DBMS note 3

Manik Chand Patnaik (M.Tech.)
Lecturer, (C.S.E), Roland Institute of Technology.
Wednesday, January 24 – 2007.

Data Models

User's Requirement: The user of the DBMS wants to represent real world data about real or logically existing entities‡. So the ultimate goal of the DBMS user is to store data about various aspects of the real world scenario with proper integrity constraints, relationships and semantics§.

- **Data models** are a collection of conceptual tools(structural high level data description constructs) for describing data, data relationships, data semantics and data constraints. All these without going into storage level details. There are three different groups:
    1. Object-based Logical Models. (Object based Semantic Data Models)
    2. Record-based Logical Models.
    3. Physical Data Models.

Object-based Logical Models *(Object based Semantic Data Models)*

A database management system can take several approaches to manage the data. Each approach constitutes a data model. Though the DBMS software hides many storage level details it is nevertheless closer to how a DBMS stores the data; not how actually the data is available in the real world. An Object based **Semantic Data Model** is a more abstract high level data model which describes the real meaning of data.

Object based models are very popular because of their flexible structuring capabilities. Various data-integrity constraints can explicitly be specified which is a big plus point.

1. Object-based logical models:
    i. Describe data at the conceptual and view levels with relationships.
2. Provide fairly flexible structuring capabilities.
    i. Allow one to specify data constraints explicitly.
    ii. Over 30 such models, including . . .
3. Entity-relationship model.
4. Object-oriented model.
5. Binary model.
6. Infological model.

‡ Entity and Entity Set : Entity is a real world object with a physical or logical existence. An entity can be either **concrete** or **abstract**. Entities which are physically present like book, student and employee are called concrete entities whereas abstract entities have a logical existence like project, presentation, history etc.
A set of entities of the same type that have the same attributes are grouped together to form an entity set.

§ Semantics is logical meaning. Here in the user's requirements logically meaningful database is a data representation which closely matches with the real world. Database is also referred as a miniworld by many authors because a database can represent only a part of the properties of real world entities.

7. Functional data model.
8. At this point, we'll take a closer look at the **entity-relationship (E-R)** and **object-oriented** models.

The E-R Model

This model is a fairly high level data model based on real world objects called entities and their relationship with each other.

1. The entity-relationship model is based on a perception of the world as consisting of a collection of basic **objects** (entities) and **relationships** among these objects.
    - An **entity** is a distinguishable object that exists, either physically or logically
    - An entity set is a set of entities of the same type
2. Each entity has associated with it a set of **attributes**[$] describing it. e.g. *number* and *balance* for a *bankacct* entity.
3. A **relationship** is an association among several entities. e.g. A *customer_bankacct* relationship associates a customer with each bank account he or she has. A relationship can also have descriptive attributes to describe about the relationship.
4. The set of all entities or relationships of the same type is called the **entity set** or **relationship set**. When E_n entities of entity set E have the same type of relationship with O_n entities of an other entity set O then both entity sets can have a relationship set.
5. The number of entity sets participating in a relationship is known as the degree of relationship set.
   If the number is 2 then it is called binary relation, if the number is 3 then the relationship is ternary.
6. Another essential element of the E-R diagram is the **mapping cardinalities**, which express the number of entities to which another entity can be associated via a relationship set. Mapping of cardinalities is done in case of binary relationship sets.
   There are four types of mappings present:-
    - One to One
      e.g. *employee* and *epf_acct*

$ Attributes are the properties of the entity. They are of different types:-

a) Simple and Composite Attributes
   - Attributes which cannot be broken down to smaller constituent parts are called simple or atomic attributes. e.g. *RegNo* of a student, *BranchCode*, *College_Name* etc.
   - Attributes that can be devided into subparts each part conveying some independent meaning are called composite attributes. e.g. *Sname* has first name, mid name and last name. Address has building name, street name, city, pincode etc.

b) Single Valued and Multi-Valued attributes
   - Attributes having a single possible value for a particular entity is single valued attribute. e.g. *RegNo*, *EPF_Number* etc.
   - Attributes for which more than one value is possible for a single entity are called multi-valued attributes. e.g. *phone_No*, *Address*, *Bank_Acct_Number* etc.

c) Derived Attributes
   - Attributes which are derived from some other attribute are called derived attributes. e.g. *Age* derived from *DOB* and current date.

d) Null Valued attributes (Many authors don't consider this as a type) When a particular attribute of a specific entity does not have a value, the DBMS stores NULL at that place. Null is neither zero nor space. It signifies that the value is either unknown or missing.

- One to Many
  e.g. *person* and *bank_acct*
- Many to One
  e.g. *student* and *college*
- Many to Many
  e.g. *student* and *teacher*

7. The overall logical/conceptual structure of a database can be expressed graphically by an **E-R diagram**:
8. **rectangles**: represent entity sets.
9. **ellipses**: represent attributes.
10. **diamonds**: represent relationships among entity sets.
11. **lines**: link attributes to entity sets and entity sets to relationships.

