

Normalization

Mirunalini.P

SSNCE

April 27, 2022

Table of Contents

- 1 Informal Design Guidelines for Relational Databases
- 2 Formal Guidelines - Normalization
 - First Normal Form
 - Second Normal Form
 - Third Normal Form
- 3 Normalization - Example
- 4 Reference

Session Objective

- Formal and Informal measures for database design
- Normalization-An introduction
- Nonloss decomposition
- Normal Forms [NF]
 - First Normal Form – 1NF
 - Second Normal Form – 2NF
 - Third Normal Form – 3NF

Informal Design Guidelines for Relational Databases

- Making sure the semantics of the attributes is clear in the schema
- Reducing the redundant information in tuples
- Reducing the Null Values in tuples
- Disallowing the possibility of generating spurious tuples

Relational Database Design by ER-Relational Mapping

Formal concepts are functional dependencies and normal forms

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)

GUIDELINE 1: Informally, each tuple in a relation should represent one entity or relationship instance.

- Attributes of different entities (EMPLOYEEs, DEPARTMENTs, PROJECTs) should not be mixed in the same relation
- Only foreign keys should be used to refer to other entities
- Entity and relationship attributes should be kept apart as much as possible.
- Relation corresponding to multiple entity types arises semantic ambiguities.

Bottom Line: Design a schema that is easy to explain and relation and the semantics of the attributes should be easy to interpret.

Semantics of the Relation Attributes

Figure 10.3

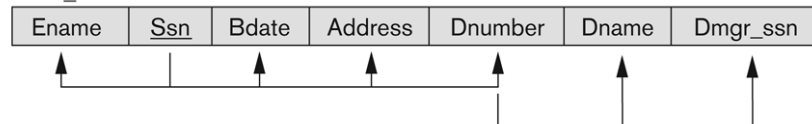
Two relation schemas suffering from update anomalies.

(a) EMP_DEPT and
(b) EMP_PROJ.

c

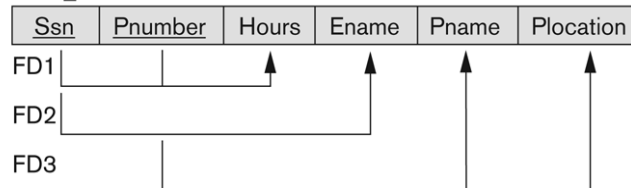
(a)

EMP_DEPT



(b)

EMP_PROJ



Semantics of the Relation Attributes

Figure 10.4

Example states for EMP_DEPT and EMP_PROJ resulting from applying NATURAL JOIN to the relations in Figure 10.2. These may be stored as base relations for performance reasons.

Redundancy

EMP_DEPT

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555

Redundancy Redundancy

EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation
123456789	1	32.5	Smith, John B.	ProductX	Bellaire
123456789	2	7.5	Smith, John B.	ProductY	Sugarland
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford

Redundant Information in Tuples and Update Anomalies

- Information is stored redundantly : Wastage of storage space
- Storing natural joins of base relations leads to additional problems referred as **update anomalies**
- Update anomalies are classified as :
 - Insertion anomalies
 - Deletion anomalies
 - Modification anomalies

Redundant Information in Tuples and Insert Anomalies

**EMP_Dept(SSN,Ename,Bdate,Address,Dnumber,
Dname,Dmgr_ssn)**

Insert Anomaly:

- To insert a new employee tuple, include either the attribute value for dept or null values if not assigned.
- Update with correct info if already inserted else leads to consistency problem
- Difficult to insert a new department without employees, if null values inserted leads to entity integrity constraint violation

Redundant Information in Tuples and Update Anomalies

EMP_PROJ(Emp_No, Proj_No, Ename, Pname, No_hours)

- Update Anomaly: Changing the name of project number P1 from “Billing” to “Customer-Accounting” may cause this update to be made for all 100 employees working on project P1.
- if not correctly updated leads to inconsistency problem

EMP_PROJ(Emp_No, Proj_No, Ename, Pname, No_hours)

Delete Anomaly:

- When a project is deleted, it will result in deleting all the employees who work on that project.
- Alternately, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

Guideline to Redundant Information in Tuples and Update Anomalies

GUIDELINE 2:

- Design a schema that does not suffer from the insertion, deletion and update anomalies.
- If there are any anomalies present, then note them so that applications can be made to take them into account.

