

RetroDisseny de Bases de Dades Relacionals.

Óscar Sellés Giner
ETIS

Anna Queralt Calafat

Data Lliurament 19/06/2006.

No hagués arribat fins aquí sense l'esforç de moltes persones.

Gràcies Susanna per la paciència infinita que has mostrat els darrers anys (ja falta poc).

Gràcies Marina per ser capaç de tirar endavant amb un pare a temps parcial.

Gràcies Joan per la màgia.

Resum.

El plantejament inicial d'aquest projecte és el de aconseguir obtenir l'esquema conceptual originari de qualsevol base de dades relacional per a fer tasques de reenginyeria.

Es pretén, a més, dotar al diagrama ER a obtenir d'extensions emprant llenguatge de definició de restriccions (OCL), per la qual cosa emprarem les llibreries de funcions Dresden OCL.

Per fer-ho, hem realitzat un estudi previ dels mecanismes de transformació entre esquemes emprats en el procés de disseny d'una base de dades relacional. Així, analitzant els mètodes que porten de l'esquema conceptual (diagrama ER), a l'esquema lògic i, d'aquest, a l'esquema físic, esperàvem trobar possibles fonts d'indeterminacions, i les seves possibles solucions, quan apliquem el procés invers per arribar a la representació original.

Finalment, mitjançant la definició d'uns mecanismes de transformació estrictes entre esquemes, s'han aconseguit eliminar els punts que podien provocar indeterminacions en el procés d'enginyeria inversa de base de dades.

En el desenvolupament de la present memòria farem esment de la necessitat de salvaguardar, tant com sigui possible, la semàntica de la realitat que pretén descriure una modelització d'aquesta mitjançant un diagrama ER i les seves successives transformacions fins al model físic.

Començant en el món de les idees, arribarem al dels aspectes físics de representació d'informació i, a partir d'aquest, farem el viatge de tornada fins als conceptes originals.

Paraules clau: Enginyeria inversa, base de dades relacional, rediseño, model ER.

Índex de continguts.

Resum.....	2
Índex de continguts.....	3
Índex de figures.....	4
Introducció.....	5
Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC. .	5
Objectius del TFC.....	6
Enfocament i mètode seguit.....	6
Planificació del projecte.....	7
Base de treball.....	9
Mòdul 1. Connexió a la base de dades i obtenció de les estructures a analitzar.....	9
Introducció.....	9
Requeriments.....	10
Requeriments d'informació.....	10
Requeriments d'independència.....	16
Solució proposada.....	19
Mòdul 2. Redisseny de la base de dades.....	22
Transformació del model ER fins al model físic.....	23
Transformació del model ER al model relacional.....	23
Transformació del model relacional al model físic.....	23
Transformació d'entitats.....	23
• Transformació d'entitats regulars.....	24
• Transformació d'entitats dèbils.....	24
• Transformació de generalitzacions / especialitzacions.....	25
• Transformació d'entitats associatives.....	27
Transformació d'interrelacions.....	29
• Transformació d'interrelacions binàries.....	29
• connectivitat 1:1.....	29
• connectivitat 1:N.....	32
• connectivitat M:N.....	33
• Transformació d'interrelacions n-àries.....	33
• Interrelacions recursives.....	37
Quadre Resum.....	38
Redisseny.....	39
Entitats Regulars.....	39
• Entitat Regular Referenciada.....	39
• Entitat Regular Referent.....	40
Entitat Dèbil.....	40
Generalització / Especialització.....	40
Interrelació 1:1.....	41
Interrelació 1:N.....	41
Interrelació M:N i n-àries amb totes les connectivitats N.....	41
Interrelació n-ària amb no totes les connectivitats N.....	42
Entitat associativa.....	42
Interrelació recursiva.....	43
Solució proposada.....	43
Mètodes i proves lògiques.....	44
Integració dels mòduls.....	46
Manual d'usuari.....	48
Conclusions i què és el que ha quedat per fer.....	52
Valoració personal.....	52
Glossari.....	53
Bibliografia.....	55
Annexos.....	¡Error! Marcador no definido.

Índex de figures.

Càrrega de dades. Diagrama de classes.....	20
Rediseño. Diagrama de classes.....	25
Diagrama de classes complet.....	47
Transformació d'esquemes. 1. Entitat regular.....	24
Transformació d'esquemes. 2 Relació 1:N.....	25
Transformació d'esquemes. 3. Categorització.....	26
Transformació d'esquemes. 4 Entitat Associativa.....	27
Transformació d'esquemes. 5 Relació 1:1.....	29
Transformació d'esquemes. 6 Relació 1:1 no obligatòria.....	30
Transformació d'esquemes. 7 Relació 1:N.....	32
Transformació d'esquemes. 8 Relació M:N.....	33
Transformació d'esquemes. 9 Relació n-ària.....	34
Transformació d'esquemes. 10 Relació n-ària.....	35
Transformació d'esquemes. 11 Relació n-ària.....	36
Transformació d'esquemes. 12 Relació recursiva.....	37

Introducció.

Les tecnologies de la informació fa temps que són una de les moltes eines de que disposa qualsevol organització en general i les empreses en particular per optimitzar els seus processos, adaptant-los d'una forma adient al seu negoci.

L'aplicació d'aquestes eines suposa una diferència competitiva vers aquelles entitats que no les utilitzen. Igualment, l'ús d'eines i sistemes amb un rendiment millorat implica una eficiència més gran per el correcte desenvolupament dels processos que modelitzen.

Aquesta política de millora de la eficàcia dels sistemes d'informació es pot aplicar a qualsevol dels àmbits on hi són presents. Anomenem reenginyeria a aquests procediments d'adaptació de sistemes vells a sistemes nous més eficients.

La millora dels procediments o dels sistemes d'un determinat ens, requereix de la disponibilitat d'una documentació descriptiva i comprensible de les estructures de dades que són el seu nucli. La falta de documentació de sistemes heretats o amb un desenvolupament pobre fa que sigui necessari obtenir la informació necessària per tal de poder migrar cap a millors sistemes.

Podem establir, doncs, que per que el procés de reenginyeria tingui èxit cal tenir suficient informació sobre allò que fan i que són els sistemes actualment en funcionament.

És en resposta a aquesta necessitat que es presenta l'anomenada enginyeria inversa. Aquesta disciplina no té per finalitat la modificació de cap dels aspectes d'un determinat sistema, aquest tipus de feina queda reservat per als processos de reenginyeria. El que veritablement necessitem és obtenir la base sobre la que treballarem per obtenir un millor producte. Aplicant aquesta tècnica obtindrem una descripció del sistema amb un grau d'abstracció suficient per tal de que pugui aplicar-se a qualsevol plataforma.

Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.

Per al present TFC es centrarem exclusivament en el món de les bases de dades relacionals. Aquest model de base de dades és un dels més emprats en els sistemes d'informació informàtics, i presenta unes característiques que el fan idoni per a l'aplicació dels processos d'enginyeria inversa.

El nostre punt de partida serà qualsevol base dades relacional. A partir d'aquesta serem capaços d'obtenir un diagrama entitat - relació que descrigui amb un nivell d'abstracció superior l'estructura de dades inherent a la base. Amb aquesta

informació ja estarem preparats per millorar l'eficàcia de la gestió de les dades migrant l'estructura cap a un altre sistema més potent.

Objectius del TFC.

L'objectiu del TFC és obtenir una aplicació capaç d'obtenir, a partir de qualsevol base de dades relacional, l'esquema entitat – relació que la modelitza a un nivell superior d'abstracció.

Per a la representació de la metainformació obtinguda emprarem el llenguatge *XMI*, definit per l'*OMG*. Aquesta informació emmagatzemada en un fitxer podrà ser representada en el format gràfic més habitual per aquest tipus d'esquemes amb l'eina *Poseidon* de la empresa *Gentleware*.

Enfocament i mètode seguit.

Per el desenvolupament del programa ens basarem en el mètode clàssic de desenvolupament de software. Amb el model en cascada partirem dels requeriments que ja hem descrit breument i que ampliarem abastament més endavant. Realitzarem les corresponents fases d'anàlisi, disseny i codificació per obtenir el producte final.

Determinades funcionalitats de l'aplicació es recolzaran en productes de tercers, emprarem l'anomenat *Dresden OCL Toolkit* i més concretament el conjunt de llibreries Java que aporta per a la representació d'esquemes UML i restriccions OCL.

Ara bé, aquest projecte porta aparellada una dosi suficient d'investigació com per que l'aplicació estricta del model esmentat sigui contraproductiu. Així doncs, optem per dividir les fases d'anàlisi i disseny del projecte en subprojectes més petits i més fàcils de manegar, dels quals presentem una llista de tasques a realitzar a títol orientatiu.

- **Connexió a base de dades** i obtenció del catàleg de les taules i de les restriccions. Per a la realització d'aquesta fase ens caldrà analitzar quines estructures de memòria són les més adients per emmagatzemar la informació a tractar.
- **Processament de les dades obtingudes.** Caldrà analitzar la informació obtinguda per esbrinar els mecanismes que ens permetran abstraure l'esquema entitat – relació des de les metadades del model analitzat. *En la mesura del possible seria desitjable formalitzar aquells aspectes de la traducció on sigui adient aplicar aquesta tècnica matemàtica.* Aquesta fase d'anàlisi serà el que esdevindrà el "cor" de la nostra aplicació.

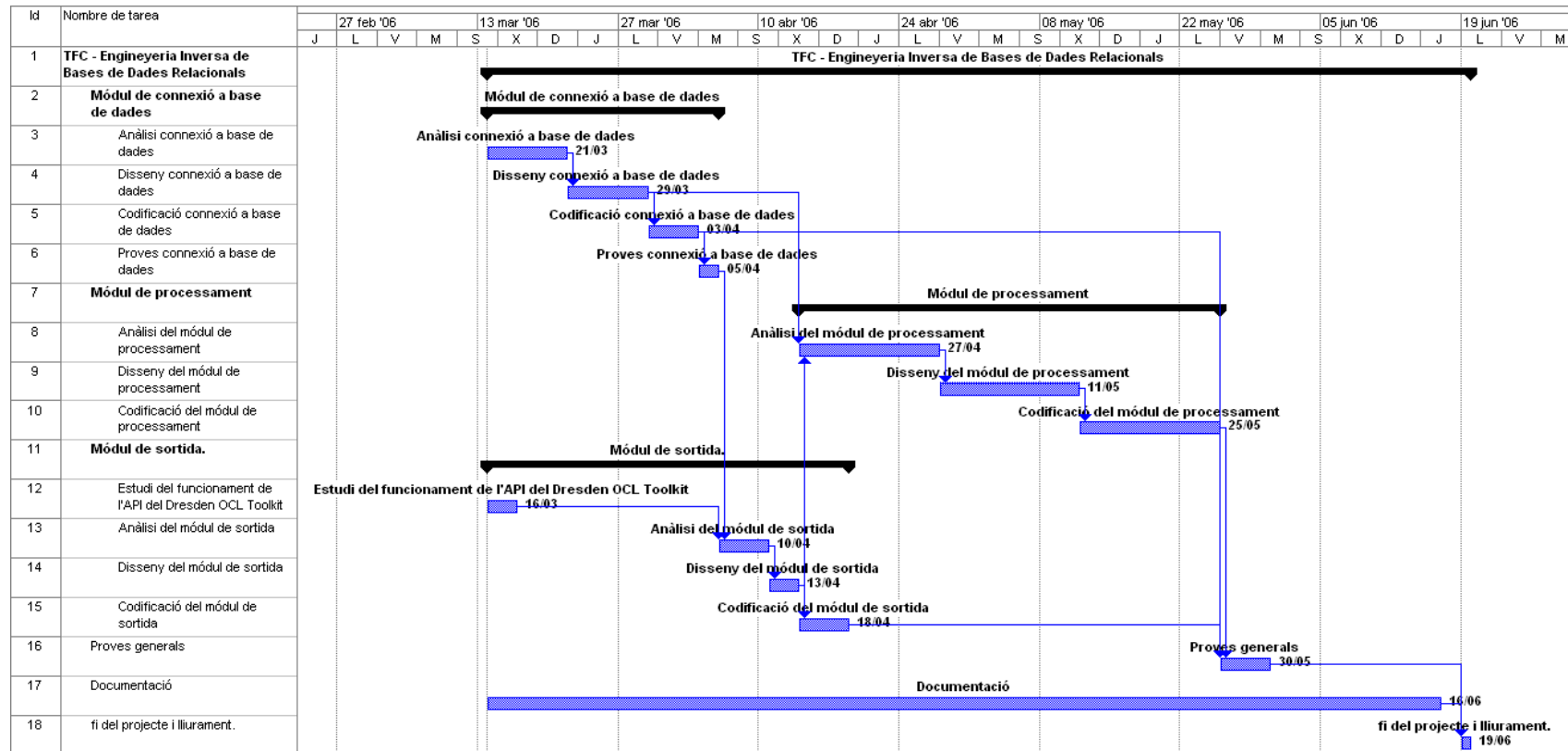
-
- **Producció d'un fitxer de sortida.** L'anàlisi de l'API de Dresden OCL Toolkit que emprarem per la descàrrega del metamodel obtingut serà necessari en aquest punt per tal de produir un fitxer de sortida amb el metamodel preparat.
 - **Codificació.** La codificació es realitzarà a mida que s'obtinguin dissenys raonablement fiables dels subprojectes esmentats.
 - **Proves.** Al igual que la fase de codificació es dissenyaran conjunts de proves parcials per subprojecte i totals per tot el producte a desenvolupar.
 - **Documentació.** Anirà realitzant-se a mida que es desenvolupi el projecte.

Planificació del projecte.

Abordem en aquest punt un dels aspectes més difícils de definir per un estudiant quan s'enfronta amb aquest tipus de projecte.

Intuïtivament, podríem planificar aquest projecte vers les fases tradicionals de desenvolupament de software. Així tindríem una fita que ens marqués el final de la fase d'anàlisi, una altra per l'acabament de la fase de disseny i una final que contemplés la finalització del codi amb les subfites adients per les tasques de documentació i proves de l'aplicació.

