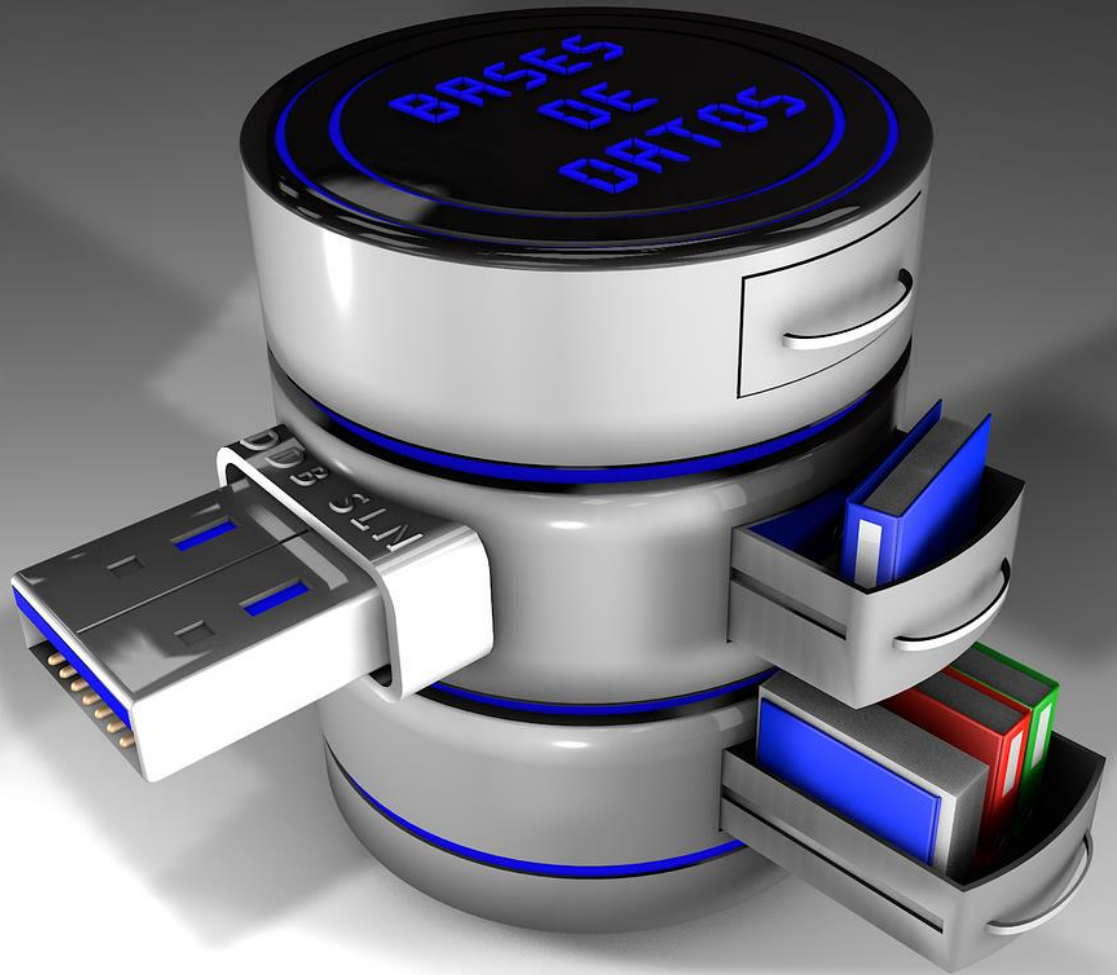


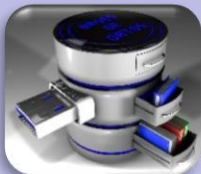
UNITAT DIDÀCTICA 4

Model Relacional



Índex

1. Introducció
2. Objectius
3. Elements
4. Claus
5. Restriccions

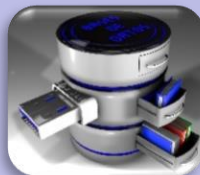
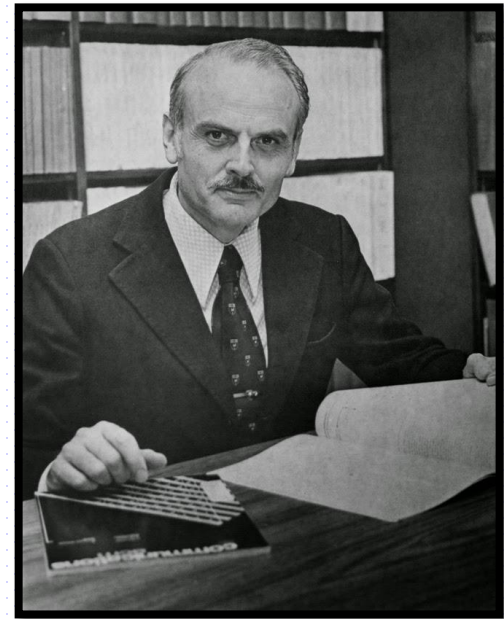


Introducció

Edgar Frank Codd va definir les bases de el model relacional a la fi dels 60.