Null Values in Tuples

Use of nulls:

- Wastage of storage space
- Problem with understanding the semantics of the attributes
- Difficulty in performing selection, aggregation operations and joins.
- Results become unpredictable

Reasons for nulls:

- Attribute not applicable or invalid
- Attribute value unknown (may exist)
- Value known to exist, but unavailable

GUIDELINE 3:

- **Relations should be designed such that their tuples will have as few NULL values as possible**
- **Attributes that are NULL frequently could be placed in separate relations (with the primary key) not in base relation**
- Emp_office(ESsn,Office_Number)

Spurious Tuples

- Bad designs for a relational database may result in spurious results for certain JOIN operations
- Spurious tuples are not valid tuples
- The "lossless join" property is used to guarantee meaningful results for join operations

Spurious Tuples

- Emp_Locs(Ename,Plocation)
- Emp_Proj1(Ssn,Pnumber,Hours,Pname,Plocation)
- Produces bad schema design, if natural join applied results in spurious tuples.
- When join them back we cannot obtain original information.
- This because the Plocation which relates the two relation is neither a foreign key nor a primary key of the relation.

GUIDELINE 4:

- The relations should be designed so that they can be joined using primary and foreign key pairs.
- No spurious tuples should be generated by doing a natural-join of any relations.

Lossless Property

- Must preserve **losslessness** of the corresponding join
- **Lossy decomposition**

Suppose

id	name	yob
1	A	81
2	A	83

 is decomposed into

c

id	name
1	A
2	A

 and

name	yob
A	81
A	83

The decomposed tables when joined, produces

id	name	yob
1	A	81
1	A	83
2	A	81
2	A	83

 with two **spurious tuples**

- Try to preserve **functional dependencies**

Normalization of Relations

- **Normalization:** The process of decomposing unsatisfactory “bad” relations by breaking up their attributes into smaller relations
- **Normal form:** Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form
- Codd proposed 1NF, 2NF, 3NF (Normal Form)
- A stronger definition of 3NF – Boyce-Codd normal form (BCNF)
 - was proposed by Boyce and Codd

Normalization of Relations

- 2NF, 3NF, BCNF based on keys and FDs of a relation schema
- Additional properties may be needed to ensure a good relational design (**lossless join, dependency preservation**)
- Normalization involves decomposing a given relvar into other relvars, and that decomposition is required to be reversible
- Resulting designs after normalization are of high quality and meet the desirable properties.
- The database designers need to normalize to the highest possible normal form (usually up to 3NF or BCNF).

The process of decomposition must also confirm the existence of additional properties:

- ① The lossless join or nonadditive join property – guarantees that the spurious tuples does not occur w.r.t. the relational schemas created after decomposition
- ② The dependency preservation property – ensures that each FD is preserved in the resulting relations

Note that property (1) is extremely important and cannot be sacrificed. Property (2) is less stringent and may be sacrificed.

Definition of Keys - Recap

- A **superkey** of a relation schema $R = A_1, A_2, \dots, A_n$ is a set of attributes S subset-of R with the property that no two tuples t_1 and t_2 in any legal relation state r of R will have $t_1[S] = t_2[S]$
- A **key** K is a superkey with the additional property that removal of any attribute from K will cause K not to be a superkey any more.
- The difference between a key and a superkey is that a key has to be a minimal. if Key $K = \{A_1, A_2, \dots, A_n\}$ of R , then $K - A_i$ is not a key of R for any A_i
- If a relation schema has **more than one key**, each is called a **candidate key**
- One of the candidate keys is arbitrarily designated to be the **primary key**, and the others are called **secondary keys**.

Definition of Keys - Recap

- A **Prime attribute** must be a member of some candidate key
- A **Nonprime** attribute is not a prime attribute - that is, it is not a member of any candidate key.
- Keys are considered from 2NF onwards

First Normal Form


- Domain of an attribute must include only **atomic** (simple, indivisible) values
- The value of any attribute in a tuple must be a single value
- Disallows **composite attributes, multivalued attributes, and nested relations**
- Disallows attributes whose values for an individual tuple are non-atomic

First Normal Form - 1NF

(a)

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	Dlocations



(b)

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	<u>Dlocation</u>
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Figure 10.8

Normalization into 1NF.

(a) A relation schema that is not in 1NF. (b) Example state of relation DEPARTMENT. (c) 1NF version of the same relation with redundancy.

First Normal Form - 1NF (Solution 1)

The domain of the Dlocations contains atomic values but some of the tuples have set of the values.