Però com ja hem esmentat al punt anterior, *Enfocament i Mètode Seguit*, aplicarem una divisió en subprojectes de la feina a fer. A més ens trobem al davant d'un cas on la planificació pot resultar força alterada per graus de dificultat incorrectament valorats (per excés o per defecte), per la qual cosa la planificació en aquest moment és merament orientativa.



Les dades proposades a la planificació oficial de l'assignatura que es troben al campus virtual gairebé coincideixen amb la finalització de les codificacions dels mòduls de connexió i de sortida (22/04) i la del mòdul de processament (29/05) per la qual cosa es proposa mantenir les dates fixades per lliurar el treball planificat.

Cal indicar que s'ha deixat un coixí de temps destinat a esmoreir qualsevol desviació de la planificació. El projecte està planificat per obtenir un producte funcional el proper 30/05.

Base de treball.

Per a la realització d'aquest projecte ens cal una base pràctica per a la realització de les proves. Així doncs, s'ha triat el sistema gestor de base de dades relacionals de l'empresa MySQL AB. Aquest producte compleix els requeriments de ésser un SGBDR suficientment estàndard com per a que les conclusions i les eines que desenvolupem siguin fàcilment transportables a altres SGBDR.

El projecte es realitzarà amb llenguatge Java, emprant el paradigma de la programació orientada a objectes.

Per a la connexió amb la base de dades des de Java, emprarem JDBC, amb els connectors proveïts pel mateix fabricant de MySQL.

Mòdul 1. Connexió a la base de dades i obtenció de les estructures a analitzar.

Introducció.

El model entitat – relació és un model conceptual de dades d'alt nivell emprat sovint en el disseny de bases de dades. Aquest model va ésser introduït al 1976 per Peter Chen. Permet la representació d'estructures d'informació que són abstracció de la realitat que es vol descriure.

L'objectiu principal d'aquest model és la creació d'un esquema, que sigui independent del Sistema Gestor de Base de Dades que s'emprarà per a la seva implementació, a partir de les especificacions dels requeriments de la realitat que es pretén descriure.

Per a la seva aplicació, distingim tres fases en aquest model:

<ul style="list-style-type: none">• Disseny conceptual	Primer pas del procés de disseny, a partir d'una realitat a representar, amb l'ajut del model conceptual, obtindrem l'anomenat esquema conceptual.
<ul style="list-style-type: none">• Disseny lògic	A partir de l'esquema conceptual obtingut a la fase anterior i aplicant les regles del model de dades propi del Sistema Gestor de Base de Dades a emprar obtindrem l'anomenat esquema lògic.
<ul style="list-style-type: none">• Disseny físic	Finalment, prenent com a base l'esquema lògic obtingut a la fase de disseny lògic, procedim a implementar la base de dades sobre el SGBD, amb l'objectiu d'aconseguir la màxima eficiència de cara a la màquina i al problema concret que estem tractant.

El procés per obtenir un esquema a partir de l'anterior es basa en una sèrie de regles de derivació que permeten transportar el model des de un estat fins al següent.

No ens estendrem, de moment, amb una explicació detallada de les característiques del model entitat – relació, ni de les regles a aplicar per a la transformació d'esquemes. Tant sols indicarem que el camí que tenim que fer és justament el contrari a les fases habituals de disseny. A partir del disseny físic, que és el que podem observar quan ens connectem a un base de dades d'un SGBD concret, ens caldrà obtenir l'esquema conceptual primigeni que donà origen al sistema.

Per a la realització del projecte ens centrarem en resoldre casos concrets de bases de dades relacionals que anirem implementant en el SGBDR triat.

Requeriments.

Els requeriments per al mòdul de càrrega de dades són de dos tipus, requeriments d'informació i requeriments d'independència del SGBDR que dona suport a la base dades que estem redissenyant.

Requeriments d'informació.

En el primer mòdul a realitzar ens ocupem de la càrrega de la informació necessària per realitzar el procés de "redisseny". Una anàlisi informal del problema, que es realitzarà amb més rigor en el següent mòdul, ens dona una idea aproximada del que necessitem:

- Llista de taules que componen la base de dades a invertir. Cada taula representarà a una entitat o a una relació en el diagrama final.
- Llista de camps que componen cadascuna de les taules de la base de dades, amb indicació de les seves característiques bàsiques (tipus de dada i si el camp admet valors nuls), aquests camps corresponen als atributs dels diferents tipus d'entitats.
- Llista dels atributs que componen la restricció de clau primària de cadascuna de les taules.
- Llista de les restriccions claus foranes que apliquen a cadascuna de les taules, amb indicació de les taules i, dins d'aquestes a les columnes, a les que es referència.
- Llista de les restriccions d'unicitat que es correspondran amb claus alternatives d'una determinada taula.

Tota la informació requerida pot ser obtinguda a partir de les taules descriptives de l'estructura interna del SGBDR. Tots els sistemes presenten aquesta mena de estructures, "catàlegs de base de dades", per descriure les diferents bases de dades que poden albergar.

Així, en el producte que hem pres com referència, MySQL 5, la informació s'emmagatzema en una base de dades interna anomenada "*information schema*".

Seguidament presentem la sentència SQL adient per a l'obtenció de cadascun dels conjunts d'informació necessari per al nostre desenvolupament. La sintaxi emprada és la corresponent a la versió del SGBDR emprat que, recordem, és MySQL 5.

- Obtenció de la llista de taules que componen la base de dades.

```
SELECT table_name FROM tables
WHERE table_schema = "biblioteca"
```

Cal destacar que el valor introduït per la condició de selecció *WHERE table_schema="biblioteca"* correspon a la base de dades creada per a la realització de les proves.

El resultat obtingut en executar aquesta consulta sobre el catàleg de MySQL és:

```
mysql> SELECT table_name FROM tables
-> WHERE table_schema="biblioteca";
+-----+
| table_name |
+-----+
| autor      |
| consta     |
| editorial  |
| escriu     |
| exemplar  |
| institucio |
| llibre     |
| prestec   |
| soci       |
| tema       |
| tracta     |
| treballa  |
+-----+
12 rows in set (0.00 sec)
```

Així doncs, la nostra base de dades, està formada per 12 taules. Cadascuna d'elles pot correspondre a diferents realitats, ja siguin entitats, relacions El procés d'anàlisi ens donarà la resposta.

- Obtenció de les columnes, i de les seves característiques, que componen cadascuna de les taules que hem trobat mitjançant la consulta anterior.

```
SELECT column_name, data_type, is_nullable
FROM   columns
WHERE  table_schema="biblioteca"
AND    table_name=nomTaula
```

Tindrem que repetir aquesta consulta per cadascuna de les taules que hem trobat en el punt anterior. D'aquesta forma obtindrem la llista de columnes que formen cada taula.

A la estructura de la taula **columns** del catàleg de MySQL existeixen altres camps que poden aportar informació útil per a la feina que tenim encomanada, però, de moment i a falta d'una anàlisi més seriosa no els inclourem a la consulta.

El resultat obtingut en executar aquesta consulta per a la taula *llibre* és:

```
mysql> SELECT column_name, data_type, is_nullable
-> FROM columns
-> WHERE table_schema="biblioteca"
-> AND table_name="llibre";
+-----+-----+-----+
| column_name | data_type | is_nullable |
+-----+-----+-----+
| id_Llibre   | int       | NO          |
| ISBN_Llibre | varchar   | NO          |
| titol_Llibre | varchar   | NO          |
| idioma      | varchar   | NO          |
| any         | int       | NO          |
| num_Copies  | int       | NO          |
| editorial   | int       | NO          |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

Obtindrem resultats semblants a mida que anem repetint la consulta variant-hi el nom de la taula de la que volem obtenir la llista de columnes.

- Obtenció de la llista dels atributs que componen la clau primària de les taules obtingudes a la consulta anterior. Aquesta consulta caldrà aplicar-la a cadascuna d'elles.

```
SELECT kcu.column_name
FROM   table_constraints tc, key_column_usage kcu
WHERE  tc.table_schema="biblioteca"
AND    kcu.table_schema = "biblioteca"
AND    tc.table_name= nomDeTaula
AND    tc.constraint_type="PRIMARY KEY"
AND    tc.constraint_name=kcu.constraint_name
AND    tc.table_name=kcu.table_name
ORDER BY kcu.ordinal_position;
```

MySQL, emmagatzema aquesta informació repartida en dos taules. Per una banda a la taula *table_constraints* es guarden les restriccions que s'han definit, amb indicació del nom de restricció, nom de taula on és d'aplicació i tipus de restricció (PRIMARY KEY, FOREIGN KEY...). A més es guarda la informació sobre els atributs que formen part de la clau primària a la taula *key_column_usage*.

Així obtindrem la llista de les columnes que componen la clau primària per cadascuna de les taules de la base de dades. A més ho obtindrem amb l'ordre en que hi són definides a la clau.

El resultat d'aplicar la consulta definida anteriorment a la taula *llibre* és:

```
mysql> SELECT kcu.column_name
-> FROM table_constraints tc, key_column_usage kcu
-> WHERE tc.table_schema="biblioteca"
-> AND kcu.table_schema="biblioteca"
-> AND tc.table_name="llibre"
-> AND tc.constraint_type="PRIMARY KEY"
-> AND tc.table_name=kcu.table_name
-> AND tc.constraint_name=kcu.constraint_name
-> ORDER BY kcu.ordinal_position;
+-----+
| column_name |
+-----+
| id_Llibre   |
+-----+
1 row in set (0.00 sec)
```

- Obtenció de la llista de atributs que són relacionats mitjançant clau forana amb altres camps d'altres taules. Aquesta consulta caldrà aplicar-la a cadascuna de les taules de la base de dades.

Per obtenir aquesta informació en el nostre cas pràctic ho farem en dos etapes, primer cal que esbrinem el nom de les restriccions de tipus "FOREIGN KEY" que estan relacionades amb cadascuna de les taules de la base de dades que analitzem.

```
SELECT constraint_name
FROM   table_constraints
WHERE  table_schema="biblioteca"
AND    table_name= nomDeTaula
AND    constraint_type="FOREIGN KEY";
```

Si apliquem la consulta per obtenir les claus foranes de la taula *llibre* obtenim:

```
mysql> select constraint_name
-> from table_constraints
-> where table_schema="biblioteca"
-> and table_name="llibre"
-> and constraint_type="FOREIGN KEY";
+-----+
| constraint_name |
+-----+
| FK_llibre_1    |
+-----+
1 row in set (0.00 sec)
```

Amb aquesta consulta obtenim la llista de claus foranes que s'apliquen a una determinada taula, ara ens cal esbrinar, per cadascuna de les claus foranes trobades, la seva composició, és a dir, el nom de columna que referència, i el nom de la taula i columnes referenciades.

```
SELECT column_name, referenced_table_name, referenced_column_name
FROM   key_column_usage
WHERE  table_schema="biblioteca"
AND    table_name= nomDeTaula
AND    constraint_name= nomDeClauForana;
```

Al igual que amb les claus primàries la informació que te relació amb la definició de les claus foranes es reparteix en les dues taules que ja hem esmentat en l'anàlisi del cas de les claus primàries. La diferència rau en el fet de que el valor per el camp *constraint_type* de la taula *table_constraints* passa a ser **FOREIGN KEY** i que els camps que indiquen la taula i columna referenciada contenen informació.

Si apliquem la consulta per a determinar la composició de la clau forane de la taula *llibre*, obtenim:

```
mysql> select column_name, ordinal_position,
-> referenced_table_name
-> referenced_column_name
-> FROM key_column_usage
-> WHERE table_schema="biblioteca"
-> AND table_name="llibre"
-> AND constraint_name="FK_llibre_1";
+-----+-----+-----+
| column_name | ordinal_position | referenced_column_name |
+-----+-----+-----+
| editorial   | 1                | editorial               |
+-----+-----+-----+
1 row in set (0.03 sec)
```

Així doncs, obtindrem la llista de claus foranes emprades a cadascuna de les taules, a més , per cada una d'elles obtindrem la columna de la taula que referència i la taula i columna a la que fan referència.

Les dades obtingudes seran emmagatzemades en objectes de classes adients per al seu tractament posterior.

Podem resumir aquest apartat d'obtenció d'informació amb el resultat i la informació que volem obtenir de cadascuna de les consultes a realitzar al catàleg del SGBDR:

Consulta	Descripció	Dades
1	Obtenció del nom de les taules.	- table_name
2	Obtenció del nom de les columnes de cadascuna de les taules	- column_name - data_type - is_nullable
3	Obtenció de les columnes que participen de la clau primària de cadascuna de les taules.	- column_name.
4	Obtenció de les columnes que participen de les claus foranes definides a cadascuna de les taules, a més de la taula i columna referenciada.	- Constraint_name - Column_name - Referenced_table_name - Referenced_column_name
5	Obtenció de les restriccions d'unicitat que puguin estar definides a cadascuna de les taules. Al igual que en el punt d'obtenció de claus foranes, emprarem idèntic mètode en dues etapes per obtenir el total de restriccions d'aquest tipus que apliquen a una determinada taula i els seus components.	- Constraint_name - Column_name.

Els requeriments d'informació que hem definit són independents del SGBDR que els emmagatzema, el programa funcionarà igualment si definim altres consultes per a la obtenció de la informació requerida. Sempre que es respecti l'ordre de les columnes en que es presenten els resultats.

Requeriments d'independència.

- Independència de les consultes.

Com ja s'ha indicat en el punt anterior "*Requeriments d'Informació*", obtindrem aquesta a partir d'unes consultes sql contra el catàleg del SGBDR on resideix la base de dades a analitzar.

Aquestes consultes i, en particular, la forma per referir el catàleg del SGBDR, poden variar d'un sistema a un altre. Una possible solució a la diversitat de consultes a plantejar (set per cada SGBDR) és la de incloure-les directament en el codi Java de l'aplicació. Aquesta solució planteja greus inconvenients a l'hora de definir un nou medi d'accés a un SGBDR no contemplat inicialment.

És per aquesta raó, que hem escollit extreure el cos de les consultes del codi de l'aplicació. Les consultes estaran guardades en un únic fitxer de configuració. Les consultes així guardades tenen que complir una sèrie de condicionants que els faran vàlids per a l'aplicació:

- Han de ser consultes SQL amb una estructura vàlida per al SGBDR al qual van destinades.
- Cal substituir les referències a la base de dades per l'etiqueta #DATABASE#.
- Cal substituir les referències al nom de taula per l'etiqueta #TABLENAME#.
- Cal substituir les referències al nom de la clau forana per l'etiqueta #FKNAME#.

Aquests fitxers seran llegits per l'aplicació per construir internament, en temps d'execució, les consultes a realitzar contra el catàleg del SGBDR. Necessàriament caldrà adequar la consulta a realitzar al que ens interessi en cada moment, per aconseguir-ho, s'han substituït en el codi de la consulta les referències al nom de la base de dades i al nom de la taula que consultem en cada moment per etiquetes en la forma #DATABASE# i #TABLENAME# que el programa s'encarregarà de substituir per les cadenes adients en cada moment.

Cal remarcar que les consultes presentades, ho són a tall d'exemple i tant sols són d'aplicació per a un SGBDR MySQL 5, si la base de dades objecte d'estudi residís en un altre sistema, caldria definir noves consultes en nous

fitxers, respectant els condicionants expressats en el format dels fitxers i l'ordre de les columnes en els resultats de cadascuna de les consultes.

- Independència de la connexió.

El programa que estem realitzant cal que sigui capaç de carregar la informació requerida amb independència del SGBDR que emprem per donar suport a la base de dades objecte del procés de redisseny.

Per la connexió de Java amb el SGBDR emprem JDBC. Els fabricants de SGBDR proporcionen biblioteques de funcions per tal d'establir aquesta mena de connexions.

El mecanisme de connexió des de Java és el següent:

1. Es registra el driver JDBC mitjançant la sentència:

```
Class.forName("driver.JDBC").newInstance();
```

On *driver.JDBC* correspon a la cadena de declaració del driver JDBC, en el cas de MySQL, la cadena és "*com.mysql.jdbc.Driver*".

2. Una vegada registrat el driver, procedim a crear una connexió contra la base de dades, per fer-ho ens cal disposar d'un usuari amb suficient nivell d'accés com per poder consultar el catàleg i el seu mot de pas. Per establir la connexió emprem la sentència:

```
Con=DriverManager.getConnection("jdbc:sgbdr://host/baseDeDades","usuari","motDePas")
```

On "*con*" és la instància d'un objecte de tipus *Connection* que creem per comunicar amb el SGBDR, "*sgbdr*" és l'identificador del tipus de base de dades al qual ens estem connectant, "*baseDeDades*" és el nom de la base de dades que conté el catàleg del SGBDR, "*usuari*" és el nom de l'usuari amb suficient nivell d'accés com per poder consultar el catàleg del SGBDR i "*motDePas*" és la paraula de pas corresponent a "*usuari*".

3. Un cop ha estat creat l'objecte de tipus "*Connection*", creem sobre ell un nou objecte, de tipus "*Statement*" aquest cop, que és el que ens permetrà realitzar consultes sobre la base de dades que hem indicat en el segon pas de la connexió.

Per la creació de l'objecte de tipus "*Statement*" no necessitem cap informació externa al programa, al contrari del que passava en les dos primeres passes del procés de connexió i consulta. És a dir, necessitem conèixer, per a l'establiment de la connexió, el driver JDBC a emprar, la cadena de connexió (composada per el tipus de SGBDR, el host on resideix la base de dades i el nom de la base de dades que conté el catàleg del

SGBDR), l'usuari amb suficients drets i el seu password. Ho presentem tot en forma de taula:

Informació necessària per a l'establiment de la connexió.			
Paràmetre	Component	Descripció	Cas pràctic
Declaració Driver	-	Cadena necessària per la declaració del driver JDBC a emprar.	com.mysql.jdbc.Driver
Cadena de connexió.	Tipus SGBDR	Correspon al tipus de SGBDR que emprem on resideix la base de dades que analitzem.	Mysql
	Host	Host on resideix el SGBDR.	Localhost
	BasedeDades	Nom de la base de dades corresponent al catàleg del SGBDR	Information_schema
	Usuari	Nom d'un usuari amb suficient nivell d'accés per consultar el catàleg del SGBDR	Root
	motDePas	Password per l'usuari.	

Així doncs, i amb l'objectiu d'establir un mecanisme que permeti la independència, proposem el pas d'aquests literals, necessaris per l'establiment de connexió, mitjançant una combinació de fitxers de configuració i paràmetres a la línia d'instruccions.

El fitxer de paràmetres proposat segueix la estructura de "*nom d'atribut = valor atribut*", aquest fitxer es passarà com un paràmetre més en el moment de executar el programa. L'aplicació s'encarregarà d'obrir el fitxer i de separar adientment la part que identifica el nom de l'atribut i el seu valor, emprant la definició del fitxer per poder establir la connexió. A més, el fitxer s'aprofitarà per incloure-hi en diferents seccions la definició de les consultes SQL que hem definit al punt de *Requeriments d'Informació*.

El fitxer de configuració per a l'accés a un SGBDR MySQL és:

```
MySQL.txt
[general]
driver=com.mysql.jdbc.Driver
connection=jdbc:mysql://localhost/information_schema

[tableQuery]
SELECT table_name
FROM tables
WHERE table_schema='#DATABASE#';

[columnQuery]
SELECT column_name, data_type, is_nullable
FROM columns
WHERE table_schema='#DATABASE#'
AND table_name='#TABLENAME#';

[PKQuery]
SELECT kcu.column_name, ordinal_position
FROM table_constraints tc, key_column_usage kcu
WHERE tc.table_schema='#DATABASE#'
AND kcu.table_schema='#DATABASE#'
AND tc.table_name='#TABLENAME#'
AND tc.constraint_type='PRIMARY KEY'
AND tc.constraint_name=kcu.constraint_name
AND tc.table_name=kcu.table_name
ORDER BY kcu.ordinal_position;

[FKQuery]
SELECT constraint_name
FROM table_constraints
WHERE table_schema='#DATABASE#'
AND table_name='#TABLENAME#'
AND constraint_type='FOREIGN KEY';

[FKQuery2]
SELECT column_name, ordinal_position, referenced_table_name,
referenced_column_name
FROM key_column_usage
WHERE table_schema='#DATABASE#'
AND table_name='#TABLENAME#'
AND constraint_name='#FKNAME#';

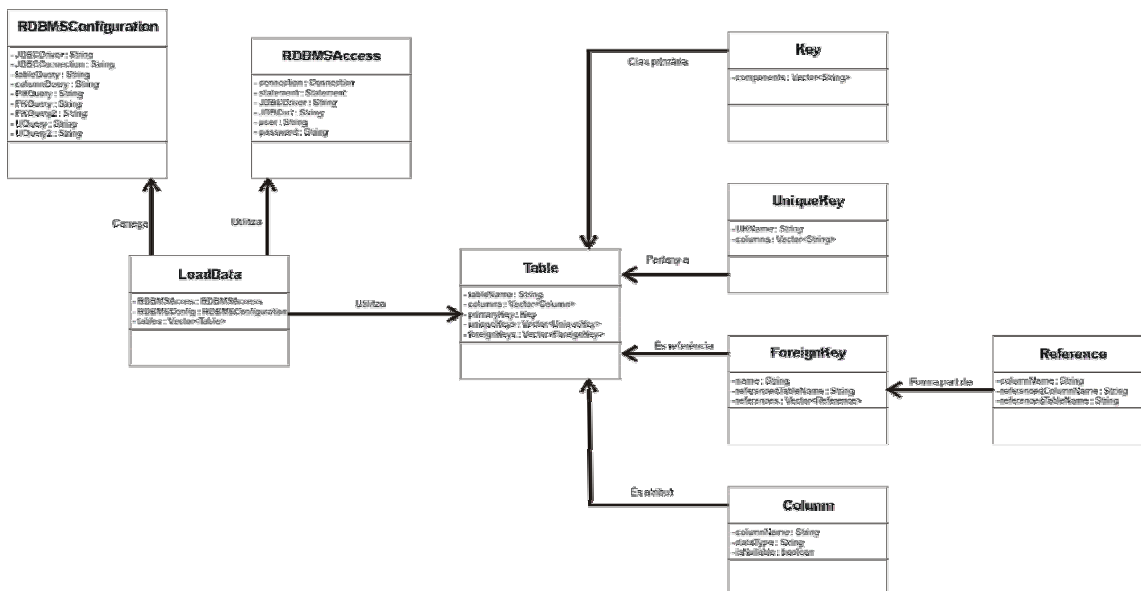
[UQuery]
SELECT constraint_name
FROM table_constraints
WHERE table_schema='#DATABASE#'
AND table_name='#TABLENAME#'
AND constraint_type='UNIQUE';

