# Database Systems

Instructor: Bilal Khalid Dar

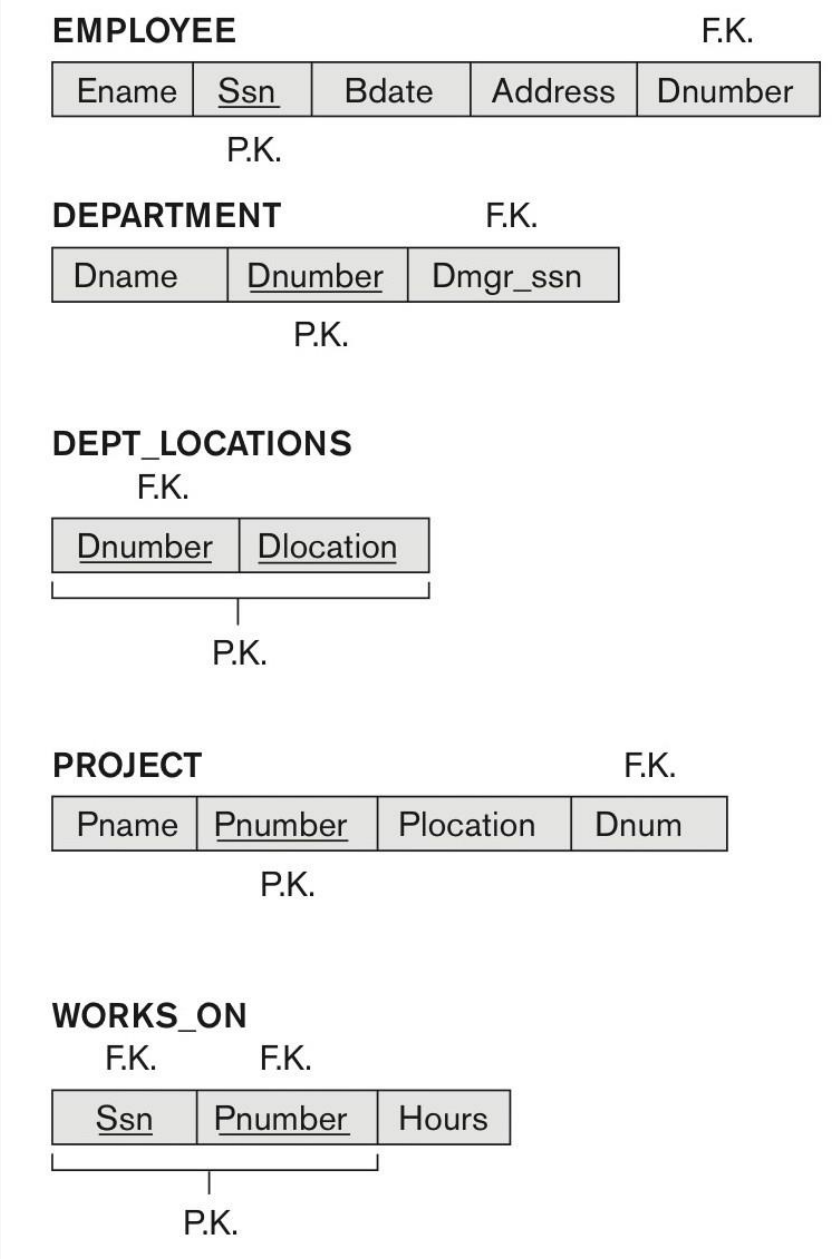
# Agenda

- Normalization

# Normalization Agenda

- We first discuss informal guidelines for good relational design
- Then we discuss formal concepts of functional dependencies and normal forms
  - - 1NF (First Normal Form)
  - - 2NF (Second Normal Form)
  - - 3NF (Third Normal Form)
  - - BCNF (Boyce-Codd Normal Form)

# A simplified COMPANY relational database schema



# Guidelines for good relational design

# Semantics of the Relational Attributes must be clear

- GUIDELINE 1: Informally, each tuple in a relation should represent one entity or relationship instance. (Applies to individual relations and their attributes).
  - 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.
- Bottom Line: *Design a schema that can be explained easily relation by relation. The semantics of attributes should be easy to interpret.*

