Models de dades

PID_00271444

Blas Torregrosa García

Temps mínim de dedicació recomanat: 2 hores

Universitat Oberta de Catalunya

Blas Torregrosa García

Enginyer en Informàtica i màster universitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions (MISTIC) per la Universitat Oberta de Catalunya (UOC). Especialitzat en ciberseguretat. Professor col·laborador del màster de Ciència de Dades de la UOC i professor associat a la Universitat de Valladolid (UVA).

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Ferran Prats Carrasco (2020)

Primera edició: febrer 2020

Av. Tibidabo, 39-43, 08035 Barcelona
Realització editorial: FUOC

Índex

Introducció

Els **models de dades** són una manera d'estructurar i organitzar les dades perquè es puguin utilitzar més fàcilment. Introduirem diferents estructures de dades, les operacions amb dades i les diferents restriccions que podem imposar a les dades. Així mateix, veurem els diferents nivells de modelatge de dades i, finalment, els tipus de models de dades.

1. Definint el model de dades

Un **model de dades** determina la manera com s'organitzen i estructuren les dades. Els models de dades són el nucli de l'emmagatzematge, l'anàlisi i el processament dels sistemes que gestionen les dades.

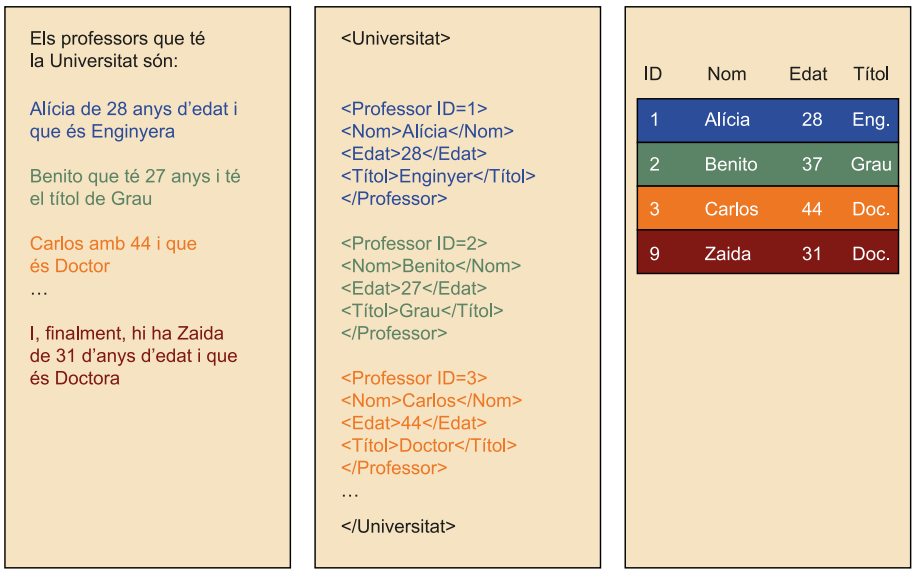
Els principals tipus d'estructures són dades estructurades, semiestructurades i no estructurades. També examinarem diferents tècniques de modelatge aplicades a aquests tipus de dades.

1.1. Estructures dels models de dades

Els models de dades tracten i descriuen una gran varietat de característiques de les dades. Pel seu origen, hi ha tres tipus principals de dades:

- Dades estructurades
- Dades no estructurades
- Dades semiestructurades

Figura 1. Diferència entre les dades no estructurades, semiestructurades i estructurades



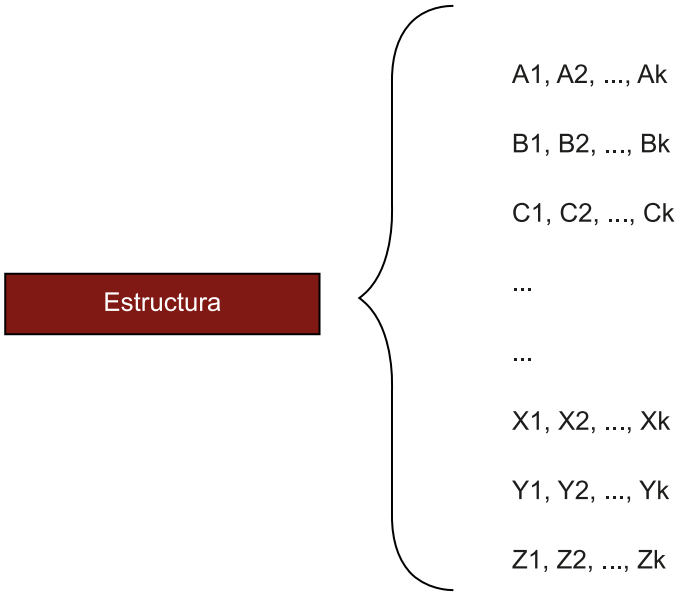
1.1.1. Dades estructurades

Les dades estructurades són dades que tenen una determinada longitud i un determinat format.