[UQuery2]
SELECT column_name, ordinal_position, referenced_table_name,
referenced_column_name
FROM key_column_usage
WHERE table_schema='#DATABASE#'
AND table_name='#TABLENAME#'
AND constraint_name='#UKNAME#';
```

Solució proposada.

Per al mòdul de càrrega de dades, proposem per una banda la definició de classes que emmagatzemin la informació necessària per els processos de connexió i

obtenció de les dades necessàries i per altra les suficients classes que representin els diferents tipus d'objectes que modelitzen la informació de l'estructura de la base de dades que analitzem.



1. Càrrega de dades. Diagrama de classes.

Classe LoadData. És la classe encarregada de realitzar la càrrega inicial de dades i de emmagatzemar la informació obtinguda. Per obtenir la informació crea una instància de la classe RDBMSConfiguration i un altra de RDBMSAccess, la qual cosa li permet, emprant els mètodes adients, executar les consultes SQL per l'obtenció de la informació. Aquesta és guardada en un objecte de tipus Table, que té una estructura complexa, integrant-hi objectes de les classes Key (clau primària), UniqueKey, ForeignKey i Column.

Classe RDBMSConfiguration. És la classe encarregada de llegir el fitxer de configuració que conté la informació necessària per la declaració del driver JDBC i l'establiment de la connexió amb la base de dades. A més, llegeix les plantilles de consultes SQL definides al fitxer de configuració i les serveix a demanda mitjançant la invocació dels seus mètodes. És utilitzada per la classe LoadData.

Classe RDBMSAccess. És la classe encarregada de realitzar les consultes sobre el SGBDR. És utilitzada per la classe LoadData.

Classe Table. És la classe on anirem guardant la informació obtinguda. Per emmagatzemar-la cal que anem creant objectes de les classes que formen l'estructura interna de la classe Table. Utilitza les classes Column, Key, UniqueKey i ForeignKey, és emprada per la classe LoadData.

Classe Column. És la modelització de la informació necessària per descriure un camp de una taula en concret. Guarda el nom del camp, el tipus de dades associat

i un booleà que informa sobre si el camp admet nuls o no. És emprada per la classe Table.

Classe Key. És la classe que guarda informació, en un vector de Strings, sobre la composició de la clau primària de la taula. És emprada per la classe Table.

Classe UniqueKey. Guarda informació sobre els camps que componen una restricció d'unicitat a la taula. És emprada per la classe Table.

Classe ForeignKey. Guarda informació sobre una restricció de clau forana definida a la taula. Utilitza la classe Reference i és emprada per la classe Table.

Classe Reference. Guarda informació sobre els components d'una determinada clau forana. És emprada per la classe ForeignKey.

Es pot trobar una descripció més amplia sobre els mètodes i atributs de les classes del mòdul de càrrega de dades a la documentació de l'aplicació generada amb JavaDoc que acompanya a aquesta memòria.

Mòdul 2. Redisseny de la base de dades.

Quan dissenyem una base de dades, ho fem perquè volem reflectir en un sistema d'informació una determinada realitat. Aquesta realitat presenta el que podríem anomenar "semàntica del cas", originalment aquesta semàntica existeix tant sols en el pensament dels usuaris del sistema. És feina del dissenyador el que el contingut d'aquesta semàntica no es perdi al aplicar-li transformacions per arribar a l'esquema físic final. És més, la centralització dins del SGBDR de la descripció d'aquesta semàntica, permet definir-la un sol cop, evitant d'aquesta forma la necessitat d'afegir controls a les aplicacions que accedeixen a la informació.

El procés en tres etapes del disseny i implementació d'una base de dades, esquema conceptual, esquema lògic, esquema físic, segueix unes determinades regles de transformació d'un esquema en el següent. L'aplicació d'aquestes normes ens permet mantenir, dins del possible, la semàntica del cas.

Cal ara esbrinar els mètodes i proves lògiques que ens serviran per determinar el diagrama entitat – interrelació original que, una vegada transportat al seu esquema lògic i a l'esquema físic, va donar com a resultat la base de dades que estem analitzant.

Per fer-ho, recordem que hem recol·lectat informació sobre les taules de la base de dades, els seus atributs, les característiques d'aquests darrers i les restriccions definides a cadascuna de les taules de tipus clau primària, clau forana i clau única. Amb aquesta informació hem d'ésser capaços d'esbrinar molts dels elements del diagrama original.

La informació obtinguda correspon a la darrera etapa del procés de creació de la base de dades, la transformació del esquema lògic al físic per aplicar-lo al SGBDR que actua com a hoste de la base de dades. El procés de transformació de l'esquema conceptual a l'esquema lògic i, d'aquest, a l'esquema físic, segueix unes certes regles i normes, com ja hem esmentat, que suposarem que s'han aplicat d'una forma raonablement correcta. Aquesta suposició serà el nostre punt de partida en el procés de redisseny.

Per a identificar les proves adients, farem un breu repàs dels mètodes de transformació fins a l'esquema físic, provant d'extreure conclusions sobre les característiques de la transformació física final que puguin aplicar-se com generalitzacions a qualsevol taula. Així, de l'admissió de valors nuls en una clau forana podrem determinar la obligatorietat o la opcionalitat d'una entitat a una interrelació, de la presència de claus foranes com a integrants de claus primàries esbrinarem la presència d'interrelacions entre entitats.

Transformació del model ER fins al model físic.

Transformació del model ER al model relacional.

Els elements bàsics del model ER són les entitats i les interrelacions.

- Les entitats quan es tradueixen al model relacional, originen relacions.
- Les interrelacions, en canvi, quan es transformen, poden donar lloc a claus foranes d'alguna relació ja obtinguda o poden donar lloc a una nova relació.

En el cas de les interrelacions, cal tenir en compte el seu grau i la seva connectivitat per poder decidir quina és la transformació adequada.

- Les interrelacions binàries 1:1 i 1:N donen lloc a claus foranes.
- Les interrelacions binàries M:N i totes les n-àries es tradueixen en noves relacions.

Transformació del model relacional al model físic.

Els mètodes per realitzar aquesta transformació poden variar notablement d'un SGBDR a un altre, per la qual cosa ens centrarem en aspectes comuns a tots ells, com són la definició de restriccions (de tipus clau primària, clau forana i clau única) i les propietats inherents als camps (admet valors nuls o no). És aquesta informació la que manté un mínim de la semàntica original del cas i és on incidirem per obtenir resultats.

Transformació d'entitats.

Com que la transformació fins al nivell físic, passa per dos etapes, model relacional i model físic, farem esment de les característiques físiques quan sigui imprescindible, en la resta de casos entendrem que quan parlem de relacions o atributs en el model relacional podem assumir que podríem parlar igualment de taules o camps en el model físic.

Cada entitat en el model ER es transforma en una relació en el model relacional i en una taula en el model físic. Els atributs de l'entitat seran atributs de la relació i, igualment, la clau primària de l'entitat serà la clau primària de la relació. Anàlogament la transformació al nivell físic donarà una taula per cadascuna de les relacions que estarà composta pels camps corresponents als atributs de la relació originària.

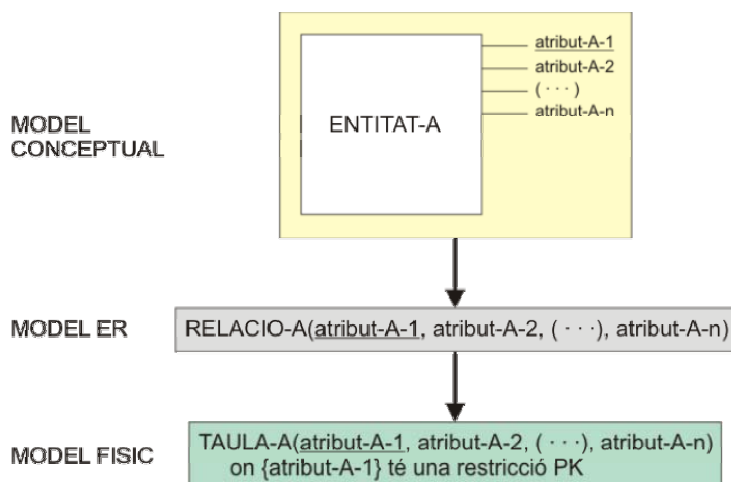
Cal fer notar que, si una entitat intervé en alguna interrelació binària 1:1 o 1:N, pot ser necessari afegir nous atributs a la relació obtinguda a partir de l'entitat.

