Hierarchical and Network Models

A) Hierarchical model

- The db model consists of a collection of records which are connected to one another through links.
- Each record is a collection of fields (attributes) and each field contains only one data value!
- A link is an association between precisely two records.

Example:
In the banking system, consider the db customer-account relationship.

We have two types of records
1. customer
2. account

and their fields:
- customer: name, street, city
- account: number, balance

This organization could be depicted like this:

```
root
/    \
|     |
|     |
BB | Victoria St. | Toronto | CB | Helen St. | Athens | PP | Eiffel St. | Paris

Dummy node
```

So a hierarchical db is a collection of such rooted trees, i.e. a forest of trees.

Problem with hierarchical db:
replication of data either in the same tree or in different trees

Drawbacks:
waste of space by replicated data
potential data inconsistency when updating
I Transformation

Consider the E-R diagram

![E-R diagram]

This diagram could be represented by two trees.

**Tree T1**

- **Customer**
  - Name
  - Street
  - City

- **Account**
  - Number
  - Balance

**Tree T2**

- **Account**
  - Number
  - Balance

- **Customer**
  - Name
  - Street
  - City

Tree T1 corresponds to the previous example.

```
<table>
<thead>
<tr>
<th>Customer</th>
<th>Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>111</td>
</tr>
<tr>
<td>Victoria St.</td>
<td>$200</td>
</tr>
<tr>
<td>Toronto</td>
<td></td>
</tr>
</tbody>
</table>

| CB       | 555     |
| Helen St. | $110 |
| Athens    |         |

| PP       | 222     |
| Eiffel St. | $300 |
| Paris     |         |
```

Tree T2 corresponds to:

```
<table>
<thead>
<tr>
<th>Customer</th>
<th>Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>111</td>
</tr>
<tr>
<td>Victoria St.</td>
<td>$200</td>
</tr>
<tr>
<td>Toronto</td>
<td></td>
</tr>
</tbody>
</table>

| CB       | 555     |
| Helen St. | $110 |
| Athens    |         |

| PP       | 222     |
| Eiffel St. | $300 |
| Paris     |         |
II Several Relationships

Consider this E-R diagram:

Three relations:
1. customer
2. account
3. branch

Note: This specifies that a customer may have several accounts each located in a specific branch and that an account may belong to several different customers.

Tree T1

Tree T2

Tree T2 would be illustrated in a similar manner.
III Several Binary Relationships

Consider the E-R diagram

![Diagram of several binary relationships]

Note:
We have to ensure, that for each single relationship, the transformation will result in diagrams which are of rooted tree form.
We work as before then the solution yields the results as being dictated by the following:

1)

![Diagram 1]

2)

![Diagram 2]

As we have seen in the simple case of single relationships.

Notes on the examples
- record templates (Structured record types)
- currency pointers are a set of pointers, one of each db tree containing the record address in that particular tree, regardless of tree type, most recently accessed by the application program.
- status flag is a variable set by the system to indicate to the application program the outcome of the last db operation. (dbstatus or if status = 0 successful!!)
B) Network Model

Definition:
A network db consists of a collection of records which are connected to one another through links.

- A record is a collection of fields (attributes) each containing a data item (value).
- A link is an binary association between two records - binary (see E-R)

Example
Consider the customer-account association

```
customer: name  street   city    account: number  balance
          1:1 relationship
```

A real situation:

```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>Victoria St.</td>
<td>Toronto</td>
</tr>
<tr>
<td>CB</td>
<td>Peanut St.</td>
<td>Athens</td>
</tr>
<tr>
<td>PP</td>
<td>Eiffel St.</td>
<td>Paris</td>
</tr>
<tr>
<td>HH</td>
<td>Main St.</td>
<td>New York</td>
</tr>
</tbody>
</table>
```

This corresponds to E-R

```
name  street   city    date    number  balance
Customer      CustAcct    Account
```

Customer  
```
nname street city
```

Account  
```
nnumber balance
```
As with hierarchical:

If we add attribute date to the customer-account relationship to denote the last time the account was accessed by the customer. We have:

Transform:
- Entities: customer, account
- Create a new record type date (single field to represent date)
- Create links:
  1) CustDate from date to Customer
  2) AcctDate from date to Account
General Relationships

As with hierarchical, consider the account, customer and branch related through the account-customer-branch relationship.

Steps for transformation

- Entity set replaced by record types account, customer and branch.
- Create a new record type link that may either have no fields or have a single field containing a unique identifier. This identifier is supplied by the system and it is not used directly by the application program.
- Referred to: dummy or link or junction record type.
- Create the links CustLink, AcctLink, BrLink from link to customer, from link to account and from link to branch respectively.
The example database is as follows:

Note the technique can be used for more than 3 entity sets. We create a many-to-one link from Link record to the record types corresponding to each entity set involved in the relationship.