DEPARTMENT		
DNAME	<u>DNUMBER</u>	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS	
<u>DNUMBER</u>	<u>DLOCATION</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

- Remove DLOCATION attribute
- Place it in a separate relation along with the primary key
- Does not suffer from redundancy

First Normal Form -1NF (Solution 2)

- Expand the key so that there will be separate tuple in the original Department Location
- Dnumber,Dlocation => Primary Key
- Introduces redundancy

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	33344555	Bellaire
Research	5	33344555	Sugarland
Research	5	33344555	Houston
Administration	4	984756112	Stafford
Headquarters	1	888665555	Houston

Table 1: 1NF version of the same relation with redundancy

First Normal Form - 1NF (Solution 3)

- if the maximum number of values is known for the attribute (3 locations for a department)
- Replace Dlocation with Dlocation1, Dlocation2, Dlocation3
- Introduces Null values if fewer values are present.

Dname	Dnumber	Dmgr_ssn	Dloc1	Dloc2	Dloc3
Research	5	33344555	Bellaire	Sland	Houston
Administration	4	984756112	Stafford	Null	Null
Headquaters	1	888665555	Houston	Null	Null

Table 2: 1NF version of the same relation with Null Values

First Normal Form -1NF

(a)

EMP_PROJ		Projs	
Ssn	Ename	Pnumber	Hours

(b)

Ssn	Ename	Pnumber	Hours
123456789	Smith, John B.	1	32.5
		2	7.5
666884444	Narayan, Ramesh K.	3	40.0
453453453	English, Joyce A.	1	20.0
		2	20.0
333445555	Wong, Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya, AliciaJ.	30	30.0
987987987	Jabbar, Ahmad V.	10	35.0
		30	5.0
987654321	Wallace, Jennifer S.	30	20.0
		20	15.0
888665555	Borg, James E.	20	NULL

(c)

EMP_PROJ1	
Ssn	Ename

EMP_PROJ2

Ssn	Pnumber	Hours
-----	---------	-------

Figure 10.9

Normalizing nested relations into 1NF. (a) Schema of the EMP_PROJ relation with a *nested relation* attribute PROJS. (b) Example extension of the EMP_PROJ relation showing nested relations within each tuple. (c) Decomposition of EMP_PROJ into relations EMP_PROJ1 and EMP_PROJ2 by propagating the primary key.

First Normal Form -1NF

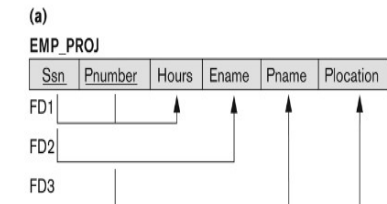
- 1NF also disallows multivalued attributes that are themselves composite leading to Nested relations.
- Emp_Proj - Each tuple represents an Employee and a relation PROJS(Pnumber,Hours) is a nested relation
- Can be represented as Emp_Proj(SSN,Ename,{PROJS(Pnumber,Hours)}) where SSN is the primary key and PNUMBER is a partial primary key.
- To decompose move nested relations into new relations and propagate the primary key into it.
- Decompose as EMP_PROJ into EMP_PROJ1 and EMP_PROJ2 by propagate the primary key along with partial key
- This procedure can be applied recursively to a relation with multiple level nesting to unnest the relation into a set of 1NF relations.

Second Normal Form - 2NF

Uses the concepts of FDs, primary key

- **Prime attribute** - attribute that is member of the primary key K
- **Full functional dependency** – a FD $X \twoheadrightarrow Y$ is a fully FD, if removal of any attribute "A" from X means the FD does not hold any more. ($A \in X$, if $(X - \{A\})$ does not functional determine Y.
- **Partial functional dependency** – a FD $X \twoheadrightarrow Y$ is a partial dependency, if removal of any attribute "A" from X means the FD still holds. ($A \in X$, if $(X - \{A\}) \twoheadrightarrow Y$).
- A relation schema R is in second normal form (2NF) if every non-prime attribute A in R is fully functionally dependent on the primary key OF R.

Second Normal Form – 2NF



C
2NF Normalization

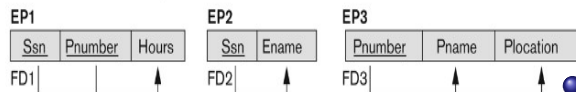


Figure
Normalizing into 2NF an
(a) Normalizing EMP_PROJ int
relations. (b) Normalizing EMP
into 3NF rel

- $SSN, PNUMBER \rightarrow HOURS$ is a full FD since neither $SSN \rightarrow HOURS$ nor $PNUMBER \rightarrow HOURS$ holds

- The dependency $SSN, PNUMBER \rightarrow ENAME$ is partial because $SSN \rightarrow ENAME$ holds.

