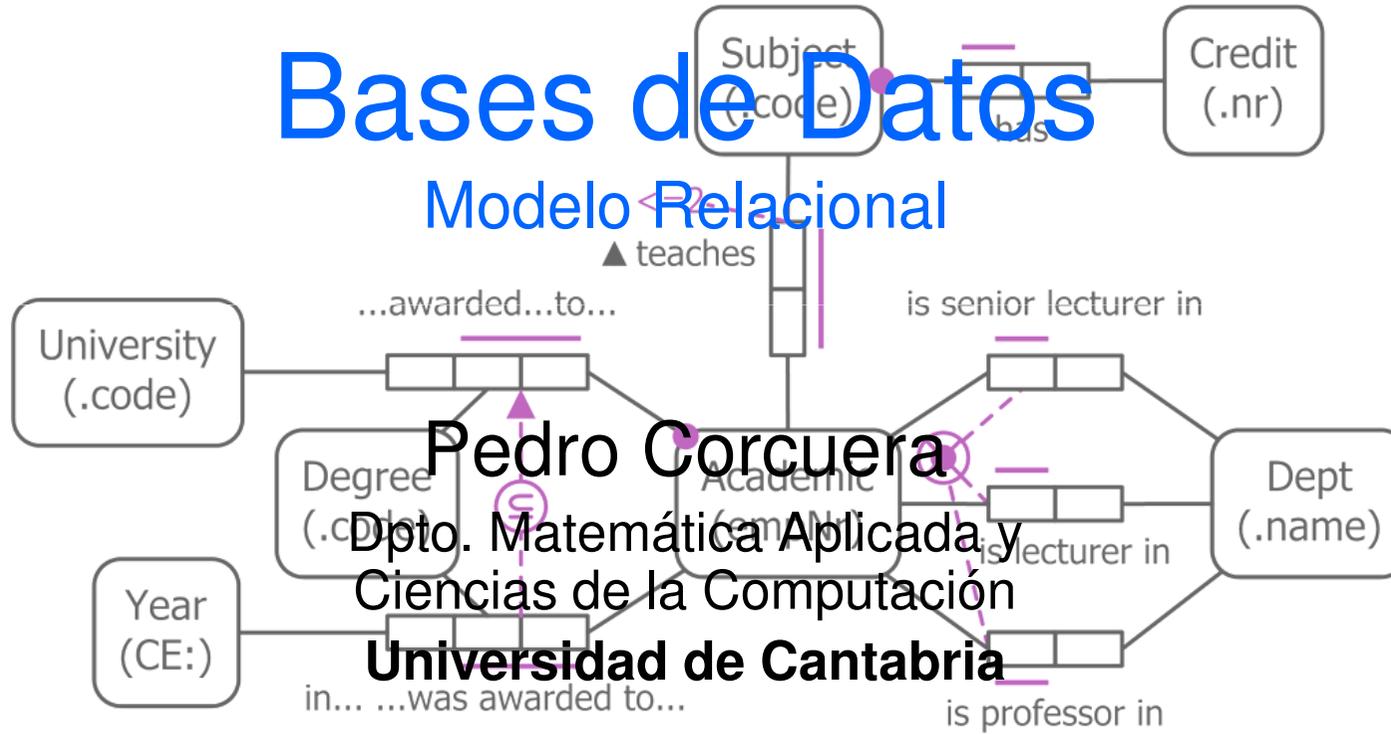


Bases de Datos

Modelo Relacional



Pedro Corcuera
 Dpto. Matemática Aplicada y
 Ciencias de la Computación
 Universidad de Cantabria

corcuerp@unican.es



Objetivos

-
- Presentar los conceptos del método relacional para modelar los datos



Índice

- Modelo Relacional
- Reglas de Codd

- Redes (1964)
 - Honeywell, IMAGE (HP), VAX DBMS, etc.
- Jerárquico (1965)
 - IMS IBM
- Relacional (1970), comercialmente (1982)
- Orientado a Objetos (1990's)
 - Persistencia en PL, OODBMS (O2, Orion, Iris)
- Objeto Relacional (mediados de los 90's)
 - Informix, Oracle-10i, SQLserver, DB2

