

Introducció al disseny de bases de dades

Jordi Casas Roma

PID_00187549



Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>

Índex

Introducció	5
Objectius	6
1. Procés de disseny d'una base de dades	7
2. Fases del disseny d'una base de dades	10
2.1. Fase 1. Recollida i anàlisi de requisits	10
2.1.1. Recollida de requisits	10
2.1.2. Estructuració i refinament dels requisits	11
2.1.3. Formalització dels requisits	11
2.2. Fase 2. Disseny conceptual	12
2.2.1. El model ER	12
2.2.2. El llenguatge unificat de modelització	13
2.3. Fase 3. Disseny lògic	13
2.3.1. Reconsideracions del model conceptual	14
2.3.2. Transformació del model conceptual al model lògic	14
2.3.3. Normalització	14
2.4. Fase 4. Disseny físic	15
2.4.1. El nivell físic i el nivell virtual	15
2.4.2. Transformació del model lògic en el model físic	16
2.5. Fase 5. Implementació i optimització	17
2.5.1. Processament i optimització de consultes	17
2.5.2. Processament de vistes	18
2.5.3. Administració de la seguretat	19
Resum	20
Glossari	21
Bibliografia	22

Introducció

El disseny de bases de dades és un procés complex que permet obtenir una implementació d'una base de dades a partir dels requisits inicials dels usuaris del sistema d'informació. Aquest procés guia el dissenyador de bases de dades per diverses etapes amb l'objectiu de segmentar un problema d'una complexitat considerable en diferents subproblemes de menor complexitat.

Cadascun dels subproblemes identificats correspon a una de les etapes del procés de disseny de bases de dades. En aquests materials didàctics es descriu el procés global de disseny de bases de dades i les diferents etapes que el componen. Aquest mòdul és només un text introductori al disseny de bases de dades, i caldrà que aprofundi en l'estudi de cadascuna de les etapes per mitjà dels diversos mòduls de l'assignatura que corresponen a les diferents etapes. Així doncs, en aquest mòdul es presenta una visió general de tot el procés de disseny de bases de dades.

Objectius

En aquests materials trobareu les eines indispensables per a assolir els objectius següents:

- 1.** Entendre en què consisteix el disseny de bases de dades i quins en són els objectius.
- 2.** Conèixer les diverses etapes que integren el procés de disseny d'una base de dades.

1. Procés de disseny d'una base de dades

El procés de disseny de bases de dades consisteix a definir l'estructura lògica i física d'una o més bases de dades per a respondre a les necessitats dels usuaris respecte a la informació i per a un conjunt concret d'aplicacions.

Mitjançant un procés de disseny de bases de dades es poden decidir les taules i relacions que ha de tenir una base de dades determinada, els atributs de les diferents taules, les claus primàries i les claus foranes que s'han de declarar en cada taula, etc. Totes aquestes tasques formen part del procés de disseny de bases de dades. Per a poder prendre aquestes decisions de la manera més correcta possible, cal tenir en compte les necessitats d'informació dels usuaris en relació amb un conjunt concret d'aplicacions.

Per tant, el disseny d'una base de dades és el procés en què es defineix l'estructura de les dades que ha de tenir la base de dades d'un sistema d'informació determinat.

Els requisits que ha de complir un sistema d'informació i la complexitat de la informació que s'hi presenta fan que el disseny d'una base de dades sigui un procés complicat. Per a simplificar aquest procés, és molt recomanable utilitzar l'estratègia de dividir per a vèncer (*divide and conquer*).

