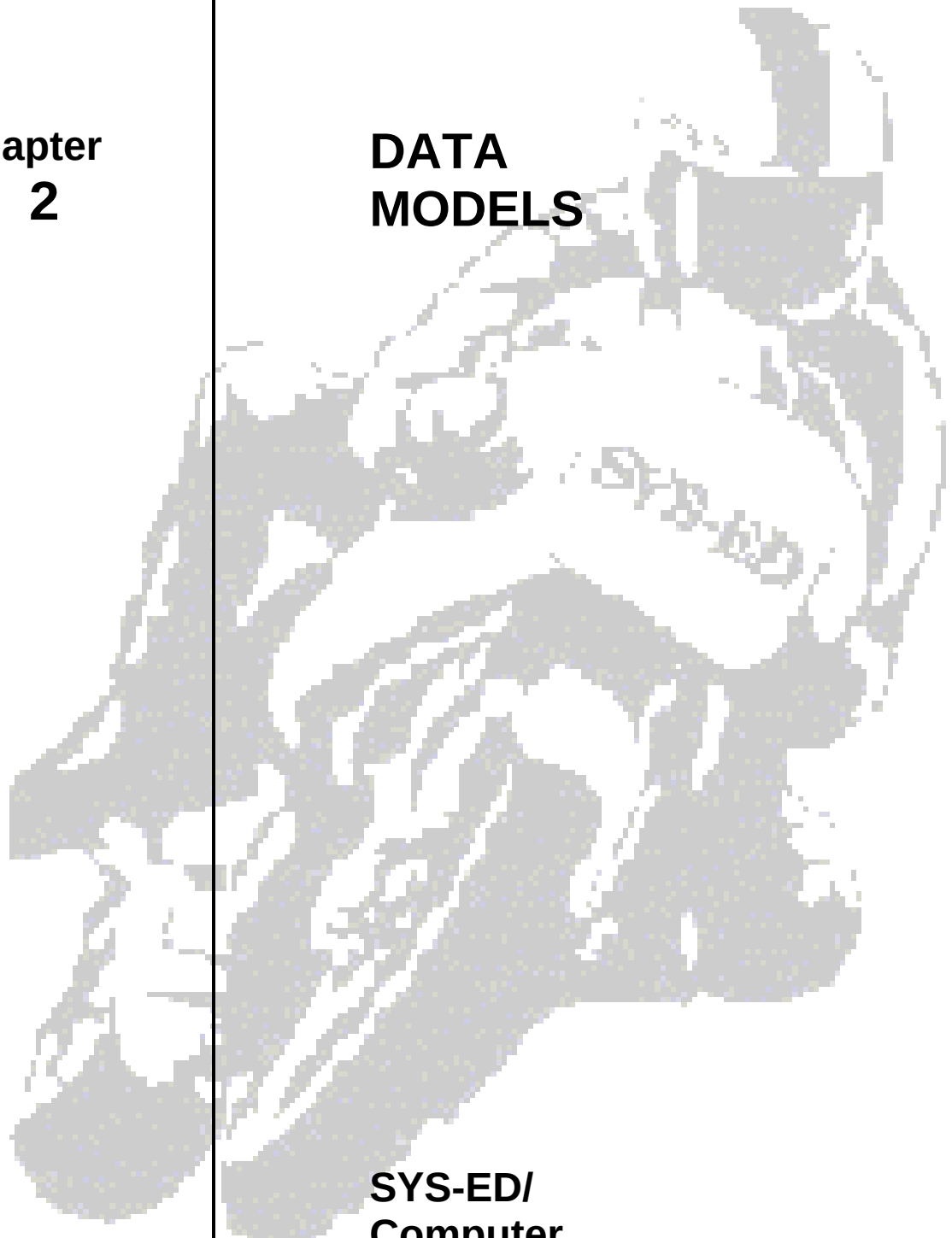
Chapter
2

# DATA MODELS

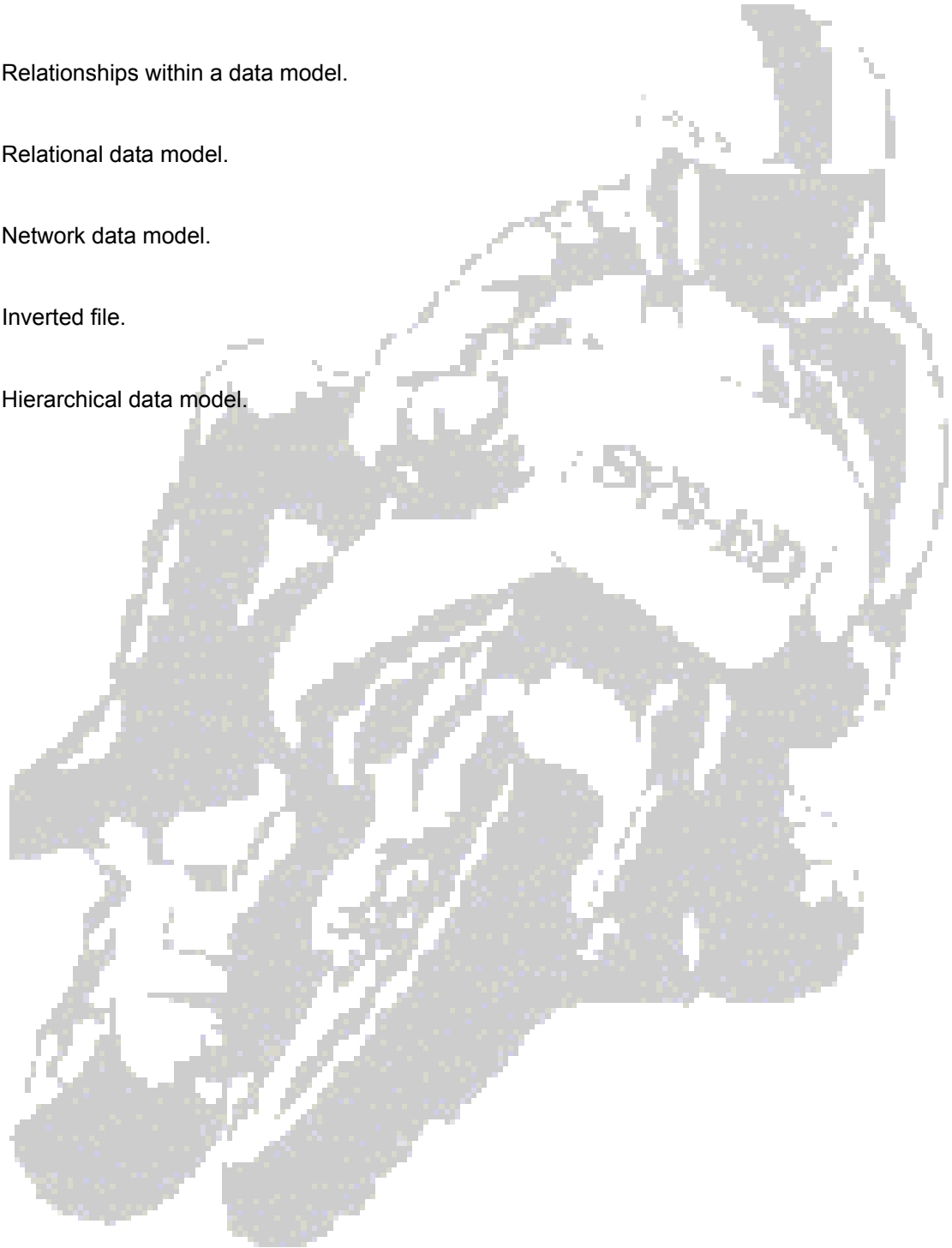


SYS-ED/
Computer
Education
Techniques, Inc.

**Objectives**

You will learn:

- What is a data model?
- Relationships within a data model.
- Relational data model.
- Network data model.
- Inverted file.
- Hierarchical data model.



# 1 What is a Data Model

A data model is a model that describes in an abstract way how data is represented in a business organization, an information system or a database management system.

Th term defines:

- how data generally is organized, such as in a database management system.

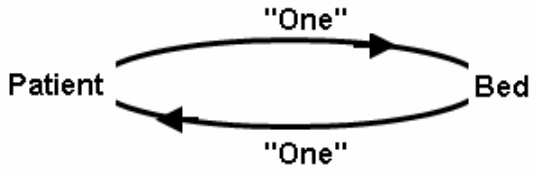
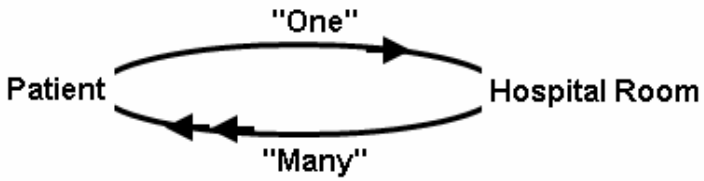
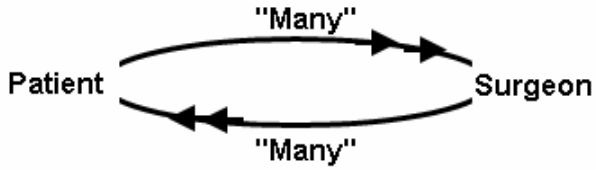
  or

- how data of a specific business function is organized logically, such as the data model of a business.

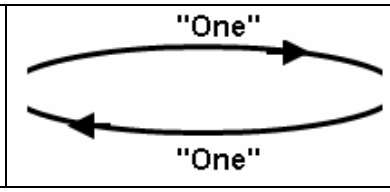
A data model will describe the:

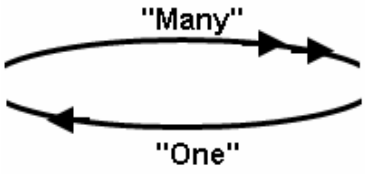
| | |
|---|---|
| Structure | Defines how data are organized: hierarchical, network, relational, object-oriented. |
| Integrity | Provides a language for the definition of rules that restrict which instances of the defined structure are allowed. |
| Manipulation | Provides a language in which updates of the data can be expressed. |
