
UNITAT FORMATIVA 1 NUCLI FORMATIU 3

MODEL RELACIONAL

1.1. ESTRUCTURA DEL MODEL RELACIONAL**Introducció**

El model relacional va ser creat per E.F. Codd i distingia dos conceptes: *relation* i *relationship*. El primer fa referència a la relació com element constitutiu de la Teoria Relacional, que es correspon amb la idea de taula o entitat. El segon concepte s'aplica a la relació entre dos taules. A l'existir sempre relacions entre camps (o dominis), no pot distingir-se quan un camp és d'una taula o d'altra, per el que s'utilitzen indistintament ambdós conceptes, als que posteriorment s'afegeixen els conceptes de clau primària i aliena, de manera que siguin distingibles les taules i els camps que pertanyen a cadascuna d'elles.

El concepte central del model relacional que proposava Codd es basa en el següent: *L'eix del model relacional són les relacions que es donen entre els camps. Són els camps els que donen origen a les taules (relation) ja que existeixen relacions (relationship) entre elles. Les taules formen bases de dades (data bank).*

Codd defineix el seu model com relacional, utilitzant la paraula relació (taula) amb el seu significat matemàtic, utilitzant un array (matriu o taula) per a la seva representació. Les característiques definides per Codd per a una relació són:

- Cada fila representa una tupla de la taula
- No és significatiu l'ordre de les files
- Totes les files són diferents
- L'ordre de les columnes és significatiu, ja que es correspon amb l'ordre dels dominis en els que es defineix la taula.

El model relacional és un model on les dades s'estructuren lògicament en forma de relacions (taules), amb l'objectiu fonamental de mantenir la independència d'aquesta estructura lògica respecte a l'emmagatzemament i altres característiques del model físic.

Una base de dades relacional pot ser estructurada físicament de formes diferents, però la representació física ha de satisfer i representar, d'alguna forma, les relacions i restriccions lògiques de l'esquema relacional.

El model relacional proposa una representació de la informació de manera que:

- Els esquemes representin fidelment la informació
- Sigui entesa fàcilment pels usuaris
- Les possibles ampliacions de l'esquema de la base de dades no modifiquin l'estructura lògica existent (Independència lògica)
- Ha de poder oferir a cada usuari les dades de la forma més adequada a l'aplicació
- Els usuaris que accedeixin a les dades no han de modificar els seus programes per canvis a l'emmagatzemament físic (Independència física), és a dir, independència de les aplicacions i del dispositiu d'emmagatzemament (no importa si la base de dades creix poc o molt)
- Les estructures lògiques de les dades presenten un aspecte uniforme (taules) que faciliten la manipulació de les dades

Dins del model relacional podem distingir tres apartats:

1. Estructura de dades (formada pel domini i la taula)
2. Regles d'integritat (específiques, diferencial i de l'entitat)
3. Manipulació de dades (àlgebra relacional i càlcul relacional)

Una de les formes més fàcil de representar la informació és mitjançant una taula bidimensional on es representen els objectes i les relacions.

Les 12 regles de Codd

1: Informació, tota la informació de la base de dades ha d'estar representada explícitament al model lògic, es a dir totes les dades estan a les taules.

2: Accés garantit, tota dada es accessible coneixent el valor de la seva clau i el nom de la columna o atribut que la conté

3: Tractament sistemàtic dels valors nuls, el SGBD ha de permetre el tractament d'aquests valors

4: Catàleg en línia basat en el model relacional, les metadades han de ser sempre accessibles utilitzant un esquema relacional

5: Subllenguatge de dades complet, al menys ha d'existir un llenguatge que permeti el tractament complert de la base de dades. Aquest llenguatge ha de permetre realitzar qualsevol operació.

6: Actualització de vistes, el SGBD ha d'encarregar-se de que les vistes mostren les ultimes informacions.

7: Alt nivell d'inserció, actualització, i cancel·lació, el sistema ha de suportar subministrar dades en el mateix temps que s'insereixi, actualitzi o esborri. Això significa que les dades es poden recuperar d'una base de dades relacional en els sistemes de dades construïts de files múltiples i/o de taules múltiples

8: Independència física de les dades, les dades han de ser accessibles des de la lògica de la base de dades encara que es modifiqui l'emmagatzemament

9: Independència lògica de les dades, els programes no han de veure's afectats pels canvis a les taules

10: Independència d'integritat, les regles d'integritat han d'emmagatzemar-se a la base de dades (al diccionari de dades) no als programes d'aplicació.

