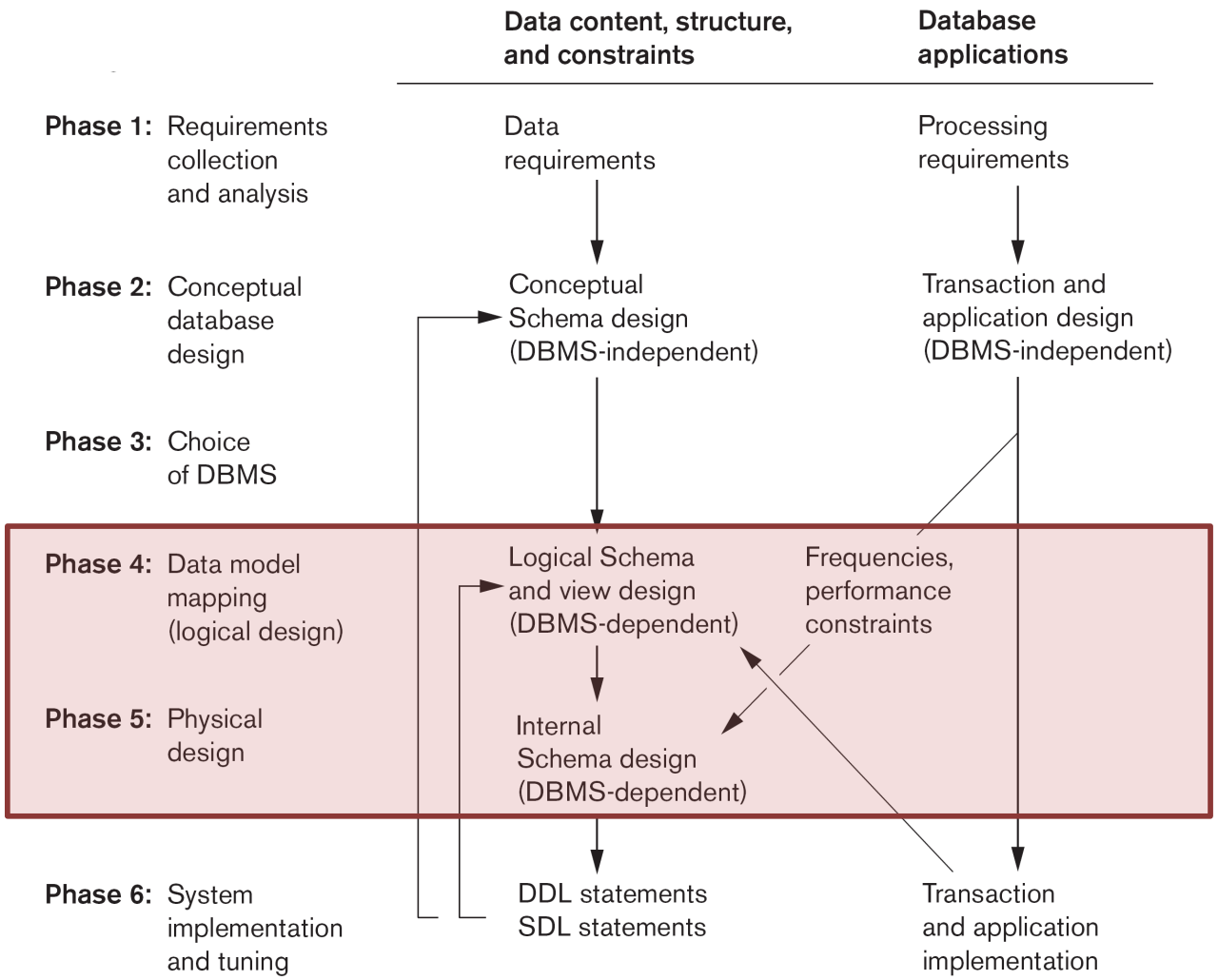
# Normalization

## Lecture 10

# Outline

1. Context
2. Normalization Objectives
3. Functional Dependencies
4. Normal Forms
   - 1NF
   - 2NF
   - 3NF

# Database Design and Implementation Process



# Normalization

- Theory and process by which to evaluate and improve relational database design
- Typically divide larger tables into smaller, less redundant tables
- Spans both logical and physical database design

# Objectives of Normalization

- Make the schema informative
- Minimize information duplication
- Avoid **modification anomalies**
- Disallow **spurious tuples**

Note: during physical tuning we may prioritize query execution speed and thus *denormalize* (e.g. OLTP vs. OLAP)

