Top-Down Database Design

Mini-world

Requirements

E1

R

Conceptual schema

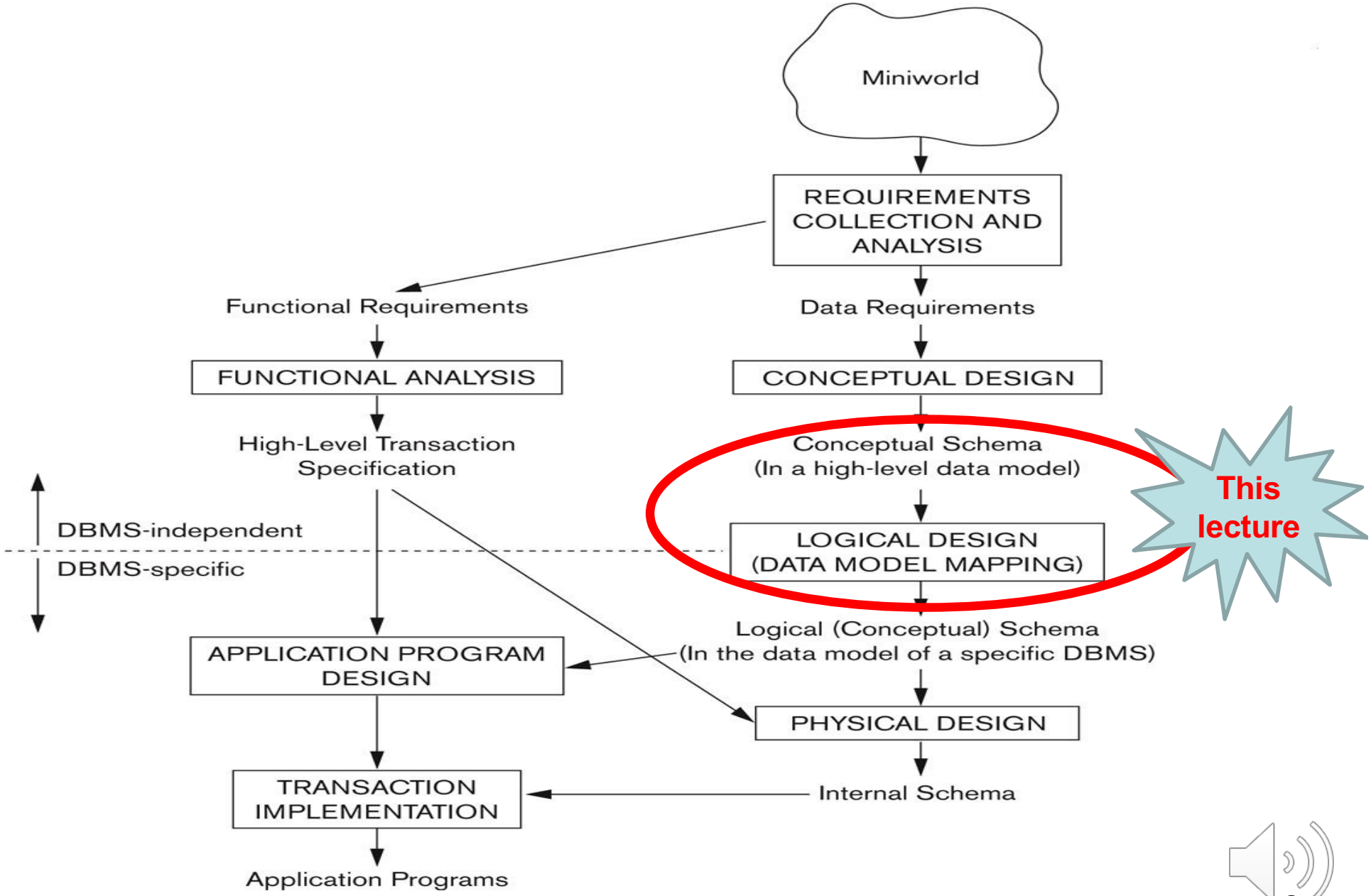
E2

Relation schemas

?

Assoc. Prof. Dr. Dang Tran Khanh (khanh@hcmut.edu.vn)

Overview of Database Design Process



Functional Dependencies & Normalization for Relational DBs

Assoc. Prof. Dr. Dang Tran Khanh
HCMUT, khanh@hcmut.edu.vn

Outline

- Introduction
- Functional dependencies (FDs)
  - Definition of FD
  - Direct, indirect, partial dependencies
- Normalization
  - 1NF and dependency problems
  - 2NF – solves partial dependency
  - 3NF – solves indirect dependency
  - BCNF – well-normalized relations
- Notes and suggestions
- Summary
- Reading suggestion: [1]: Chapters 14, 15

Introduction

- Each *relation schema* consists of a number of attributes and the *relational database schema* consists of a number of relation schemas
- Attributes are grouped to form a relation schema
- Need some formal measure of why one grouping of attributes into a relation schema may be better than another

Introduction

- “Goodness” measures:
  - Redundant information in tuples
  - Update anomalies: modification, deletion, insertion
  - Reducing the NULL values in tuples
  - Disallowing the possibility of generating spurious tuples

Introduction

- Redundant information in tuples: the attribute values pertaining to a particular department (DNUMBER, DNAME, DMGRSSN) are repeated for *every employee who works for that department*

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
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

Introduction

- Update anomalies: modification, deletion, insertion
  - Modification

    →As the manager of a dept. changes we have to update many values according to employees working for that dept.

    →Easy to make the DB **inconsistent**

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
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

Introduction

- Deletion: if Borg James E. leaves, we delete his tuple and lose the existing of dept. 1, the name of dept. 1, and who is the manager of dept. 1

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
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

Introduction

- Insertion:
  - How can we create a department before any employees are assigned to it ??