Aquests atributs formaran claus foranes de la relació en el model relacional i restriccions de clau forana en el nivell físic. A més, la obligatorietat o no de la participació d'una determinada entitat a la interrelació (a nivell conceptual), quedarà reflectida en el model físic amb l'admissió o no de valors nuls als camps.

A més, cal considerar els dos tipus d'entitats que podem definir, entitats regulars (entitats que representen a un tipus de cosa que es pot identificar clarament) i entitats dèbils (aquelles entitats la existència o identificació de les quals depèn d'un altra entitat).

- *Transformació d'entitats regulars.*

Així, si considerem el cas d'una entitat regular, ignorant les interrelacions a les que participi:



Transformació d'esquemes. 1. Entitat regular

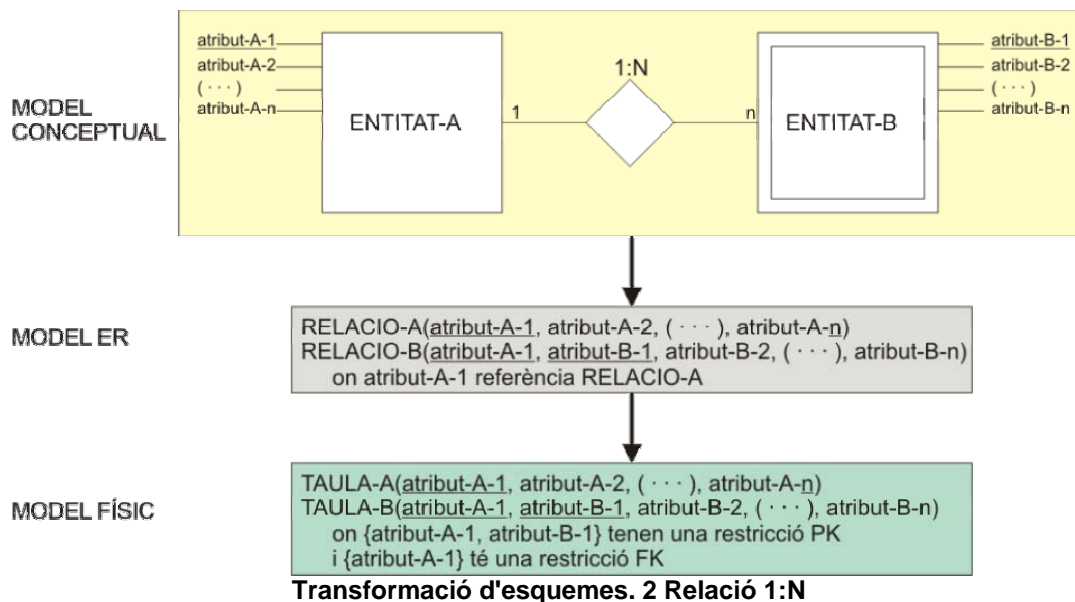
On podem observar que la transformació correspon amb el que hem exposat. Les entitats en un diagrama entitat – relació participen en interrelacions. El cas que acabem de descriure, correspondria a una entitat que es referenciada per altres relacions.

A nivell físic, en la definició de la taula serà necessari indicar quin conjunt de camps (un o més d'un) identifica unívocament cada tupla de la taula, creant-hi una restricció de tipus "clau primària".

- *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 interrelació 1:N que completa la seva identificació.

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



El procés de transformació fa que la relació en que es transforma l'entitat dèbil presenti un atribut afegit als atributs definits en el diagrama. En el cas de que la clau primària de l'entitat principal fos formada per més d'un atribut, caldria afegir el mateix nombre de nous atributs a la relació dèbil. Aquest conjunt d'atributs afegits (un o més d'un) fa referència a l'entitat de la qual depèn l'entitat dèbil en forma de clau forana. L'atribut afegit més un conjunt d'atributs propis de l'entitat dèbil formen la clau primària de la relació resultant.

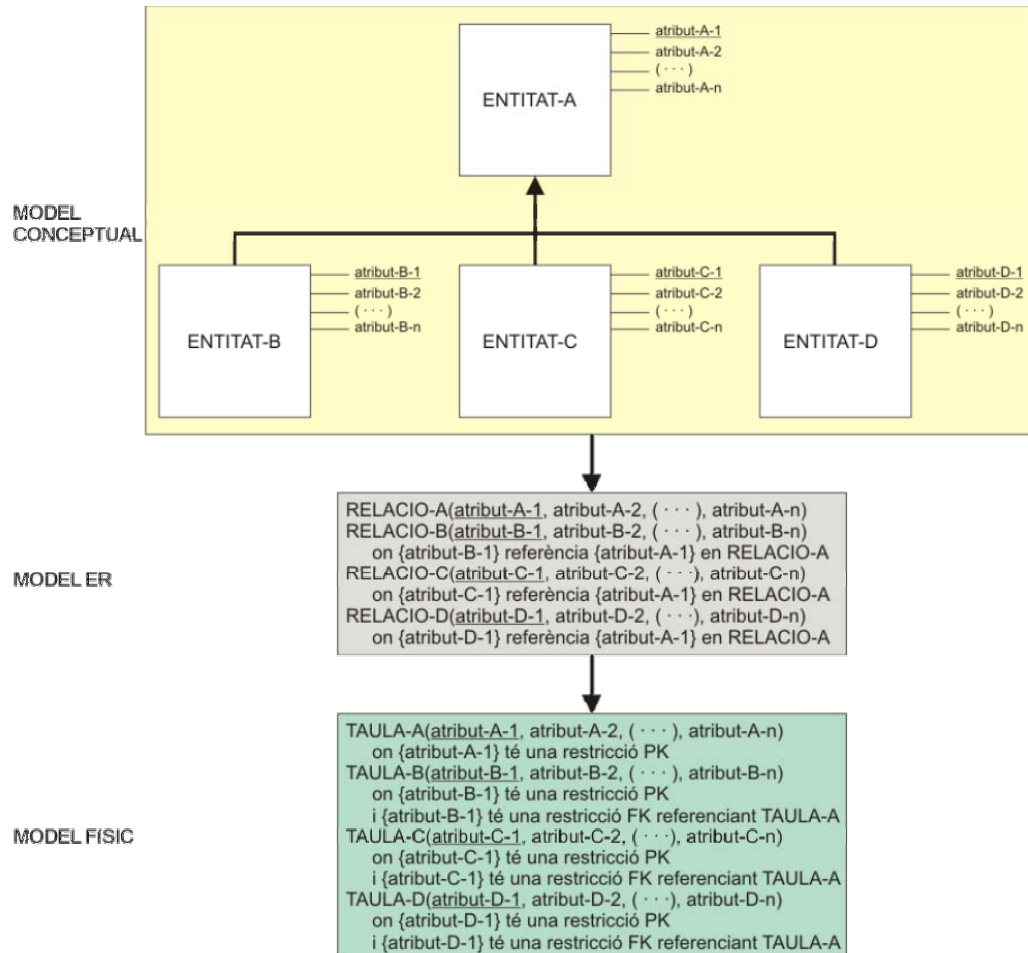
A nivell físic, existiran dos restriccions sobre els camps que componen la clau primària de la taula resultant de la transformació de l'entitat dèbil. Per una banda la restricció de clau primària, que exigeix que els valors atòmics dels camps que formen la clau siguin únics i que no admet valors nuls per a cap dels camps que hi formen part (aquesta darrera condició ens indica l'obligatorietat de la participació de l'entitat del costat 1 a la relació). Per altra banda existirà una altra restricció de tipus clau forana sobre la part de la clau primària que té que referenciar, necessàriament, a l'entitat forta.

- *Transformació de generalitzacions / especialitzacions.*

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

- a) La relació de l'entitat superclasse té com a clau primària la clau de l'entitat superclasse i conté tots els atributs comuns.

- b) Les relacions de les entitats subclasse tenen com a clau primària la clau de l'entitat superclasse i contenen els atributs específics de la subclasse.



Transformació d'esquemes. 3. Categorització

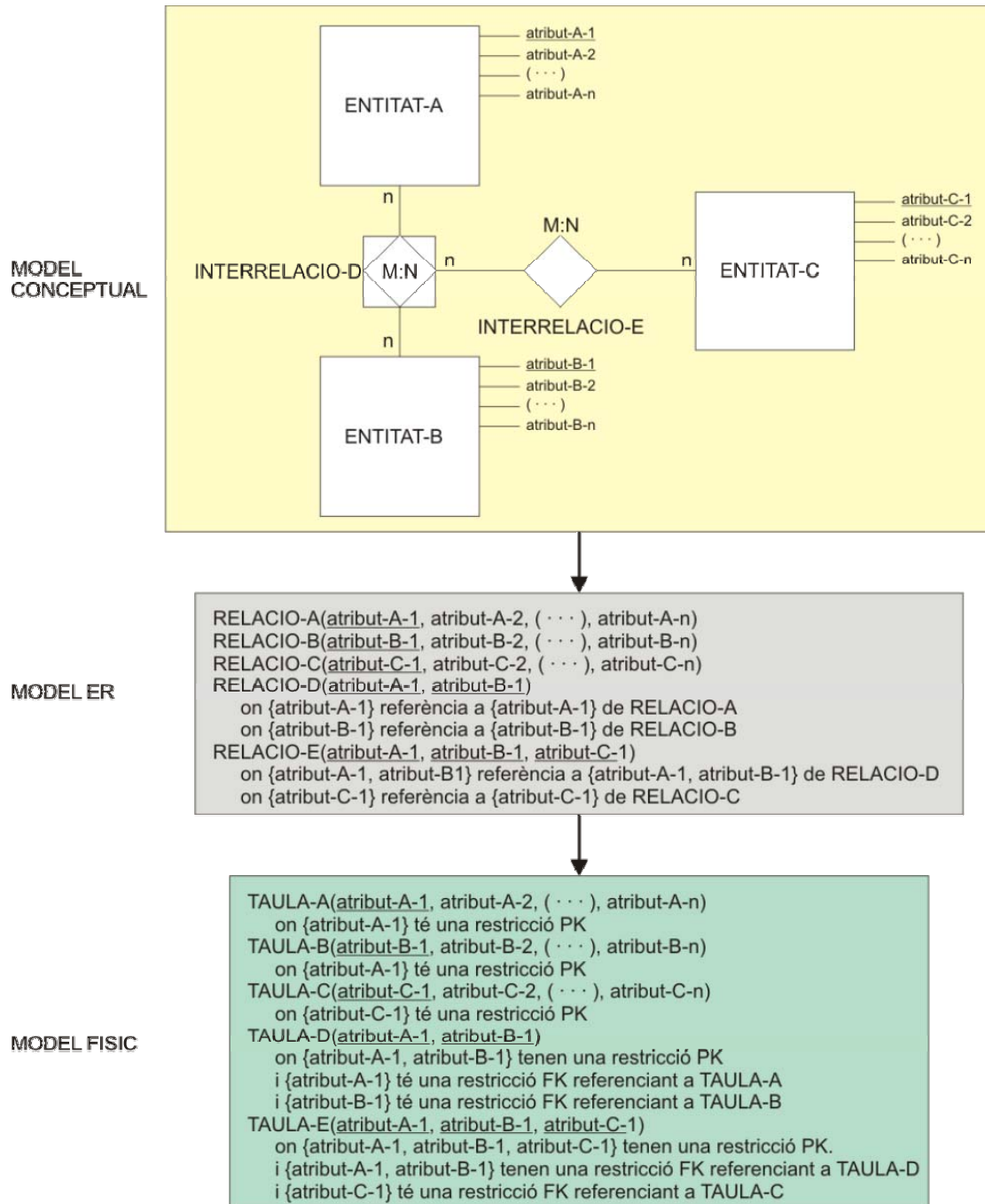
El procés de transformació deixa a la relació resultant de l'entitat superclasse tal i com està definida, però les relacions corresponents a les entitats subclasse, tenen a la seva clau primària referenciada mitjançant clau forana a l'entitat superclasse.

Tota la clau primària de les entitats subclasse es correspon exactament amb la clau primària de l'entitat superclasse.

A nivell físic, la taula corresponent a la transformació de l'entitat superclasse, es tracta com qualsevol taula (tindrà, com a mínim, una restricció de tipus clau primària). Les taules corresponents a les transformacions realitzades sobre les entitats subclasse, presentaran dues restriccions sobre tots els camps que componen la clau primària, per una banda la pròpia restricció de clau primària i per altra una restricció de clau forana que referenciarà la taula de l'entitat superclasse.

- *Transformació d'entitats associatives.*

L'anàlisi del procés de transformació d'aquesta mena d'entitats queda inclòs en aquest apartat, malgrat que podria anar perfectament en l'apartat d'anàlisi de transformacions d'interrelacions, atès que l'origen d'aquesta mena d'entitat és una interrelació.



Transformació d'esquemes. 4 Entitat Associativa

Una entitat associativa té el seu origen en una interrelació. En conseqüència, succeeix que la transformació de la interrelació originària és, alhora, la transformació de l'entitat associativa.

L'entitat associativa del cas presentat correspon a l'anomenada "INTERRELACIO-D" que presenta, a l'hora, característiques d'entitat al participar en una interrelació.

Cal fer notar que a l'exemple descrit hem fet la transformació de la interrelació "E", amb connectivitat M:N, cas que veurem més endavant.

A nivell físic, si pensem la nostra atenció exclusivament a la resolució de l'entitat associativa, veurem que dona lloc a una taula amb la clau primària composta per dos claus foranes en aquest cas particular. Per al cas de que l'entitat associativa fos fruit de la transformació d'una interrelació n-ària, la clau primària estaria formada seguint les regles de transformació que es detallen en el punt corresponent.

La característica que diferencia la transformació física d'una entitat associativa d'una altra binària M-N o n-ària, rau en el fet que la taula corresponent es referenciada per altres taules o en que té atributs propis no referenciats per cap clau forana participant de la clau primària.

Transformació d'interrelacions.

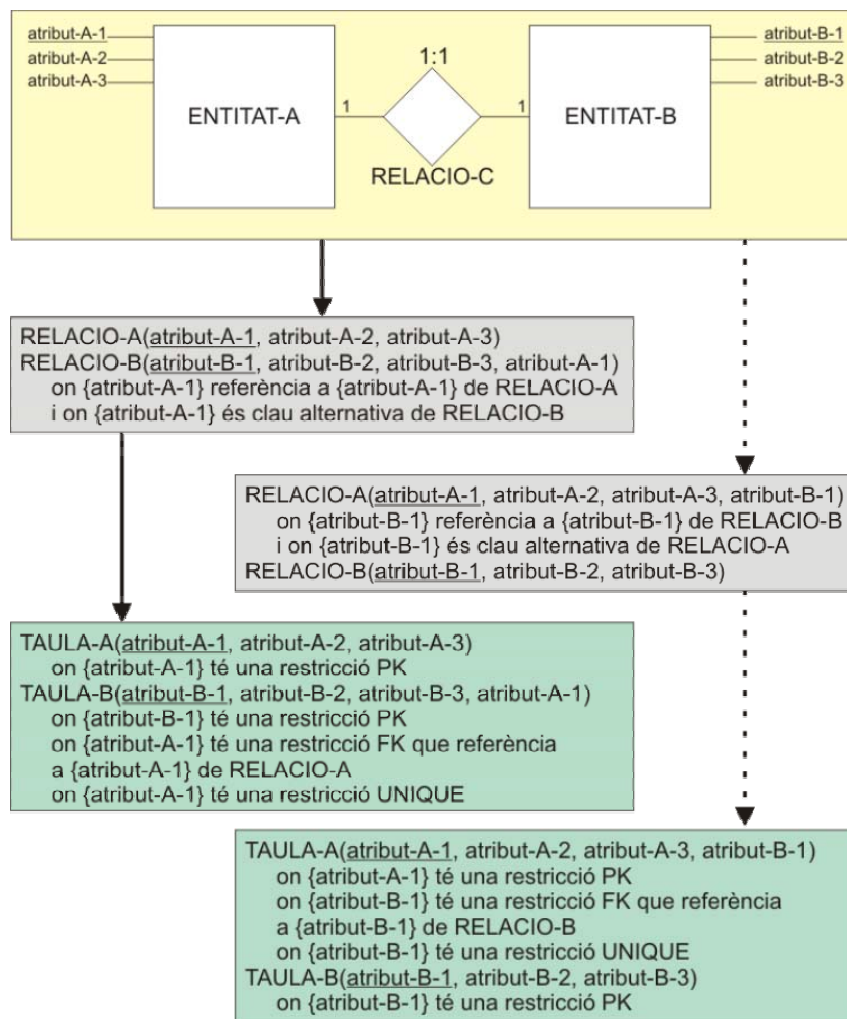
- *Transformació d'interrelacions binàries.*

Per a transformar una interrelació binària cal tenir en compte la seva connectivitat, i si les entitats són obligatòries o opcionals a la interrelació.

- connectivitat 1:1

El punt de partida és que les entitats que intervenen en la interrelació 1:1 ja s'han transformat en relacions amb els seus corresponents atributs.

Aleshores només caldrà afegir a qualsevol d'aquestes dues relacions una clau forana que referenciï l'altra relació.

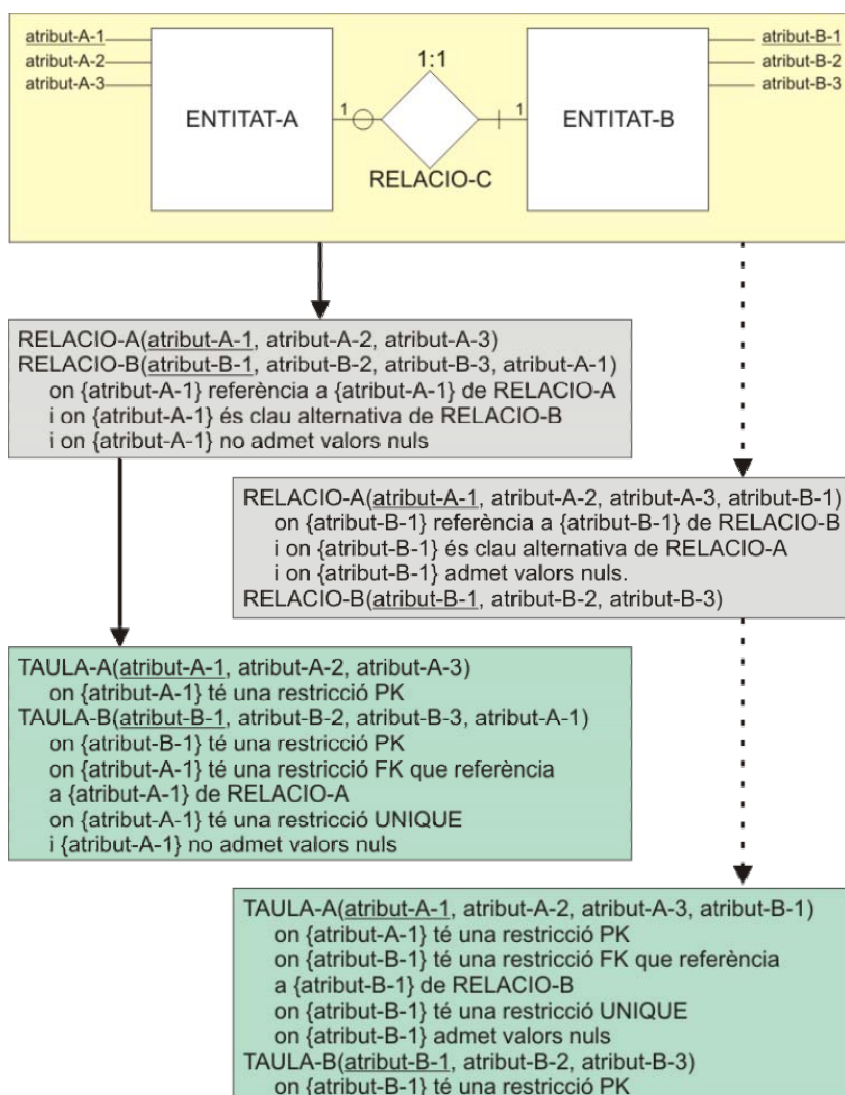


Transformació d'esquemes. 5 Relació 1:1

Com que hi ha dues entitats participants en la interrelació i, sense tenir més dades, qualsevol de les dues opcions proposades és correcta.

La interrelació té connectivitat 1:1, açò vol dir que per cada ocurrència d'una entitat existirà una i no més una ocurrència de l'altra entitat, per aquesta raó, la clau forana passa a ser clau candidata de la taula on s'insereix el nou camp i, per tant, presenta una restricció, a nivell físic, de tipus únic.

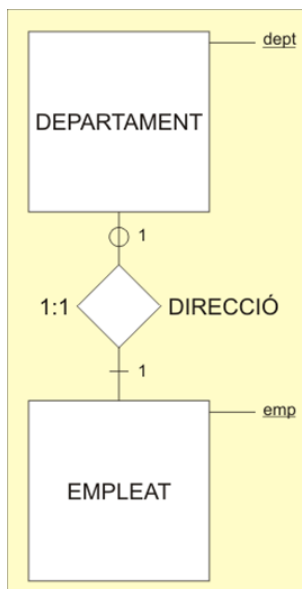
Ara bé, aquesta descripció és vàlida si tenim present tant sols la connectivitat entre les entitats. La obligatorietat o no d'una de les entitats participants a la interrelació fa que el resultat final sigui lleugerament diferent.



Transformació d'esquemes. 6 Relació 1:1 no obligatòria

Malgrat que totes dues solucions són correctes, l'admissió de valors nuls quan s'haurien pogut evitar (una de les solucions els admet i l'altra no) fa que la solució que no els admet sembli, a priori, més correcta que l'altra. Cal tenir present, en aquesta mena de casos, la semàntica del sistema que ja hem esmentat. Probablement una de les dos solucions s'ajusti millor al cas que l'altra.

Per exemple, en el cas proposat en els apunts de l'assignatura Base de Dades I de la UOC se'ns presenta el cas de la interrelació "DIRECCIÓ" entre les entitats "DEPARTAMENT" i "EMPLEAT", dins d'aquesta interrelació la participació d'empleat és obligatòria (segur que un empleat serà cap d'un departament concret) però la participació de departament no ho és (no tots els empleats són caps de departament).



Com a possibles transformacions d'aquest esquema se'ns presenten les solucions:

- Primera opció:

```
DEPARTAMENT(dept, ..., emp_cap)
  on {emp_cap} referència a {emp} a EMPLEAT.
EMPLEAT(emp, ...)
```

- Segona opció:

```
DEPARTAMENT(dept, ...)
EMPLEAT(emp, ..., dept)
  on {dept} referència a {dept} a
  DEPARTAMENT
  i {dept} pot prendre valors nuls.
```

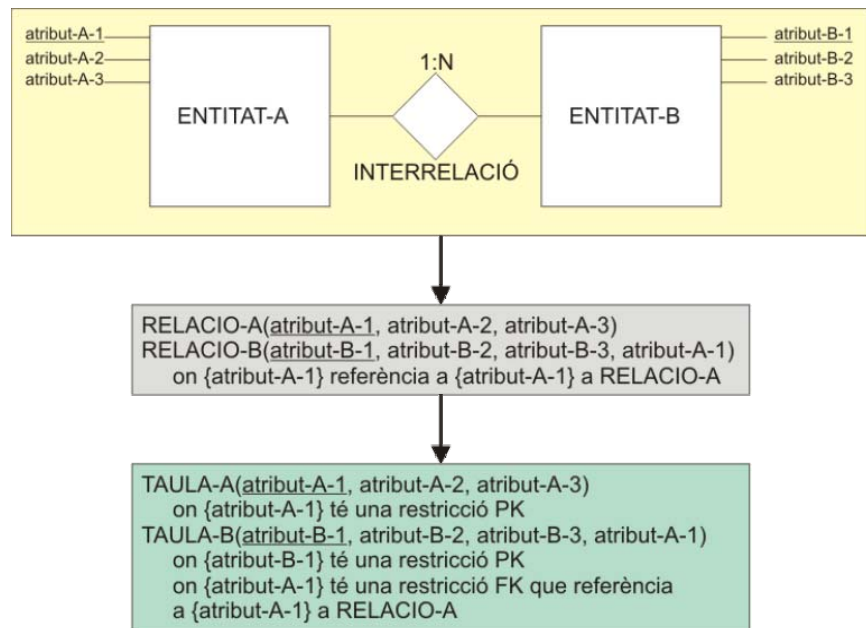
Malgrat que, formalment, qualsevol de les dues solucions reflecteix igualment la realitat, no passa el mateix amb la "semàntica" del cas.

Quan parlem d'empleats i departaments, el que és habitual és referir-nos al "cap d'un departament", és a dir, un atribut de Departament. Com que la participació de l'entitat "empleat" és obligatòria en la relació presentada, podem assegurar que tots els camps corresponents a l'atribut "emp_cap" no seran nuls.

L'aplicació de la segona solució, és a dir, la d'incloure l'atribut a l'entitat obligatòria, pot donar-nos valors nuls en el camp afegit, en aquest cas concret, tots els empleats que no fossin caps de departament tindrien aquest atribut amb valor nul, amb la consegüent pèrdua de contingut semàntic i el malbaratament d'espai d'emmagatzematge.

- connectivitat 1:N.

Partim del fet que les entitats que intervenen en la interrelació 1:N ja s'han transformat en relacions amb els seus corresponents atributs. En aquest cas només cal afegir a la relació corresponent a l'entitat del costat N una clau forana que referencii a l'altra relació.



Transformació d'esquemes. 7 Relació 1:N

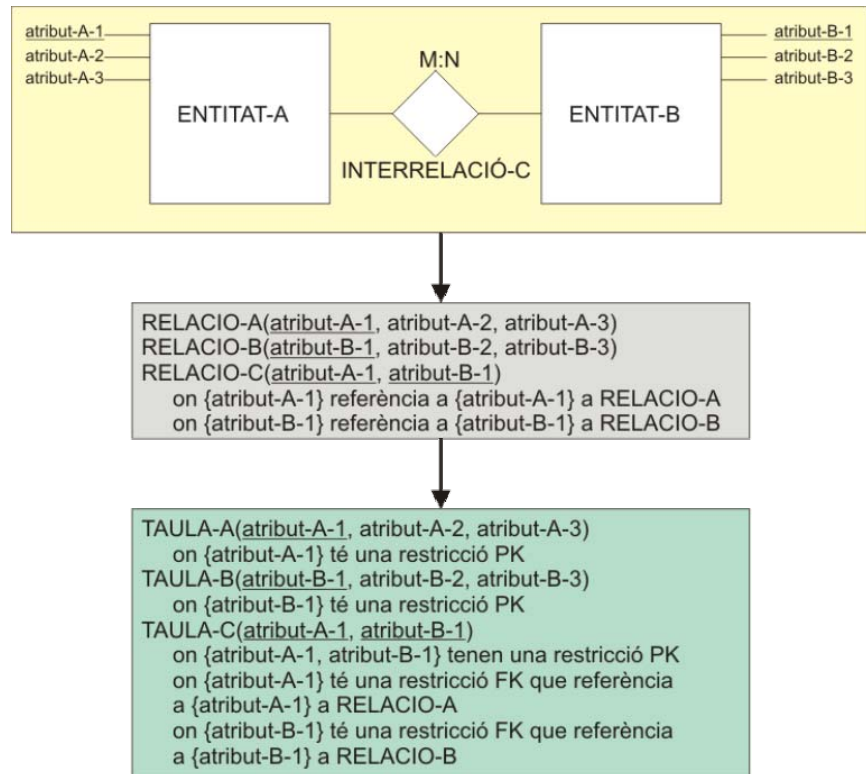
Al igual que en el cas de la connectivitat 1:1, el nou atribut de la relació referent no forma part de la clau primària d'aquesta.

A més i seguint amb el paral·lelisme amb la connectivitat 1:1, podria donar-se el cas en que la entitat del costat "1" (referenciada) fos opcional en la interrelació. En aquest cas la clau forana a la entitat del costat "N" (referent) caldria que admetés valors nuls. En aquest cas no tenim alternatives a escollir per salvaguardar la semàntica del cas, la única transformació possible és la que acabem de plantejar.

Així doncs, a nivell físic, la característica d'admetre valors nuls o no del camp que fa referència mitjançant la clau forana és el que determina si aquest camp admet valors nuls o no.

- connectivitat M:N.

Una interrelació M:N es transforma sempre en una relació. La seva clau primària estarà formada pels atributs de la clau primària de les dues entitats interrelacionades. Els atributs de la interrelació seran els atributs de la nova relació.



Transformació d'esquemes. 8 Relació M:N

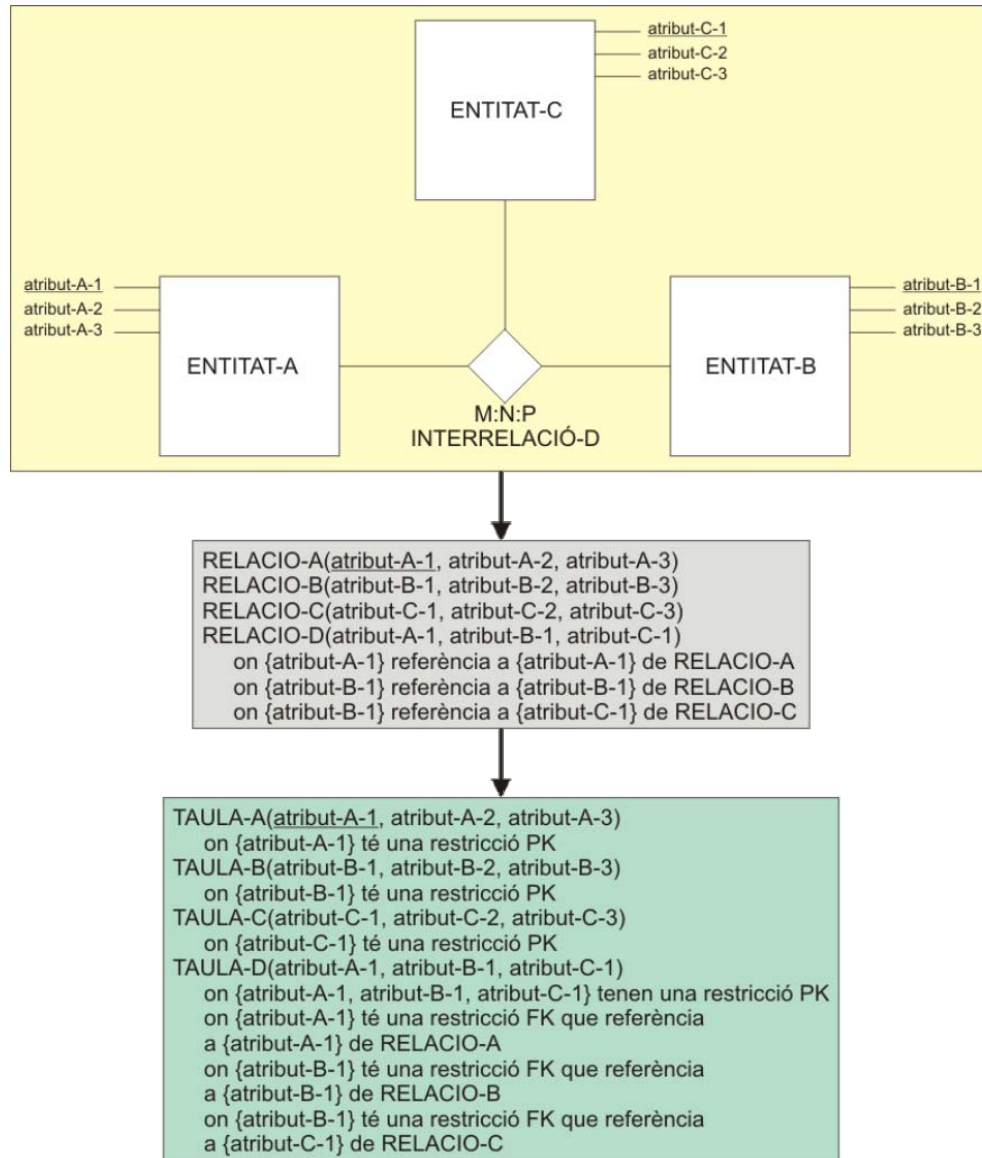
A nivell físic, la taula creada per representar la interrelació, tindrà la clau primària composta per exactament dues claus foranes que referenciaran a les taules derivades de les entitats participants a la interrelació.

- *Transformació d'interrelacions n-àries.*

En tots els casos la transformació d'una relació n-ària consistirà en l'obtenció d'una nova relació que conté tots els atributs que formen les claus primàries de les n entitats interrelacionades i tots els atributs de la interrelació.

Es poden distingir dos casos:

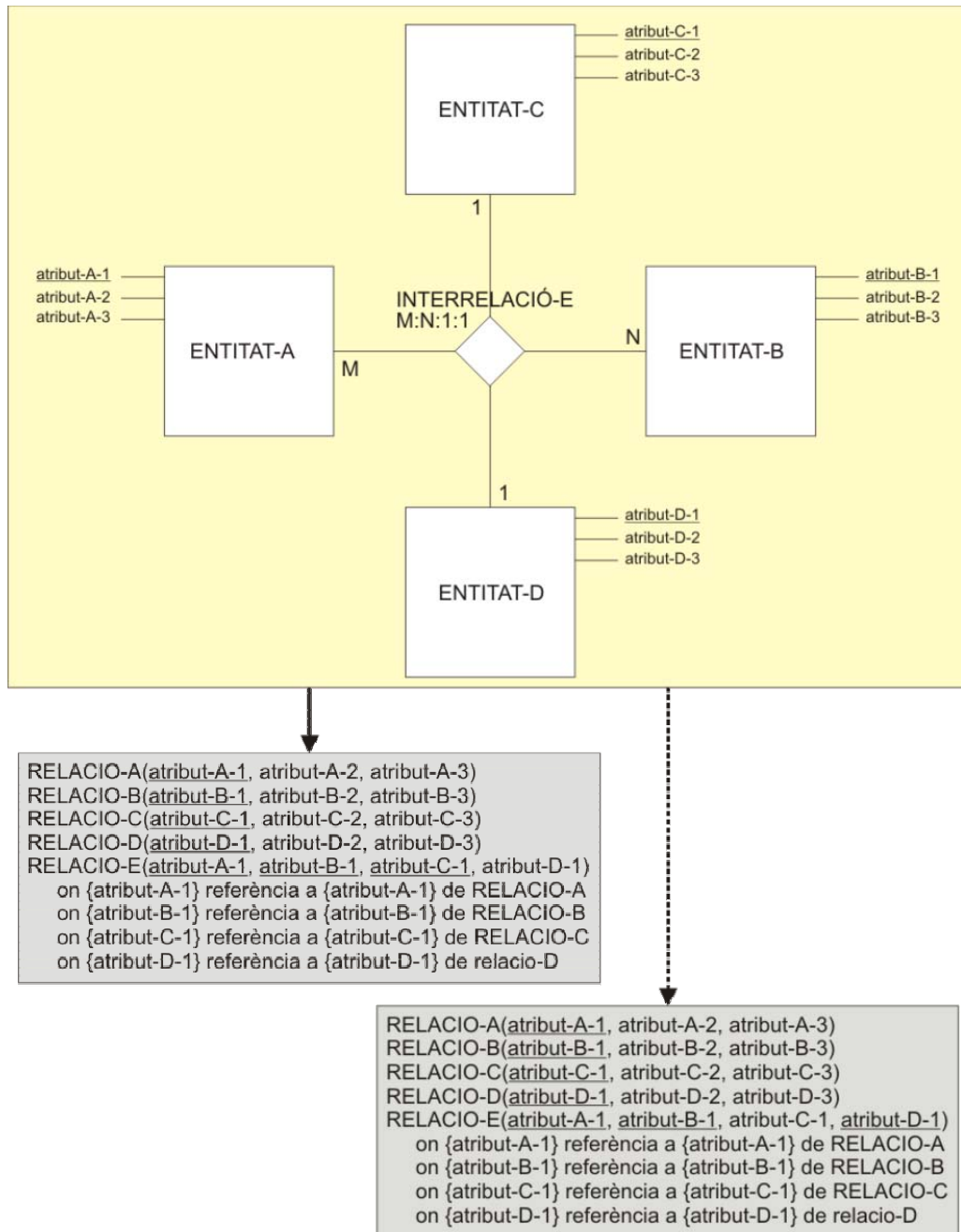
- a) totes les entitats estan connectades a la interrelació amb cardinalitat “molts”, aleshores la clau primària de la nova relació estarà formada per tots els atributs que formen les claus de les n entitats interrelacionades.



Transformació d'esquemes. 9 Relació n-ària

Hem pres com exemple el cas d'una interrelació ternària amb totes les connectivitats a “N”.

- b) Si una o més entitats estan connectades amb cardinalitat “un”, la clau primària de la nova entitat estarà formada per les claus de $n - 1$ de les entitats interrelacionades, amb la condició que l'entitat, la clau de la qual no s'ha inclòs ha de ser una de les que està connectada amb cardinalitat “un”.



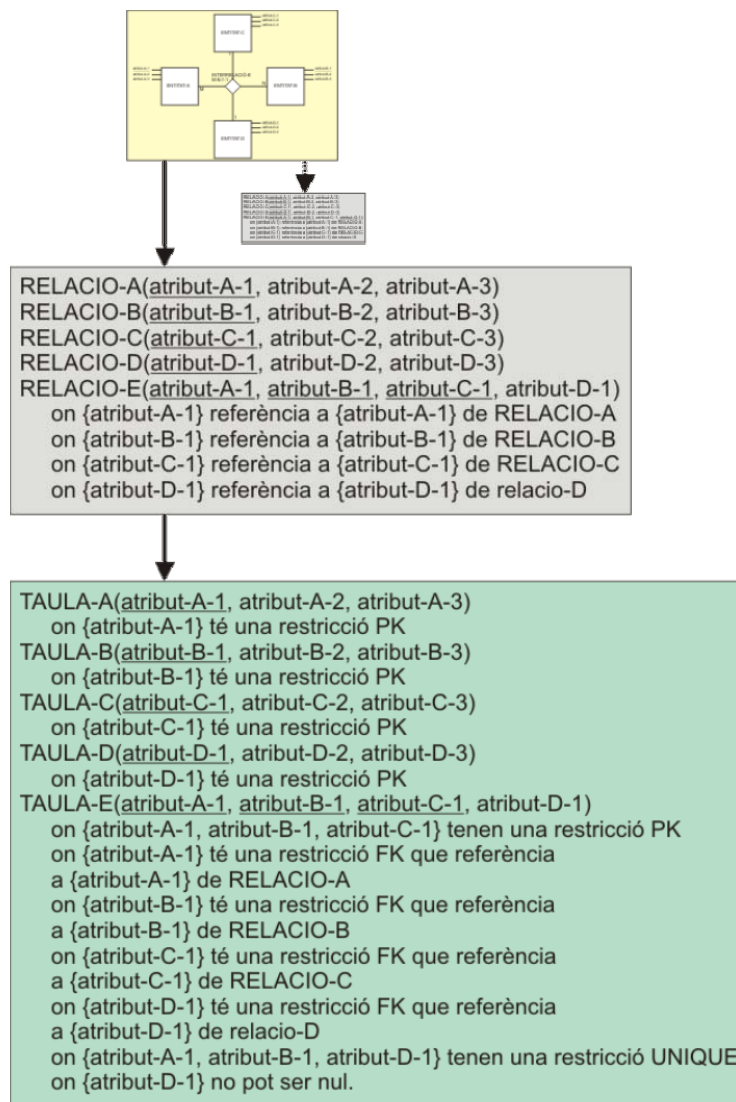
Transformació d'esquemes. 10 Relació n-ària

Cal fer notar que amb les condicions de transformació per interrelacions amb el tipus de connectivitat que compleixi les restriccions de l'enunciat de la norma de transformació, podem tenir diverses solucions vàlides per la transformació. En principi hi haurà tantes solucions com entitats amb connectivitat 1 participants a la interrelació. La única diferència entre una solució i l'altra rau en els atributs escollits com a clau primària de la relació resultant de la transformació de la interrelació.

El cas representat gràficament correspon a una interrelació quaternària on dos de les seves relacions participants presenten connectivitat 1, hi haurà doncs 2 transformacions possibles.

La transformació a nivell físic, una vegada hem triat quins atributs formaran la clau primària, és directa. Però cal tenir present que les combinacions descartades són claus candidates de la taula, per tant cal que presentin una restricció de tipus UNIQUE per salvaguardar adientment la semàntica del cas.

Així, suposant que la solució adoptada fos la primera, la transformació a nivell físic quedaria:



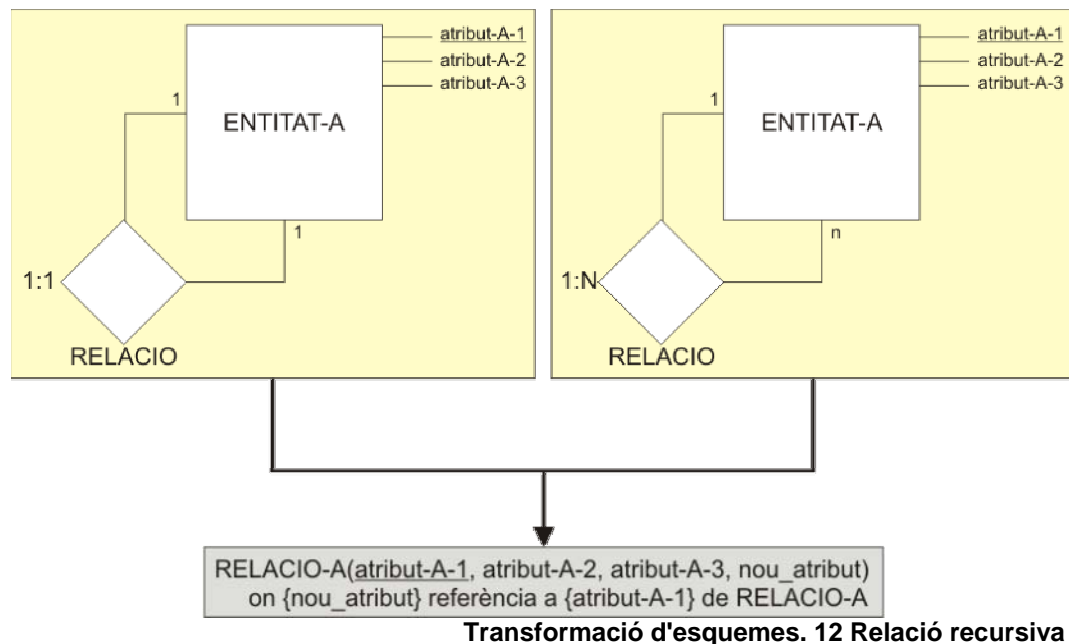
Transformació d'esquemes. 11 Relació n-ària

La salvaguarda de la semàntica del cas ens exigeix la declaració d'una restricció de tipus UNIQUE sobre la clau alternativa descartada, a més cal definir als atributs descartats com no nuls.

- *Interrelacions recursives.*

Les interrelacions recursives representen el cas de les entitats que es relacionen amb elles mateixa. El tractament que donarem a aquests casos és el mateix que per els que ja hem vist a altres interrelacions.

Així, doncs, si una interrelació recursiva té connectivitat 1:1 o 1:N, dóna lloc a una clau forana i, si té connectivitat M:N o és n-ària, origina una nova relació.



Malgrat que la transformació a nivell lògic és igual per aquests casos, la transformació a nivell físic presenta diferències.

Així en el cas d'una relació recursiva 1:1, al camp que afegim a la taula resultant (*nou_atribut*) caldrà que li apliquem una restricció de tipus UNIQUE. Si a més, la participació de la entitat a la interrelació és obligatòria per tots dos costats, caldrà que l'atribut afegit sigui declarat com "no nul".

En el cas d'una relació recursiva 1:N, al camp que afegim a la taula resultant no tindrà sentit l'aplicació de la restricció UNIQUE. Tant sols la obligatorietat o no de la participació de la entitat per tots dos costats, determinarà la característica de "nullable" del nou atribut.

Les interrelacions recursives binàries M:N o n-àries, es transformaran seguint idèntiques pautes que les definides per a interrelacions no recursives.

Quadre Resum.

Presentem de forma esquemàtica un resum de les característiques que presentarà l'esquema físic a partir de les transformacions aplicades a l'esquema conceptual.

ENTITATS	Regulars		Presenta una restricció PK.
	Dèbils		Presenta una restricció PK. Part de la PK presenta una restricció FK que referència a l'entitat forta.
	Generalització/ Especialització		Superclasse: - És un cas d'entitat regular. Subclasse: - Presenta una restricció PK. - La PK, tota ella, presenta una sola restricció FK que referència a la superclasse.
	Associatives		El seu origen és una interrelació. Altres entitats (de qualsevol tipus) poden establir interrelacions amb ella.
INTERRELACIONS	Binàries	1:1	No tots dos costats obligatoris: - S'afegeix un camp a qualsevol de les dues entitats. - El camp afegit té una restricció FK que referència a l'altra entitat participant. - El camp afegit és clau candidata de la entitat, a nivell físic presenta una restricció de tipus UNIQUE. - El camp afegit pot ser nul. Tots dos costats obligatoris. - Igual que el cas en que no tots dos costats són obligatoris però en aquest cas el camp afegit no pot ser nul.
		1:N	S'afegeix un nou atribut a la taula resultat de la transformació de l'entitat de la part "N". El nou camp presenta una restricció de tipus FK que referència a la clau primària de l'entitat del costat 1. Si la participació de l'entitat del costat 1 és obligatòria el camp afegit no podrà ser nul.
		M:N	Es crea una nova taula, els atributs de la qual són formats per els atributs que componen les claus primàries de cadascuna de les entitats participants a la interrelació. La clau primària, tota ella i en dos parts, presenta restriccions FK a les entitats que participen de la interrelació.
