پوهنتون کابل

پوهنحی کمپیوترساینس

Introduction to Database and Data Models

Lectures 05-06

تهیه کننده : پوهنیار محمد شعیب "زرین خیل"
سال : 1389

Introduction to Database and Data Models - Relational Model

05

By: M Shuaib Zarinkhail

2010

Database As a Model

- A database is a model
- Not a model of some actual business reality!
- Rather, it is a model of the user's 'MODEL' of that reality
  - i.e.a representation of a view of reality that allows the user to keep track of important information

Database As a Model

- The amount of detail in the model depends on the level of information required
  - This is a critical part of defining a database
- Must be dynamic
  - Support changes

DB Creation -Summary of Steps

1. Create a Data Model
2. Transpose model into normalized relations
3. Create the database Tables
4. Define relationships between tables
5. Define component applications (metadata)
   - Forms, queries, reports, menus and / or application programs

Creating a Database – Step #1

- Create a Data Model:
  - Identify the Universe of Discourse (UoD) of a field or subject
- Definitions: UoD = the subset of model of reality that we are interested in implementing in a database

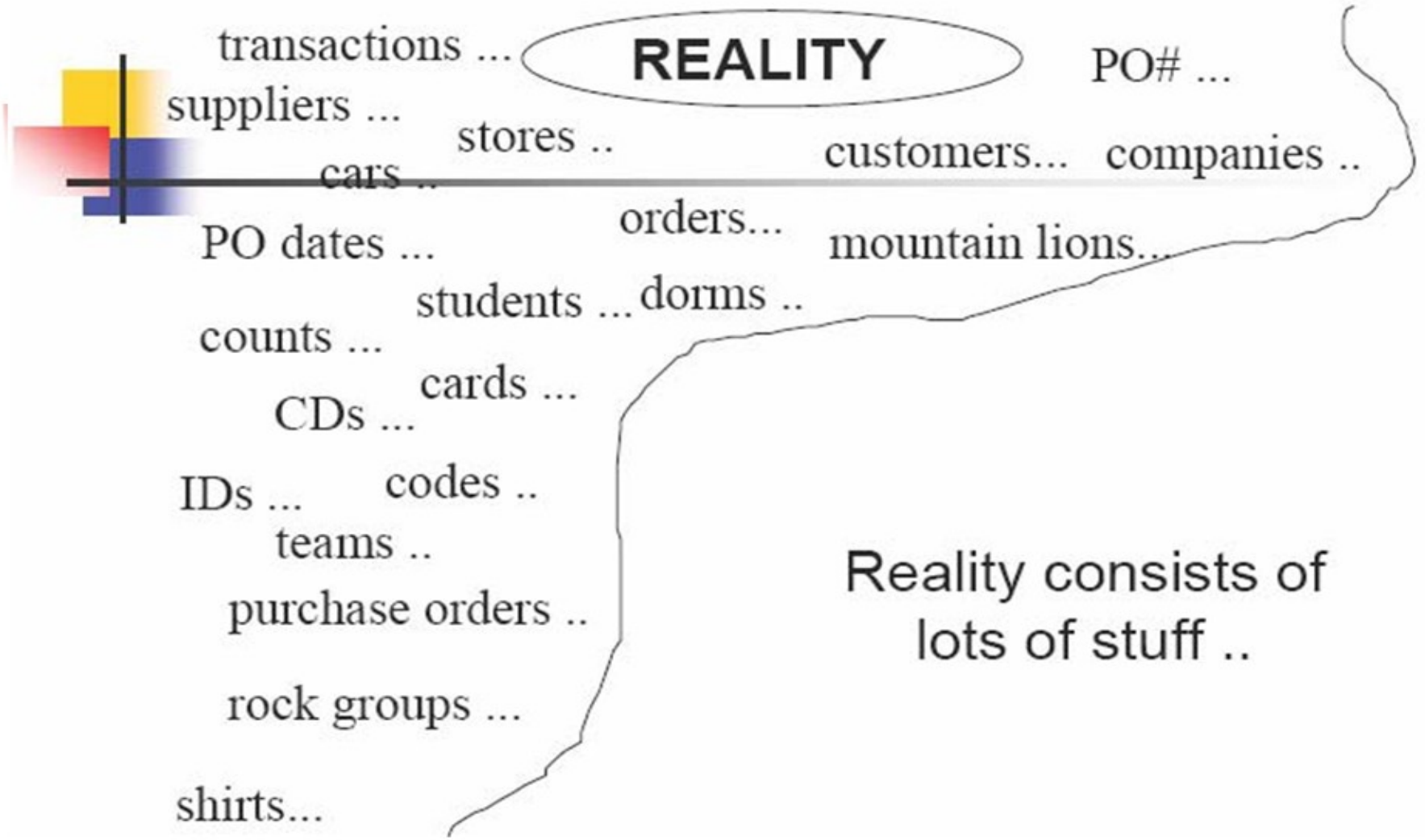
Creating a Database – Step #1

- A Model may consists of:
  - Entity Classes - Object Classes
  - Attributes of properties
    - Domains
  - Relationships
  - Business Rules
- As UoD changes, change Data Model to reflect new reality

Creating a Database – Step #1

Data Modeling

- The process of determining what items are of importance to the:
  - Users
  - Relationships between items
  - Rules that users need to be in forced
  - Creating a Data Model to represent all them


transactions ...
REALITY
PO# ...
suppliers ...
stores ..
customers...
companies ..
cars ..
orders...
PO dates ...
mountain lions...
students ...
dorms ..
counts ...
cards ...
CDs ...
IDs ...
codes ..
teams ..
purchase orders ..
rock groups ...
shirts...
Reality consists of
lots of stuff ..

transactions ...