Alguns exemples de tipus de dades estructurades inclouen els nombres, les dates i els grups de paraules i nombres denominats **cadenes**.[1] En general, les dades estructurades segueixen un patró com el de la figura 2.

(1)En anglès, *strings*.

Figura 2. Forma general de les dades estructurades



Habitualment, les **dades estructurades** tenen predefinit el nombre de columnes, és a dir, el nombre *k* és fix. Algunes d'aquestes columnes podrien no sempre tenir dades, sent en aquest cas encara dades estructurades.

La major part de les dades estructurades es poden classificar en dades generades per màquines i dades generades per persones. Entre les dades estructurades generades per màquines s'inclouen:

- Dades de sensors, per exemple, les targetes d'identificació per radiofreqüència (RFID), mesuradors intel·ligents, dispositius mèdics, sensors en rellotges intel·ligents.
- Dades del sistema de posicionament global (GPS).
- Dades de registre d'activitat (*logs*).
- Dades financeres.
- Dades de punts de venda.

Les dades generades per l'ésser humà inclouen activitats i esdeveniments a les xarxes socials:

- **Dades d'entrada**, pot ser qualsevol dada que una persona pugui introduir en un sistema informàtic, com ara el seu nom, el telèfon, l'edat, el correu

electrònic, els ingressos, les adreces físiques o les respostes d'enquestes. Són útils per entendre el comportament dels usuaris.

- **Dades de *clickstream*** (tràfic), quan els usuaris naveguen per mitjà de llocs web o xarxes socials generen una gran quantitat de dades. Aquestes dades es poden registrar i analitzar per determinar el comportament dels clients o els patrons de compra, o per descobrir defectes en els processos.

- **Dades de videojocs** que inclouen l'activitat dels usuaris mentre juguen per internet a videojocs en una varietat de plataformes, com ara ordinadors, mòbils i consoles.

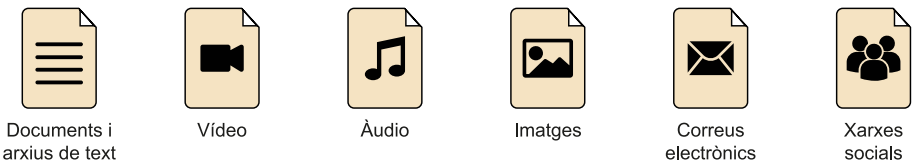
1.1.2. Dades no estructurades

Les dades que no s'ajusten a un model de dades o a un esquema de dades es coneixen com a **dades no estructurades**.

Qualsevol document que es compongui principalment de text, amb poca o cap estructura que descrigui el contingut del document, es pot classificar com a dades no estructurades. Les dades no estructurades es diferencien de les dades estructurades en el sentit que la seva estructura no és predictible.

Alguns exemples de dades no estructurades són documents, correus electrònics, blogs, informes, notes, imatges digitals, vídeos i imatges per satèl·lit. També poden ser dades no estructurades algunes dades generades per màquines o sensors (figura 3). De fet, les dades no estructurades representen la majoria de dades (s'estima que entorn del 80%) tant dins com fora de les organitzacions, i també de la web (LinkedIn, Twitter, Snapchat, Instagram, YouTube, Facebook, etc.).

Figura 3. Diversos tipus de dades no estructurades



Les dades no estructurades també es poden classificar com generades per màquines o per persones. La majoria dels conjunts de dades **generades per màquines** són imatges de satèl·lits, dades científiques, fotografies i vídeos. La majoria dels conjunts de dades no estructurades **generades per persones** són documents, dades de xarxes socials, dades de dispositius mòbils o llocs web.

Tècnicament, tant els arxius de text com els arxius d'àudio o vídeo tenen una estructura definida pel propi format d'arxiu, però aquest aspecte no és important aquí. La idea de «no estructurat» depèn del contingut de l'arxiu i no del format.

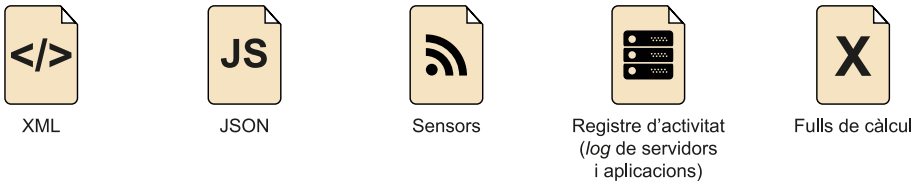
1.1.3. Dades semiestructurades

De vegades, les dades porten adjuntes etiquetes de metadades que proporcionen informació i context del contingut de les dades. Aquestes dades es consideren **dades semiestructurades**.