11: Independència de la distribució, el subllenguatge de base de dades ha de permetre que les seves instruccions funcionen igualment en una base de dades distribuïda o en una que no ho és.

12: No subversió, si el SGBD té un llenguatge que permet el recorregut registre a registre aquest no pot utilitzar-se per a incomplir les regles de la base de dades

Estructura de les bases de dades relacionals

Les bases de dades relacionals posseeixen una estructura a tres nivells: *emmagatzemament*, *intern* i *conceptual*. El nivell extern i les seves característiques són implementades pels fabricants de cada base de dades.

A nivell d'emmagatzemament és l'administrador de la base de dades el responsable de definir les estructures d'emmagatzemament, fent servir tant les ordres del sistema operatiu com les del llenguatge de definició d'emmagatzemament de dades, que és un subconjunt del DDL.

A l'àmbit intern també és l'administrador de la base el responsable de definir les estructures lògiques de dades, utilitzant les ordres DDL necessàries.

A l'àmbit lògic o conceptual les diverses persones que es comuniquen amb la base de dades utilitzen el DML per seleccionar o emmagatzemar informació.

La implementació del model relacional en la base de dades la farem més endavant amb l'SQL.

Conceptes bàsics

A partir d'ara quan parlem de "relació" ens estarem referint al concepte que fins ara tenim de "taula".

■ Una **taula** és una estructura formada per files i columnes que serveixen per emmagatzemar dades d'una entitat.

■ Una **tupla** és cada registre que pertany a la relació. És a dir, cada fila és una tupla. És el conjunt de camps que formen la relació que es refereix a un mateix objecte. Una de les propietats de les tuples és que no hi poden haver dos registres amb tots els seus camps iguals, duplicats.

■ Un **atribut** és cadascuna de les columnes de la relació. Es corresponen amb els camps.

- El **grau** de la relació és el nombre d'atributs d'una relació. El grau és fix, no varia, a no ser que es modifiqui l'estructura de la relació. És el número de camps que posseeix una taula
- La **cardinalitat** de la relació és el nombre de tuples d'una relació. La cardinalitat no es fixa, sinó que pot variar afegint-hi o esborrant-ne tuples.
- **Vista** és una nova taula creada a partir d'altra o altres taules utilitzant les operacions relacionals entre les taules.
- El **domini** són tots els valors possibles que pot agafar un atribut.

La taula relacional té les següents característiques:

- Té només un tipus de fila
- Les files tenen totes les mateixes columnes
- La representació gràfica d'una taula es realitza mitjançant un conjunt de files i columnes. La primera fila conté el descriptor de cada columna. La resta de les files conté cadascun dels registres de la taula. El nom de la taula s'escriu, normalment, en la primera cel·la de la columna esquerra, deixant la resta d'eixes columnes sense contingut.
- Fins ara hem definit dos tipus de relacions: les pròpiament dites (taules) i les vistes. Quan de l'execució d'un conjunt d'ordres DML sobre varies relacions s'obté una vista i s'emmagatzema la seva estructura i dades, es diu que és una instantània (*snapshot*), ja que cada vegada que s'executen aquestes ordres s'obtindrà una vista diferent de les dades.

Les característiques de les columnes són:

- El número de columnes per taula és fixa i predeterminat, encara que pot modificar-se
- Cada columna ha de ser única i amb un nom específic en aquesta taula, és a dir, no poden haver columnes duplicades i el valor de cada columna ha de ser únic
- El valor de cada columna ha d'estar contingut en el domini que representa, encara que pot ser un valor nul.

Domini

- Un **domini** D és un conjunt finit de valors homogenis i atòmics (que podem anomenar V1, V2,..., Vn), caracteritzat per un nom. Diem que els valors són homogenis perquè són tots del mateix tipus, i atòmics perquè són indivisibles referits al model.

Per exemple, el domini de nacionalitats té els valors: espanyola, francesa, nordamericana, etc. que són tots del mateix tipus i que no es poden dividir sense perdre el seu significat (si descomposem "espanyola" en "e", "s",... es perd la seva semàntica, ja que les lletres per separat ja no tenen el significat que tenia "espanyola" com un valor de la nacionalitat).