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
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

Introduction

- Reducing the NULL values in tuples
  - Employees not assigned to any dept.: waste the storage space
  - Other difficulties: aggregation operations (e.g., COUNT, SUM, …) and joins

Introduction

- Disallowing the possibility of generating spurious tuples

EMP_PROJ(SSN, PNUMBER, HOURS, ENAME, PNAME, PLOCATION)

EMP_LOCS(ENAME, PLOCATION)

EMP_PROJ1(SSN, PNUMBER, HOURS, PNAME, PLOCATION)

- Generation of invalid and spurious data during JOINS: PLOCATION is the attribute that relates EMP_LOCS and EMP_PROJ1, and PLOCATION is neither a primary key nor a foreign key in either EMP_LOCS or EMP_PROJ1 (*cf. chapter 14 [1] for more details*)

Introduction

- "Goodness" measures:
  - Redundant information in tuples
  - Update anomalies: modification, deletion, insertion
  - Reducing the NULL values in tuples
  - Disallowing the possibility of generating spurious tuples

  ☞ Normalization
- It helps DB designers determine the best relation schemas
  - A formal framework for analyzing relation schemas based on their keys and on the functional dependencies among their attributes
  - A series of normal form tests that can be carried out on individual relation schemas so that the relational database can be normalized to any desired degree
- It is based on the concept of normal form 1NF, 2NF, 3NF, BCNF, 4NF, 5 NF
- It is a process which ensures that the data is structured in such a way that attributes are grouped with the PK. Attributes that do not directly depend on PK may be extracted to form a new relation

Introduction

- There are two important properties of decompositions:

  (a) non-additive or losslessness of the corresponding join
  (b) preservation of the functional dependencies

- Note that property (a) is extremely important and *cannot* be sacrificed. Property (b) is less stringent and may be sacrificed (see chapter 14)

Functional Dependencies (FDs)

- Definition of FD
- Direct, indirect, partial dependencies
- Homework: chapter 15
  - *Inference Rules for FDs*
  - *Equivalence of Sets of FDs*
  - *Minimal Sets of FDs*

Functional Dependencies (FDs)

- Functional dependencies (FDs) are used to specify *formal measures* of the "goodness" of relational designs
- FDs and keys are used to define **normal forms** for relations
- FDs are **constraints** that are derived from the *meaning* and *interrelationships* of the data attributes
- A set of attributes X *functionally determines* a set of attributes Y if the value of X determines a unique value for Y

Functional Dependencies (FDs)

- 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 t1 and t2 in any relation instance r(R): *If* t1[X]=t2[X], *then* t1[Y]=t2[Y]
- X -> Y in R specifies a *constraint* on all relation instances r(R)
- Examples:
  - social security number determines employee name:

    SSN -> ENAME
  - project number determines project name and location:

    PNUMBER -> {PNAME, PLOCATION}
  - employee ssn and project number determines the hours per week that the employee works on the project:

    {SSN, PNUMBER} -> HOURS

Functional Dependencies (FDs)

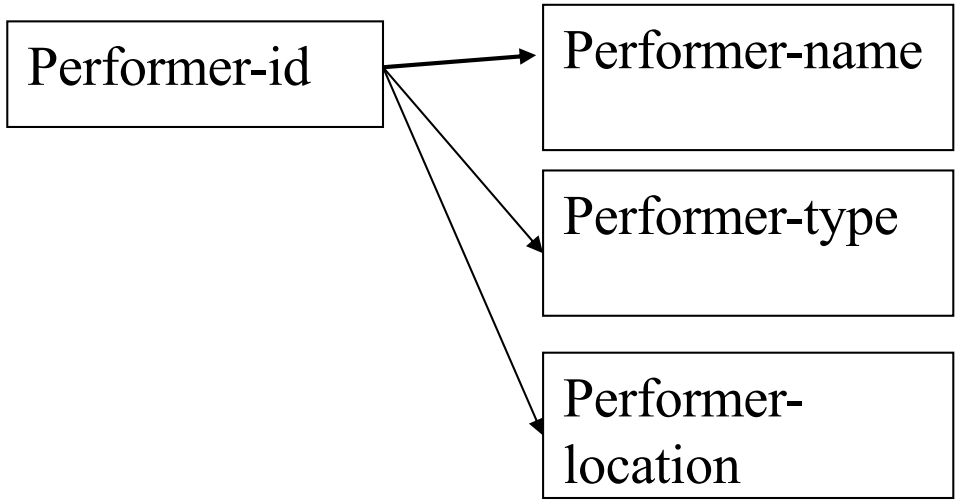
- If K is a key of R, then K functionally determines all attributes in R (since we never have two distinct tuples t1 & t2 with t1[K]=t2[K])

Functional Dependencies (FDs)

- Definition of FD
- Direct, indirect, partial dependencies
- Inference Rules for FDs
- Equivalence of Sets of FDs
- Minimal Sets of FDs

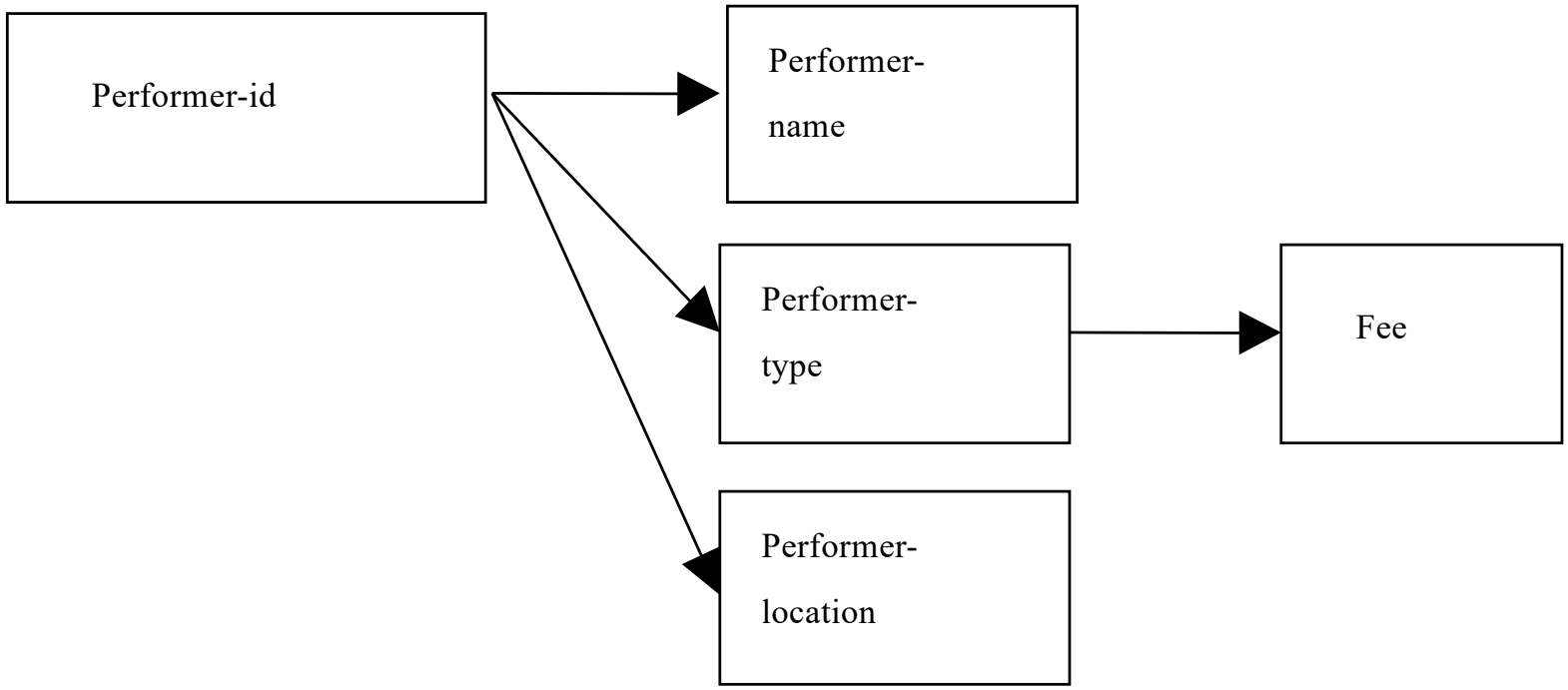
Functional Dependencies (FDs)