# Example Schema

**EMPLOYEE**

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

**DEPARTMENT**

| Dname | Dnumber | Dmgr_ssn |
|---|---|---|
| Research | 5 | 333445555 |
| Administration | 4 | 987654321 |
| Headquarters | 1 | 888665555 |

# Straw Man Schema

**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 |

# Make the Schema Informative

- Design a relational schema so that it is easy to explain its meaning
- Do **not** combine attributes from multiple entity types and relationship types into a single relation; semantic ambiguities will result and the relation cannot be easily explained
- Normalized tables, and the relationship between one normalized table and another, mirror real-world concepts and their interrelationships

# Example Schema

## What is this table about?

- Employees? Departments?

**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 |

# Minimize Information Duplication

- Avoid data redundancies

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 |

- Avoid excessive use of NULLs (e.g. fat tables)
  - Wastes space
  - Can make information querying/understanding complicated and error-prone

# Avoid **Modification Anomalies**

An undesired side-effect resulting from an attempt to modify a table [that has not been sufficiently normalized]

Types of updates:

- Insertion
- Update
- Deletion

# Insertion Anomaly

Difficult or impossible to insert a new row

- Add a new employee
  - Unknown manager
  - Typo in department/manager info
- Add a new department
  - Requires at least one employee

**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 |

# Update Anomaly

Updates may result in logical inconsistencies

- Change the department name/manager

**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 |

# Deletion Anomaly

Deletion of data representing certain facts necessitates deletion of data representing completely different facts

- Delete James E. Borg

**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 |

# Disallow **Spurious Tuples**

Avoid relational design that matches attributes across relations that are not (foreign key, primary key) combinations because joining on such attributes may produce invalid tuples

# Example Decomposition

**CAR**

| ID | Make | Color |
|---|---|---|
| 1 | Toyota | Blue |
| 2 | Audi | Blue |
| 3 | Toyota | Red |

**CAR1**

| ID | Color |
|---|---|
| 1 | Blue |
| 2 | Blue |
| 3 | Red |

**CAR2**

| Make | Color |
|---|---|
| Toyota | Blue |
| Audi | Blue |
| Toyota | Red |

# Natural Join

| ID | Make | Color |
|---|---|---|
| 1 | Toyota | Blue |
| 1 | Audi | Blue |
| 2 | Toyota | Blue |
| 2 | Audi | Blue |
| 3 | Toyota | Red |

**CAR1**

| ID | Color |
|---|---|
| 1 | Blue |
| 2 | Blue |
| 3 | Red |

**CAR2**

| Make | Color |
|---|---|
| Toyota | Blue |
| Audi | Blue |
| Toyota | Red |

# Additive Decomposition

**CAR**

| ID | Make | Color |
|---|---|---|
| 1 | Toyota | Blue |
| 2 | Audi | Blue |
| 3 | Toyota | Red |

**JOIN**

| ID | Make | Color |
|---|---|---|
| 1 | Toyota | Blue |
| **1** | **Audi** | **Blue** |
| **2** | **Toyota** | **Blue** |
| 2 | Audi | Blue |
| 3 | Toyota | Red |

# Functional Dependency (FD)

In a relation *r*, a set of attributes **Y** is *functionally dependent* upon another set of attributes **X** ($X \rightarrow Y$) iff…

for all pairs of tuples $t_1$ and $t_2$ in *r*…

if $t_1$[**X**]=$t_2$[**X**]…

it MUST be the case that $t_1$[**Y**]=$t_2$[**Y**]

# FD Example (1)

| | StudentID | Year | Class | Instructor |
|---|---|---|---|---|
| $t_1$ | 1 | Sophomore | COMP355 | Wu |
| $t_2$ | 2 | Sophomore | COMP285 | Wu |
| $t_3$ | 3 | Junior | COMP355 | Wu |
| $t_4$ | 3 | Junior | COMP285 | Wu |
| $t_5$ | 2 | Sophomore | COMP355 | Russo |
| $t_6$ | 4 | Sophomore | COMP355 | Russo |

What FDs hold in the current state of this relation?

$$\{StudentID\} \rightarrow \{Year\}$$

$$\{StudentID, Class\} \rightarrow \{Instructor\}$$

# FDs & Keys

- One cannot determine whether FDs hold for *all* relation states unless the meaning of and relationships among the attributes are known

# FD Example (2)

| | StudentID | Year | Class | Instructor |
|---|---|---|---|---|
| $t_1$ | 1 | Sophomore | COMP355 | Wu |
| $t_2$ | 2 | Sophomore | COMP285 | Wu |
| $t_3$ | 3 | Junior | COMP355 | Wu |
| $t_4$ | 3 | Junior | COMP285 | Wu |
| $t_5$ | 2 | Sophomore | COMP355 | Russo |
| $t_6$ | 4 | Sophomore | COMP355 | Russo |

What FDs hold in the current state of this relation?

$$\{StudentID\} \rightarrow \{Year\}$$

$$\{StudentID, Class\} \rightarrow \{Instructor\}$$

Multiple sections?
Co-instruction?
Re-takes?