# Redundant Information in Tuples and Update Anomalies

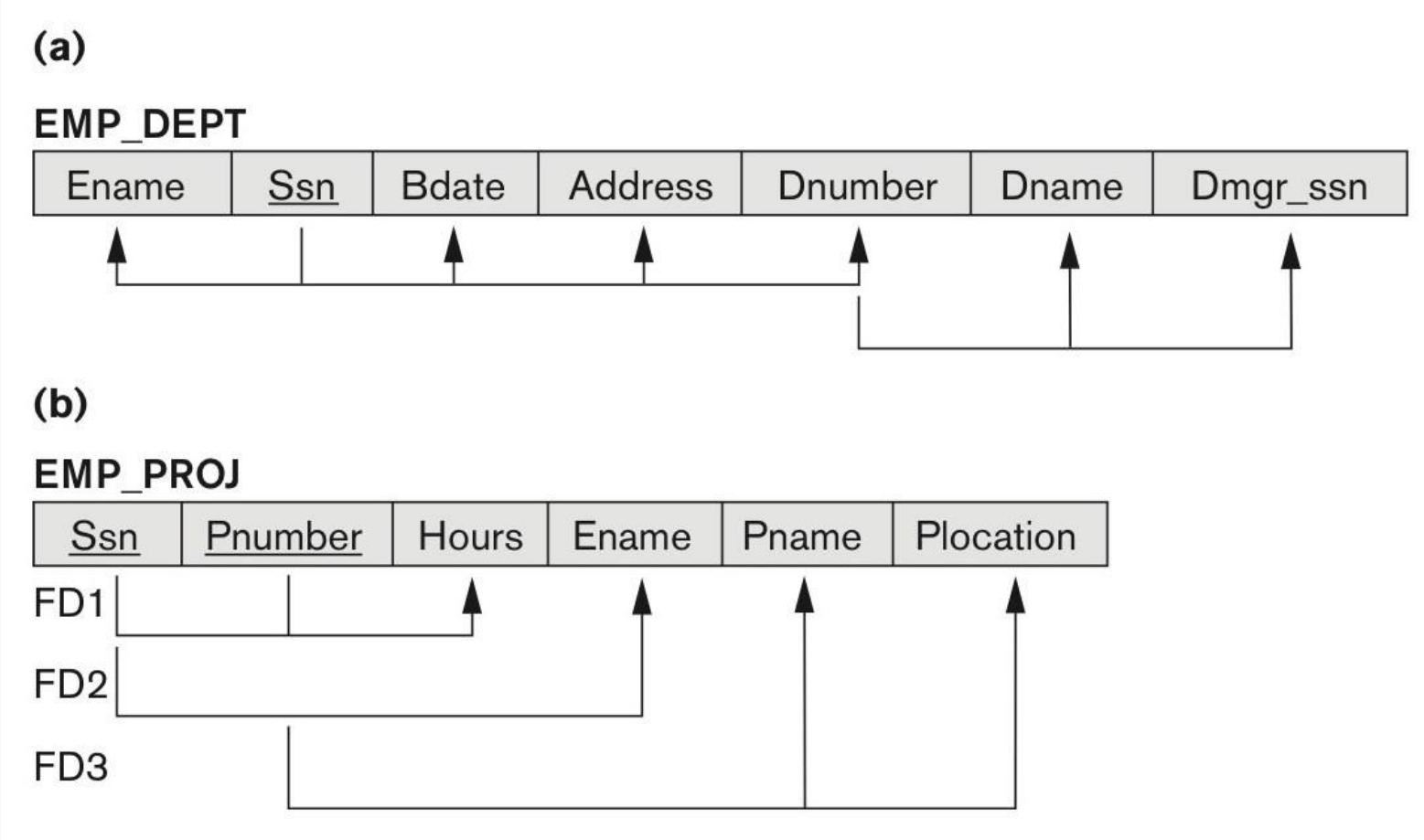
- Information is stored redundantly
  - Wastes storage
  - Causes problems with update anomalies
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies

# EXAMPLE OF AN UPDATE ANOMALY

- Consider the relation:
  - EMP_PROJ(Emp#, Proj#, 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.

# Two relation schemas suffering from update anomalies

**Figure 14.3**
Two relation schemas suffering from update anomalies. (a) EMP_DEPT and (b) EMP_PROJ.



# Sample states for EMP_DEPT and EMP_PROJ

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 |

# Sample states for EMP_DEPT and EMP_PROJ

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 |
| 987987987 | 30 | 5.0 | Jabbar, Ahmad V. | Newbenefits | Stafford |
| 987654321 | 30 | 20.0 | Wallace, Jennifer S. | Newbenefits | Stafford |
| 987654321 | 20 | 15.0 | Wallace, Jennifer S. | Reorganization | Houston |
| 888665555 | 20 | Null | Borg, James E. | Reorganization | Houston |

# EXAMPLE OF AN INSERT ANOMALY

- Consider the relation:
  - EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)
- Insert  Anomaly:
  - Cannot insert a project unless an employee is assigned to it.
- Conversely
  - Cannot insert an employee unless an he/she is assigned to a project.

