The Relational Model

IDBS - Lecture 1.pdf

The Relational model is the foundation of relational databases. The model is founded by the Turing Award winning computer scientist Edgar F. Codd.

The **Relational Model** is

> a formal data model with a sound mathematical foundation, based on Set theory and first order predicate logic in which a database is represented as a collection of Relations.
> PDBM , page 1.703

Database

a collection of related data items within a specific business process or problem setting stored on a computer system through the organization and management of a database management system.

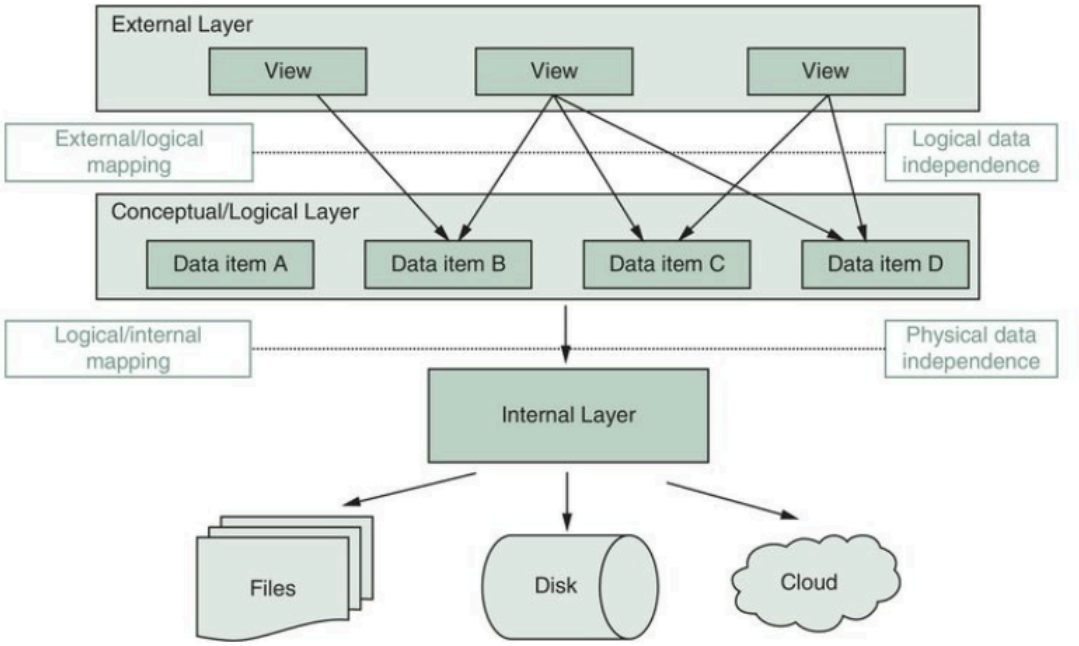
Database management system (DBMS)

A database management system (DBMS) is the software package used to define, create, use and maintain a database. These softwares could be MySQL.

Three-Layer Architecture

The three-layer architecture a description of how the underlying data models of a database are related. The three layers are

- **External**
  - A view is a subset of the data items in the logical model tailored toward the needs of a specific application or group of users. Often called a virtual table in a relational setting. It is used to control data access and enforce security.
- **conceptual/logical**,
  - data is represented in the form of various database tables.
- **Internal**
  - Specifies how the data are stored or organized physically. (Harddrives, servers, the cloud etc.)



PDBM , page 76

Basic Concept of Relational Model

> In the relational model, a database is represented as a collection of relations. A relation is defined as a set of Tuples that each represent a similar real-world entity such as a product, a supplier, an employee, etc..