Definició dels dominis:

Tot domini ha de tenir un nom, pel qual es puguin referir a ell, i un tipus de base de dades, a més d'unes restriccions a aquest tipus.

Aquestes restriccions poden ser:

- a) Sobre el rang del tipus (p.e. dins del conjunt dels positius, només agafar els parells)
- b) Agafar només un conjunt de valors (dins del conjunt dels habitants d'una ciutat, fer una enquesta als que tinguin entre 18 i 65 anys)
- c) Cridar a una funció per veure si el valor és possible

En algun moment es podria pensar que un domini és el mateix que una taula d'un grau (taula amb una columna), però això no és cert, ja que el domini conté tots els possibles valors que pot prendre un atribut i és estàtic (és a dir, aquests valors no canvien al llarg del temps, i si canvien, s'hauria de considerar un domini diferent), en canvi la taula és dinàmica per pròpia definició (ja que podem incloure sempre més tuples).

■ Un **atribut** A és el paper que representa un determinat domini D a una relació. Cada atribut només pot prendre valors d'un únic domini mentre que un domini pot tenir assignat tota una sèrie d'atributs.

Per exemple, suposem una taula *Client* amb els atributs *nom_client*, *DNI* i *adreça*, i una taula *Empleat* que inclou l'atribut *nom_empleat*. Llavors els atributs *nom_client* i *nom_empleat* poden tenir el mateix domini, el conjunt de tots els noms de persones.

Un domini *simple* és aquell en què els seus valors són valors atòmics.

Un domini *compost* es pot definir com una combinació de dominis simples al que podem aplicar certes restriccions d'integritat.

Per exemple, *Data*, compost pels dominis simples *Dia*, *Mes* i *Any*.

De la mateixa manera que definim dominis compostos també podem definir atributs compostos, que prendran els seus valors dels dominis compostos que tenen el mateix nom. Per exemple, atribut *Data* i domini *Data*.

Claus: conceptes i tipus

Quan un domini o combinació de dominis d'una relació posseeixen un conjunt de valors que identifiquen únicament a cada element de la relació, es denomina **clau primària**. La clau primària identifica unívocament a cada registre d'una taula. Les seves característiques són:

- És única i coneguda
- És no nul·la
- No es permeten claus primàries repetides o redundants
- Els camps que formen la clau primària reben el nom de primers.

Existeixen taules amb un o més camps o grups de camps que posseeixen característiques de clau primària. A aquests camps o grups de camps que poden originar la clau primària de la seva taula se'ls denominen **claus candidates**.

Les característiques de les claus candidates són:

- Al menys hi ha una per taula i està formada pel conjunt de tots els camps.
- Ha de tenir el mínim número possible de camps. Entre un i el total de camps de la taula.
- La clau candidata seleccionada es converteix en la clau primària. La resta de les claus candidates es denominen **claus alternatives**.

Quan un domini o combinació de dominis d'una taula posseeixen un conjunt de valors que no formen la clau primària en la taula, però són el conjunt de camps que formen la clau primària en altra taula se'ls denomina **clau aliena**.

Les claus alienes al compartir camps permeten relacionar taules, pel que faciliten la relació entre les mateixes, per originar noves taules o vistes. En quant a les característiques de les claus alienes, destaquen:

- El seu domini ha de ser el mateix que el domini de la clau primària de la taula a la que fa referència.
- Es permet que el nom dels camps sigui diferent de la clau aliena i que el nom del camp de la clau primària i el de la taula associada siguin diferents.
- El seu valor pot estar duplicat o ser nul.
- Els camps que la formen poden formar part de la clau primària de la taula a la que pertanyen.
- Una taula que no es relacioni amb altra no té clau aliena.

Esquemes de Relacions

Una relació(taula) es defineix sobre un conjunt de dominis $D_1...D_n$ i està formada per tres parts:

- Nom: Les relacions s'identifiquen normalment per un nom
- Una capçalera formada per un conjunt d'atributs $A_1...A_n$ on cada A_i està definit sobre el domini D_i i n és fix i representa el grau de la taula
- Un cos format per un conjunt variable de tuples. Una tupla ve definida per les parelles $(A_i:v_i)$ $i=1...m$, on A_i és un atribut, v_i pertany a D_i i m és la cardinalitat.

