COSC 3380
Design of Database Systems

Functional Dependencies and Normalization for Relational Databases

March 27, 2024

Guideline 1

- Design relation schema so that it is easy to explain its real-world meaning
- Do not combine attributes from multiple entity types and relationship types into a single relation

Guideline 2

- Design base relation schemas so that no update anomalies are present in the relations
- If any anomalies are present:
  - Note them clearly
  - Make sure that the programs that update the database will operate correctly
  - Use triggers or stored procs for automatic updates

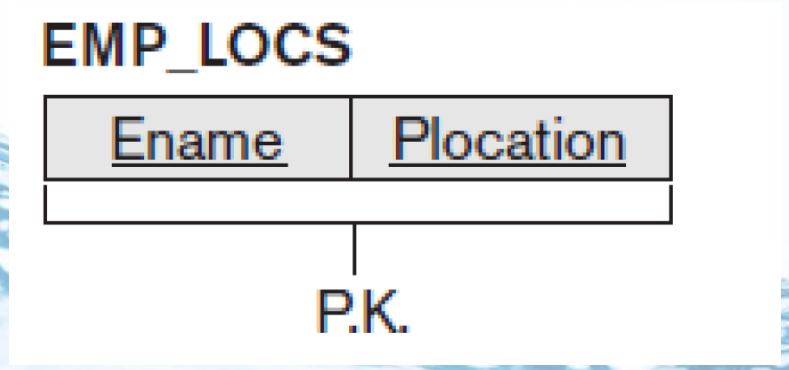
Guideline 3

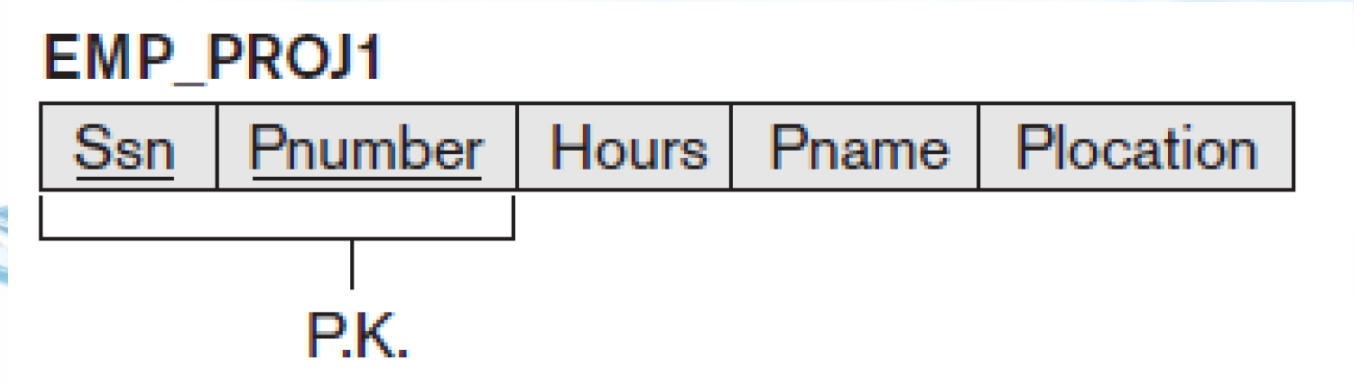
- Avoid placing attributes in a base relation whose values may frequently be NULL
- If NULLs are unavoidable:
  - Make sure that they apply in exceptional cases only, not to a majority of tuples

Guideline 4

- Design relation schemas to be joined with equality conditions on attributes that are appropriately related
  - Guarantees that no spurious tuples are generated
- Avoid relations that contain matching attributes that are not (foreign key, primary key) combinations

Generation of spurious tuples





Generation of spurious tuples

EMP_LOCS

Ename	Plocation
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin T.	Houston
Wong, Franklin T.	Stafford
Zelaya, Alicia J.	Stafford
Jabbar, Ahmad V.	Stafford
Wallace, Jennifer S.	Stafford
Wallace, Jennifer S.	Houston
Borg, James E.	Houston

EMP_PROJ1

Ssn	Pnumber	Hours	Pname	Plocation
123456789	1	32.5	ProductX	Bellaire
123456789	2	7.5	ProductY	Sugarland
666884444	3	40.0	ProductZ	Houston
453453453	1	20.0	ProductX	Bellaire
453453453	2	20.0	ProductY	Sugarland
333445555	2	10.0	ProductY	Sugarland
333445555	3	10.0	ProductZ	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston
999887777	30	30.0	Newbenefits	Stafford
999887777	10	10.0	Computerization	Stafford
987987987	10	35.0	Computerization	Stafford
987987987	30	5.0	Newbenefits	Stafford
987654321	30	20.0	Newbenefits	Stafford
987654321	20	15.0	Reorganization	Houston
888665555	20	NULL	Reorganization	Houston

