UT2. Relational Data Bases

“Data Bases”

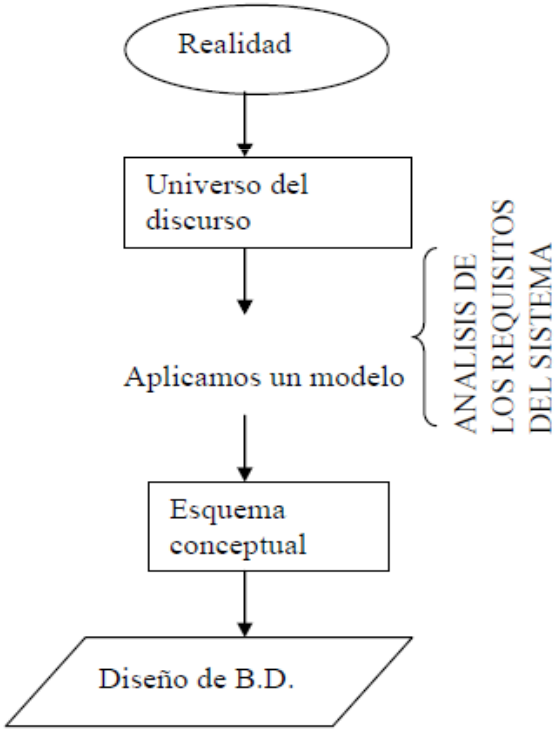
DAM-DAW

Index

- Data model
- Relations, attributes and tuples
- Data type and Domain
- Keys
  - Primary Key
  - Foreign keys
- Referential integrity
- Views
- Security Management
- Indexes
- SQL
  - Data types and domains
  - DDL, DML, DCL
  - TCL
- Logical Design vs. Physical Design

Data model

- We can define data model like de set of concepts, rules and standards that allows to define the data domain (universo del discurso)



Ejemplo: Centro de Estudios Privados

Universo del Discurso 1 (UD1) { Docentes + Especialidades; Alumnos + Expediente

Universo del Discurso 2 (UD2) { Gestión personal, nóminas; Contabilidad, Control recibos

Gestión pedagógica frente a gestión empresarial.

Relations, attributes and tuples

- The relational model is a group of rules set forth by E. F. Codd based on mathematical principles (relational algebra), and it defines how database management systems should function. The basic structures of a relational database (as defined by the relational model) are tables, columns (or fields), rows (or records), and keys.
- **A relation** or table is a collection of logically related information treated as a unit. Tables are organized by rows and columns
  - Order of columns is irrelevant (better primary key at left)
  - There is no order in rows
  - No duplicate rows in a table
  - Other characteristics:
    - NULL value: no information
- Escriba el texto aquí
- each element (column) is termed an **attribute** value
  - An attribute value is an attribute name paired with an element of that attribute's domain
- **Tuple**, equivalent to row in a table

Si una persona (Javier) tiene un role (estudiante) con distintos permisos definidos en una BD, NO se le puede quitar ninguno predefinido en su rol (select, insert), pero sí se le pueden dar permisos adicionales(update, delete)

Domain

- **Domain**
  - a **data domain** refers to all the values which a data element may contain.
  - The rule for determining the domain boundary may be as simple as a data type with an enumerated list of values
  - For example, a database table that has information about people, with one record per person, might have a "gender" column. This gender column might be declared as a string data type, and allowed to have one of two known code values:
    - "M" for male, "F" for female, and NULL for records where gender is unknown or not applicable. The data domain for the gender column is: "M", "F".
  - In a normalized data model, the reference domain is typically specified in a reference table. Following the previous example, a Gender reference table would have exactly two records, one per allowed value—excluding NULL. Reference tables are formally related to other tables in a database by the use of foreign keys.