# FDs & Keys

- One cannot determine whether FDs hold for *all* relation states unless the meaning of and relationships among the attributes are known
  - These are the "data dependencies" foreshadowed in Lecture 2 (Relational Model)
  - If you *do* have this domain knowledge, it is possible to identify candidate keys (minimal subsets of attributes that FD all attributes)

- One can state an FD does *not* hold given a relation state by identifying violating tuple(s)

# FD Example (3)

| | StudentID | Year | Class | Instructor |
|---|---|---|---|---|
| $t_1$ | 1 | Sophomore | COMP355 | Wu |
| $t_2$ | 2 | Sophomore | COMP285 | Wu |
| $t_3$ | 3 | Junior | COMP355 | Wu |
| $t_4$ | 3 | Junior | COMP285 | Wu |
| $t_5$ | 2 | Sophomore | COMP355 | Russo |
| $t_6$ | 4 | Sophomore | COMP355 | Russo |

$$\{StudentID\} \rightarrow \{Year\}$$
$$\{StudentID, Class\} \rightarrow \{Instructor\}$$

- *Every student is classified as either a Freshman, Sophomore, Junior, or Senior.*
- *Students can take only a single section of a class, taught by a single instructor.*

**Key(s):** $\{StudentID, Class\}$

# FD Example (4)

| | StudentID | Year | Class | Instructor |
|---|---|---|---|---|
| $t_1$ | 1 | Sophomore | COMP355 | Wu |
| $t_2$ | 2 | Sophomore | COMP285 | Wu |
| $t_3$ | 3 | Junior | COMP355 | Wu |
| $t_4$ | 3 | Junior | COMP285 | Wu |
| $t_5$ | 2 | Sophomore | COMP355 | Russo |
| $t_6$ | 4 | Sophomore | COMP355 | Russo |

$$\{StudentID\} \nrightarrow \{Instructor\}$$
$$\{StudentID\} \nrightarrow \{Class\}$$
$$\{Year\} \nrightarrow \{StudentID\}$$
$$\{Year\} \nrightarrow \{Instructor\}$$
$$\{Year\} \nrightarrow \{Class\}$$

$$\{Class\} \nrightarrow \{Year\}$$
$$\{Class\} \nrightarrow \{StudentID\}$$
$$\{Class\} \nrightarrow \{Instructor\}$$
$$\{Instructor\} \nrightarrow \{Class\}$$
$$\{Instructor\} \nrightarrow \{Year\}$$
$$\{Instructor\} \nrightarrow \{StudentID\}$$

# FD Example (5)

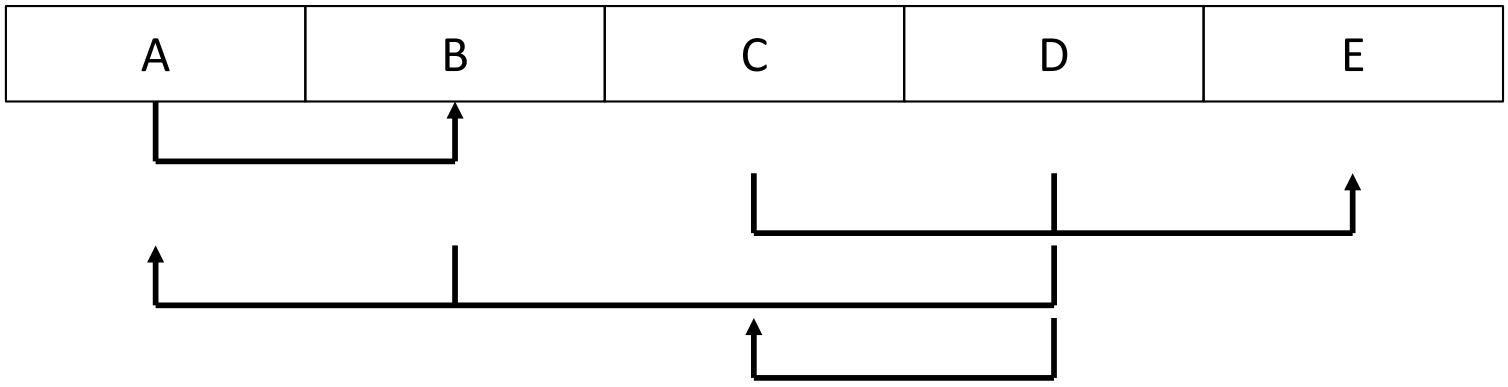
| | StudentID | Year | Class | Instructor |
|---|---|---|---|---|
| $t_1$ | 1 | Sophomore | COMP355 | Wu |
| $t_2$ | 2 | Sophomore | COMP285 | Wu |
| $t_3$ | 3 | Junior | COMP355 | Wu |
| $t_4$ | 3 | Junior | COMP285 | Wu |
| $t_5$ | 2 | Sophomore | COMP355 | Russo |
| $t_6$ | 4 | Sophomore | COMP355 | Russo |