- FD2 and FD3 violates 2NF because the non_prime key attributes are determined by the part of the primary key viz, SSN AND Pnumber.

- Decompose the table based on partial FD's to achieve 2NF

R can be decomposed into 2NF relations via the process of 2NF normalization

Third Normal Form - 3NF

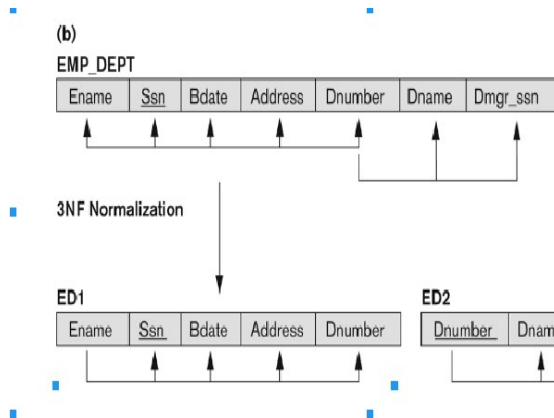
Based on the concept of Transitive Dependency

- **Transitive functional dependency** – a FD $X \rightarrow Z$ that can be derived from two FDs $X \rightarrow Y$ and $Y \rightarrow Z$
- A FD $X \rightarrow Z$ in a R is a **transitive dependency** if there is a set of attributes Y that is **neither a candidate key nor a subset of any key of R** and both $X \rightarrow Y$ and $Y \rightarrow Z$ holds
- A relation schema R is in third normal form (3NF) if it is in 2NF and no non-prime attribute A in R is transitively dependent on the primary key of R.

Third Normal Form - 3NF

- In $X \rightarrow Y$ and $Y \rightarrow Z$, with X as the primary key, we consider this as a problem only if Y is not a candidate key.
- When Y is a candidate key, there is no problem with the transitive dependency.

Third Normal Form - 3NF



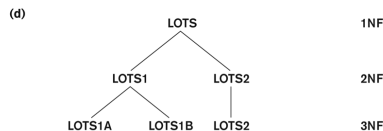
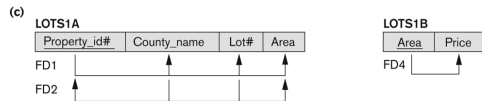
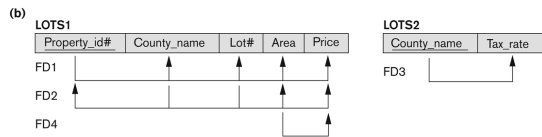
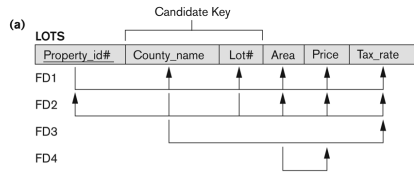
- $SSN \rightarrow DNUMBER$ and $DNUMBER \rightarrow DMGRSSN$ hence $SSN \rightarrow DMGRSSN$ is a transitive FD.
- DNUMBER is neither a candidate key nor a subset of the key
- $DNUMBER \rightarrow DMGRSSN$ violates 3NF
- R can be decomposed into 3NF relations via the process of 3NF normalization

Third Normal Form - 3NF

NOTE:

- In $X \rightarrow Y$ and $Y \rightarrow Z$, with X as the primary key, we consider this a problem only if Y is not a candidate key. When Y is a candidate key, there is no problem with the transitive dependency .
- E.g., Consider EMP (SSN, Emp_No, Salary).
- Here, $SSN \rightarrow Emp_No \rightarrow Salary$ and Emp_No is a **candidate key**.

Normalization - Example



- Two candidate keys:
PROPERTY_ID and
COUNTY_NAME, LOT_No
- The LOTS relation with its
functional dependencies FD1
though FD4.
- Decomposing into the 2NF
relations LOTS1 and LOTS2 .
- Decomposing LOTS1 into the
3NF relations LOTS1A and
LOTS1B .