| Querying | Provides a language in which the data can be queried. |

## 2 Relationships Within a Data Model

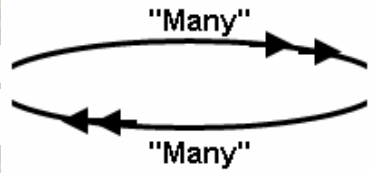
| | |
|---|---|
| 1. |  |
| 2. |  |
| 3. |  |

| **Relationship** | **Explanation** |
|---|---|
| One-to-one | At a given point in time one patient is assigned to one bed. |
| One-to-many | At a given point in time, zero, one, or many patients are assigned to one hospital room; the hospital room may have zero, one, or many patients. However a patient is assigned to only one hospital room. |
| Many to Many | A surgeon may have operated on several patients. A patient may have been operated on by several surgeons on several visits to the hospital. |

| | | |
|---|---|---|
| Patient's Number |  | Another Unique Identifier |
| "One-to-one" relationship between to attributes. | | |

| | | |
|---|---|---|
| Patient's Name |  | Patient's Number |
| "One-to-one" relationship between to attributes. | | |
| John White | | 1111 |
| Mary Jones | | 1234 |
| Charles Brown | | 2345 |
| Hal Kane | | 4876 |
| Paul Kosher | | 5123 |
| Ann Hood | | 6845 |
| John White | | 9843 |

"One-to-Many" Relationship Between Two Attributes. There can be many patients within the same name, but the patient numbers are unique.

