What Goes Around Comes Around

IMS Era

IMS was released around 1968, and initially had a hierarchical data model. It understood the notion of a **record type,** which is a collection of named fields with their associated data types. Each **instance** of a record type is forced to obey the data description indicated in the definition of the record type.

IMS在1968年左右发布，而最初有一个层次数据模型。据了解记录类型的概念，这是与它们相关联的命名字段的集合数据类型。一个记录类型的每个实例都被迫服从数据描述在记录类型的定义表示。

**This requirement of tree-structured data presents a challenge for our sample data：**

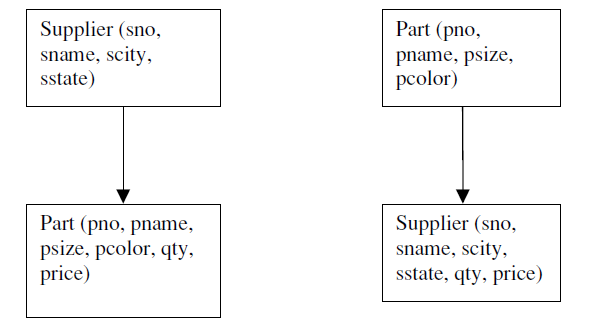
**1. Information is repeated.** In the first schema, Part information is repeated for

each Supplier who supplies the part. In the second schema, Supplier information

is repeated for each part he supplies. 信息被重复。在第一模式中，第一部分信息重复

每个供应商提供谁的一部分。在第二个模式，供应商信息重复他每个部分供应。重复的信息是不可取的，因为它提供了可能性不一致的数据

**2.Existence depends on parents**. In the first schema it is impossible for there to be a part that is not currently supplied by anybody. In the second schema, it is impossible to have a supplier which does not currently supply anything.



IMS supported four different storage formats for hierarchical data. Basically root records can either be:

1. Stored sequentially

2. Indexed in a B-tree using the key of the record

3. Hashed using the key of the record

4. Dependent records are found from the root using either Physical sequentially Various forms of pointers.

Some of the storage organizations impose restrictions on DL/1 commands. the storage organization that hashes root records on a key cannot support “get next”

1.HSAM： 层次顺序访问方法，片段按层次顺序作物理邻接存储。

2.HISAM：层次索引顺序访问方法，非根片段按层次顺序值升序邻接存储，根片段用顺序域索引的方法组织并指向下属区域。

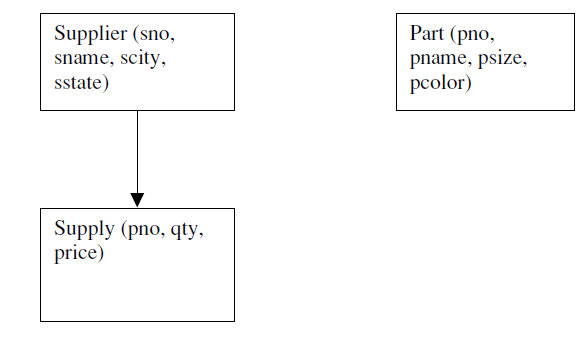
3.HDAM： 层次直接访问方法，片段的存储采用离散分布方式，根片段用顺序域HASH方法组织，从根片段出发用指针按层次顺序值的顺序把物理上分散的从属片段链接起来。

4.HIDAM：层次索引直接访问方法，类似于HDAM，不同的是，根片段采用顺序域索引技术组织，而不是HASH方法。

**physical data independence:** The ability of a data base application to continue to run, regardless of what tuning is performed at the physical level. Physical data independence is important because a DBMS application is not typically written all atonce. As new programs are added to an application, the tuning demands may change, and better DBMS performance could be achieved by changing the storage organization.

IMS supports a certain level of **logical data independence**,because DL/1 is actually defined on a **logical data base**, not on the actual physical data base that is stored. Hence, a DL/1 program can be written initially by defining the logical

IMS支持逻辑数据独立性的一定水平，因为DL /1实际上是在逻辑数据的基础上定义的，而不是在存储的实际物理数据的基础上。因此，在DL/1程序最初可以通过定义逻辑写入



假设一个人构建只包含部分信息的两个物理数据基地，一个和含第二供应商和供应信息如图所示的图中,DL/1方案在树上定义;因此它们不能用直接在图3相反的结构，IMS允许的逻辑的定义如图4这里所示的数据的基础上，从两个不同的供应和零件记录类型数据库被“融合”（加盟）的零件号的共同价值为分层所示的结构。

We will summarize the lessons learned so far, and then turn to the CODASYL proposal.

