



The Relational Database

UNIT 2



Objetives

- To introduce the most useful concepts about relational databases.



Introduction

Relational DBs are based on a model created by Edgar Codd in 1969.

Since R System (IBM - 1980) until now more than 30 years have passed and it is still the system mostly used.

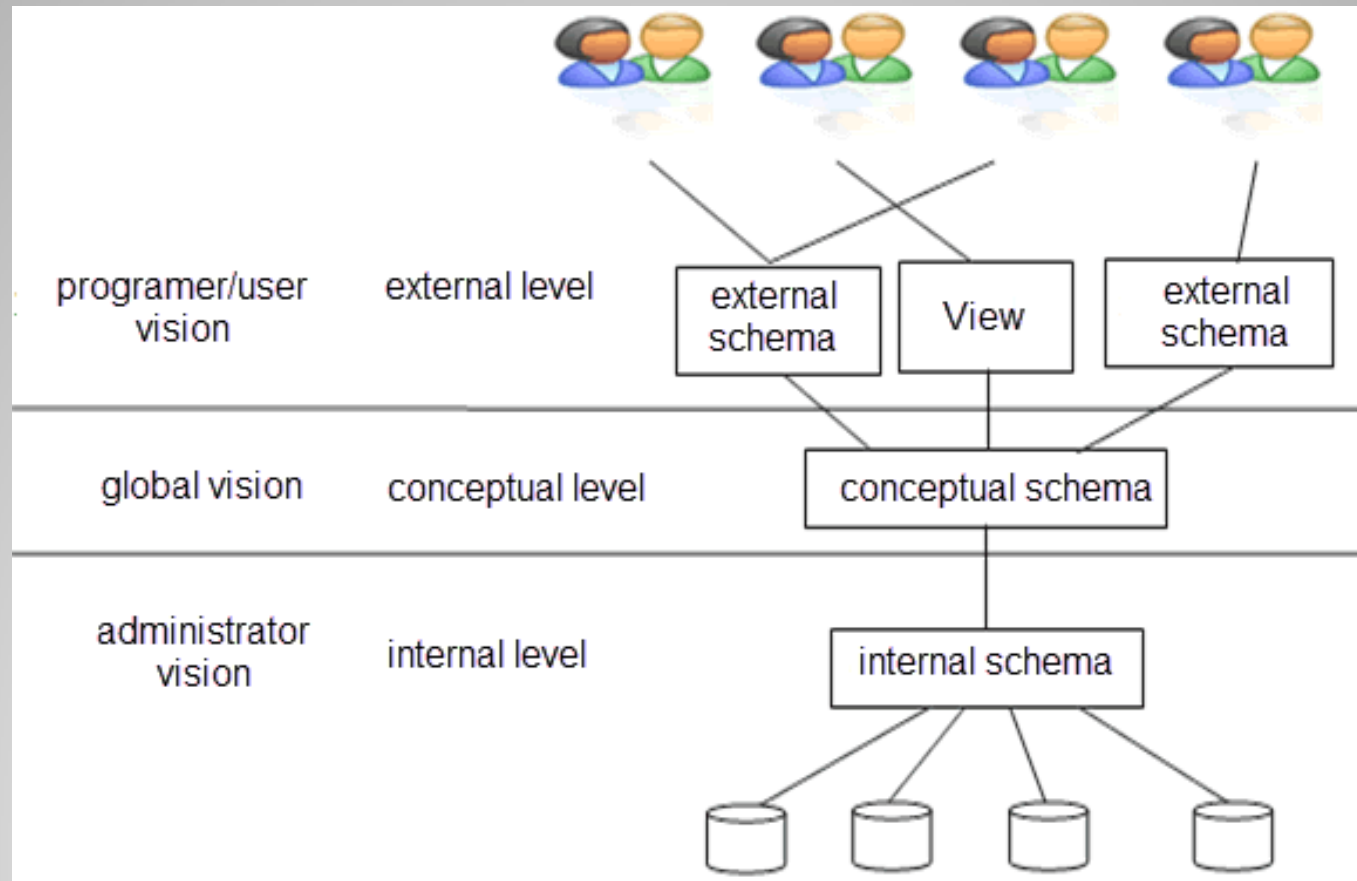


RDBMS aims

- Data independence
 - Physical
 - Logical
- Security and integrity
- Minimum redundancy
- Data are easily accessed.



RDBMS architecture





RDB elements

- Data are organized in *TABLES* with *ROWS* and *COLUMNS*.
- *ROW* → *RECORD*
- *COLUMN* → *FIELD*
- Each table has an *UNIQUE NAME*
- Can contain 0, 1 or more rows.
- Rows are not ordered.
- A row cannot be duplicated.