	N-àries	Tot N	- Nova relació amb clau primària formada per les claus primàries de les entitats participants. A més existiran restriccions FK a nivell físic per referenciar a les entitats participants.
		No tot N	- Nova relació amb clau primària formada per les claus primàries de les 2^{N-1} entitats participants i les seves corresponents FK. - Restricció de tipus UNIQUE a les $2^{N-1}-1$ claus candidates.
	Recursives		S'apliquen idèntics criteris que en la resta d'interrelacions.

Retrodisseny.

Per a la realització de l'anàlisi i el disseny del mòdul que s'encarregarà de realitzar el procés d'enginyeria inversa sobre la base de dades, hem realitzat un estudi previ sobre el procés de transformació entre diferents nivells d'esquema, des de l'esquema conceptual, passant per el lògic, fins a arribar al físic. Aquest estudi es pot trobar com el annexe 1 a aquesta memòria.

Una vegada hem analitzat les característiques físiques que el procés de transformació entre esquemes, deixa en les taules finals de la base de dades a analitzar, podem començar el procés d'anàlisi.

Aquest procés es durà a terme taula per taula, provant d'esbrinar tanta informació de l'esquema conceptual original com sigui possible.

Recordem que les estructures que cerquem es poden resumir en:

- Entitats Regulars.
- Entitats Dèbils.
- Generalitzacions / Especialitzacions.
- Entitats Associatives.
- Relacions binàries 1:1.
- Relacions binàries 1:N.
- Relacions binàries M:N.
- Relacions n-àries.
- Relacions recursives.

Així doncs, determinarem en els següents punts, els mètodes i les proves lògiques adients per esbrinar si la taula que analitzem és correspon a qualsevol de les estructures esmentades.

Entitats Regulars.

Dins d'un esquema conceptual, una entitat regular, pot relacionar-se amb qualsevol altra entitat mitjançant qualsevol tipus de relació. Ara bé, en determinades ocasions trobarem entitats que tant sols són referenciades per altres, en altres ocasions les entitats faran referència a altres entitats.

Aquesta diferència és important ja que la marca física final és substancialment diferent una de l'altra.

- *Entitat Regular Referenciada.*

La taula corresponent a una entitat regular referenciada, no té cap restricció de tipus clau forana. Si en el nostre anàlisi, taula per taula, ens trobem en

aquest cas, podem guardar la informació sobre la taula en la estructura adient per a Entitats Regulars i passar a la següent taula.

- *Entitat Regular Referent.*

La taula corresponent a una entitat regular referent, té alguna restricció de tipus clau forana que no participa de la restricció de clau primària. El sistema per determinar el tipus d'interrelació que determina la clau forana detectada serà tractat en els apartats corresponents.

Entitat Dèbil.

Hem comprovat al punt anterior, que les entitats regulars presenten, totes elles, restriccions de clau primària que no tenen relació amb les possibles claus foranes declarades a les seves taules. Aquesta situació es dona exclusivament a les entitats regulars. A partir d'aquest punt, totes les taules tindran claus primàries participades per claus foranes.

Així, per a una taula, on la clau primària sigui composta (dos o més atributs) i una (i només una), part de la clau primària sigui referenciada per una sola clau forana, aleshores podem dir que l'estructura de l'esquema conceptual original per a aquesta taula és el d'una entitat dèbil, sent l'entitat forta la taula referenciada a la clau forana.

Una vegada determinat el caràcter original de l'entitat, continuarem analitzant les claus foranes que no participen de la clau primària de la taula per determinar possibles interrelacions on hi participés.

Generalització / Especialització.

Una vegada més determinarem que una taula concreta correspon a una estructura de tipus Generalització / Especialització, parant atenció a com es relacionen la seva clau primària i les seves claus foranes (si hem arribat fins aquí, sabem que en té).

Així, si la taula que analitzem té una clau primària que està referenciada, tota ella, per una, i només una, clau forana, podem determinar que la taula correspon a una entitat subclasse dins d'una generalització / especialització, sent la entitat superclasse la taula referenciada per la clau forana que compona totalment a la clau primària.

Una vegada determinat el caràcter original de l'entitat, continuarem analitzant les claus foranes de la taula per determinar possibles interrelacions on hi participés.

Interrelació 1:1.

En l'anàlisi de transformació entre diferents nivells d'esquemes, hem determinat que una interrelació 1:1 no crea taules físiques en el seu procés fins a l'esquema físic. El resulta d'aquesta transformació per a aquest tipus d'interrelació, és la incorporació en la taula corresponent a la entitat que participa de la interrelació de forma obligatòria (en qualsevol de les dos, si totes dues tenen participació obligatòria), de nous atributs que referenciaran mitjançant clau forana a la clau primària de l'altra entitat.

A més, i donada la connectivitat de la interrelació, el camp afegit passa a ser clau alternativa de la relació, per la qual cosa es tindrà que haver afegit una restricció de tipus UNIQUE al camp afegit (recordem que aquest tipus de restricció assegura que els valors per aquest camp seran diferents a qualsevol de les tuples de la taula, el valor nul es considerat com "no definit" i per tant no compta per a la restricció d'unicitat declarada).

Podem determinar, fins i tot, l'obligatorietat de la participació de l'entitat referenciada a la interrelació. Si el camp afegit admet nuls, la participació de l'entitat referenciada és optativa, obligatòria altrament. Ara bé, l'adopció d'aquesta solució és ineficient perquè pot provocar el malbaratament d'espai d'emmagatzematge a la base de dades.

Interrelació 1:N.

Al igual que les interrelacions 1:1, aquest tipus d'interrelacions donen com a resultat taules amb camps afegits que formaran una clau forana que referenciarà a l'entitat del costat 1 de la interrelació. La diferència rau en el fet de que no tindran associada cap restricció d'unicitat als camps afegits.

Igualment, la determinació de la obligatorietat de participació de l'entitat del costat 1 a la interrelació, es realitzarà en base a si els camps afegits admeten valors nuls o no.

Interrelació M:N i n-àries amb totes les connectivitats N.

Aquest punt analitza al mateix temps dos tipus d'estructures diferents, però podem considerar que una interrelació M-N és un cas particular d'una interrelació n-ària amb totes les connectivitats a N.

Així si la taula que estem analitzant presenta una clau primària que està composta, tota ella, per més d'una clau forana, i, a més, tots els camps de la taula formen part de la clau primària, podem assegurar que correspon a la transformació d'una interrelació amb totes les connectivitats a N.

El grau de la interrelació ens donarà el nombre de claus foranes que formen la clau primària (una per cada entitat participant).

Si la taula presentés qualsevol altra clau forana fora de les esmentades, és a dir, si tingués camps fora de la clau primària, aleshores estaríem parlant d'una entitat associativa. El mateix passaria si la taula tingués camps propis fora de la clau primària.

Interrelació n-ària amb no totes les connectivitats N.

Per a que la taula que analitzem doni com a estructura originària aquesta mena d'interrelacions, cal que es compleixin diverses condicions.

1. La clau primària serà composta.
2. Cadascun dels conjunts de camps components de la clau primària tindrà la seva correspondència amb una clau forana.
3. Existiran restriccions d'unicitat per cadascuna de les combinacions de les claus foranes descartades per participar a la clau primària de l'entitat amb els atributs que si ho han fet. És a dir, existiran tantes restriccions d'unicitat com entitats participants a la interrelació ho facin amb connectivitat "1".

A diferència de les interrelacions n-àries amb totes les connectivitats a N, la presència de camps fora de la clau primària no implica que es tracti d'una entitat associativa. Els camps que queden fora de la definició de clau primària, si hi són inclosos en les restriccions d'unicitat amb participació d'altres elements que si estan a la clau primària, són indicació de la presència d'altra entitat amb connectivitat "1" que participa de la interrelació.