# EXAMPLE OF A DELETE ANOMALY

- Consider the relation:
  - EMP_PROJ(Emp#, Proj#, 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 for 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

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

# Generation of Spurious Tuples – avoid at any cost

- Bad designs for a relational database may result in erroneous results for certain JOIN operations
- The "lossless join" property is used to guarantee meaningful results for join operations

- GUIDELINE 4:
  - The relations should be designed to satisfy the lossless join condition.
  - No spurious tuples should be generated by doing a natural-join of any relations.

- **Before Normalization**
- Begin with a list of all of the fields that must appear in the database. Think of this as one big table.

– Do not include computed fields

– One place to begin getting this information is from a printed document used by the system.

– Additional attributes besides those for the entities described on the document can be added to the database.

# Some Important Concepts

# Functional Dependencies

## Functional Dependencies

We say an attribute, B, has a *functional dependency* on another attribute, A, if for any two records, which have the same value for A, then the values for B in these two records must be the same. We illustrate this as:

A → B

**Example**: Suppose we keep track of employee email addresses, and we only track one email address for each employee. Suppose each employee is identified by their unique employee number. We say there is a functional dependency of email address on employee number:

employee number → email address

# Functional Dependencies

| EmpNum | EmpEmail | EmpFname | EmpLname |
|---|---|---|---|
| 123 | jdoe@abc.com | John | Doe |
| 456 | psmith@abc.com | Peter | Smith |
| 555 | alee1@abc.com | Alan | Lee |
| 633 | pdoe@abc.com | Peter | Doe |
| 787 | alee2@abc.com | Alan | Lee |

If EmpNum is the PK then the FDs:

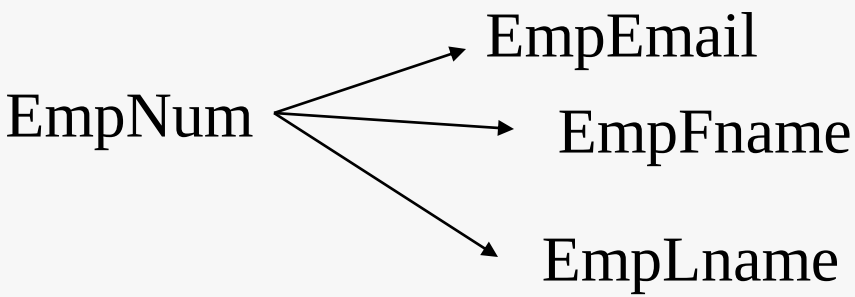
EmpNum → EmpEmail
EmpNum → EmpFname
EmpNum → EmpLname

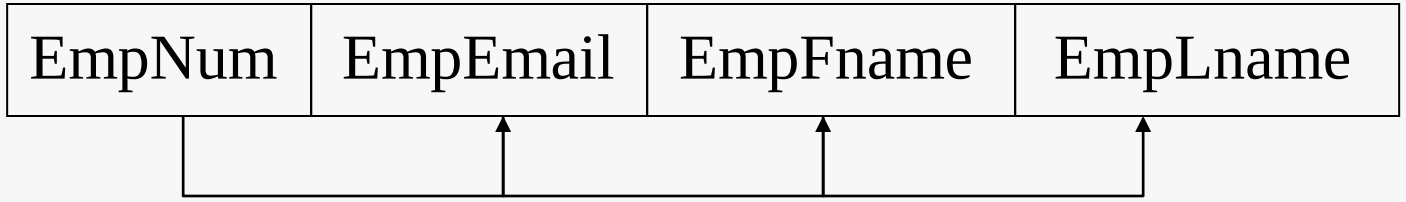
must exist.

# Functional Dependencies

EmpNum → EmpEmail
EmpNum → EmpFname
EmpNum → EmpLname

*3 different ways you might see FDs depicted*



# Determinant

Functional Dependency

EmpNum → EmpEmail

Attribute on the LHS is known as the ***determinant***

- EmpNum is a determinant of EmpEmail

# Transitive dependency

**Transitive dependency**

Consider attributes A, B, and C, and where

A → B and B → C.