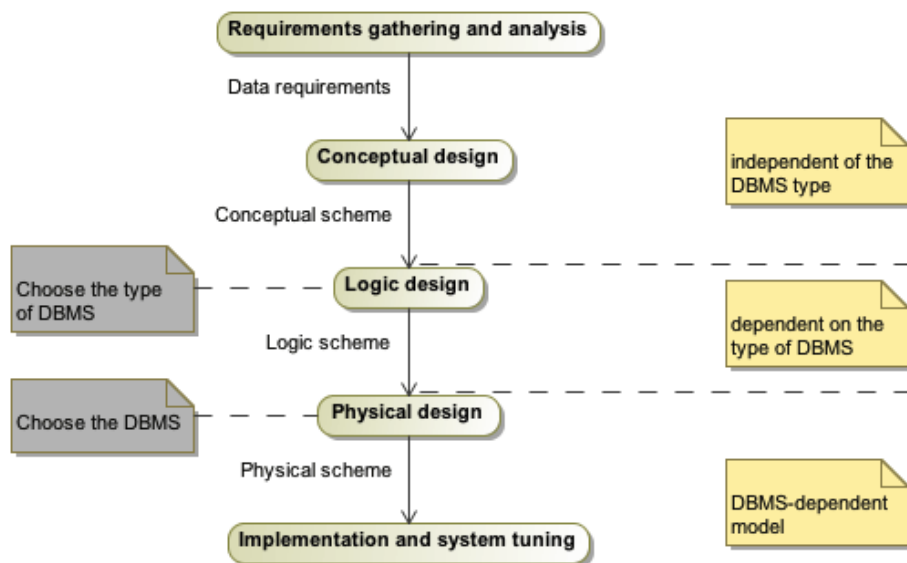
Aplicant aquest concepte obtenim les diferents etapes del disseny de bases de dades. Aquestes etapes són seqüencials i el resultat de cadascuna serveix de punt de partida de l'etapa següent. El resultat de la darrera etapa serà el disseny final de la nostra base de dades. D'aquesta manera, un procés d'una certa complexitat es descompon en diferents processos de menor complexitat. La figura 1 mostra les diverses etapes del disseny de bases de dades.

En primer lloc tenim la **recollida i anàlisi de requisits**. Aquesta etapa ha de permetre obtenir els requisits i les restriccions de les dades del problema. Per a obtenir aquesta informació caldrà mantenir converses amb els diferents usuaris de la futura base de dades i de les aplicacions que hi estan relacionades. Només si s'encreuen els requisits dels diferents perfils d'usuaris serà possible establir un marc complet dels requisits i les restriccions de les dades relacionades amb la futura base de dades.

Dividir per a vèncer

L'estratègia de dividir per a vèncer proposa resoldre un problema complex mitjançant la subdivisió en un conjunt de problemes més senzills en què la resolució dels diferents sub-problemes comporta solucionar el problema inicial.

Figura 1. Etapes del disseny de bases de dades



Sistema gestor de bases de dades

Un sistema gestor de bases de dades (SGBD; en anglès *database management system*, DBMS) és un tipus de programari específic que serveix d'interfície entre la base de dades, l'usuari i les aplicacions que la utilitzen.

A continuació s'inicia el **disseny conceptual**. En aquesta etapa es crea un esquema conceptual d'alt nivell a partir de les especificacions i els requisits obtinguts en l'etapa anterior. En aquest procés cal extreure les necessitats i els requisits de la problemàtica i sintetitzar-los en un model visual que permeti representar les dades i les restriccions dels conceptes que es volen modelitzar en el sistema d'informació. Aquest model s'anomena *esquema conceptual*.

Fins aquesta etapa del disseny de bases de dades encara no ha calgut escollir el tipus de bases de dades que s'utilitzarà (relacional, orientada a objectes, documental, etc.) ni tampoc el sistema gestor de bases de dades (SGBD¹) que es farà servir o el llenguatge concret amb què s'implementarà la base de dades.

⁽¹⁾SGBD és la sigla de *sistema gestor de bases de dades*.

En el moment en què s'inicia la tercera etapa del procés de disseny, el **disseny lògic**, s'ha de determinar el tipus de bases de dades que s'utilitzarà. És a dir, no és necessari encara escollir un SGBD concret, però sí el tipus de bases de dades que es vol emprar. En aquesta etapa l'esquema conceptual es converteix en un esquema lògic adequat al tipus de bases de dades que es pretén fer servir.

Per *tipus de bases de dades* entenem els diferents grups de bases de dades segons el model de dades que apliquen. Actualment hi ha diversos tipus d'SGBD, entre els quals els més utilitzats són les bases de dades relacionals, orientades a objectes, documentals, geogràfiques o multidimensionals, entre d'altres. Per exemple, les bases de dades relacionals són el conjunt de tots els SGBD que apliquen models de dades relacionals.

En aquest punt, i abans d'iniciar l'etapa de **disseny físic**, cal escollir un SGBD concret sobre el qual es pretén implementar la base de dades. L'etapa de disseny físic adapta l'esquema lògic a les necessitats específiques d'un SGBD concret i, posteriorment, ajusta alguns paràmetres per al funcionament correcte de la base de dades. Per *base de dades concreta* o *SGBD concret* entenem una

aplicació concreta de bases de dades. En el cas de bases de dades relacionals, exemples de models concrets d'SGBD són Oracle Database, Mysql, SQL Server o IBM Informix, entre d'altres.

