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)

mutually exclusive frameworks

  - Participation and Cardinality – property of an ***entity***
  - (min, max) – range of an entity's interaction with a ***relationship***
- Understand the correspondence between the next two slides.

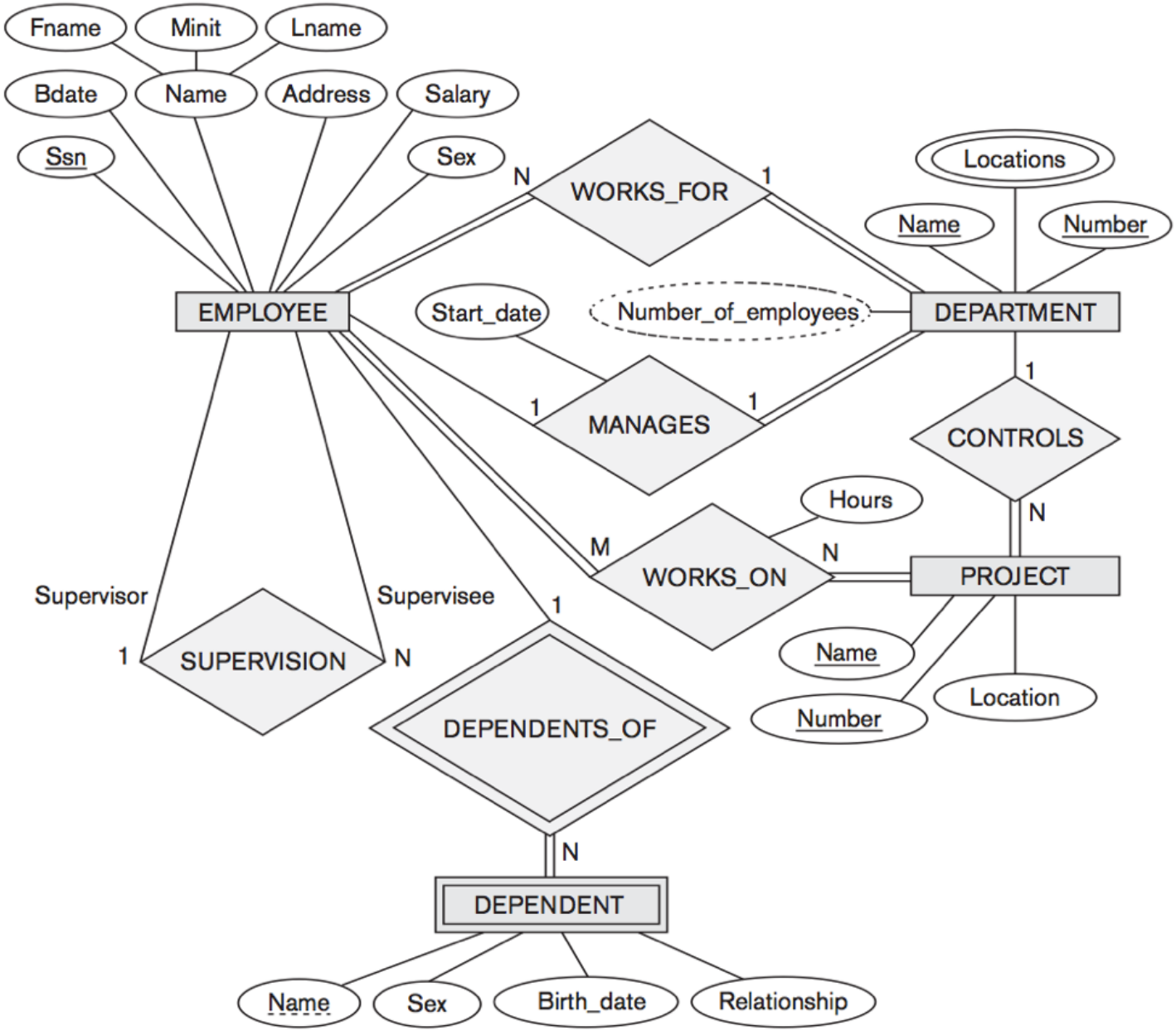
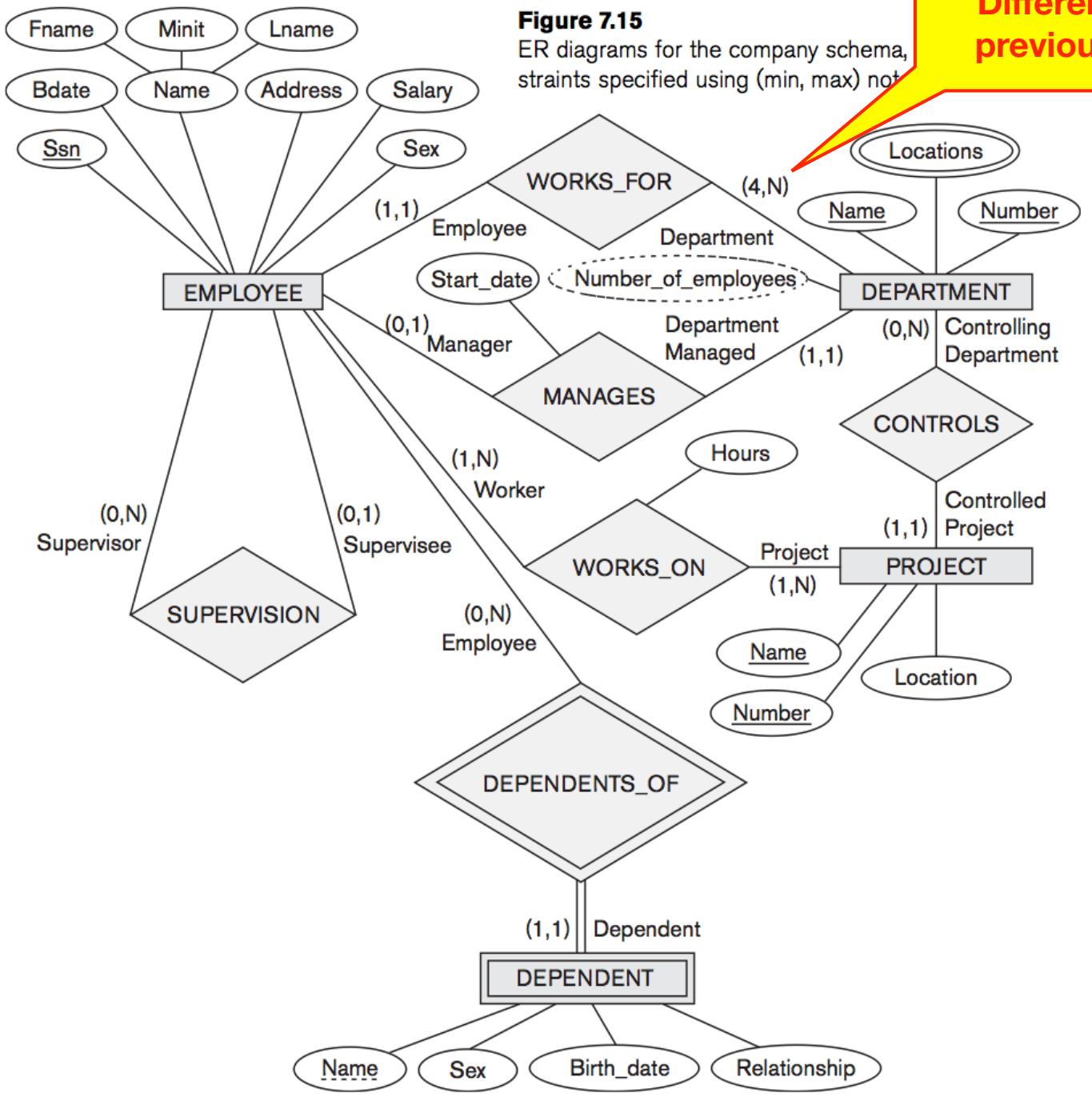