Denominem **esquema de la relació** a l'esquema constituït pel nom de la relació (si existeix) i la capçalera, i el denotem per $R(\{A_i:D_i\}_{i=1}^n)$.

Denominem **Relació**, al conjunt format per l'esquema i el cos de la relació, és a dir, una relació r sobre R és un subconjunt d'elements d' R format per m tuples, el grau de la qual és n i la cardinalitat m .

Així doncs, una base de dades relacional és una base de dades percebuda pels usuaris com una col·lecció d'esquemes de relació.

Per a anotar que un atribut A pertany a una taula s'utilitza $A_i(R(A))$

Tipus de relacions

- Persistents: Són aquelles relacions que la seva definició roman a la base de dades i s'esborra només amb l'acció explícita de l'usuari.
- Temporals: Són aquelles que desapareixen de la base de dades sense necessitat de que l'usuari l'esborri, per exemple a l'acabar una sessió.

Un esquema relacional es representarà mitjançant un graf, conegut com graf relacional. Es tracta d'un graf dirigit els nodes del qual són les relacions de la base de dades i els arcs representen les restriccions de clau aliena, i en el que apareixeran a més de les diferents relacions amb els seus atributs i les restriccions de clau aliena les restriccions de clau primària, unicatat i obligatorietat. Les convencions empleades per a la representació d'aquest graf són:

- El nom de les taules està representat en majúscules i en negreta. Primer apareix el nom de la taula i a continuació els seus atributs entre parèntesi.
- Les claus primàries apareixen subratllades
- Les claus alienes estan representades en lletra cursiva i referencien a la taula en la que són clau primària mitjançant una fletxa.
- Els atributs que poden agafar valors nuls apareixen amb un asterisc

1.2. REGLES D'INTEGRITAT: D'ENTITAT, CLAU I REFERENCIAL

L'objectiu de les regles d'integritat és comprovar la consistència de les dades de la base de dades.

Anomenem integritat la propietat de les dades de correspondre a representacions plausibles del món real.

Les operacions que poden afectar a la integritat són la inserció, la modificació i l'esborrat de registres de la base de dades.

Hem de distingir entre dos tipus de restriccions:

- **Restriccions del model o inherents**: són les obligades pel propi model.

Per una part, existeixen una sèrie de restriccions que es deriven de la pròpia definició de relació i són:

- **Regla d'integritat d'unicitat de la clau primària**: La regla d'integritat d'unicitat està relacionada amb la definició de clau primària. Concretament, estableix que tota clau primària que s'esculli per a una relació no ha de tenir valors repetits.
- **Regla d'integritat d'entitat de la clau primària**: La regla d'integritat d'entitat de la clau primària disposa que els atributs de la clau primària d'una relació no poden tenir valors nuls o desconeguts.
- **Regla d'integritat referencial**: La regla d'integritat referencial està relacionada amb el concepte de clau forana. Concretament, determina que tots els valors que pren una clau forana han de ser valors nuls o valors que existeixen a la clau primària a la que fa referència.

La necessitat de la regla d'integritat referencial prové del fet que les claus foranes tenen per objectiu establir una connexió amb la clau primària que referencien. Si un valor d'una clau forana no fos present a la clau primària corresponent, representaria una referència o una connexió incorrecta.

Així doncs, donades dues relacions R1 i R2, amb K1 com a clau primària de R1 i K2 com a clau externa a R2 que fem coincidir amb K1, llavors cada valor de K2 ha de ser igual a algun valor de K1 o bé ser nul. Això vol dir que per a tota clau externa diferent del valor nul existeix una tupla a la taula referenciada on la clau primària és igual al valor de la clau externa.

- **Regla d'integritat de domini**: La regla d'integritat de domini està relacionada, com el nom ho indica, amb la noció de domini. Aquesta regla estableix dues condicions.

La **primera condició** és que un valor no nul d'un atribut A_i ha de pertànyer al domini de l'atribut A_i, és a dir, ha de pertànyer a domini(A_i).

La **segona condició** serveix per a establir que els operadors que és possible aplicar sobre els valors depenen dels dominis d'aquests valors. És a dir, un operador determinat només es pot aplicar sobre valors que tinguin dominis adequats per a aquell operador.