Finalment, la darrera etapa és la **implementació i optimització** de la base de dades. Aquesta etapa permet carregar les dades i posteriorment permet ajustar alguns paràmetres del model físic i de la base de dades per a optimitzar-ne el rendiment.

Aquestes etapes del disseny no s'han de seguir estrictament de manera seqüencial, i en molts casos és habitual refer el disseny de l'etapa anterior a partir de necessitats detectades en fases posteriors. Aquests **bucles de retroalimentació** són habituals i permeten afinar els dissenys de les diverses etapes d'una manera iterativa.

El procés que mostra la figura 1 es basa en el model de **disseny orientat a dades**. Aquest model se centra en el disseny dels contenidors de la informació i en l'estructura de la base de dades. Paral·lelament a aquest model existeix el model de **disseny orientat a processos**, que es concentra en les aplicacions de bases de dades per a determinar les dades i l'ús que en fan les aplicacions. Tradicionalment, el disseny d'aplicacions s'ha basat en aquest segon model, però cada vegada queda més clar que totes dues activitats són paral·leles i que estan estretament interrelacionades. Les eines de disseny de bases de dades i d'aplicacions les combinen cada vegada més.

2. Fases del disseny d'una base de dades

A continuació veurem amb una mica més de detall les fases que componen el procés de disseny d'una base de dades.

2.1. Fase 1. Recollida i anàlisi de requisits

La primera fase en el disseny d'una base de dades consisteix a conèixer i analitzar amb detall les expectatives, les necessitats i els objectius dels futurs usuaris de la base de dades. Aquest procés s'anomena *recollida i anàlisi de requisits*.

La fase de recollida i anàlisi de requisits es pot dividir en tres subfases seqüencials: la recollida de requisits; l'estructuració i el refinament dels requisits, i la formalització dels requisits.

2.1.1. Recollida de requisits

Per a determinar els requisits, en primer lloc s'han d'establir els actors del sistema d'informació que interaccionaran amb la base de dades. Això inclou els usuaris i les aplicacions, tant si són nous com si no ho són. Normalment, un grup d'analistes s'encarrega de fer l'anàlisi dels requisits, i, molt probablement, aquesta anàlisi serà informal, incompleta i fins i tot incoherent en algun punt. Per tant, cal dedicar molts esforços a treballar aquesta informació i convertir-la en una especificació que els dissenyadors de bases de dades puguin utilitzar per a modelitzar i implementar el sistema d'informació.

En aquesta fase, no solament cal recollir i analitzar els requisits referents a l'estructura o la forma de la informació (tipus de dades i relacions entre ítems de dades), sinó que s'ha de capturar i analitzar qualsevol tipus de requisit, independentment de quin tipus sigui. Cal recollir, analitzar i documentar qualsevol requisit que els usuaris esperin de la base de dades. Això inclou els processos que s'han d'executar sobre la base de dades, les restriccions que cal implementar sobre les dades, les restriccions en el rendiment del sistema d'informació, les restriccions en la implementació (tant pel que fa al maquinari² com al programari³), requisits de seguretat o requisits de rendiment (per exemple, temps de resposta), entre d'altres. Algunes de les activitats més habituals d'aquesta fase són les següents:

- Identificar els grups d'usuaris i les principals àrees d'aplicació que faran ús de la base de dades i que es veuran directament o indirectament afectats

⁽²⁾En anglès, *hardware*.

⁽³⁾En anglès, *software*.

per la base de dades. Dins de cada grup cal escollir usuaris clau i formar comitès per a portar a terme la recollida i l'especificació dels requisits.

- Estudiar i analitzar la documentació existent relativa a les aplicacions en ús.
- Estudiar l'entorn actual i l'ús que es vol donar a la informació. Això inclou estudiar les entrades, el flux i les sortides d'informació, a més de les freqüències i els usos de les diferents tasques dins el sistema d'informació.
- Fer entrevistes i enquestes als futurs usuaris, per tal que puguin manifestar la seva opinió i les prioritats que tenen en el nou sistema d'informació.

2.1.2. Estructuració i refinament dels requisits

S'ha de tenir en compte que alguns d'aquests requisits, molt probablement, canviaran durant el procés de disseny, i que cal estar atents i en contacte d'una manera continuada amb els usuaris de la base de dades per a detectar problemes en els requisits. És una bona pràctica incorporar els usuaris de la base de dades en el procés de desenvolupament, ja que així s'incrementa el grau d'implicació i de satisfacció d'aquests usuaris. Hi ha algunes propostes de metodologies per a la recollida i l'anàlisi de requisits basades en el treball conjunt amb els usuaris de la base de dades, com, per exemple, el disseny conjunt d'aplicacions (JAD⁴).

