Chapter 2

Data Models

2.1 DATA MODELS

- Underlying structure of the database is called as data models.
- It is a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.

Different types of data models are:

- Hierarchical model
- Network model
- Object oriented model
- Object relational model
- Relational model
- Entity relationship model

2.2 HIERARCHICAL MODEL

The hierarchical data model organizes data in a tree structure. In this model, each entity has only one parent but can have several children. Only one entity at the top of the hierarchy is called as Root. The structure is based on the rule that one parent can have many children but children are allowed only one parent. Linkages are only possible vertically but not horizontally or diagonally, i.e. there is no relation between different trees at the same level unless they share the same parent.

Advantages

- High speed of access to large datasets.
- **Data security:** Hierarchical model was the first database model that offered the data security that is provided and enforced by the DBMS.
- **Efficiency:** The hierarchical database model is very efficient when the database contains a large number of transactions using data whose relationships are fixed.
- The model allows easy addition and deletion of new information. Data at the top of the Hierarchy is very fast to access. It is very easy to work with the model because it works

well with linear type data storage such as tapes. The model relates very well to natural hierarchies such as assembly plants and employee organization in corporations. It relates well to anything that works through one to many relationships.

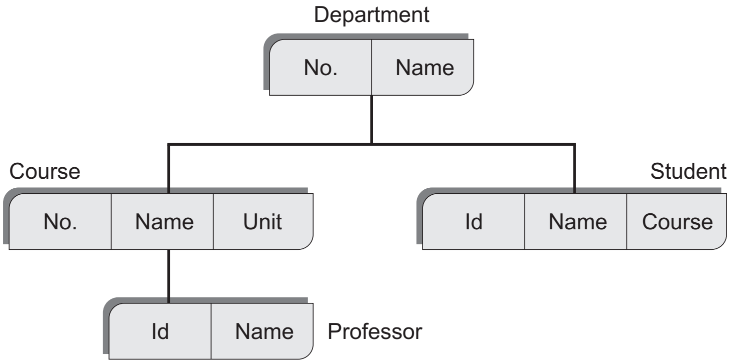


Fig. 2.1. *Hierarchical model.*

Disadvantages

- Implementation complexity
- Database management problems
- Lack of structural independence.
- This model cannot be used for more sophisticated relationships. It requires data to be repetitively stored in many different entities. The database can be very slow when searching for information on the lower entities.
- We no longer use linear data storage mediums such as tapes so that advantage is null.
- Searching for data requires the DBMS to run through the entire model from top to bottom until the required information is found, making queries very slow. It can only model one to many relationships; many to many relationships are not supported.

2.3 NETWORK MODEL

The Network Data Model is also known as the “CODASYL Data Model” or sometimes as “DBTG Data Model.” The model is based on directed graph theory. The network model replaces the hierarchical tree with a graph thus allowing more general connections among the nodes. The main difference of the network model from the hierarchical model is its ability to handle many-to-many (n : n) relationship or in other words, it allows a record to have more than one parent.

Advantages

- Conceptual simplicity
- Capability to handle more relationship types
- Data independence

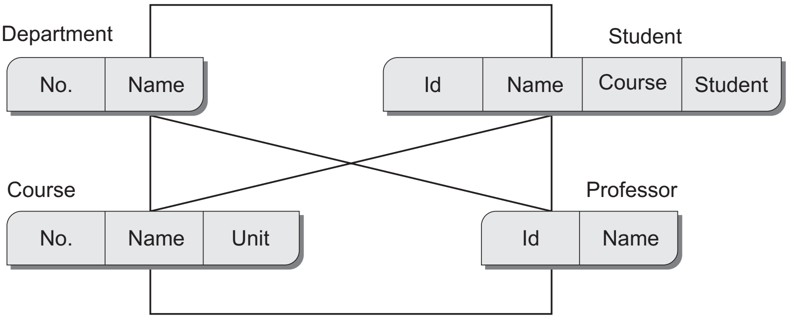


Fig. 2.2. *Network model.*

Disadvantages

- Detailed structural knowledge is required
- Lack of structural independence

2.4 OBJECT-ORIENTED MODEL

Object DBMSs add database functionality to object programming languages. Object DBMSs extend the semantics of the C++, Smalltalk and Java object programming languages to provide full-featured database programming capability, while retaining native language compatibility. A major benefit of this approach is the unification of the application and database development into a seamless data model and language environment. As a result, applications require less code, use more natural data modeling, and code bases are easier to maintain. Object developers can write complete database applications with a modest amount of additional effort.

