# Database Systems

# Lecture #12

Sang-Wook Kim

Hanyang University

# Objectives

◆ To learn design guidelines for relation schemas and functional dependency

- Discuss *goodness* of database schemas

- Designing good database schemas

- Concepts of functional dependency

◆ *Informal* Design Guidelines for Relation Schemas

relational schema .

ER -> relational

- Semantics of Attributes in Relations

- Redundant Information and Update Anomalies

- NULL Values in Tuples

- Spurious Tuples

◆ Functional Dependencies

# Informal Design Guidelines for Relation Schemas

◆ Designing a relational database

- Process to define a set of *good table schemas* for a given application

- Decide the attributes to be included in each table

# Informal Design Guidelines for Relation Schemas

◆ Rough criteria for a *good table schema*

- Semantics of attributes

- Redundant information in tuples

- NULL values in tuples

- Spurious tuples

◆ Formal criteria

- *Normalization*

◆ Determining a table schema

- All the attributes belonging to one relation should have meanings mutually-related in the real-world

- Ex: STUDENT (name, age, gender, address)

◆ Example of a good design

**EMPLOYEE**                                    F.K.

| Ename | Ssn | Bdate | Address | Dnumber |
|-------|-----|-------|---------|---------|

P.K.

**DEPARTMENT**            F.K.

| Dname | Dnumber | Dmgr_ssn |
|-------|---------|----------|

P.K.

**PROJECT**                                    F.K.

| Pname | Pnumber | Plocation | Dnum |
|-------|---------|-----------|------|

P.K.

**DEPT_LOCATIONS**
F.K.

| Dnumber | Dlocation |
|---------|-----------|

P.K.

**WORKS_ON**
F.K.          F.K.

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

P.K.

◆ Each tuple should *correspond to only one* entity instance or relationship instance

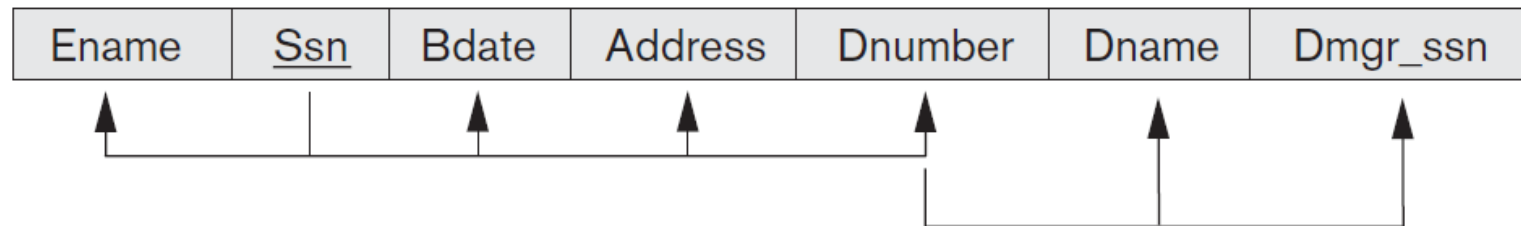◆ Do not combine attributes from multiple entity types and relationship types into a single relation

◆ Example of violating Guideline 1

**(a)**

**EMP_DEPT**

| Ename | Ssn | Bdate | Address | Dnumber | Dname | Dmgr_ssn |
|-------|-----|-------|---------|---------|-------|----------|

**(b)**

**EMP_PROJ**

| Ssn | Pnumber | Hours | Ename | Pname | Plocation |
|-----|---------|-------|-------|-------|-----------|

FD1

FD2

FD3
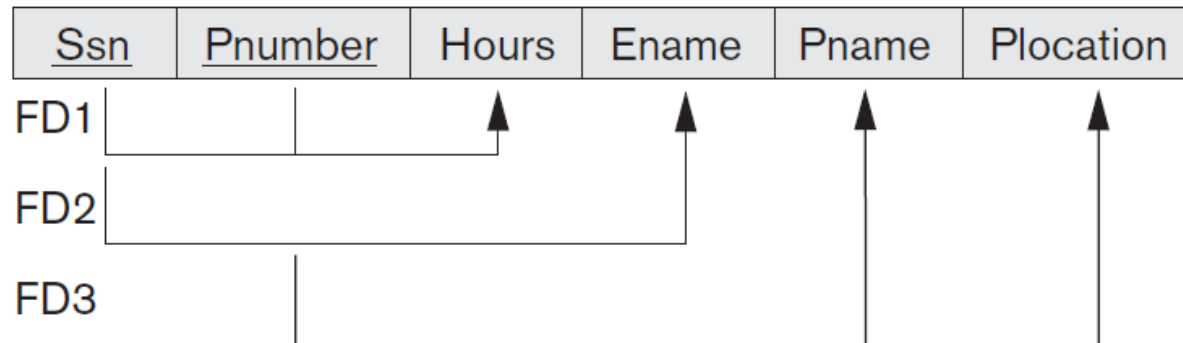
# Redundant Information

◆ Minimizing storage space

- One of main goals of table schema design

- By eliminating redundant information

◆ Example of a database instance

**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 |

**DEPT_LOCATIONS**

| Dnumber | Dlocation |
|---|---|
| 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Sugarland |
| 5 | Houston |