- **Restriccions d'usuari o semàntiques**: són condicions específiques d'una base de dades concreta, és a dir, són les que s'han de complir en una base de dades particular amb uns usuaris concrets, però que no són necessàriament rellevants en una altra base de dades.

Així, en la pràctica, els mecanismes que proporciona el model relacional per recollir les restriccions semàntiques o d'usuari són:

- a) La restricció de **clau primària** (PRIMARY KEY) que permet declarar un atribut o conjunt d'atributs com la clau primària d'una relació.
- b) La restricció d'**unicitat** (UNIQUE) ens permet definir claus alternatives (els valors d'un o varis atributs no poden repetir-se en diferents tuples d'una relació).

- c) La restricció d'**obligatorietat** (NOT NULL) ens permet declarar si un o varis atributs d'una relació han d'agafar sempre un valor, és a dir, no poden agafar valors nuls.
- d) La restricció de **clau aliena** (FOREIGN KEY), també denominada d'integritat referencial, s'utilitza per a, mitjançant claus alienes (conjunt d'atributs en una relació que és una clau primària en altra o la mateixa relació) enllaçar relacions d'una base de dades. La integritat referencial ens indica que els valors de la clau aliena en la relació fill han de correspondre amb els valors de la clau primària en la relació pare o bé ser nuls si admeten nuls. Els atributs que són clau aliena en una relació no necessiten tenir els mateixos noms que els atributs de la clau primària amb la qual ells es corresponen.

A més de la integritat referencial que ens permet enllaçar relacions entre sí donant lloc a l'estructura de la base de dades, el model relacional permet també definir les opcions d'esborrat i modificació en les claus alienes. Aquestes opcions indiquen les accions que hi ha que portar enlloc quan es produeix un esborrat o modificació d'una tupla en la relació pare referenciada per una relació filla. Les possibilitats per a una operació d'actualització (esborrar o modificació) són:

- *Esborrat/modificació en cascada* (CASCADE): L'esborrat (o modificació) d'una tupla en la relació pare ocasiona un esborrat (o modificació) de totes les tuples en la relació filla (tuples la clau aliena de les quals coincideixi amb el valor de la clau primària de la tupla eliminada o modificada en la relació pare).
 - *Esborrat/modificació restringit* (RESTRICT): En aquest cas si existeixin tuples en la relació filla relacionades amb la tupla de la relació pare sobre la que es realitza l'operació, aleshores no es podria esborrar un departament que tingui empleats que treballen en ell.
 - *Esborrat/modificació amb posada a nuls* (SET NULL): Aquesta possibilitat ens permet posar el valor de la clau referenciada a NULL quan es produeix l'esborrat o modificació d'una tupla en la relació pare.
 - *Esborrat/modificació amb posada a un valor per defecte* (SET DEFAULT): el seu funcionament és semblant al cas anterior, amb l'excepció de que el valor al que es posen les claus alienes referenciades és un valor per defecte que s'haurà especificat en la definició de la taula corresponent.
- e) Restriccions de **verificació** (CHECK): En alguns casos pot ocórrer que sigui necessari especificar una condició que han de complir els valors de determinats atributs d'una relació de la base de dades apart de les restriccions ja vistes de clau primària, unicatat, obligatorietat i clau aliena. Per això es defineixen les restriccions de verificació que sempre porten implícites un rebuig en cas de que no s'acompleixi la condició especificada i que també es comproven davant una inserció, esborrat o modificació. Com les restriccions de verificació sempre van lligades a un únic element de la base de dades no és necessari que tinguin un nom.
 - f) **Assercions** (ASSERTION): Altre tipus de restricció, que generalitza a l'anterior, el formaran les assercions en les que la condició s'estableix sobre elements de diferents relacions (per això ha de tenir un nom que la identifiqui). El seu funcionament és idèntic al de les restriccions de verificació, tenen com àmbit una única relació.
 - g) **Disparadors** (TRIGGER): A vegades pot interessar especificar una acció diferent del rebuig quan no s'acompleix una determinada restricció semàntica. L'ús dels disparadors ens permet indicar una condició i especificar l'acció que volem que es porti enlloc si la condició és vertadera. Són regles esdeveniment-condició-acció (ECA) que poden interpretar-se com regles que especifiquen que quan es produeix un esdeveniment, si s'acompleix una condició, aleshores es realitza una determinada acció.