UoD

PO# ...

suppliers ...

customers...

PO dates ...

orders...

counts ...

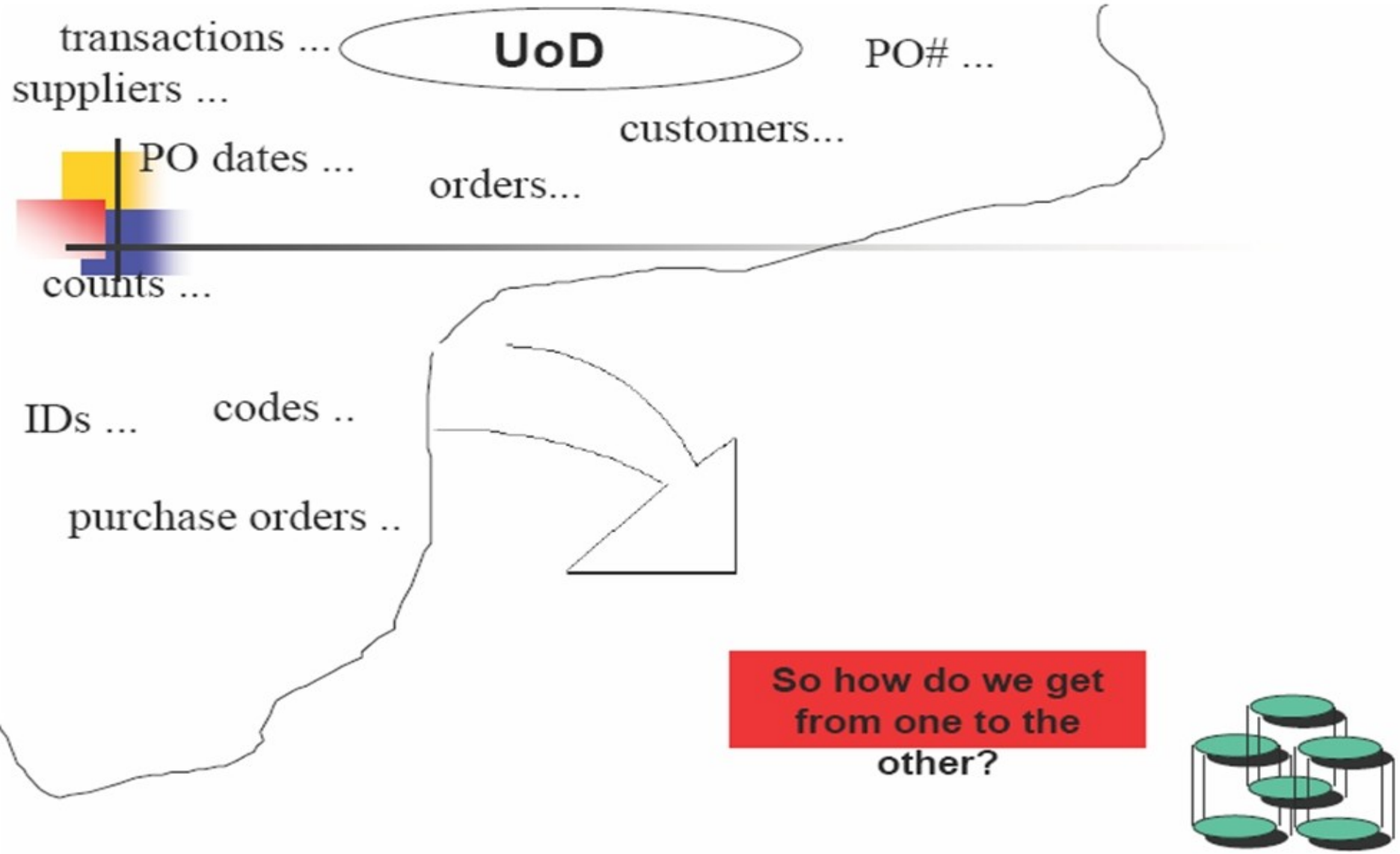
IDs ...

codes ..

purchase orders ..

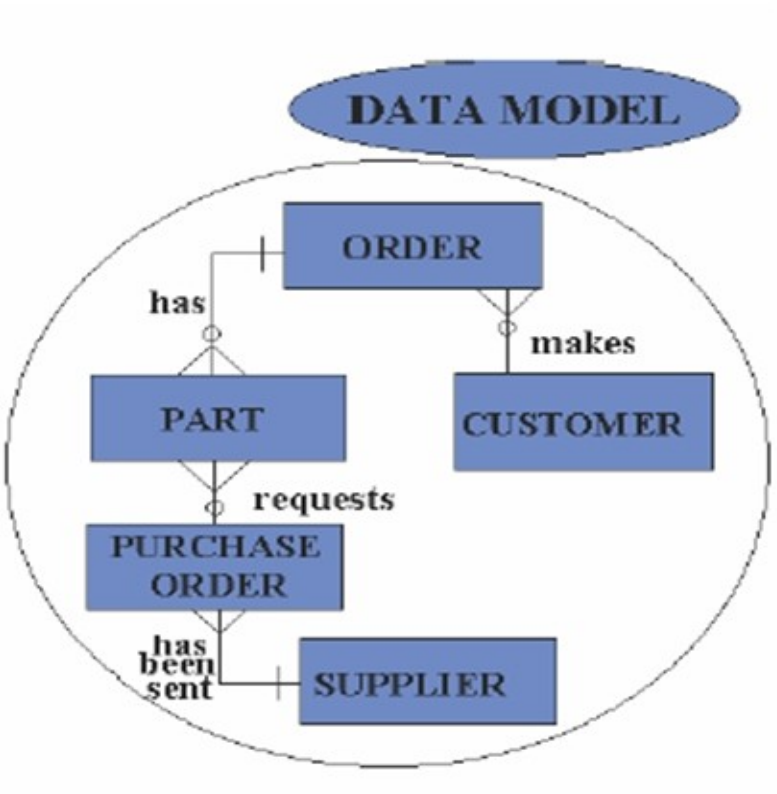
A Universe of Discourse (UoD) is some subset of that stuff that we are interested in representing with a computer-based information system (usually a DBMS).

