Object-Oriented and Object-Relational Databases

CS 5513

1. Evolution of DBMS - History

- File system
- Hierarchical DBMS
  - Using the hierarchical data model: a hierarchical tree structure
  - Data are represented by collections of *records*
  - Relationships between data are represented by *links*
  - Records are organized as collections of trees

Example:

1. Evolution of DBMS - History (cont.)

- Network DBMS
  - Using the network data model: a graph structure
  - Data are represented by collections of *records*
  - Relationships between data are represented by *links*
  - Records are organized as arbitrary graphs.
  - Example:

1. Evolution of DBMS - History (cont.)

- Disadvantages of hierarchical and network data models:
  - Complex programs has to be written to answer even simple queries based on navigational record-oriented access;
  - There was minimal data independence;
  - There was no widely accepted theoretical foundation.

1. Evolution of DBMS - History (cont.)

- Relational DBMS
  - Using the relational data model
  - A relational database: a set of relations/tables
  - Relation:
    - attributes/columns; every attribute is atomic (has one single value);
    - tuples/rows/records; a tuple is identified by its key
  - Using a declarative query language: SQL
  - Example:

1. Evolution of DBMS - History (cont.)

- Object-oriented DBMS
  - Using an object-oriented (OO) data model: set of classes
  - Class:
    - Attributes: simple/complex/relationships
    - Methods/operations
    - Encapsulation
  - Class hierarchy: inheritance
    - Superclass
    - Subclass
  - Example: See Section 3

1. Evolution of DBMS - History (cont.)

- Object-relational DBMS
  - Using the relational data model with added complex attributes and other OO features to accommodate complex objects and operations
  - Example: See Section 4

- Other DBMS based on later data models: NoSQL, XML, etc.

1. Evolution of DBMS - Applications

- Simple (earlier) database applications
  - Small data items
  - Uniformity in data
  - Fixed length data
  - Atomic fields
  - Short transactions

1. Evolution of DBMS – Applications (Cont.)

- Complex (later) database applications
  - CAD Engineering drawings
    - Components
    - Versions
    - Relationships between components and versions
  - Software engineering
    - Source code
    - Module specifications
    - Relationships between modules
    - Definitions and usage of variables/parameters
    - Development history
  - Multimedia data
    - Text, audio, video, image
  - Hypertext data
    - Web database
    - Want to retrieve documents via links and structures
  - Social media data
    - Graph data
    - Want to retrieve data via links and weights on links

2. Advantages & Disadvantages of Relational DBMS

- Advantages:
  - Strong mathematical foundation
  - Supports SQL, a simple query language
- Disadvantages:
  - Atomic attributes => cannot store complex values
  - Fixed predefined data types => limited
  - Semantic overloading => cannot distinguish relationship representation from data representation
  - Normalization => creation of relations that may not represent real-world entities => may need join operations to get back to the real-world entities => inefficient
  - Fixed operations
  - Impedance mismatch
    - Query in SQL, but applications are in other languages (C, Java, etc.)
    - Need conversion to map different data types from SQL and app languages => inefficient
  - Other problems
    - Short transactions
    - Schema change is difficult

