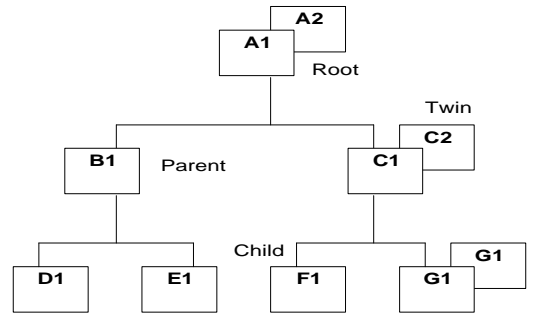


# Datenmodell

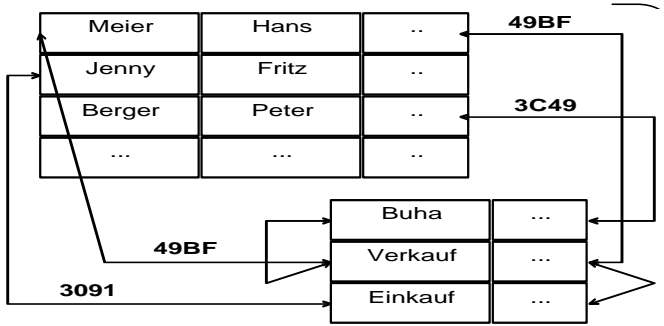
## Hierarchisches Datenbankmodell

- Ablage der Daten in Segmenten in einer Baumstruktur
  - Häufig im Einsatz, ca. 50 %, hauptsächlich bei Banken und Versicherungen
  - Oberstes Segment (Root) und übergeordnete Segmente (Parent) kann beliebig viele untergeordnete Segmente (Childs) besitzen
- Vorteile**
  - Gute Performance bei Batch
  - Relative Einfachheit und Benutzbarkeit
  - Existenz bewährter DBMS
  - Weit verbreitetes Know-how
- Nachteile**
  - Löschen der Parents löscht auch die Childs
  - Datendefinitionssprache erlaubt nur eine aufwendige nachträgliche Änderung der phys. Datenstrukturen
  - Regeln und Navigation



## Netzwerk Datenbankmodell

- Statische Verknüpfung der Datensätze mittels Pointer. Gezielter Zugriff auf jede beliebige Satzart
- Vorteile**
  - Gute Performance
  - Existenz bewährter DBMS
  - Leichte Implementierung von komplexen Beziehungen
- Nachteile**
  - Hohe Komplexität Der Programmierer muss die log. Datenstrukturen kennen
  - Kompliziert
  - Selten im Einsatz



## Relationales Datenbankmodell

- Ansammlung zweidimensionaler Tabellen welche über Schlüssel und damit ihre Datensätze beliebig untereinander verbunden werden
- Relation (zweidimensionale Tabelle)**
  - Eindeutiger Tabellenname
  - Innerhalb Tabelle eindeutiger Merkmalsname
  - Eines der Merkmale (Primarykey) identifiziert eindeutig die Tupel innerhalb der Tabelle
  - Merkmale und ihre Ordnung sind bedeutungslos
  - Anzahl der Tupel und die Ordnung innerhalb der Tabelle sind bedeutungslos
- Begriffe**
  - Datensatz einer Relation = Tupel (Zeile)
  - Merkmal (Attribut) = Bezeichnung jedes Eintrags in der Tabelle
  - Primärschlüssel = Eindeutige Identifikation der Datensätze einer Relation
  - Domäne = Merkmalswerte eines Attributs (Häufig Datum, Zeit oder numerische Werte)
- Sichten / Views**
  - View = virtuelle Zusammensetzung physisch getrennt gespeicherter Datensätze
  - Relationen werden mittels rel. Operationen so bearbeitet, dass die gewünschte Benutzersicht entsteht
  - Einsatzmöglichkeit: Ausblenden bestimmter Spalten oder Tupel
- Vorteile**
  - Unterstützung des Standards SQL
  - Datensicherung/Datenschutz
  - Portierbarkeit/Skalierbarkeit
  - Transaktionssteuerungskonzept
  - Werkzeuge und Tools
- Relationale Algebra**
  - Siehe Datenbank-Grundlagen