$$\{StudentID, Instructor\} \nrightarrow \{Class\}$$
$$\{Year, Class\} \nrightarrow \{Instructor\}$$
$$\{Year, Class\} \nrightarrow \{StudentID\}$$
$$\{Class, Instructor\} \nrightarrow \{StudentID\}$$
$$\{Class, Instructor\} \nrightarrow \{Year\}$$
$$\{Year, Class, Instructor\} \nrightarrow \{StudentID\}$$

# Exercise

Consider the following visual depiction of the functional dependencies of a relational schema.

1. List all FDs in algebraic notation
2. Identify all key(s) of of this relation



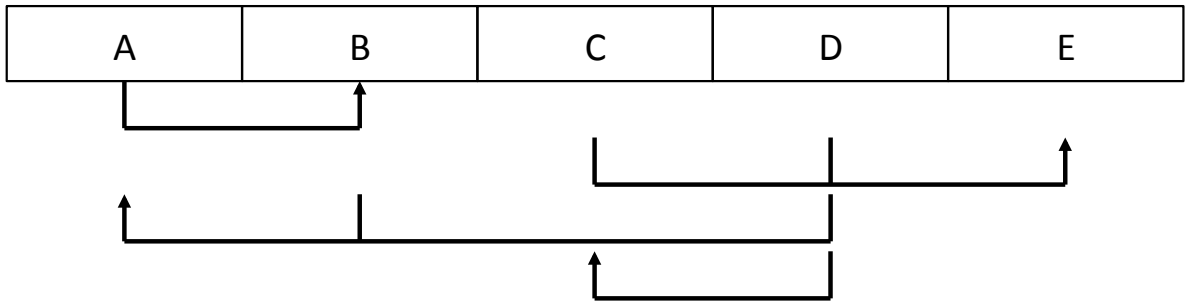
# Answer

**Functional Dependencies**

$$A \rightarrow B$$
$$CD \rightarrow E$$
$$BD \rightarrow A$$
$$D \rightarrow C$$

**Keys**

$$DA$$
$$DB$$



# Normalization Process

- Submit a relational schema to a set of tests (related to FDs) to certify whether it satisfies a **normal form**
- If it does not pass, decompose into smaller relations that satisfy the normal form
  - Must be non-additive (i.e. no spurious tuples!)
- The normal form of a relation refers to the highest normal form that it meets
  - As of 2002 the most constraining is 6NF
- The normal form of a database refers to the lowest normal form that any relation meets
  - Practically, a database is normalized if all relations ≥ 3NF

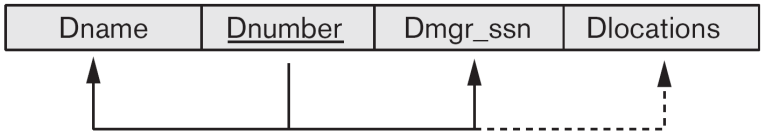
# 1NF – First Normal Form

- The domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute
- No relations within relations or relations as attribute values within tuples
- Considered part of the formal definition of a relation in the basic (flat) relational model
  - In other words, an *implicit* constraint (Lecture 2)