⁽⁴⁾JAD és la sigla de l'expressió anglesa *joint application design*.

2.1.3. Formalització dels requisits

El pas següent és convertir els requisits a un format estructurat, mitjançant tècniques d'especificació de requisits, com, per exemple, l'anàlisi orientada a objectes (OOA⁵), diagrames de flux de dades (DFD⁶) o la notació Z. Aquestes tècniques utilitzen diferents tipus de recursos (diagrames, text, taules, gràfics, diagrames de decisió, etc.) per a organitzar i representar d'una manera clara els requisits.

⁽⁵⁾OOA és la sigla de l'expressió anglesa *object oriented analysis*.

⁽⁶⁾DFD és la sigla de l'expressió anglesa *data flow diagrams*.

La notació Z

La notació Z és un llenguatge formal utilitzat en enginyeria del programari.

Aquesta fase pot representar un cost important dins el procés de disseny d'una base de dades, però és molt important i pot ser determinant per a l'èxit o el fracàs del sistema d'informació. Detectar i corregir els errors o problemes en les fases inicials del projecte és molt menys costós que arrossegar els errors fins a les fases finals, quan corregir-los tindrà uns costos molt més importants. La satisfacció de l'usuari final serà determinada per la capacitat de recollir i captar les necessitats que té i d'implementar-les de manera correcta en la solució final.

2.2. Fase 2. Disseny conceptual

La fase de **disseny conceptual** té com a objectiu crear un esquema conceptual d'alt nivell i independent de la tecnologia a partir dels requisits, les especificacions i les restriccions que s'han recollit en la fase anterior.

En aquesta fase es parteix de la recollida i l'anàlisi de requisits obtinguts en la fase anterior, i té com a objectiu dissenyar un esquema conceptual de la base de dades que sigui consistent amb els requisits, les especificacions i les restriccions imposats per la problemàtica que s'ha de resoldre.

Un **esquema conceptual** és una descripció concisa dels requisits de dades, que s'expressa mitjançant conceptes proporcionats per un model de dades d'alt nivell, fàcil d'entendre i que no conté detalls d'implementació. L'esquema, a més, ha de servir de referència per a verificar que s'han agrupat tots els requisits i que no hi ha cap conflicte entre ells.

En aquesta fase del disseny encara no es té en compte el tipus de base de dades que s'utilitzarà. I, per tant, tampoc no es considera l'SGBD ni el llenguatge concret amb què s'implementarà la base de dades. En aquesta etapa ens concentrarem en l'estructura de la informació, sense resoldre de moment qüestions relacionades amb la tecnologia.

2.2.1. El model ER

Hi ha diversos models de dades d'alt nivell que permeten modelitzar els requisits, les especificacions i les restriccions que s'han obtingut en la primera fase del disseny d'una base de dades. Un dels més coneguts i utilitzats és el model entitat-interrelació, que abreuja com a **model ER**. Aquest model és un dels més utilitzats en el disseny conceptual de les aplicacions de bases de dades, principalment, a causa de la seva simplicitat i facilitat d'ús.

Els principals elements que inclou el model són els tipus d'entitat, els atributs i les classes de relacions entre entitats. L'objectiu principal del model ER és permetre als dissenyadors reflectir en un model conceptual els requisits del món real que siguin interessants per a la problemàtica que s'ha de resoldre. El model ER facilita el disseny conceptual d'una base de dades i, com ja hem comentat, és aplicable al disseny de qualsevol tipus de bases de dades.

Model ER

En anglès s'anomena *entity-relationship model*. Atesa l'ambigüitat de la traducció, en llengua catalana trobem autors que ho tradueixen com a *model entitat-relació* i d'altres que ho tradueixen com a *model entitat-interrelació*. Tots dos conceptes es refereixen al mateix model.