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

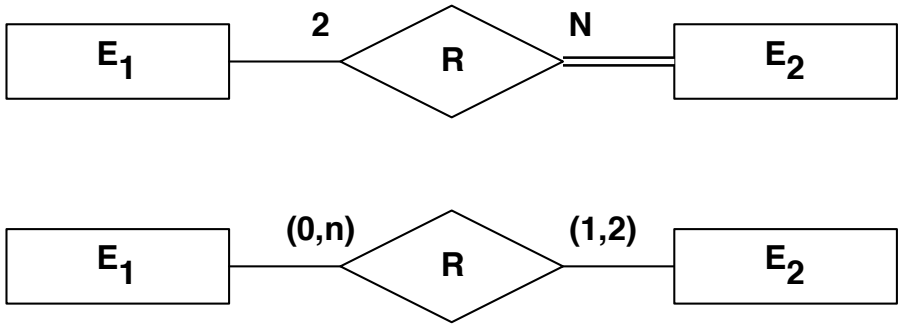
Figure 7.15
ER diagrams for the company schema,
straints specified using (min, max) not

Different than previous slide



Cardinality, Participation VS (min,max)

- CANNOT Mix the two on the same Relationship Type
  - 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
- Basic Syntax: `SELECT-FROM-WHERE`
- Optional Syntax
  - `GROUP BY` (**cannot come before** `WHERE`)
  - `HAVING` (**cannot come before** `GROUP BY`)
- Aggregate Functions: `COUNT`, `SUM`, `MIN`, `MAX`, `AVG`
  - **Cannot appear in** `WHERE` clause
  - Don't confuse `COUNT` and `SUM`!
- `JOIN`: Natural Join, Inner Join, Outer Join

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 (∪,∩,⟍,−, ÷)
- ~~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

- Be able to normalize into 1NF
- Be able to normalize → 2NF → 3NF → BCNF (**incl. 15.1**)
  - Schema diagram
  - Text schema
- Be able to normalize a relation and its data into either
  - 4NF (given ER/EER or data)
  - 5NF (given ER/EER)
- Both 4NF and 5NF violations may be detected using an accompanying ER diagram.
  - However, should also be able to detect 4NF violations based upon data analysis only.

Detect all minimal keys

Ch. 16: Disk Storage and Hashing

UTD

- 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
  - Hex-only XOR shortcut

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-tree *or* B^+-tree
  - Insert
  - Delete

ucsf.edu
online link

Chapter 18: Strategies for Query Optimization

UTD

- **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~~

You will not be asked to reproduce a memorized list of these algorithms. However, you should be familiar with them if asked questions about them.

Chapter 19: Query Optimization

- **Q Type**: Multiple Choice, Multiple Answer, Matching, T/F
- Query Optimization (**Problems of BOTH**)
  - **§19.1** - Heuristic Optimization (schema-based) (**p.700**)
    - Five steps heuristic (*MEMORIZE*)
  - **§19.5.2** - Cost-estimation Optimization (*Cost functions given*)
    - Five examples (**p.721**)
    - Extra study problems

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

Ch. 20: Transaction Processing

- 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 20.1 (Hardcopy problems involving Serializability graphs for transactions)**
  - Be able to create a conflict graph from a schedule of transactions presented in either:
    - A table of Transactions/Operations
    - A text list of Transactions/Operations

- 20.5 – Characterizing Schedules Based on **Serializability**
  - If a schedule is serializable, show *ALL* equivalent serial schedules, e.g.
    - T1 → T4 → T3
    - T1 → T3 → T4
  - If a schedule is *not* serializable, show *ALL* cycles
    - ***X***(T1 → T2), ***Y***,***Z***(T2 → T1)
    - ***Y***(T2 → T3); ***Z***(T3 → T4); , ***X***,***Z***(T4 → T2)\
- ~~20.6 Transaction Support in SQL~~