# 1NF Violation (1)

**(a)**

**DEPARTMENT**




**(b)**

**DEPARTMENT**

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

**(c)**

**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 |

# 1NF Violation (2)

**(a)**

**EMP_PROJ**

| | | Projs | |
|---|---|---|---|
| Ssn | Ename | Pnumber | Hours |

**(b)**

**EMP_PROJ**

| 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, Alicia J. | 30 | 30.0 |
| | | 10 | 10.0 |
| 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 |
|---|---|---|

# Important FD Definitions

| | |
|---|---|
| **Trivial FD** | $X \rightarrow Y,\ Y \subseteq X$ |
| **Non-Prime** | An attribute that does not occur in any key (opposite: Prime) |
| **Full FD** | $X \rightarrow Y,\ \forall A \in X((X - \{A\}) \nrightarrow Y)$ |
| **Transitive FD** | $X \rightarrow Z \therefore X \rightarrow Y\ and\ Y \rightarrow Z$ |

# 2NF – Second Normal Form

- 1NF **AND** every non-prime attribute is fully FD on the primary key
  - Must test all FDs whose LHS is part of the PK

- To fix, decompose into relations in which non-prime attributes are associated only with the part of the primary key on which they are fully functionally dependent

# 2NF Example

| StudentID | Course | StudentAddress |
|---|---|---|
| 1 | COMP570 | 555 Huntington |
| 1 | COMP285 | 555 Huntington |
| 2 | COMP570 | 610 Huntington |
| 3 | COMP355 | Louis Prang |
| 3 | COMP553 | Louis Prang |

$$\{StudentID, Course\} \rightarrow \{StudentAddress\}$$
$$\{StudentID\} \rightarrow \{StudentAddress\}$$

| StudentID | StudentAddress |
|---|---|
| 1 | 555 Huntington |
| 2 | 610 Huntington |
| 3 | Louis Prang |

| StudentID | Course |
|---|---|
| 1 | COMP570 |
| 1 | COMP285 |
| 2 | COMP570 |
| 3 | COMP355 |
| 3 | COMP553 |

# 2NF Can Suffer Update Anomalies

| Year | Winner | Nationality |
|---|---|---|
| 1994 | Miguel Indurain | Spain |
| 1995 | Miguel Indurain | Spain |
| 1996 | Bjarne Riis | Denmark |
| 1997 | Jan Ullrich | Germany |

- Relation is in 2NF?
  - Trivially true (why?)
- List all non-trivial FDs for this relation state

$$\{Year\} \rightarrow \{Winner, Nationality\}$$
$$\{Winner\} \rightarrow \{Nationality\}$$

- What if we insert (1998, Jan Ullrich, USA)?

# 3NF – Third Normal Form

- 2NF **AND** every non-prime attribute is non-transitively dependent on every key

  *"A non-key field must provide a fact about the key, the whole key, and nothing but the key. So help me Codd."*

- To fix, decompose into multiple relations, whereby the intermediate non-key attribute(s) functionally determine other non-prime attributes

# 3NF Example

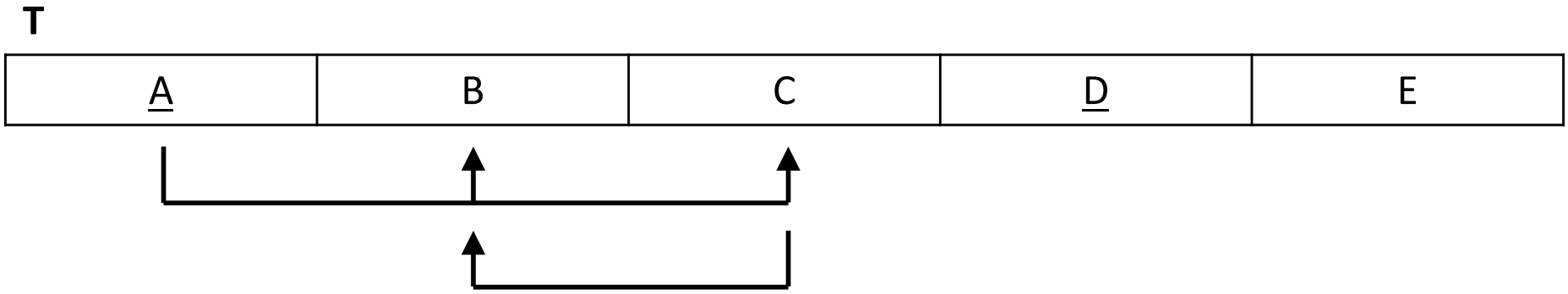
| Year | Winner | Nationality |
|---|---|---|
| 1994 | Miguel Indurain | Spain |
| 1995 | Miguel Indurain | Spain |
| 1996 | Bjarne Riis | Denmark |
| 1997 | Jan Ullrich | Germany |

