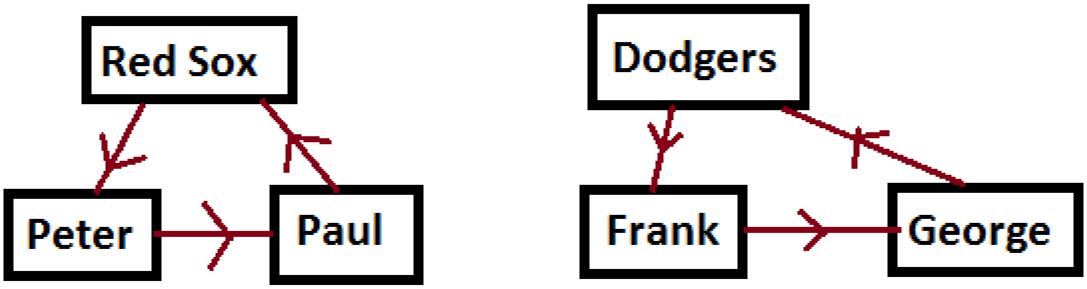
Network Model

The Network Model is based on a directed graph.

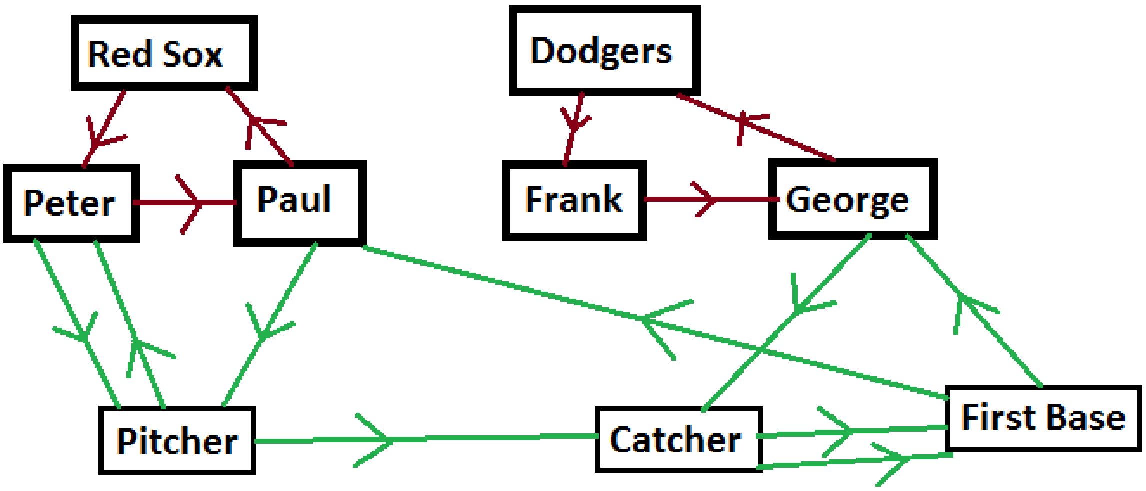
This model was developed by the Data Base Task Group of the CODASYL COBOL committee. In 1971 they proposed a schema DDL, a subschema DDL and a DML (to be embedded in COBOL programs).

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.

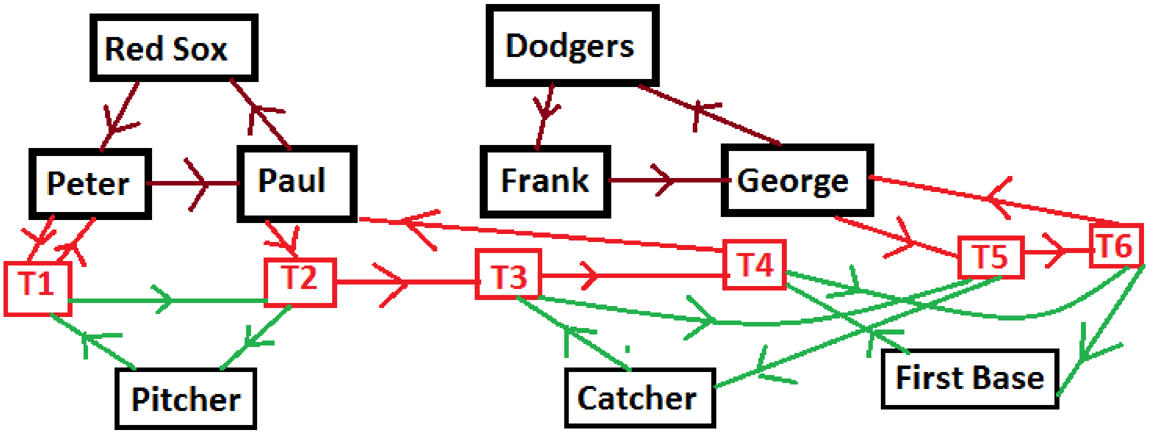


But the connection between Player and Position is really many- many. So we start with Peter and follow the green link to Pitcher. Which green edge do we follow OUT of Pitcher? Back to Peter? or on to Catcher?