The Object-Oriented Model

1. The object-oriented model is based on a collection of objects, like the E-R model.
2. An object contains values stored in **instance variables** within the object. Unlike the record-oriented models, these values are themselves objects. Thus objects contain objects to an arbitrarily deep level of nesting.
3. An object also contains bodies of code that operate on the the object. These bodies of code are called **methods**.
4. Objects that contain the same types of values and the same methods are grouped into **classes**. A class may be viewed as a type definition for objects. Analogy: the programming language concept of an abstract data type.
5. The only way in which one object can access the data of another object is by invoking the method of that other object. This is called **sending a message** to the object.
6. Internal parts of the object, the instance variables and method code, are not visible externally. Result is two levels of data abstraction. For example, consider an object representing a bank account. The object contains instance variables *number* and *balance*. The object contains a method *pay-interest* which adds interest to the balance. Under most data models, changing the interest rate entails changing code in application programs. In the object-oriented model, this only entails a change within the *pay-interest* method.
7. Unlike entities in the E-R model, each object has its own unique identity, independent of the values it contains: Two objects containing the same values are distinct. Distinction is created and maintained in physical level by assigning distinct object identifiers.

OO Model, a supplementary reading:-

Relational database technology is not good at handling the real world data. Any data to be represented using relational model has to confine to tables. So relational database design is actually a process of trying to figure out how to represent real-world objects using tables in such a manner that there is good performance and at the same time proper data integrity is enforced. This process involves mapping between the object based model and the relational model. Object Oriented database Modeling is a part of the application design process. The same object classes used by the programming languages are what will be used by the object database. As the models are consistent there is no need to transform from one model to another model.

Object Oriented model represents an entity as a class. A class represents both entity attributes as well as the behaviour of the entity. For example a student class will have the details like *s_name*, *reg_no*, *phone*, etc alongwith procedures that a student has like *register*, *log_in*, etc. Within an object the class attributes take specific values which distinguishes one object instance from another whereas the behaviour is shared among all members of the class.

As the OO Model does not restrict the attribute values to a set of pre-defined data types such as character, integer etc. Instead the values can be objects so any kind of data can be stored in the database. It may be text, image, video or video-player it does not matter.

Advantages:-

Capabilities to handle large no of different data types: As discussed in previous paragraph, there is no restriction on the attribute data types.

Combination of OOP and OODB:This is the most significant benefit of the technology as Object Oriented Programming can be clubbed together with an Object Oriented Database to produce an integrated application. The OODB provides the persistence for the object instances.

Access to information: OODB represents relationships explicitly. Both navigational and associative access to information is provided. As the complexity of inter-relationship between information stored within database increases, the greater advantages are visible to the user.

OO Features: The OO features like inheritance and polymorphism makes the objects more closely resemble with the real world entities. And this also makes solution development easier because there is code-reuse and integrity of data.

Disadvantages:-

Difficult to maintain: In a real-world situation the data model is not static. It changes to adopt to the organizational needs. So the objects may need to be modified periodically and this makes data migration a routine job.

Implementation Problems: As each object instance is quite a complex piece of information comprising of data an behaviour, storing and retrieving them is not an easy job so handling objects is very resource intensive. So it is not suitable for small and simple applications. It is popular in E-Commerce, Engineering, Animal and Plant sciences where complexity of the data items(objects) is quite high.

Record-based Logical Models

In record based models the database is organized in fixed-format records. A fixed number of fields are defined for each record type. Each field usually have a fixed length.

1. Record-based logical models:
    - Also describe data at the conceptual and view levels.
2. Unlike object-oriented models, are used to
    - Specify overall logical structure of the database, **and**
    - Provide a higher-level description of the implementation.
3. Named so because the database is structured in fixed-format records of several types.
4. Each record type defines a fixed number of fields, or attributes.
5. Each field is usually of a fixed length (this simplifies the implementation).
6. Data Integrity constraints cannot be explicitly specified in respect to each record.
7. Separate languages associated with the model are used to express database queries and updates.
8. The three most widely-accepted models are the **relational, network**, and **hierarchical**.

The Relational Model

- Data and relationships are represented by a collection of **tables**.
- Each **table** has a number of columns with unique names, e.g. *customer*, *acct_no*.

The Network Model

- Data are represented by collections of records.
- Relationships among data are represented by links.
- Organization is that of an **arbitrary graph**.

The Hierarchical Model

- Similar to the network model.
- Organization of the records is as a collection of **trees**, rather than arbitrary graphs.

The relational model does not use pointers or links, but relates records by the values they contain. This allows a formal mathematical foundation to be defined.