2.2.2. El llenguatge unificat de modelització

El llenguatge unificat de modelització (UML) és un llenguatge gràfic dissenyat per a especificar, visualitzar, modificar, construir i documentar un sistema. El llenguatge UML incorpora una gran quantitat de diagrames que permeten representar, des de perspectives diferents, el model d'un sistema. Amb vista al disseny conceptual de bases de dades, ens interessa especialment el diagrama de classes, que permet representar informació del domini de discurs. Els diagrames de classes són diagrames estàtics que descriuen l'estructura d'un sistema a partir de les classes o tipus d'entitat del sistema, els atributs que tenen i les associacions o tipus de relacions que s'estableixen. Aquests diagrames han mostrat una capacitat excel·lent per a la modelització de dades. Per aquest motiu, han esdevingut cada vegada més importants en el disseny conceptual de bases de dades

Llenguatge UML

El llenguatge unificat de modelització (en anglès, *unified modeling language*) és un llenguatge de propòsit general per a modelitzar sistemes de programari. L'estàndard va ser creat, i actualment és mantingut, pel Grup de Gestió d'Objectes.

2.3. Fase 3. Disseny lògic

Prèviament a la fase de disseny lògic, cal escollir un tipus de base de dades. És a dir, no s'ha de triar, encara, un SGBD concret, sinó que només cal seleccionar el tipus de base de dades que es vol implementar. És important que quedi clar que el tipus de base de dades determina l'esquema de disseny lògic que s'ha de construir. Una vegada escollit el tipus d'SGBD on es vol implementar la base de dades, ja es pot iniciar la fase del disseny lògic.

En la fase de **disseny lògic** es transforma el model conceptual, independent del tipus de tecnologia, en un model lògic dependent del tipus d'SGBD en el qual es vol implementar la base de dades.

En aquesta etapa es parteix del disseny conceptual desenvolupat en el pas anterior i s'obté un disseny lògic de la futura base de dades. En aquesta transformació s'ajusta el model tenint en compte el tipus d'SGBD en el qual es vol implementar la base de dades. Per exemple, si es vol crear la base de dades en un sistema relacional, aquesta etapa obtindrà un conjunt de relacions amb els atributs, les claus primàries i les claus foranes corresponents.

En aquesta etapa ens concentrem en les qüestions tecnològiques relacionades amb el model de la base de dades, assumint que en l'etapa anterior ja hem deixat resolta la problemàtica de l'estructuració de la informació des del punt de vista conceptual.

Per tant, el resultat d'aquesta etapa serà un model lògic de l'estructura de la informació. Generalment, quan aquest model lògic fa referència a un SGBD relacional, s'anomena *model relacional*. En aquest material ens centrarem en la transformació del model conceptual en un model relacional, és a dir, en un model lògic per a una base de dades relacional.

En el disseny lògic es poden veure tres subfases o parts independents, que s'apliquen de manera seqüencial per a transformar el model conceptual obtingut en la fase anterior en un model lògic que serà el resultat d'aquesta fase. En primer lloc, en la subfase anomenada *reconsideracions del model conceptual* revisem el model conceptual per assegurar-nos que està lliure d'alguns errors tipificats i identificables. A continuació, es produeix la transformació del model conceptual en el model lògic i, finalment, apliquem la teoria de la normalització al model lògic.

2.3.1. Reconsideracions del model conceptual

En aquesta primera part es realitza una anàlisi acurada del model conceptual obtingut en la fase anterior, amb la intenció de detectar i corregir alguns errors que se solen produir en els models conceptuals i que cal detectar i reparar al més aviat possible, per a evitar que es propaguin en fases posteriors. Aquests errors s'anomenen *paranys de disseny*. Cal conèixer cadascun d'aquests errors i assegurar-se que el model conceptual que es vol transformar n'està lliure abans de continuar amb el disseny lògic.

2.3.2. Transformació del model conceptual al model lògic

En aquests materials ens centrarem en el disseny lògic de bases de dades relacionals. Per tant, el model lògic generat serà aplicable a qualsevol base de dades relacional. Partirem del resultat de l'etapa del disseny conceptual expressat mitjançant un diagrama de classes UML i veurem com es pot transformar utilitzant una estructura de dades del model relacional. Aquest procés mostra la manera de dur a terme la transformació dels diferents elements que componen el model conceptual en elements que constitueixen el model lògic o, més concretament, el model relacional.

El model relacional

El model relacional és el model lògic específic per a bases de dades relacionals.

2.3.3. Normalització

La teoria de la normalització aplica la teoria de conjunts, la lògica i l'àlgebra relacional per a formalitzar un conjunt d'idees simples que guien un bon disseny de bases de dades relacionals.