- Direct dependency (fully functional dependency): All attributes in R must be fully functionally dependent on the primary key (or the PK is a determinant of all attributes in R)



Functional Dependencies (FDs)

- Indirect dependency (transitive dependency): Value of an attribute is not determined directly by the primary key



Functional Dependencies (FDs)

- Partial dependency
  - **Composite determinant** - more than one value is required to determine the value of another attribute, the combination of values is called a composite determinant

    EMP_PROJ(SSN, PNUMBER, HOURS, ENAME, PNAME, PLOCATION)

    {SSN, PNUMBER} -> HOURS

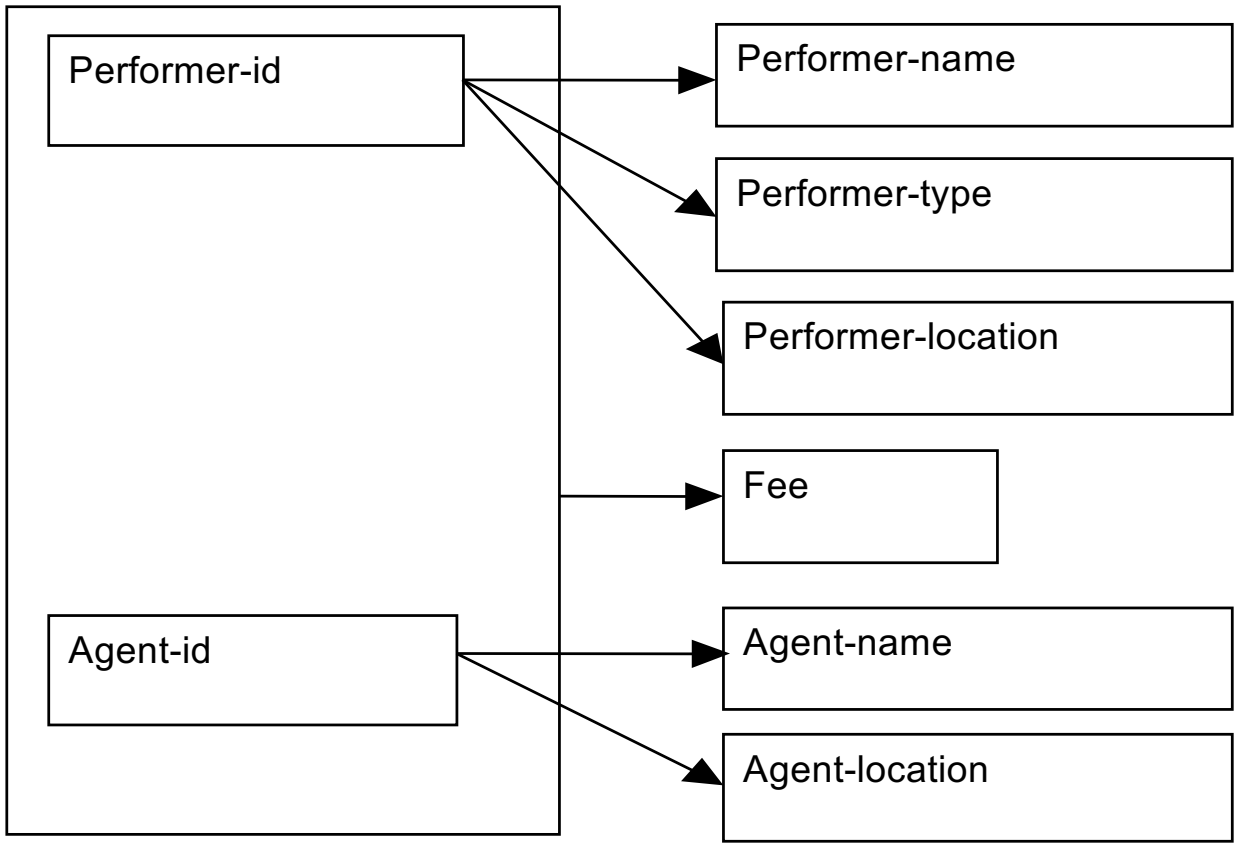
  - **Partial dependency** - if the value of an attribute does not depend on an entire composite determinant, but only part of it, the relationship is known as the partial dependency

    SSN -> ENAME

    PNUMBER -> {PNAME, PLOCATION}

Functional Dependencies (FDs)

- Partial dependency



Outline

- Introduction
- Functional dependencies (FDs)
  - Definition of FD
  - Direct, indirect, partial dependencies
  - Homework: Inference Rules for FDs, Equivalence of Sets of FDs, Minimal Sets of FDs
- Normalization
  - 1NF and dependency problems
  - 2NF – solves partial dependency
  - 3NF – solves indirect dependency
  - BCNF – well-normalized relations
- Notes and suggestions
- Summary
- Reading suggestion: [1]: Chapters 14, 15

Normalization

- **Normalization**: The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations
- **Normal form**: Using keys and FDs of a relation to certify whether a relation schema is in a particular normal form
- **Normalization** is carried out in practice so that the resulting designs are of high quality and meet the desirable properties
- The database designers ***need not*** normalize to the highest possible normal form (3NF, BCNF or 4NF) for practical DB applications (see chapter 14)