1.3. TRADUCCIÓ DEL MODEL ENTITAT-RELACIÓ AL MODEL RELACIONAL

Preparació del model entitat-relació

- *Eliminació dels atributs múltiples*

Tots els atributs múltiples, és a dir, els atributs que poden agafar més d'un valor del seu domini, s'han de transformar en entitats febles. Aquesta entitat contindrà l'atribut múltiple.

A més, s'haurà de tenir en compte que si l'atribut no fos clau primària es procedirà d'una de les formes següents:

- a) L'entitat feble heretarà la clau de l'entitat o relació que tenien els atributs múltiples.
- b) S'afegirà un nou atribut que permeti identificar de manera única les tuples d'aquesta entitat.

- *Eliminació dels atributs compostos*

Tots els atributs compostos han de ser descompostos en atributs simples.

Transformació del model entitat-relació

Transformació d'entitats

Totes les entitats es transformen en taules mantenint el número i tipus d'atributs, així com la seva clau. El tipus de nomenclatura que farem servir serà: el nom de la taula i entre parèntesi tots els seus atributs subratllant els que formin part de la clau.

• **Transformació de les relacions binàries**

Transformacions de relacions 1 a 1

La transformació es pot fer de tres maneres:

a) Les dues entitats participen totalment (1,1) i (1,1) o les dues entitats participen parcialment (0,1), (0,1) i s'accepten valors nuls en la clau aliena:

- No fer cap taula per la relació
 - o Es fa una taula per cada entitat
 - o Els atributs clau d'una d'elles indistintament passen a la taula de l'altra com a clau externa
 - o Si la relació té atributs, passen a la taula que assumeix la clau externa.

b) Una de les taules participa totalment i altra parcialment (1,1) i (0,1)

- No fer cap taula per la relació
 - o Es fa una taula per cada entitat
 - o Els atributs clau de la taula que participa parcialment passen a la taula que participa totalment
 - o Si la relació té atributs, passen a la taula que assumeix la clau externa.

c) Les dues entitats participen parcialment (0,1), (0,1) i no s'accepten valors nuls en la clau aliena:

- Es fa una taula per la relació
 - o Una taula per cada entitat
 - o La taula de la relació té com atributs els atributs clau de les entitats. La clau serà qualsevol d'aquests atributs, i junt amb l'altre seran claus alternatives.
 - o Si la relació té atributs, es posen a la nova taula.

Transformacions de 1 a N

- a) Les dues entitats participen totalment (1,1) i (1,n) o les dues entitats participen parcialment (0,1) i (0,n) o alguna de elles participa parcialment (1,1) i (0,n) o (0,1) i (1,n) i a tots els casos s'accepten valors nuls en la clau aliena.
- Cada entitat es transforma en una taula
 - L'atribut clau de l'entitat amb cardinalitat màxima molts passa a la taula amb cardinalitat màxima 1, i es transforma en clau externa.
 - Si la relació té atributs, es posen a la taula amb cardinalitat màxima 1
- b) Les dues entitats participen parcialment: (0,1) i (0,n) o l'entitat amb cardinalitat màxima 1 participa parcialment (0,1) i (1,n) i no s'accepten en cap cas valors nuls a la clau aliena:
- Cada entitat es transforma en una taula
 - Es crea una taula per la relació, formada pels atributs clau de les entitats, a més dels atributs de la relació.
 - La clau serà l'atribut clau de l'entitat que participa amb cardinalitat màxima 1

Transformació de M a N

- Cada entitat es transforma en una taula
- Es crea una taula per la relació, formada pels atributs clau de les entitats i els atributs de la relació. La clau estarà formada pels atributs clau de les entitats (clau composta)
- Poden existir casos en els que algun atribut de la relació hagi de formar part de la clau primària

- **Transformació de les relacions ternàries**

Transformació M:N:P

Quan la connectivitat de la relació és M:N:P, la taula que s'obté de la seva transformació té com a clau primària tots els atributs que formen les claus primàries de les tres entitats relacionades.

Transformació M:N:1

Quan la connectivitat de la relació és M:N:1, la taula que s'obté de la seva transformació té com a clau primària tots els atributs que formen les claus primàries de les dues entitats dels costats de la relació etiquetada amb M i amb N.

