Network Model

The Network Model is based on a directed graph.

This model was developed by the Data Base Task Group of the CODASYL COBOL committee. In 1971 they proposed a schema DDL, a subschema DDL and a DML (to be embedded in COBOL programs).

Consider a database to hold the players for baseball teams:

Each Team has multiple players. This is shown by a linked list or directed graph structure. The structure is called a **DBTG Set**. Each set has an owner type of record (in this case team) and a member type of record (in this case player). Each owner can have many members. A DBTG set is strictly 1-many.

Now let’s add Player Postitions – Assume that each player plays multiple position. So we want another DBTG set with Player as the owner and Position as the member.
But the connection between Player and Position is really many-many. So we start with Peter and follow the green link to Pitcher. Which green edge do we follow OUT of Pitcher? Back to Peter? or on to Catcher?

DBTG set's can't handle many-many. So we have to “fix” the problem:

Create 3 DBTG set's:

1. Owner: Team  Member: Players
2. **Owner: Player**  Member: Temp
3. Owner: Position  Member: Temp
<table>
<thead>
<tr>
<th>DBTG DDL</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SCHEMA NAME IS name</strong></td>
<td>Gives name of database</td>
</tr>
<tr>
<td><strong>RECORD NAME IS name</strong></td>
<td>Gives name of record (node on graph)</td>
</tr>
<tr>
<td><strong>KEY name IS ascending/descending</strong></td>
<td>Field in record to be used for sorting. Optional</td>
</tr>
<tr>
<td><strong>DUPLICATES ARE NOT ALLOWED FOR name</strong></td>
<td>This field cannot contain duplicate values. Optional</td>
</tr>
<tr>
<td><strong>02 name TYPE is type</strong></td>
<td>Gives name and type of field. Types are given using COBOL syntax.</td>
</tr>
<tr>
<td><strong>SET NAME IS name</strong></td>
<td>Gives name of the set</td>
</tr>
<tr>
<td><strong>OWNER IS name</strong></td>
<td>Record that is owner of set</td>
</tr>
<tr>
<td><strong>MEMBER IS name</strong></td>
<td>Record that is member of set</td>
</tr>
<tr>
<td><strong>ORDER IS next/first/last/prior/system default/sorted</strong></td>
<td>Describes how members are ordered within the set</td>
</tr>
<tr>
<td><strong>INSERTION is automatic/manual</strong></td>
<td>Automatic: DBMS inserts member into set when it is created. Manual: programmer must issue statement to insert into set</td>
</tr>
<tr>
<td><strong>RETENTION is fixed/mandatory/optional</strong></td>
<td>Fixed: members cannot be moved from one set occurrence to another. Mandatory: member must belong to some set but can move from one occurrence to another. Optional: member does not have to be part of a set</td>
</tr>
<tr>
<td><strong>SET SELECTION IS KEY name</strong></td>
<td>Field of owner used to choose set occurrence.</td>
</tr>
</tbody>
</table>
**Schema Name is College-Life**

**Record Name is Student**
- **Key**: Student-ID is Ascending
- **Duplicates are not allowed for**: Student-ID
  - 02 Student-ID: Type is Decimal 9
  - 02 Student-Name: Type is Character 30
  - 02 Student-Address: Type is Character 30
  - 02 Major: Type is Character 08
  - 02 Birth-Date: Type is Character 08

**Record Name is Car**
- **Duplicates are not allowed for**: Car-Serial-No
- **Duplicates are not allowed for**: Car-License
  - 02 Car-Serial-No: Type is Character 18
  - 02 Car-Make: Type is Character 12
  - 02 Car-Model: Type is Character 17
  - 02 Car-Year: Type is Decimal 2
  - 02 Car-License: Type is Character 6

**Record Name is Tickets**
  - 02 Ticket-Date: Type is Character 8
  - 02 Offense: Type is Decimal 6
  - 02 Ticket-Number: Type is Character 12
  - 02 Ticket-Status: Type is