Normalization

- Two new concepts:
  - A **Prime attribute** must be a member of *some candidate key*
  - A **Nonprime attribute** is not a prime attribute: it is not a member of any candidate key

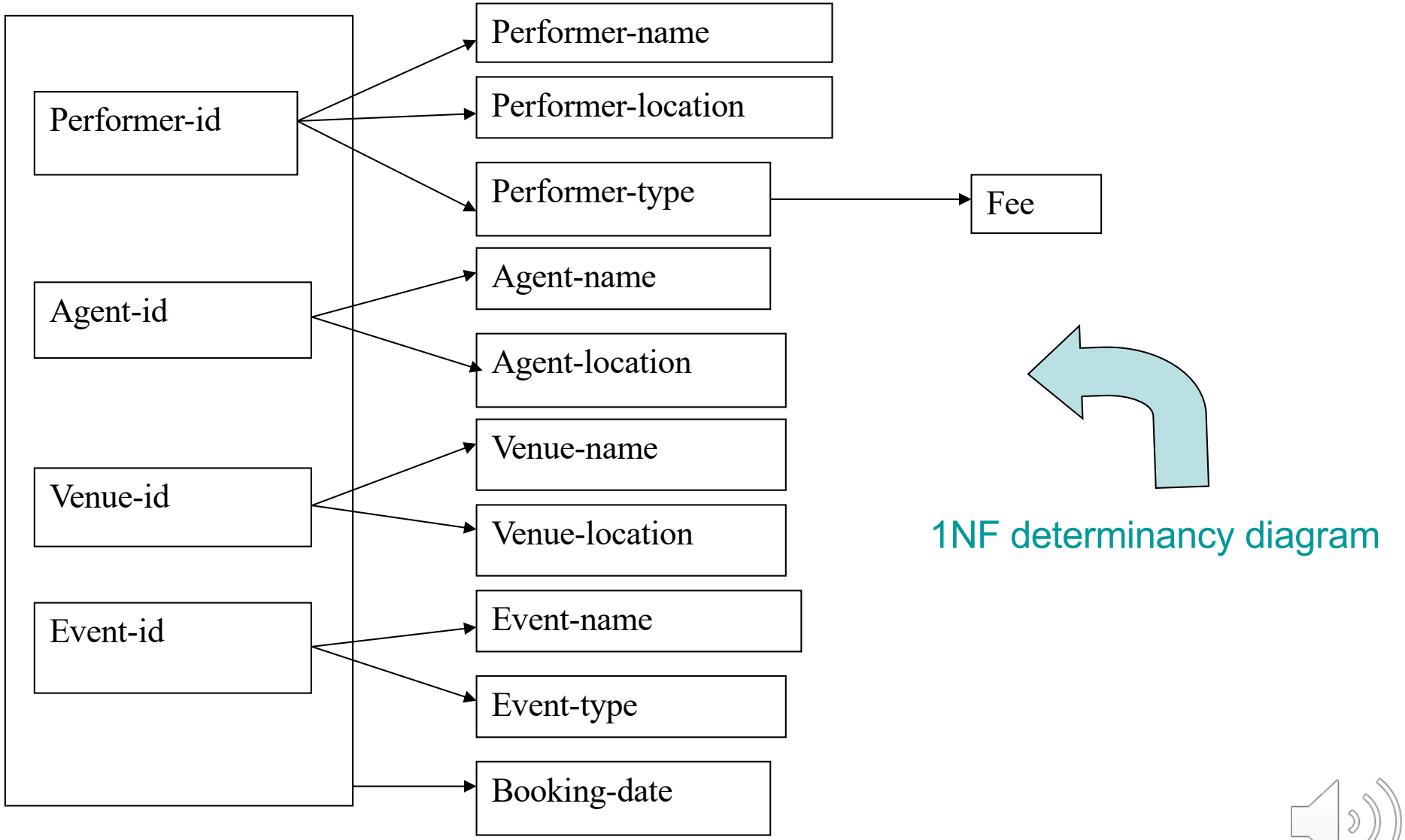
Normalization

- 1NF and dependency problems
- 2NF – solves partial dependency
- 3NF – solves indirect dependency
- BCNF – well-normalized relations

Normalization

- First normal form (1NF): there is only one value at the intersection of each row and column of a relation - no set valued attributes in 1 NF → Disallows composite attributes, multivalued attributes, and **nested relations**
- To be part of the formal definition of a relation in the basic (flat) relational model

1NF



1NF determinancy diagram

Normalization

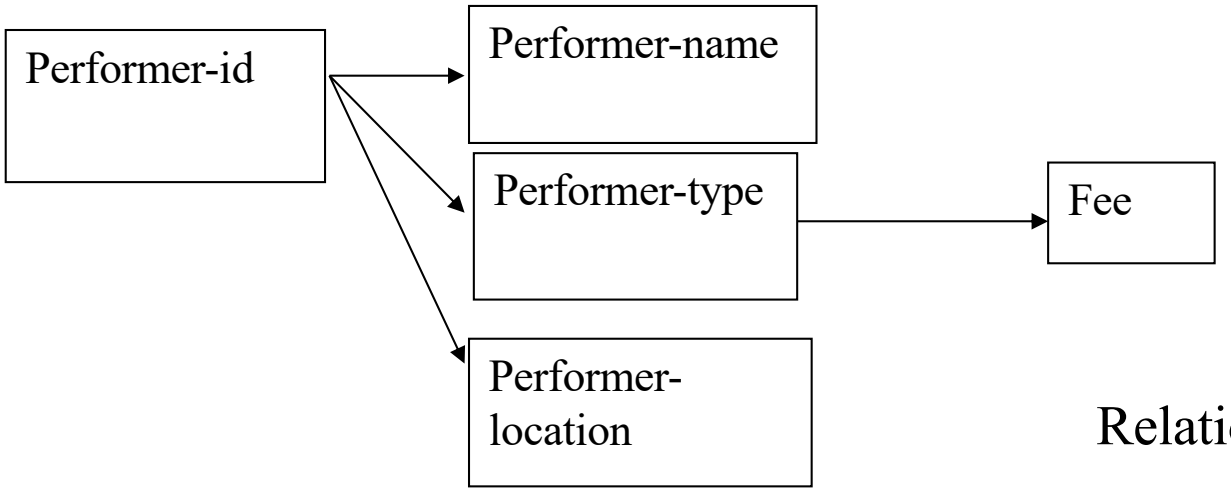
- 1NF and dependency problems
- 2NF – solves partial dependency
- 3NF – solves indirect dependency
- BCNF – well-normalized relations