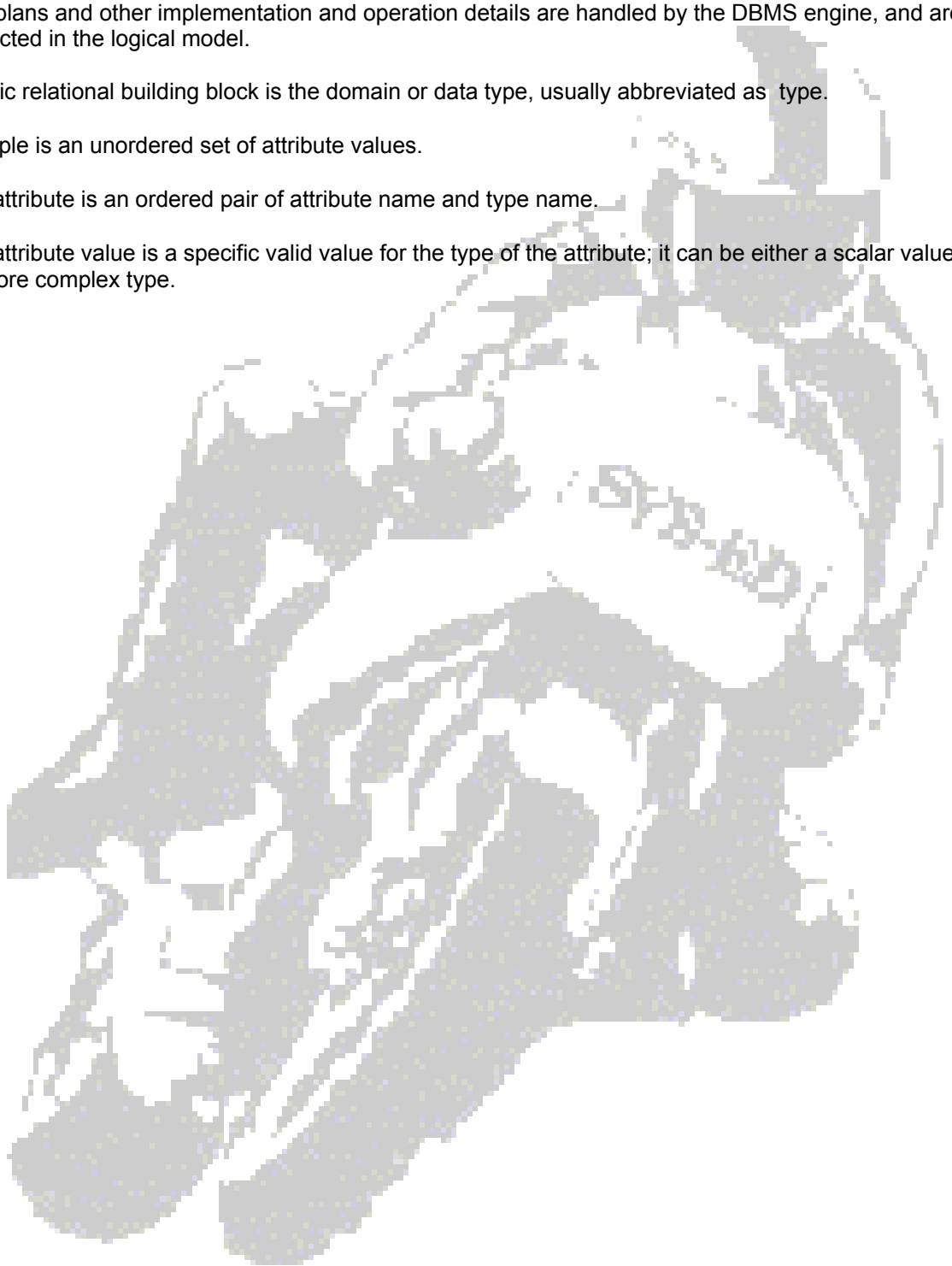
| | | |
|---|---|---|
| Patient's Number |  | Surgeon's Name |
| "Many-to-many" relationship between two attributes. | | |

## 3 The Relational Data Model

The relational model provides for a consistent, logical representation of information. Consistency is achieved by including declared constraints in the database design, which is known as the schema. The access plans and other implementation and operation details are handled by the DBMS engine, and are not reflected in the logical model.

The basic relational building block is the domain or data type, usually abbreviated as type.

- A tuple is an unordered set of attribute values.
- An attribute is an ordered pair of attribute name and type name.
- An attribute value is a specific valid value for the type of the attribute; it can be either a scalar value or a more complex type.



