Lecture 6

Steps in Normalization

- Summary of Definitions of the Normal Forms
- Functional Dependency and Determinants
- The 1st Normal Form (1NF)
- The 2nd Normal Form (2NF)
- Anomalies and Normalization
- Turning a Table with Anomalies into Single-Theme Tables
- The 3rd Normal Form (3NF)
- The Boyce-Codd Normal Form (BCNF)
- The 4th Normal Form (4NF)
- The 5th Normal Form (5NF)
- The Domain-Key Normal Form (DKNF)

Logical designing of database

There are two approaches to the logical design of the database:
-The top-down approach
-Bottom-up approach

- Method of E/R model is top-down approach. Method includes defining the entities, relationships and attributes tracing scheme E/R and mapping schema.

- Normalization is a bottom-up approach. This is a step in the decomposition of complex records are simple.

- Normalization reduces redundancy, using the principle of partition.

- Splitting is the conversion table in the smaller tables without losing information.

- Top-down approach is best suited to test the existing developments.

Through normalization we want to design for our relational database a set of files that

- (1) contain all the data necessary for the purposes that the database is to serve,
- (2) have as little redundancy as possible,
- (3) accommodate multiple values for types of data that require them,
- (4) permit efficient updates of the data in the database, and
- (5) avoid the danger of losing data unknowingly.

Data redundancy

- **Data redundancy** means their repeatability. Redundancy increases the time it takes to update, add, and delete data.

- Redundancy also increases the use of disk space, and, as a consequence, increases the number of disk accesses.

- Consequence of redundancy can be:
  - Update anomalies - insertion, updation and deletion of data can cause errors.
  - Inconsistency - the error rate increases with repeated recording of facts.
  - Undue consumption of disk space.

Example of data redundancy

Consider the structure of the table student of

STUDENT (_Code, _Name, _DateOfBirth, _Address, _Sity, _Specialty,_ Group, _Semester, _Quiz1, _ Quiz2)

How to fill in the data table Student:

Code	**Name**	**…**	**Semester**	**Quiz1**	**Quiz2**
001	Aida	…	SEM-1	40	65
001	Aida	…	SEM-2	56	48
002	Zhandos	…	SEM-1	93	84
002	Zhandos	…	SEM-2	85	90

The need for data normalization

- Normalization can be viewed as a series of steps (i.e., levels) designed, one after another, to deal with ways in which tables can be "too complicated for their own good".
- The purpose of normalization is to reduce the chances for anomalies to occur in a database.
- The definitions of the various levels of normalization illustrate complications to be eliminated in order to reduce the chances of anomalies.
- At all levels and in every case of a table with a complication, the resolution of the problem turns out to be the establishment of two or more simpler tables which, as a group, contain the same information as the original table but which, because of their simpler individual structures, lack the complication.

Functional Dependency and Determinants

The essence of this idea is that if the existence of something, call it A, implies that B must exist and have a certain value, then we say that "B is functionally dependent on A." We also often express this idea by saying that "A determines B," or that "B is a function of A," or that "A functionally governs B."

Often, the notions of functionality and functional dependency are expressed briefly by the statement, "If A, then B." It is important to note that the value B must be *unique* for a given value of A, i.e., any given value of A must imply just one and only one value of B, in order for the relationship to qualify for the name "function." (However, this does not necessarily prevent different values of A from implying the same value of B.)

Value of x **("argument," or "A")**	**Value of $y = x^2$** **("the function," or "the result", or "B")**
3	9
4	16
-3	9

- In general, a functional dependency is a relationship among attributes.
- In relational databases, we can have a determinant that governs one other attribute or several other attributes.
- To go back to our mathematical examples for a moment, we could view the situation of functional dependency of several attributes on one determinant as being like having several linked functions that share an argument and can be displayed economically in just one table.
- For example, consider the following table that displays sample values of the algebraic functions y = x2, y = x3, and y = x4.