Normalization

- Second normal form (2NF) - all attributes must be fully functionally dependent on the primary key
- 2NF solves partial dependency problem in 1NF
- Method: identify primary keys and group attributes that relate to the key together to form separate new relations

2NF

2NF determinancy diagram

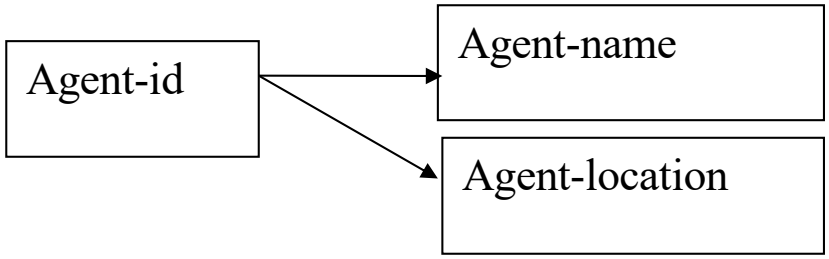


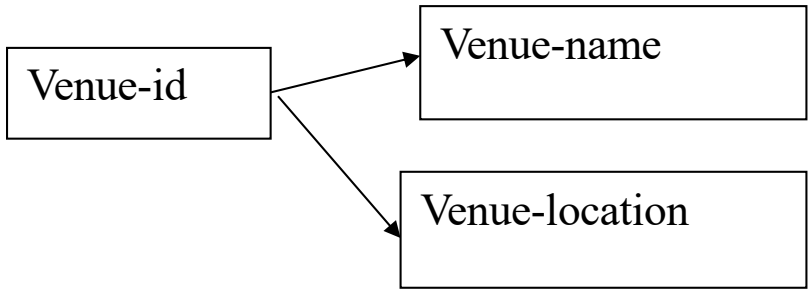
Relation in 2NF

P-id	P-name	P- type	Fee	P-location
101	Baron	Singer	75	York
105	Steed	Dancer	60	Berlin
108	Jones	Actor	85	Bombay
112	Eagles	Actor	85	Leeds
118	Markov	Dancer	60	Moscow
126	Stokes	Comedian	90	Athens
129	Chong	Actor	85	Beijing
134	Brass	Singer	75	London
138	Ng	Singer	75	Penang
140	Strong	Magician	72	Rome
141	Gomez	Musician	92	Lisbon
143	Tan	Singer	75	Chicago
147	Qureshi	Actor	85	London
149	Tan	Actor	85	Taipei
150	Pointer	Magician	72	Paris
152	Peel	Dancer	60	London

2NF

2NF determinancy diagram





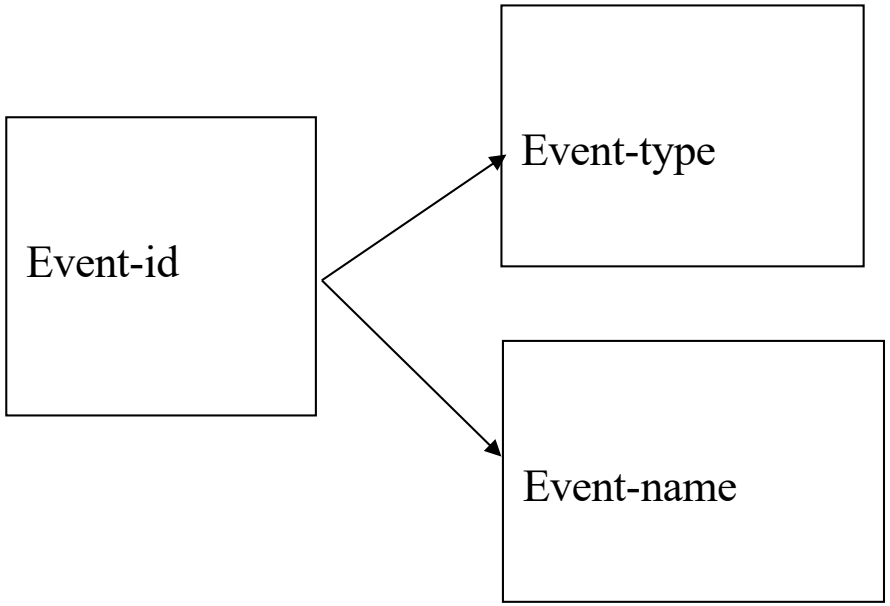
Relation in 2NF

A-id	A-name	A- location
1295	Burton	Lonton
1435	Nunn	Boston
1504	Lee	Taipei
1682	Tsang	Beijing
1460	Stritch	Rome
1522	Ellis	Madrid
1509	Patel	York
1478	Burns	Leeds
1377	Webb	Sydney
1478	Burns	Leeds
1190	Patel	Hue
1802	Chapel	Bristol
1076	Eccles	Oxford
1409	Arkley	York
1428	Vemon	Cairo

V-id	V-name	V-location
59	Atlas	Tokyo
35	Polis	Athens
54	Nation	Lisbon
17	Silbury	Tunis
46	Royale	Cairo
75	Vostok	Kiev
79	Festive	Rome
28	Gratton	Boston
84	State	Kiev
82	Tower	Lima
92	Palace	Milan
62	Shaw	Oxford

2NF

2NF determinancy diagram

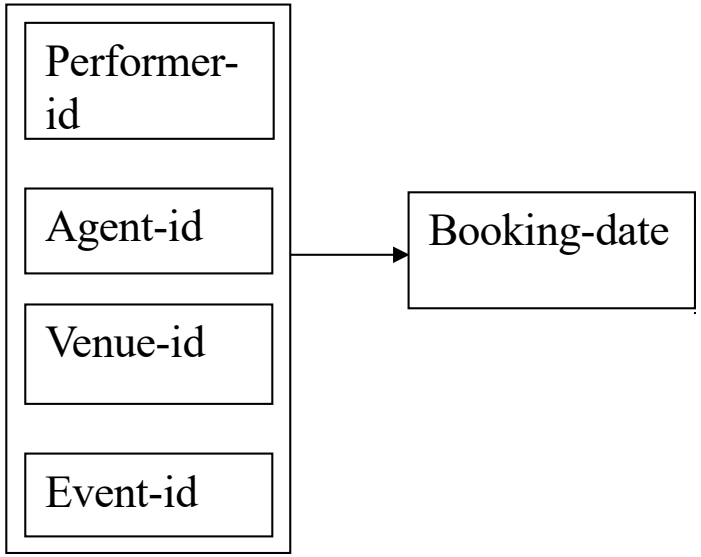


Relation in 2NF

E-id	E-name	E-type
959	Show Time	Musical
907	Elgar 1	Concert
921	Silver Shoe	Ballet
942	White Lace	Ballet
901	The Dark	Drama
913	What Now	Drama
926	Next Year	Drama
952	Gold Days	Drama
934	Angels	Opera
945	Trick-Treat	Variety show
938	New Dawn	Drama
981	Birdsong	Musical
957	Quicktime	Musical
963	Vanish	Magic show
941	Mahler 1	Concert
964	The Friends	Drama
927	Chanson	Opera
971	Card Trick	Magic show
988	Secret Tape	Drama
978	Swift Step	Dance