RDB elements

- The *TABLE SCHEMA* is the list of attributes (columns) the table is formed.
- A *ROW INSTANCE* are the values of the row.
- In a column all values have the same *DATA TYPE*, and those are extracted from the *DOMAIN* (set of permitted values).
- A column name cannot be used for another column in the same table, but it can in another table.



The NULL value

- It is a special value, it represents the lack of value.
- It is not equivalent to 0 or empty string.
- It has its own functionality.

For example:

campo = 0 is correct

campo = NULL not correct → campo IS NULL



Primary key

- A table must have a PRIMARY KEY (clave primaria o principal) .
- The primary key is a field or a set of fields that is used to identify each row of the table.
- Null values are not permitted, neither duplicated values.
- A table cannot have two primary keys, but one key including several columns.



Alternate key

- Sometimes a table can have *ALTERNATE KEYS* (claves alternativas).
- Un alternate key is a field or set of fields that is used to identify each row of the table and it is not the primary key.
- No null values
- No duplicated values.



FOREIGN KEY

- A table can have one or several FOREIGN KEYS (claves ajenas/foráneas/externas).
- It is a column or set of columns that points to a row in a table. The table can be the same table (less frequent) or another table.
- The value included in the foreign key must identify the row in the related table.
- It can have null values and duplicated values.



INTEGRITY RULES

- The DBMS (DataBase Management System) must seek to ensure the data integrity.
- To enforce the business rules associated with the database and prevent the entry of invalid information into tables *INTEGRITY RULES (CONSTRAINTS)* are defined.
- Then before adding information in the DB, the system automatically checks that the information is correct.
- There are different types of constraints.



INTEGRITY RULES

Table DEPT

DEPTNO	DNAME	LOC
20	RESEARCH	DALLAS
30	SALES	CHICAGO

Each value in the DNAME column must be unique

Each row must have a value for the ENAME column

Each value in the DEPTNO column must match a value in the DEPTNO column of the DEPT table

Table EMP

EMPNO	ENAME	... Other Columns ...	SAL	COMM	DEPTNO
6868	MULDER		5500.00		20
7329	SMITH		9000.00		20
7499	ALLEN		7500.00	100.00	30
7521	WARD		5000.00	200.00	30
7566	JONES		2975.00	400.00	30

Each row must have a value for the EMPNO column, and the value must be unique

Each value in the SAL column must be less than 10,000



Integrity rules

- *NULL* rule. Defined on a single column, allows or disallows inserts or updates of rows containing a null (the absence of a value) in that column.
- *UNIQUE* rule. Defined on a column (or set of columns) allows the insert or update of a row only if it contains a unique value in that column (or set of columns).
- *KEY INTEGRITY*. (integridad de claves). Each row in the table can be uniquely identified by the values in the key. So the key must be NOT NULL and UNIQUE.
- *REFERENTIAL INTEGRITY* (Integridad referencial). Guarantees that the values in that key match the values in a key (primary/alternate) in the related table.
- *COMPLEX* integrity checking. Complex rules defined on a column, set of columns or tables.



Integrity rules

How to define integrity rules?

- Including the rule in the data definition
 - NULL clause
 - UNIQUE clause
 - PRIMARY KEY constraint
 - FOREIGN KEY constraint
 - CHECK clause
- Defining triggers
 - For some complex rules



Architecture

- A RDBMS has a 3 levels architecture:
 - *EXTERNAL LEVEL* contains *VIEWS* (vistas) , the user's view of the DB describes a part of the DB that is relevant to a particular user.
 - *CONCEPTUAL LEVEL* describes what data is stored within the whole database and how the data is inter-related. It does not specify how the data is physically stored.
 - *INTERNAL LEVEL* describes how the data is actually stored in the DB and on the computer hardware.



The language

- To define, manage and control the whole data stored in the database, the RDBMS incorporates a specific language based on Codd's rules and relational theory: SQL
- SQL is a high-level language, follows a standard and includes sentences based on relational algebra.



Summary

- Data are organized in related tables.





Summary

- Example:

Piezas				
Codigo	Denominacion	Precio	Fabricante	Codigo_segun_fab
1	Taburete 3 patas	25	Fab1	T123-34
2	Mesa ovalada	1200	Fab1	M234-87
3	Taburete 3 patas	78	Fab2	T2S-0P

Fabricantes		
IdFab	Nombre	Direccion
Fab1	Muebles La Madera	Null
Fab2	Maderas Asociados	Avda. del Sol, nave3
Fab5	Industrias Madereras	Pol. Ind 3 fuentes, nave 23

Schema
Instances
Primary key

Alternate key
Foreign key
Integrity