Generation of spurious tuples – NATURAL JOIN

	Ssn	Pnumber	Hours	Pname	Plocation	Ename
	123456789	1	32.5	ProductX	Bellaire	Smith, John B.
*	123456789	1	32.5	ProductX	Bellaire	English, Joyce A.
	123456789	2	7.5	ProductY	Sugarland	Smith, John B.
*	123456789	2	7.5	ProductY	Sugarland	English, Joyce A.
*	123456789	2	7.5	ProductY	Sugarland	Wong, Franklin T.
	666884444	3	40.0	ProductZ	Houston	Narayan, Ramesh K.
*	666884444	3	40.0	ProductZ	Houston	Wong, Franklin T.
*	453453453	1	20.0	ProductX	Bellaire	Smith, John B.
	453453453	1	20.0	ProductX	Bellaire	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Smith, John B.
	453453453	2	20.0	ProductY	Sugarland	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Wong, Franklin T.
*	333445555	2	10.0	ProductY	Sugarland	Smith, John B.
*	333445555	2	10.0	ProductY	Sugarland	English, Joyce A.
	333445555	2	10.0	ProductY	Sugarland	Wong, Franklin T.
*	333445555	3	10.0	ProductZ	Houston	Narayan, Ramesh K.
	333445555	3	10.0	ProductZ	Houston	Wong, Franklin T.
	333445555	10	10.0	Computerization	Stafford	Wong, Franklin T.
*	333445555	20	10.0	Reorganization	Houston	Narayan, Ramesh K.
	333445555	20	10.0	Reorganization	Houston	Wong, Franklin T.

Guideline 4

- Design relation schemas to be joined with equality conditions on attributes that are appropriately related
  - Guarantees that no spurious tuples are generated
- Avoid relations that contain matching attributes that are not (foreign key, primary key) combinations

Recap: Informal Design Guidelines

- Design relation schema so that it is easy to explain its real-world meaning
- Do not combine attributes from multiple entity types and relationship types into a single relation

- Design base relation schemas so that no update anomalies are present in the relations
- If any anomalies are present:
  - Note them clearly
  - Make sure that the programs that update the database will operate correctly
  - Use triggers or stored procs for automatic updates

- Avoid placing attributes in a base relation whose values may frequently be NULL
- If NULLs are unavoidable:
  - Make sure that they apply in exceptional cases only, not to a majority of tuples

- Design relation schemas to be joined with equality conditions on attributes that are appropriately related
  - Guarantees that no spurious tuples are generated
- Avoid relations that contain matching attributes that are not (foreign key, primary key) combinations

Functional Dependency

- Constraint between two sets of attributes from the database

Definition. A **functional dependency**, denoted by $X \rightarrow Y$, between two sets of attributes X and Y that are subsets of R specifies a *constraint* on the possible tuples that can form a relation state r of R. The constraint is that, for any two tuples t_1 and t_2 in r that have $t_1[X] = t_2[X]$, they must also have $t_1[Y] = t_2[Y]$.

- Property of the semantics or meaning of the attributes

Functional Dependency

- Y component of a tuple depends on or is determined by values of X component
- Values of X component of a tuple uniquely/functionally determine the values of Y component
- Y is functionally dependent on X