2NF

2NF determinancy diagram

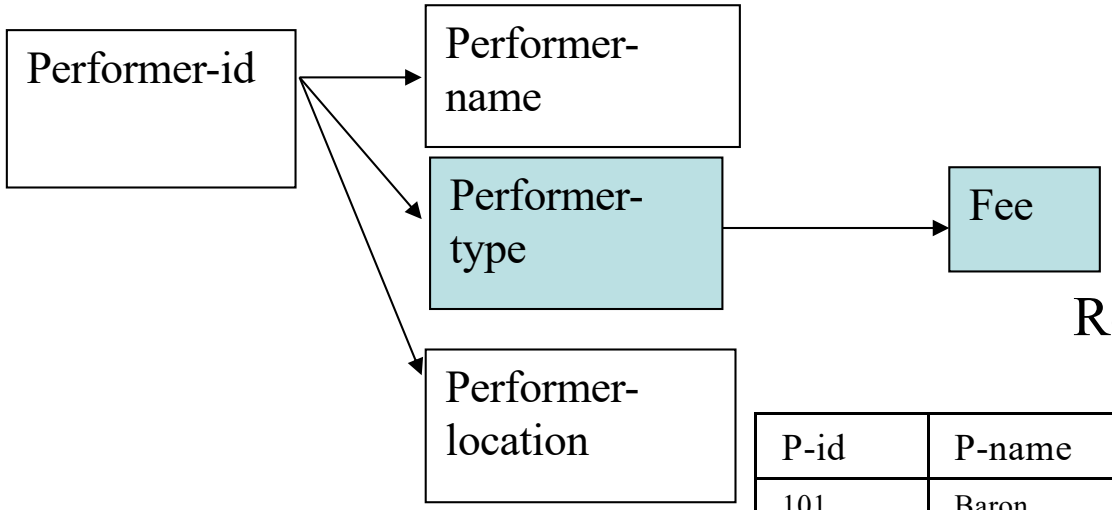


Relation in 2NF

P-id	A-id	V-id	E-id	Booking-date
101	1295	59	959	25-Nov-99
105	1435	35	921	07-Jan-02
105	1504	54	942	10-Feb-02
108	1682	79	901	29-Jul-03
112	1460	17	926	13-Aug-00
112	1522	46	952	05-May-99
112	1504	75	952	16-Mar-99
126	1509	59	945	02-Sept-01
129	1478	79	926	22-Jun-00
134	1504	28	981	18-Sept-01
138	1509	84	957	18-Aug-99
140	1478	17	963	18-Aug-99
141	1478	84	941	21-Jul-00
143	1504	79	927	21-Nov-02
147	1076	17	952	30-Apr-00
147	1409	79	988	17-Apr-00
152	1428	59	978	01-Oct-01

2NF

2NF determinancy diagram



Relation in 2NF

P-id	P-name	P- type	Fee	P-loc'n
101	Baron	Singer	75	York
105	Steed	Dancer	60	Berlin
108	Jones	Actor	85	Bombay
112	Eagles	Actor	85	Leeds
118	Markov	Dancer	60	Moscow
126	Stokes	Comedian	90	Athens
129	Chong	Actor	85	Beijing
134	Brass	Singer	75	London
138	Ng	Singer	75	Penang
140	Strong	Magician	72	Rome
141	Gomez	Musician	92	Lisbon
143	Tan	Singer	75	Chicago
147	Qureshi	Actor	85	London
149	Tan	Actor	85	Taipei
150	Pointer	Magician	72	Paris
152	Peel	Dancer	60	London

- Problem with 2NF:
  - Insertion
  - Modification
  - Deletion

Normalization

- 1NF and dependency problems
- 2NF – solves partial dependency
- 3NF – solves indirect dependency
- BCNF – well-normalized relations

Normalization

- 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

NOTE:

In X -> Y and Y -> 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#, Salary).

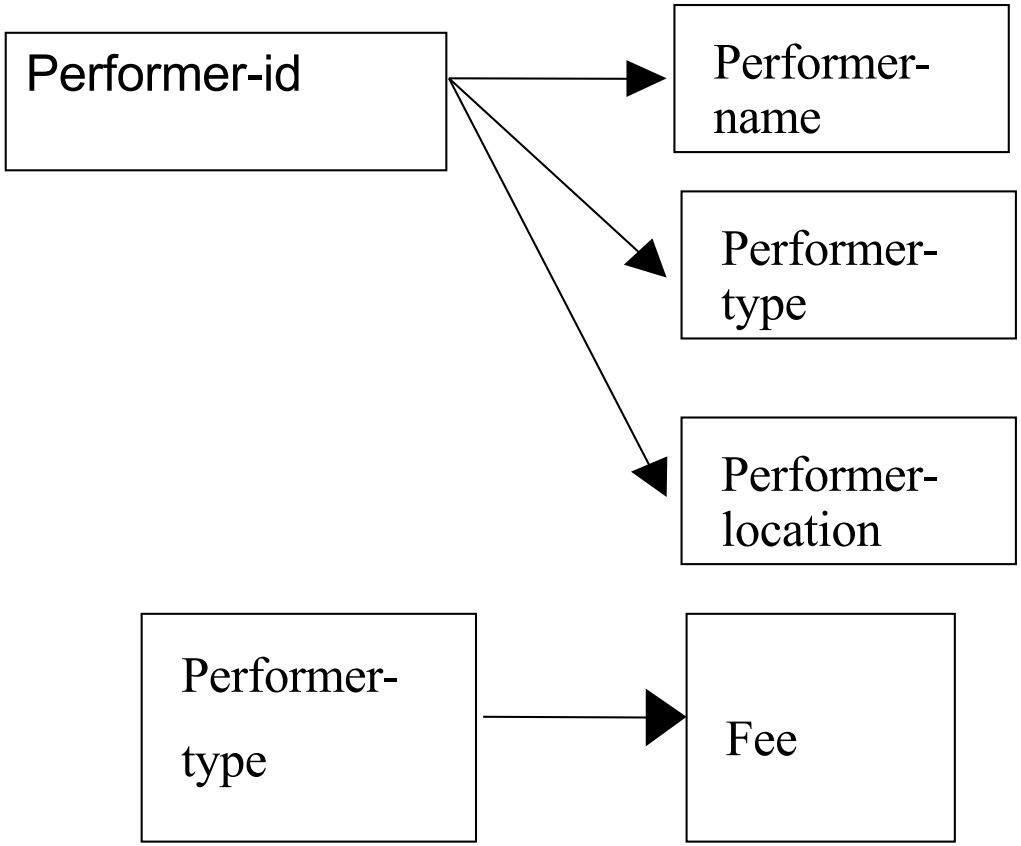
Here, SSN -> Emp# -> Salary and Emp# is a candidate key

