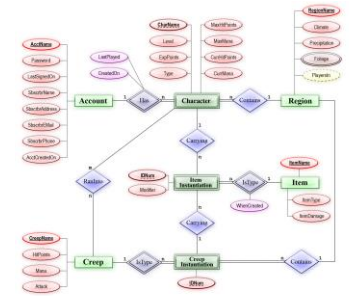
Modern Database Systems

Relational model, relational databases. History and overview of database models and systems.

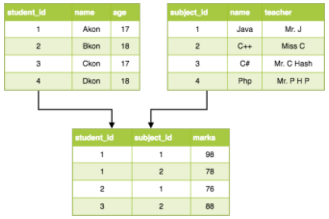
Doc. RNDr. Irena Holubova, Ph.D.
Irena.Holubova@matfyz.cuni.cz

Three layers of database modelling

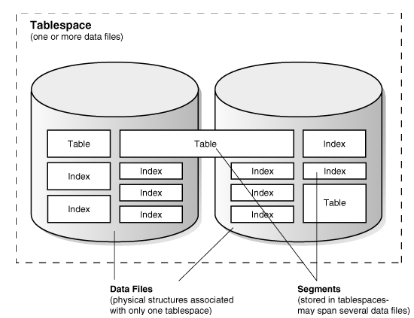
- Conceptual
  - Highest level of abstraction
  - Modelling of real-world objects and relationships
  - e.g., ER, UML, …



- Logical
  - Machine interpretable data structures
    for storing the modelled data
  - e.g., object, relational, object-relational, XML, graph, …



- Physical
  - How logical database structures are implemented in a specific technical environment
  - e.g., data files, index structures, …

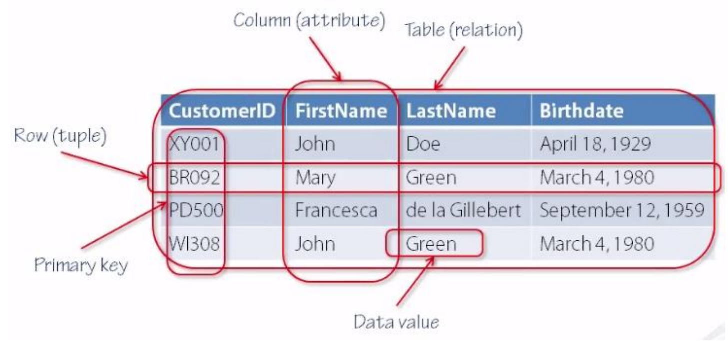


Database = relational database?

- A common assumption for many years
- Relational databases are able to store and process various data structures
- Advantages:
  - Simplicity
    - of the model
    - of the respective query language
  - After so many years mature and verified database management systems (DBMSs)
  - Strong mathematical background
  - …

ORACLE DATABASE MySQL Microsoft SQL Server IBM DB2 PostgreSQL

Relational model



- Proposed by E.F. Codd in 1970
  - Paper: “A relational model of data for large shared data banks”
  - IBM Research Labs
- Basic idea:
  - Storing of objects and their mutual associations in tables (relations)
    - A **relation** R from X to Y is a subset of the Cartesian product X × Y.
  - Row in a table (member of relation) = object/association
  - Column (attribute) = attribute of an object/association
  - Table (relational) schema = name of the schema + list of attributes and their types
  - Schema of a relational database = set of relational schemas

Relational model

- Basic integrity constraints
  - Unique identification of a row
    - Super key vs. key
  - Simple type attributes
  - NULL values
    - No “holes”
- Keys/foreign keys

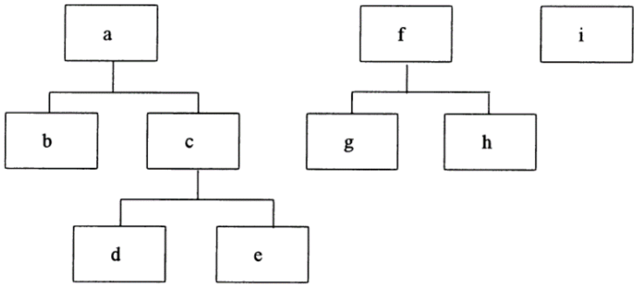
Further details: course Database Systems (NDBI025)

But the relational model was not the first one…

time

- First generation: navigational
  - Hierarchical model
  - Network model
- Second generation: relational
- Third generation: post-relational
  - Extensions of relational model
    - Object-relational
  - New models reacting to popular technologies
    - Object
    - XML
    - NoSQL (key/value, column, document, graph, …) - Big Data
  - Multi-model systems
  - …
  - Back to the relations
    - NewSQL