La teoria de la normalització utilitza les formes normals (FN) per a reconèixer els casos en què no s'apliquen bons criteris de disseny. Una relació està en una forma normal determinada si satisfà un conjunt de restriccions específiques que són pròpies d'aquesta forma normal. La infracció d'aquestes restriccions origina que la relació tingui un conjunt d'anomalies i les redundàncies

d'actualització no desitjables. Les formes normals són declaratives, és a dir, cada forma normal indica les restriccions que s'han de complir, però no descriu cap procediment per a aconseguir-ho.

Quan una relació no compleix una forma normal és perquè infringeix la restricció associada a aquesta forma normal. Per a aconseguir que es verifiqui la forma normal, caldrà evitar la condició que fa que s'infringeixi la restricció en qüestió. El procediment que aplicarem per a aconseguir que no s'infringeixi la restricció associada a la forma normal s'anomena *normalització*.

Hi ha diverses formes normals entre les quals estudiarem les sis més importants. Cada forma normal indica les restriccions específiques que ha de complir una relació. Com més gran és el grau d'una forma normal, més restrictiva és aquesta forma normal i elimina més redundàncies que les formes normals de grau inferior, atès que inclou les restriccions d'aquelles i agrega una restricció addicional.

2.4. Fase 4. Disseny físic

Prèviament a la fase de disseny físic s'ha d'escollir un SGBD concret. Cal estudiar els diferents sistemes comercials o lliures que hi ha en el mercat i seleccionar un SGBD on es pugui implementar el sistema d'informació que s'ha anat gestant en les fases anteriors del procés de disseny.

Els components físics que formen cada SGBD són específics. Els fabricants utilitzen estratègies i tecnologies diferents per a maximitzar el rendiment dels seus sistemes gestors de bases de dades. En aquest nivell no existeix cap estàndard i, per tant, caldrà adaptar l'esquema lògic, obtingut en el pas anterior, tenint presents les característiques de cada sistema gestor. El dissenyador ha de considerar els aspectes d'implementació física i d'eficiència que depenen específicament de l'SGBD escollit.

El **disseny físic** és una fase del procés de disseny de bases de dades que adapta l'esquema lògic obtingut en la fase anterior a l'SGBD concret on caldrà desplegar el sistema d'informació.

2.4.1. El nivell físic i el nivell virtual

L'estudi dels nivells físic i virtual de les bases de dades permet veure aspectes com les estructures d'emmagatzematge i les rutes d'accés pels fitxers de la base de dades. Cada SGBD ofereix diferents opcions per a organitzar fitxers i rutes d'accés, i cal que el dissenyador conegui la implementació concreta per a poder implementar d'una manera més eficient el disseny físic del sistema d'informació.

També és important conèixer les característiques dels processos que consulten i actualitzen la base de dades, com, per exemple, les freqüències d'execució i els volums que s'espera tenir de les diferents dades que es volen emmagatzemar, per a poder aconseguir un bon rendiment de la base de dades.

Alguns criteris importants que poden servir per a escollir les opcions de disseny físiques de la base de dades són els següents:

- **Temps de resposta.** És el temps que transcorre entre que s'envia una petició a l'SGBD fins que aquest retorna les dades de la resposta. Una part important d'aquest temps està sota el control de l'SGBD i fa referència al temps d'accés per part de l'SGBD a les dades requerides per a generar la resposta. Altres aspectes no estan sota el control de l'SGBD, com ara la planificació del sistema operatiu o els temps d'accés als medis físics d'emmagatzematge de les dades.
- **Ús de l'espai.** És la quantitat d'espai de disc utilitzat pels fitxers de la base de dades i les estructures de rutes d'accés al disc, incloent-hi índexs i altres rutes d'accés.
- **Rendiment.** És la quantitat mitjana de transaccions que es poden processar en un minut de temps. Aquest factor pot ser crític per a sistemes transaccionals, com ara línies aeronàutiques o entitats bancàries.

2.4.2. Transformació del model lògic en el model físic

En aquest pas es transforma el model lògic d'una base de dades en un model físic, depenent de l'SGBD escollit per a implementar el sistema d'informació, les característiques concretes del maquinari utilitzat i el sistema operatiu i el programari bàsic. Cada SGBD ha desenvolupat un llenguatge propi, fet a mida pel mateix constructor, per a implementar el disseny físic de la base de dades, d'acord amb les característiques de l'entorn, i per a obtenir el màxim rendiment del maquinari, del sistema operatiu i del mateix gestor. De fet, es podria considerar una ampliació del llenguatge SQL estàndard, amb les clàusules pròpies, que cada gestor necessita per a definir els components del disseny físic. No obstant això, hi ha una gran semblança o equivalència entre una part important dels diferents gestors.

