DATA WAREHOUSING LECTURE 4

ENGR. MADEHA MUSHTAQ

DEPARTMENT OF COMPUTER SCIENCE

IQRA NATIONAL UNIVERSITY

NORMALIZATION

- Normalization is fundamentally about not having the same information twice or minimizing duplication.
- Normalization is to reduce and even eliminate data redundancy.
- Goals of normalization:
  - Eliminate redundant data.
  - Ensure data dependencies make sense.

NORMALIZATION

- Normalization is necessary to remove file maintenance anomalies, which are side effects of deletion, updating and insertion.
- 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

- Deletion anomalies:
  - We tried to delete a record, but parts of it was left undeleted because of unawareness, the data is also saved somewhere else.
- Insert anomalies:
  - We tried to insert data in a record that does not exist at all.
  - Normalization is a method to remove all these anomalies and bring the database to a consistent state.

NORMALIZATION

- Functional Dependency:
  - Functional dependency (FD) is a set of constraints between two attributes in a relation.
  - 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

- Partial Dependency:
  - PD exists, when for a composite primary key, any attribute in the table depends only on a part of the primary key and not on the complete primary key.
  - To remove Partial dependency, we can divide the table, remove the attribute which is causing partial dependency, and move it to some other table where it fits in well.

NORMALIZATION

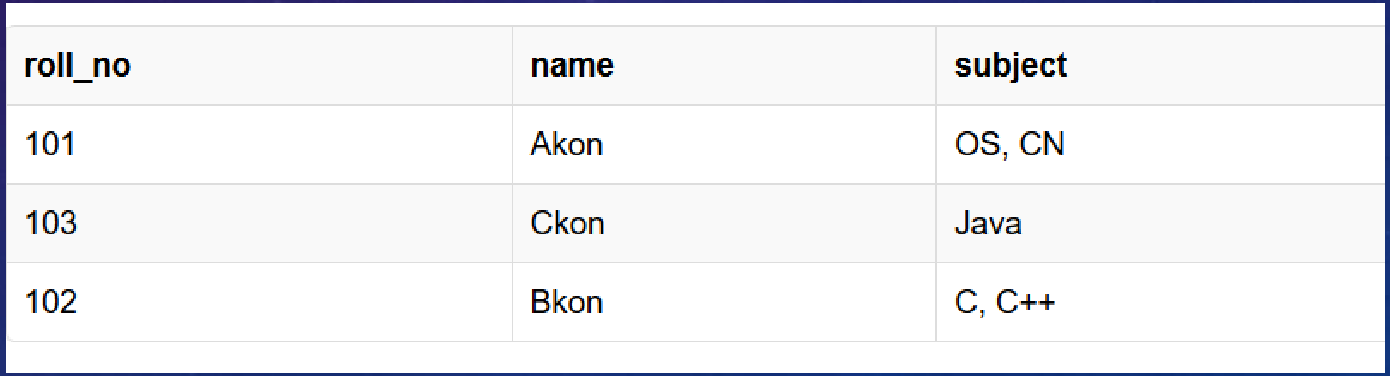
- Transitive Dependency:
  - When a non-key attribute depends on other non-key attributes rather than depending upon the primary key.
  - A transitive dependency is when changing a non-key column, might cause any of the other non-key columns to change.

NORMALIZATION

- Codd in 1971 in his seminal paper formulated a number of design principles for a relational database that can be formalized into three normal forms:
  - First Normal Form
  - Second Normal Form
  - Third Normal Form

NORMALIZATION :1NF

- All attributes must have a single(atomic) value, BUT it can contain redundant data.
- Each attribute must contain only a single value from its pre-defined domain.
- Table 1 is not in 1NF as each column is not containing single/atomic values.