- El modelo Relacional fue propuesto por Edgar F. Codd de IBM en 1970 en el artículo: "A Relational Model for Large Shared Data Banks", Communications of the ACM, June 1970.
- Por este trabajo, Ted Codd obtuvo el premio Turing en 1981 (ACM Turing Award)
- Primer sistema comercial: 1981-82

- Algunos sistemas comerciales actuales:

The logo for Microsoft SQL Server, featuring the word "Microsoft" in a small font above "SQL Server" in a larger, bold font.The logo for IBM DB2, with "IBM" in its characteristic eight-bar font and "DB2" in a green box.The logo for Oracle, with the word "ORACLE" in a red, stylized font.The logo for Teradata, with "Teradata" in orange and "a division of NCR" in a smaller font below it.The logo for Sybase, with the word "SYBASE" in a serif font.

- Sistemas open source actuales:

The logo for MySQL, featuring a blue fish icon and the text "MySQL" in a blue font.The logo for PostgreSQL, featuring a blue elephant icon and the text "PostgreSQL" in a blue font.

- El modelo relacional es muy usado en los DBMS
 - Obtiene su nombre del concepto matemático de **relación matemática**, donde cada entidad es representada por medio de una relación y que puede verse como un conjunto de ‘valores’ diferentes de una entidad dada
- Se caracteriza por:
 - ser sencillo y uniforme (colección de tablas y lenguajes declarativos)
 - sólida fundamentación teórica: rigor matemático del modelo
 - se independiza del almacenamiento físico y de las aplicaciones.

- Una base de datos relacional representa los datos mediante **tablas** bidimensionales
 - La tabla por sí misma, y no los datos almacenados en ella, es lo que se llama **relación**. Las relaciones (tablas) poseen un nombre que refleja su contenido
 - Los datos almacenados en una tabla son llamados ocurrencias, instancias o **tuplas** de la relación
 - Las tablas están formadas por una colección de **atributos**:
 - Describe a la relación, con referencia a los datos que almacena.
 - Forman las columnas de una relación.
 - Tiene un nombre que se relaciona con el tipo de datos que el atributo almacena

- conjunto de valores permitido para un atributo (nominado, finito y homogéneo de valores atómicos)
- se identifica por un nombre
- Cada dominio puede definirse de dos maneras:
 - Extensión (dando sus posibles valores):
 - días de la semana = {lunes, martes, miércoles, ... sábado, domingo}
 - Intensión (mediante un tipo de datos):
 - peso = decimal
 - A veces se asocia al dominio su unidad de medida (kilos, metros, etc.) y/o ciertas restricciones (como un rango de valores)

- Un **atributo** es la interpretación de un determinado dominio en una relación, es decir el “papel” que juega en la misma. se identifica por un nombre
- Atributo y dominio pueden llamarse igual, pero ...
 - Un atributo está siempre asociado a una relación, mientras que un dominio tiene existencia propia con independencia de las relaciones
 - Un atributo representa una propiedad de una relación
 - Un atributo toma valores de un dominio
 - Varios atributos distintos (de la misma o de diferentes relaciones) pueden tomar sus valores del mismo dominio

- Una relación (matemáticamente) es un subconjunto del producto cartesiano de la lista de dominios $\{D_i\}$
- Una relación consta de los siguientes elementos:
 - Nombre de la relación
 - Cabecera: conjunto de n pares atributo-dominio
 - Cuerpo: Conjunto de m tuplas
 - Esquema: constituido por el nombre de la relación y la cabecera
 - Estado: constituido por el esquema y cuerpo
 - La **cardinalidad** de una relación es el número de tuplas que hay en un esquema. El **grado** el número de atributos

