Final Exam Review

CS-6360 Database Design

Chris Irwin Davis, Ph.D.

Email: chrisirwindavis@utdallas.edu
Phone: (972) 883-3574
Office: ECSS 4.705

Nothing

Nada

Ch. 3 & 4: ER & EER Models

UTD

- **Q Type**: Multiple Choice, Multiple Answer, Matching, T/F
- Answer questions about ER/EER diagrams
- Participation and Cardinality ⟺ (min, max)
  - Participation and Cardinality – property of an ***entity***
  - (min, max) – range of an entity's interaction with a ***relationship***

mutually exclusive frameworks

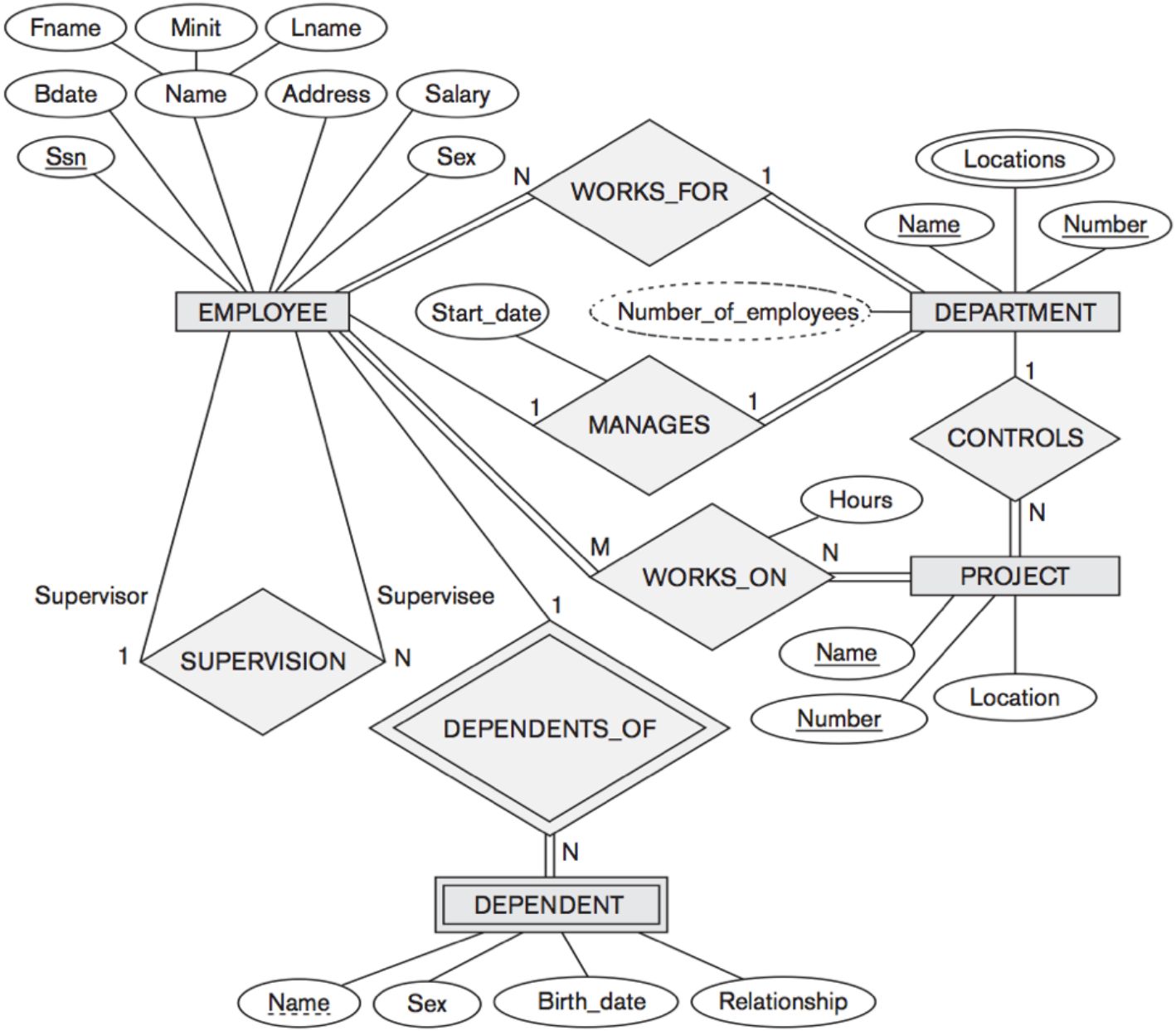


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

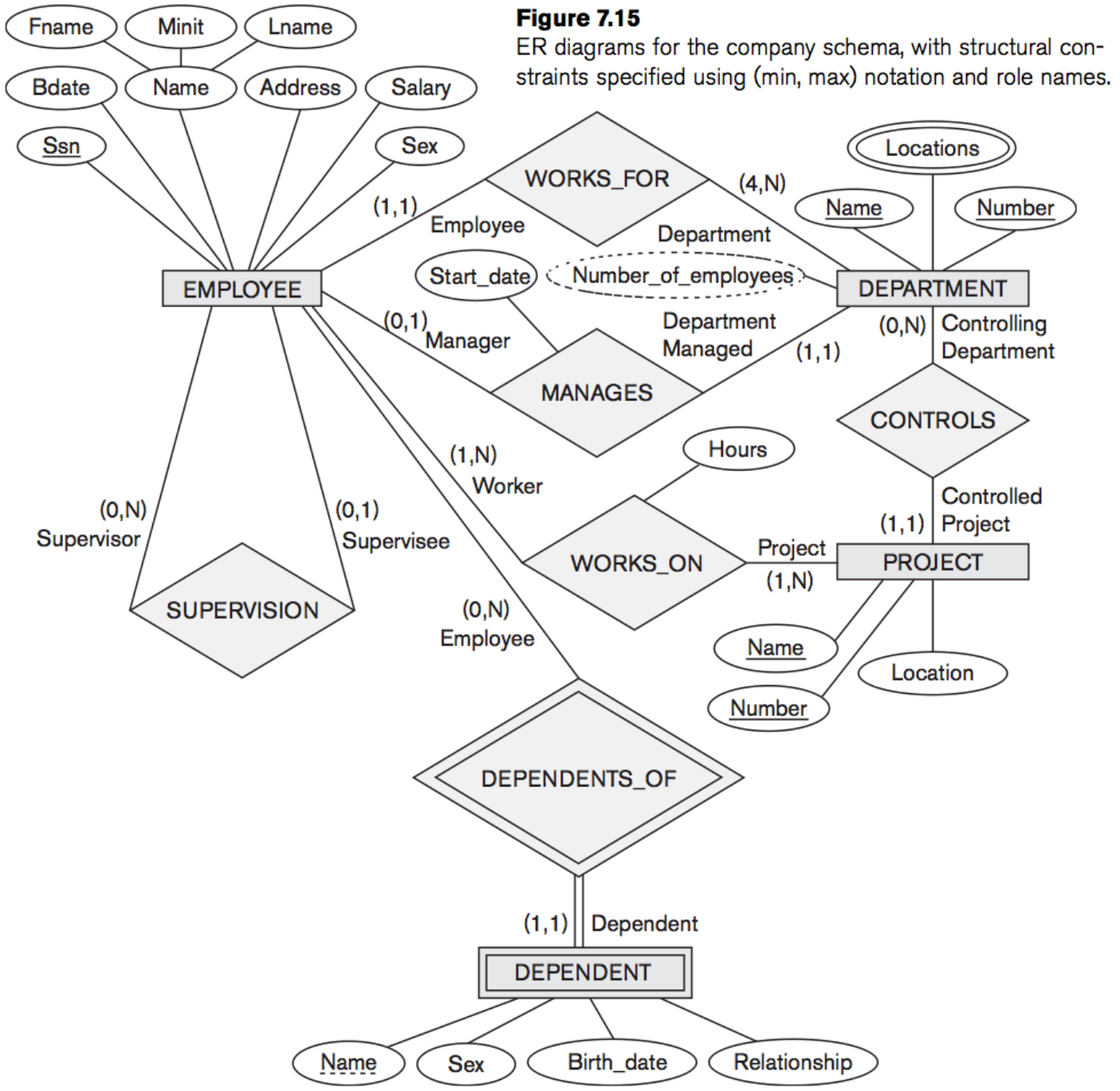
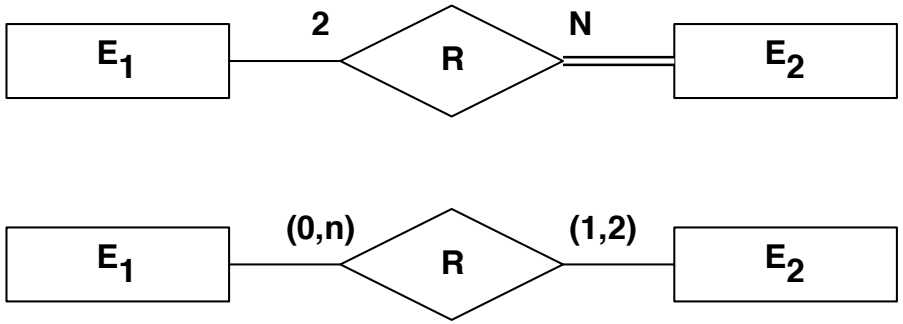


Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

Cardinality, Participation and (min,max)

- Cardinality ≡ max
- Participation ≡ min
- If there is **one** number, it is Cardinality (i.e. max)
- Participation has only two options
  - single line: min = 0
  - double line: min = 1



Nichts

Ch. 6 & 7: SQL

- **Q Type**: Short answer, Multiple Choice, True/False
- Required Syntax: `SELECT FROM`
- Optional Syntax
  - `WHERE`
  - `GROUP BY` (after `WHERE`)
  - `HAVING` (after `GROUP BY`)
- Aggregate Functions: `COUNT`, `SUM`, `MIN`, `MAX`, `AVG`
  - **Never** in `WHERE` clause
- `JOIN`: Natural Join, Inner Join, Outer Join

- NOT COVERED
  - ~~UPDATE, DELETE~~
  - ~~VIEW~~
  - ~~Admin functions: DESCRIBE, MODIFY, ALTER, DROP~~

Ch. 8: Relational Algebra & Relational Calculus

UTD

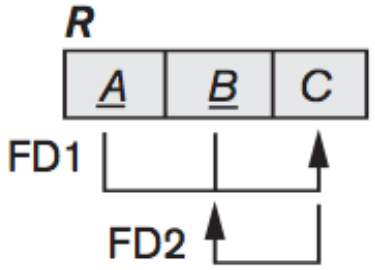
- **Q Type**: Multiple Choice, Multiple Answer, Matching, T/F
- Relational Algebra
  - Basic syntax (σ,π,ρ,⋈,⟕,⟖,⟗,*)
  - Set functions (∪,∩,⟍,−, *but not* ÷)
- ~~Relational Calculus (Boolean ops and Quantifiers)~~
  - ~~Tuple Relational Calculus~~
  - ~~Domain Relational Calculus~~

- Map ER diagram onto relation schema using 7-step algorithm
- Map EER diagram onto relation schema using 9-step algorithm (i.e. 7-step + 2-step algorithm).
- NOTE: Common Misunderstanding!!!
  - ER Step 7: Mapping of *N*-ary Relationship Types (p.296, 7ed.)
  - EER Step 8: Four different options
    - Superclass and subclasses
    - Subclass relations only
    - Single relation (superclass) with one type attribute
    - Single relation (superclass) with multiple type attributes