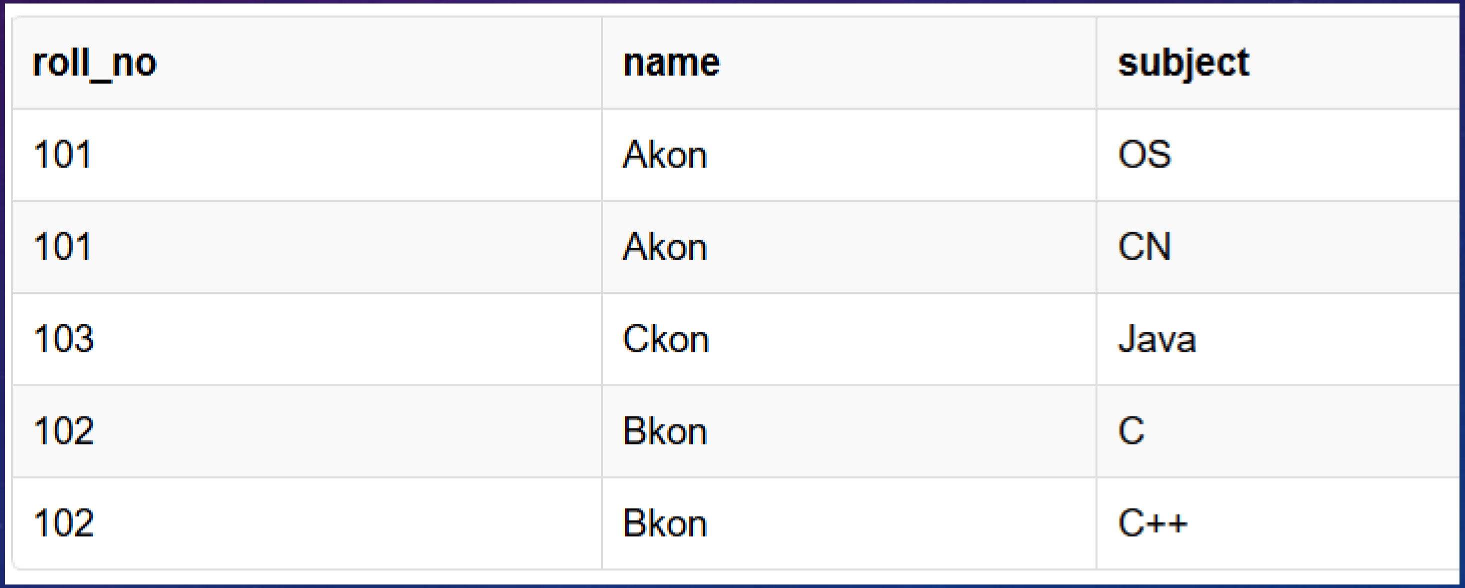


roll_no	name	subject
101	Akon	OS, CN
103	Ckon	Java
102	Bkon	C, C++

Table 1

NORMALIZATION :1NF

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

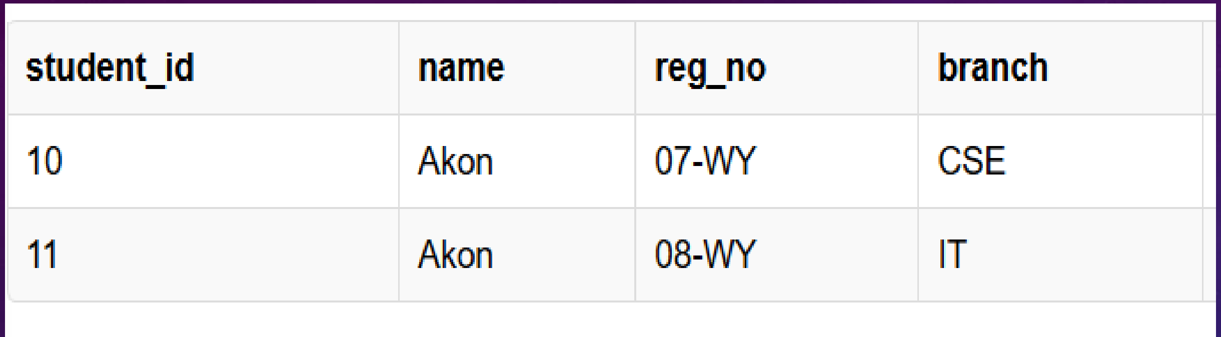


roll_no	name	subject
101	Akon	OS
101	Akon	CN
103	Ckon	Java
102	Bkon	C
102	Bkon	C++