L'estàndard SQL incorpora la definició de tots els components del disseny lògic de la base de dades. En canvi, no conté cap element del disseny físic.

Per a transformar el disseny lògic de la base de dades en un SGBD concret, partim de les definicions de taules (amb tota la informació relacionada; és a dir, atributs, claus primàries, claus foranes i claus alternatives). A continuació es

relaciona cada element amb un espai adequat en el nivell virtual, i finalment, es relaciona cada espai virtual amb un fitxer físic, que constitueix el nivell físic del sistema d'informació.

2.5. Fase 5. Implementació i optimització

La darrera etapa és la implementació i l'optimització de la base de dades.

L'etapa d'implementació i optimització consisteix a realitzar la càrrega de les dades, i, posteriorment, ajustar alguns paràmetres relacionats amb el model físic de la base de dades per a optimitzar-ne el rendiment.

L'objectiu principal d'aquesta etapa és optimitzar el rendiment de la base de dades. En primer lloc, però, cal realitzar la càrrega de les dades, ja que no és possible optimitzar els accessos a les dades sense poder determinar la mida de les taules, els tipus d'accessos i consultes, la freqüència de les consultes i accessos, etc.

Finalment, també caldrà concretar els diferents rols d'usuaris i aplicacions per a poder determinar els permisos dels diferents grups. El component de gestió de la seguretat i les vistes permeten limitar els accessos i, d'aquesta manera, reduir el risc de problemes derivats d'accessos no autoritzats.

2.5.1. Processament i optimització de consultes

El llenguatge SQL emprat per les bases de dades relacionals és un llenguatge declaratiu. Això implica que s'especifica el resultat que es vol obtenir a partir d'una consulta realitzada, en comptes de determinar l'algorisme o el mètode que s'ha d'emprar per a obtenir el resultat. Per tant, és necessari entendre el conjunt de tasques que realitzen els SGBD relacionals per a obtenir la resposta que es vol.

Els SGBD relacionals avaluen sistemàticament les possibles estratègies alternatives que es poden presentar, amb l'objectiu d'escollir la que es considera òptima. El processament de consultes recull tot el conjunt d'activitats realitzades per l'SGBD que tenen com a objectiu l'extracció d'informació de la base de dades per tal d'assolir l'estratègia més eficient i proporcionar un més bon rendiment del sistema d'informació.

El processament de consultes es pot dividir en les quatre etapes principals següents:

1) **Descomposició de la consulta.** Traducció de la consulta expressada en llenguatge SQL a una representació interna basada en l'àlgebra relacional.

2) Optimització de la consulta. Procés de selecció del pla d'execució de la consulta més eficient entre les moltes estratègies generalment disponibles en el processament d'una consulta. L'optimització de consultes es pot realitzar sobre tres vessants:

a) Optimització semàntica basada en les restriccions especificades en l'esquema de la base de dades.

b) Optimització sintàctica, que consisteix a transformar heurísticament l'expressió relacional original en una altra d'equivalent però que sigui molt més eficient.

c) Optimització física, amb l'objectiu d'escollir entre els diversos plans d'avaluació, que poden tenir costos diferents, el que és més eficient. Per a determinar el cost més eficient cal conèixer el cost de cada operació, que sovint depèn de diferents paràmetres.

3) Generació de codi.

4) Execució de la consulta.

El procés d'optimització de consultes, és a dir, el tractament que dona un SGBD a les consultes fetes pels usuaris mitjançant SQL. Aquest punt és un dels més importants que s'ha de tenir en compte quan es dissenya un SGBD relacional, ja que l'opció triada afecta directament el rendiment global del sistema.

L'optimització de consultes és un aspecte molt important que cal considerar quan es dissenya i es construeix un SGBD relacional. Les tècniques que s'utilitzen per a optimitzar consultes condicionen el rendiment global del sistema, ja que determinen el temps que necessita el sistema gestor per a resoldre les consultes que han fet els usuaris.

2.5.2. Processament de vistes

Una vista és una taula lògica que permet accedir a la informació d'una o diverses taules per mitjà d'una consulta predefinida. No conté informació per si mateixa sinó que es basa en informació d'altres taules. Les vistes proporcionen mecanismes de seguretat i permeten al dissenyador de la base de dades personalitzar la vista que tenen els diferents usuaris de la base de dades.