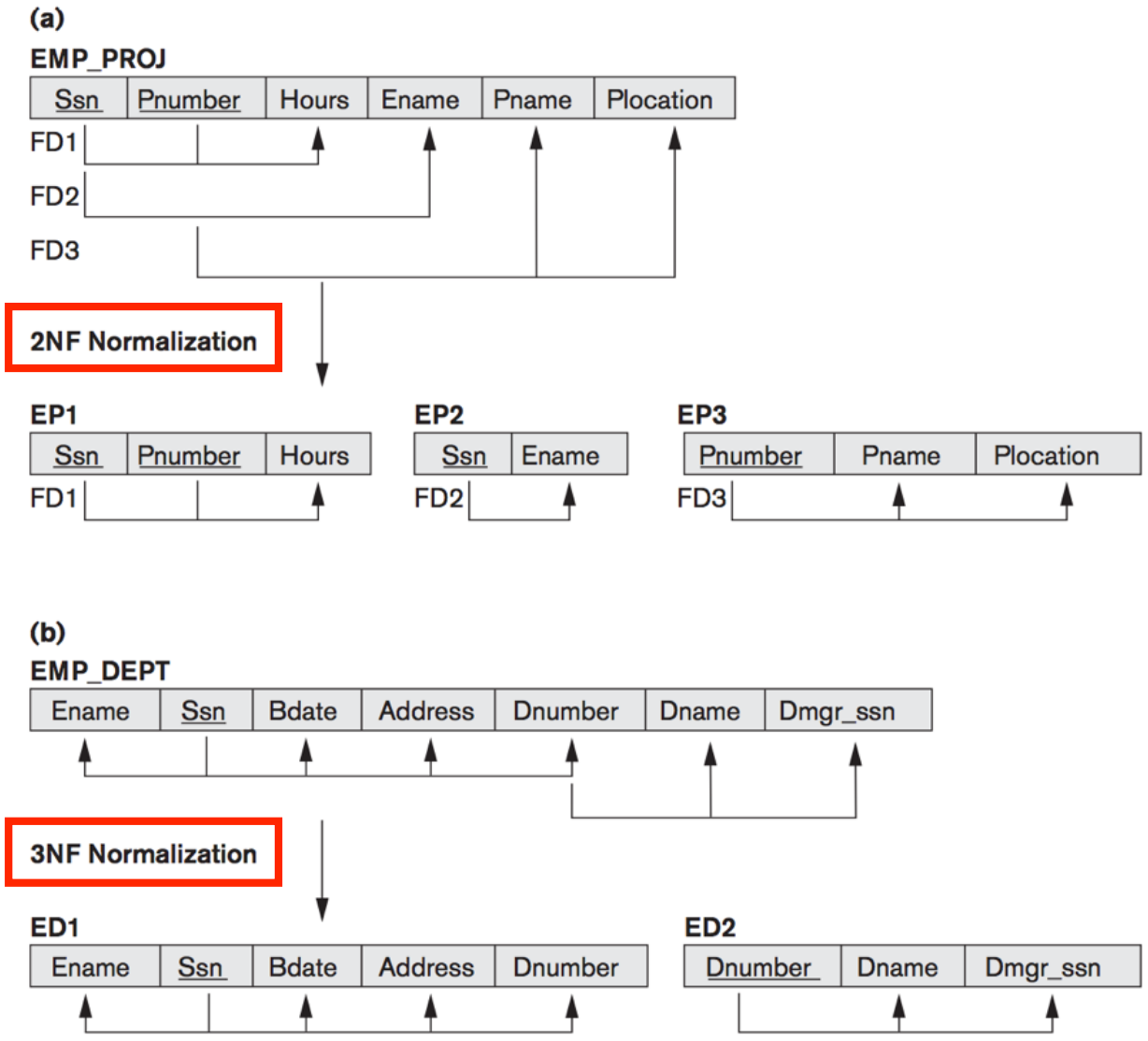
- **1NF** – The only attribute values permitted by 1NF are single atomic (or indivisible) values. That is, no attribute for a given tuple is multi-valued, i.e. “nested relations”
- **1NF** violations are based on violations of (**Data**)

Ch. 14: Functional Dependencies and Normalization

- Normal Forms Based on Primary Keys (**Schema**)
  - **2NF** – A relation schema *R* is in 2NF if every nonprime attribute *A* in *R* is fully functionally dependent on the primary key of *R*
  - **3NF** – Relation should not have a non-key attribute functionally determined by another non-key attribute (or by a set of non-key attributes). That is, there should be no transitive dependency of a non- key attribute on the primary key.
  - **BCNF**