**Record Name is Insurance**
  - 02 Company-Name: Type is Character 43
  - 02 Company-Address: Type is Character 35

**Record Name is Course**
  - 02 Course Description: Type is Character 35
  - 02 Course-ID: Type is Character 5

**Record Name is Class**
  - 02 Course-ID: Type is Character 5
  - 02 Class-Section: Type is Character 8
  - 02 Class-Time: Type is Character 13
SET NAME IS STUDENT-CAR
    OWNER IS STUDENT
    ORDER IS SORTED
    MEMBER IS CAR
    INSERTION IS MANUAL
    RETENTION IS OPTIONAL
    SET SELECTION IS KEY STUDENT-ID

SET NAME IS CAR-TICKET
    OWNER IS CAR
    ORDER IS NEXT
    MEMBER IS TICKETS
    INSERTION IS AUTOMATIC
    RETENTION IS FIXED
    SET SELECTION IS KEY CAR-LICENSE

SET NAME IS INSURANCE-CAR
    OWNER IS INSURANCE
    ORDER IS SYSTEM DEFAULT
    MEMBER IS CAR
    INSERTION IS MANUAL
    RETENTION IS OPTIONAL

SET NAME IS STUDENT-CLASS
    OWNER IS STUDENT
    ORDER IS FIRST
    MEMBER IS CLASS
    INSERTION IS MANUAL
    RETENTION IS OPTIONAL
    SET SELECTION IS KEY STUDENT-ID

SET NAME IS COURSE-CLASS
    OWNER IS COURSE
    ORDER IS LAST
    MEMBER IS CLASS
    INSERTION IS MANUAL
    RETENTION IS OPTIONAL
**Subschema DDL**

<table>
<thead>
<tr>
<th>SS name WITHIN schema name</th>
<th>Names subschema and associates it with schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPPING DIVISION</td>
<td>Rename set or record</td>
</tr>
<tr>
<td>STRUCTURE DIVISION</td>
<td>Gives records and sets to include</td>
</tr>
</tbody>
</table>

```
TITLE DIVISION.
SS TICKET-PROCESSING WITHIN COLLEGE-LIFE.
MAPPING DIVISION.
ALIAS SECTION.
AD RECORD STUDENT IS CAR-OWNER.
AD SET CAR-TICKET IS VIOLATION.
AD STUDENT-ID IS OWNER-ID.
AD STUDENT-NAME IS OWNERS-NAME.
AD STUDENT-ADDRESS IS OWNERS-ADDRESS.
STRUCTURE DIVISION.
RECORD SECTION.
01 CAR-OWNER.
   05 OWNERS-NAME PIC X(30).
   05 OWNERS-ADDRESS PIC X(60).
   05 OWNER-ID PIC 9(9).
01 CAR.
01 TICKETS.
SET SECTION.
SD VIOLATION.
SD STUDENT-CAR.
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>READY name USAGE MODE IS exclusive/protected</td>
<td>Gives name of subschema to use. Exclusive: me only Protected: allow other readers</td>
</tr>
<tr>
<td>FIND record WITHIN set USING x</td>
<td>Go to the first record within the given set type that has the given key value</td>
</tr>
<tr>
<td>FIND next/prior/first/last/owner WITHIN set</td>
<td>Moves through members and back to owner. Must state what set you are moving through as record can be in more than one set type.</td>
</tr>
<tr>
<td>GET</td>
<td>Copy current record into application program</td>
</tr>
<tr>
<td>MODIFY</td>
<td>Replace current record with data in application program</td>
</tr>
<tr>
<td>STORE</td>
<td>Insert new record into database. Does not connect to a set if insertion is manual</td>
</tr>
<tr>
<td>CONNECT record TO set</td>
<td>For manual insertion</td>
</tr>
<tr>
<td>DISCONNECT record FROM SET</td>
<td>Can’t do with fixed insertion</td>
</tr>
<tr>
<td>ERASE</td>
<td>If owner and members are fixed – delete whole set. If owner and members are mandatory – disallow. If owner and members are optional – delete owner only.</td>
</tr>
</tbody>
</table>

FIND owner-id WITHIN CAR-OWNER USING V1234567.
FIND NEXT WITHIN CAR-OWNER.
FIND OWNER WITHIN CAR-OWNER.