NORMALIZATION: 2NF

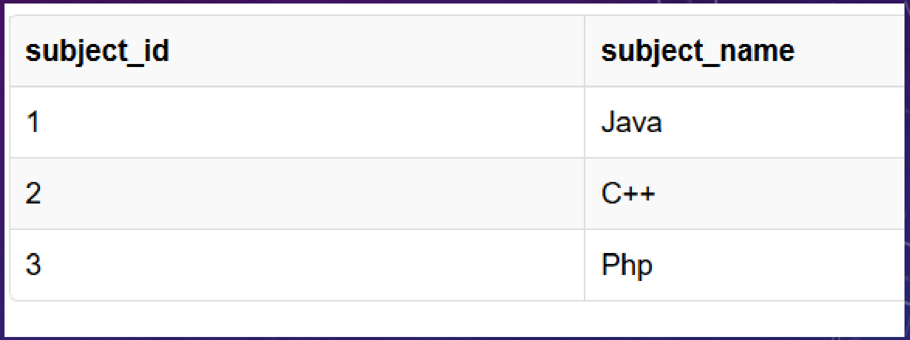
- Must be in First Normal Form and every non-key attribute is fully dependent on the primary key.
- This implies eliminating attributes which are not dependent on primary key.
- There should be no Partial Dependency.
- For student table, student_id is the primary key.
- For Subject table, subject_id is the primary key.
- For score table, we have a composite primary key (student_id, subject_id).

NORMALIZATION: 2NF



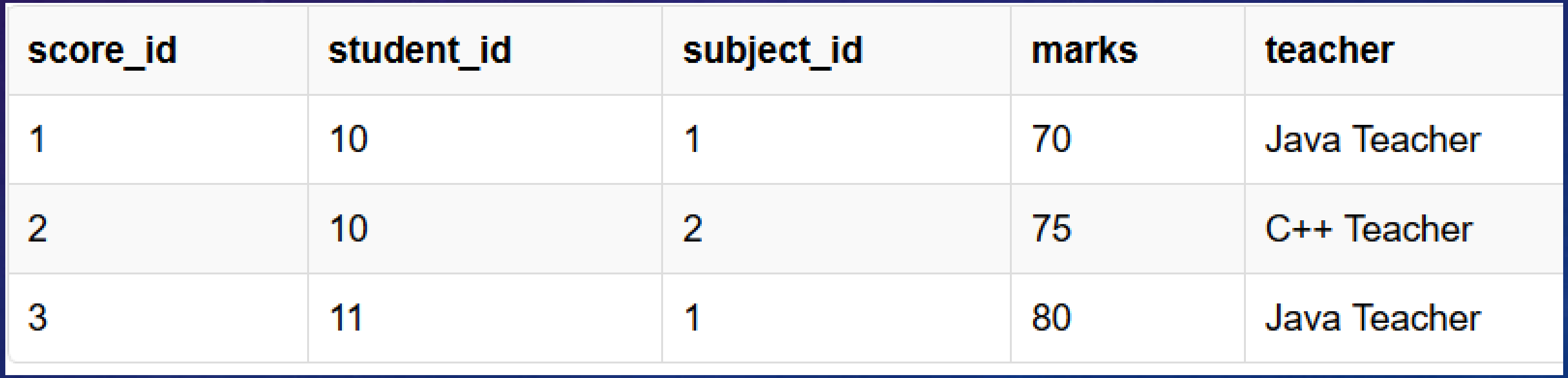
student_id	name	reg_no	branch
10	Akon	07-WY	CSE
11	Akon	08-WY	IT