La línia divisòria entre les dades no estructurades i les semiestructurades no és fàcil d'establir, encara que alguns autors consideren que fins i tot les dades no estructurades tenen un cert grau d'estructura.

Les dades semiestructurades tenen un nivell definit d'una certa estructura i coherència, però no són de naturalesa relacional. En el seu lloc, les dades semiestructurades solen ser jeràrquiques o grafs. Aquest tipus de dada s'emmagatzema habitualment com a arxiu de text. Els arxius XML o JSON són formes freqüents d'emmagatzematge de dades semiestructurades (figura 4).

Figura 4. Alguns formats de dades semiestructurades



1.2. Operacions amb dades

El segon component d'un model de dades és un conjunt d'operacions que es poden realitzar amb les dades.

Les **operacions** indiquen els mètodes per tractar les dades.

Atès que els diferents models de dades solen estar associats amb estructures diferents, les operacions també seran diferents, encara que alguns tipus d'operacions es poden realitzar amb tots els models de dades.

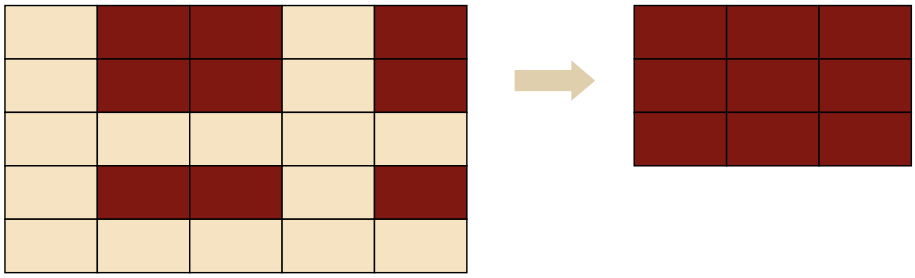
1.2.1. Subconjunt

Sovint, quan estem treballant amb un conjunt de dades gran, solament necessitarem una part d'aquest per a la seva anàlisi o tractament.

El procés de generar **subconjunts** (*subsetting*) consisteix a extreure les variables i les observacions necessàries per a una operació.

Depenent del context, també es denomina **selecció** o **filtratge**. Un subconjunt pot estar format per un nombre indeterminat de camps (columnes) i dades (files).

Figura 5. Exemple de subconjunts

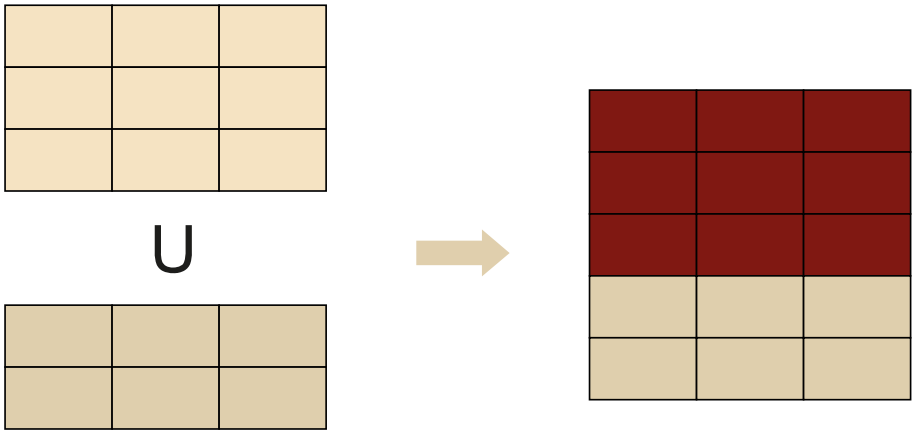


1.2.2. Unió

La suposició que subjau a l'operació d'**unió** és que les dades involucrades tenen la mateixa estructura.

Consisteix a incloure totes les dades dels diferents conjunts de dades en solament una unió. Aquesta operació també elimina les dades duplicades.

Figura 6. Exemple d'unió

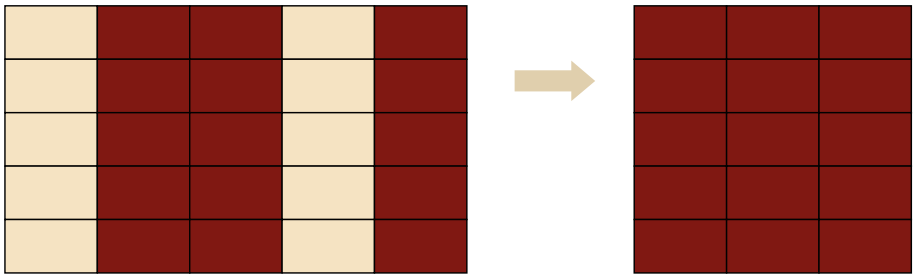


1.2.3. Projecció

Una altra operació usual consisteix a recuperar una part concreta de les dades. En aquest cas, especifiquem que estem interessats solament en certs camps d'una col·lecció de dades. Això produeix una nova col·lecció de dades que

conté exclusivament aquests camps. La diferència amb els subconjunts és que en la **projecció** s'inclouen totes les dades (files) de cadascun dels camps seleccionats (columnes).