$$X \rightarrow Y$$

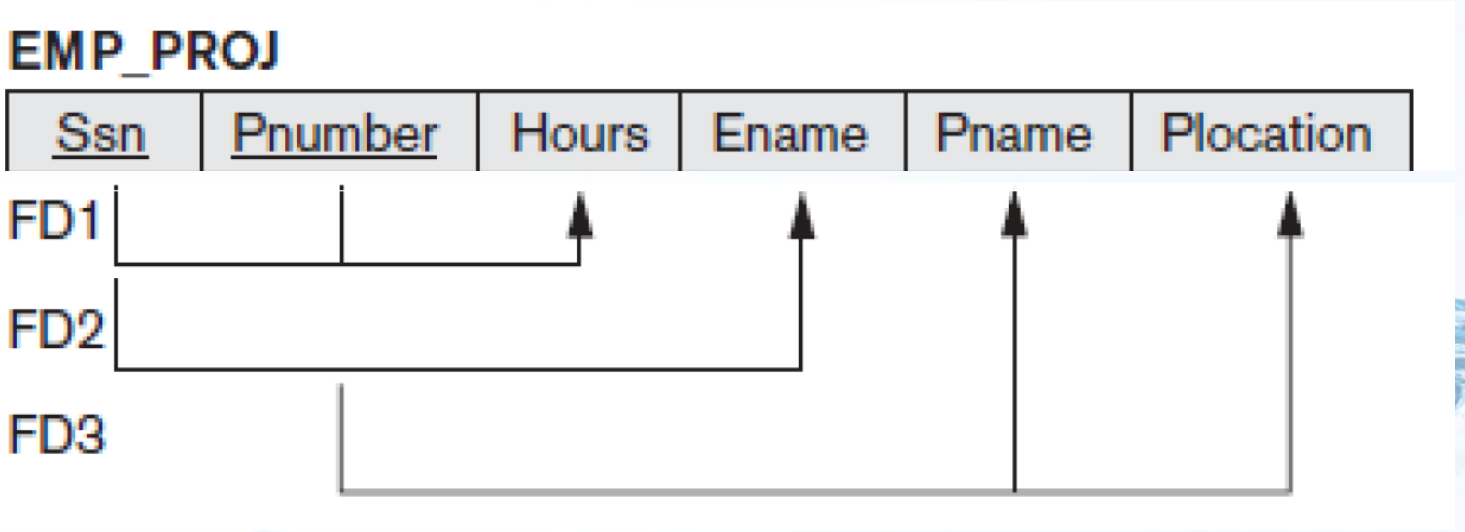
Functional Dependency

- If X is a candidate key of relation, R
  - Cannot be more than one tuple with a given X value
  - X →Y for any subset of attributes Y in R
  - With X as candidate key, X→R
  - X→Y in R does not mean Y→X in R

Functional Dependency

- X → Y holds if whenever two tuples have the same value for X, they *must have* the same value for Y
  - For any two tuples, t_1 and t_2, in any relation instance r(R): If $t_1[X]=t_2[X]$, *then* $t_1[Y]=t_2[Y]$
- X → Y in R specifies a *constraint* on all relation instances r(R)
- Can be displayed graphically on a relation schema
- FDs are derived from the real-world constraints on the attributes

Functional Dependency



a. Ssn → Ename

b. Pnumber →{Pname, Plocation}

c. {Ssn, Pnumber} → Hours

Functional Dependency

- A FD is a property of the attributes in the schema R
- The constraint must hold on *every* relation instance r(R)
- If K is a key of R, then K functionally determines all attributes in R
  - As we never have two distinct tuples with t1[K]=t2[K]

Defining FDs from instances

- To define the FDs, we need to understand the meaning of the attributes involved and the relationship between them.
- Given the instance (population) of a relation, all we can conclude is that an FD *may exist* between certain attributes.
- What we can definitely conclude is – that certain FDs *do not exist* because there are tuples that show a violation of those dependencies.

Functional Dependency

TEACH

Teacher	Course	Text
Smith	Data Structures	Bartram
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

TEXT → COURSE

TEACHER → COURSE

What FDs may exist?

- A relation *R*(A, B, C, D) with its extension.

A	B	C	D
a1	b1	c1	d1
a1	b2	c2	d2
a2	b2	c2	d3
a3	b3	c4	d3

- Which FDs *may exist* in this relation?

Normalization of Relations

- Normal forms are based on functional dependencies amongst the attributes of a relation
- Take a relation schema through a series of tests
  - Certify whether it satisfies a certain normal form
  - Proceed in a top-down fashion

Normalization of Relations

- Analyze a given relation schema based on their FDs and primary keys
  - Minimize redundancy
  - Minimize insertion/deletion/update anomalies
- Filtering/purification process to achieve better quality design

Normalization of Relations

Definition. The **normal form** of a relation refers to the highest normal form condition that it meets, and hence indicates the degree to which it has been normalized.

Objectives of normalization:

1. To free the collection of relations from undesirable insertion, update and deletion dependencies;
2. To reduce the need for restructuring the collection of relations, as new types of data are introduced, and thus increase the life span of application programs;
3. To make the relational model more informative to users;
4. To make the collection of relations neutral to the query statistics, where these statistics are liable to change as time goes by.

— E.F. Codd, "Further Normalization of the Data Base Relational Model"[9]

Normalization of Relations

- Properties of a normalized relational schema
  - Nonadditive/lossless join property
    - Ensures spurious tuples are not created with schemas created after decomposition
    - Extremely critical and must be achieved at any cost
  - Dependency preservation property
    - Each FD represented in some relation resulting after decomposition
    - Desirable but sometimes sacrificed due to other factors

Practical Use of Normal Forms

- Normalization carried out in practice
  - Resulting designs are of high quality and meet the requirements/goals
  - Pays particular attention to normalization only up to 3NF, BCNF (3.5NF), or at most 4NF
- Do not need to normalize to the highest possible normal form!
- Denormalization

Definition. Denormalization is the process of storing the join of higher normal form relations as a base relation, which is in a lower normal form.