The object-oriented model is based on a collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate in the object, these bodies of code are called methods. Objects that contain the same types of values and the same methods are grouped together into classes.

Advantages

- Applications require less code
- Applications use more natural data model
- Code is easier to maintain
- It provides higher performance management of objects and complex interrelationships between objects
- Object-oriented features improve productivity
- Data access is easy.

2.5 OBJECT RELATIONAL MODEL

Object relational database management systems (ORDBMSs) add new object storage capabilities to the relational systems at the core of modern information systems. These new facilities integrate management of traditional fielded data, complex objects such as time-series and geospatial data and diverse binary media such as audio, video, images, and applets. By encapsulating methods with data structures, an ORDBMS server can execute complex analytical and data manipulation operations to search and transform multimedia and other complex objects.

A system that includes both object infrastructure and set relational extenders. Object-relational systems combine the advantages of modern object-oriented programming languages with relational database features such as multiple views of data and a high-level, non-procedural query language. Some of the object-relational systems available in the market are IBM DB2 universal server, oracle corporation's oracle 8, Microsoft Corporations SQL server 7 and so on.

2.6 RELATIONAL MODEL

The relational model was introduced by Dr. E. F. Codd in 1970. The relational model represents data in the form of two dimensional tables. The organization of data into relational tables is known as the logical view of the database. Software such as Oracle, Microsoft SQL Server, Sybase, are based on the relational model.

2.6.1 Characteristics of Relational Model

- The relational model eliminated all parent child relationships and instead represented all data in the database as simple row/column tables of data values.
- A relation is similar to a table with rows/columns of data values.
- Each table is an independent entity and there is no physical relationship between tables.
- Most data management systems based on the relational model have a built-in support for query languages like ANSI SQL or QBE. These queries are simple English constructs that allow adhoc data manipulation from a table.
- Relational model of data management is based on set theory.
- The user interface used with relational models is non-procedural because only what needs to be done is specified and not how it has to be done.

2.6.2 E. F. Codd's Laws for a Fully Functional Relational Database Management System

CODD's 12 rules define an ideal relational database which is used as a guideline for designing relational database systems. Though no commercial database system completely conforms to all 12 rules, they do interpret the relational approach. The CODD's 12 rules are as follows:

Rule 0: Foundation rule: The system must qualify as relational both as a database and as a management system.

Rule 1: The information rule: All information in the database must be represented in one and only one way (that is, as values in a table).

Rule 2: The guaranteed access rule: All data should be logically accessible through a combination of table name, primary key value and column name.

Rule 3: Systematic treatment of null values: A DBMS must support Null Values to represent missing information and inapplicable information in a systematic manner independent of data types.

Rule 4: Active online catalog based on the relational model: The database must support online relational catalog that is accessible to authorized users through their regular query language.

Rule 5: The comprehensive data sublanguage rule: The database must support at least one language that defines linear syntax functionality, supports data definition and manipulation operations, data integrity and database transaction control.

Rule 6: The view updating rule: Representation of data can be done using different logical combinations called Views. All the views that are theoretically updatable must also be updatable by the system.

Rule 7: High-level insert, update, and delete: The system must support set at the time of insert, update and delete operators.

Rule 8: Physical data independence: Changes made in physical level must not have any impact and require a change to be made in the application program.

Rule 9: Logical data independence: Changes made in logical level must not impact and require a change to be made in the application program.

Rule 10: Integrity independence: Integrity constraints must be defined and separated from the application programs. Changing Constraints must be allowed without affecting the applications.

Rule 11: Distribution independence: The user should be unaware about the database location i.e. whether or not the database is distributed in multiple locations.

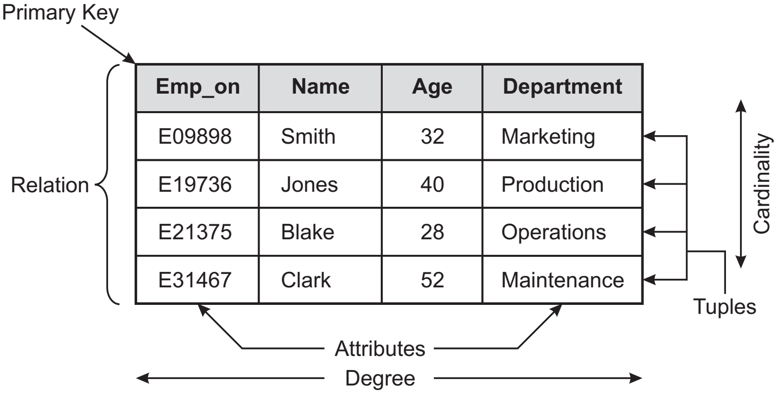
Rule 12: The non subversion rule: If a system provides a low level language, then there should be no way to subvert or bypass the integrity rules of high-level language. Of all the rules, rule 3 is the most controversial. This is due to a debate about three-valued or ternary logic. Codd's rules and SQL use ternary logic, where null is used to represent missing data and comparing anything to null results in an unknown truth state. However, when both Booleans or operands are false, the operation is false; therefore, not all data that is missing is unknown, hence the controversy.

