PROG2111 – Relational Databases
Final Project
Normalization Report
2025 – 12 – 03
Kevin Napoles, Eric Moutoux

1NF

The first form consists in the attributes being atomic values, so to achieve it, we create a single table with one attribute per column without repeating them

StudentID

FirstName

LastName

Address

PhoneNumber

DateOfBirth

ProgramID

ProgramName

ProgramType

CourseID

CourseName

Section

Term

Grade

ContactID

ContactName

ContactAddress

ContactPhoneNumber

RelationshipToStudent

Data Assumptions:

There is one row per student per course per emergency contact.

All attributes are atomic values (no lists, no repeated columns like Course1, Course2, etc.).

2NF

To perform a second form normalization we need to get rid of all the partial dependencies so every non key attribute depends on the whole key not just part of it

We can identify the following dependencies:

A student's personal details depend only on StudentID:

StudentID → FirstName, LastName, Address, PhoneNumber, DateOfBirth, ProgramID

Program details depend only on ProgramID:

ProgramID → ProgramName, ProgramType

Course details depend only on CourseID:

CourseID → CourseName, Section

Emergency contact details depend only on ContactID:

ContactID → ContactName, ContactAddress, ContactPhoneNumber, RelationshipToStudent

Grade and term depend on the combination of student and course:

(StudentID, CourseID) → Term, Grade

It is apparent that several of the non-key fields (i.e. FirstName, ProgramName, CourseName, ContactName) are based on only part of the composite primary key (StudentID, CourseID, ContactID) and not the entirety of it.

This is an issue that violates 2NF and establishes redundancy. For example: ProgramName and ProgramType will be the same across each row for a student if they're taking multiple courses in the same program. CourseName and Section will be the same across each student who is

taking that course. ContactName and ContactAddress will be the same across each course the student takes.

To remove these partial dependencies, we split the big table into smaller relations where each non-key attribute depends on the whole key of that relation.

All attributes that describe the student in a student table:

Student:

Student(StudentID(PK), FirstName, LastName, Address, PhoneNumber, DateOfBirth, ProgramID)

Every attribute depends on StudentID

Program:

Program(ProgramID(PK), ProgramName, ProgramType)

Course:

Course(CourseID(PK), CourseName, Section)

EmergencyContact:

EmergencyContact(ContactID(PK), StudentID(FK), ContactName, ContactAddress, ContactPhoneNumber, RelationshipToStudent)

StudentID is a foreign key indicating which student the emergency contact belongs to

Enrollment:

Enrollment(StudentID, CourseID, Term, Grade)

3NF

The goal of Third Normal Form (3NF) is to remove transitive dependencies.

A transitive dependency exists when A → B and B → C, so A → C: a non-key attribute depends on the key *through* another non-key attribute.

In the non-normalized table, we know the following:

- StudentID → ProgramID and ProgramID → ProgramName, ProgramType
  ⇒ transitive: StudentID → ProgramName, ProgramType
- CourseID → CourseName, Section
  ⇒ for each entry (StudentID, CourseID) → CourseName, Section
- ContactID → ContactName, ContactAddress, ContactPhoneNumber, RelationshipToStudent
  Because this ContactID was kept with Student and the course info, this means this same contact info was found with every course ever taken by this student.

To achieve 3NF we want to make sure that in all tables:

- Every non-key depends upon the entire key
- There is no existing non-key → non-key dependency within a given table.

Following are the decomposed tables:

- Student(StudentID (PK), FirstName, LastName, Address, PhoneNumber, DateOfBirth, ProgramID (FK))
- Program(ProgramID (PK), ProgramName, ProgramType)
- Course(CourseID (PK), CourseName, Section)
- EmergencyContact(ContactID (PK), StudentID (FK), ContactName, ContactAddress, ContactPhoneNumber, RelationshipToStudent)
- Enrollment(StudentID (PK), CourseID (PK), Term, Grade)

Now the only functional dependencies that exist are:

- StudentID → FirstName, LastName, Address, PhoneNumber, DateOfBirth, ProgramID
- ProgramID → ProgramName, ProgramType
- CourseID → CourseName, Section

- ContactID → StudentID, ContactName, ContactAddress, ContactPhoneNumber, RelationshipToStudent
- (StudentID, CourseID) → Term, Grade

Where the left-hand side of the dependency is a key (or part of a composite key in Enrollment) and where no attributes depend on another non-key attribute. Thus all relationships are in Third Normal Form (3NF).

Eliminated Redundant Data

In the original one big table, there were so many redundancies:

ProgramName, ProgramType

Each student has a program and the table repeats these columns for every row where a student in that program takes multiple courses. After normalization: they exist in the Program table only once per ProgramID.

CourseName, Section

Students take different courses and each course is repeated for every student taking the same course. After normalization: they exist in the Course table only once per CourseID.

ContactName, ContactAddress, ContactPhoneNumber, RelationshipToStudent

An emergency contact is listed for every course the student takes, but the same emergency contact is effectively repeated on each row. After normalization: they exist in the EmergencyContact table only once per ContactID, linked to the student by StudentID.

Term, Grade

These attributes were initially tied to the composite (StudentID, CourseID) inside the one big table where they were repeated along with all of the other student, course and contact information. After normalization: Term and Grade now exist only once in the Enrollment table for each (StudentID, CourseID) pair.