- RELACIÓN
 - Es la estructura básica del modelo relacional. Se representa mediante una **tabla**.
 - DOMINIO
 - Es el conjunto válido de valores que toma un **atributo**. Existen con independencia de cualquier otro elemento.
 - ATRIBUTO
 - Representa las propiedades de la relación. Se representa mediante una **columna**.
 - TUPLA
 - Es una ocurrencia de la relación. Se representa mediante una **fila**
-



-
- Clave Candidata (Candidate Key): conjunto de atributos que identifican unívoca y mínimamente cada tupla de la relación
 - Una clave debe ser única
 - Pueden estar formadas por un solo atributo o por la concatenación de varios atributos
 - Para poder definir las es necesario conocer las reglas bajo las cuales operan los datos en la relación

- Una relación puede poseer más de una llave.
 - **Super Llave:** Atributo o conjunto de atributos que identifica de manera única una relación
 - **Llaves candidatas:** llaves que posee una relación, número mínimo de atributos
 - **Llave primaria:** llave escogida de entre las llaves candidatas para trabajar con ella en la relación. Se selecciona buscando que posea el menor número de atributos, y que no pueda tener valores nulos
 - **Llaves secundarias:** llaves candidatas que no fueron seleccionadas como llave primaria.

- **Nulo**: es un valor, no es lo mismo que cero o blanco. Significa que el valor particular de ese atributo se desconoce en ese momento
- **Llave primaria** (primary key): no permite nulos en algún atributo de la misma (regla de integridad de entidad)
- **Llave foránea** (foreign key): atributo simple o compuesto, en una relación, que está definido en el mismo dominio que la llave primaria de otra relación

- **Llaves primarias y foráneas:** forman las asociaciones entre las relaciones de la base de datos.
- **Base de datos 'completamente' relacional:** no tiene conexiones físicas (apuntadores) entre relaciones, solo las conexiones establecidas por los valores duplicados.
- **Regla de Integridad Referencial:** Una llave foránea referencia a una primaria, por lo que un valor de ésta existe para cada valor de la llave foránea

- Regla de Integridad de la llave (Key Constraint):
 - Cada tabla debe tener una llave que identifique de manera única cada tupla.
- Regla de Integridad de la Entidad (Entity Integrity Constraint):
 - La llave Primaria no puede ser Nula
- Regla de Integridad de Referencia (Referential Integrity Constraint):
 - Una tupla de una tabla que referencía otra tabla DEBE referirse a una tupla existente en esa tabla

Reglas de Codd para considerar un DBMS relacional

- Toda la información debe estar almacenada en tablas.
- Toda la información debe poder ser consultada con solo utilizar un nombre de tabla, un nombre de columna y un valor de llave primaria.
- Los valores nulos deben ser manejados de manera sistemática.
- Debe soportar un catálogo (término relacional para un diccionario de datos) en línea y dinámico.
- Debe tener lenguajes que manejen la definición de datos, la definición de vistas, la manipulación de datos, las reglas de integridad, y transacciones.
- Debe ser capaz de actualizar cualquier 'vista' de datos teóricamente actualizable, que se pueda crear.

Reglas de Codd para considerar un DBMS relacional

- Los comandos de Altas, Cambios y Bajas deben trabajar en tablas completas
- Los cambios en los métodos de almacenamiento físico no deben afectar la ejecución de las aplicaciones.
- Los cambios al esquema no deben afectar a las aplicaciones que no utilicen la porción de la BD afectada por dichos cambios.
- Debe soportar la definición de reglas de integridad. Las reglas deberán estar definidas en el catálogo. El SMDBD deberá comprobar que estas reglas se cumplan.
- Si la BD es distribuída, deberá aparecer a los usuarios como si estuviera centralizada.
- NO debe haber forma de violar las reglas de integridad especificadas en el catálogo.