Student



subject_id	subject_name
1	Java
2	C++
3	Php

Subject



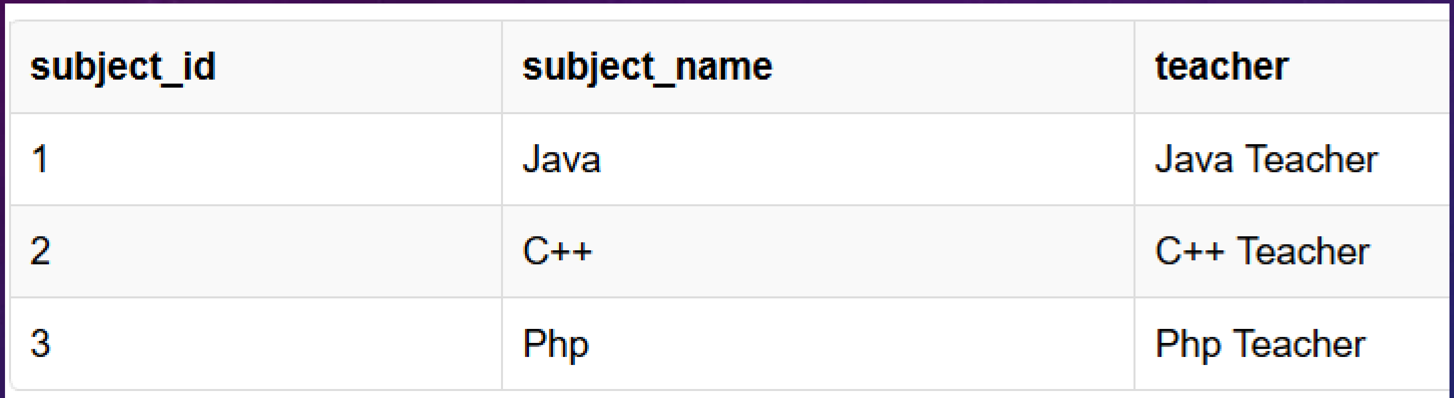
score_id	student_id	subject_id	marks	teacher
1	10	1	70	Java Teacher
2	10	2	75	C++ Teacher
3	11	1	80	Java Teacher

Score

NORMALIZATION: 2NF

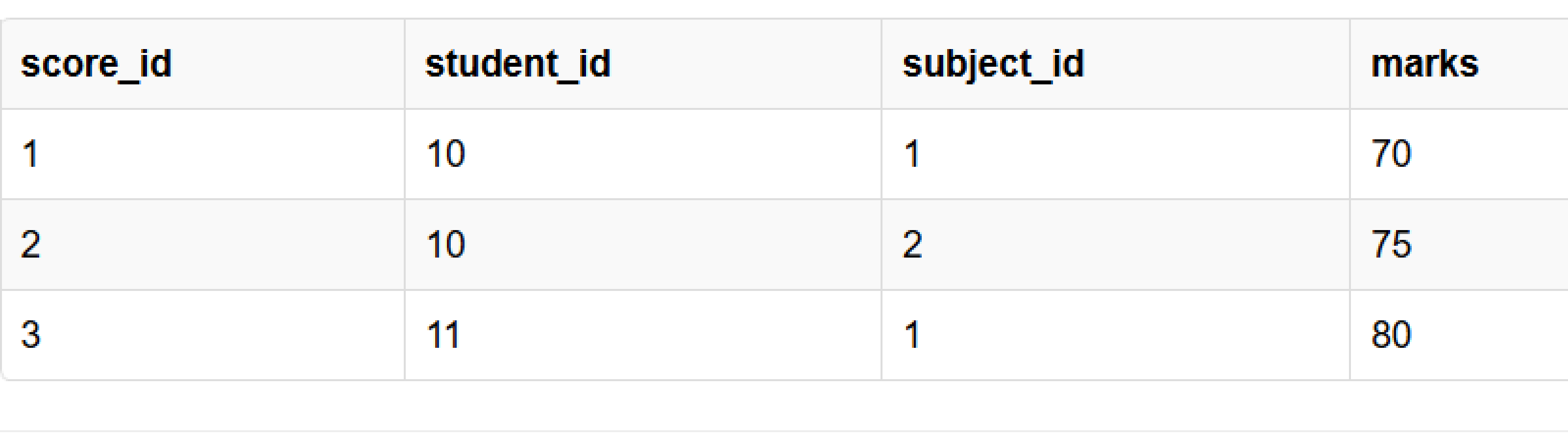
- Now if we look at the Score table, we have a column named teacher which is only dependent on the subject.
- Now as we just discussed that the primary key for this table is a composition of two columns which is student_id & subject_id but the teacher's name only depends on subject, hence the subject_id, and has nothing to do with student_id.
- This is Partial Dependency, where an attribute in a table depends on only a part of the primary key and not on the whole key.

NORMALIZATION: 2NF



subject_id	subject_name	teacher
1	Java	Java Teacher
2	C++	C++ Teacher
3	Php	Php Teacher

Our Score table is now in the second normal form, with no partial dependency.



