Introduction to databases from a bioinformatics perspective

Misha Taylor

Overview

- Background
- Flat text files
- ISAM Databases
- SQL/Relational Databases
- Object-Oriented/XML Databases
- The Future

What is “informatics”

- Derived from the French word informatique
- Tends to get associated with specific application areas
  - Medical informatics
  - Bioinformatics
  - Nursing informatics
  - Business informatics (MIS/IMS)
  - Social-science informatics

A good definition

- Informatics is the science that deals with information, its structure, its acquisition and its use
Informatics is not computer science

- Emphasis is on the acquisition, modeling, and representation of data and knowledge – not on the building of computational artifacts
- However, understanding computational artifacts very much helps to illustrate the underlying principles
- It’s impossible to provide examples of the principles independent of any application domain

Informatics is about systems modeling

- Creating and enhancing models of application areas
- Identifying relationships among models
- Creating algorithms that can automate domain tasks

Informatics is about knowledge and its representation

- Conceptualizing the knowledge required to drive applications
- Building useful, maintainable systems
- Developing better methods for management of knowledge within organizations and scientific communities

Problem-solving knowledge automates specific tasks

Domain knowledge
+ Problem-solving method
Intelligent behavior
Databases & Knowledge

- Databases are a tool for storing knowledge
  - Data
  - Relationships

A parable: Amazon vs. CDNOW

Database concepts

- Entity – thing that is being stored and is representative of something in the real world
- Attribute – descriptor of an entity
- Relationships

Flat text files

- Flat text files can act as the basis of these concepts (entity, attribute, relationships)
But...

- Most applications require that specific information can be quickly and efficiently retrieved.
- Sometimes critical that performance does not degrade as more entities are added.
- Flat text files don’t always fulfill these requirements, especially when there are many entities and/or relationships.

Solution – indexes and keys

- Performance requirement is most often met through the use of indexes or keys.
- More sophisticated database paradigms:
  - ISAM
  - SQL/Relational
  - Object-oriented/XML

What is ISAM?

- Indexed Sequential Access Method
- Used in:
  - Cobol
  - Btrieve
  - dBase
  - FoxPro
  - Faircom c-tree Plus

ISAM

- Entities are records
- Attributes are understood to be data stored starting at a specific offset in the record
- Data & indexes are stored in files
- Applications are responsible for maintaining relationships and knowing which set of records is in which file.
ISAM (contd.)

- ISAM database/library manages index and data files

SQL/Relational

- Entities are represented by rows
- Collections of entities are represented as tables
- Collections of entities and attributes may be arbitrarily defined at runtime.
- Applications are not responsible for maintaining relationships, but are responsible for conforming to the model

SQL/Relational (contd.)

- Incorporates an easy-to-use query language - SQL

Object-oriented/XML

- Ties data and behavior together - entities are objects, which have both attributes and methods
- XML is used as a portable persistence mechanism
- Applications can discover data and relationships at runtime – need not conform to an application-specific model
## Comparing ISAM, SQL/Relational, and OO/XML

<table>
<thead>
<tr>
<th>ISAM</th>
<th>SQL/Relational</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>User operates on file</td>
<td>User operates on a file within a database</td>
<td>User operates on objects</td>
</tr>
<tr>
<td>The file may contain multiple entity types</td>
<td>The table has a single defined entity type</td>
<td>Objects may encapsulate multiple entity types</td>
</tr>
</tbody>
</table>

## Comparing ISAM, SQL/Relational, and OO/XML (contd.)

<table>
<thead>
<tr>
<th>ISAM</th>
<th>SQL/Relational</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>All instances of an entity type are contained in one file</td>
<td>All instances of an entity type are maintained in one table</td>
<td>Instances of an entity type may occur in multiple objects</td>
</tr>
<tr>
<td>Every instance of a given entity type has the same composition.</td>
<td>Every instance of a given entity type has the same composition.</td>
<td>Every instance of a given entity type may have a different composition.</td>
</tr>
</tbody>
</table>

## Comparing ISAM, SQL/Relational, and OO/XML

<table>
<thead>
<tr>
<th>ISAM</th>
<th>SQL/Relational</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>The application is responsible for extracting attributes from entity instances</td>
<td>The DBMS is responsible for extracting attributes from entity instances</td>
<td>The data contains the description of the attributes for any particular entity instance.</td>
</tr>
<tr>
<td>Relationships are maintained by the application code.</td>
<td>Relationships are maintained by the DBMS.</td>
<td>Relationships are described within the data itself.</td>
</tr>
</tbody>
</table>

## Comparing ISAM, SQL/Relational, and OO/XML (contd.)

<table>
<thead>
<tr>
<th>ISAM</th>
<th>SQL/Relational</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexes are granular to the file level</td>
<td>Indexes are granular to the DBMS-understood table level</td>
<td>Indexes must be granular to the element level.</td>
</tr>
</tbody>
</table>