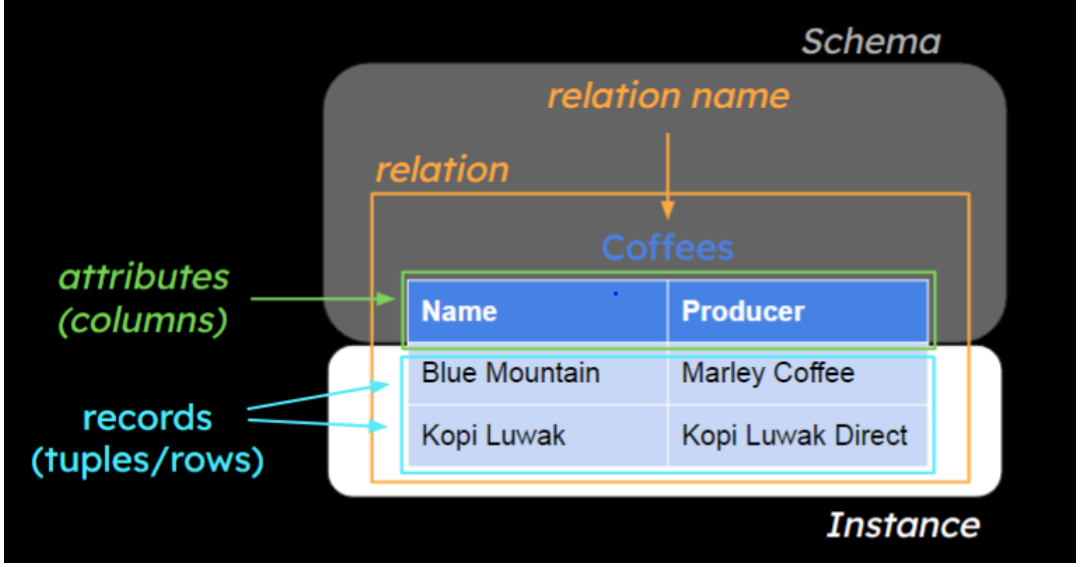
```
[[Principles of database management the practical guide to storing, managing and analyzing big and small data - PDF Room.pdf#page=290&selection=2,0,27,4|PDBM , page 290]]
```

It is recommended to use meaningful names for each relation and its attribute types. Here you can see some examples of relations:

- Student (Studentnr, Name, HomePhone, Address)
- Professor (SSN, Name, HomePhone, OfficePhone, Email)
- Course (CourseNo, CourseName)

Relation

A relation is a Table in the database. It has a *relation name*, *Attributes* (columns). *records* (Tuples/rows)



IDBS - Lecture 1, page 25

Formal definition of a database relations

> A relation $R(A1, A2, A3, \ldots A_n)$ (e.g., SUPPLIER(SUPNR, SUPNAME, …)) can now be formally defined as a set of m tuples $r = t1, t2, t3, \ldots t_m$ whereby each tuple t is an ordered list of n values $t =< v1, v2, v3, \ldots v_n >$ corresponding to a particular entity (e.g., a particular supplier). Each value vi is an element of the corresponding Domain, $dom(A_i)$, or is a special $NULL$ value. A $NULL$ value means that the value is missing, irrelevant, or not applicable.
> Some example tuples for the student, professor, and course relations are as follows:
> Student(100, Michael Johnson, 123 456 789, 532 Seventh Avenue)
> Professor(50, Bart Baesens, NULL, 876 543 210, Bart.Baesens@kuleuven.be)
> Course(10, Principles of Database Management)
> PDBM , page 293

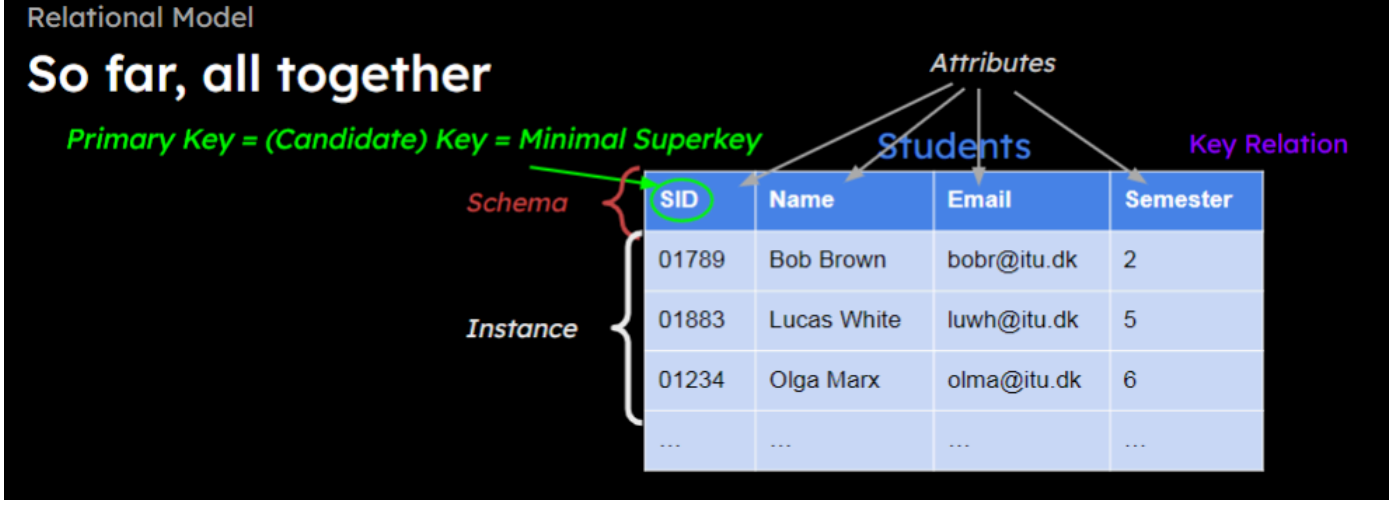
A relation R of degree n on the Domains $dom(A_1), dom(A_2), dom(A_3), \ldots, dom(A_n)$ can also be alternatively defined as a Subset of the Cartesian product of the domains that define each of the attribute types.