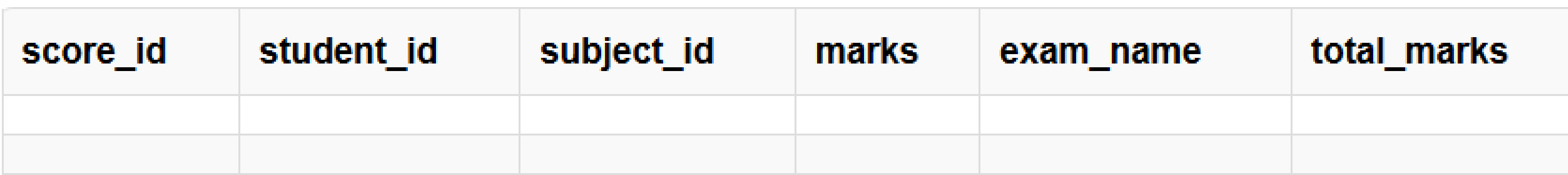
score_id	student_id	subject_id	marks
1	10	1	70
2	10	2	75
3	11	1	80

NORMALIZATION: 3NF

- Must be in second normal form and every non-key attribute is dependent on the primary key, the whole primary key and nothing but the primary key.
- This implies eliminating attributes which are dependent on a non-key attribute.
- There should be no Transitive Dependency.

NORMALIZATION: 3NF

- In the Score table, we need to store some more information, which is the exam name and total marks, so let's add 2 more columns to the Score table.



score_id	student_id	subject_id	marks	exam_name	total_marks

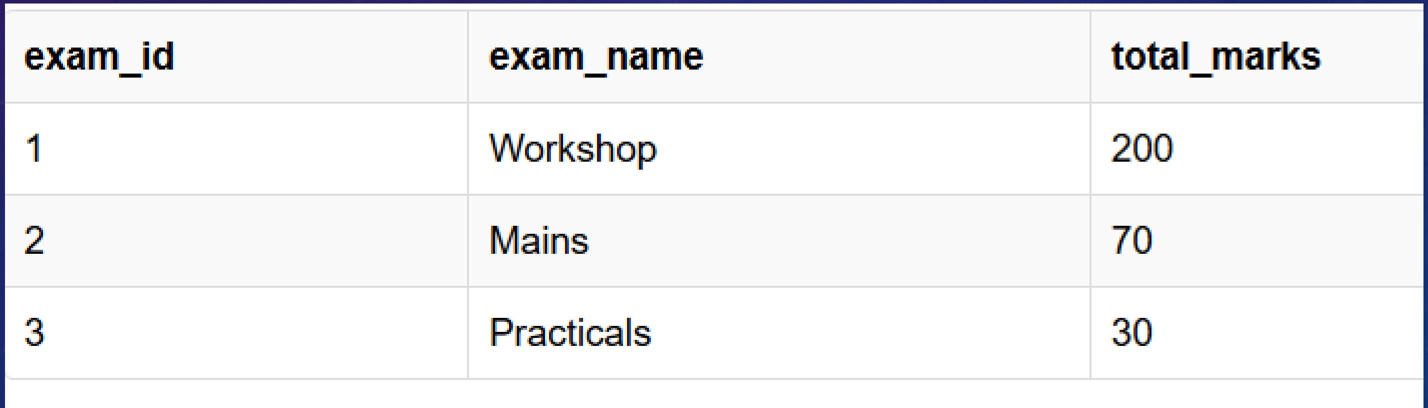
Primary key for our Score table is a composite key, **(student_id, subject_id)**

NORMALIZATION: 3NF

- Our new column exam_name is dependent on both student_id and subject_id.
- However, the other new column total_marks depends on exam_name as with exam type the total score changes.
- But exam_name is just another column in the score table. It is not a primary key.
- This is transitive dependency.

NORMALIZATION: 3NF

- To remove the transitive dependency, we can create a new table Exam and put the columns exam_name and total_marks in this new table.



exam_id	exam_name	total_marks
1	Workshop	200
2	Mains	70
3	Practicals	30

NORMALIZATION

- Conclusions:
- Generally a good idea is to only ensure 2NF.
- 3NF is at the cost of simplicity and performance.
- Normalization usually results in the formation of many tables.
- Normalization is well suited for transactional processing.
- But not for DSS as they aim at analyzing rather than processing.
- The multitude of tables causes problems in many analysis techniques that are designed to work in a DWH environment.

END OF SLIDES