$Year \rightarrow Nationality \therefore$
$Year \rightarrow Winner$ **and**
$Winner \rightarrow Nationality$

| Year | Winner |
|---|---|
| 1994 | Miguel Indurain |
| 1995 | Miguel Indurain |
| 1996 | Bjarne Riis |
| 1997 | Jan Ullrich |

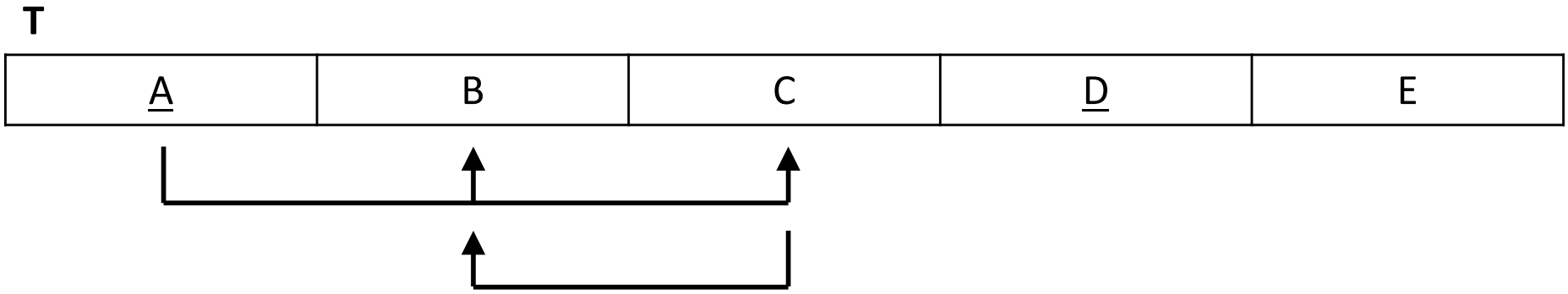
| Winner | Nationality |
|---|---|
| Miguel Indurain | Spain |
| Bjarne Riis | Denmark |
| Jan Ullrich | Germany |

# Exercise




Consider the schema for relation **T**, as well as all FDs. What is the normal form of **T**? If **T** violates 3NF, provide a 3NF decomposition that satisfies the FDs (including the primary key) and does not produce spurious tuples. Show and explain all steps of your analysis and decomposition (if applicable).

# Answer (1)




List non-trivial FDs

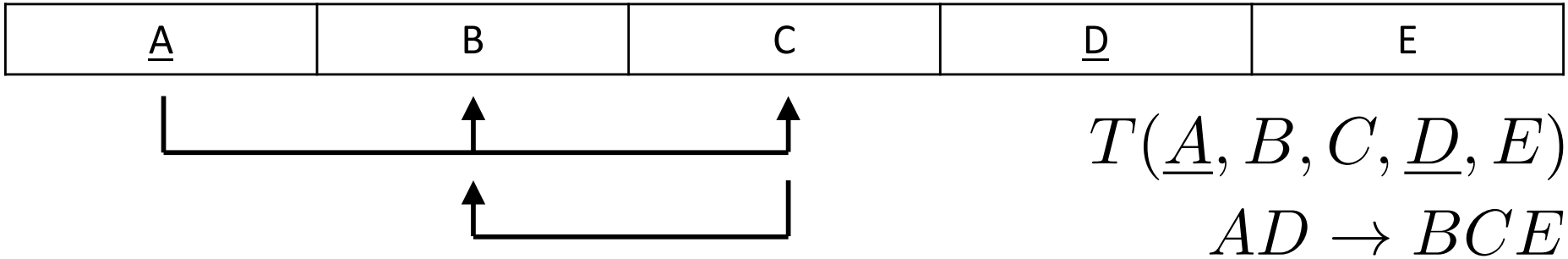
$$AD \rightarrow BCE$$
$$A \rightarrow BC$$
$$C \rightarrow B$$