As our UoD changes, we want to modify the computer-based information system to reflect those changes

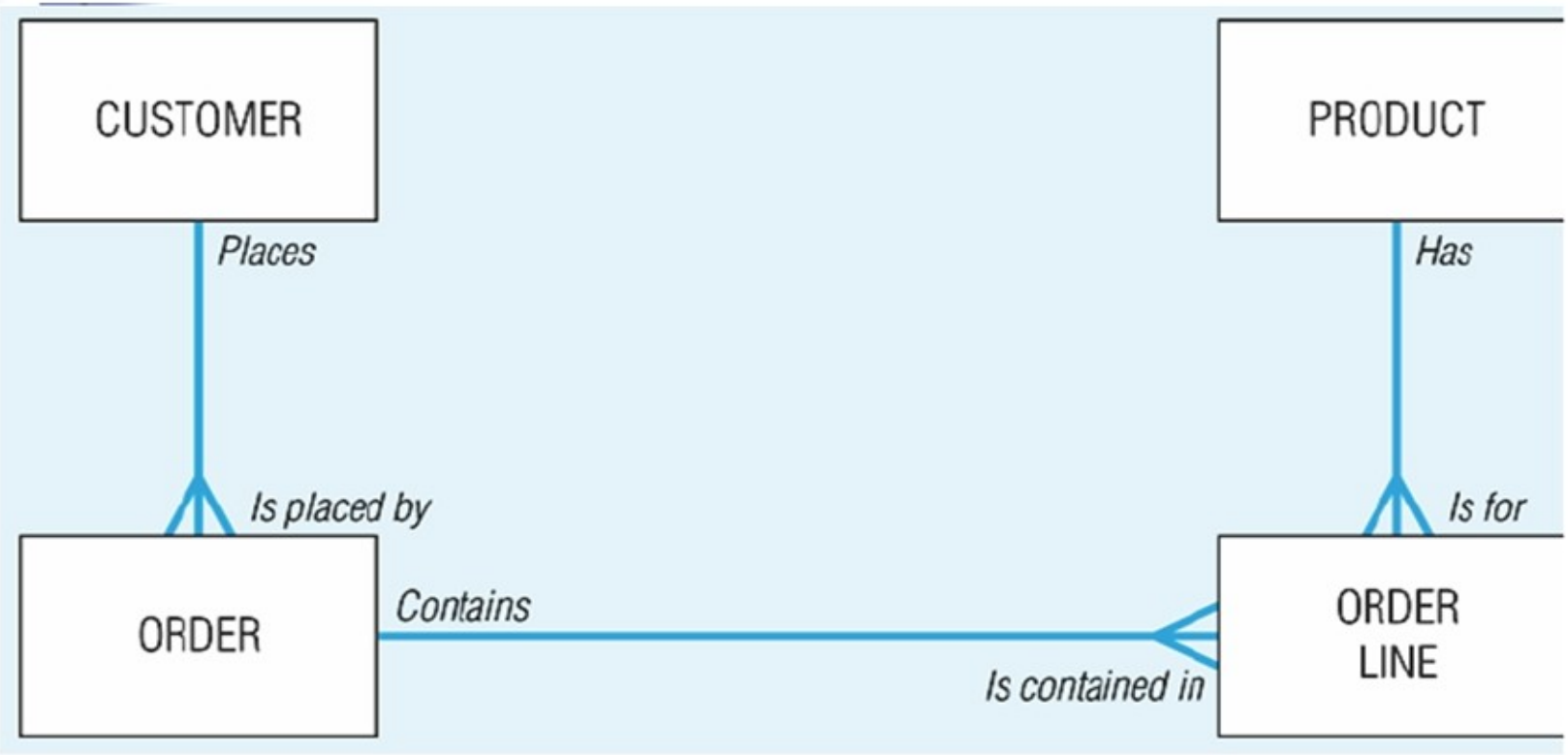

transactions ...
UoD
PO# ...
suppliers ...
customers...
PO dates ...
orders...
counts ...
IDs ...
codes ..
purchase orders ..
So how do we get from one to the other?