Figura 7. Exemple de projecció

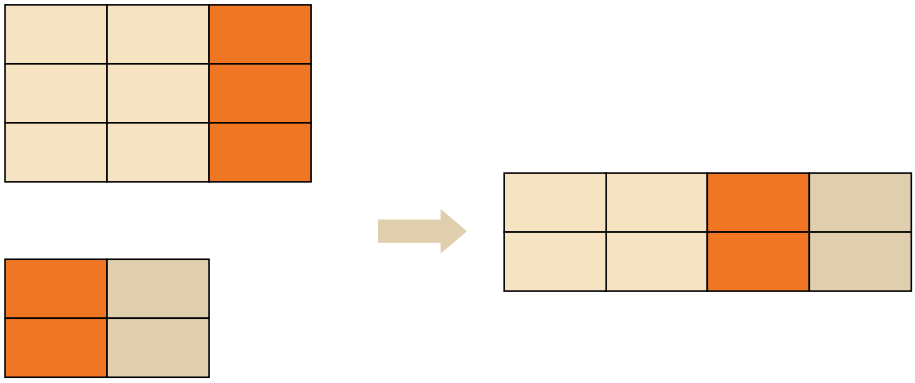


1.2.4. Connexió (*join*)

El segon tipus de combinació, denominada **connexió**,[2] es pot realitzar quan les dues col·leccions de dades tenen contingut diferent, però tenen alguns elements comuns (camps i dades).

(2)*Join*, en anglès, d'ús molt habitual.

Figura 8. Exemple de connexió o *join*



1.3. Restriccions de les dades

El tercer factor d'un model de dades són les restriccions sobre les dades.

Les **restriccions** són les instruccions lògiques que es poden imposar a les dades.

Hi ha diferents tipus de restriccions i els diferents models de dades tenen diferents formes d'expressar les restriccions.

1) Restriccions de valor. Una restricció de valor és una declaració lògica sobre el valor que poden tenir les dades. Per exemple, posem que un cert valor d'un atribut (edat) no pot ser negatiu.

2) Restriccions d'unicitat. Aquesta és una de les limitacions més importants. Aquesta restricció permet identificar de manera única cada element de la col·lecció. Per exemple, fent que el correu electrònic sigui únic per poder accedir a llocs web. És possible tenir més d'un atribut únic en una col·lecció.

3) Restriccions de cardinalitat. Requereix que es comptabilitzi el nombre de valors associats amb cada objecte i que es comprovi si es troben entre uns límits superior i inferior.

4) Restriccions de tipus. Per evitar que es pugui posar qualsevol valor a un atribut, és possible imposar un tipus de dades. Una restricció de tipus imposa un tipus de dades a un atribut. Per exemple, podem imposar que un atribut que conté el cognom d'una persona hagi de ser una cadena alfabètica i no pugui ser un nombre o una data. Una restricció d'aquest tipus és un cas particular de restricció de domini.

5) Restriccions de domini. El domini d'un atribut és el conjunt de possibles valors permesos per a aquest atribut. Per exemple, els mesos de l'any són entre 1 i 12, o que la puntuació d'un examen a la UOC és entre 0 i 10.

6) Restriccions estructurals. Una restricció estructural imposa restriccions a l'estructura de les dades en lloc de valors de les dades en si. Per exemple, podem imposar que l'estructura de les dades sigui matricial i que el nombre de files i columnes sigui el mateix.

2. Nivells de modelatge de dades

Un **model de dades** proporciona una representació visual de diversos aspectes funcionals, estructurals o de gestió d'una organització.

A més, un model de dades actua com una forma de comunicació entre les diferents parts interessades, tant tècniques com no tècniques.

De vegades, un model de dades il·lustra conceptes que s'han de comunicar o acordar, com si fos un plànol d'obra. Aquests plànols es construeixen amb diversos nivells de detall diferents, que van des d'uns requisits de disseny bàsics fins a especificacions de disseny detallades. Aquests plànols (o models) s'han de construir amb diferents nivells de disseny, denominats **nivells de modelatge de dades**.

La taula 1 presenta els tres nivells de modelatge de dades existents, de menor a major nivell de detall:

Taula 1. Diferents nivells de modelatge de dades

Nivells	Propòsit	Audiència
Modelatge conceptual de dades	Comunicació i definició de termes i regles.	Gestors i personal interessat en el projecte (*stakeholders*).
Modelatge lògic de dades	Clarificació i detall d'estructures de dades i regles.	• Arquitectes de dades • Analistes de negoci
Modelatge físic de dades	Implementació tècnica en sistemes reals.	• Desenvolupadors • Administradors de sistemes i de bases de dades

2.1. Modelatge conceptual de dades

El modelatge conceptual de dades és el primer pas en una perspectiva de menor a major nivell de detall, en què l'objectiu és capturar des d'una vista aèria els requisits de les dades d'una organització.