Fer un ús adient de les vistes és un aspecte clau per a assolir un bon disseny d'una base de dades, ja que ens permet ocultar els detalls de les taules i mantenir la visió de l'usuari independentment de l'evolució que tingui l'estructura de taules.

Les vistes havien estat durant molt de temps un simple mecanisme de simplificació de consultes, però actualment tenen una importància cabdal en diverses àrees, com el disseny extern, els magatzems de dades⁷ o la informàtica distribuïda.

⁽⁷⁾En anglès, *data warehouse*.

2.5.3. Administració de la seguretat

Finalment, cal tenir en compte les tècniques que s'empren per a protegir la base de dades contra accessos no autoritzats i els mecanismes per a assignar i revocar privilegis als diferents usuaris. D'aquestes i d'altres accions, se n'encarrega el component de seguretat de l'SGBD. Aquest component esdevé cada dia més important, atès que actualment una gran quantitat d'ordinadors i altres tipus de dispositius estan interconnectats i, per tant, qualsevol persona podria esdevenir usuari, i un possible atacant, d'una base de dades.

En moltes organitzacions, la informació és un actiu intangible i de naturalesa sensible, que té un valor molt important. Per a preservar aquesta informació s'ha de protegir el sistema d'informació i saber les obligacions legals que cal complir.

Resum

En aquest mòdul hem vist, d'una manera molt general, el procés de disseny d'una base de dades.

Hem introduït, tot i que molt breument, les diverses etapes que formen el procés de disseny d'una base de dades. Hi entrarem en detall en la resta de mòduls de l'assignatura.

El procés de disseny d'una base de dades s'inicia amb la fase de recollida i anàlisi de requisits, que permet recollir i centralitzar les necessitats dels diferents grups d'usuaris i aplicacions. A partir d'aquesta anàlisi, en la segona fase, es modelitza un esquema conceptual que permet descriure el model de dades d'una manera independent de la tecnologia.

L'etapa següent en aquest procés és el disseny lògic, que requereix haver escollit prèviament el tipus de base de dades que es vol utilitzar en la implementació del sistema d'informació. El tipus de base de dades determina el model lògic que cal desenvolupar. Per exemple, en el cas d'utilitzar un tipus de base de dades relacional, es transforma el model conceptual en un model lògic específic per a bases de dades relacionals, anomenat *model relacional*.

El disseny físic permetrà adaptar el model lògic a un sistema gestor de bases de dades (SGBD) concret. Per tant, prèviament a aquest pas caldrà haver escollit l'SGBD específic que es vol utilitzar per a implementar el sistema d'informació. En aquesta etapa es crea l'estructura física que emmagatzemarà les dades de la base de dades.

Finalment, la darrera etapa permet l'optimització de la base de dades i la gestió de la seguretat relacionada amb els usuaris i les aplicacions de la base de dades. Lògicament, caldrà realitzar la càrrega de les dades prèviament, ja que la mida de les taules, els tipus de consultes i les freqüències d'aquestes són elements importants amb vista a optimitzar l'accés a les dades per part del sistema gestor.

Glossari

database management system *f* Vegeu **sistema gestor de bases de dades**. sigla **DBMS**

divide and conquer *loc* Vegeu **dividir per a vèncer**.

dividir per a vèncer *loc* Estratègia que proposa resoldre un problema complex per mitjà de la subdivisió en un conjunt de problemes més senzills en què la resolució dels diferents subproblemes comporta solucionar el problema inicial.
en divide and conquer

entity relation-ship model *m* Vegeu **model ER**.

llenguatge unificat de modelització *m* Llenguatge de propòsit general per a modelitzar sistemes de programari. L'estàndard va ser creat, i actualment és mantingut, pel Grup de Gestió d'Objectes.

en unified modeling language sigla **UML**

model ER *m* Model entitat-interrelació de dades d'alt nivell que permet modelitzar els requisits, les especificacions i les restriccions.

en entity-relationship model

SGBD *m* Vegeu **sistema gestor de bases de dades**.

sistema gestor de bases de dades *m* Tipus de programari específic que serveix d'interfície entre la base de dades, l'usuari i les aplicacions que la utilitzen.

en database management system sigla **SGBD**

UML *m* Vegeu **llenguatge unificat de modelització**.

Bibliografia

Elmasri, Ramez; Navathe, Shamkant, B. (2007). *Fundamentos de sistemas de bases de datos*. Madrid: Pearson Educación.

Ramakrishnan, Raghu; Gehrke, Johannes (2002). *Database management systems*. Boston: Mc-Graw-Hill Higher Education.