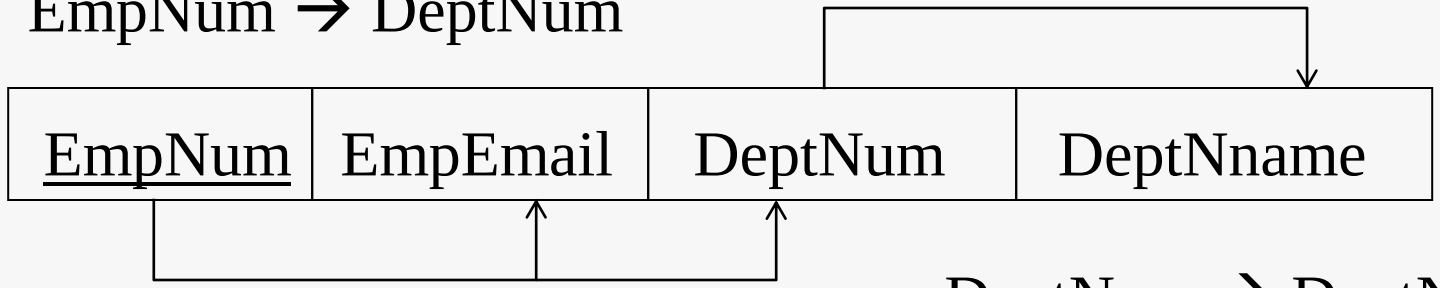
Functional dependencies are transitive, which means that we also have the functional dependency

A → C

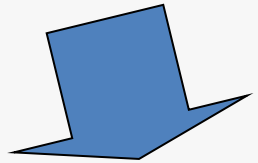
We say that C is transitively dependent on A through B.

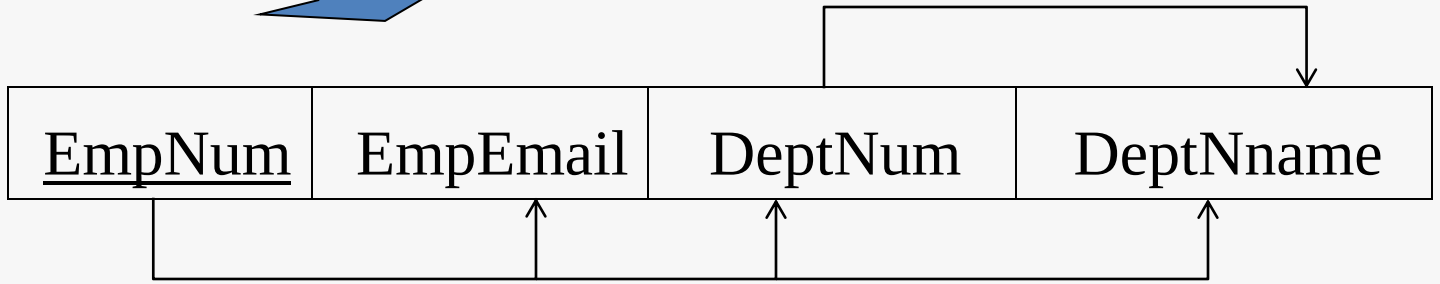
# Transitive dependency

EmpNum → DeptNum




DeptNum → DeptName






DeptName is *transitively dependent* on EmpNum via DeptNum

EmpNum → DeptName

# Normalization

## Definition

- This is the process which allows you to winnow out redundant data within your database.
  - The results of a well executed normalization process are the same as those of a well planned E-R model
- This involves restructuring the tables to successively meeting higher forms of Normalization.
- A properly normalized database should have the following characteristics
  - Scalar values in each fields
  - Absence of redundancy.
  - Minimal use of null values.
  - Minimal loss of information.

(Note: Winnow(Webster): To get rid of / eliminate inferior material

# Normalization

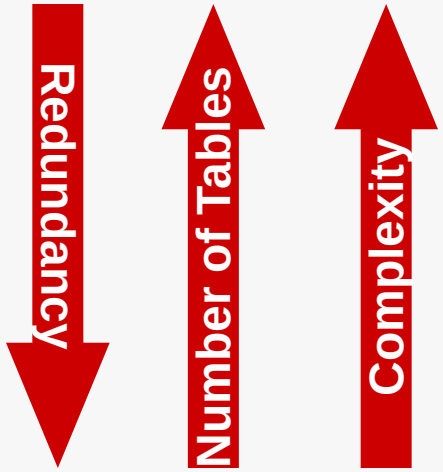
## Process

- Eliminate Repeating Groups
  - Make a separate table for each set of related attributes and give each table a primary key.
- Eliminate Redundant Data
  - If an attribute depends on only part of a multivalued key, remove it to a separate table.
- Eliminate Columns not dependent on key
  - If attributes do not contribute to a description of the key, remove them to a separate table.
- Isolate Independent multiple relationships
  - No table may contain two or more 1:n or n:m relationships that are not directly related.
- Isolate Semantically Related Multiple Relationships
  - There may be practical constraints on information that justify separating logically related many-to-many relationships