Value of x	**Value of x^2**	**Value of x^3**	**Value of x^4**
3	9	27	81
4	16	64	256
-3	9	-27	81

Key concept in terms of functional dependencies

A simple example of the functional dependence

Table “Employee”:

Student_ID	**Name**	**Sity**
E1	Mark	New York
E2	Sandra	California
E3	Henry	Paris

All attributes in the table must be functionally dependent on the key.

However, the attribute should be the key to functionally define other attributes.

Key concept is in terms of functional dependencies

Functional dependence can be given the following definition:
In this relation, R attribute A is functionally dependent on B, if the matching of the two tuples that are in R, their values B, they must be matched by the value of A.

Functional relationships are due "many-to-one."

Student_ID	**Sity**	**Subject_ID**	**Exam_scores**
AD0036	London	C1	90
AD0078	New York	C1	88
CC0075	New York	C2	93
CC0097	Florida	C1	75
AD0036	London	C2	87
CC0075	New York	C1	66

The 1st Normal Form (1NF)

Definition:

A table (relation) is in 1NF if

1. There are no duplicated rows in the table.

2. Each cell is single-valued (i.e., there are no repeating groups or arrays).

3. Entries in a column (attribute, field) are of the same kind.

- Note: The order of the rows is immaterial; the order of the columns is immaterial.
- Note: The requirement that there be no duplicated rows in the table means that the table has a key (although the key might be made up of more than one column--even, possibly, of all the columns).

Example1 Consider a table “Projects”

Employee _ID	**Department**	**Department_head _ID**	**project code**	**total time**
E101	Systems	E901	P27 P51 P20	90 101 60
E305	Sales	E906	P27 P22	109 98
E508	Administration	E908	P51 P27	NULL 72

Applying the requirements of 1NF, we obtain the following table:

Employee _ID	**Department**	**Department _head_ID**	**project code**	**total time**
E101	Systems	E901	P27	90
E101	Systems	E901	P51	101
E101	Systems	E901	P20	60
E305	Sales	E906	P27	109
E305	Sales	E906	P22	98
E508	Administration	E908	P51	NULL
E508	Administration	E908	P27	72

Table1 satisfies the definition of 1NF: viz., it has no duplicated rows; each cell is single-valued (i.e., there are no repeating groups or arrays); and all the entries in a given column are of the same kind.

In this table we can see that the key, SSN, functionally determines the other attributes; i.e., a given Social Security Number implies (determines) a particular value for each of the attributes FirstName, LastName, and Major (assuming, at least for the moment, that a student is allowed to have only one major). In the arrow notation: **SSN → FirstName, SSN → LastName, and SSN →Major.**

Table1

Social Security Number	FirstName	LastName	Major
123-45-6789	Jack	Jones	Library and Information Science
222-33-4444	Lynn	Lee	Library and Information Science
987-65-4321	Mary	Ruiz	Pre-Medicine
123-54-3210	Lynn	Smith	Pre-Law
111-33-5555	Jane	Jones	Library and Information Science

- A key attribute will, by the definition of key, uniquely determine the values of the other attributes in a table; i.e., all non-key attributes in a table will be functionally dependent on the key.
- But there may be non-key attributes in a table that determine other attributes in that table.
- Consider the following table2:

Table2

FirstName	**LastName**	**Major**	**Level**
Jack	Jones	LIS	Graduate
Lynn	Lee	LIS	Graduate
Mary	Ruiz	Pre-Medicine	Undergraduate
Lynn	Smith	Pre-Law	Undergraduate
Jane	Jones	LIS	Graduate

- In Table2 the Level attribute can be said to be functionally dependent on the Major attribute.

- Thus we have an example of **an attribute that is functionally dependent on a non-key attribute**.

- This statement is true in the table *per se*, and that is all that the definition of functional dependence requires;

  but the statement also reflects the real-world fact that Library and Information Science is a major that is open only to graduate students and that Pre-Medicine and Pre-Law are majors that are open only to undergraduate students.

The 2nd Normal Form (2NF)

Definition:

A table is in 2NF if it is in 1NF and if all non-key attributes are dependent on all of the key.

Note: Since a partial dependency occurs when a non-key attribute is dependent on only a part of the (composite) key, the definition of 2NF is sometimes phrased as, "A table is in 2NF if it is in 1NF and if it has no partial dependencies."

