

Database Normalization Tutorial — Student-Course Example

مقدمة

Data Integrity هي عملية تصميم قاعدة البيانات بطريقة تقلل التكرار وتزيد **Database Normalization**.

أهم الأهداف

1. Ensure data integrity
2. No Redundancy
3. No Update or insert Anomalies
4. Simplicity and clearness

المراحل الرئيسية

1NF — First Normal Form

2NF — Second Normal Form

3NF — Third Normal Form

المثال المستخدم

لدينا جدول بيانات الطلاب والدورات

Primary Key: (StudentID + CourseID)

| StudentID | Student Name | Student Skills | Course1 | Course2 | Course3 | Course1Instructor | Course2Instructor | Course3Instructor | SectionID | Section Name |
|-----------|--------------|----------------|---------|---------|-----------|-------------------|-------------------|-------------------|-----------|--------------|
| 1 | Ahmed | SQL, C# | Math | Physics | Chemistry | Dr.Ali | Dr.Sara | Dr.Bob | 10 | A |
| 2 | Omar | Python | Math | Biology | NULL | Dr.Ali | Dr.Lina | NULL | 11 | B |

مشاكل هذا الجدول

Repeating Groups: Course1, Course2, Course3 و Course1Instructor...

Repeating Group يعني أن نفس نوع البيانات يتكرر عدة مرات لنفس الكيان (**Entity**) داخل الجدول.

Multi-valued Attributes: StudentSkills يحتوي أكثر من مهارة في نفس الخلية

Partial Dependency: StudentSkills

Transitive Dependency: SectionName على SectionID وليس على StudentID مباشرة

Step 1 — First Normal Form (1NF)

1. كل خلية (Cell) تحتوي قيمة واحدة فقط (Atomic Value)

2. إزالة Repeating Groups

3. Unique يكون كل صف (Row)

Aply1NF

No Multi-valued Attributes

| StudentID | StudentName | StudentSkill | CourseID | Instructor | SectionID | SectionName |
|-----------|-------------|--------------|----------|------------|-----------|-------------|
| 1 | Ahmed | SQL | Math | Dr.Ali | 10 | A |
| 1 | Ahmed | C# | Math | Dr.Ali | 10 | A |
| 1 | Ahmed | SQL | Physics | Dr.Sara | 10 | A |
| 1 | Ahmed | C# | Physics | Dr.Sara | 10 | A |
| 2 | Omar | Python | Math | Dr.Ali | 11 | B |
| 2 | Omar | Python | Biology | Dr.Lina | 11 | B |

Then

No Repeating Groups

Students Table

| StudentID | StudentName | CourseID | Instructor | SectionID | SectionName |
|-----------|-------------|----------|------------|-----------|-------------|
| 1 | Ahmed | Math | Dr.Ali | 10 | A |
| 1 | Ahmed | Physics | Dr.Sara | 10 | A |
| 2 | Omar | Math | Dr.Ali | 11 | B |
| 2 | Omar | Biology | Dr.Lina | 11 | B |

StudentSkills Table

| StudentID | Skill |
|-----------|--------|
| 1 | SQL |
| 1 | C# |
| 2 | Python |

Step 2 — Second Normal Form (2NF)

1- Must be in 1NF

2. إزالة **Partial Dependency**: أي عمود غير المفتاح يعتمد على جزء فقط من المفتاح المركب

الجدول:

StudentSkills Table

| StudentID | Skill |
|-----------|--------|
| 1 | SQL |
| 1 | C# |
| 2 | Python |

Students Table

| StudentID | StudentName | SectionID | SectionName |
|-----------|-------------|-----------|-------------|
| 1 | Ahmed | 10 | A |
| 2 | Omar | 11 | B |

Courses Table

| CourseID | Course Name | Instructor |
|----------|-------------|------------|
| 1 | Math | Dr.Ali |
| 2 | Physics | Dr.Sara |
| 3 | Chemistry | Dr.Bob |

StudentCourses Table

| StudentID | CourseID |
|-----------|----------|
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |

Step 3 — Third Normal Form (3NF)

1- Must be in 2NF

2. إزالة **Transitive Dependency**: أي عمود غير مفتاح يعتمد على عمود غير مفتاح آخر

3NF: الجداول بعد 3

Students Table

| StudentID | StudentName | SectionID |
|-----------|-------------|-----------|
| 1 | Ahmed | 10 |
| 2 | Omar | 11 |

Sections Table

| SectionID | SectionName |
|-----------|-------------|
| 10 | A |
| 11 | B |

StudentSkills Table

| StudentID | Skill |
|-----------|--------|
| 1 | SQL |
| 1 | C# |
| 2 | Python |

Courses Table

| CourseID | Course Name | Instructor |
|----------|-------------|------------|
| 1 | Math | Dr.Ali |
| 2 | Physics | Dr.Sara |
| 3 | Chemistry | Dr.Bob |

StudentCourses Table

| StudentID | CourseID |
|-----------|----------|
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |

Final Tables After 3NF

1 Students Table

StudentID | StudentName | SectionID |

2 Sections Table

SectionID | SectionName |

3 Courses Table

CourseID | Instructor |

4 StudentCourses Table (*PK = StudentID + CourseID*)

StudentID | CourseID |

5 StudentSkills Table (*PK = StudentID + Skill*)

Relationships (ERD Conceptual)

- Students ↔ Sections → 1-to-1
 - Students ↔ Courses → many-to-many via StudentCourses
 - Students ↔ Skills → 1-to-many via StudentSkills
-

Normalization Summary

| Step | Action |
|------|--|
| 1NF | Atomic values, remove repeating groups |
| 2NF | Remove Partial Dependencies |
| 3NF | Remove Transitive Dependencies |