Data
And
Knowledge
Engineering

◆ Example of a database instance

**WORKS_ON**

| Ssn | Pnumber | Hours |
|---|---|---|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | Null |

**PROJECT**

| Pname | Pnumber | Plocation | Dnum |
|---|---|---|---|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

# Redundant Information

◆ Example of tables with redundancy (1:N)

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 |

Data And Knowledge Engineering

Page 14

◆ Example of tables with redundancy (N:M)

| | | 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 |

# Update Anomalies

◆ Updating tuples won't work as expected

- Insertion

- Deletion

- Modification

# Update Anomalies

◆ Insertion anomalies

- Difficult to insert a new department that *has no employees yet* in the EMP_DEPT table

- This problem will not occur if we build *separate tables* DEPARTMENT and EMPLOYEE

◆ Deletion anomalies

- When deleting the last employee working for a particular department from the EMP_DEPT table

  - Information concerning that department is lost from the database

- This problem will not occur if we build *separate tables* DEPARTMENT and EMPLOYEE

◆ Modification anomalies

- If we change the manager of a particular department in the EMP_DEPT table

  - Must update those tuples of all employees who work in that department

- This problem will not occur if we build *separate tables* DEPARTMENT and EMPLOYEE

# 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 updating the database will operate correctly

# NULL Values in Tuples

◆ Meanings of NULLs

- Unknown value

- Unavailable or withheld value

- Not applicable attribute

# NULL Values in Tuples

◆ Problems with NULL values

- Wasted storage space

- Ambiguous meaning

- Difficult to use aggregate functions

  ▪ Each individual NULL value considered to be different

# Guideline 3

◆ Avoid placing attributes in a base relation of their values may frequently be NULL

◆ If NULLs are unavoidable:

- Make sure that they apply in exceptional cases only, not to a majority of tuples

- Create a *new relation for only those tuples* with NULL values

◆ Example

- If only 10 percent of employees have their own individual offices

- Method 1:
  - Include Office_number attribute to EMPLOYEE relation
  - 90% of tuples will have NULLs

◆ Example

- If only 10 percent of employees have their own individual offices

- Method 2:

  - Create EMP_OFFICES (Essn, Office_number) relation

  - EMP_OFFICES will have tuples for only the employees with office

# Spurious Tuples

◆ NATURAL JOIN can produce tuples with invalid information

- Called spurious tuples

- Caused by inappropriately designed relations

◆ Example

**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 |

◆ Example

● Join EMP_LOCS and EMP_PROJ1 with Plocation

| | 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. |

*
*
*

◆ Avoid relations that contain join attributes that are *not (foreign key, primary key) combinations*

# Functional Dependencies

◆ Concepts

- Formal tool for analysis of relational schemas

- Used to define *normal forms* along with keys

# Functional Dependencies

◆ Format

- Denoted by $X \rightarrow Y$

  ▪ $X$, $Y$: set of attributes that are subset of $R$

- Values of $X$ *functionally determine* the values of $Y$

- Or $Y$ is *functionally dependent on $X$*

# Functional Dependencies

◆ Meanings

- For any two tuples $t_1$ and $t_2$ in a particular relation that have $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$

  - Thus the tuples with same values for $X$ also have same values for $Y$

  - A value of attribute set $X$ uniquely determines the value of attribute set $Y$

- Derived from the constraints required in the real-world application

  - Major -> Department

  - Department -> School

# Functional Dependencies

◆ Note that...