Physical Data Models

1. Are used to describe data at the lowest level. i.e. The storage structure and the access mechanism. They describe how data is stored in a computer, representing record structures, record ordering and access paths.
2. Very few models, e.g.
    - Unifying model.
    - Frame memory model.

The Relational Model

The relational model uses tables to represent the data and the relationship among those data. Each table has multiple columns and each column is identified by a unique name.

Advantages:-

Structural Independence: Changes made in the database structure does not affect the DBMS's capability to access data. The database users are oblivious of the details of the data storage because the relational database does not depend upon the navigational data access system.

Simplicity: The relational model is the simplest model at the conceptual level. It allows the designer to concentrate on the logical view of the database, leaving the physical data storage details.

Ease of design, implementation, maintenance and use: Due to the inherent features of data independence and structural independence, the relational model makes it easy to design, implement, maintain and use the databases.

Adhoc query capability: One of the main reasons for the popularity of this model is the presence of powerful and flexible query capabilities. The query language for relational databases is SQL, a 4^{th} generation language to extract relevant data using simple commands.

Disadvantages:-

Ease of Design can lead to bad design: As relational databases are easy to design, it can result in the development and implementation of poorly designed systems which were made to satisfy current needs rather than designed similar to real world data representation.

Information Island phenomenon: Though a relational model offers capabilities to establish relationship among data, no one prevents if a department creates its own database and does not integrate itself with the rest of the system. So the problems of data inconsistency and redundancy will still be there despite the usage of a relational database system.

Network Model

In the network model data are represented by collection of records and relationships among data are represented by links. A network model permits a record to have more than one parent – the resultant is a graph or network. The network model was developed specifically to handle non-hierarchical relationships.

Advantages:-

Simplicity: Conceptually simple and easy to design.

Ability to handle more relation types: Network model can handle one to many and many to many relation types.

Ease of data access: In a network database terminology, a relationship is a set. Each set comprises of two types of records, an owner record and a member record. In a network model an application can access an owner record and all the member records within a set.

Data Integrity: In a network model, no member can exist without an owner. A user must therefore first define the owner record and then the member record. This ensures data integrity.

Data Independence: The network model clearly separates the programs and the complex physical storage details. The application programs work independently of the data. Any changes made in the data characteristics do not affect the application program.

Disadvantages:-

System complexity: In a network model, data are accessed one record at a time. This makes it essential for the database designers,administrators and programmers to be familiar with the internal data structures to gain access to the data. So an user-friendly DBMS cannot be created using the network database model.

Lack of structural independence: Making structural modifications to the database is very difficult in the network database model as the data access method is navigational. Any changes made to the database structure requires the application programs to be modified before they can access data. Though the network database model achieves data independence, it still fails to achieve structural independence.

Hierarchical Model

Hierarchical database model is one of the oldest database models. It organizes data elements in parent-child type of hierarchy. It is identical to the network model in the sense that data and relationships among data are represented by records and links respectively. The hierarchical model is overly restrictive in defining relationships.

Advantages:-

Simplicity: Since the database is based on the hierarchical structure, the relationship is logically simple. Thus the design of database is simple.

Data Security:This is the first data model to offer data security which is enforced not by the programmer but by the DBMS.

Data Integrity: Hierarchical model is based on the parent-child relationship so there is always a link between the parent segment an the child segments. The child segments are automatically referenced by the parent segment so this model has some data integrity.

Efficiency: This model is very efficient when the database contains a large number of one-to-many relationships. And transactions which need these relationships.

Disadvantages:-

Complexity in Implementation: Although this model is conceptually simple and easy to design, it is very complex to implement.

Database management problems: If any changes are made to the database structure then it is required to make the necessary changes in all the application programs that access the data.

Structural Dependence: Hierarchical databases use physical storage paths to navigate to different data segments. So if the physical structure is changed then the application must be altered too.

Implementational limitation: Most of the common relationships do not conform to the one-to-many relationship type. Many-to-many type is very difficult to implement.

Key Concepts related to Relational Model:-

The data item which the computer uses to identify a row in a database system is

referred to as key. Key is a single attribute or a set of attributes of an entity set that can uniquely identify an entity. There are different types of keys:-

1. Primary Key
   It is also called the entity identifier. For example the registration number uniquely identifies a student.
2. Secondary Key
   It identifies all those records which have a special property. It means that it classifies the entities. For example BranchID of a student.
3. Super Key
   A super key includes any number of attributes which uniquely identify an entity. If RollNo and CollegeCode can together uniquely identify a student then it is a superkey. Taken BranchID together will also uniquely identify a student. Registration Number can uniquely identify a student as well as Registration Number and RollNo can together uniquely identify a student. All of these are called Super Keys.
   The primary key is a minimal SuperKey.
4. Candidate Key
   If there are more than one possible candidates for the primary key then all those keys are called Candidate Keys. For example RollNo and CollegeCode can together uniquely identify a student and Registration_Number can also uniquely identify a student. Both are candidate keys.