Schema

A Schema is the name of the Relation and a list of Attributes. A database schema is a set of all relations names in the database.

Example of a database Schema

```
Students (SID : INT, Name : STRING, Email : STRING, Semester : INT)
```



Domain

Specifies and organizes the range of admissible values for an attribute type.

For example, a domain can consist of

- all integer values between 1 and 9999 and can be used to define the attribute type "SUPNR"
- gender domain, which contains the values "male" and "female"

Each attribute type of a relation is defined using a corresponding domain - and can be used in a relation multiple times.

Attribute Type

An attribute type is a representation of a specific, defined property of an entity type.

Tuple

A tuple is an ordered list of attribute values that each describe an aspect of this entity, such as supplier number, supplier name, supplier address, etc.

Keys

Keys uniquely identify tuples as well as to establish relationships between relations

Superkey

A superkey is a set of attributes that uniquely identifies records. The entire set of attributes of a relation is a superkey

- Superkeys conatin at least one Candidate key.
- Important to define Indexes for storage.
- Used to establish Relationships.
- From all Candidate keys only one can be the Primary Key - the remaining are known as Alternative Keys.

Definition

A superkey is defined as a subset of attribute types of a relation R with the property that no two tuples in any relation state should have the same combination of values for these attribute types. In other words, a superkey specifies a uniqueness constraint in the sense that no two distinct tuples in a state can have the same value for the superkey. Every relation has at least one default superkey – the set of all its attribute types. A superkey can have redundant attribute types.
As an example, for the relation Student, (Studentnr, Name, HomePhone) is a superkey, but note that both Name and

HomePhone are redundant and Studentnr is a superkey as such.
PDBM , page 296

Candidate key

The attribute type(s) of a relation that are unique, such as a unique product number.

- Superkeys contain atleaste one candidate key / all candidate keys are superkeys
- A Relation can have many candidate keys
  As an example, a product relation may have both a unique product number and a unique product name. Each of these keys is called a candidate key. One of them is designated as the primary key of the relation.

In Normalization & Functional Dependencies, the attribute that is **not** present in the right hand side of a Functional Dependency, will be present in the candidate key.

PDBM , page 297

Primary Key

A selected candidate key that identifies tuples in the relation and is used to establish connections to other relations; must be unique within the relation. The primary key identify records in a relation.

- It cannot be `NULL`
- A Relation can have many candidate keys
- From all Candidate keys only one can be the primary key
  This could be a "id" key of a row in the table

Foreign Keys

an attribute type or attribute types of one relation that is/are referring to the primary key of another.
That is - values appearing in attributes of one relation must be key of another relation.

> **Defines the relationship between relations**
> - A key FK in a Relation R is a *foreign key* iff:
> - The attributes in FK matches a primary key PK of a relation S and they are of the same type
> - Any record i in R has a value in FK that either
> - occurs as a value of PK for some record j in S, or
> - is null
> - I.e., FK = PK (domain and values)
> A relation can have several foreign keys

IDBS - Lecture 1, page 38

> A set of attribute types FK in a relation R_1 is a foreign key of R_1 if two conditions are satisfied.
>
> 1. First, the attribute types in FK have the same domains as the Primary Key attribute types PK of a relation R_2.
> 2. a value FK in a tuple t_1 of the current state r_1 either occurs as a value of PK for some tuple t_2 in the current state r_2 or is $NULL$. These conditions for a foreign key specify a so-called referential integrity constraint between two relations R_1 and R_2.
>   PDBM , page 298

Relational Constraints

Constraint	Description
Domain constraint	The value of each attribute type A must be an atomic and single value from the domain dom(A).
Key constraint	Every relation has a key that allows one to uniquely identify its tuples.
Entity integrity constraint	The attribute types that make up the primary key should always satisfy a NOT NULL constraint.
Referential integrity constraint	A foreign key FK has the same domain as the primary key PK attribute type(s) it refers to and occurs as a value of either PK or NULL.

PDBM , page 301