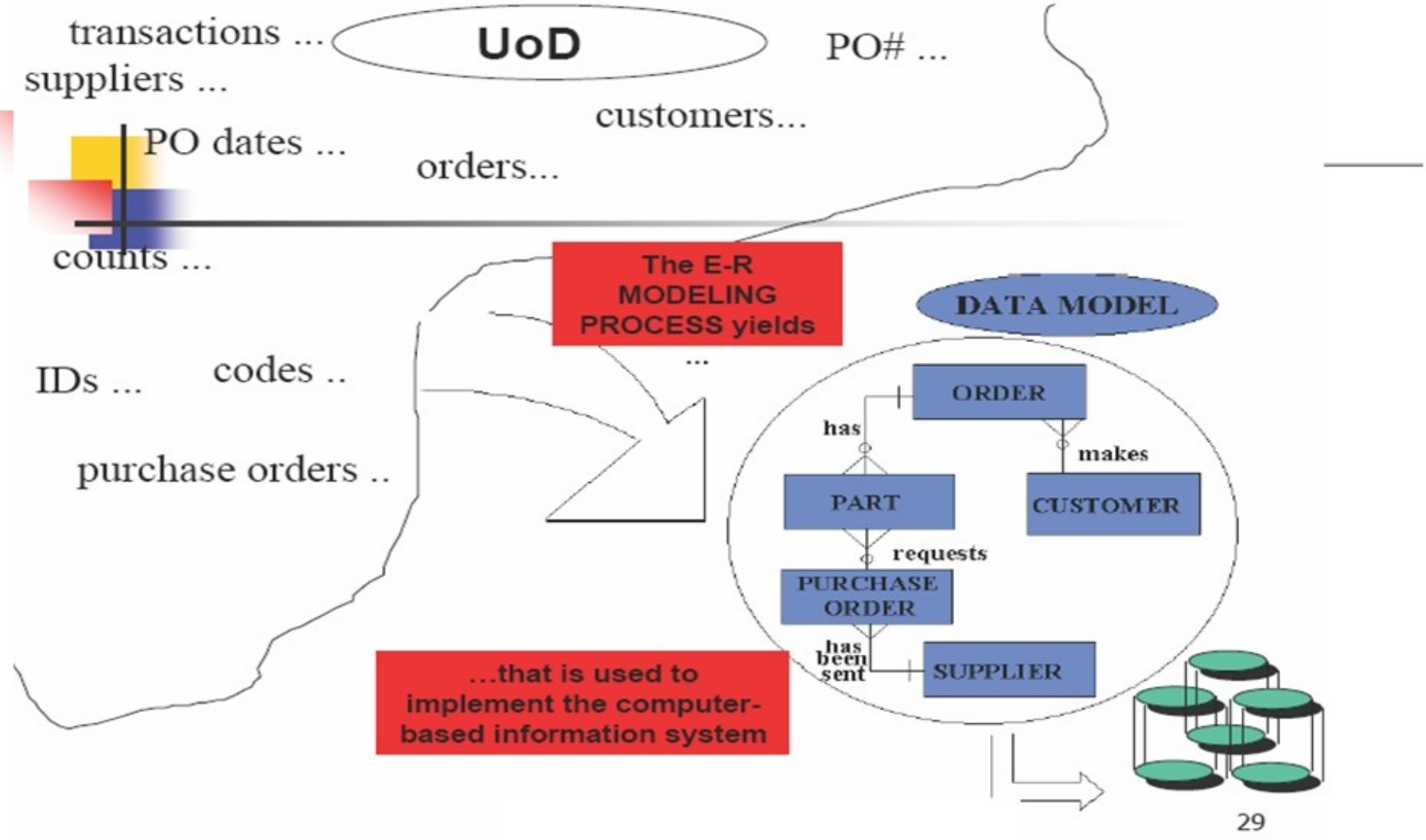
We begin with a .. DATA MODEL

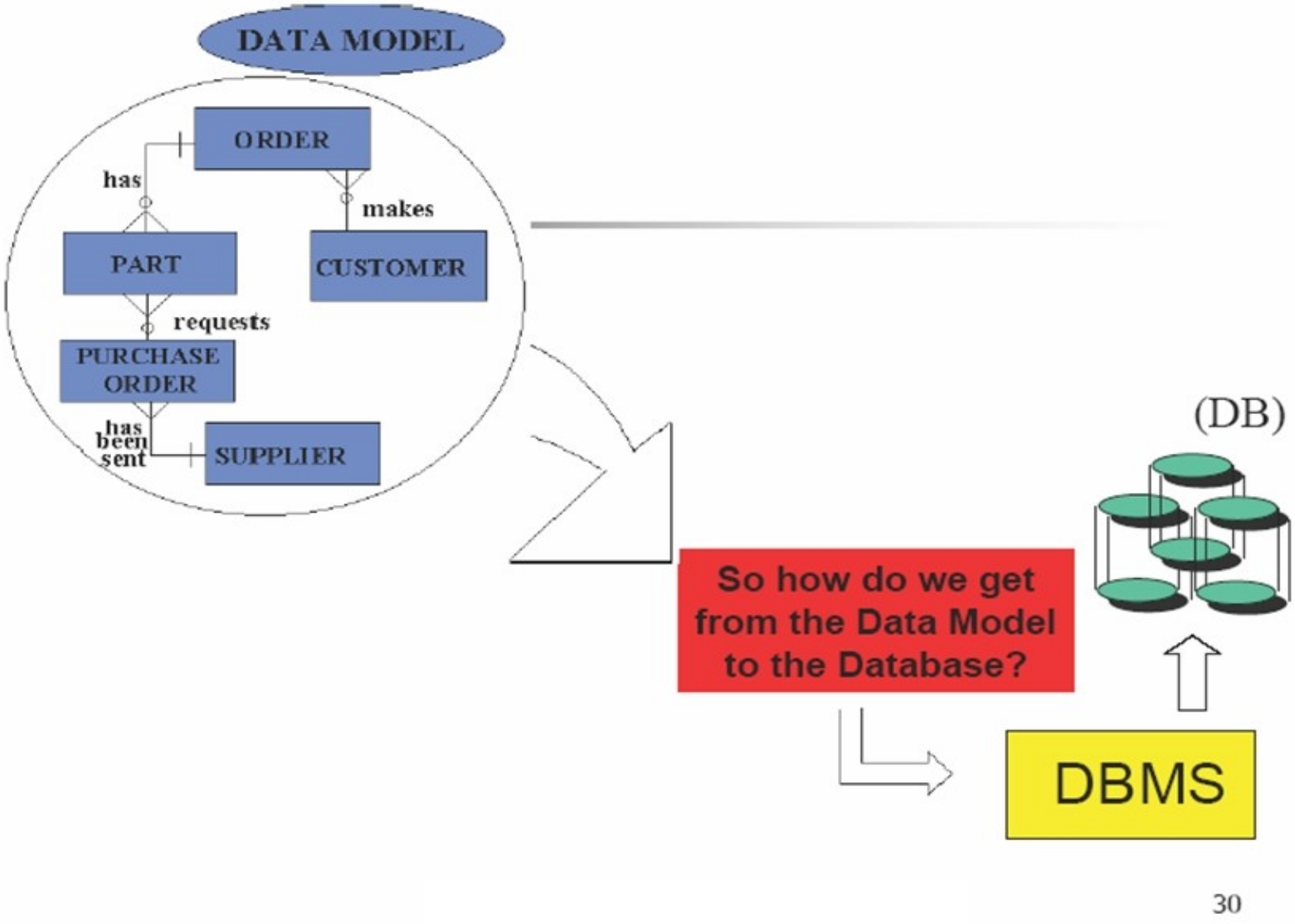
- A data model is a set of concepts that can be used to capture a UoD which in turn can be used to implement a computer-based information system.



A segment from enterprise data model




transactions ...
UoD
PO# ...
suppliers ...
customers...
PO dates ...
orders...
counts ...
The E-R MODELING PROCESS yields
...
DATA MODEL
IDs ...
codes ..
purchase orders ..
ORDER
has
makes
PART
CUSTOMER
requests
PURCHASE ORDER
has been sent
SUPPLIER
...that is used to implement the computer-based information system
29


DATA MODEL
ORDER
has
makes
PART
CUSTOMER
requests
PURCHASE ORDER
has been sent
SUPPLIER
(DB)
So how do we get from the Data Model to the Database?
DBMS
30

UoD

HIGH-LEVEL or *conceptual* level
(e.g., E-R diagram)

DBMS - INDEPENDENT

DBMS - SPECIFIC

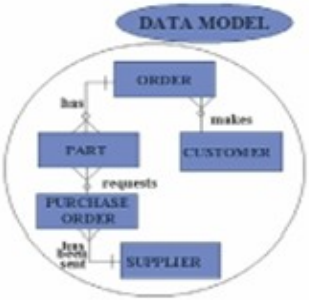
LOW-LEVEL or *physical* level
(e.g., file layouts/structures,
indexing, OS access strategies)

31

This leads to an additional layer of modeling!

UoD

HIGH-LEVEL or *conceptual* level
(e.g., E-R diagram)



DBMS - INDEPENDENT

DBMS - SPECIFIC

REPRESENTATIONAL level

DBMS software
which (to access the data), uses
the ...

DBMS

LOW-LEVEL or *physical* level
(e.g., file layouts/structures,
indexing, OS access strategies)

32

Creating a Database – Step #2

- Transpose the model into a set of relations
- Then form the schema that have to be implemented
  - Def Schema: the design for a database, including its tables, relationships, domains and constraints, that is used as the foundation for creating a database and its applications

Creating a Database – Step #2

Syntax:

- TableName (list of column data items)
  - Identifiers
- Relationship lines and possibly names
  - Including relationship types
- Different data model techniques
  - Hierarchical, network, relational, etc
- Apply Normalization rules to eliminate data anomalies

Creating a Database – Step #3

- Change data model relations to database tables
  - This should be based on the schema or data model already explained in step #2
- Name tables for identity within a database file
- Name columns for identity within a table
  - Table names in a database and column names in a table should not duplicate

Creating a Database – Step #3

- Set data type for each column
  - This is important for data quality
  - This required for relationships
- For relationships between two tables "RIC" should perform
  - In RIC, domain of two fields (the index field from parent table, and the foreign key field from child table should use the same data types and domains)

Creating a Database – Step #4

- Define relationships between database tables
  - This can be done during step #3 of creating DB
- Define relationship rules (optional)
  - i.e. on delete / on update (cascade, no action, …)
- Implement business rules (if any)

Creating a Database – Step #5

- Define the Application Metadata Components
- Forms
  - Data display that typically show one record
  - Usually on screen in a window
  - All or selected fields of a record are displayed
  - Usually allow updating

Creating a Database – Step #5

- Define the Application Metadata Components
- Queries
  - Views of retrieved data that answers a question
  - Can be done by
    - Structured Query Language (SQL)
    - Query by Example (QBE)
    - Query by Form (QBF)