En 1970 publica el document "*A Relational Model of data for large Shared data Banks*" ("*Un model relacional de dades per a grans bancs de dades compartits*").

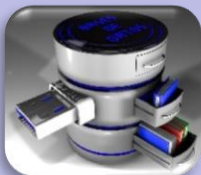
Actualment es considera que aquest és un dels documents més influents de tota la història de la informàtica. Ho és perquè en ell es van definir les bases de l'anomenat Model Relacional de Bases de Dades.



Introducció

El que intentava era evitar que els usuaris de les bases de dades, hagueren d'aprendre els secrets interns de el sistema. Pretenia que els usuaris treballessin de forma senzilla i independentment del funcionament físic de la base de dades en si.

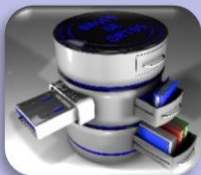
El seu enfocament va ser revolucionari al evitar conceptes del món de la computació en el seu model.



Introducció

Encara que treballava per IBM, aquesta empresa no va rebre molt bé les seues teories. De fet, IBM va continuar treballant en el seu sistema gestor de bases de dades en xarxa IMS. Van ser altres empreses (especialment Oracle) les que van implementar les seues teories.

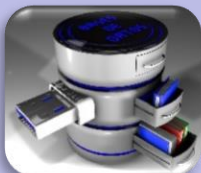
Pocs anys després, el model es va començar a utilitzar cada vegada més fins, finalment, ser el model de bases de dades més popular. Hui en dia pràcticament totes les bases de dades segueixen aquest model, encara que en aquests anys han aparegut rivals cada vegada més forts en les anomenades bases de dades NoSQL, que han demostrat un gran eficàcia en bases de dades que necessiten una enorme quantitat d'instruccions de modificació per minut.



Objectius

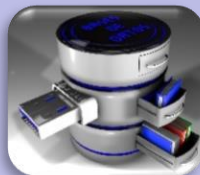
Els objectius que perseguia amb el seu model eren:

- ✓ **Independència física.** La forma d'emmagatzemar les dades, no ha d'influir en la seua manipulació lògica. Si la forma d'emmagatzemar les dades canvia, el usuari no s'han de veure afectats en el seu treball. Això permet que els usuaris es concentren en què volen consultar a la base de dades, en lloc de com consultar-lo.



Objectius

- ✓ **Independència lògica.** Les aplicacions que utilitzen la base de dades, no han de ser modificades perquè es modifiquen elements de la base de dades. És a dir, afegir, esborrar i suprimir dades, no influeix en les vistes dels usuaris.
- ✓ **Flexibilitat:** la base de dades ofereix fàcilment diferents vistes en funció dels usuaris i aplicacions.
- ✓ **Uniformitat.** Les estructures lògiques sempre tenen una única forma conceptual (les taules).
- ✓ **Senzillesa.** Facilitat de maneig pel que fa a sistemes anteriors.

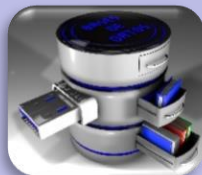


Elements

Taula

L'element fonamental és el que es coneix com a **relació**, encara que més habitualment se l'anomena **taula**.

Codd va definir les relacions utilitzant un llenguatge matemàtic, però es poden associar a la idea taula (de files i columnes) ja que és més fàcil d'entendre.

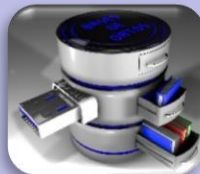


Elements

Taula

Les taules consten de:

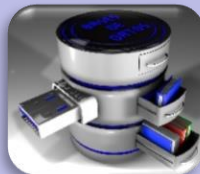
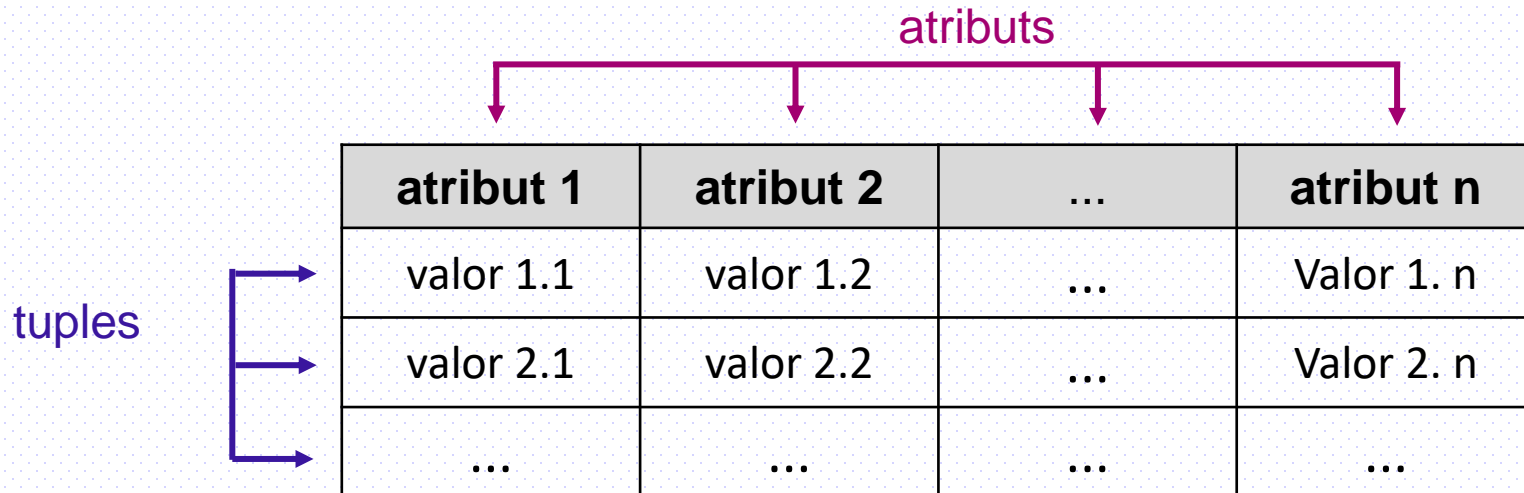
- atributs. Cadascuna de les propietats de les dades que s'emmagatzemen en la relació (nom,dni, ...)
- tuples. Cada element o exemplar de la relació. Per exemple si es té una relació que emmagatzema persones, una tupla representaria cadascuna de les persones emmagatzemades (Anna, Pere, Carme, ...)



Elements

Atributs. Tuples

Ja que una relació es representa com una taula, es pot dir que les columnes de les taules són els atributs i les files, les tuples.

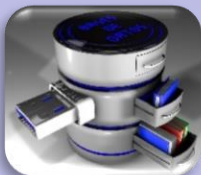


Elements

Atributs. Tuples

Suposem una taula en la qual es guardarà informació sobre professors

DNI	Nom	Cognom1	Cognom2
20556987P	Maria	García	Jiménez
42336998L	Lucas	Ruíz	Ruíz
33569004T	Carla	Esteban	Pérez

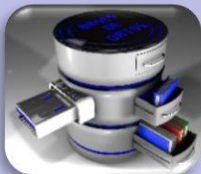


Elements

Domini

Un domini conté tots els possibles valors que pot prendre un determinat atribut. Dos atributs diferents poden tenir el mateix domini.

És un conjunt finit de valors del mateix tipus. Als dominis se'ls assigna un nom i així podem referir-nos a aquest nom en més d'un atribut.

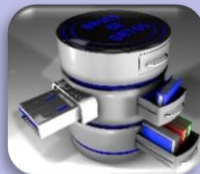


Elements

Domini

Es confon la idea entre tipus de dades i domini. La veritat és que són conceptes semblants, però no iguals. La diferència és que el domini imposa més restriccions que els tipus de dades.

Un tipus de dades pot ser integer, Però un domini pot ser valors entre [0-110] si parlem, per exemple, d'un domini per indicar l'edat.

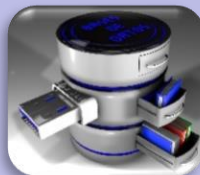


Elements

Propietats de les taules

Les taules han de complir una sèrie de propietats:

- Cada taula té un nom diferent.
- Cada atribut de la taula pren un sol valor en cada tupla.
- Els noms dels atributs EN UNA MATEIXA TAULA són diferents, encara que poden repetir-se en taules diferents.
- Cada tupla és única. No hi ha tuples duplicades.
- L'ordre dels atributs, no importa.
- L'ordre de les tuples, no importa.