Hierarchical model



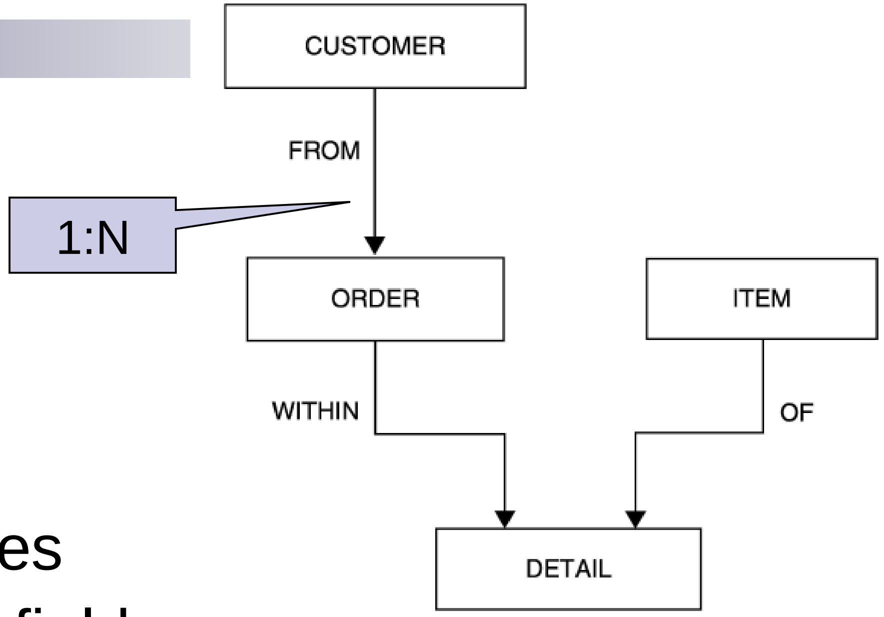
- Idea: Data are organized into records that are recursively composed of other records
- IBM's IMS (Information Management System)
  - Released in 1968
    - Still used! (https://www.ibm.com/it-infrastructure/z/ims)
  - One of the first commercially available DBMS
- Forest of trees
  - One-to-many relationships
  - First independent = redundancy
    - A record cannot be stored in two different trees without duplication
  - Later links and sharing
- Data processing: hierarchical, starting from the root, depth-first, left-to-right traversal order
  - First storage on tapes – linear access
  - Later (arrival of discs) direct access thanks to hashing and B-tree techniques

Suitable for the original use case but not in general

Network model

- Also CODASYL data model
  - Conference/Committee on Data Systems Languages
    - Consortium formed in 1959 to guide the development of a standard programming language (COBOL)
    - Also focussed on databases
- Defined in 1971
- Idea: data records connected through binary relationships
  - Data processing: navigational primitives according to which records are accessed and updated one at a time
    - Relational query languages: set orientation
- 1973 – report describing:
  - General architecture of a DBMS
  - Schema DDL + DML + Sub-schema DDL (interfaces, i.e., views) + DSDL (data storage, i.e., physical structure)

Network model



- Nodes = record types
  - Represent real-world entities
  - Have atomic or compound fields
  - Record = a data unit
    - Has an identifier
- Edges = set types
  - 1:N relationship type
  - A list of records = head record + members of the set
- Close to the ER model

Network model

user work area

```
CUSTOMER.CUST-NO := "C400"                   /* store search key value in the UWA
find CUSTOMER record                         /* find CUSTOMER record "C400"
find first ORDER record of FROM set          /* find first ORDER record owned by this CUSTOMER record
while ERROR-COUNT = 0
    get ORDER                                /* get item values of current ORDER record
    <process item values of current ORDER>
    find next ORDER record of FROM set       /* find first ORDER record
end-while

CUSTOMER.CUST-NO := "C400"
find CUSTOMER record                         /* make CUSTOMER record "C400" the current of FROM
if ERROR-COUNT = 0 then
    ORDER.ORD-NO := 30183                    /* store value of ORD-NO in the UWA
    ORDER.CUST-DATE := "2008/08/19"          /* store value of ORD-DATE in the UWA
    store ORDER                              /* store ORDER record and insert it in the current FROM set
end-if
```

Relational model

- Optimal for may applications, **but…**
- New application domains have appeared
  - e.g., GIS
  - Complex data types not supported by the relational model
- Normalizing data into table form affects performance for the retrieval of large, complex, and hierarchically structured data
  - Numerous joins
- Object-oriented programming languages (OOPLs) have appeared
  - Defined the concept of user-defined classes

