

## 12: Normalization

Normalization should be part of the database design process. However, it is difficult to separate the normalization process from the ER modelling process so the two techniques should be used concurrently.

Use an entity relation diagram (ERD) to provide the big picture, or macro view, of an organization's data requirements and operations. This is created through an iterative process that involves identifying relevant entities, their attributes and their relationships.

Normalization procedure focuses on characteristics of specific entities and represents the micro view of entities within the ERD.

### What Is Normalization?

*Normalization* is the branch of relational theory that provides design insights. It is the process of determining how much redundancy exists in a table. The goals of normalization are to:

- Be able to characterize the level of redundancy in a relational schema
- Provide mechanisms for transforming schemas in order to remove redundancy

Normalization theory draws heavily on the theory of functional dependencies. Normalization theory defines six normal forms (NF). Each normal form involves a set of dependency properties that a schema must satisfy and each normal form gives guarantees about the presence and/or absence of update anomalies. This means that higher normal forms have less redundancy, and as a result, fewer update problems.

### Normal Forms

All the tables in any database can be in one of the normal forms we will discuss next. Ideally we only want minimal redundancy for PK to FK. Everything else should be derived from other tables. There are six normal forms, but we will only look at the first four, which are:

- First normal form (1NF)
- Second normal form (2NF)
- Third normal form (3NF)
- Boyce-Codd normal form (BCNF)

BCNF is rarely used.

### First Normal Form (1NF)

In the *first normal form*, only single values are permitted at the intersection of each row and column; hence, there are no repeating groups.

To normalize a relation that contains a repeating group, remove the repeating group and form two new relations.

The PK of the new relation is a combination of the PK of the original relation plus an attribute from the newly created relation for unique identification.

### Process for 1NF

We will use the **Student\_Grade\_Report** table below, from a School database, as our example to explain the process for 1NF.

**Student\_Grade\_Report** (StudentNo, StudentName, Major, CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)

- In the Student Grade Report table, the repeating group is the course information. A student can take many courses.
- Remove the repeating group. In this case, it's the course information for each student.
- Identify the PK for your new table.
- The PK must uniquely identify the attribute value (StudentNo and CourseNo).
- After removing all the attributes related to the course and student, you are left with the student course table (**StudentCourse**).
- The Student table (**Student**) is now in first normal form with the repeating group removed.
- The two new tables are shown below.

**Student** (StudentNo, StudentName, Major) **StudentCourse** (StudentNo, CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)

### How to update 1NF anomalies

**StudentCourse** (StudentNo, CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)

- To add a new course, we need a student.
- When course information needs to be updated, we may have inconsistencies.
- To delete a *student*, we might also delete critical information about a course.

## Second Normal Form (2NF)

For the *second normal form*, the relation must first be in 1NF. The relation is automatically in 2NF if, and only if, the PK comprises a single attribute.

If the relation has a composite PK, then each non-key attribute must be fully dependent on the entire PK and not on a subset of the PK (i.e., there must be no partial dependency or augmentation).

### Process for 2NF

To move to 2NF, a table must first be in 1NF.

- The Student table is already in 2NF because it has a single-column PK.
- When examining the Student Course table, we see that not all the attributes are fully dependent on the PK; specifically, all course information. The only attribute that is fully dependent is grade.
- Identify the new table that contains the course information.
- Identify the PK for the new table.
- The three new tables are shown below.

**Student** (StudentNo, StudentName, Major) **CourseGrade** (StudentNo, CourseNo, Grade) **CourseInstructor** (CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation)

### How to update 2NF anomalies

- When adding a new instructor, we need a course.
- Updating course information could lead to inconsistencies for instructor information.
- Deleting a course may also delete instructor information.

## Third Normal Form (3NF)

To be in *third normal form*, the relation must be in second normal form. Also all transitive dependencies must be removed; a non-key attribute may not be functionally dependent on another non-key attribute.

### Process for 3NF

- Eliminate all dependent attributes in transitive relationship(s) from each of the tables that have a transitive relationship.
- Create new table(s) with removed dependency.
- Check new table(s) as well as table(s) modified to make sure that each table has a determinant and that no table contains inappropriate dependencies.
- See the four new tables below.

