SCHOOL ENROLMENT TABLE

Consider a table tracking **student enrolments in courses**.

A student can enrol in multiple courses, and each course has a specific teacher.

STU_ID	STU_NAME	COURSE_ID	COURSE_NAME	TEACHER_ID	TEACHER_NAME	GRADE
101	Alice	CSE101	Databases	T001	Dr Smith	A
101	Alice	CSE102	Networking	T002	Dr Johnson	B
102	Bob	CSE101	Databases	T001	Dr Smith	B+
103	Charlie	CSE103	AI Basics	T003	Dr Allen	A-
104	David	CSE102	Networking	T002	Dr Johnson	C

Step 1: Identify the Composite Primary Key

Since a student can take multiple courses, we need a composite primary key:

PK = ({STU_ID}, {COURSE_ID})

This ensures that each student-course combination is unique.

Step 2: Identify Functional Dependencies (FDs)

A functional dependency is when one attribute determines another.

- Student Dependencies:
  - STU_ID → STU_NAME (A student's name depends only on their ID)
- Course Dependencies:
  - COURSE_ID → COURSE_NAME, TEACHER_ID (Each course has a fixed name and teacher)
- Teacher Dependencies:
  - TEACHER_ID → TEACHER_NAME (Each teacher has a unique ID and name)
- Grade Dependencies:
  - (STU_ID, COURSE_ID) → GRADE (A student receives a unique grade per course)

Step 3: Identify Partial Dependencies (Violates 2NF)

A partial dependency occurs when a non-key attribute depends on only part of a composite primary key.

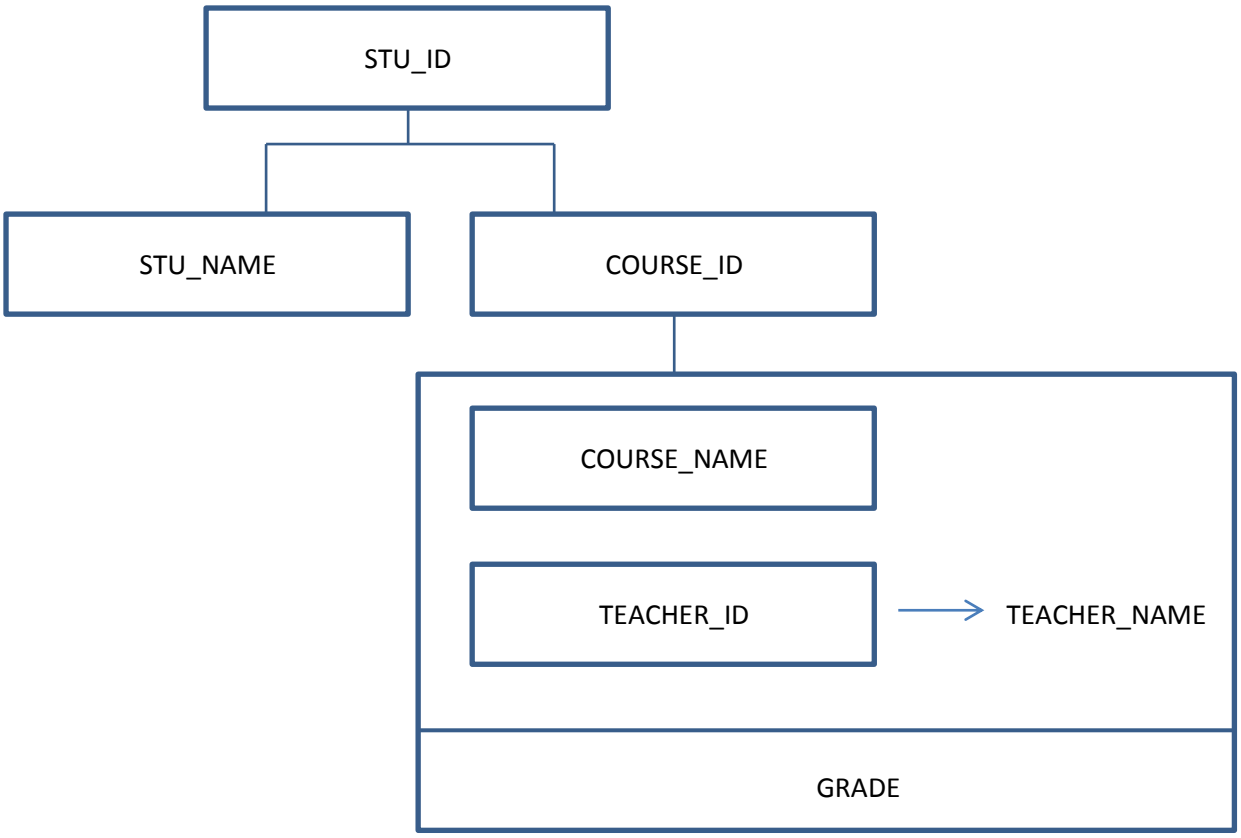
- STU_NAME depends only on STU_ID, not on COURSE_ID.
  - Partial Dependency: STU_ID → STU_NAME
- COURSE_NAME and TEACHER_ID depend only on COURSE_ID, not on STU_ID.
  - Partial Dependency: COURSE_ID → COURSE_NAME, TEACHER_ID
- TEACHER_NAME depends only on TEACHER_ID, not on the full primary key.
  - Partial Dependency: TEACHER_ID → TEACHER_NAME

Step 4: Identify Transitive Dependencies (Violates 3NF)

A transitive dependency occurs when a non-key attribute depends on another non-key attribute rather than directly on the primary key.

- TEACHER_NAME depends on TEACHER_ID, which depends on COURSE_ID.
  - Transitive Dependency: COURSE_ID → TEACHER_ID → TEACHER_NAME

Step 5: Draw the Dependency Diagram



Step 6: Normalize the Table into 3NF

To eliminate partial and transitive dependencies, we split the table into multiple normalised tables.

1. Students Table (Only Student-Related Data)

STU_ID (PK)	**STU_NAME**
101	Alice
102	Bob
103	Charlie
104	David

2. Courses Table (Only Course-Related Data)

COURSE_ID (PK)	COURSE_NAME	TEACHER_ID
CSE101	Databases	T001
CSE102	Networking	T002
CSE103	AI Basics	T003

3. Teachers Table (Only Teacher Data)

TEACHER_ID (PK)	TEACHER_NAME
T001	Dr Smith
T002	Dr Johnson
T003	Dr Allen

4. Enrolment Table (Links Students and Courses)

STU_ID (FK)	COURSE_ID (FK)
101	CSE101
101	CSE102
102	CSE101
103	CSE103
104	CSE102

Summary

- The original ENROLMENT table violated 2NF due to partial dependencies.
- It also violated 3NF due to transitive dependencies.
- We normalised it into four tables (Students, Courses, Teachers, and Enrolment).
- Now, data redundancy is reduced, and updates are more efficient.