3. Rules to convert from an ER Diagram to OODB

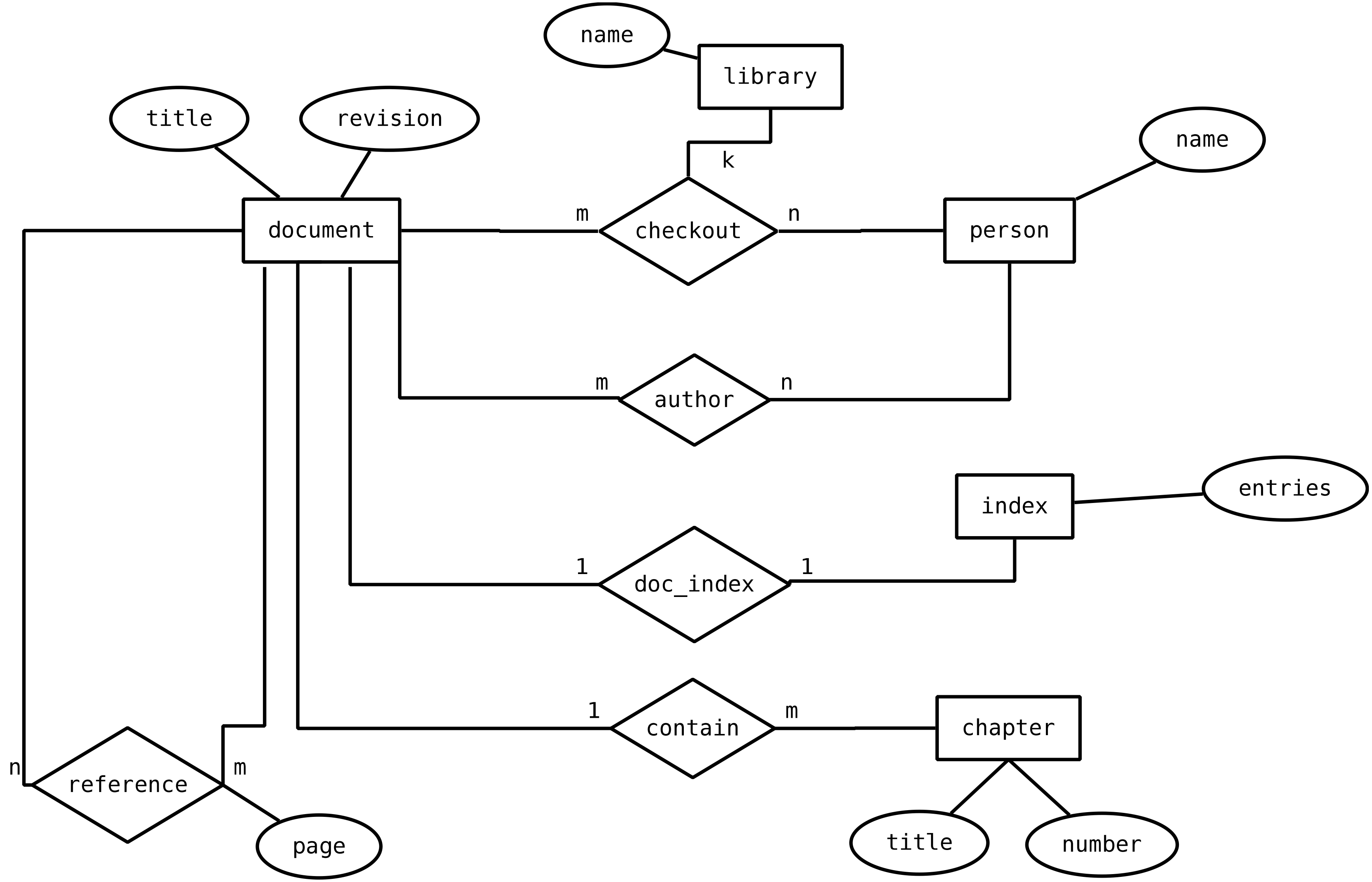
- ER:
  - Set of entities
  - Set of relationships between entities
  - Entity: has attributes
- OODB:
  - Classes:
    - Attributes: simple/complex/relationship
    - Operations/methods

Rules to convert from an ER Diagram to OODB (Cont.)

- 1-1 Relationship Set
- m:1 Relationship Set
- m:n Relationship Set
- Non-Binary Relationship Set

Rules to convert from an ER Diagram to OODB (Cont.)

- Example:


name
library
title
revision
name
k
document
m
checkout
n
person
m
author
n
index
entries
1
doc_index
1
1
contain
m
chapter
n
reference
m
page
title
number

4. Object-Relational DBMS

- Extending relational model by
  - Adding OO features
  - Adding constructs to SQL

4. Object-Relational DBMS - Oracle

- Example:

5. Object ID (OID)

- OID: persistent and unique; created by the system when an object is created
- What is the difference between OID and a primary key?
  - Answer:
- Why does the system use OID?
  - Answer:

5. Object ID (cont.)

- How is an OID implemented?
- Answer:

6. Storage Structure

- What is the storage structure of an object in an OODBMS?
- Answer:

7. Schema Evolution

- Different types of schema evolution:
  - Change to structure of a database
  - Add/delete/modify the data type of an attribute
  - Add/delete/modify a method
  - Change a class hierarchy
  - Change a composite hierarchy
- Database reorganization due to schema evolution: what are the problems and what are the solutions?
- Answers:

8. OODB Clustering

- What is the purpose of database clustering?
  - Answer:
- What are some possible approaches for database clustering in OODB?
  - Answer:

9. Advantages and Disadvantages of OODB

- Advantages:
  - Enriched modeling capability
  - Extensibility => allows user-defined data types
  - Removal of impedance mismatch
  - Support for schema evolution
  - Support for long-lived transactions
  - More expressive query languages
  - Applicability to complex advanced database applications

Advantages and Disadvantages of OODB (Cont.)

- Disadvantages:
  - Lack of experience
  - Lack of universal data model (however the ODMG (Object Data Management Group) has proposed an object model that has become the de factor standard for OODBMS)
  - Complexity
  - More complex for implementing components such as query processing and concurrency control.

End of Topic 5