**Student** (StudentNo, StudentName, Major) **CourseGrade** (StudentNo, CourseNo, Grade) **Course** (CourseNo, CourseName, InstructorNo) **Instructor** (InstructorNo, InstructorName, InstructorLocation)

At this stage, there should be no anomalies in third normal form. Let's look at the dependency diagram (Figure 12.1) for this example. The first step is to remove repeating groups, as discussed above.

**Student** (StudentNo, StudentName, Major)

**StudentCourse** (StudentNo, CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)

To recap the normalization process for the School database, review the dependencies shown in Figure 12.1.

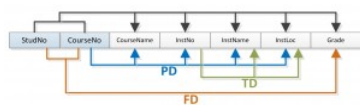


Figure 12.1 Dependency diagram, by A. Watt.

The abbreviations used in Figure 12.1 are as follows:

- PD: partial dependency
- TD: transitive dependency
- FD: full dependency (Note: FD typically stands for **functional** dependency. Using FD as an abbreviation for full dependency is only used in Figure 12.1.)

## Boyce-Codd Normal Form (BCNF)

When a table has more than one candidate key, anomalies may result even though the relation is in 3NF. *Boyce-Codd normal form* is a special case of 3NF. A relation is in BCNF if, and only if, every determinant is a candidate key.

### BCNF Example 1

Consider the following table (**St\_Maj\_Adv**).

Student_id	Major	Advisor
111	Physics	Smith
111	Music	Chan
320	Math	Dobbs
671	Physics	White
803	Physics	Smith

The *semantic rules* (business rules applied to the database) for this table are:

1. Each Student may major in several subjects.
2. For each Major, a given Student has only one Advisor.
3. Each Major has several Advisors.
4. Each Advisor advises only one Major.
5. Each Advisor advises several Students in one Major.

The functional dependencies for this table are listed below. The first one is a candidate key; the second is not.

1. Student\_id, Major  $\longrightarrow$  Advisor
2. Advisor  $\longrightarrow$  Major

Anomalies for this table include:

1. Delete – student deletes advisor info
2. Insert – a new advisor needs a student
3. Update – inconsistencies

**Note:** No single attribute is a candidate key.

PK can be Student\_id, Major or Student\_id, Advisor.

To reduce the **St\_Maj\_Adv** relation to BCNF, you create two new tables:

1. **St\_Adv** (Student\_id, Advisor)
2. **Adv\_Maj** (Advisor, Major)

**St\_Adv** table

Student_id	Advisor
111	Smith

111	Chan
320	Dobbs
671	White
803	Smith

**Adv\_Maj** table

Advisor	Major
Smith	Physics
Chan	Music
Dobbs	Math
White	Physics

## BCNF Example 2

Consider the following table (**Client\_Interview**).

ClientNo	InterviewDate	InterviewTime	StaffNo	RoomNo
CR76	13-May-02	10.30	SG5	G101
CR56	13-May-02	12.00	SG5	G101
CR74	13-May-02	12.00	SG37	G102
CR56	1-July-02	10.30	SG5	G102

FD1 – ClientNo, InterviewDate → InterviewTime, StaffNo, RoomNo (PK)

FD2 – staffNo, interviewDate, interviewTime → clientNO (candidate key: CK)

FD3 – roomNo, interviewDate, interviewTime → staffNo, clientNo (CK)

FD4 – staffNo, interviewDate → roomNo

A relation is in BCNF if, and only if, every determinant is a candidate key. We need to create a table that incorporates the first three FDs (**Client\_Interview2** table) and another table (**StaffRoom** table) for the fourth FD.

**Client\_Interview2** table

ClientNo	InterviewDate	InterViewTime	StaffNo
CR76	13-May-02	10.30	SG5
CR56	13-May-02	12.00	SG5
CR74	13-May-02	12.00	SG37
CR56	1-July-02	10.30	SG5

**StaffRoom** table

StaffNo	InterviewDate	RoomNo
SG5	13-May-02	G101
SG37	13-May-02	G102
SG5	1-July-02	G102

## Normalization and Database Design

During the normalization process of database design, make sure that proposed entities meet required normal form before table structures are created. Many real-world databases have been improperly designed or burdened with anomalies if improperly modified during the course of time. You may be asked to redesign and modify existing databases. This can be a large undertaking if the tables are not properly normalized.