- **The table is in 2NF if it is in 1NF and every attribute in a row is functionally dependent upon the key to the whole, not only on his part.**

Instructions for converting tables in 2NF:

- Locate and delete the attributes that are functionally dependent only on the part of the key, not the key to the whole.

- Put this attributes in a separate table.

- Group the remaining attributes.

Table2 has another interesting aspect.

- Its key is a composite key, consisting of the paired attributes, FirstName and LastName.

- The Level attribute is functionally dependent on this composite key, of course; but, in addition, Level can be seen to be dependent on only the attribute LastName.

- (This is true because each value of Level is paired with a distinct value of LastName. In contrast, there are two occurrences of the value Lynn for the attribute FirstName, and the two Lynns are paired with different values of Level, so Level is not functionally dependent on FirstName.)

- Thus this table fails to qualify as a 2nd Normal Form table, since the definition of 2NF requires that all non-key attributes be dependent on all of the key.
- (Admittedly, this example of a partial dependency is artificially contrived, but nevertheless it illustrates the problem of partial dependency.)

- We can turn Table 2 into a table in 2NF in an easy way, by adding a column for the Social Security Number, which will then be the natural thing to use as the key.

Example1

With the SSN defined as the key, Table 3 is in 2NF, as you can easily verify.

This illustrates the fact that any table that is in 1NF and has a single-attribute (i.e., a non-composite) key is automatically also in 2NF.

Table 3 still exhibits some problems, however. For example, it contains some repeated information about the LIS-Graduate pairing.

Table 3

SSN	FirstName	LastName	Major	Level
123-45-6789	Jack	Jones	LIS	Graduate
222-33-4444	Lynn	Lee	LIS	Graduate
987-65-4321	Mary	Ruiz	Pre-Medicine	Undergraduate
123-54-3210	Lynn	Smith	Pre-Law	Undergraduate
111-33-5555	Jane	Jones	LIS	Graduate

Anomalies and Normalization

- At this point it is appropriate to note that the main thrust behind the idea of normalizing databases is the avoidance of insertion and deletion anomalies in databases.

How do anomalies relate to normalization?

- The simple answer is that by arranging that the tables in a database are sufficiently normalized (in practice, this typically means to at least the 4th level of normalization), we can ensure that anomalies will not arise in our database.
- Anomalies are difficult to avoid directly, because with databases of typical complexity (i.e., several tables) the database designer can easily overlook possible problems.
- Normalization offers a rigorous way of avoiding unrecognized anomalies.

Turning a Table with Anomalies (Table 3) into Single-Theme Tables

SSN	FirstName	LastName
123-45-6789	Jack	Jones
222-33-4444	Lynn	Lee
987-65-4321	Mary	Ruiz
123-45-4321	Lynn	Smith
111-33-5555	Jane	Jones
999-88-7777	Newton	Gingpoor

Major	Level
LIS	Graduate
Pre-Medicine	Undergraduate
Pre-Law	Undergraduate
Public Affairs	Graduate

SSN	Major
123-45-6789	LIS
222-33-4444	LIS
987-65-4321	Pre-Medicine
123-54-3210	Pre-Law
111-33-5555	LIS

Example2 Consider a table “Project”

Employee _ID	project code	Department	Department_head_ID	Total time
E101	P27	Systems	E901	90
E305	P27	Finance	E909	10
E508	P51	Administration	E908	NULL
E101	P51	Systems	E901	101
E101	P20	Systems	E901	60
E508	P27	Administration	E908	72

Instructions for applying the changes to the table Project in 2NF, we obtain the following table:

Employee _ID	Department	Department_ head_ID
E101	Systems	E901
E305	Finance	E909
E508	Administration	E908

Employee _ID	project code	total time
E101	P27	90
E305	P27	10
E508	P51	NULL
E101	P51	101
E101	P20	60
E508	P27	72

The 3rd Normal Form (3NF)

- **Definition:**
- **A table is in 3NF if it is in 2NF and if it has no transitive dependencies.**