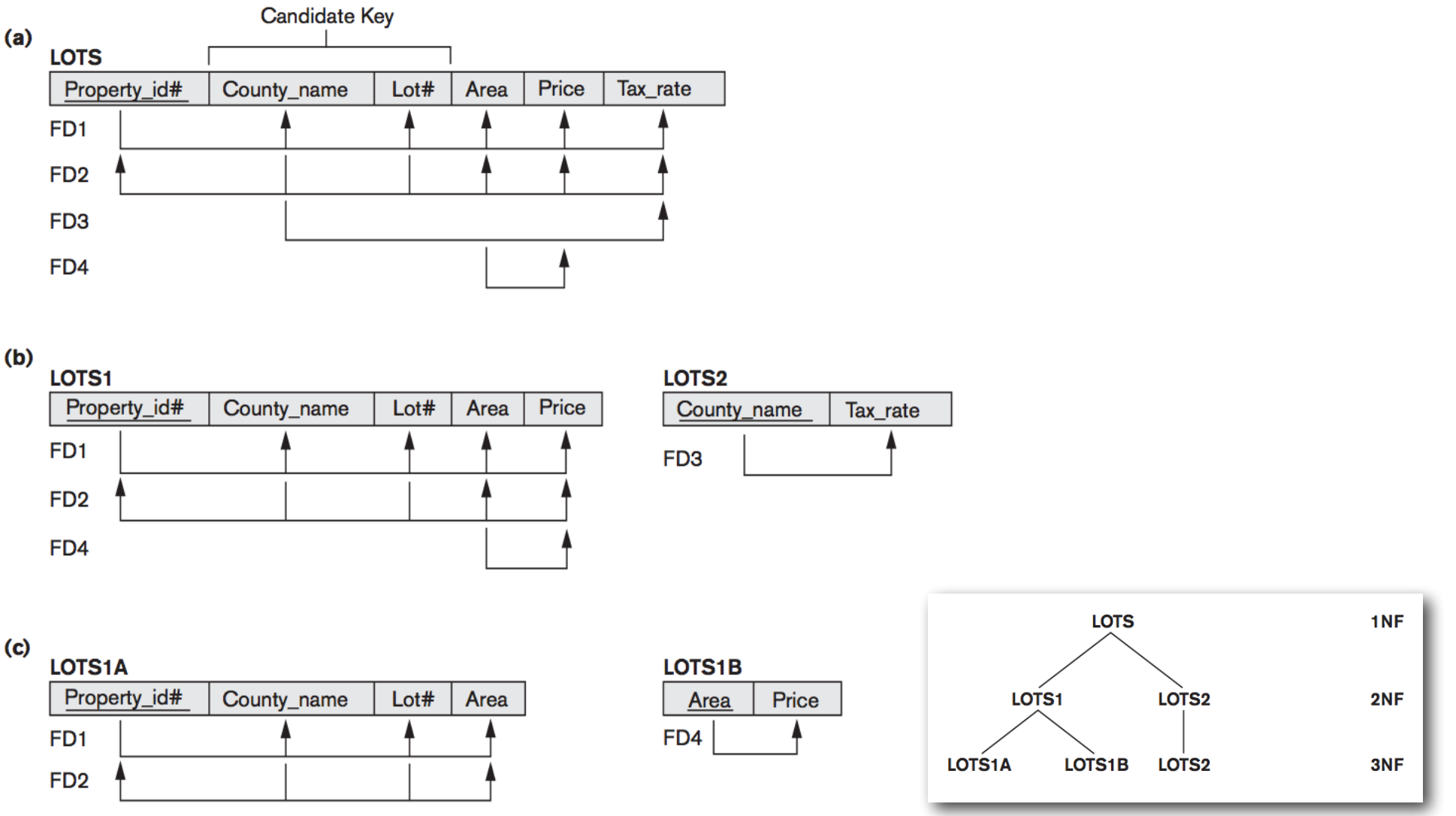


Normalization Example



Normalizing into 2NF and 3NF. (a) Normalizing EMP_PROJ into 2NF relations. (b) Normalizing EMP_DEPT into 3NF relations.

