

Phase 3: Normalized Relational Schema

Schema :

- Students (student_id, name, field_of_study, registration_status, internShip_id (FK references to internship (internShip_id)))

Domains:

student_id: INT (PK)

name: String(100)

field_of_study: String(100)

registration_status: String ('approved', 'rejected', 'pending')

FK: Students (internShip_id) references internShip (internShip_id)

- Enrollment (Student_id , Course code, since)

Domains:

student_id , : INT (PK)

course_code: String(20) (PK)

since : Date

- Organization (org_id, name, approval_status, Address , Approved , rejected)

Domains:

org_id: INT (PK)

name: String(100)

approval_status: String('approved', 'rejected', 'pending')

Address: String (200)

- Internship (internship_id, org_id(Fk), monitor_id(Fk), start_date, end_date)

Domains:

internship_id: INT (PK)

org_id: INT (FK → Organization)

monitor_id: INT (FK → Students)

start_date: DATE

end_date: DATE

FK: InternShip (org_id) references Organiztaion (Organization_id)

FK: InternShip (monitor_id) references monitor (monitor_id)

- Professor (Evaluation_id, ssn, grade, Name, feedback, prof_id, department)

Domains:

Evaluation_id : int (Pk)

Ssn : int (pk)

Grade : string (100)

Name : string (100)

Feedback: string (100)

prof_id : int

Department: string (100)

- Teaching_Assistant (Evaluation_id, ssn, grade, Name, feedback, assistant_id)

Domains:

assistant_id: INT (PK)

Evaluation_id : int (Pk)

Ssn : int (pk)

Grade : string (100)

Name : string (100)

Feedback: string (100)

- Monitor (monitor id, name, contact_details)

Domains:

monitor_id: INT (PK)

name: String(100)

Contact_Details: String(100)

- Course (course code, name, room_no)

Domains:

course_code: String(20) (PK)

name: String(100)

room_no: String(20)

- Report (Evaluation id, report id, Type, Submission_date)

Domains:

report_id: INT (PK)

type: String(50)

submission_date: DATE

evaluation_id: INT (Pk)

feedback: TEXT

student_id: INT (FK → Students)

Partial key : Reportid

- Teach's (proff id, Ass id, course code, Date)

functional dependencies:

1. Students Table

Fields: student_id, name, field_of_study, registration_status, internShip_id

Functional Dependencies:

student_id → name, field_of_study, registration_status

internShip_id is a foreign key (no functional dependency in this table)

Candidate Key: student_id

2. Enrollment Table

Fields: Student_id, Course_code, since

Functional Dependencies:

(Student_id, Course_code) → since

Candidate Key: Composite (Student_id, Course_code)

3. Organization Table

Fields: org_id, name, approval_status, Address, Approved, rejected

Functional Dependencies:

org_id → name, Address, approval_status

approval_status → Approved, rejected (transitive dependency - violates 3NF)

Candidate Key: org_id

4. Internship Table

Fields: internship_id, org_id, monitor_id, start_date, end_date

Functional Dependencies:

internship_id → org_id, monitor_id, start_date, end_date

Candidate Key: internship_id

5. Professor Table

Fields: Evaluation_id, ssn, grade, Name, feedback, prof_id, department

Functional Dependencies:

prof_id \rightarrow Name, ssn, department

Evaluation_id \rightarrow grade, feedback (mixed with professor data - violates 1NF)

Candidate Keys:

prof_id (for professor attributes)

Evaluation_id (for evaluation attributes)

6. Teaching_Assistant Table

Fields: Evaluation_id, ssn, grade, Name, feedback, assistant_id

Functional Dependencies:

assistant_id \rightarrow Name, ssn

Evaluation_id \rightarrow grade, feedback (mixed with TA data - violates 1NF)

Candidate Keys:

assistant_id (for TA attributes)

Evaluation_id (for evaluation attributes)

7. Monitor Table

Fields: monitor_id, name, contact_details

Functional Dependencies:

monitor_id \rightarrow name, contact_details

Candidate Key: monitor_id

8. Course Table

Fields: course_code, name, room_no

Functional Dependencies:

course_code → name, room_no

Candidate Key: course_code

9. Report Table

Fields: Evaluation_id, report_id, Type, Submission_date

Functional Dependencies:

report_id → Type, Submission_date, Evaluation_id

Candidate Key: report_id

10. Teach Table

Fields: proff_id, Ass_id, course_code, Date

Functional Dependencies:

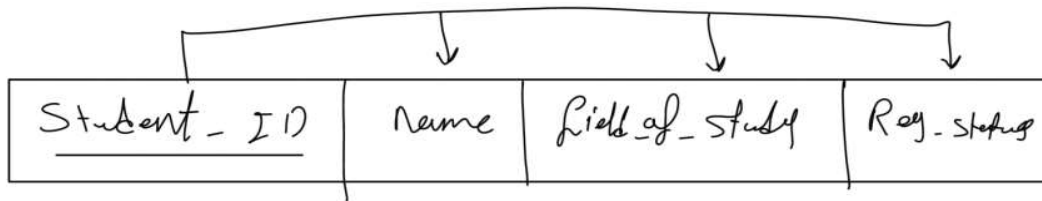
(proff_id, course_code) → Date

(Ass_id, course_code) → Date

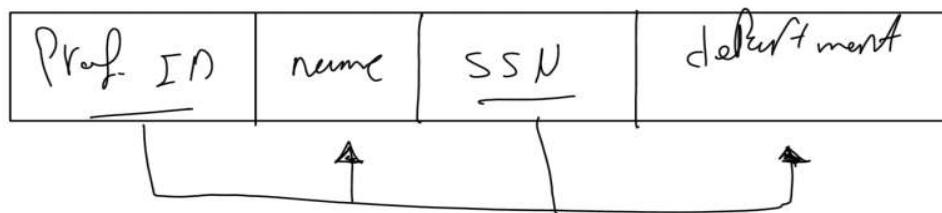
Candidate Key: Composite (proff_id, course_code, Ass_id)

Normalization:

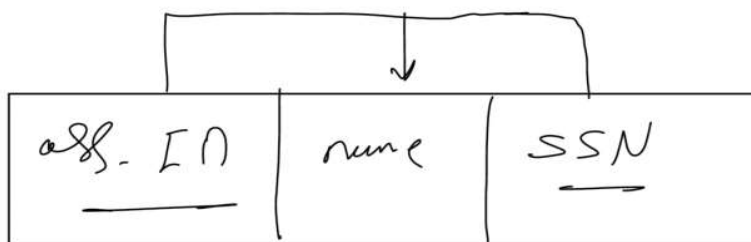
Student



Professor



Teaching Assistant



Report

<u>Report_ID</u>	<u>Nation_ID</u>	type	submission date
------------------	------------------	------	-----------------

organization

<u>org_ID</u>	name	address	approved-status
---------------	------	---------	-----------------

Course

<u>Course-code</u>	name	room_no
--------------------	------	---------

Monitor

<u>monitor_Id</u>	name	Contact-details
-------------------	------	-----------------

internship

<u>intern ID</u>	<u>org. ID</u>	<u>stud. ID</u>	<u>monitor. ID</u>	Start date
------------------	----------------	-----------------	--------------------	------------

Evaluation

<u>ev. ID</u>	<u>prof. ID</u>	<u>stud. ID</u>	grade	feedback	sub. date
---------------	-----------------	-----------------	-------	----------	-----------

Course Assignment

<u>Course-Code</u>	<u>Prof. ID</u>	<u>ass. ID</u>	ass. Date
--------------------	-----------------	----------------	-----------