University of Central Florida
CGS 2545
Database Concepts

DEPARTMENT OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE
COMPUTER SCIENCE DIVISION

Normalization

- Functional Dependency
  - 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 A1, A2,..., An, then those two tuples must have to have same values for attributes B1, B2, ..., Bn.

Normalization

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

Normalization

- Armstrong's Axioms
  - If F is a set of functional dependencies then the closure of F, denoted as F^+, is the set of all functional dependencies logically implied by F.
  - Armstrong's Axioms are a set of rules, that when applied repeatedly, generates a closure of functional dependencies.

Normalization

- Armstrong's Axioms
  - **Reflexive rule**
    - If alpha is a set of attributes and beta is_subset_of alpha, then alpha holds beta.

Normalization

- Armstrong's Axioms
  - **Augmentation rule**
    - If a → b holds and y is attribute set, then ay → by also holds.
    - That is adding attributes in dependencies, does not change the basic dependencies.`

Normalization

- Armstrong's Axioms
  - **Transitivity rule**
    - Same as transitive rule in algebra, if a → b holds and b → c holds, then a → c also holds
    - a → b is called as a functionally that determines b.

Normalization

- Trivial Functional Dependency
  - **Trivial**
    - If a functional dependency (FD) X → Y holds, where Y is a subset of X, then it is called a trivial FD.
    - Trivial FDs always hold.

Normalization

- Trivial Functional Dependency
  - **Non-trivial**
    - If an FD X → Y holds, where Y is not a subset of X, then it is called a non-trivial FD.

Normalization

- Trivial Functional Dependency
  - **Completely non-trivial**
    - If an FD X → Y holds, where x intersect Y = Φ, it is said to be a completely non-trivial FD.

Normalization

- Normalization
  - 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.
  - Normalization is a method to remove all these anomalies and bring the database to a consistent state.

Normalization

- Normalization
  - **Update anomalies**
    - 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.

Normalization

- Normalization
  - **Deletion anomalies**
    - Tried to delete a record, but parts of it was left undeleted because of unawareness, the data is also saved somewhere else.
  - **Insert anomalies**
    - Tried to insert data in a record that does not exist at all.

Normalization

- First Normal Form
  - First Normal Form is defined in the definition of relations (tables) itself.
  - 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

Normalization

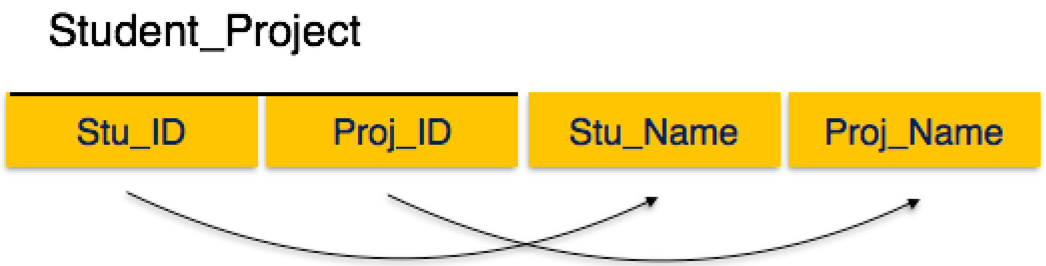
Course	Content
Programming	Java, c++
Web	HTML, PHP, ASP

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

Normalization

- Normalization
  - Second Normal Form
    - **Prime attribute -** An attribute, which is a part of the candidate-key, is known as a prime attribute.
    - **Non-prime attribute** - An attribute, which is not a part of the prime-key, is said to be a non-prime attribute.
    - Every non-prime attribute should be fully functionally dependent on prime key attribute.
    - That is, if X → A holds, then there should not be any proper subset Y of X, for which Y → A also holds true.

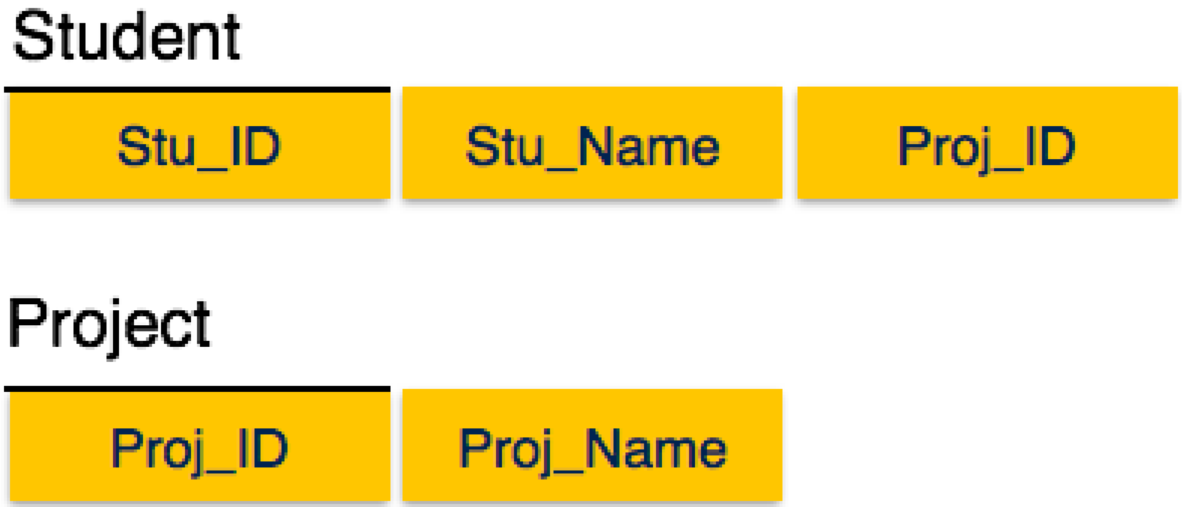
Normalization



- in Student_Project relation that the prime key attributes are Stu_ID and Proj_ID.
- According to the rule, non-key attributes, i.e. Stu_Name and Proj_Name must be dependent upon both and not on any of the prime key attribute individually.
- But we find that Stu_Name can be identified by Stu_ID and Proj_Name can be identified by Proj_ID independently.
- This is called **partial dependency**, which is not allowed in Second Normal Form.