- In order to discuss the 3rd Normal Form, we need to begin by discussing the idea of transitive dependencies.
- In mathematics and logic, a transitive relationship is a relationship of the following form: "If A implies B, and if also B implies C, then A implies C."
- "If A functionally governs B, and if B functionally governs C, then A functionally governs C." In the arrow notation, we have:

$$[(A \rightarrow B) \text{ and } (B \rightarrow C)] \rightarrow (A \rightarrow C)$$

Example1. Consider the table Employees

Employee _ID	Department	Department_head_ID
E101	Systems	E901
E305	Finance	E909
E402	Sales	E906
E508	Administration	E908
E607	Finance	E909
E608	Finance	E909

Applying the guidelines to the transformation of the employee table in 3NF, we obtain the following tables:

Department	**Department_head_ID**
Systems	E901
Sales	E906
Administration	E908
Finance	E909

Employee _ID	**Department**
E101	Systems
E305	Finance
E402	Sales
E508	Administration
E607	Finance
E608	Finance

Example2. **The following table, Table 4, provides an example of how transitive dependencies can occur in a table in a relational database.**

Author Last Name	Author First Name	Book Title	Subject	Collection or Library	Building
Berdahl	Robert	The Politics of the Prussian Nobility	History	PCL General Stacks	Perry-Castañeda Library
Yudof	Mark	Child Abuse and Neglect	Legal Procedures	Law Library	Townes Hall
Harmon	Glynn	Human Memory and Knowledge	Cognitive Psychology	PCL General Stacks	Perry-Castañeda Library
Graves	Robert	The Golden Fleece	Greek Literature	Classics Library	Waggener Hall
Miksa	Francis	Charles Ammi Cutter	Library Biography	Library and Information Science Collection	Perry-Castañeda Library
Hunter	David	Music Publishing and Collecting	Music Literature	Fine Arts Library	Fine Arts Building
Graves	Robert	English and Scottish Ballads	Folksong	PCL General Stacks	Perry-Castañeda Library

- By examining Table 4 we can infer
  - that books dealing with history, cognitive psychology, and folksong are assigned to the PCL General Stacks collection;
  - that books dealing with legal procedures are assigned to the Law Library; that books dealing with Greek literature are assigned to the Classics Library;
  - that books dealing with library biography are assigned to the Library and Information Science Collection (LISC);
  - and that books dealing with music literature are assigned to the Fine Arts Library.
- Further, we can infer
  - that the PCL General Stacks collection and the LISC are both housed in the Perry-Castañeda Library (PCL) building;
  - that the Classics Library is housed in Waggener Hall;
  - and that the Law Library and Fine Arts Library are housed, respectively, in Townes Hall and the Fine Arts Building.

- Thus we see that there is a transitive dependency in Table4: any book that deals with

  - history,

  - cognitive psychology,

  - or library biography will be physically housed in the PCL building (unless it is temporarily checked out to a borrower);

  - any book dealing with legal procedures will be housed in Townes Hall;

  - and so on.

- In short, if we know what subject a book deals with, we also know not only what library or collection it will be assigned to but also what building it is physically housed in.

- What is wrong with having a transitive dependency or dependencies in a table?
  - For one thing, there is duplicated information: from three different rows we can see that the PCL General Stacks are in the PCL building.
  - For another thing, we have possible deletion anomalies: if the Yudof book were lost and its row removed from Table4, we would lose the information that books on legal procedures are assigned to the Law Library and also the information the Law Library is in Townes Hall.
  - As a third problem, we have possible insertion anomalies: if we wanted to add a chemistry book to the table, we would find that Table4 nowhere contains the fact that the Chemistry Library is in Robert A.Welch Hall.
  - As a fourth problem, we have the chance of making errors in updating: a careless data-entry clerk might add a book to the LISC but mistakenly enter Townes Hall in the building column.
- The solution to the problem is, once again, to place the information in Table4 into appropriate single-theme tables.
- Here is one such possible arrangement:

Author Last Name	Author First Name	Book Title
Berdahl	Robert	The Politics of the Prussian Nobility
Yudof	Mark	Child Abuse and Neglect
Harmon	Glynn	Human Memory and Knowledge
Graves	Robert	The Golden Fleece
Miksa	Francis	Charles Ammi Cutter
Hunter	David	Music Publishing and Collecting
Graves	Robert	English and Scottish Ballads

Table 5

Book Title	Subject
The Politics of the Prussian Nobility	History
Child Abuse and Neglect	Legal Procedures
Human Memory and Knowledge	Cognitive Psychology
The Golden Fleece	Greek Literature
Charles Ammi Cutter	Library Biography
Music Publishing and Collecting	Music Literature
English and Scottish Ballads	Folksong

Subject	Collection or Library
History	PCL General Stacks
Legal Procedures	Law Library
Cognitive Psychology	PCL General Stacks
Greek Literature	Classics Library
Library Biography	Library and Information Science Collection
Music Literature	Fine Arts Library
Folksong	PCL General Stacks

Collection or Library	Building
PCL General Stacks	Perry-Castañeda Library
Law Library	Townes Hall
Classics Library	Waggener Hall
Library and Information Science Collection	Perry-Castañeda Library
Fine Arts Library	Fine Arts Building

- You can verify for yourself that none of these tables contains a transitive dependency; hence, all of them are in 3NF (and, in fact, in DKNF).
- We can note in passing that the fact that Table5 contains the first and last names of Robert Graves in two different rows suggests that it might be worthwhile to replace it with two further tables, along the lines of:

Author Last Name	Author First Name	Author Identification Number
Berdahl	Robert	001
Yudof	Mark	002
Harmon	Glynn	003
Graves	Robert	004
Miksa	Francis	005
Hunter	David	006

That would be more economical of storage space than Table 5.

Furthermore, the structure of these Tables lessens the chance of making updating errors.

Author Identification Number	Book Title
001	The Politics of the Prussian Nobility
002	Child Abuse and Neglect
003	Human Memory and Knowledge
004	The Golden Fleece
005	Charles Ammi Cutter
006	Music Publishing and Collecting
004	English and Scottish Ballads

The Boyce-Codd Normal Form (BCNF)

- **Definition: A table is in BCNF if it is in 3NF and if every determinant is a candidate key.**

- The Boyce-Codd Normal Form (BCNF) deals with the anomalies that can occur when a table fails to have the property that every determinant is a candidate key.

- Here is an example, Table_6, that fails to have this property.
- (In Table_6 the SSNs are to be interpreted as those of students with the stated majors and advisers.
- Note that each of students 123-45-6789 and 987-65-4321 has two majors, with a different adviser for each major.)

Example1.

We begin by showing that Table_6 lacks the required property, viz., that every determinant be a candidate key.

What are the determinants in Table_6? One determinant is the pair of attributes, SSN and Major.

Each distinct pair of values of SSN and Major determines a unique value for the attribute, Adviser. Another determinant is the pair, SSN and Adviser, which determines

unique values of the attribute, Major.

Table_6

SSN	Major	Adviser
123-45-6789	Library and Information Science	Dewey
123-45-6789	Public Affairs	Roosevelt
222-33-4444	Library and Information Science	Putnam
555-12-1212	Library and Information Science	Dewey
987-65-4321	Pre-Medicine	Semmelweis
987-65-4321	Biochemistry	Pasteur
123-54-3210	Pre-Law	Hammurabi

- Still another determinant is the attribute, Adviser, for each different value of Adviser determines a unique value of the attribute, Major.
- (These observations about Table_6 correspond to the real-world facts that each student has a single adviser for each of his or her majors, and each adviser advises in just one major.)

- Now we need to examine these three determinants with respect to the question of whether they are candidate keys.
- The answer is that the pair, SSN and Major, is a candidate key, for each such pair uniquely identifies a row in Table6.
- In similar fashion, the pair, SSN and Adviser, is a candidate key.
- But the determinant, Adviser, is not a candidate key, because the value Dewey occurs in two rows of the Adviser column.
- So Table 6 fails to meet the condition that every determinant in it be a candidate key.