Normalization

- 3NF solves indirect (transitive) dependencies problem in 1NF and 2NF
- Method: identify all transitive dependencies and each transitive dependency will form a new relation, with non-prime attributes participating in the transitive dependency and the attribute which determines others as the attributes for the new relation

3NF

3NF determinancy diagram



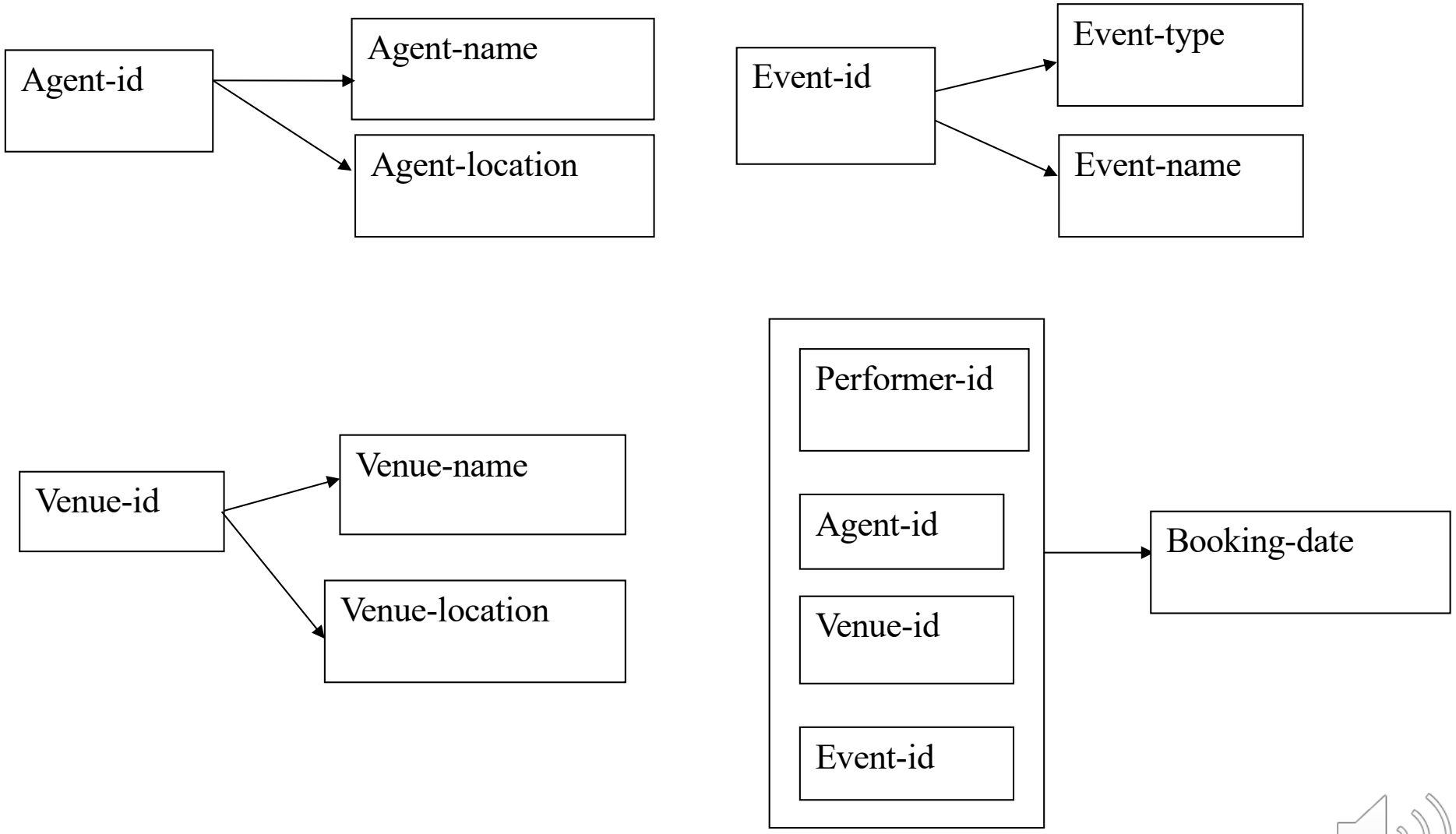
Relation in 3NF

P-id	P-name	P- type	P-loc'n
101	Baron	Singer	York
105	Steed	Dancer	Berlin
108	Jones	Actor	Bombay
112	Eagles	Actor	Leeds
118	Markov	Dancer	Moscow
126	Stokes	Comedian	Athens
129	Chong	Actor	Beijing
134	Brass	Singer	London
138	Ng	Singer	Penang
140	Strong	Magician	Rome
141	Gomez	Musician	Lisbon
143	Tan	Singer	Chicago
147	Qureshi	Actor	London
149	Tan	Actor	Taipei
150	Pointer	Magician	Paris
152	Peel	Dancer	London

P- type	Fee
Singer	75
Dancer	60
Actor	85
Comedian	90
Magician	72
Musician	92

3NF

3NF determinancy diagram



Normalization

- 1NF and dependency problems
- 2NF – solves partial dependency
- 3NF – solves indirect dependency
- BCNF – well-normalized relations

SUMMARY OF NORMAL FORMS based on Primary Keys

Summary of Normal Forms Based on Primary Keys and Corresponding Normalization

Normal Form	Test	Remedy (Normalization)
First (1NF)	Relation should have no multivalued attributes or nested relations.	Form new relations for each multivalued attribute or nested relation.
Second (2NF)	For relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.	Decompose and set up a new relation for each partial key with its dependent attribute(s). Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it.
Third (3NF)	Relation should not have a nonkey attribute functionally determined by another nonkey attribute (or by a set of nonkey attributes). That is, there should be no transitive dependency of a nonkey attribute on the primary key.	Decompose and set up a relation that includes the nonkey attribute(s) that functionally determine(s) other nonkey attribute(s).