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



DBTG DDL

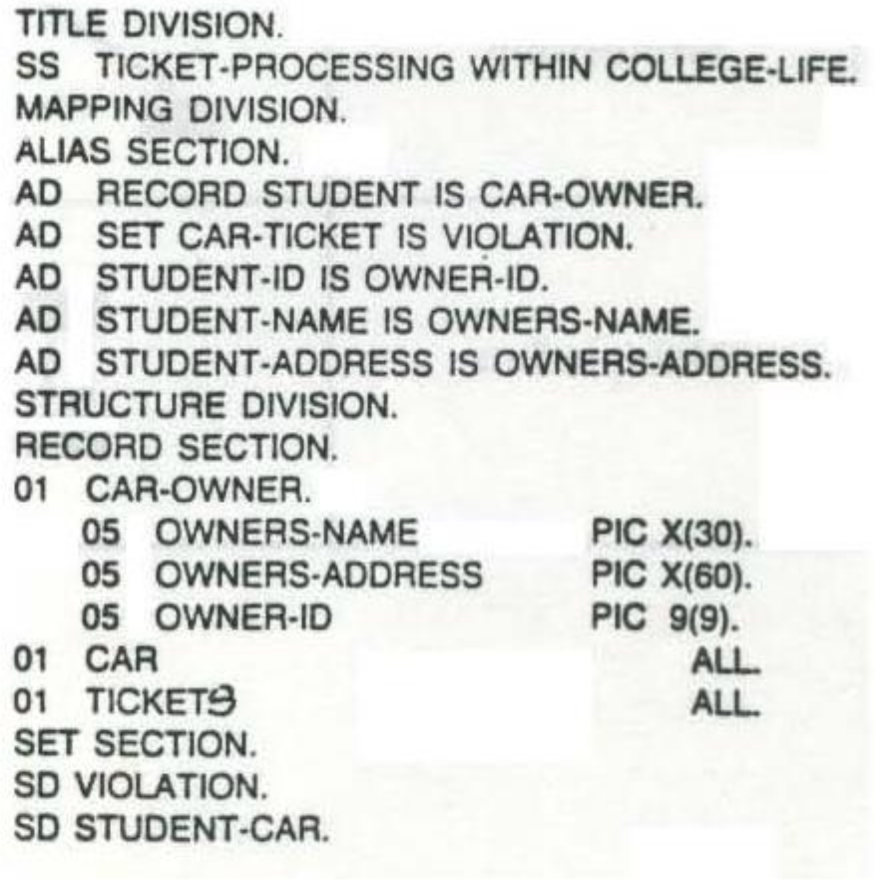
SCHEMA NAME IS name	Gives name of database
RECORD NAME IS name	Gives name of record (node on graph)
KEY name IS ascending/descending	Field in record to be used for sorting. Optional
DUPLICATES ARE NOT ALLOWED FOR name	This field cannot contain duplicate values. Optional
02 name TYPE is type	Gives name and type of field. Types are given using COBOL syntax.
SET NAME IS name	Gives name of the set
OWNER IS name	Record that is owner of set
MEMBER IS name	Record that is member of set
ORDER IS next/first/last/prior/system default/ sorted	Describes how members are ordered within the set
INSERTION is automatic/manual	Automatic: DBMS inserts member into set when it is created Manual: programmer must issue statement to insert into set
RETENTION is fixed/mandatory/optional	Fixed: members cannot be mored 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
SET SELECTION IS KEY name	Field of owner used to choose set occurrence.

```
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

SS name WITHIN schema name	Names subschema and associates it with schema
MAPPING DIVISION	
AD alias IS name	Rename set or record
STRUCTURE DIVISION	Gives records and sets to include



```
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                                ALL.
01   TICKETS                            ALL.
SET SECTION.
SD VIOLATION.
SD STUDENT-CAR.
```

DML

READY name USAGE MODE IS exclusive/protected	Gives name of subschema to use. Exclusive: me only Protected: allow other readers
FIND record WITHIN set USING x	Go to the first record within the given set type that has the given key value
FIND next/prior/first/last/owner WITHIN set	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.
GET	Copy current record into application program
MODIFY	Replace current record with data in application program
STORE	Insert new record into database. Does not connect to a set if insertion is manual
CONNECT record TO set	For manual insertion
DISCONNECT record FROM SET	Can't do with fixed insertion
ERASE	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.

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