Creating a Database – Step #5

- Structured Query Language (SQL)
  - Standard query language
  - Uses by most of DBMSs
- Query by Example (QBE)
  - Use DBMS tools to identify tables, relationships and data items to be analyzed
- Query by Form (QBF)
  - Queries done by entering search constraints

Creating a Database – Step #5

- Define the Application Metadata Components
- Reports
  - Neatly organizes and summarizes data in a DB
  - Can be based on a query, view, or a table
  - Data can be summarized in sections
  - Typically meant to be printed, also good for display

Creating a Database – Step #5

- Define the Application Metadata Components
- Menus
  - Allow DB developers to tailor application ot a specific user
  - Control access to application components
  - Make applications easier to use by displaying options and helping users to choose actions

Creating a Database – Step #5

- Define the Application Metadata Components
- Application Programs
  - Special data retrieval or update software that are written in either
    - A built-in languages (i.e. MS Access has Macro Language and VBA)
    - A standard programming language that interfaces DBMS

Maintenance

- After a database has been created, it then must be maintained
- For this purpose, the followings are needed
  - Trained personnel
  - Updated software
  - Capable hardware
  - Enough storage media

Introduction to Database and Data Models - Relational Model

06

By: M Shuaib Zarinkhail

2010

DBMS - INDEPENDENT

Functions

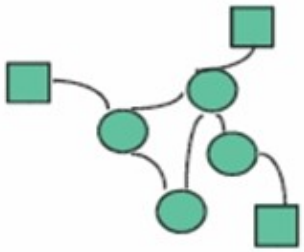
(functions, inputs, outputs)

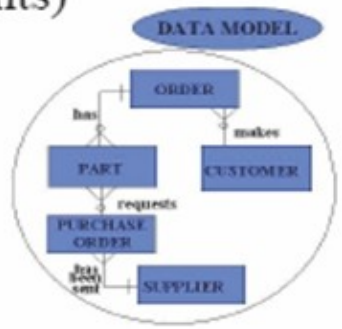
DBMS - INDEPENDENT

Objects

(entities, attributes, relationships, constraints)

HIGH-LEVEL or *conceptual* level





REPRESENTATIONAL level

DBMS software

which (to access the data), uses the ...

LOW-LEVEL or *physical* level

DBMS - SPECIFIC

Processes

(macros, forms., reports, queries, etc.)

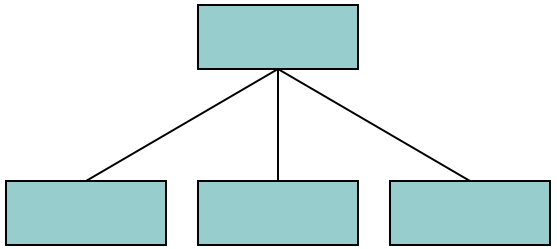
DBMS - SPECIFIC

Structures

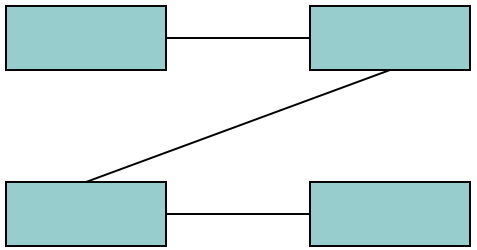
(tables, indexes, file structures, etc.)

Hierarchical and Network Models

- Hierarchical Model
  - Very limited
  - Very slow



- Network Model
  - Less restrictive than hierarchical model



Relational Model

- First proposed by Dr E F Codd in 1970s
- Then finalized in 1985, applicable up to date
- Applied Relational Algebra to the problem of managing and accessing large amounts of data

Relational Model

- New design intended to eliminate problems with databases
- Especially, Large Institutional Databases, accessible only via complicated programs and file systems
- Not responsive to user needs

Relational Model – Design Features

- Data stores as rows and columns in tables
- Values in row / columns use to relate / link rows across tables
- Provides a visualization of relationships
- Keeps data item duplication between tables to a minimum
- Uses a formal process called Normalization to setup optimum tables

Relational Model

GOAL:

- Simplicity
  - Allow end users to access databases directly and simply
  - By process, it is too difficult, but professional can easily develop and access for them

The Relational Model Cont…

- In 1985 Dr. Codd finalized 13 (12 + 1) rules for relational model
- These rules are still being used for relational databases

Details of those rules are in coming slides

13 Rules – Relational DB 1

- **DB management**
  - A relational DBMS must be able to manage database entirely through its relational capabilities
- **Information rule**
  - All information is represented explicitly as a value in tabular format

13 Rules – Relational DB 2

- **Guaranteed access**
  - Every value in a relational database is guaranteed to be accessible by using a combination of
    - table name, column name, primary key value (Index)
- **Systematic null value support**
  - The DBMS provides systematic support for the treatment of *null values*
    - distinct from default values
    - independent of any domain

Null Value: unknown or inapplicable data

13 Rules – Relational DB 3

1. **Active, online relational catalog**
   - The description of the database and its contents is represented at the logical level in tabular format
   - and can therefore be queried using the database language

13 Rules – Relational DB 4

- **Comprehensive data sublanguage**
  - At least one supported language must have a well-defined syntax and be comprehensive

    It must support
    - data definition - manipulation
    - integrity rules - authorization
    - transactions