PATIENT TABLE

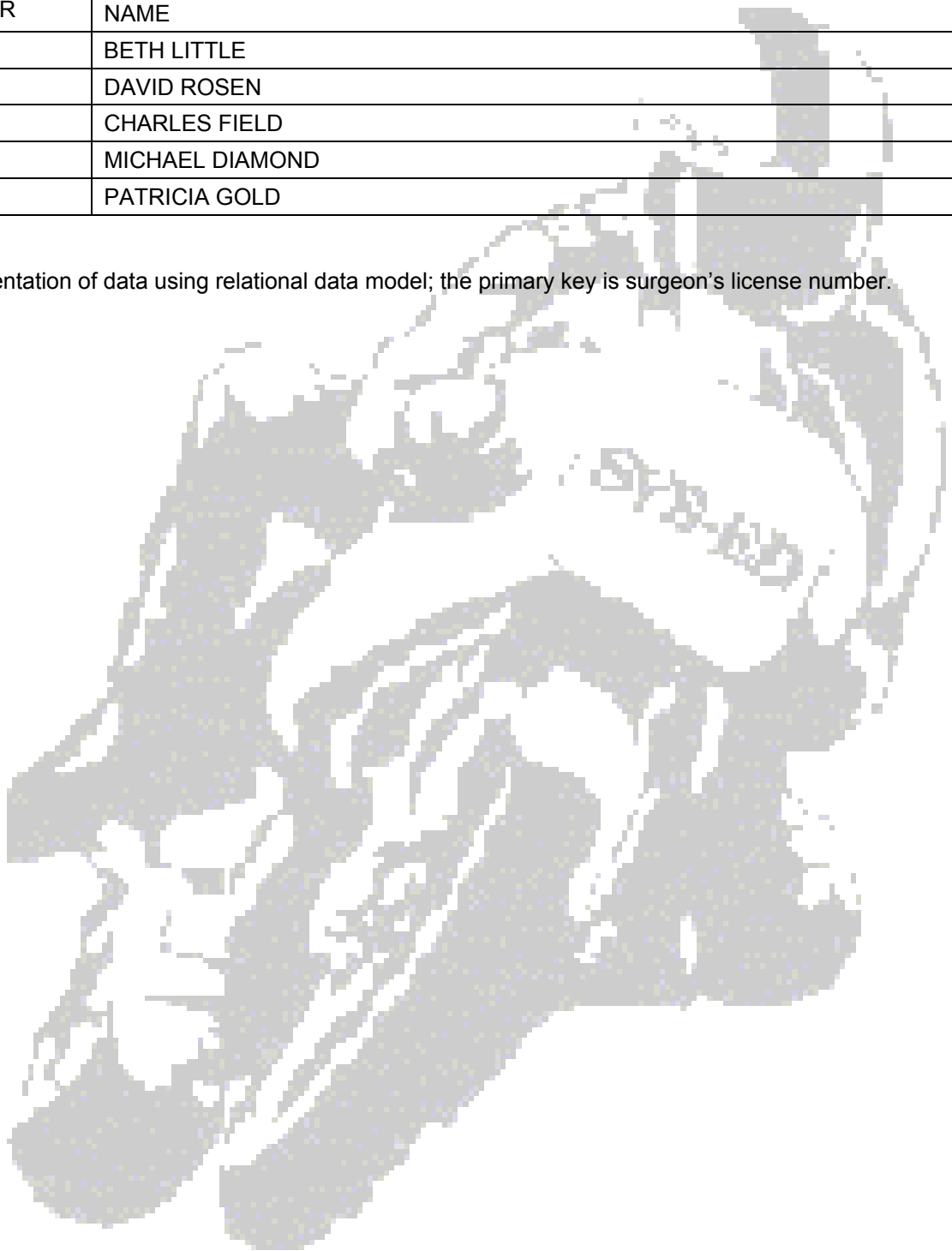
| | | ATTRIBUTE | |
|---|---|---|---|
| KEY | | ↓ | |
| PATIENT'S NUMBER | PATIENT'S NAME | PATIENT'S ADDRESS | |
| 1111 | JOHN WHITE | 15 NEW STREETS NEW YORK, NY | |
| 1234 | MARY JONES | 10 MAIN STREET, RYE, NY | TUPLE |
| 2345 | CHARLES BROWN | DOGWOOD LANE, HARRISON, NY | |
| 4876 | HAL KANE | 55 BOSTON POST ROAD, CHESTER, CT | |
| 5123 | PAUL KOSHER | BLIND BROOK, MAMARONECK, NY | |
| 6845 | ANN HOOD | HILTON ROAD, LARCHMONT, NY | |

This is a representation of data using the relational data model. The primary key is the patient's number; the patient's number is unique. The PATIENT table has three attributes and at this point in time has six tuples. The intention is not to split the entries in the column headed 'Patient's Address' into additional columns. The house number, the street, the name of the city, and the name of the state will be a single attribute.

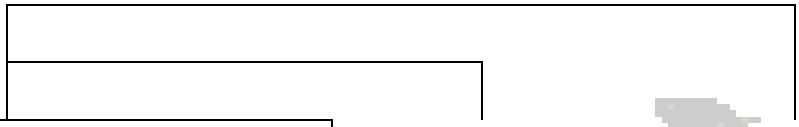
SURGEON TABLE

| KEY | |
|---|---|
| LICENSE NUMBER | SURGEON'S NAME |
| 145 | BETH LITTLE |
| 189 | DAVID ROSEN |
| 243 | CHARLES FIELD |
| 311 | MICHAEL DIAMOND |
| 467 | PATRICIA GOLD |

Representation of data using relational data model; the primary key is surgeon's license number.