General Normal Form Definitions

The following more general definitions take into account relations with multiple candidate keys

- A relation schema R is in second normal form (2NF) if every non-prime attribute A in R is fully functionally dependent on every key of R
- A relation schema R is in third normal form (3NF) if whenever a FD $X \rightarrow A$ holds in R , then either:
 - (a) X is a superkey of R , or
 - (b) A is a prime attribute of R

Condition (a) checks the two types of dependencies:

- A non-prime attribute determines another non-prime attribute signals the transitive FD that violates 3NF
- A proper subset of a key of R functionally determines a non-prime attribute – signals the partial FD that violates 2NF

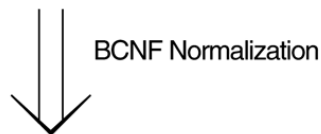
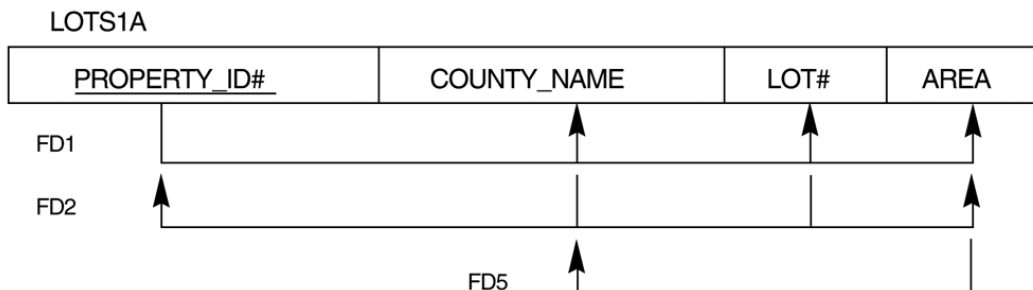
Boyce-Codd Normal Form – BCNF

- Boyce-Codd Normal Form is stricter than 3NF.
- Each normal form is strictly stronger than the previous one
 - Every 2NF relation is in 1NF
 - Every 3NF relation is in 2NF
 - Every BCNF relation is in 3NF
- There exist relations that are in 3NF but not in BCNF
- The goal is to have each relation in BCNF (or 3NF)
- 3NF allows FD's that conform to the clause (b) of general definition of 3NF.
- BCNF disallows them and hence it is stricter definition of a normal form.

Boyce-Codd Normal Form – BCNF

- A relation schema R is in Boyce-Codd Normal Form (BCNF) if whenever an FD $X \rightarrow A$ holds in R, then X is a superkey of R
- FD5 violates BCNF because Area is not superkey of LOTS1A.
- This decomposition loses the functional dependency FD2 because its attributes no longer coexist in the same relation after decomposition.

(a)



LOTS1AX

<u>PROPERTY_ID#</u>	AREA	LOT#
---------------------	------	------

LOTS1AY

<u>AREA</u>	COUNTY_NAME
-------------	-------------

Boyce - Codd Normal Form – BCNF

- Every relation in BCNF is also in 3NF, a relation in 3NF is not necessarily in BCNF.

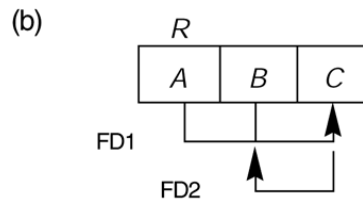


Figure 2: Relation in 3NF but not in BCNF due to the $FD2$

Boyce - Codd Normal Form – BCNF

- A relation TEACH with the following dependencies:
FD1: Student, Course \rightarrow Instructor
FD2:14 Instructor \rightarrow Course
Student, Course is a candidate key for this relation
- Decomposition of this relation schema into two schemas is not straightforward because it may be decomposed into one of the three following possible pairs:
 - R1 (Student, Instructor) and R2(Student, Course)
 - R1 (Course, Instructor) and R2(Course, Student)
 - R1 (Instructor, Course) and R2(Instructor, Student)
- All the three decomposition loose FD1.
- A relation R not in BCNF can be decomposed so as to meet the nonadditive join property

Boyce - Codd Normal Form – BCNF

- The FD $((R1 \cap R2) - \rightarrow (R1 - R2))$ is in F^+
- The FD $((R1 \cap R2) - \rightarrow (R2 - R1))$ is in F^+
if we apply the test only third decomposition meets the test
 $R1 \cap R2$ is Instructor
 $R1 - R2$ is course
- Let R be the relation not in BCNF, let $X \subseteq R$, and let $X - \rightarrow A$ be the FD that causes a violation of BCNF. R may be decomposed into two relations:
 - a. $R - A$
 - b. XA
- If either $R - A$ or XA is not in BCNF, repeat the process.



Fundamentals of Database systems 7th Edition by Ramez Elmasri.