Written algebraically

$$T(\underline{A}, B, C, \underline{D}, E)$$

# Answer (2)

T



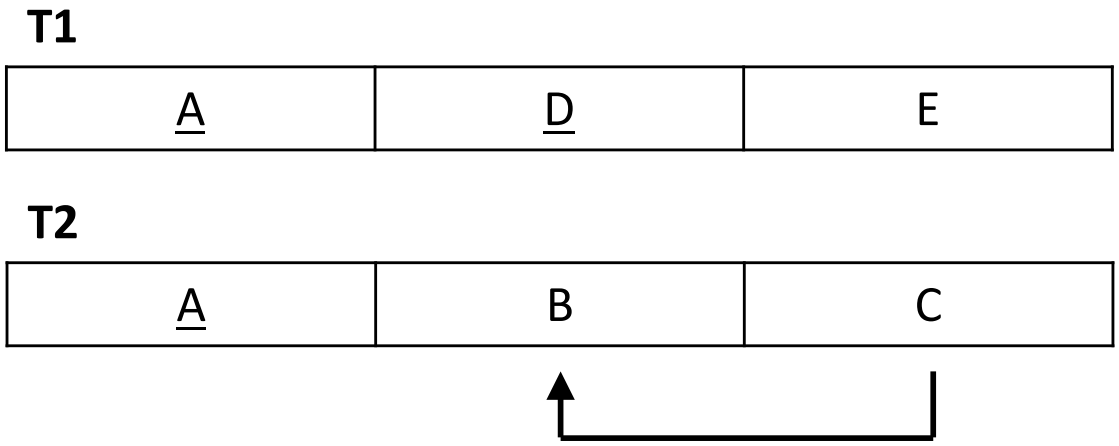

$$T(\underline{A}, B, C, \underline{D}, E)$$
$$AD \rightarrow BCE$$
$$A \rightarrow BC$$
$$C \rightarrow B$$

T is in …

- Both B & C are FD on A
  - Thus not fully FD on PK (AD)

Decompose!

# Answer (3)




$$T1(\underline{A}, \underline{D}, E)$$
$$T2(\underline{A}, B, C)$$

$$AD \rightarrow E$$

$$A \rightarrow BC$$

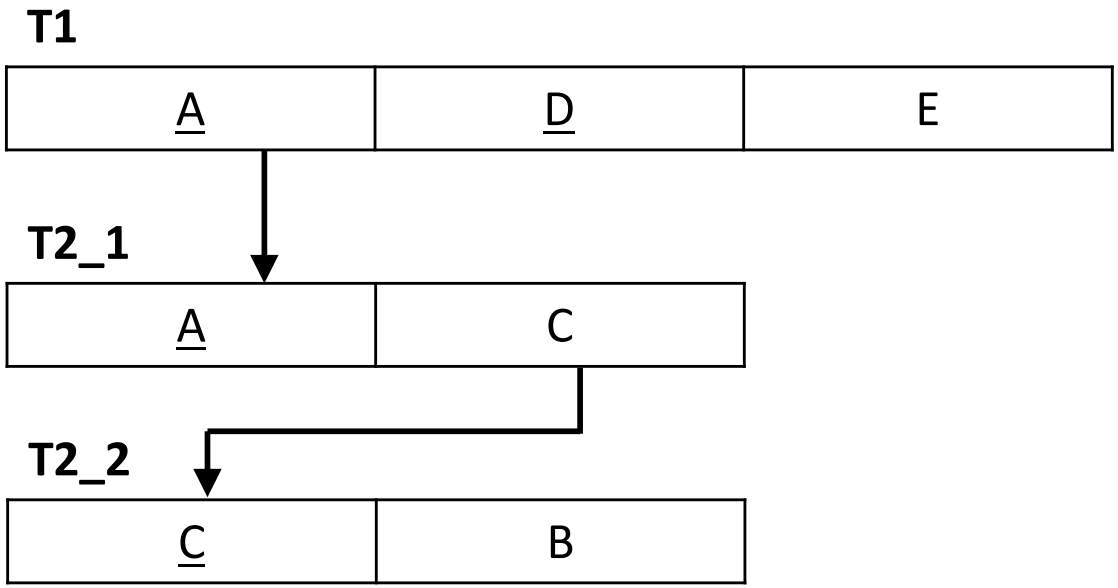
$$C \rightarrow B$$

T1 is in …
- 2NF: E is fully FD on AD
- 3NF: No transitive FDs (trivially true)

T2 is in …
- 2NF: B and C fully FD on A (trivially true)
- !3NF: B is transitively FD on A [via C]

Decompose!