2.6.3 Principle Components of Relational Model

The relational model consists of following three basic components:

1. Data structure
2. Data integrity
3. Data manipulation

1. **The Relational Data Structure**: The relational data structure is shown in Fig 2.3. It is based on the employee relation.



Emp_on	Name	Age	Department
E09898	Smith	32	Marketing
E19736	Jones	40	Production
E21375	Blake	28	Operations
E31467	Clark	52	Maintenance

Fig. 2.3. *The relatinal data structure.*

Structural terminology summary

Formal relational term	*Informal Equivalence*
Relation	Table
Tuple	Row or record
Cardinality	Number of rows
Attribute	Column or field
Degree	Number of columns
Primary key	Unique identifier
Domain	Pool of legal values

2. **Relational Integrity:** Integrity constraints means when changes made to the database by authorized users that should not result in a loss of data consistency.

   There are two main types of integrity constraints.

   - Domain constraints
   - Referential integrity

   Domain Constraints

   - Domain constraints specify the set of values that can be associated with an attribute.
   - Domain constraints are tested easily by the system whenever a new data item is entered into the database.
   - Domain constraints also prohibit use of null values for particular fields.

   Referential Integrity

   - A value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation. This is called referential integrity.

   Referential Integrity in the E-R Model: Referential integrity constraints arise frequently. If we derive our relational database scheme by constructing tables from E-R diagrams then every relation arising from a relationship set has referential integrity constraints.

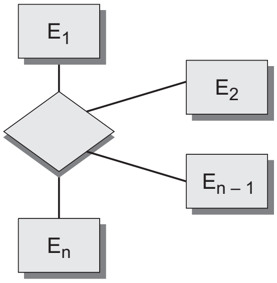


Fig. 2.4. *Referential integrity in the E-R model.*

As shown in Fig. 2.4, an n-array relationship set R, relating entity sets E_1, E_2,……..E_n. Let K_i denote the primary key of E_i. The attributes of the relation scheme for relationship set

R include $K_1 \cup K_2 \cup \ldots\ldots \cup K_n$. Each K_i in the scheme for R is a foreign key that leads to a referential integrity constraints.

Referential Integrity in SQL: Using SQL primary key, candidate key, and foreign key are defined as part of the create table statement as given below

Example: **Create** table deposit (br-name char (15), acc-no char(10), cust-name cher (20) not null, balance integer, primary key (acc-no, cust-name), foreign key (branch-name), Foreign key(cist-name) references customer);

Null: "Null represents a value for an attribute that is currently unknown or is not applicable for this tuple."

Other integrity constraints are:

- Entity integrity
- Enterprise constraints.

Entity Integrity: "In a base relation, no attribute of a primary key can be null".

A primary key is used to identify tuples uniquely. This means that no subset of the primary key is sufficient to provide unique identification of tuples. Therefore, primary key should not be null.

Enterprise Constraints: These are additional rules specified by the uses or database administrators of a database.

3. **Data Manipulation:** The manipulated part of the relational algebra model consists of a set of operators known collectively as the relational algebra together with relational calculus.

Advantages of Relational Model

- Structural independence
- Conceptual simplicity
- Design, implementation, maintenance and usage ease
- Good for adhoc requests
- It is simpler to navigate
- Greater flexibility.

Disadvantages of Relational Model

- Significant hardware and software overheads
- Not as good for transaction process modeling as hierarchical and network models
- May have slower processing than hierarchical and network models.

2.7 COMPARISON BETWEEN THE VARIOUS DATABASE MODELS

Model	*Data element organization*	*Relationship representation*	*Identity*	*Access language*
Hierarchical	Files, records	Tree	Record based	Procedural
Network	Files, records	Graph	Record based	Procedural
Relational	Tables	Foreign key concept	Value based	Non-Procedural
Object-oriented	Objects	Logical containment	Record based	Procedural
Object- Relational	Objects	Relational extenders	Value based	Non-Procedural