APPENDICES

Appendix A

Legacy System Models: Hierarchical and Network

Chapter 1 briefly introduced the hierarchical and network data models. These data models were widely used before the adoption of the relational data model. The earlier generation of data systems composed of these two data models are generally known as legacy system models. The information legacy in many enterprises is still residing on databases supporting these two data models.

As you know, the relational data model is the one that is widely used now because of its superiority and strength. Nevertheless, you need to review the earlier data models for two reasons. First, hierarchical and network databases are still in use. Second, a brief review of the earlier data models will enhance your appreciation of the relational model and its superior features.

In this appendix, we will summarize the basic concepts of the two legacy system models and provide a mapping of the semantic data model with each of the two models.

HIERARCHICAL DATA MODEL

This model is the oldest of the database models. Unlike the relational data model or the network data model, the hierarchical data model did not originate with any systematic document or research results. The network model is based on the stipulations by the DBTG of CODASYL. On the other hand, the relational model originated from the research specifications of Dr. Codd.

Database Design and Development: An Essential Guide for IT Professionals by Paulraj Ponniah
ISBN 0-471-21877-4

The hierarchical data model, however, has been derived from IBM's Information Management System (IMS) database that has been in wide use. In an IMS database, the data structures are placed in a hierarchical top-down arrangement.

Summary of Basic Concepts

- Data is organized in the form of data segments being arranged as an inverted tree.
- Two fundamental modeling concepts make up the hierarchical data model: segment types and parent-child relational types.
- Segments are linked in parent-child relationships, as a top-down structure.
- A child segment is linked to just one parent segment whereas a parent segment may be linked to many child segments.
- A single segment, called the root segment, exists at the top of the inverted tree. The root segment does not participate as a child in any parent-child relationship.
- All other segments participate in relationships as child segments.
- A segment without any child segments is known as a leaf segment.
- For any segment type in the hierarchical tree, a single path exists from the root.
- Each segment type generally represents an entity type of the organization. Data fields in the record types denote the attributes of the entity type.
- Logical links between related segments are implemented through physical addresses (pointers) in the segment itself.

Semantic to Hierarchical Data Model

The following indicates the mapping of components of the semantic data model into components of the hierarchical data model.

Object Sets

- Object set is transformed into segment type.
- The name of the object set becomes the name of the record type. CUSTOMER object is transformed into CUSTOMER segment type.
- The object instances of an object set become the occurrences of the transformed segment type.
- The complete set of object instances becomes the total set of segment occurrences.

Attributes

- Attributes of an object are transformed into the data fields of the record type.
- The names of the attributes become the names of the data fields.
- Domain of values of each attribute becomes the domain for the corresponding data field.

- A single-valued or derived attribute is transformed into a single field.
- For a composite attribute, as many fields are used in the record type as the number of component attributes in the composite attribute.

Instance Identifiers

- The attribute or attributes forming the instance identifier maps into a set of sequence fields.
- Segment occurrences are sorted based on the values of the sequence fields.
- A sequence field may be declared to be "unique" in the sense that no duplicate values are allowed in the sequence field. This is similar to the function of a primary key.
- Only when a sequence field is defined to be "unique" it has a function similar to that of a primary key. Sequence fields may also be declared to be "non-unique."

Relationships Each relationship in the hierarchical data model becomes a binary link between the parent and the child segment types.

One-to-One or One-to-Many Relationship

- For each object set in the semantic data model, create a segment type.
- If two object sets A and B are in a one-to-one or one-to-many relationship, create a tree structure with A as the parent segment type and B as the child segment type with link between A and B.
- The object set on the "one" side of the relationship becomes the parent segment type and the object set on the "many" side of the relationship becomes the child segment type.

Many-to-Many Relationship

- For each object set in the semantic data model, create a segment type.
- If two object sets A and B are in a many-to-many relationship, map this as two one-to-many relationships.
- Create a tree structure with A as the parent segment type and B as the child segment type with link between A and B.
- Create another tree structure with B as the parent segment type and A as the child segment type with link between B and A.

NETWORK DATA MODEL

The Conference on Data Systems Languages (CODASYL), the organization comprising of vendor representatives and user groups, developed the language COBOL. In the late 1960s, CODASYL appointed a subgroup known as the Database Task Group (DBTG) to develop standards for database systems. DBTG published a preliminary report in 1969. Based on revisions and suggestions made for improvement, DBTG published a revised version of the report in 1971.

Essentially, the network data model is based on the 1971 DBTG report. This data model conforms to a three-level database architecture: conceptual, external, and internal levels. A number of commercial database systems were developed to implement the network data model.

Summary of Basic Concepts

- Data is organized in the form of records being arranged as a network of nodes.
- Two fundamental modeling concepts make up the network data model: record types and set.
- Two record types are linked as a set. The set expresses the one-to-one or one-to-many relationship between two record types.
- A set expressing the relationship between two record types consists of a member record type and an owner record type.
- One owner record type may be part of different sets with different member record types.
- Similarly, one member record type may have multiple owner record types.
- A network consisting of one-to-one or one-to-many relationships is known as a simple network. A complex network, on the other hand, contains many-to-many relationships also.
- Each record type generally represents an entity type of the organization. Data fields in the segment types denote the attributes of the entity type.
- An instance of a set type represents one occurrence of the entity represented by the record type.
- Logical links between related records are implemented through physical addresses (pointers) in the record itself.

Semantic to Network Data Model

The following indicates the mapping of components of the semantic data model into components of the network data model.

Object Sets

- Object set is transformed into record type.
- The name of the object set becomes the name of the segment type. CUSTOMER object is transformed into CUSTOMER record type.
- The object instances of an object set become the occurrences of the transformed record type.
- The complete set of object instances becomes the total set of record occurrences.

Attributes

- Attributes of an object are transformed into the data fields of the field type.
- The names of the attributes become the names of the data fields.

- Domain of values of each attribute becomes the domain for the corresponding data field.
- A single-valued or derived attribute is transformed into a single field.
- For a composite attribute, as many fields are used in the segment type as the number of component attributes in the composite attribute.

Instance Identifiers

- The attribute or attributes forming the instance identifier maps into a set of fields termed the key of the record type.
- Record occurrences may be sorted based on the values of the key fields.
- No special significance is attached to the key fields as all navigation within the data model is through physical pointers.

Relationships Each relationship in the network data model becomes a binary link between the owner and the member record types.

One-to-One or One-to-Many Relationship

- For each object set in the semantic data model, create a record type.
- If two object sets A and B are in a one-to-one or one-to-many relationship, connect them with A as the owner record type and B as the member record type.
- The object set on the "one" side of the relationship becomes the owner record type and the object set on the "many" side of the relationship becomes the member segment type.

Many-to-Many Relationship

- For each object set in the semantic data model, create a segment type.
- If two object sets A and B are in a many-to-many relationship, map this as two one-to-many relationships.
- Create a link record type C to contain key values of both A and B.
- Define two sets, one with A and the link record type C, the other with B and the link record type C.
- The set with A as the owner record type and C as the member record type represents one of the transformed one-to-many relationship. The set with B as the owner record type and C as the member record type represents the other transformed one-to-many relationship.