- If $X$ is a *candidate key* of $R$

    - Then $X \rightarrow Y$

    - For any subset of attributes $Y$ of $R$

- If $X \rightarrow Y$ in $R$

    - This does not say whether or not $Y \rightarrow X$ in $R$

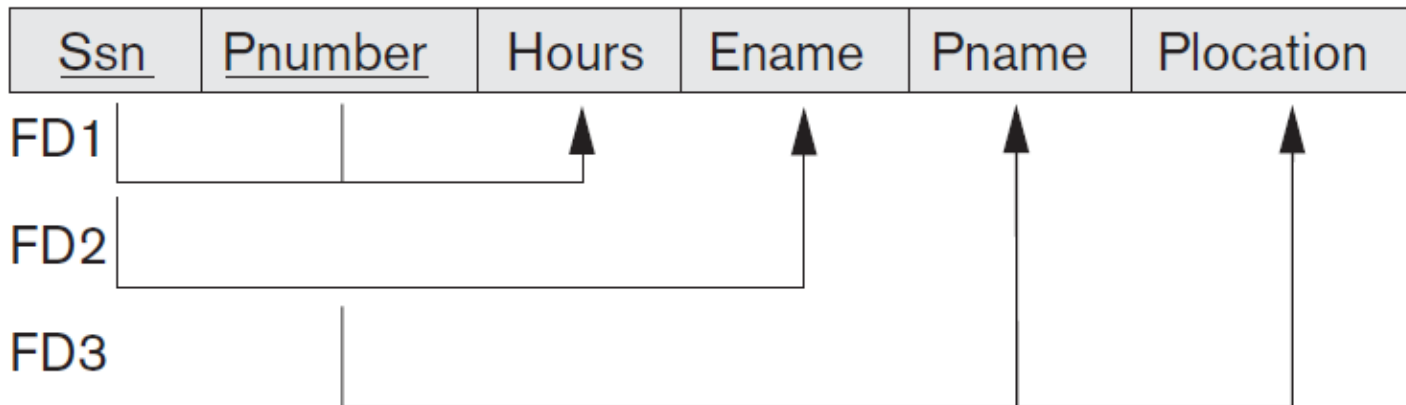◆ FD is a *property of table schema R*

- ● Not of a particular table instance *r*

  - ▪ The constraint must hold for all instance *r*(*R*)

- ● Cannot be inferred automatically from a given *r*

  - ▪ Must defined explicitly before the database is built

◆ Example

● For EMP_PROJ table

a. Ssn $\rightarrow$ Ename

b. Pnumber $\rightarrow$ {Pname, Plocation}

c. {Ssn, Pnumber} $\rightarrow$ Hours

**EMP_PROJ**

| Ssn | Pnumber | Hours | Ename | Pname | Plocation |
|-----|---------|-------|-------|-------|-----------|

FD1

FD2

FD3

# Inference Rules for FDs

◆ *F*:

- Set of functional dependencies that are specified on table schema *R*

◆ $F^+$:

- Set of all dependencies that can be inferred from *F*

- Also include *F*

- Called the *closure* of *F*

◆ Example

- $F$ = {Ssn → {Ename, Bdate, Address, Dnumber},

    Dnumber → {Dname, Dmgr_ssn} }

- Additional FDs that can be inferred from $F$:

    ▪ Ssn → {Dname, Dmgr_ssn}

    ▪ Ssn → Ssn

    ▪ Dnumber → Dname

◆ Armstrong's inference rules

- IR1 (reflexive rule): If $X \supseteq Y$, then $X \rightarrow Y$

- IR2 (augmentation rule): $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

- IR3 (transitive rule): $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$

- *Sound* and *complete (Proof?)*

◆ Other inference rules

- IR4 (decomposition rule): $\{X \rightarrow YZ\} \models X \rightarrow Y$

- IR5 (union rule): $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$

- IR6 (pseudo-transitive rule): $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$

- Can prove IR4~6 by using IR1~3

# Systematic Way to Determine Additional FDs

◆ *X:*

- Set of attributes that appear as a left-hand side of some FD in *F*

◆ *X⁺:*

- Set of all the attributes that are *functionally determined* by *X* based on *F*

- Called *closure of X under F*

- Determined for each *X*

- Can be calculated by applying inference rules repeatedly

◆ Algorithm for determining $X^+$

$$X^+ := X;$$
$$\text{repeat}$$
$$\quad \text{old}X^+ := X^+;$$
$$\quad \text{for each functional dependency } Y \rightarrow Z \text{ in } F \text{ do}$$
$$\quad \quad \text{if } X^+ \supseteq Y \text{ then } X^+ := X^+ \cup Z;$$
$$\text{until } (X^+ = \text{old}X^+);$$

◆ Example

$$F = \{ \text{Ssn} \rightarrow \text{Ename},$$
$$\text{Pnumber} \rightarrow \{\text{Pname, Plocation}\},$$
$$\{\text{Ssn, Pnumber}\} \rightarrow \text{Hours}\}$$

◆ Example

$\{Ssn\}^+ = \{Ssn, Ename\}$

$\{Pnumber\}^+ = \{Pnumber, Pname, Plocation\}$

$\{Ssn, Pnumber\}^+ = \{Ssn, Pnumber, Ename, Pname, Plocation, Hours\}$

# Summary

◆ Informal guidelines for good design

◆ Functional dependency

- Basic tool for analyzing relational schemas

- Inference rules

# References

1. Codd, Edgar F. "A relational model of data for large shared data banks."*Communications of the ACM* 13.6 (1970): 377-387.

2. Ullman J. *Principles of Database and Knowledge-Base Systems*, Vol. 1, Computer Science Press, 1988.

3. Maier, David. *The theory of relational databases*. Vol. 11. Rockville: Computer science press, 1983.

4. Atzeni, Paolo, and Valeria De Antonellis. *Relational database theory*. Benjamin-Cummings Publishing Co., Inc., 1993.

5. Armstrong, William Ward. "Dependency Structures of Data Base Relationships." *IFIP congress*. Vol. 74. 1974.

Data
And
Knowledge
Engineering

# Have a nice day!

Data
And
Knowledge
Engineering