PATIENT AND SURGEON TABLE



| COMPOUND KEY | | | | |
|---|---|---|---|---|
| PATIENT'S NUMBER | SURGEON'S LICENSE NUMBER | DATE Of SURGERY | SURGERY | POSTOPERATIVE DRUG ADMINISTERED |
| 1111 | 145 | JAN 1,1977 | GALLSTONES REMOVAL | PENICILLIN |
| 1111 | 311 | JUNE 12,1977 | KIDNEY STONES REMOVAL | ---------- |
| 1234 | 243 | APRL 5, 1976 | EYE CATARACT REMOVAL | TETRACYCLINE |
| 1234 | 467 | MAY 10, 191 | THROMBOSIS REMOVAL | ---------- |
| 2345 | 189 | JAN 8,1978 | OPEN HEART SURGERY | CEPHALOSPORIN |
| 4876 | 145 | NOV 5.1977 | CHOLECY SECTOMY | DEMICILLIN |
| 5123 | 145 | MAY 10,1977 | GALLSTONES REMOVAL | ---------- |
| 6845 | 243 | APRIL 5, 1976 | EYE CORNEA REPLACEMENT | TETRACYCLINE |

This is a representation of data using the relational data model. The primary key is patient's number + surgeon's license number + date of surgery. The major assumption is that a patient may receive only one postoperative drug, if any, for a given specific surgery performed.

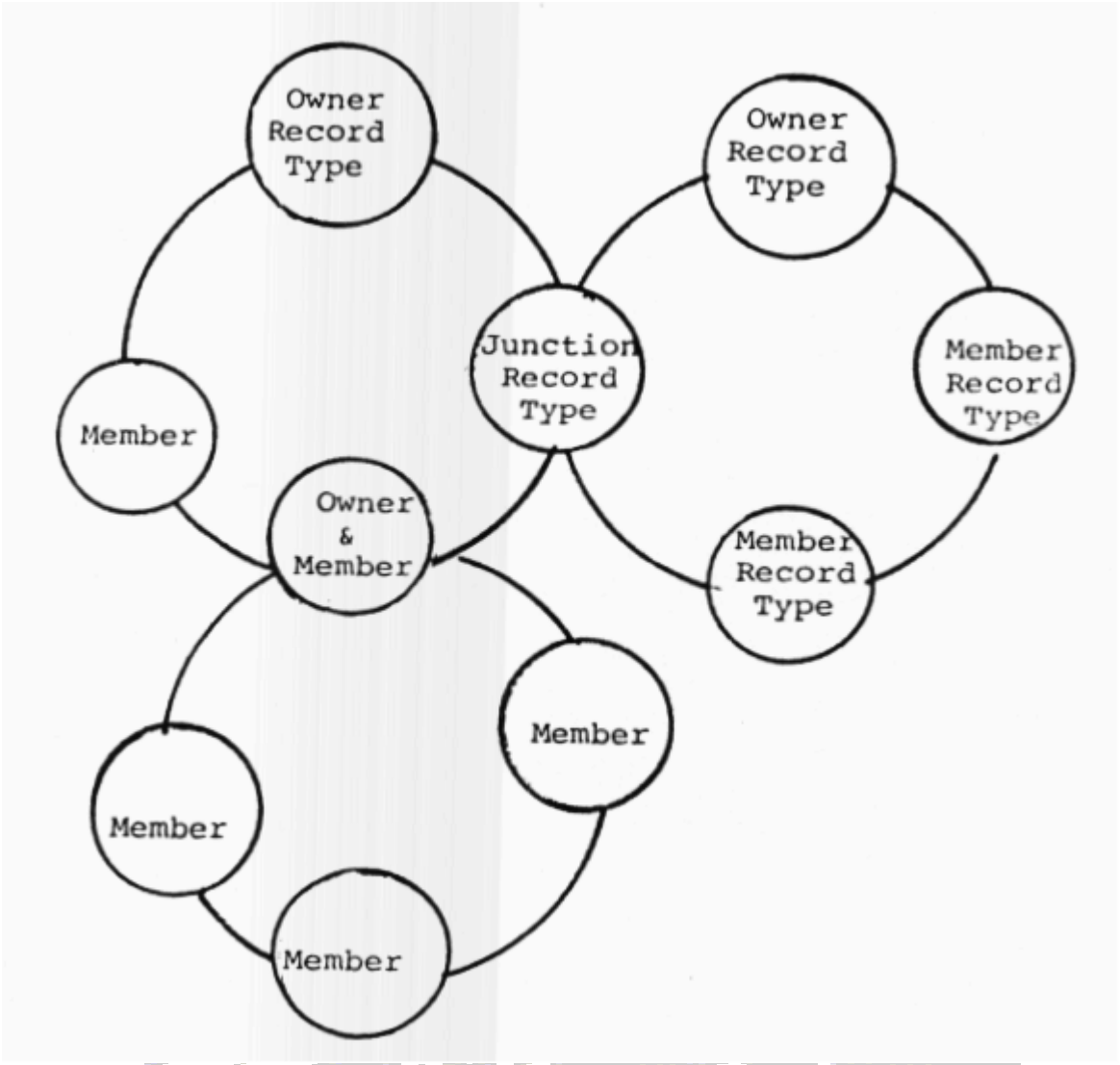
DRUG RELATION

KEY

| POSTOPERATIVE DRUG ADMINISTERED | SIDE EFFECT OF DRUGS |
|---|---|
| PENICILLIN | RASH |
| TETRACYCLINE | FEVER |
| CEPHALOSPORIN | ----- |
| DEMICILLIN | ----- |

The data is being represented using the relational data model. The primary key is postoperative drug administered. The assumption is that there can be only one side effect of a drug, if any. Another assumption is that the side effect of a drug depends only on the drug having been administered.