Normalization Example



- 4NF and 5NF normal forms are based on violation of both (**Data & Schema**)
- Note that if proper Chapter 9 schema design principles are observed, 4NF and 5NF violations will not occur
  - In practice however, schemas evolve over time
  - Later DBAs may not have access to original design requirements
- 15.6 – Multivalued Dependency and 4NF
  - Figure 15.15
- 15.7 – Join Dependencies and 5NF
  - Figure 15.15

Ch. 16: Disk Storage and Hashing

- Type: Multiple Choice, Multiple Answer, Matching, T/F
- ~~16.1 – Intro~~
- ~~16.2 – Secondary Storage Devices~~
- ~~16.3 – Placing File Records on Disk~~
- ~~16.5 – Operations on files~~
- ~~16.6 – Files of Unordered Records (Heap Files)~~
- ~~16.7 – Files of Ordered Records (Sorted Files)~~

Ch. 16: Disk Storage and Hashing

UTD

- Type: Multiple Choice, Multiple Answer, Matching, T/F
- 16.8 Hashing Techniques
  - Extendible hashing (p.612-614)
  - Dynamic hashing (p. 614)
  - Linear hashing (p. 614-616)
- ~~16.9 Other Primary File Organizations~~
- 16.10 – RAID (Problem)
  - Hex ⇔ Binary

Ch. 17: Indexing Structures for Files

- **Q Type**: Multiple Choice, Multiple Answer, Matching, T/F
- Single-level Ordered Indexes
  - Primary Indexes
  - Clustering Indexes
  - Secondary Indexes
- Multilevel Indexes
- Dynamic Multilevel Indexes: B-Trees and B^+-Trees
  - (**Hardcopy** Problem) *either* B *or* B+
  - Difference?
  - Properties?

ucsf.edu
online link

Chapter 18: Strategies for Query Optimization

- **Q Type**: Multiple Choice, Multiple Answer, Matching, T/F
- ~~§18.2 Algorithms for External Sorting~~
- §18.3 Algorithms for SELECT Operation (7 strategies)
  - i.e. filter: SQL WHERE
- §18.4 Implementing the JOIN Operation (4 strategies)
- §18.5 Algorithms for PROJECT Operation (7 strategies)
  - i.e. display: SQL SELECT
- ~~§18.6,~~
- ~~§18.7,~~
- ~~§18.8~~

Chapter 19: Query Optimization

- **Q Type**: Multiple Choice, Multiple Answer, Matching, T/F
- Selection (filter, e.g. SQL WHERE)
- Projection (display, e.g. SQL SELECT)
- Query Optimization (**Problems**)
  - **§19.1** - Heuristic Optimization (schema-based) (**p.700**)
  - **§19.5.2** - Cost-based Optimization (data-based)
    - Five examples (**p.721**)

Important

Important

- Steps in converting a query tree during heuristic optimization.
  - (a) Initial (canonical) query tree for SQL query Q.
  - (b) Moving SELECT operations down the query tree.
  - (c) Applying the more restrictive SELECT operation first.
  - (d) Replacing CARTESIAN PRODUCT and SELECT with JOIN operations.
  - (e) Moving PROJECT operations down the query tree.

Ch. 20: Transaction Processing

- **Q Type**: Multiple Choice, Multiple Answer, Matching, T/F
- 20.1 – Introduction to Transaction Processing
  - 20.1.3 – Four Potential Concurrency Issues
  - 20.1.4 – Six Types of Failures (pp. 750-1)
- 20.2 Transaction and System Concepts
  - States and Operations
  - System Log

- 20.3 Desirable Properties of Transactions (ACID)
  - Atomicity
  - Consistency Preservation
  - Isolation
  - Durability
- 20.4 Characterizing Schedules Based on Recoverability
  - Schedule conflicts
  - (Non-)Recoverable Schedule
  - Cascading Rollback
  - Strict schedule

- 20.5 – Characterizing Schedules Based on **Serializability**
  - **Algorithm 21.1 (Hardcopy problems involving Serializability graphs for transactions)**
  - Serializable? Show *ALL* equivalent serial schedules, e.g.
    - T1 → T4 → T3
    - T1 → T3 → T4
  - Non-serializable? Show *ALL* cycles
    - ***X***(T1 → T2), ***Y***,***Z***(T2 → T1)
    - ***Y***(T2 → T3); ***Z***(T3 → T4); , ***X***,***Z***(T4 → T2)