NOTE:

- A set of fields is a superkey if:
  - No two distinct tuples can have same values in all key fields
- A set of fields is a key for a relation if :
  - It is a superkey
  - No subset of the fields is a superkey
- what if >1 key for a relation?
  - One of the keys is chosen (by DBA) to be the primary key.
  - Other keys are called candidate keys.

    E.g. reg_no is a key for Students.

    What about name?

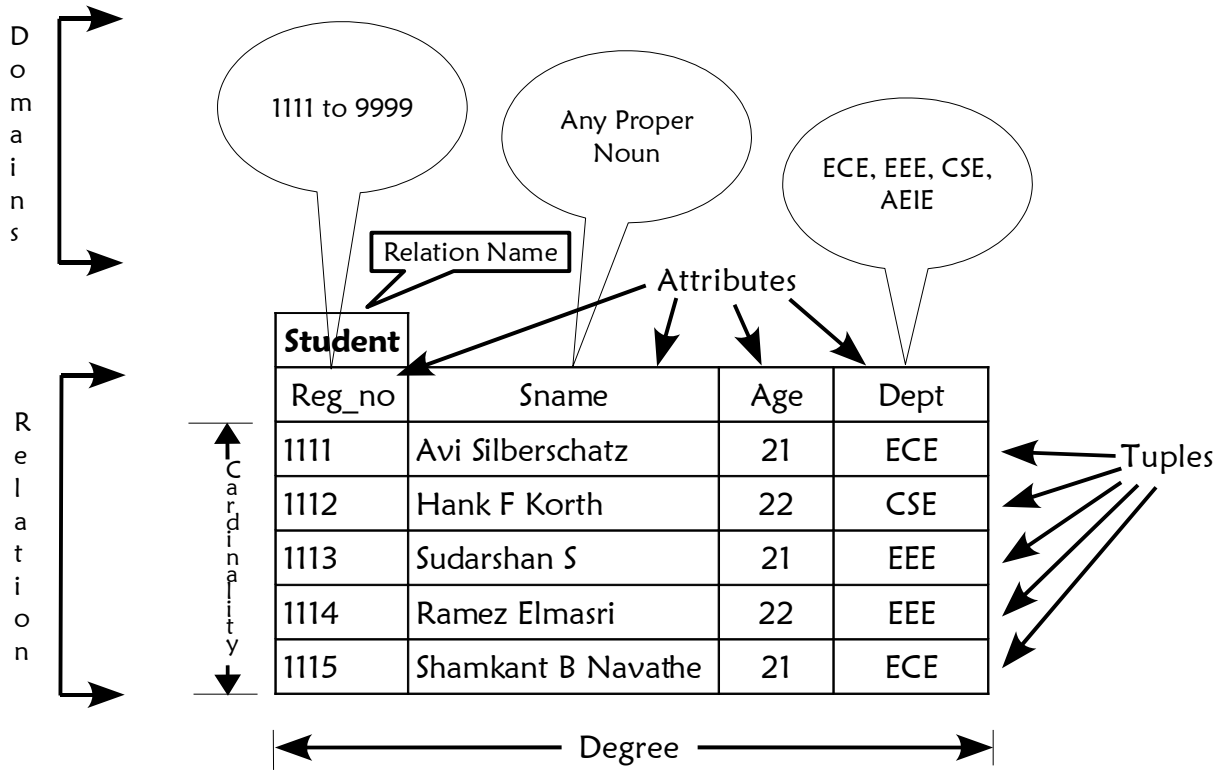
    The set {reg_no,sname} is a superkey.

Foreign key: Set of fields in one relation that is used to `refer' to a tuple in another relation.

Must correspond to the primary key of the other relation. Like a `logical pointer'.

If all foreign key constraints are enforced, referential integrity is achieved (i.e., no dangling references.)

Diagram showing different concepts related to relational model



In this diagram I have tried to show the conceptual points related to the Relational Model. The special point here is that the table is called the relation.

1. Tuple : A row in the table
2. Attribute : The column of the table
3. Cardinality of Relation : Number of rows in the table
4. Domain : A set of permitted values for an attribute
5. Degree of Relation : Number of attributes

NOTE:

Relational database: a set of relations.

Relation: made up of 2 parts:

Schema : specifies name of relation, plus name and type of each column.
E.g. Students(reg_no: number, sname: varchar[20], age: integer, dept: varchar[4])

Instance : a table, with rows and columns.
#rows = cardinality (above table's cardinality = 5)
#fields = degree / arity (above table's arity = 4)

You can think of a relation as a set of rows or tuples.
i.e., all rows are distinct even if they have the exact data.