### Key Terms and Abbreviations

**Boyce-Codd normal form (BCNF):** a special case of 3rd NF

**first normal form (1NF):** only single values are permitted at the intersection of each row and column so there are no repeating groups

**normalization:** the process of determining how much redundancy exists in a table

**second normal form (2NF):** the relation must be in 1NF and the PK comprises a single attribute

**semantic rules:** business rules applied to the database

**third normal form (3NF):** the relation must be in 2NF and all transitive dependencies must be removed; a non-key attribute may not be functionally dependent on another non-key attribute

### Exercises

Complete chapters 11 and 12 before doing these exercises.

1. What is normalization?
2. When is a table in 1NF?
3. When is a table in 2NF?
4. When is a table in 3NF?
5. Identify and discuss each of the indicated dependencies in the dependency diagram shown in Figure 12.2.

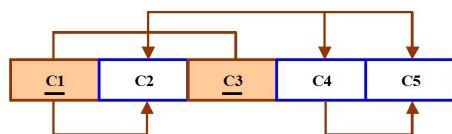


Figure 12.2 For question 5, by A. Watt.

6. To keep track of students and courses, a new college uses the table structure in Figure 12.3.

Attribute Name	Sample Value	Sample Value	Sample Value
StudentID	1	2	3
StudentName	John Smith	Sandy Law	Sue Rogers
CourseID	2	2	3
CourseName	Programming Level 1	Programming Level 1	Business
Grade	75%	61%	81%
CourseDate	Jan 5 <sup>th</sup> , 2014	Jan 5 <sup>th</sup> , 2014	Jan 7 <sup>th</sup> , 2014

Draw the dependency diagram for this table.  
question 6, by A. Watt.

Figure 12.3 For

7. Using the dependency diagram you just drew, show the tables (in their third normal form) you would create to fix the problems you encountered. Draw the dependency diagram for the fixed table.
8. An agency called Instant Cover supplies part-time/temporary staff to hotels in Scotland. Figure 12.4 lists the time spent by agency staff working at various hotels. The national insurance number (NIN) is unique for every member of staff. Use Figure

NIN	ContractNo	Hours	eName	hNo	hLoc
1135	C1024	16	Smith J.	H25	East Killbride
1057	C1024	24	Hocine D.	H25	East Killbride
1068	C1025	28	White T.	H4	Glasgow
1135	C1025	15	Smith J.	H4	Glasgow

12.4 to answer questions (a) and (b).

Figure 12.4 For question 8, by A. Watt.

1. This table is susceptible to update anomalies. Provide examples of insertion, deletion and update anomalies.
  2. Normalize this table to third normal form. State any assumptions.
9. Fill in the blanks:
1. \_\_\_\_\_ produces a lower normal form.
  2. Any attribute whose value determines other values within a row is called a(n) \_\_\_\_\_.
  3. An attribute that cannot be further divided is said to display \_\_\_\_\_.
  4. \_\_\_\_\_ refers to the level of detail represented by the values stored in a table's row.
  5. A relational table must not contain \_\_\_\_\_ groups.

Also see *Appendix B: Sample ERD Exercises*

## Bibliography

Nguyen Kim Anh, *Relational Design Theory*. OpenStax CNX. 8 Jul 2009 Retrieved July 2014 from <http://cnx.org/contents/606cc532-0b1...4e2e2d533b@1@1>

Russell, Gordon. Chapter 4 – Normalisation. *Database eLearning*. N.d. Retrived July 2014 from [db.grussell.org/ch4.html](http://db.grussell.org/ch4.html)

---

12: Normalization is shared under a [CC BY](#) license and was authored, remixed, and/or curated by LibreTexts.

- **1.12: Normalization** by Adrienne Watt is licensed [CC BY 4.0](#). Original source: <https://opentextbc.ca/dbdesign01/>.