Transformació N:1:1

Quan la connectivitat de la relació és N:1:1, la taula que s'obté de la seva transformació té com a clau primària els atributs que formen la clau primària de l'entitat del costat N i els atributs que formen la clau primària de qualsevol de les dues entitats que estan connectades amb 1.

Transformació 1:1:1

Quan la connectivitat de la relació és 1:1:1, la taula que s'obté de la seva transformació té com a clau primària els atributs que formen la clau primària de dues entitats qualsevol de les tres interrelacionades.

- **Transformació de les relacions recursives**

Si una relació recursiva té connectivitat 1:1 o 1:N, dóna lloc a una clau forana o nova taula depenent dels casos anteriors i, si té connectivitat M:N, origina una nova taula.

- **Transformació d'entitats dèbils**

Les entitats dèbils es tradueixen al model relacional igual que la resta d'entitats, amb una petita diferència. Aquestes entitats sempre estan al costat N d'una relació 1:N que completa la seva identificació.

Així, doncs, la clau forana originada per aquesta relació 1:N ha de formar part de la clau primària de la relació corresponent a l'entitat dèbil.

- **Transformació de la generalització/especialització**

Cadascuna de les entitats superclasse i subclasse que formen part d'una generalització/especialització es transforma en una taula:

- a) La taula de l'entitat superclasse té com a clau primària la clau entre l'entitat superclasse i conté tots els atributs comuns
- b) Les taules de les entitats subclasse tenen com a clau primària, i aliena a la superclasse, la clau de l'entitat superclasse i contenen els atributs específics de la subclasse

1.4. NORMALITZACIÓ

Introducció

Fins ara hem vist la forma de expressar gràficament les dades d'una base de dades mitjançant el diagrama entitat-relació corresponent i, posteriorment, traslladar aquest esquema al model relacional.

Així i tot, la pròpia transformació al model relacional presenta una sèrie de problemes que fins ara no havíem tingut en compte, per exemple:

- *Informació redundant.* Consisteix en la repetició innecessària de dades. La redundància no necessària dóna origen a anomalies en la base de dades. Aquestes anomalies són de tres tipus:

Anomalies d'inserció. No es pot insertar un registre si no es coneix el valor dels atributs primaris.

Anomalies d'actualització. Si s'esborra un registre es perd tota la informació de tots els seus atributs.

Anomalies d'esborrat. A l'eliminar un registre hi ha que modificar una gran quantitat d'ells, ja que pot haver molta redundància.

- *Ambigüetat.* Es produeix quan en dos taules s'utilitzen els mateixos noms dels camps per referir-se a conceptes diferents o, a l'inrevés, es donen diferents noms a atributs que contenen la mateixa informació.

Molts d'aquests problemes es poden eliminar si s'utilitza correctament una metodologia que permeti obtenir el model lògic de dades. Encara utilitzant unes regles determinades es possible que hi apareixen problemes derivats de les restriccions imposades entre atributs de diferents entitats o entre atributs de la mateixa entitat, això és el que, a continuació veurem com "dependències".

Dependència funcional

Com a primera idea, podríem dir que la dependència és un conjunt de restriccions que s'imposen a determinats atributs de les taules.

Així, el disseny d'esquemes es basa en el concepte de dependència de dades, és a dir, la capacitat de determinar el valor d'un atribut coneixent altre o altres.

Formalment, podríem dir que, en una relació R amb atributs A_1, \dots, A_n , X determina funcionalment a Y , o també, Y depèn funcionalment d' X ($X \rightarrow Y$) si per a tota ocurrència d' R , el valor d' X determina unívocament el valor d' Y o, dit d'altra manera, no és possible que existeixen dues ocurrències d' R que tinguin el mateix valor per a X i diferent valor per a Y .

X i Y són subconjunts d' A_1, \dots, A_n són subconjunts d' A_1, \dots, A_n (atributs o conjunts d'atributs). Al conjunt d'atributs de la part esquerra de la dependència se la denomina determinant.

Per a determinar les dependències funcionals d'una base de dades és necessari conèixer el significat dels seus atributs, la seva funció i les regles de funcionament de la base.

