

# The Nested Relational Model

April 12, 2002

Michael Minock

## Nested Relational Model

Removes the restriction of 1st normal form on relations.

Also known as **NFNF** - non-first normal form databases.

```
DEPT= (DNO, DNAME, MANAGER, EMPLOYEES, PROJECTS, LOCATIONS)
EMPLOYEES = (ENAME, DEPENDENTS)
PROJECTS = (PNAME, PLOC)
LOCATIONS = (DLOC)
DEPENDENTS = (DNAME, AGE)
```

All attribute names are distinct.

A nested attribute may be a **multi-valued composite attribute** (EMPLOYEES), or a **multi-valued simple attribute** (LOCATIONS).

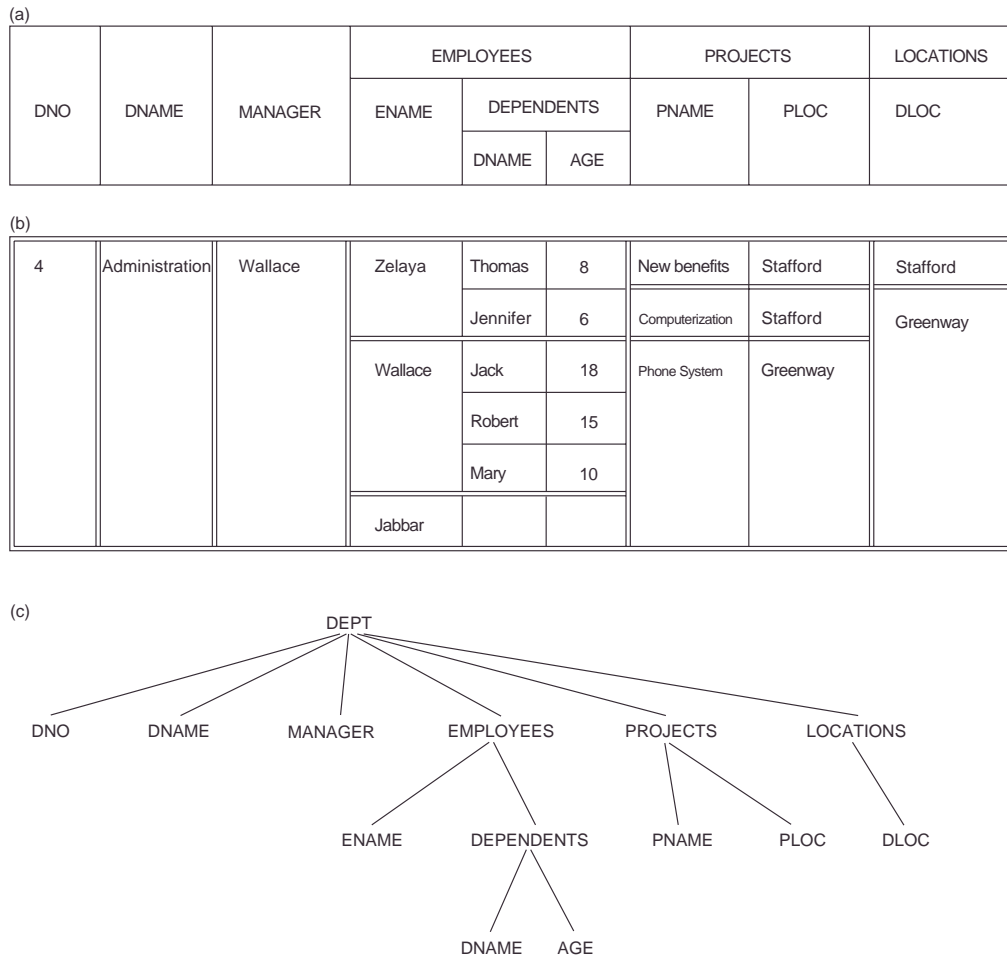
Some models accept treat specifically **single valued composite attributes**, but we will not.

What about **single valued simple attributes**?

So this gives us collection types and row types!

# Nested Relational Model

**Figure 13.1** Illustrating a nested relation. (a) DEPT schema. (b) Example of a Non-1NF tuple of DEPT. (c) Tree representation of DEPT schema.



## Nested Relational Model

Note that full relations are being *nested*.

- *external relational schemas* (root nodes)
  - DEPT
- *internal relational schemas* (internal nodes)
  - EMPLOYEES, PROJECTS, LOCATIONS, and DEPENDENTS
- simple attributes leaf level
  - DNO, DNAME, MANAGER, ENAME, DNAME, AGE, PNAME, PLOC, and DLOC.

## M:N Relationships?

The nested relations represent *independent information*.

How do you represent the M:N relationship between employees and projects?

Difficult to capture in a hierarchical structure.

But nested model is strictly more expressive than relational – so just do it the standard way at the proper level of nesting.

## Nested Relational Model

**Figure 14.4** Example relations for the schemas in Figure 14.3 that result from applying NATURAL JOIN to the relations in Figure 14.2. These may be stored as base relations for performance reasons.