La principal raó per al **modelatge conceptual de dades** és capturar el panorama general i comprendre l'abast dels requisits d'alt nivell del projecte de dades en una organització.

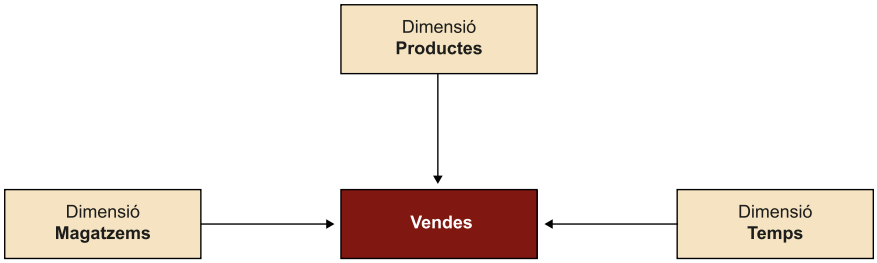
El modelatge conceptual de dades intenta respondre les preguntes següents:

- Quin problema de dades incideix en el negoci i necessita una solució?
- Quins són els conceptes principals relatius a aquest problema?
- Com es relacionen aquests conceptes entre si?

El resultat d'això és un model conceptual de dades. El model generat ha d'il·lustrar els conceptes clau necessaris per resoldre el problema de dades en particular. A més, hauria de proporcionar una descripció de l'esforç requerit per l'organització per realitzar-ho.

Per exemple, suposem un supermercat que vol mesurar les seves vendes. El supermercat pretén mesurar les vendes per producte, per magatzem i per data.

Figura 9. Modelatge conceptual de dades



El modelatge conceptual de dades ajuda a obtenir una anàlisi preliminar i les definicions dels termes clau, ajudant a comprendre els conceptes més importants del projecte i definint i documentant els requisits per a cadascun d'aquests conceptes. També permet explorar les relacions més importants entre els conceptes. Per tant, aquest tipus de models se solen denominar **models de domini**.

2.2. Modelatge lògic de dades

El **modelatge lògic** s'utilitza com a pas previ al modelatge físic i representa les estructures detallades, i també les seves relacions detallades.

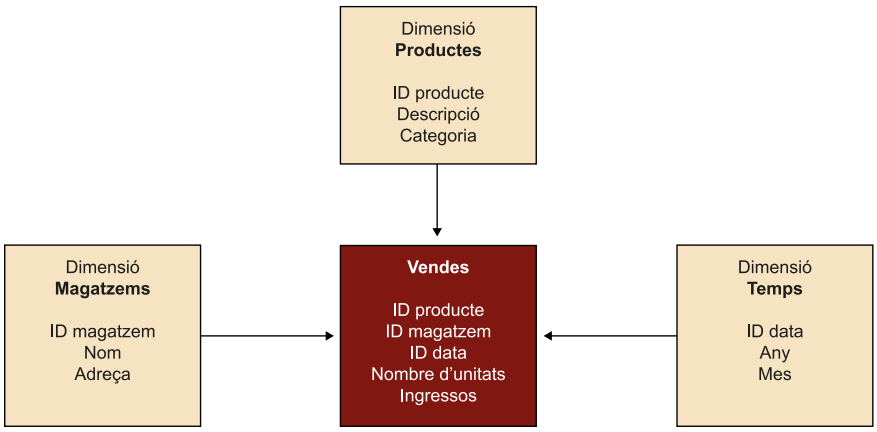
El model hauria de contenir totes les entitats essencials i els atributs, a més de les relacions que tenen entre si. El model no depèn de la implementació. Per tant, els models lògics han de ser flexibles i adaptables independentment del sistema en què s'implementarà.

Els **beneficis** principals de construir un model lògic són:

- Facilita la comprensió dels elements de dades i els seus requisits.
- Ajuda a evitar la duplicitat i inconsistència de les dades.
- Promou la reutilització i l'intercanvi de dades.

En la figura 10 es representa el model lògic aplicat a l'exemple del supermercat.

Figura 10. Modelatge lògic de dades



En aquest nivell de modelatge, les preguntes que s'haurien de respondre serien:

- Com donar resposta a qüestions relatives a les dades al més aviat possible?
- Com és la forma òptima d'organitzar la informació?
- Com emmagatzemar les dades històriques?
- Com fer que la informació sigui segura?