Keys

- **Key**
  - Each row or tuple has to be identified uniquely by a key, which can be one or several columns.
    - Candidate keys: when there are more than one key. They can´t have null values
    - The **primary key** will be chosen among the candidate keys.
    - **Foreign key**: field **(or collection of fields)** in one table that uniquely identifies a row of another table or the same table
    - Since the purpose of the foreign key is to identify a particular row of referenced table, it is generally required that the foreign key is equal to the candidate key in some row of the primary table, or else have no value (the NULL value.). This rule is called a referential integrity constraint between the two tables

DNI	NOMBRE ALUMNO	TUTOR	DELEGADO
D1	JAIME	P1	D2
D2	PEPE	P2	D2
D3	LUIS		D2

LA COLUMNA DELEGADO ES UN FOREIGN KEY DE LA MISMA TABLA DE ALUMNOS (PORQUE EL DELEGADO ES UN ALUMNO)

Referential Integrity

- ***Foreign keys establish relationships between tables.***

- **A *referential integrity constraint* is the rule** that the **non null** values of a foreign key are valid only if they also appear as values of a parent key. The table that contains the parent key is called the *parent table* of the referential constraint, and the table that contains the foreign key is a *dependent* of that table.
  - On delete Cascade
  - On update set null

DEPARTAMENTOS

Clave Primaria		
NumDepto	NombreDepto	Presupuesto
D1	Marketing	1.000.000
D2	Desarrollo	1.200.000
D3	Investigación	500.000

EMPLEADOS

Clave Primaria				
NumEmpleado	Apellido	Salario	Comisión	Clave Ajena
E1	López	4.000.000	24%	D2
E2	Fernández	3.000	1%	D3
E3	Martínez	2.000.000	3.5%	D2

Views

- View
  - Remember external level
  - Views can represent a subset of the data contained in a table. Consequently, a view can limit the degree of exposure of the underlying tables to the outer world: a given user may have permission to query the view, while denied access to the rest of the base table.
  - A **view** is the result set of a *stored* query on the data
  - Example: in an enterprise a specific user only needs to have access to the employee's name and de department's name (which is stored in to different tables – employees and Departments)
  - **CREATE VIEW AS …..**

Apellido	Nombre Departamento
López	Marketing
Fernández	Desarrollo
Martínez	Investigación
Sánchez	Desarrollo

Security management

- User
  - To access a database we need an user with a password. It's a need both for users and software applications
  - CREATE USER
- ~~Roll~~ Role
  - Roles grant and deny permissions to groups of users
  - A roll can include other roles a User can have more than 1 role.
  - One user can have several roles
- Permission
  - What a user or application can or cannot do in a database
  - It can refer to data contents or database actions
  - GRANT

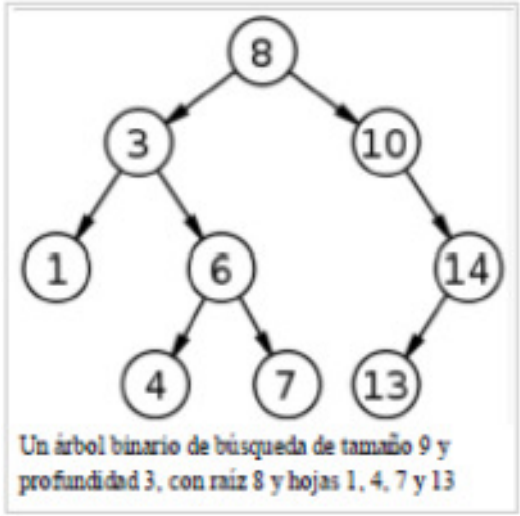
Si una persona (Javier) tiene un role (estudiante) con distintos permisos definidos en una BD, NO se le puede quitar ninguno predefinido en su rol (select, insert); pero sí se le pueden dar permisos adicionales(update, delete) a través de GRANT

Exercise

- Optional
  - Design a security subsystem which allows to control which permissions can be assigned to an specific user
    - The data of the user has to be stored
    - Role management (creation as well as assignment to an user)
    - Permissions can be assigned to roles
    - Do you think that the system should allow to assign individual permissions to an user?

Indexes

- An **index** in a database operates like the index tab on a file folder. It points out one identifying column, such as a customer's name, that makes it easier and quicker to find the information you want.
- Rule of thumb: 10%- 15%.
- Index can have one or more columns
- It requires space in disk
- Many indexes affect the update, delete and insert operations.
- Index can allow duplicate values or not
- Unique indexes prevent from duplicated values in a column (columns)
- In a relational database
  - An unique index is created with the fields of the define primary key
  - Foreign keys are suitable for defining an index (not always)
  - B-tree structure
- CREATE INDEX



Un árbol binario de búsqueda de tamaño 9 y profundidad 3, con raíz 8 y hojas 1, 4, 7 y 13

Exercise

- Example
  - Student table: Ies, DNI , N.Matrícula, Nombre,edad
  - Teacher table: Ies, DNI, nombre
  - One student only have a Tutor (which is a teacher)
    - Define candidate keys
    - Choose primary key
    - How to resolve uniqueness constraint of the key not chosen as primary key?
    - What other fields of student table are suitable to define an index?
      - Will it allow duplicates?