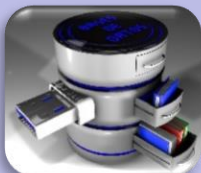
Elements

Claus.

Definició

Una **clau candidata** és un atribut que pot identificar inequívocament a una tupla d'una taula. Aquesta clau ha de complir dues propietats: que no puga repetir el valor i que no puga estar en blanc.

Tota taula en el model relacional ha de tindre al menys una clau candidata, aquesta passarà a convertir-se en **clau principal**.



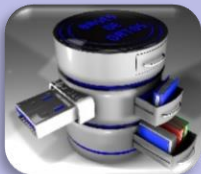
Elements

Claus.

Definició

Una **clau principal** és una clau candidata que es tria com a identificador de la taula. Ha de complir, per tant, les mateixes propietats que les claus candidates.

Tota taula en el model relacional ha de tindre al menys una clau candidata, aquesta passarà a convertir-se en **clau principal**.



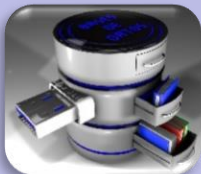
Elements

Claus.

Si hi ha diverses claus candidates, es tria com a principal la que identifique millor a cada tupla en el context de la base de dades.

Per exemple, tenim una taula en la qual es va a emmagatzemar clients. Es disposen dels atributs *DNI* i *código_client*, les dos **claus candidates**. Es triaria com a **clau principal** el *código_client*, ja que és l'atribut específic per al client.

L'atribut DNI passaria a ser **clau alternativa**.



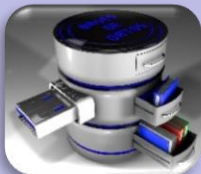
Elements

Claus.

Definició

Una **clau alternativa** és una clau candidata que NO es tria com a clau principal. Ha de complir, per tant, les mateixes propietats que les claus candidates.

Les taules poden tindre o no claus alternatives



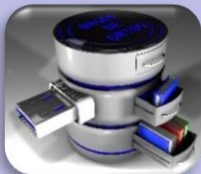
Elements

Claus.

Definició

Una **clau aliena** és un atribut que està en una taula, però que serveix per identificar a un altre element d'una altra taula.

Les taules poden tindre o no claus alienes



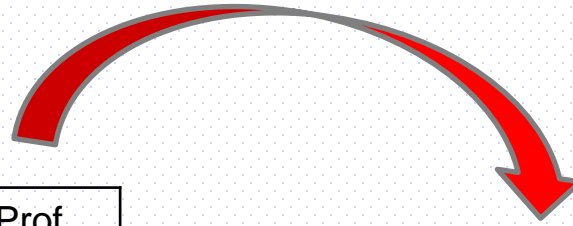
Elements

taula Alumne

expedient	nom	Cod_Prof
175	Andrés	2
102	Sofia	3
305	Pau	2
419	Julia	1
533	Inés	1

**clau
Principal**

**clau
aliena**



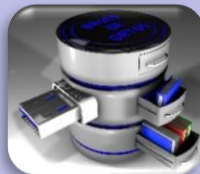
taula Professor

nom	codi	DNI
àngel	1	20145668K
Martina	2	70224896M
Carmen	3	30898557F

**clau
inici**

**clau
alternativa**

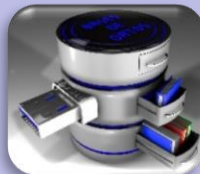
l'atribut Cod_Prof en la segona taula, és una **clau aliena**, ja que serveix per identificar elements d'una altra taula



Elements

Restriccions

- ✓ **clau principal** (primary key). Marca un o més atributs com a identificadors de la taula.
- ✓ **unicitat** (unique). Impedeix que els valors dels atributs marcats d'aquesta forma puguin repetir-se. Aquesta restricció s'ha d'indicar en totes les claus alternatives.
- ✓ **obligatorietat** (not null). Prohibeix que l'atribut marcat d'aquesta manera quedi buit, és a dir, que pugui contenir un valor nul. Ha de marcar-se en les claus alternatives.



Elements

Restriccions

- ✓ **integritat referencial** (foreign key). Serveix per indicar una clau aliena. Només poden contenir valors que coincideixen amb la clau principal de la taula que relacionen.
- ✓ **check**. És una restricció que imposa una condició lògica que haurà de complir la columna. Per exemple, podríem restringir la columna anomenada salari perquè sempre accepti valors majors de 1000; no es permetria indicar salaris menors de 1000 en aquesta columna. De vegades les regles impliquen a diverses columnes, com ara que la data d'inici sigui major que la data final.