El modelatge lògic proporciona un mapa general que pot incloure múltiples tecnologies. Encara que el model s'entén des d'una perspectiva independent de la tecnologia, el modelatge lògic implica normalitzar i abstreure per obtenir les característiques següents:

- Totes les entitats i relacions entre elles.
- Tots els atributs de totes les entitats.
- Les restriccions sobre les dades (de valor, unicitat, cardinalitat, tipus i domini).

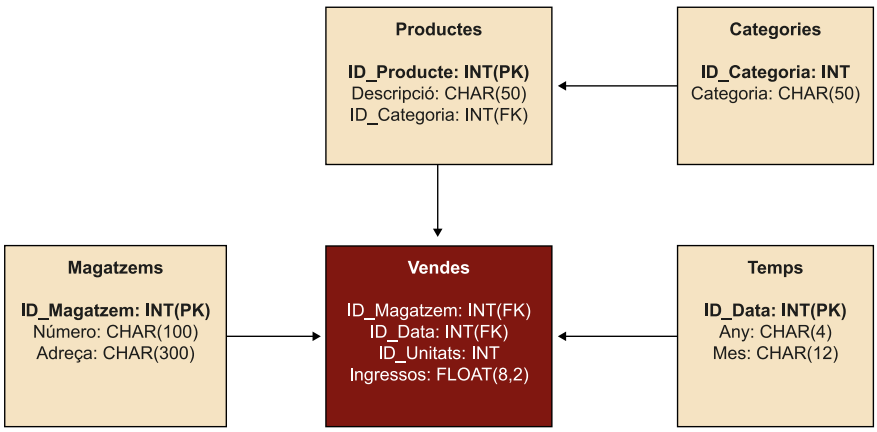
2.3. Modelatge físic de dades

El **modelatge físic de dades** il·lustra com es construiran els diferents elements de dades, segons els requisits proporcionats pel model lògic.

Facilita una visió de com és l'estructura física del conjunt de dades en la implementació. També es poden incloure qüestions d'optimització o de millora de rendiment. El model físic conté totes les estructures de dades, inclosos els noms dels atributs, els tipus de dades, les restriccions dels atributs i les relacions entre aquestes estructures.

En la figura 11 es representa el model físic aplicat a l'exemple del supermercat.

Figura 11. Modelatge físic de dades



El disseny d'un model físic de dades es determina per la tecnologia amb la qual s'implementarà i estarà optimitzat segons els requisits d'aquesta tecnologia. Algunes **tecnologies d'implementació** freqüents serien:

- Bases de dades relacionals
- Bases de dades no relacionals o NoSQL
- Esquemes XML
- Sistemes llegats (*legacy*)

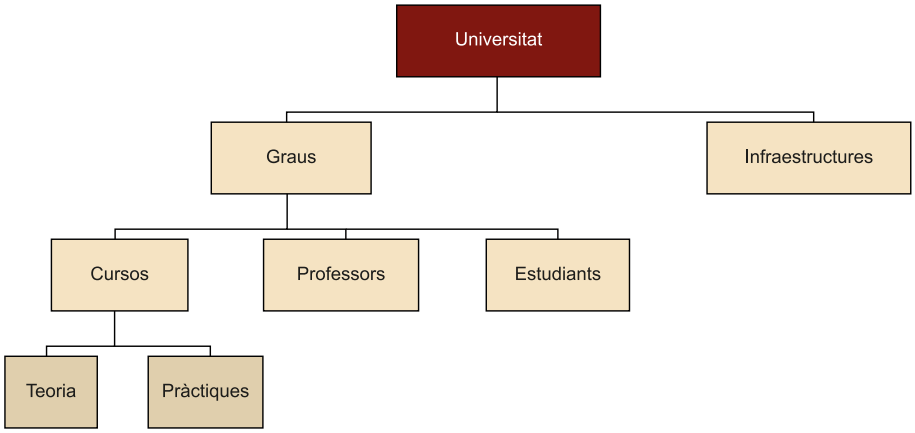
3. Tipus de models de dades

Una vegada conegut que els models de dades són molt importants en la construcció d'una estructura de gestió de dades, aquests models es poden classificar en diferents tipus, en funció de les estructures que utilitzen. En aquesta secció veurem les més comunes.

3.1. Model jeràrquic

Un model jeràrquic utilitza una estructura en forma d'arbre per representar les dades, amb un sol origen (denominat **arrel**) per a cada registre. El registre conté informació sobre un tema en particular i està connectat amb altres registres mitjançant enllaços. Hi ha un ordre concret d'organització dels registres dins de cada nivell de l'arbre (figura 12).

Figura 12. Model jeràrquic de dades



Com es mostra en la figura 12, cada branca de la jerarquia representa una sèrie de registres relacionats. Una relació en un model jeràrquic és una relació pare/fill. Per accedir a les dades dins d'aquest model cal començar a l'arrel i descendir per l'arbre fins a les dades de destinació. Cal conèixer l'estructura del model per poder accedir a les dades.