# Answer (4)



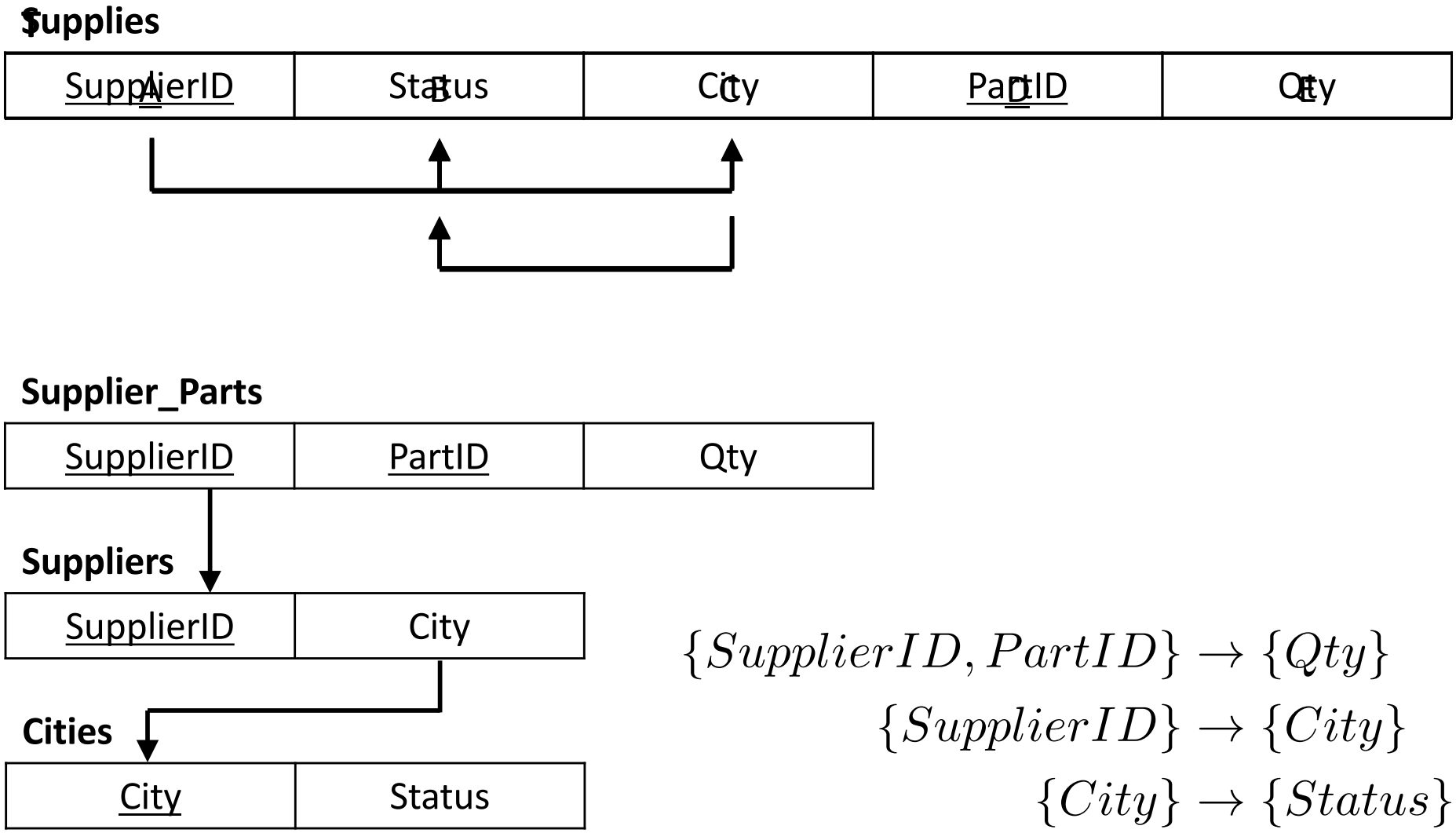

$$T1(\underline{A}, \underline{D}, E)$$
$$T2\_1(\underline{A}, C)$$
$$T2\_2(\underline{C}, B)$$

$$AD \rightarrow E$$
$$A \rightarrow C$$
$$C \rightarrow B$$

Database is in 3NF

- Why?

# Answer (5)




$$\{SupplierID, PartID\} \rightarrow \{Qty\}$$
$$\{SupplierID\} \rightarrow \{City\}$$
$$\{City\} \rightarrow \{Status\}$$

# Summary

- Normalization is the theory and process by which to evaluate and improve relational database design
  - Makes the schema informative
  - Minimizes information duplication
  - Avoids modification anomalies
  - Disallows spurious tuples

- Make sure all your relations are *at least* 3NF!
  - Higher normal forms exist
  - We may reduce during physical design