Unit 2: The relational model

1 - What the relational model is

2 - Main elements

3 - Keys

4 - Indexes

5 - Scheme of a relation

6 - Integrity rules

The relational model

Source: An article by Edgar F. Codd (1970):

- Relation → table (relational theory)
- Relationship → association between tables

Features:

- The database looks like a collection of tables
- Each row within a table represents a record
- The order of the rows is not significant at all, whereas the order of the columns is
- All the rows are different
- Each column is labeled with a name (field)
- A number of operations acting on groups of tuples are defined

Elements of the relational model

- **Tables**: a structure made up of rows and columns that is used to store the data of an entity
- **Fields**: it is the name given to each column of a table, and they represent attributes of the entities. The set of potential, possible values is called the <u>domain</u> name
- **Tuples**: each row of a table is a combination of values, and it is called a tuple. It matches a single record and there cannot be two of them with the same values in their fields
- **Views**: a view can be considered a special case of a table. They are “created” from other tables by using operations between them. They are often used to restrict or facilitate access and for the presentation of data to some of the users

Elements of the relational model

tabla

Cve. cliente	Nombre	Direccion	Ciudad	Estado
1	Alfredo Godinez	Fresnillo #47	Veracruz	Veracruz
2	Gabriela Mora	El crespo #81	Guadalajara	Jalisco
3	Alejandra Avalos	Casa Mata #1	Morelia	Michoacan
4	Jaime Quintero	Miraflores #23	Uruapan	Michoacan
5	Carlos Miranda	Rio Bravo #95	Matamoros	Tamaulipas

registro

campo clave

campo

Advantages

- Data independent
- Application independent
- Storage independent
- Simplification of the data display to the users
- Simplification of the data logical representation
- Allows the optimisation and treatment of redundancy and consistency issues
- Data dictionary representation (also with tables)

Abstraction

The relational model allows the creation of universal languages based on the calculus of relational predicates:

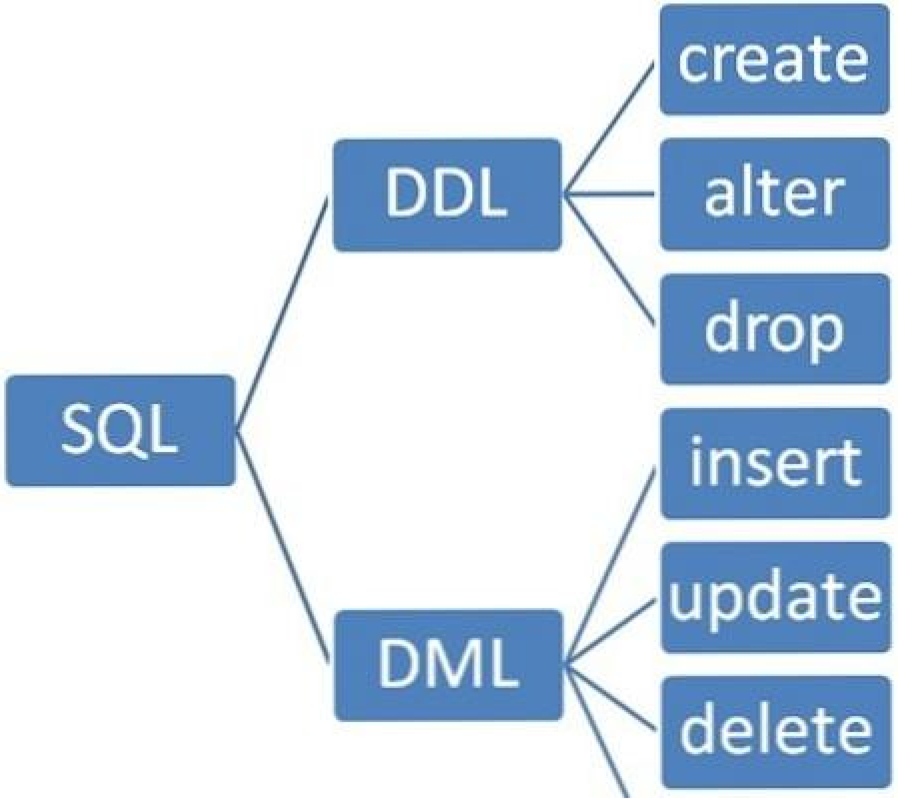
- Relational algebra
- Relational calculus
- SQL (standard → free and commercial implementations)

IMPORTANT: The logical design of a relational database is defined by an **abstract pattern** which is totally independent of the implementation of any DBMS (however, this dependency does exist the other way around)

Referential architecture

Relational database systems have the same structure as other models:

- Physical layer (storage) → definition of storage structures using a specific language (subset of the DDL)
- Logical data structure → also via the DDL
- Information retrieving → by using the DML



Keys

In the relations there are domains (or combinations of them) whose values allow each element to be uniquely identified. These are what we call keys:

- Candidate/unique key:
  - There is at least one for each table
  - Not null
  - Not ambiguous
  - Not repeated
  - It is minimal
- Primary key
  - It is the candidate key that the DB designer chooses as the main one to identify every record
- External / foreign key:
  - There are sets of values that do not form a primary key in their own table, but they actually do in another, external one
  - Therefore, they allow different tables to be related by marching the values of fields in common
  - The may have duplicate or null values

Indexes

An index is a component (from the point of view of data representation) that allows to display the data in a particular order.

They are useful to perform faster and more efficient (certain) query operations and data updates (sort → select / modify)

They have to do with sets of attributes called projections:

- Disadvantage: the elements are already present in the tables → redundant
- Advantage: increase of speed to go through the tables

Indexes can be stored. If the attributes of the table are modified, they are automatically updated. Those fields designated as primary keys are always indexes

Schema of a relation

DB schema → logical design of the database

Scheme of a relation → set of attributes and domains that define the table

R(A) = R(A1:D1, A2:D2, …, An:Dn)

Person(<u>ID</u>: numeric, name: text, birth_date: date)

An <u>instance</u> of the database is the data contained in a relational schema at a given moment (it is like a photography of the data at that instant)

Integrity constraints

They are rules that allow the data to meet certain properties, in order to guarantee or improve its correctness and consistency.

Some of them are already given implicitly in the relational model, e.g.:

- Absence of repeated tuples
- Irrelevance in the order of the tuples
- Relevance in the order of attributes

...whereas some others might be declared or implemented by the designers when creating the database

Integrity constraints

- Key integrity: certain values of a subset of fields cannot be null, and they must differ to each other on different records
- Referential integrity: in a foreign key, there might be no value if there is no match with another record in the referenced table (in other words, there can be null values) → effects on update, delete
- User integrity: those rules defined by the database designer to meet or fulfil the needs on each specific situation

Codd's 12 rules:

https://en.wikipedia.org/wiki/Codd's_12_rules

Normalisation

- This is the method that makes it possible to ensure that a relational design is (more or less) correct
- It consists of a process that reduces data redundancies, obtaining tables with an optimal and efficient structure
- This process is based on the application of rules to meet certain conditions called **normal forms**
- We will learn how to apply this process in the next unit of relational databases design