13 Rules – Relational DB 5

- **View updating rule**
  - All views that are theoretically updatable can be updated through the system
- **Set-level insertion, update, and deletion**
  - Supports not only set-level retrievals but also set-level inserts, updates, and deletes

13 Rules – Relational DB 6

- **Physical data independence**

Application programs and ad hoc programs are logically unaffected when

- physical access methods or storage structures are altered

- **Logical data independence**

Application programs and ad hoc programs are logically unaffected, to the extent possible when

- changes are made to the table structures

13 Rules – Relational DB 7

- **Integrity independence**

The database language must be capable of defining integrity rules

  - These rules must be stored in the online catalog, and they cannot be bypassed

- **Distribution independence**

Application programs and ad hoc requests are logically unaffected when

  - data is first distributed - data is redistributed

13 Rules – Relational DB 8

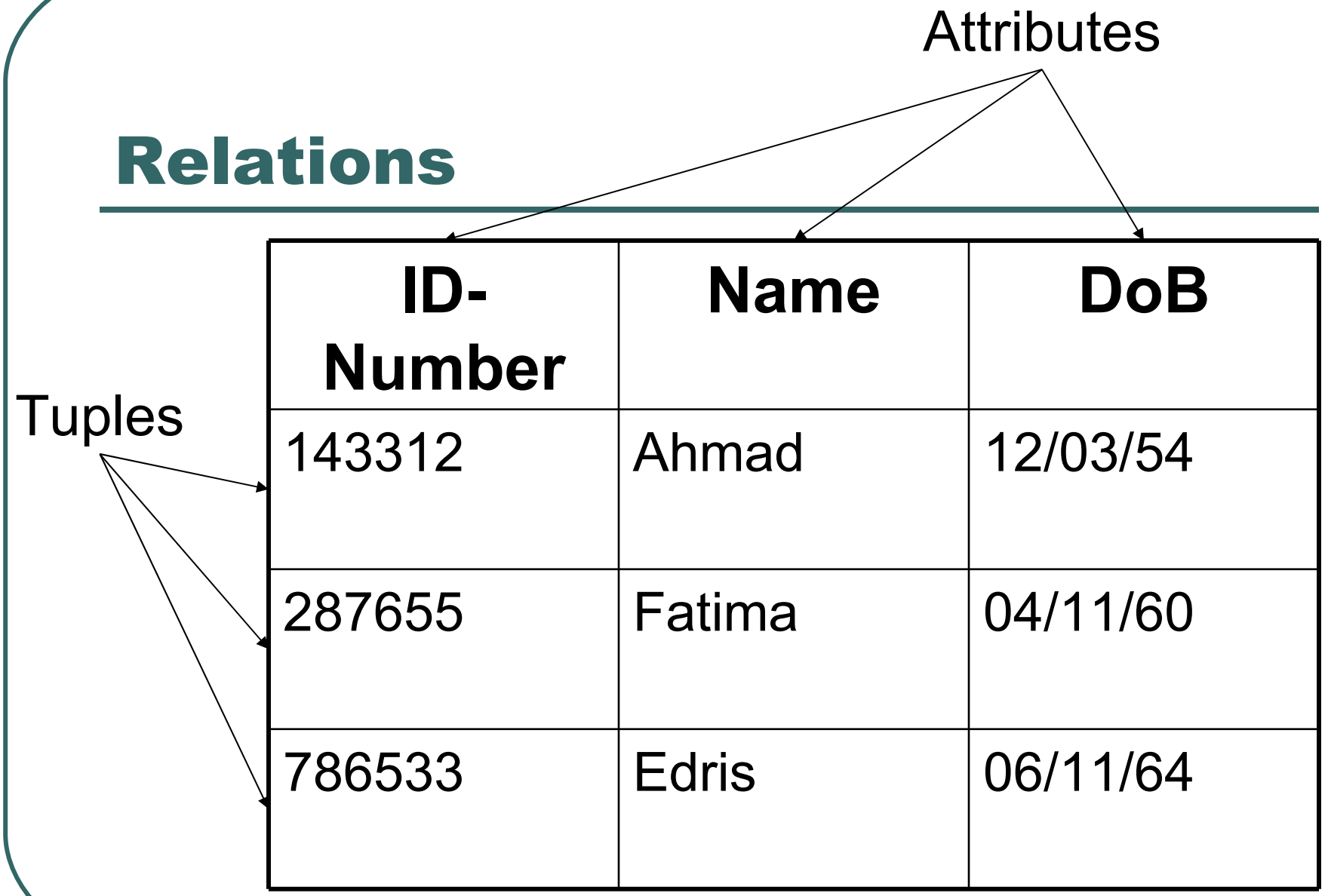
- **Non-subversion**
  - There should be no way to modify the DB structure other than through the multiple row database language (like SQL)
  - It must not be possible to bypass the integrity rules defined through the database language by using lower-level languages

Relations

Definition:

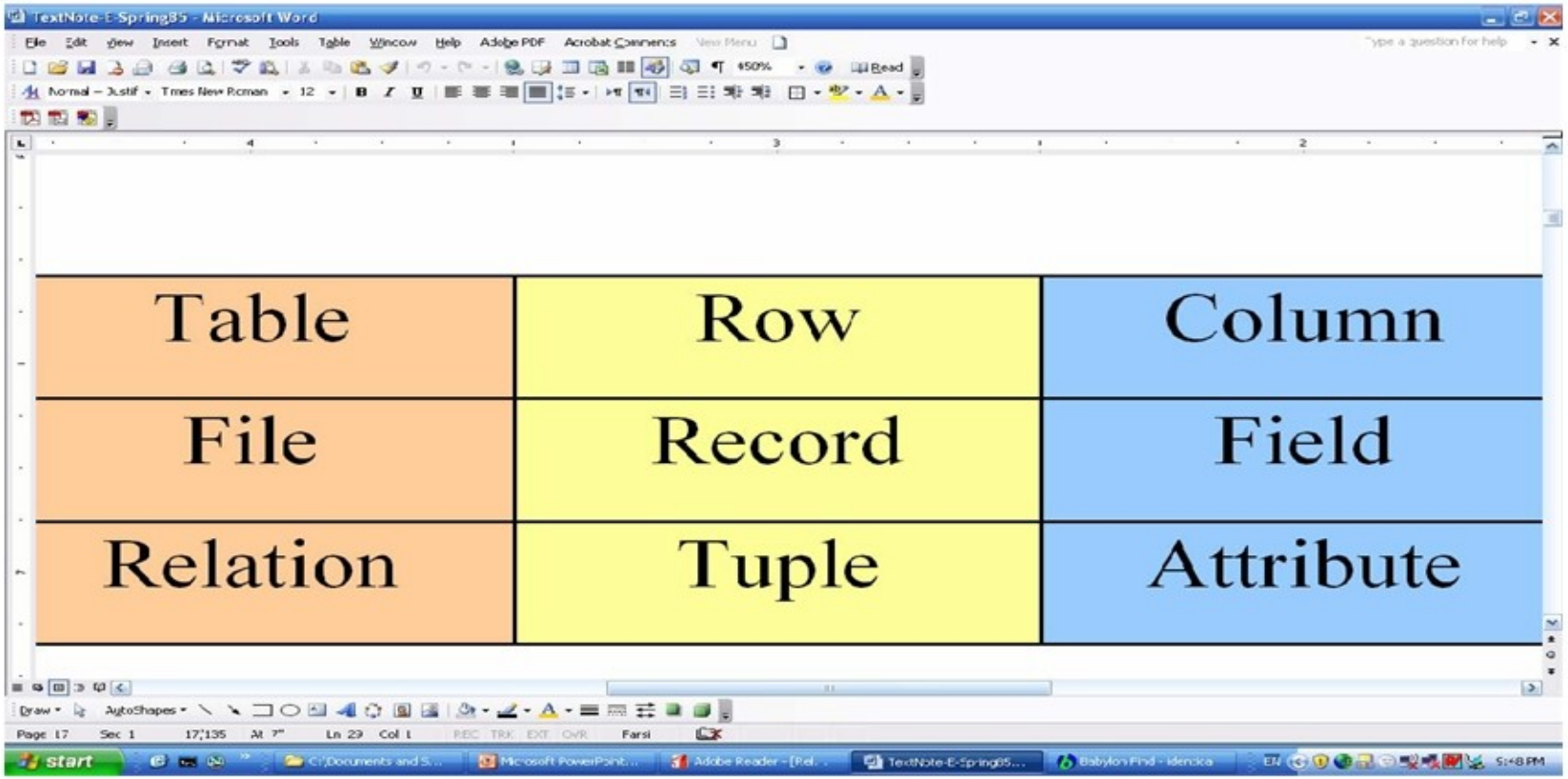
- A relation is a two dimensional table that holds data pertaining to an object of interest called Entity.

Relations



ID-Number	Name	DoB
143312	Ahmad	12/03/54
287655	Fatima	04/11/60
786533	Edris	06/11/64

Relation, Table and File



Properties of Relations

- All values in a column must be of the same type
- All of the cells in a relation must be single value (No repeating groups or arrays in a cell)
- Column names must be unique within a relation
- Order of columns and rows is not relevant
- No two rows can be identical

Valid Relation?

ID-Number	Name	DoB
143312	Ahmad	12/03/54
287655	Fatima	04/11/60
143312	Ahmad	12/03/54

Valid Relation?

ID-Number	Name	DoB
143312	Ahmad	12/03/54
287655	Fatima	04/11/60
786533	Edris	06/11/64

Valid Relation?

ID-Number	Name	DoB
143312	Ahmad	12/03/54
287655	Fatima	Accounting
786533	Edris	06/11/64

Valid Relation?

ID-Number	Name	Phone
143312	Ahmad	0786111111
287655	Fatima	0786222222 0786333333
786533	Edris	0786444444

Valid Relation?

ID-Number	Name	DoB
143312	Ahmad	12/03/54
287655	Fatima	04/11/60
786533	Edris	06/11/64

ID-Number	DoB	Name
143312	12/03/54	Ahmad
287655	04/11/60	Fatima
786533	06/11/64	Edris

Valid Relation?

ID-Number	Name	DoB
143312	Ahmad	12/03/54
287655	Fatima	04/11/60
786533	Edris	06/11/64

ID-Number	Name	DoB
287655	Fatima	04/11/60
143312	Ahmad	12/03/54
786533	Edris	06/11/64

Notation

- Relations can be represented in text forms:

RELNAME(Attribute1, Attribute2, … AttributeN)

Example:

STUDENT(ID_Number, Name, DoB)

Relations in Entity Relationship Diagrams (ERD)

- Entities are represented as Relations
- Relations are shown as boxes in ERD
- Relationships between entities shown by lines in ERD
- Then, the relationship lines shown by ‘Foreign Keys’ in practical part