## 4 The Network (CODASYL) Data Model

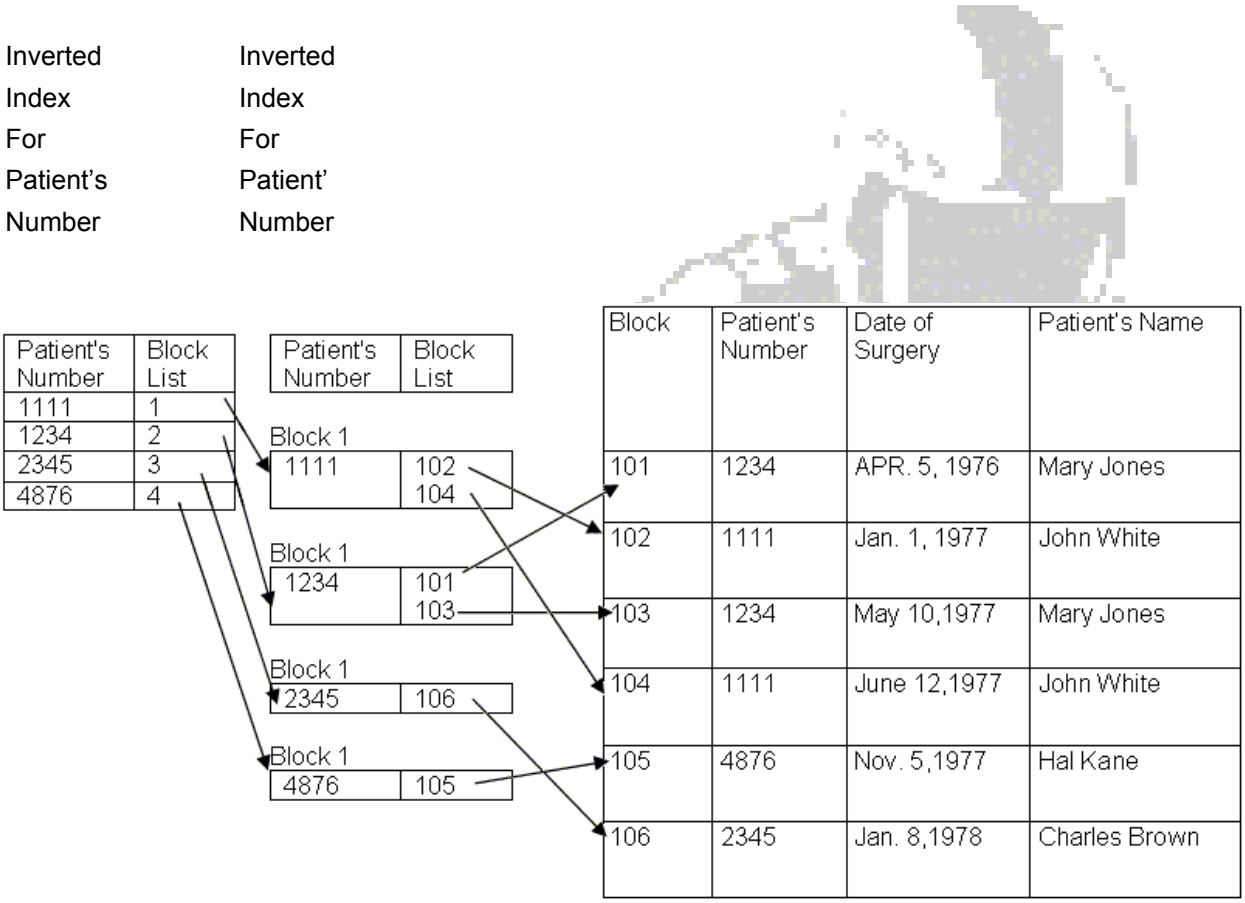


## 5 Inverted File

Inverted Index for Patient's Number

Index To

| Inverted Index For Patient's Number | Inverted Index For Patient' Number |
|---|---|



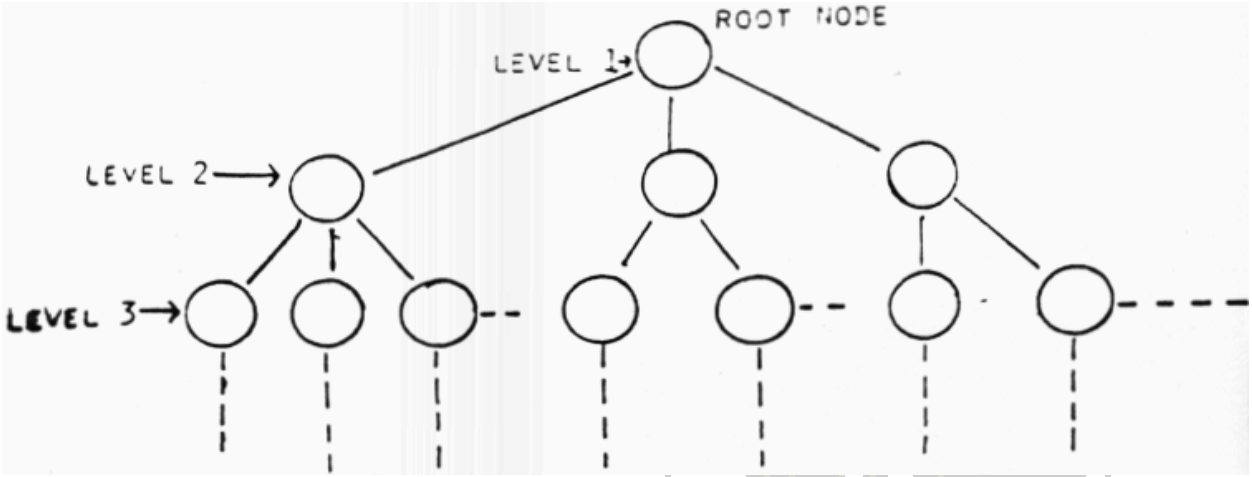
## 6 The Hierarchical Data Model

In a hierarchical data model, data is organized into a tree-like structure based upon a reasonable conservation relationships. The structure allows repeating information using parent/child relationships. All attributes of a specific record are listed under an entity type. In a database, an entity type is the equivalent of a table; each individual record is represented as a row and an attribute as a column. Entity types are related to each other using 1: N mapping, also known as one to many relationships.