Structured Query Language (SQL) /ˈsiːkwəl/ "sequel"

- SQL is a standard language for storing, manipulating and retrieving data in databases.
- Originally based upon relational algebra and relational calculus
- SQL consists of many types of statements which may be informally classed as sublanguages, commonly:
  - data query language “Select”
  - data definition language (DDL) CREATE, DROP, ALTER,……
  - data manipulation language (DML) INSERT, UPDATE, DELETE..
  - data control language (DCL) CREATE USER, GRANT..
- BE CAREFUL!: SQL implementations are incompatible between vendors and do not necessarily completely follow standards. Migration from one database to another can be more difficult if we use particularities of one vendor
- Origins and evolution

SQL languages

- **DDL** (**Data Definition Language**)
  - CREATE
  - ALTER
  - DROP
- **DCL** (**Data Control Language**) :
  - Manage users: CREATE …
  - Manage permissions: GRANT … .
- **DML** (**Data Manipulation Language**) :
  - Query the database: SELECT
  - UPDATE table contents
  - DELETE rows in tables

- THEY ARE NOT CASE SENSITIVE.
  - Usually reserved words in capital letters

SQL: DDL (Data Types)

- https://www.w3resource.com/sql/data-type.php
- **Character String types**
  - Fixed length: CHAR(long) / CHARACTER(long)
  - Variable Length: VARCHAR(long)
  - **Character large object: CBLOB**
- **Numeric Types**
  - Integer, SmallInt , Bigint ,…
- **Decimal Numbers**
  - NUMERIC(precisión, escala) / DECIMAL(precisión, escala) / DEC(precisión,escala)
- **Floating-point numbers (**The term *floating point* is derived from the fact that there is no fixed number of digits before and after the decimal point; that is, the decimal point can float)
  - FLOAT (precision), REAL, DOBLE PRECISION
- **Boolean types**
  - Boolean: Stores truth values - either TRUE or FALSE
- **Date/time Types**
  - DATE, TIME …

- … Consult your database to check data types allowed.
  - https://www.w3schools.com/sql/sql_datatypes.asp

DDL: Create Domain

- Cada dominio E/R se representa mediante una sentencia CREATE DOMAIN en SQL-2:

```
CREATE DOMAIN e_civil AS CHAR(1)
        CHECK (VALUE IN ('S', 'C', 'V', 'D'))
```

NOTE: No domains in Microsoft Access

DDL: CREATE TABLE

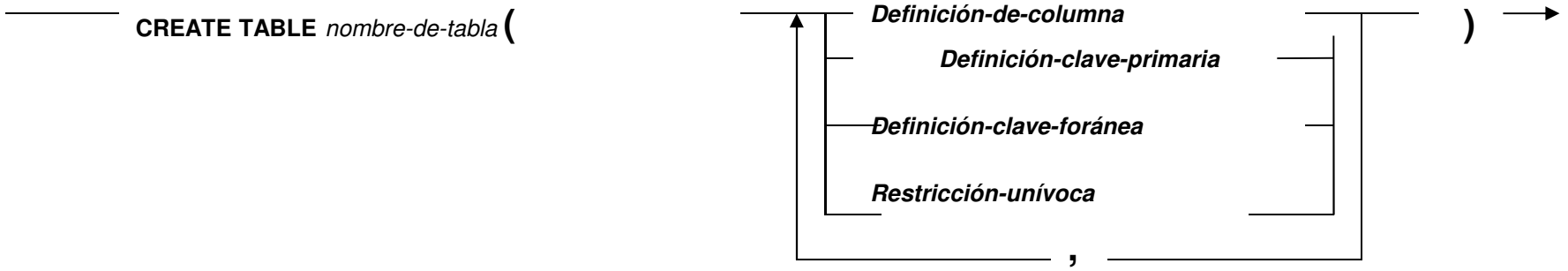
- **CREATE TABLE tabla (**
  campo1 typo(length),
  campo2 type (length),
  ...**);**