General Normal Form Definitions

- The above definitions consider the primary key only
- The following more general definitions take into account relations with multiple candidate keys

General Normal Form Definitions

- 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 -> A holds in R, then either:

  (a) X is a superkey of R, or

  (b) A is a prime attribute of R

Normalization

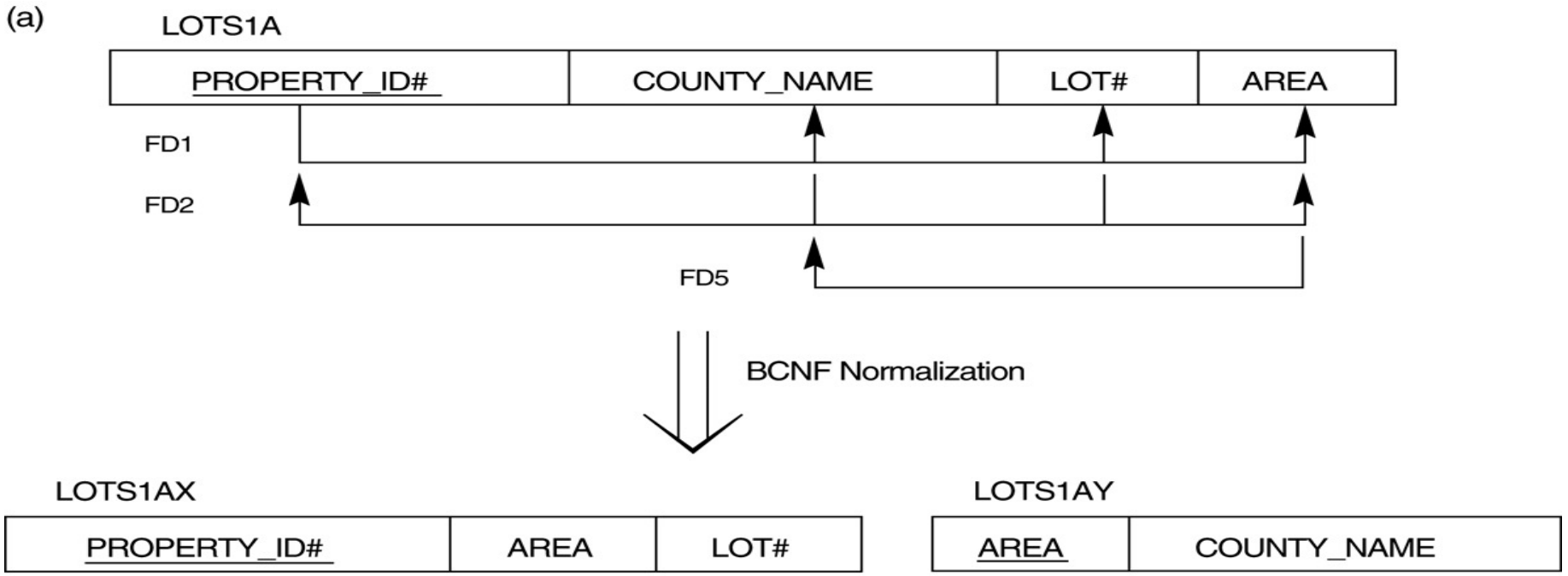
- 1NF and dependency problems
- 2NF – solves partial dependency
- 3NF – solves indirect dependency
- BCNF – well-normalized relations

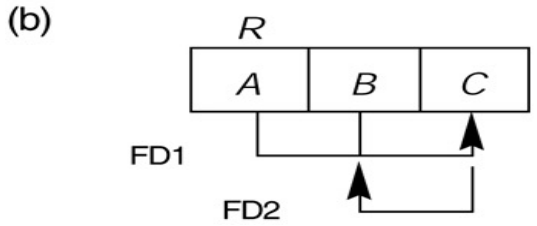
Normalization

- A relation schema R is in **Boyce-Codd Normal Form** (**BCNF**) if whenever an FD X -> A holds in R, then X is a superkey of R

- More details: [1] -> Chapters 14, 15
- The goal is to have each relation in BCNF (or 3NF)

BCNF





Boyce-Codd normal form. (a) BCNF normalization of LOTS1A with the functional dependency FD2 being lost in the decomposition. (b) A schematic relation with FDs; it is in 3NF, but not in BCNF.

Outline

- Introduction
- Functional dependencies (FDs)
  - Definition of FD
  - Direct, indirect, partial dependencies
  - Homework: Inference Rules for FDs, Equivalence of Sets of FDs, Minimal Sets of FDs
- Normalization
  - 1NF and dependency problems
  - 2NF – solves partial dependency
  - 3NF – solves indirect dependency
  - BCNF – well-normalized relations
- Notes and suggestions
- Summary
- Reading suggestion: [1]: Chapters 14, 15

Notes & Suggestions

- [1]: Chapter 15
  - 4NF: based on *multivalued dependency* (MVD)
  - 5NF: based on join dependency

    →Such a dependency is very difficult to detect in practice and therefore, normalization into 5NF is considered very rarely in practice
  - Other normal forms & algorithms
  - ER modeling: top-down database design

    →Bottom-up database design ??

Summary

- Introduction: “goodness” measures for relations
- Functional dependencies (FDs)
  - Definition of FD
  - Direct, indirect, partial dependencies
  - Homework:
    - → Inference Rules for FDs
    - → Equivalence of Sets of FDs
    - → Minimal Sets of FDs
- Normalization
  - 1NF and dependency problems
  - 2NF – solves partial dependency
  - 3NF – solves indirect dependency
  - BCNF – well-normalized relations
- Notes and suggestions: 4NF, 5NF, other normal forms, and bottom-up DB design
- Next Week: see my HP

Q&A