# Normalization

## Levels

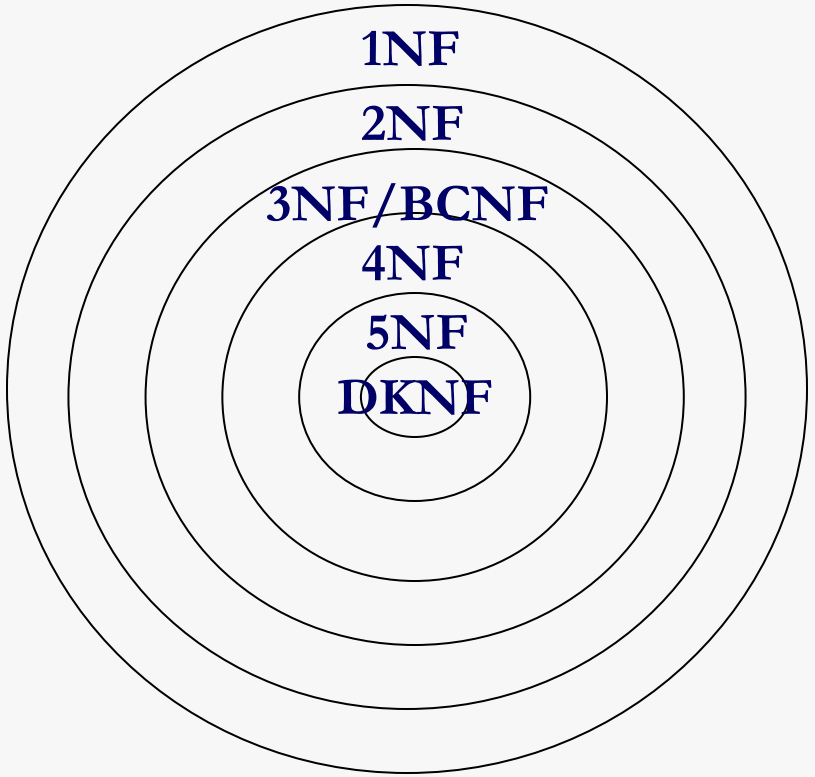
- Levels of normalization based on the amount of redundancy in the database.
- Relational theory defines a number of structure conditions called Normal Forms that assure that certain data anomalies do not occur in a database.
- Various levels of normalization are:
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce-Codd Normal Form (BCNF)
  - Fourth Normal Form (4NF)
  - Fifth Normal Form (5NF)
  - Domain Key Normal Form (DKNF)




Most databases should be 3NF or BCNF in order to avoid the database anomalies.

# Normalization

## Levels




| | |
|---|---|
| **1NF** | *Keys; No repeating groups or multi-valued* |
| **2NF** | *No partial dependencies* |
| **3NF** | *No transitive dependencies* |
| **BCNF** | *Determinants are candidate keys* |
| **4NF** | *No multivalued dependencies* |
| **5NF** | *No multivalued dependencies* |
| **4NF** | *No multivalued dependencies* |

Each higher level is a subset of the lower level

# Normalization

## First Normal Form (1NF)

A table is considered to be in 1NF if all the fields contain only scalar values (as opposed to list of values).

### Example (Not 1NF)

| ISBN | Title | AuName | AuPhone | PubName | PubPhone | Price |
|---|---|---|---|---|---|---|
| 0-321-32132-1 | Balloon | Sleepy, Snoopy, Grumpy | 321-321-1111, 232-234-1234, 665-235-6532 | Small House | 714-000-0000 | $34.00 |
| 0-55-123456-9 | Main Street | Jones, Smith | 123-333-3333, 654-223-3455 | Small House | 714-000-0000 | $22.95 |
| 0-123-45678-0 | Ulysses | Joyce | 666-666-6666 | Alpha Press | 999-999-9999 | $34.00 |
| 1-22-233700-0 | Visual Basic | Roman | 444-444-4444 | Big House | 123-456-7890 | $25.00 |