Constraints:

- **Primary Key:** PRIMARY KEY(campo1, campo2, ...)
- **Uniqueness:** UNIQUE(campo1, campo2,...)
- **Mandatory:** NOT NULL
- **Referential Integrity:**

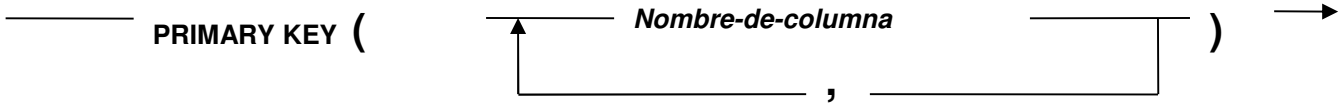
  FOREING KEY(campo1, campo2, ...) REFERENCES tabla [(campoE1,campoE2,..)]

.

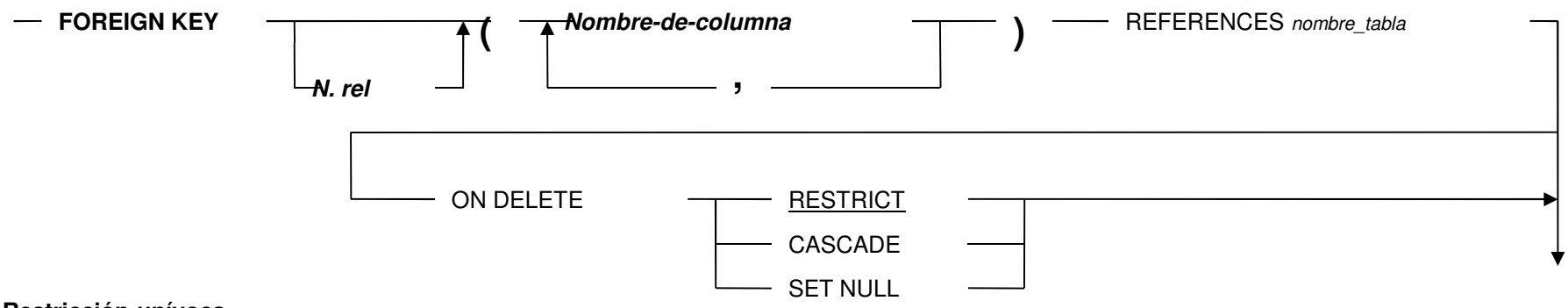


Definición-de-columna

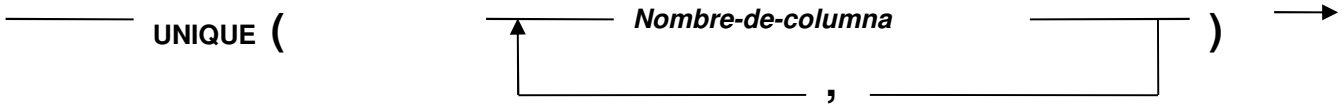
Definición-clave-primaria



Definición-clave-foránea



Restricción-unívoca



DDL: Create Table

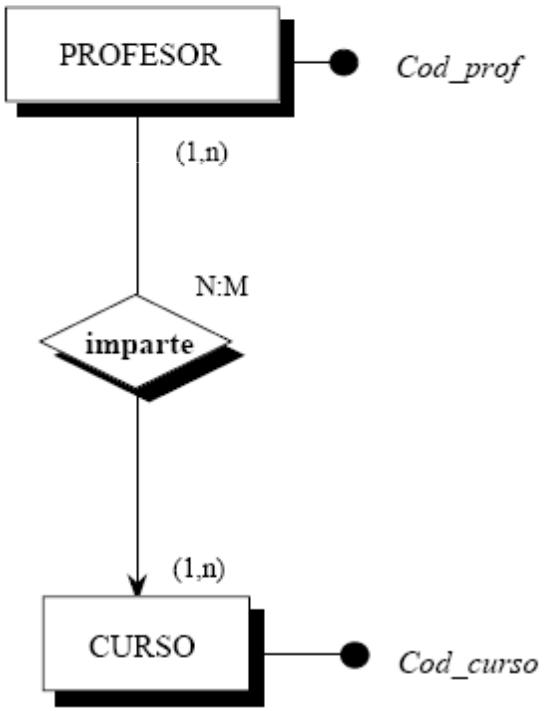
ME/R

MR – SQL

```
CREATE TABLE Profesor (
                Cod_Profesor Códigos,
                Nombre Nombres,
                DNI DNIS, NOT NULL
                Dirección Lugares,
                Teléfono Nos_Teléfono,
                Materia Materias,
                PRIMARY KEY (Cod_Profesor),
                UNIQUE (DNI));
```

DDL: Create Table

Modelo E/R



Modelo Relacional

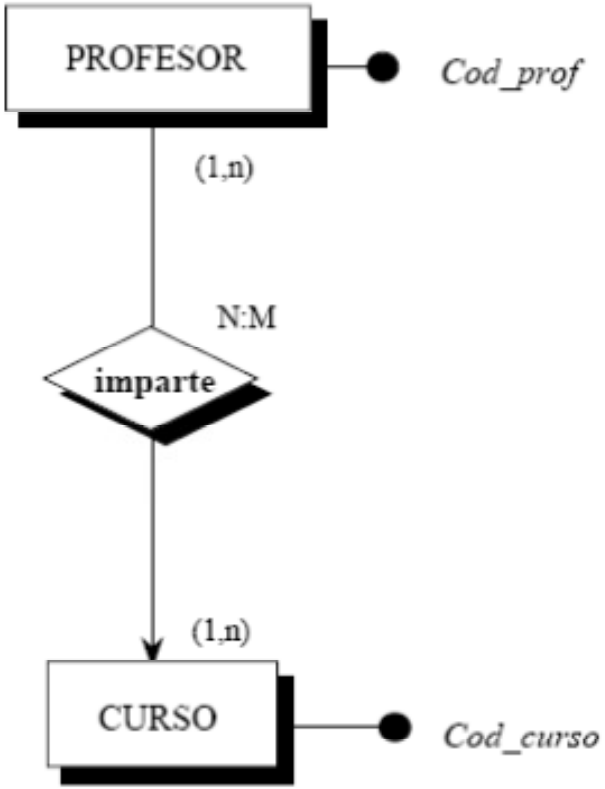
PROFESOR (*Cod_prof*, ...)

CURSO (*Cód_curso*, ...)

IMPARTE (*Cód_curso, Cod_prof*, ...)

DDL: Create Table

Modelo E/R



SQL

```
CREATE TABLE Imparte
(Cod_Profesor Codigos_P,
Cod_Curso Codigos_C,
…. ,
  PRIMARY KEY (Cod_Profesor, Cod_Curso),
  FOREIGN KEY (Cod_Profesor) REFERENCES Profesor
