencies and Normalization for Relational Databases

# Why do we need to normalize relations?

- To avoid redundancies and
- 'anomalies' = inconsistencies
  - Insertion anomaly
  - Update anomaly
  - Deletion anomaly

# How can we normalize relations?

- By 'decomposing' relations into smaller relations while ensuring that the 'non-additive join property' or 'lossless join property' holds
- and optionally the 'dependency preservation property' holds
- Thus, we can achieve higher 'normal forms'.
  - What are the widely used normal forms?

# First normal form (1NF)

- A relation R is in 1NF if it **does not** possess any **multivalued attributes**.

# Second normal form (2NF)

- A relation R is in 2NF if
  - it is in **1NF**, and
  - no non-prime attributes are **partially functionally dependent** on the primary key.



# Third normal form (3NF)

- A relation R is in 3NF if
  - it is in **2NF**, and
  - no non-prime attributes are **transitively functionally dependent** on the primary key.

# Boyce-Codd normal form (BCNF)

- A relation  $R$  is in BCNF if
  - it is in **3NF**, and
  - whenever a nontrivial functional dependency  $X \rightarrow A$  holds in  $R$ , then  $X$  must be a **superkey** of  $R$ .

# When do we need to denormalize relations?

- In situations where we need to frequently 'join' multiple relations to serve incoming queries
- Types of joins
  - Inner Join
  - Left (outer) join
  - Right (outer) join
  - Full join
  - Natural join

# Things we discussed today

- Why do we need to normalize relations?
  - To avoid redundancies and 'anomalies'
- How can we normalize relations?
  - By 'decomposing' relations into smaller relations
- When do we need to denormalize relations?
  - When we have frequent 'join' queries

# Reference

- R. ELMASRI, S.B. NAVATHE (2017), Fundamentals of Database Systems, Pearson Education, 7<sup>th</sup> Edition.
  - Chapter 14: Basics of Functional Dependencies and Normalization for Relational Databases

# In the next class, we will discuss

- NoSQL databases

Thank you