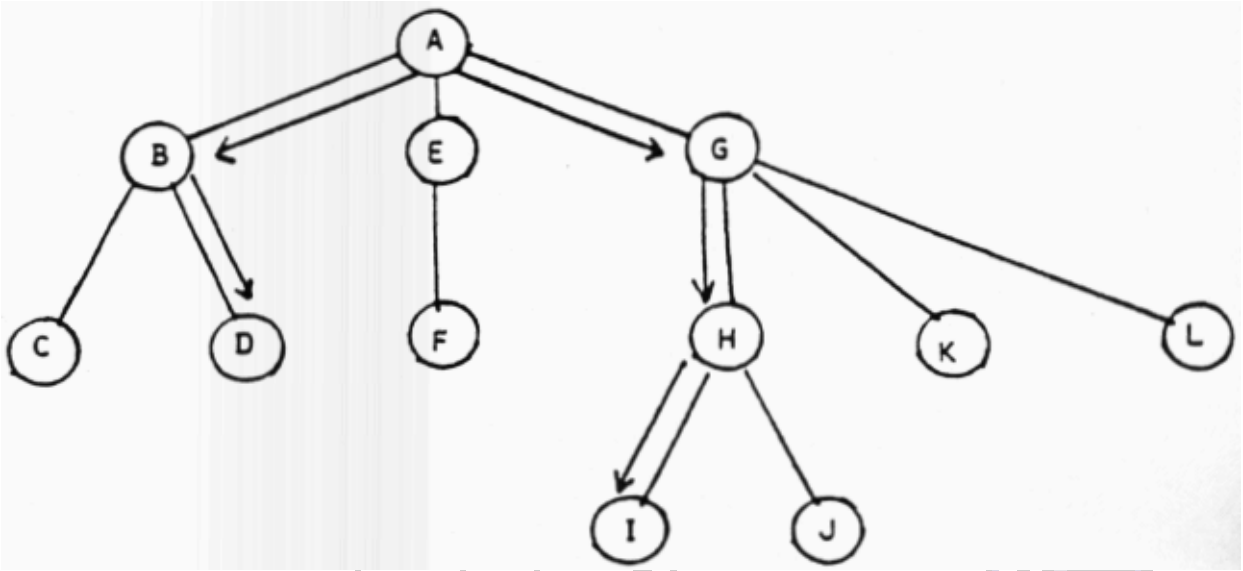
Hierarchical structures were widely used in the first mainframe database management systems. However, owing to their restrictions, they often cannot be used to relate structures that exist in the real world.

While the hierarchical model is rare in modern databases, it is common in many other means of storing information:

- File systems
- Windows registry
- XML documents




A hierarchical tree structure is main up of nodes and branches. A node is a collection of data attributes describing the entity at that node. The highest node of a hierarchical tree structure is called a “root” node; it is the dominant entity type. A root node is at the first level. The dependent nodes, which are the subordinate entity types, are at the second and third levels.