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (Cod_Curso) REFERENCES Curso
    ON DELETE CASCADE
    ON UPDATE CASCADE)
```

DDL: Alter Table

- **ALTER TABLE**
  - ALTER TABLE profesor ADD (apellidos VARCHAR(100));
  - ALTER TABLE profesor MODIFY (nombre VARCHAR(50));

DDL: Drop Table

- **DROP TABLE**
  - DROP TABLE profesor;
  - DROP TABLE imparte;
  - DROP TABLE curso;
- Which should be the order of the previous sentences?

Logical Design vs. Physical Design

- Logical Design
  - Normalization
    - the process of restructuring a relational database in accordance with a series of so-called normal forms in order to **reduce data redundancy** and **improve data integrity**.
    - It was first proposed by Edgar F. Codd as an integral part of his relational model.
- Physical design
  - **Denormalization** is a strategy used on a previously-normalized database to increase performance
  - https://searchdatamanagement.techtarget.com/definition/denormalization

Exercise

- Look for DCL syntax for
  - Creating users
  - Assign permissions

References

- Tutorials and exercises: http://www.w3schools.com/
- Quick reference: http://sql.1keydata.com/es
- Tutorial : http://www.aulaclic.es/sql
- MySQL:
  - http://www.mysql.com/
  - http://dev.mysql.com/
- Data types
  - https://www.w3resource.com/sql/data-type.php