Accedir a les dades

Per exemple, en el model jeràrquic que mostra la figura 12, per accedir a les dades d'«Estudiants», cal conèixer tot el model.

3.2. Model relacional

Els models relacionals organitzen les dades en taules denominades **relacions**. Les relacions es componen de files i columnes. Cada fila té un valor únic, anomenat *tupla*, i cada columna alberga un valor, anomenat *atribut*:

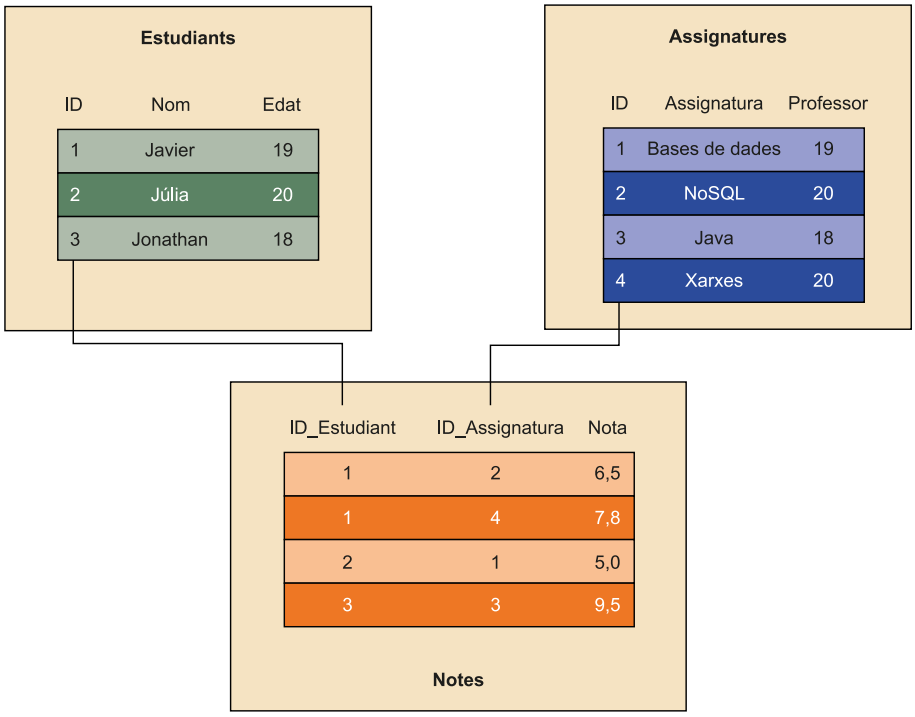
- Una **clau primària** (*primary key*) és un atribut o una combinació d'atributs que tenen la restricció d'unicitat i sempre tenen valors.

- Una clau primària en una altra taula es denomina **clau externa** (*foreign key*).
- Una **tupla** inclou les dades pròpies d'una entitat.
- El **grau** d'una relació és el nombre d'atributs en la relació.
- La **cardinalitat** és el nombre de tuples d'una relació.

Un **model relacional** és un mètode declaratiu per especificar tant dades com consultes.

Això implica que cal declarar directament les dades existents i, amb això, es possibilita que el programari defineixi les estructures de dades per gestionar-les i recuperar-les de forma eficient.

Figura 13. Exemple de model relacional



El model relacional és el més popular i estès de tots els models de dades per ser el més utilitzat en l'ecosistema dels sistemes de gestió de bases de dades. Els **beneficis** d'usar un model de dades relacional són els següents:

- L'avantatge principal és la seva capacitat per descriure dades de forma senzilla.
- El procés de recuperació de registres se simplifica usant els atributs clau.

- És possible representar diferents tipus de relació amb aquest model.

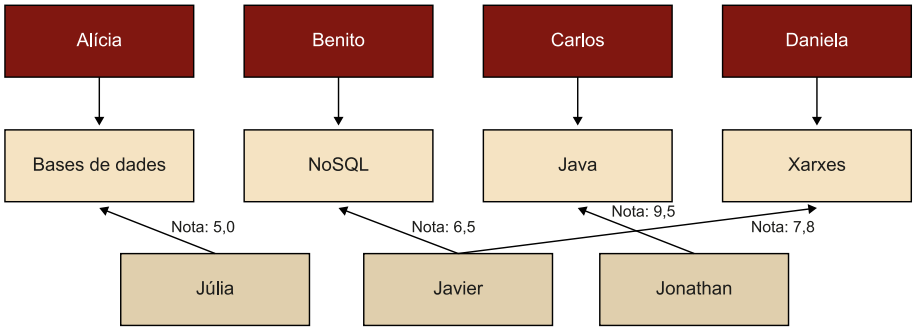
3.3. Model en xarxa

Un **model en xarxa** està dissenyat com un enfocament flexible per representar dades i les seves relacions. En un model de dades en xarxa, les dades es representen en termes de **nodes** i d'**enllaços**:

- Un **node** representa un registre, que és una col·lecció d'atributs.
- Un **enllaç** representa la relació entre dos nodes.