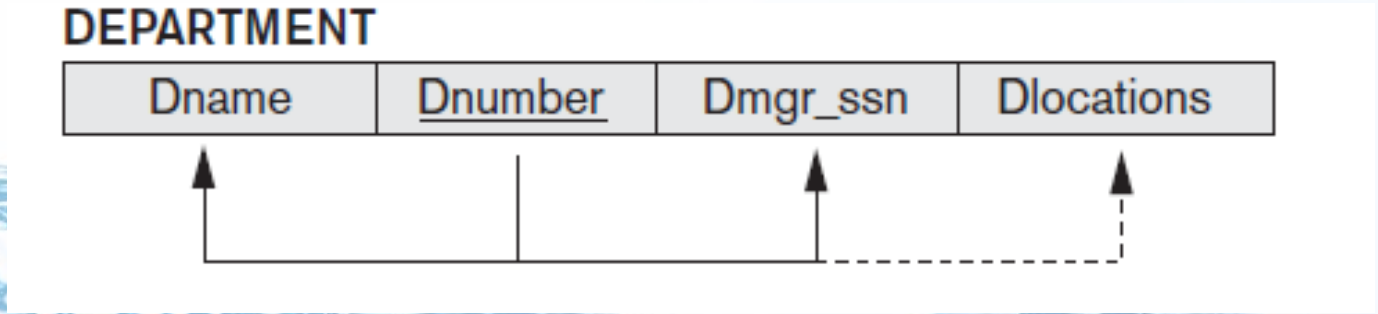
Recall time!

- Candidate Key
- Primary Key
- Super Key
- 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.

First Normal Form

- Domain of an attribute must include only **atomic** (**simple, indivisible**) values
- Value of any attribute in a tuple must be a single value
- Disallows a set of values as an attribute value in a single tuple
  - Disallows composite attributes
  - Disallows multivalued attributes
  - Disallows relations within relations
- Most RDBMSs allow only those relations to be defined that are in 1NF

First Normal Form

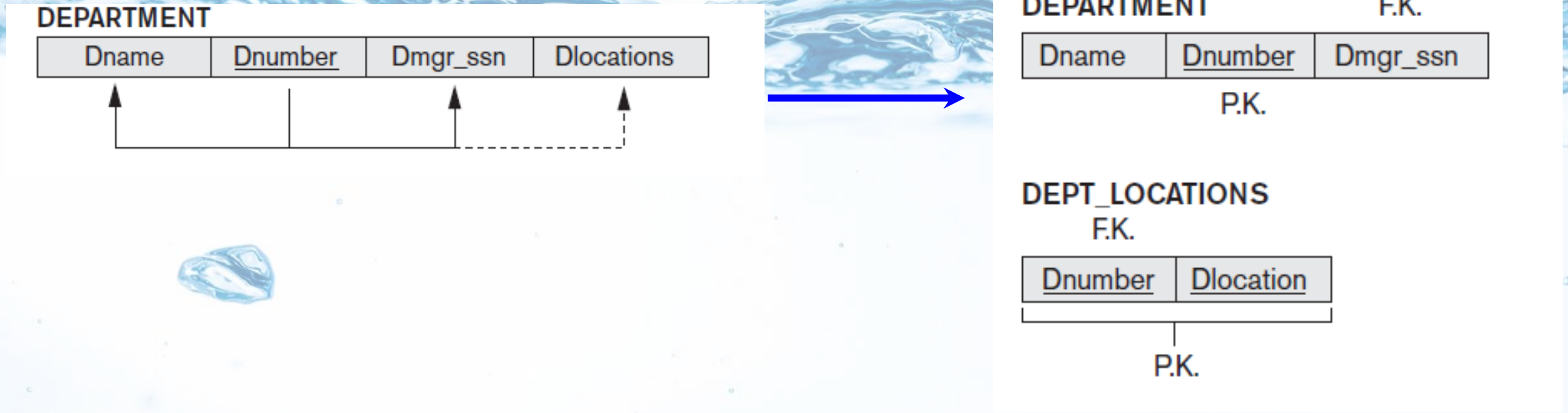


DEPARTMENT

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

Techniques to achieve 1NF

- Remove attribute that violates 1NF and place in separate relation



Techniques to achieve 1NF

- Expand the key
  - Introduces redundancy

DEPARTMENT

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

Techniques to achieve 1NF

- Use several atomic attributes
  - (DLocation1, DLocation2, DLocation3)
  - NULL Values
  - Querying and ordering problems
  - Scalability?