**EMP\_DEPT**

| ENAME               | <u>SSN</u> | BDATE      | ADDRESS                | DNUMBER | DNAME          | DMGRSSN   |
|---------------------|------------|------------|------------------------|---------|----------------|-----------|
| Smith,John B.       | 123456789  | 1965-01-09 | 731 Fondren,Houston,TX | 5       | Research       | 333445555 |
| Wong,Franklin T.    | 333445555  | 1955-12-08 | 638 Voss,Houston,TX    | 5       | Research       | 333445555 |
| Zelaya, Alicia J.   | 999887777  | 1968-07-19 | 3321 Castle, Spring,TX | 4       | Administration | 987654321 |
| Wallace,Jennifer S. | 987654321  | 1941-06-20 | 291 Berry,Bellaire,TX  | 4       | Administration | 987654321 |
| Narayan,Ramesh K.   | 666884444  | 1962-09-15 | 975 FireOak,Humble,TX  | 5       | Research       | 333445555 |
| English,Joyce A.    | 453453453  | 1972-07-31 | 5631 Rice,Houston,TX   | 5       | Research       | 333445555 |
| Jabbar,Ahmad V.     | 987987987  | 1969-03-29 | 980 Dallas,Houston,TX  | 4       | Administration | 987654321 |
| Borg,James E.       | 888665555  | 1937-11-10 | 450 Stone,Houston,TX   | 1       | Headquarters   | 888665555 |

**EMP\_PROJ**

| <u>SSN</u> | <u>PNUMBER</u> | HOURS | ENAME               | PNAME           | PLOCATION |
|------------|----------------|-------|---------------------|-----------------|-----------|
| 123456789  | 1              | 32.5  | Smith,John B.       | ProductX        | Bellaire  |
| 123456789  | 2              | 7.5   | Smith,John B.       | ProductY        | Sugarland |
| 666884444  | 3              | 40.0  | Narayan,Ramesh K.   | ProductZ        | Houston   |
| 453453453  | 1              | 20.0  | English,Joyce A.    | ProductX        | Bellaire  |
| 453453453  | 2              | 20.0  | English,Joyce A.    | ProductY        | Sugarland |
| 333445555  | 2              | 10.0  | Wong,Franklin T.    | ProductY        | Sugarland |
| 333445555  | 3              | 10.0  | Wong,Franklin T.    | ProductZ        | Houston   |
| 333445555  | 10             | 10.0  | Wong,Franklin T.    | Computerization | Stafford  |
| 333445555  | 20             | 10.0  | Wong,Franklin T.    | Reorganization  | Houston   |
| 999887777  | 30             | 30.0  | Zelaya,Alicia J.    | Newbenefits     | Stafford  |
| 999887777  | 10             | 10.0  | Zelaya,Alicia J.    | Computerization | Stafford  |
| 987987987  | 10             | 35.0  | JabbarAhmad V       | Computerization | Stafford  |
| 987987987  | 30             | 5.0   | JabbarAhmad V       | Newbenefits     | Stafford  |
| 987654321  | 30             | 20.0  | Wallace,Jennifer S. | Newbenefits     | Stafford  |
| 987654321  | 20             | 15.0  | Wallace,Jennifer S. | Reorganization  | Houston   |
| 888665555  | 20             | null  | Borg,James E.       | Reorganization  | Houston   |

## Unary operators NEST (and UNNEST)

$EMP\_PROJ\_FLAT \leftarrow \pi_{SSN,ENAME,PNUMBER,HOURS}(EMP\_PROJ)$

NEST groups together the tuples *with the same value* for attributes *not specified* in the NEST operation.

$EMP\_PROJ\_NESTED \leftarrow$

$NEST_{PROJS=(PNUMBER,HOURS)}(EMP\_PROJ\_FLAT)$

Note similarity to group by in SQL.

There is the inverse operation as well:

$EMP\_PROJ\_NESTED \equiv$

$UNNEST_{PROJS=(PNUMBER,HOURS)}(EMP\_PROJ\_NESTED)$

**Figure 14.9** Normalizing nested relations into 1NF. (a) Schema of the EMP\_PROJ relation with a “nested relation” PROJS. (b) Example extension of the EMP\_PROJ relation showing nested relations within each tuple. (c) Decomposing EMP\_PROJ into 1NF relations EMP\_PROJ1 and EMP\_PROJ2 by propagating the primary key.

(a)

**EMP\_PROJ**

| SSN | ENAME | PROJS   |       |
|-----|-------|---------|-------|
|     |       | PNUMBER | HOURS |

(b)

**EMP\_PROJ**

| SSN       | ENAME               | PNUMBER | HOURS |
|-----------|---------------------|---------|-------|
| 123456789 | Smith,John B.       | 1       | 32.5  |
|           |                     | 2       | 7.5   |
| 666884444 | Narayan,Ramesh K.   | 3       | 40.0  |
| 453453453 | English,Joyce A.    | 1       | 20.0  |
|           |                     | 2       | 20.0  |
| 333445555 | Wong,Franklin T.    | 2       | 10.0  |
|           |                     | 3       | 10.0  |
|           |                     | 10      | 10.0  |
|           |                     | 20      | 10.0  |
| 999887777 | Zelaya,Alicia J.    | 30      | 30.0  |
|           |                     | 10      | 10.0  |
| 987987987 | Jabbar,Ahmad V.     | 10      | 35.0  |
|           |                     | 30      | 5.0   |
| 987654321 | Wallace,Jennifer S. | 30      | 20.0  |
|           |                     | 20      | 15.0  |
| 888665555 | Borg,James E.       | 20      | null  |

(c)

**EMP\_PROJ1**

| <u>SSN</u> | ENAME |
|------------|-------|
|------------|-------|

**EMP\_PROJ2**

| <u>SSN</u> | <u>PNUMBER</u> | HOURS |
|------------|----------------|-------|
|------------|----------------|-------|



## Nested Relational Model

SQL1999 has proposed operations to carry out such nesting and un-nesting in the DML.

So what do we do in the DDL?

Nested tuples resemble complex objects, but with a strictly hierarchical structure.

This data model is close to how XML is defined. Except that XML has more syntactic sugar and has IDs that can point to arbitrary nodes back in the “tree”.

And XML has a weak type system.