Object model and object databases

- Approach I.: extend objects with data persistence, i.e., databases
  - Approx. early to mid-1970s
- Objects = basis for modelling in a database application
  - An instance of a class
- Data stored as a graph of objects (in terms of OOP)
  - Suitable for individual navigational access to entities
  - Not suitable for “batch operations” (data-intensive applications)
- The goal: the programmer does not have to take care of object hierarchy persistency
  - Comfort support in software development platforms
    - e.g., Hibernate in Java or ADO.NET Entity Framework
  - Application data is loaded/stored from/to the database as needed
  - The data exists regardless of the application runtime

Object-relational databases

- Approach II.: extend databases with objects
  - Approx. early 1990s
  - Aim: to bridge the gap between relational databases and object-oriented modelling techniques used in programming languages
- Relational model enriched with:
  - Objects, classes, inheritance, complex types of attributes
  - Custom data types, methods/functions
- A middle ground between relational databases and object-oriented databases

Object-relational databases

```
CREATE TYPE StockItem_objtyp AS OBJECT
  (
    StockNo NUMBER,
    Price NUMBER,
    TaxRate NUMBER );

CREATE TYPE LineItem_objtyp AS OBJECT (
    LineItemNo NUMBER,
    Stock_ref REF StockItem_objtyp,
    Quantity NUMBER,
    Discount NUMBER );

CREATE TYPE PhoneList_vartyp AS VARRAY(10) OF VARCHAR2(20);

CREATE TABLE Customer_objtab OF Customer_objtyp (CustNo PRIMARY KEY) OBJECT
      IDENTIFIER IS PRIMARY KEY;
```

Object-relational databases

```
CREATE TYPE PurchaseOrder_objtyp AUTHID CURRENT_USER AS OBJECT (
  PONo                NUMBER,
  Cust_ref            REF Customer_objtyp,
  OrderDate           DATE,
  ShipDate            DATE,
  LineItemList_ntab   LineItemList_ntabtyp,
  ShipToAddr_obj      Address_objtyp,

  MAP MEMBER FUNCTION
    getPONo RETURN NUMBER,

  MEMBER FUNCTION
    sumLineItems RETURN NUMBER
  );
```

Object-relational databases

```
CREATE OR REPLACE TYPE BODY PurchaseOrder_objtyp AS

MAP MEMBER FUNCTION getPONo RETURN NUMBER is
  BEGIN
    RETURN PONo;
  END;

MEMBER FUNCTION sumLineItems RETURN NUMBER is
    i          INTEGER;
    StockVal     StockItem_objtyp;
    Total        NUMBER := 0;

  BEGIN
    FOR i in 1..SELF.LineItemList_ntab.COUNT LOOP
      UTL_REF.SELECT_OBJECT(LineItemList_ntab(i).Stock_ref,StockVal);
      Total := Total + SELF.LineItemList_ntab(i).Quantity * StockVal.Price;
    END LOOP;
    RETURN Total;
  END;
END;
```

XML model and databases

Word Wide Web Consortium

- XML – W3C markup language
  - DTD, XML Schema, XPath, XQuery, XSLT, …
- XML databases
  - Native vs. XML-enabled
  - Support for XML data type + related technologies
- SQL/XML (≠ SQLXML !)
  - XML data type (XML value)
  - Extension of SQL
    - Data publication (XMLELEMENT, XMLATTRIBUTES, XMLAGG, …)
    - Querying (XMLFOREST, XMLTABLE, XMLEXISTS)

XML model and databases

```
-<bookstore>
  -<book category="cooking">
      <title lang="en">Everyday Italian</title>
      <author>Giada De Laurentiis</author>
      <year>2005</year>
      <price>30.00</price>
   </book>
  -<book category="children">
      <title lang="en">Harry Potter</title>
      <author>J K. Rowling</author>
      <year>2005</year>
      <price>29.99</price>
   </book>
  -<book category="web">
      <title lang="en">XQuery Kick Start</title>
      <author>James McGovern</author>
      <author>Per Bothner</author>
      <author>Kurt Cagle</author>
      <author>James Linn</author>
      <author>Vaidyanathan Nagarajan</author>
      <year>2003</year>
      <price>49.99</price>
   </book>
  -<book category="web" cover="paperback">
      <title lang="en">Learning XML</title>
      <author>Erik T. Ray</author>
      <year>2003</year>
      <price>39.95</price>
   </book>
 </bookstore>
```

```
for $x in doc("books.xml")/bookstore/book
where $x/price > 30
order by $x/title
return $x/title
```

```
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

But the relational model still beats them all…

And then the Big Data has arrived…

References

- Ling Liu, M. Tamer Özsu: Encyclopedia of Database Systems. Springer 2009
- https://docs.oracle.com/cd/B19306_01/appdev.102/b14260/adobjxmp.htm