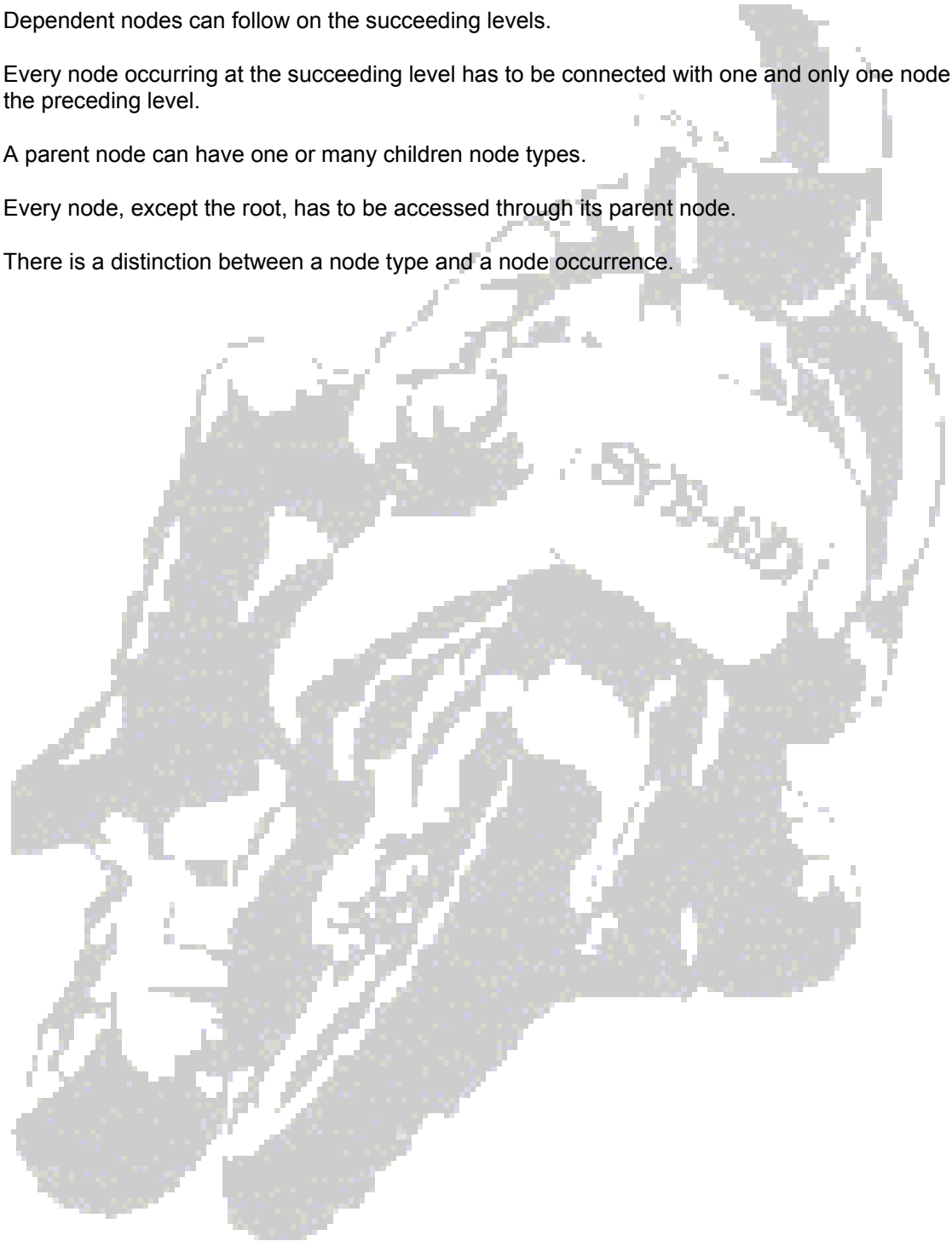


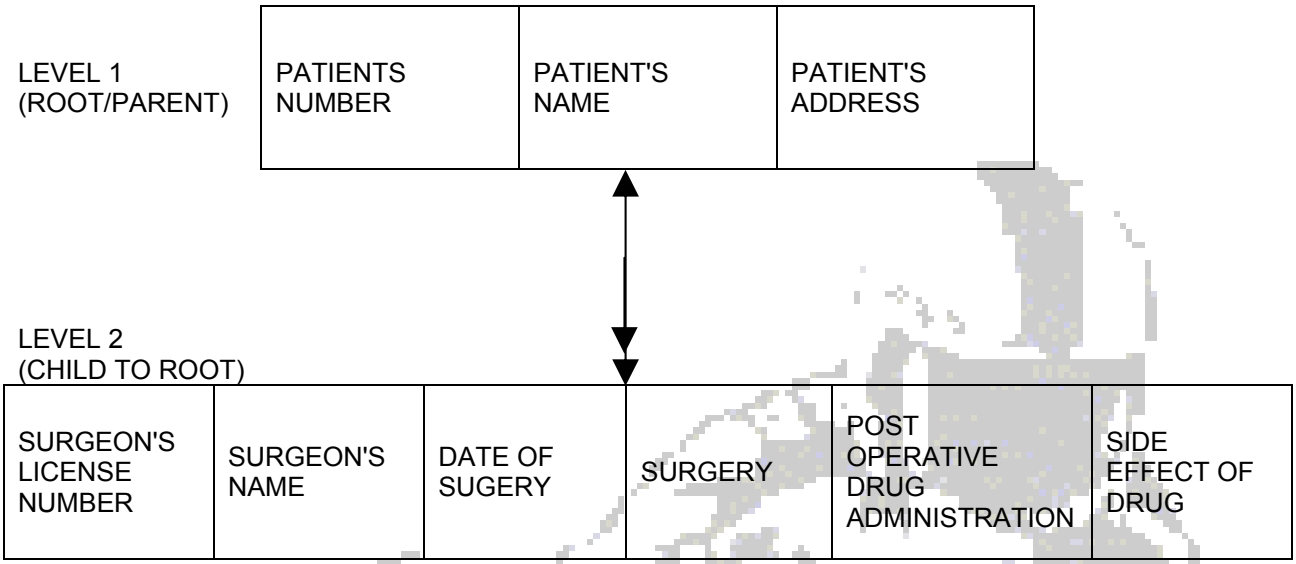

The access path to every node within a hierarchical data model is unique.

For example, node I can be reached only through the path A to G to H to I. Node D can be accessed only through the path A to B to D. Hence a hierarchical data model consists of linear paths.

A hierarchical tree structure has to satisfy the following conditions:

- A hierarchical data model always starts with a root node.
- Every node consists of one or more attributes describing the entity at that node.
- Dependent nodes can follow on the succeeding levels.
- Every node occurring at the succeeding level has to be connected with one and only one node at the preceding level.
- A parent node can have one or many children node types.
- Every node, except the root, has to be accessed through its parent node.
- There is a distinction between a node type and a node occurrence.






This is a representation of data using the hierarchical data model for PATIENT database.

The entity PATIENT is the root node type. The entities SURGEON, SURGERY, and DRUG are combined into one node type as the dependent of the root.

| | | | | | | |
|---|---|---|---|---|---|---|
| | 111 | JOHN WHITE | 15 NEW ST NEW YORK, NY | | | |
| | 311 | MICAHEL DIAMOND | JUNE 12, 1977 | KIDNEY STONES REMOVEL | --------- --------- | --------- --------- |
| 145 | BETH LITTLE | JAN 1, 1977 | GALLSTONES REMOVAL | PENICILLIN | RASH | |

This is a hierarchical database record occurrence for the hierarchical data model in the previous figure. The data base record is for the patient John White. The dependent node type, which is child to root in this case, has two occurrences for the surgeries gallstones removal and kidney stones removal.