El model de dades en xarxa és semblant al model de dades jeràrquiques, però és més senzill i la seva implementació és més fàcil.

Figura 14. Exemple de model en xarxa



3.4. Model orientat a objectes

Un **model de dades orientat a objectes** és un model de dades que tracta els conjunts de dades com a «objectes».

Aquests objectes són entitats que inclouen tant atributs com comportaments, és a dir, combinen dades i procediments per treballar amb les dades.

Els **elements** d'un model de dades orientat a objectes són:

- Els **objectes** que representen les entitats o situacions del món real.

- Els **atributs** representen certes característiques dels objectes i els **mètodes** representen el comportament dels objectes.

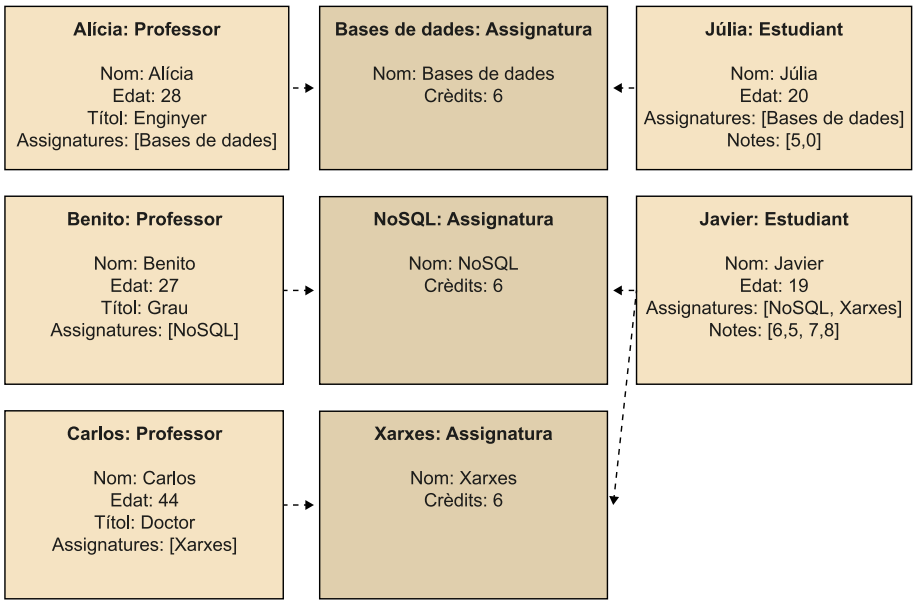
- Les **classes** són generalitzacions d'objectes que comparteixen atributs i mètodes. Un objecte es diu que és una **instància** d'una classe.

Amb aquest tipus de model es poden respondre preguntes sobre els conjunts de dades del tipus:

- Quants d'aquests «objectes» s'ajusten a un determinat format?
- Quantes dades contenen cadascun d'aquests?

L'avantatge principal d'aquest enfocament és que s'adapta perfectament a treballar amb llenguatges orientats a objectes. No obstant això, el model relacional segueix sent el més utilitzat. Per tant, també hi ha un **model objecte-relacional**, un model híbrid que combina un model relacional amb algunes funcionalitats superiors del model orientat a objectes. És a dir, permet un model de dades orientat a objectes virtualitzat sobre un model relacional.

Figura 15. Exemple de model de dades orientat a objectes



Bibliografia

Abiteboul, S.; Buneman, P.; Suciu, D. (2000). *Data on the Web: From Relations to Semistructured Data and XML. The Morgan Kaufmann Series in Data Management Systems*.

Batini, C. (1991). *Conceptual Database Design: An Entity-Relationship Approach*. Londres: Pearson Education.

Google. «Comprende cómo funcionan los datos estructurados». Disponible a: https://developers.google.com/search/docs/guides/intro-structured-data

Inmon, W. H.; Linstedt, D. (2014). *Data Architecture: A Primer for the Data Scientist: Big Data, Data Warehouse, and Data Vault*. Massachusetts: Morgan Kaufmann.

Singh Birgi, J.; Khaire, M.; Hira, S. (2016). «Data Model: A Blueprint for Data Warehouse». *International Journal of Scientific and Research Publications* (vol. 6, núm. 1).

Wei, T.; Lee, J.; Kumar Mukhiya, S. (2018). *Hands-On Big Data Modeling*. Packt Publishing.

Yannakoudakis, E. J. (2013). *The Architectural Logic of Database Systems*. Berlín: Springer Verlag.