Normalization

- Break the relation in two as depicted in the above picture.
- So there exists no partial dependency.



Student

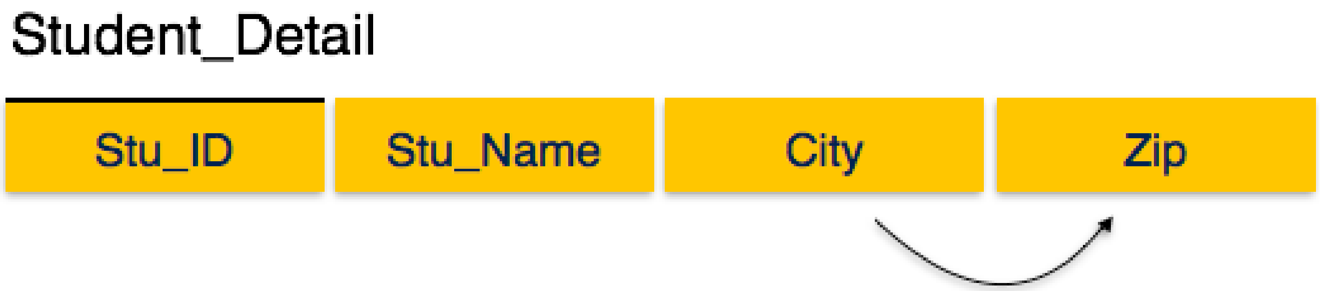
Stu_ID	Stu_Name	Proj_ID

Project

Proj_ID	Proj_Name

Normalization

- Third Normal Form
  - No non-prime attribute is transitively dependent on prime key attribute.
  - For any non-trivial functional dependency, X → A, then either –
    - X is a superkey or,
    - A is prime attribute.

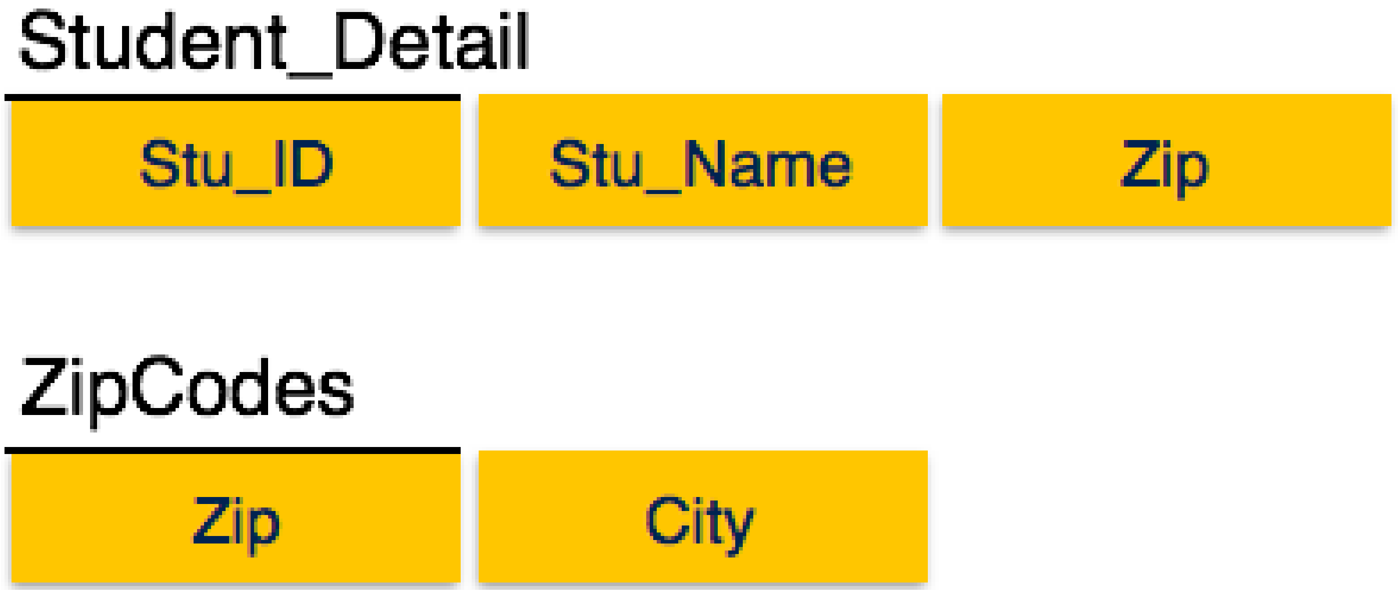


Normalization

- Third Normal Form
  - in the above Student_detail relation, Stu_ID is the key and only prime key attribute.
  - City can be identified by Stu_ID as well as Zip itself.
  - Neither Zip is a superkey nor is City a prime attribute. Additionally, Stu_ID → Zip → City, so there exists **transitive dependency**.

Normalization

- Break the relation into two relations as follows



Student_Detail

Stu_ID	Stu_Name	Zip

ZipCodes

Zip	City