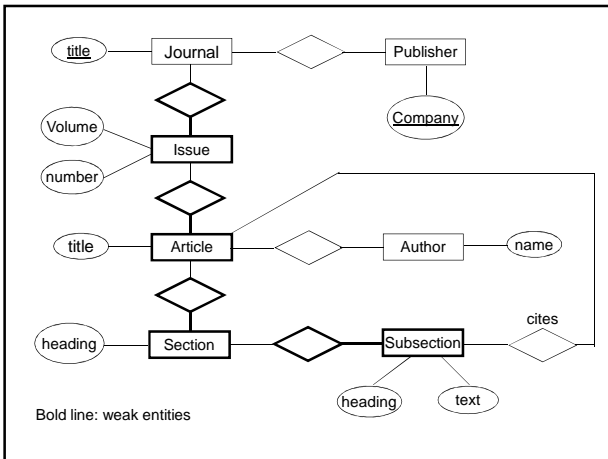


Tutorial 6: Complex Data Models

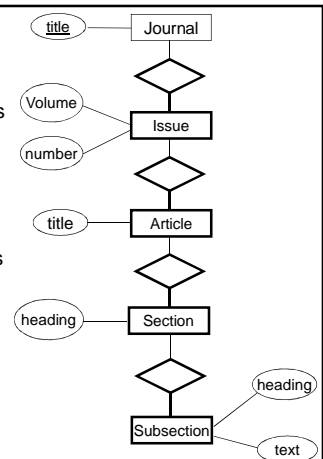
INFS3200/7907
Advanced Database Systems

TEVAL



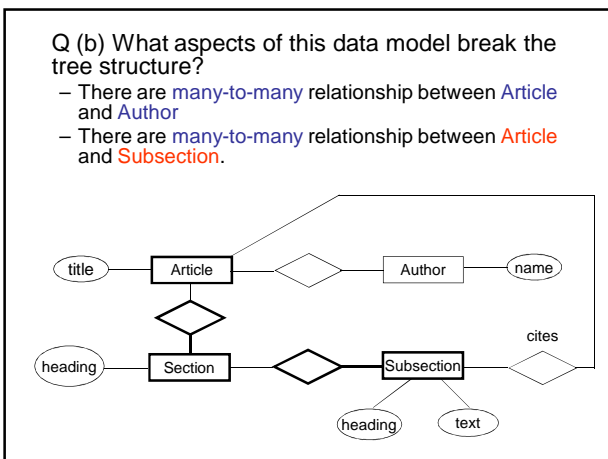
Q (a) What aspects of this data model are tree-structured?

- The chain of weak entities from **Journal** to **Subsection**.
- As a tree, it is a **directed acyclic graph**.
- There are **one-to-many** relationship between the owner and its dependents since there must be one starting node – **root (Journal)** for a tree and one parent to many children.



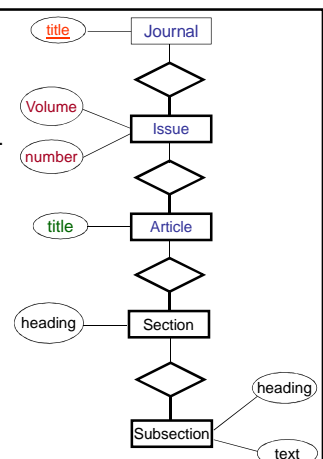
Q (b) What aspects of this data model break the tree structure?

- There are **many-to-many** relationship between **Article** and **Author**
- There are **many-to-many** relationship between **Article** and **Subsection**.



Q (c) How is the **Article** entity identified?

- **Journal.title**, **Issue.volume**, **Issue.number**, **Article.title**.



Q (d) How would you navigate this data model to assemble the contents of an issue to be sent to typesetter?

- Select an instance of **Issue**, then assemble the corresponding instances of **Article**, **Section** and **Subsection**.
- The associated instances of **Author**.

Q (d) Does the tree-structured aspect of this data model help?

- The tree structure helps in that if we selected an instance of **Issue**, the dependent entities would come automatically.
- The only problem is that we need an instance of **Author**, which may be shared by many instances of **Issue**.

Q (e) How would you navigate this data model to assemble a citation index (Article cites article)?

- The citation index is a relationship between **Article** and itself.
- The dependent entities such as **Section** and **Subsection**, are no longer required. Instead, we need to construct a derived cites relationship, taking the union of the population of cites over all dependent instances of **Subsection**.

Question (e) Does the tree-structured aspect of this data model help?

- It doesn't help.
- In fact, the derivation of **Article.cites** removes the tree structured aspects of the data.

Q (f) What sort of queries would use the attributes **Section.heading**, **Subsection.heading**, **Subsection.text**?

- Probably search engine style queries restricted to headings as heading are considered to be rich in content-specific terms.

Q (g) What parts of this data model would it make sense to package as an XML-structured data type?

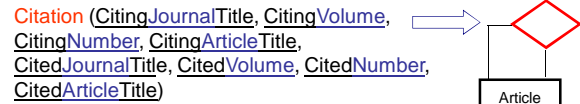
- Certainly **Article** and its dependent entities.
- Provision would have to be made to refer to **Author** (assuming one author per article).
- Probably also **Issue** packaged with its dependent entity **Article**.

Q (h) Suppose you packaged the XML-structured data type as the value set of a relational attribute. What would a relational schema for the data model look like?

- Article (JournalTitle, IssueVolume, IssueNumber, ArticleTitle, AuthorName, Content: XML Doc) (assuming one author per article)
- Citation (CitingJournalTitle, CitingVolume, CitingNumber, CitingArticleTitle, CitedJournalTitle, CitedVolume, CitedNumber, CitedArticleTitle)
- Author (Name)
- Journal (Title, PublisherCompany)
- Publisher (Company)

Q (i) Suppose you wanted to distribute the data, partly to a search engine and partly to a relational system supporting citation index queries. What table and fragments (possibly replicated) would go where? Why?

- Relational system (SQL Query): Publisher, Journal, Author, Citation and Article except Content to the system supporting citation index queries, because citation index queries require all this data.



Q (i)

- Search engine (IEEE Xplorer):
 - Article, with identifying attributes necessarily replicated and Author name also replicated, to the search engine.
 - Identifying attributes replicated because they form the primary key.
 - AuthorName replicated because search engine queries can combine author name with (index) terms from the content.

Article (JournalTitle, IssueVolume, IssueNumber, ArticleTitle, AuthorName, Content: XML Doc)