Author and AuPhone columns are not scalar

# Normalization

## 1NF: Decomposition

1. Place all items appearing in the repeating group in a new table
2. Designate a primary key for each new table produced.
3. Create a relationship between the two tables
   - For 1:N relation duplicate the P.K. from 1 side to many side
   - For M:N relation create a new table with P.K. from both tables

## Example (1NF)

| ISBN | Title | PubName | PubPhone | Price |
|---|---|---|---|---|
| 0-321-32132-1 | Balloon | Small House | 714-000-0000 | $34.00 |
| 0-55-123456-9 | Main Street | Small House | 714-000-0000 | $22.95 |
| 0-123-45678-0 | Ulysses | Alpha Press | 999-999-9999 | $34.00 |
| 1-22-233700-0 | Visual Basic | Big House | 123-456-7890 | $25.00 |

| ISBN | AuName | AuPhone |
|---|---|---|
| 0-321-32132-1 | Sleepy | 321-321-1111 |
| 0-321-32132-1 | Snoopy | 232-234-1234 |
| 0-321-32132-1 | Grumpy | 665-235-6532 |
| 0-55-123456-9 | Jones | 123-333-3333 |
| 0-55-123456-9 | Smith | 654-223-3455 |
| 0-123-45678-0 | Joyce | 666-666-6666 |
| 1-22-233700-0 | Roman | 444-444-4444 |

# Normalization

## Functional Dependencies

1. If one set of attributes in a table determines another set of attributes in the table, then the second set of attributes is said to be functionally dependent on the first set of attributes.

### Example 1

| ISBN | Title | Price |
|---|---|---|
| 0-321-32132-1 | Balloon | $34.00 |
| 0-55-123456-9 | Main Street | $22.95 |
| 0-123-45678-0 | Ulysses | $34.00 |
| 1-22-233700-0 | Visual Basic | $25.00 |

**Table Scheme: {ISBN, Title, Price}**

**Functional Dependencies: {ISBN} → {Title}**

**{ISBN} → {Price}**

# Normalization

## Functional Dependencies

### Example 2

| PubID | PubName | PubPhone |
|---|---|---|
| 1 | Big House | 999-999-9999 |
| 2 | Small House | 123-456-7890 |
| 3 | Alpha Press | 111-111-1111 |

**Table Scheme: {PubID, PubName, PubPhone}**
**Functional Dependencies: {PubId} → {PubPhone}**
**{PubId} → {PubName}**

### Example 3

| AuID | AuName | AuPhone |
|---|---|---|
| 1 | Sleepy | 321-321-1111 |
| 2 | Snoopy | 232-234-1234 |
| 3 | Grumpy | 665-235-6532 |
| 4 | Jones | 123-333-3333 |
| 5 | Smith | 654-223-3455 |
| 6 | Joyce | 666-666-6666 |
| 7 | Roman | 444-444-4444 |

**Table Scheme: {AuID, AuName, AuPhone}**
**Functional Dependencies: {AuId} → {AuPhone}**
**{AuId} → {AuName}**

# 1NF Repeat

# First Normal Form

**First Normal Form**

We say a relation is in **1NF** if all values stored in the relation are single-valued and atomic.

1NF places restrictions on the structure of relations. Values must be simple.

# First Normal Form

The following in **not** in 1NF

| **EmpNum** | **EmpPhone** | **EmpDegrees** |
|---|---|---|
| 123 | 233-9876 | |
| 333 | 233-1231 | BA, BSc, PhD |
| 679 | 233-1231 | BSc, MSc |

EmpDegrees is a multi-valued field:

employee 679 has two degrees: *BSc* and *MSc*

employee 333 has three degrees: *BA*, *BSc*, *PhD*

# First Normal Form

| EmpNum | EmpPhone | EmpDegrees |
|---|---|---|
| 123 | 233-9876 | |
| 333 | 233-1231 | BA, BSc, PhD |
| 679 | 233-1231 | BSc, MSc |

To obtain 1NF relations we must, without loss of information, replace the above with two relations - see next slide

# First Normal Form

**Employee**

| EmpNum | EmpPhone |
|---|---|
| 123 | 233-9876 |
| 333 | 233-1231 |
| 679 | 233-1231 |

**EmployeeDegree**

| EmpNum | EmpDegree |
|---|---|
| 333 | BA |
| 333 | BSc |
| 333 | PhD |
| 679 | BSc |
| 679 | MSc |


Thank
You