Lesson 1: Physical and logical data independence are highly desirable

Lesson 2: Tree structured data models are very restrictive

Lesson 3: It is a challenge to provide sophisticated logical reorganizations of tree

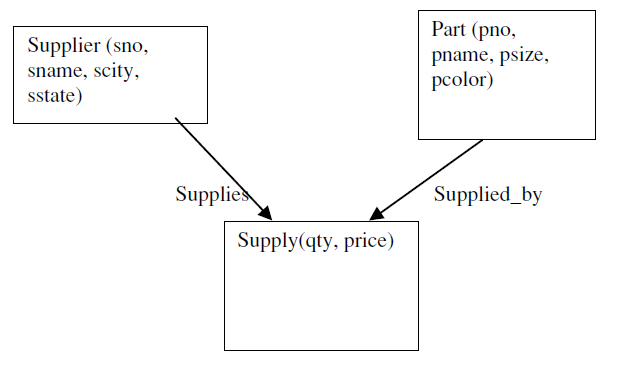
structured data

Lesson 4: A record-at-a-time user interface forces the programmer to do manual query

optimization, and this is often hard.

**CODASYL Era -**Committee on Data Systems Languages

CODASYL was an ad-hoc committee that championed a network data model along with a record-at-a-time data manipulation language.This model organized a collection of record types, each with keys, into a network, rather than a tree. Hence, a given record instance could have multiple parents, rather than a single one,



Supplier-Parts-Supply example

**To find the red parts supplied by Supplier 16 in CODASYL, one can use the following code:**

查询方法：One enters the data base at supplier 16, and then iterates over the members of the Supplies set. This will yield a collection of Supply records. For each one, the owner in the Supplied\_by set is identified, and a check for redness performed

查询语句：

Find Supplier (SNO = 16)

Until no-more {

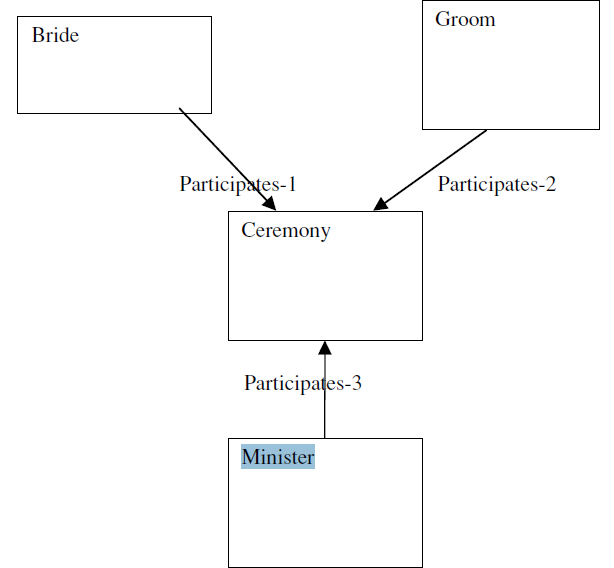
Find next Supply record in Supplies

Find owner Part record in Supplied\_by

Get current record

-check for red—

}



3-way relationship between a bride, a groom, and a minister

**IMS move to network model’s advantage:**

1.no kludges are required to implement graph-structured data 不需要组装件实现图结构的数据

**IMS move to network model’s features:**

2.The CODASYL proposal suggested that the records in each entry point be hashed on the key in the record CODASYL模式建议在每个入口点的记录被散列的记录关键字。

2.The CODASYL proposal provided essentially no physical data independence CODASYL模式基本上没有物理数据独立性，而且也没有逻辑独立性

注：

In contrast, a CODASYL programmer must keep track of the:

The last record touched by the application

The last record of each record type touched

The last record of each set type touched

**Hence, the lessons learned in CODASYL were:**

Lesson 5: Networks are more flexible than hierarchies but more complex

Lesson 6: Loading and recovering networks is more complex than hierarchies

**Relational Era**

proposal was threefold:

1．Store the data in a simple data structure (tables)——one has a better change of providing logical data independence

2. Access it through a high level set-at-a-time DML——one can provide a high degree of physical data independence

3. No need for a physical storage proposal——because of DML