- It is easy to check on the anomalies in Table6.
- For example, if student 987-65-4321 were to leave Enormous State University, the table would lose the information that Semmelweis is an adviser for the Pre-Medicine major.
- As another example, Table 6 has no information about advisers for students majoring in history.
- As usual, the solution lies in constructing single-theme tables containing the information in Table 6.
- Here are two tables that will do the job.

SSN	Adviser
123-45-6789	Dewey
123-45-6789	Roosevelt
222-33-4444	Putnam
555-12-1212	Dewey
987-65-4321	Semmelweis
987-65-4321	Pasteur
123-54-3210	Hammurabi

Major	Adviser
Library and Information Science	Dewey
Public Affairs	Roosevelt
Library and Information Science	Putnam
Pre-Medicine	Semmelweis
Biochemistry	Pasteur
Pre-Law	Hammurabi
History	Herodotus

- The basic definition of NF 3 is inadequate and inappropriate for the tables:
  -Having multiple candidate keys.
  -Possible with composite keys.
  -Share overlapping candidate keys.

- To normalize the table under these conditions was proposed normal form Boyce-Codd (BCNF).
  Relation is in BCNF if it is in 3NF and every determinant is a candidate key.
  Instructions to convert a table in BCNF:
  - Locate and remove the overlapping candidate keys.
  - Place a part of the possible key and attribute from which it is functionally dependent in a separate table.
  - Group the remaining items in the table.

Example2. Consider a table “Projects”

Employee _ID	Name	Project_code	Total time
E1	Veronica	P2	48
E2	Paul	P5	100
E3	Igor	P6	15
E4	Akbota	P2	250
E4	Akbota	P5	75
E1	Veronica	P5	40

After applying the changes to the table "Projects" in BCNF, we obtain the following table:

Employee _ID	**Name**
E1	Veronica
E2	Paul
E3	Igor
E4	Akbota

Employee _ID	**Project_code**	**Total time**
E1	P2	48
E2	P5	100
E3	P6	15
E4	P2	250
E4	P5	75
E1	P5	40

Denormalization

- Input in the table intentional redundancy to improve query performance is called denormalization.

- Denormalization is a decision to implement a compromise between performance and consistency of the data.

- Denormalization increases the usable space on the disk.

Product_ID	Description	Price
P1	XXX	20
P2	УУУ	10
P3	ZZZ	12

Order_ID	Product_ID	Amount
101	P1	2
102	P3	1
103	P1	1
104	P2	3
105	P2	3

After applying denormalization table “Orders”, get the following table:

Order_ID	Product_ID	Amount	Sales	Tax	Commodity price
101	P1	2	40	4	44
102	P3	1	12	1,2	13,2
103	P1	1	20	2	22
104	P2	3	30	3	33
105	P2	3	20	2	22

Conclusion

- In this lesson, you learned that:
  There are two approaches to the logical design of the database:
  A "top down"
  Bottom up approach

- Methods of E/R model is a "top down" and normalization is a "bottom up".

- Normalization is used to simplify the table structure.
  Normalization is the design of the tables in accordance with the specified conditions in the form of certain normal forms.

- Table structure is always in a certain normal form.

- The most important and commonly used normal forms are:
  -First Normal Form (1 NF)
  -Second Normal Form (2 NF)
  -Third Normal Form (3 NF)
  -Normal Form Boyce-Codd (BCNF)

- Normalization theory is based on the fundamental concept of functional dependence. Functional relationships are due "many-to-many."

- A table is in 1NF, if each box contains a single value.

- A table is in 2NF, if it is in 1NF and every attribute in the line depends on the whole key, not a part of it.

- A table is in 3NF, if it is in 2NF and every non-key attribute is functionally dependent only on the primary key.

- The basic definition of 3NF is inadequate and is not suitable for tables, at which:
  -There are multiple possible keys.
  -Candidate keys are composite.
  -Candidate keys overlap.

- The relation is in normal form Boyce-Codd (BCNF) if and only if every determinant is a candidate key.

- Intentional redundancy in the input table to improve query performance is called denormalization.

- Denormalization is a compromise between performance and consistency of the data.

- Denormalization increases the usable space on the disk.