Normalització

El procés de normalització es basa en aconseguir la independència de les dades respecte a les aplicacions que els fan servir, obtenint taules amb una estructura òptima i eficaç, és a dir, consisteix en seguir uns passos per a obtenir dades agrupades en relacions de manera que la seva implementació en el nivell físic sigui òptima.

Hi ha cinc nivells de normalització, sent cada vegada més complexe el procés d'obtenció de taules normalitzades. Per a bases de dades relativament senzilles es pot deixar de normalitzar una vegada aconseguida la tercera forma normal (3FN).

El procés de normalització es basa en la descomposició sense pèrdua de les taules que estan en una forma normal inferior, obtenint-se una forma normal superior. Descomposició sense pèrdua significa que s'ha de dividir o descomposar la taula en altres amb menor quantitat d'atributs, sense que hi hagi pèrdua d'informació.

Primera Forma Normal

Una taula està en **Primera Forma Normal (1FN)** si el valor que conté cada atribut és únic i elemental. En cadascun dels atributs només es pot incloure una dada encara que sigui composta, però no es pot incloure una llista de dades.

Segona forma normal

Direm que un atribut o conjunt d'atributs té dependència funcional d'altre o altres si a cadascun dels primers li correspon només un dels segons.

Una taula està en **Segona Forma Normal (2FN)** quan està en 1FN i no hi ha cap atribut que sense pertànyer a la clau primària no té una dependència funcional total d'ella.

Si la clau primària està formada per només un atribut i està en 1FN, ja està en 2FN.

Per transformar una taula amb dependències funcionals, la clau de la qual estigui formada per més d'un camp, en una taula en 2FN es necessitarà crear taules noves per eliminar les dependències funcionals. Les noves taules tindran atributs que dependran funcionalment de la clau i els que formen la part de la clau de la que depenent. Una vegada creades les noves taules, s'eliminaràn de la taula primera els atributs que tenien dependències funcionals.

Tercera forma normal

Es diu que hi ha dependència funcional transitiva entre dos atributs quan un atribut que no pertany a la clau primària permet conèixer el valor d'altre atribut.

Una taula està en **Tercera Forma Normal (3FN)** si està en 2FN i no existeixen atributs que no pertanyen a la clau primària que puguin ser coneguts mitjançant altre atribut que no formi part de la clau primària, és a dir, no hi ha dependències funcionals transitives.

Quan hi han dependències funcionals transitives, es crea una nova taula amb els atributs que tenen dependència funcional transitiva, eliminant-se l'atribut dependent de la taula original. Si en la taula creada hi ha algun tipus de dependència funcional es descomposa fins eliminar aquesta dependència.

Forma normal de Boyce-Codd

Una taula està en Forma Normal de Boyce-Codd (FNBC) si només existeixen dependències funcionals elementals que depenen de la clau primària o de qualsevol clau alternativa. Si la clau primària està formada per només un atribut i està en 3FN, ja està en FNBC.

1.5. LLENGUATGES RELACIONALS

Encara que la formulació sintàctica d'expressions mitjançant l'àlgebra o càlcul relacional permet crear les consultes que es desitgen, no són un interfaç fàcil per als usuaris, de manera que els sistemes comercials, sense apartar-se gaire de la formalitat de les expressions, permeten escriure expressions en llenguatge natural.

Originàriament eren llenguatges de consulta, als que se'ls afegien noves funcionalitats: modificar dades, estructures, seguretat, etc. Destaquen els següents llenguatges: SQL (*Structured Query Language*) que utilitzen àlgebra relacional, QUEL (*Query Language*) que utilitza càlcul relacional orientat a tuples i QBE (*Query By Example*) que utilitza càlcul relacional orientat a dominis.

Com hem dit, el llenguatge més important que fa ús de l'àlgebra relacional és l'SQL. Tots els grans tipus de bases de dades relacionals l'utilitzen amb petites variants depenent de l'empresa propietària del sistema gestor de la base de dades. Entre els sistemes que el fan servir tenim l'ORACLE, INFORMIX, ACCESS,...

És important destacar també, que en l'actualitat hi ha versions de quasi tots els llenguatges de programació que poden treballar amb bases de dades relacionals fent servir l'àlgebra relacional, ja sigui amb SQL o amb el seu propi motor. Alguns d'aquests llenguatges serien, per exemple, Visual C/C++, Visual Basic,...