Entitat associativa.

Hem detectat en els punts corresponents a la determinació d'interrelacions n-àries (les úniques que creen taules per el procés de transformació, si fem extensiu el concepte de "n-àries" a les interrelacions binàries M:N), determinats casos on es poden presentar claus foranes que no tinguin res a veure amb les claus primàries o candidates de les interrelacions.

En aquests casos estem al davant d'una entitat associativa que es relaciona amb una entitat mitjançant una interrelació 1:1 o 1:N, participant la entitat associativa del costat N de la interrelació.

Per a la resta de casos (participació en interrelacions n-àries o binàries 1:1 o 1:N, però del costat 1) l'únic mètode que podem seguir és el de detectar primer a la entitat associativa i, després, comprovar si altres entitats hi fan referència.

Un altre fet que indica la presència d'una entitat associativa és la concurrència d'una clau primària composta per més d'una clau forana i la presència d'atributs propis a la taula (no referenciats per cap restricció de clau forana).

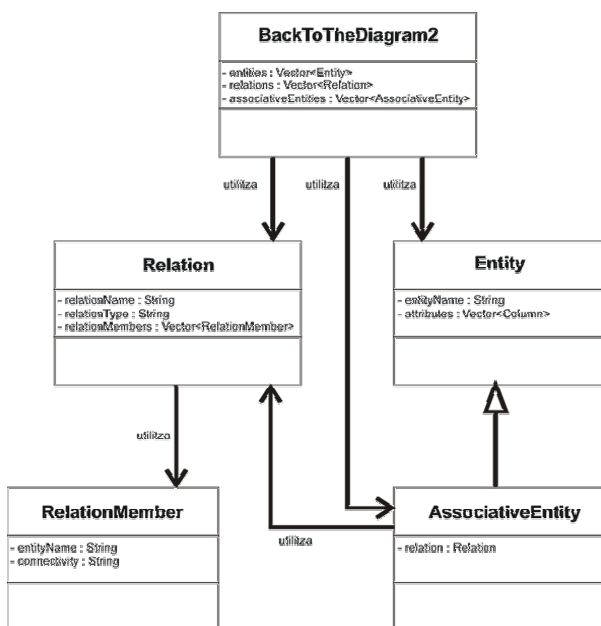
Interrelació recursiva.

El criteri únic per determinar la recursivitat d'una interrelació és que alguna de les claus foranes de la taula, faci referència a la mateixa taula.

La determinació del tipus d'interrelació, el seu grau i les connectivitats presents, es realitzaran en base a l'anàlisi dels diferents tipus d'interrelacions descrits als punts anteriors.

Solució proposada.

Per al mòdul de retrodisseny, proposem la integració en una única classe de tots els mètodes i les proves lògiques per obtenir la transformació de taules i camps en entitats, relacions i atributs.



2.RetroDisseny. Diagrama de classes.

Classe BackToTheDiagram2: És 2 perquè és la segona versió que s'ha implementat. És l'encarregada de guardar les estructures corresponents al diagrama entitat-relació que donà com a fruit la base de dades que estem retrodissenyant. A més guarda els mètodes i proves lògiques per determinar el caràcter original que correspon a una determinada taula.

Classe Entity: Classe que modelitza la informació necessària per descriure una entitat en un model ER. És emprada per la classe **BackToTheDiagram2** per emmagatzemar les entitats que va determinant.

Classe Relation: Classe que modelitza la informació necessària per descriure una interrelació en un model ER. És utilitzada per la classe **BackToTheDiagram2** per guardar la informació referent a les relacions que va determinant. Utilitza la classe **RelationMember**.

Classe RelationMember: Classe que modelitza la participació d'una entitat en una interrelació, guardant el nom de l'entitat i la connectivitat amb que participa.

Classe AssociativeEntity: Estén a Entity. Modelitza la informació requerida per descriure una entitat associativa, és a dir, hem afegit un objecte de tipus *Relació* als atributs que ja tenia la classe Entity. És utilitzada per la classe BackToTheDiagram per guardar els objectes que determina d'aquest tipus.

Mètodes i proves lògiques.

És adient fer una petita exposició dels mètodes que permeten aplicar el procés de redisseny a la base de dades.

Com ja hem dit el depositari dels mètodes i proves lògiques per determinar el caràcter original d'una taula, és la classe BackToTheDiagram2, a més dels mètodes purament lògics la classe també disposa d'un conjunt de funcions d'utilitat que permeten manegar les dades d'una forma més senzilla.

Aquests mètodes són:

- Vector<ForeignKey> getFKinPK(Table table).

Aquesta funció retorna un vector amb els objectes de tipus ForeignKey que fan referència a camps que formen part de la clau primària de la taula.

- int FKsinPK(table).

Retorna el nombre de claus foranes que participen de la clau primària de la taula.

- Vector<ForeignKey> getFKoutPK(Table table).

Mètode que retorna els objectes de tipus ForeignKey definits a la taula que no referencien a cap dels components de la clau primària.

- int FKsoutPK(Table table).

Mètode que retorna el nombre de claus foranes definides a la taula que no participen de la clau primària.

- Vector<UniqueKey> getUKinPK(Table table).

Funció que retorna els objectes de tipus UniqueKey definits a la taula que contenen alguna referència als camps que formen part de la clau primària.

- `int UKsinPK(Table table)`.

Funció que retorna el nombre de claus úniques que contenen alguna referència als camps que formen part de la clau primària a la seva definició.

- `Vector<Column> getColumnOutPK(Table table)`

Funció que retorna un vector d'objectes de tipus `Column` corresponents als atributs de la taula que no participen de la clau primària.

- `boolean allPKisFK(Table table)`

Funció que retorna cert si tota la clau primària està referenciada en una o més d'una claus foranes.

- `Vector<Column> removeFKfromPK(Table table)`

Mètode que retorna un vector d'objectes de tipus `Column`, corresponents a la eliminació d'aquells atributs de la clau primària que tinguin relació amb alguna clau forana.

- `void analyzeRelations(Table table)`

Mètode que s'encarrega de cercar relacions de tipus 1:1 i 1:N a entitats regular, dèbils, categoritzacions o entitats associatives. S'encarrega de la seva creació al vector de relacions definit a la classe `BackToTheDiagram2`.

- `boolean isRegularEntity(Table Table)`

Mètode que utilitza la funció `FKsinPK`, comprovant-hi que el resultat d'aquesta sigui igual a zero, en aquest cas, és a dir, quan no hi ha claus foranes que referencien a camps que participen de la clau primària, podem assegurar que estem davant d'una entitat regular.

- `boolean isWeakEntity(Table table)`

Mètode que determina si una taula en concret correspon a una entitat dèbil i, a més, crea les oportunes entrades en el vector d'entitats i en el de relacions. Per fer-ho comprova que el nombre de `FKsinPK` sigui igual a 1 i que no tota la clau primària sigui referenciada per claus foranes (`allPKisFK()`), és a dir, existirà un conjunt d'atributs propis de la entitat corresponent que no són clau forana, que és la característica de les entitats dèbils.

- boolean isCategorization(Table table)

Al igual que l'anterior mètode, aquest comprova que el nombre de claus foranes participants de la clau primària sigui exactament 1 i, a més, comprova que tota la clau primària sigui referenciada per una clau forana, que és la característica de les categoritzacions. En el cas de les categoritzacions detectem a l'entitat corresponent a la superclasse com una entitat regular, la categorització la detectem a partir de les entitats corresponents a les subclasse.

- boolean isNariaRelation(Table table)

Aquest mètode comprova que el nombre de claus foranes participants de la clau primària sigui superior a 1, per la qual cosa detectarà interrelacions a partir de grau 2. A més és capaç de determinar la connectivitat en que participen cadascuna de les entitats a la interrelació, creant-hi les entitats i relacions en els vectors corresponents.

- boolean isAssociativeEntity(Table table)

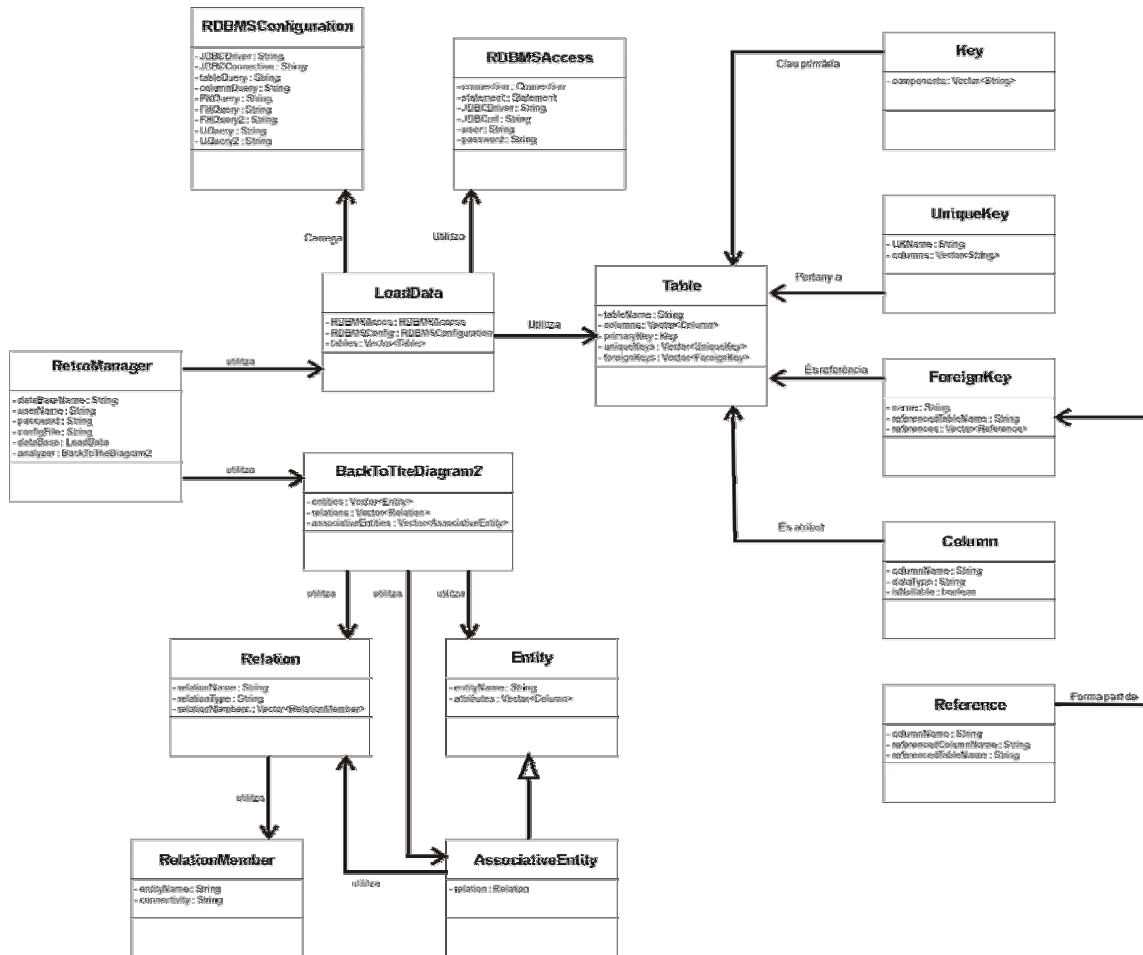
Per tal de que una taula correspongui a una entitat associativa cal que es donin les següents condicions, el nombre d'entitats participants de la clau primària mitjançant claus foranes, serà més gran que 1, a més, d'aquesta condició obligatòria, cal que es doni una qualsevol de les dos següents, cal que no tota la clau primària no sigui referenciada per claus foranes, o bé, cal que no tots els atributs de la taula formin part de la clau primària, aquestes darreres condicions indiquen la presència d'atributs propis de la taula que, al tenir més d'una referència a la clau primària, sabem que és una entitat associativa.

- void report()

Funció que produeix una sortida per pantalla amb la informació sobre el diagrama entitat-relació original que hem guardat a les estructures internes de la classe BackToTheDiagram2.

Integració dels mòduls.

Per tal de que l'aplicació funcioni, cal integrar tots dos mòduls, el de càrrega de dades i el de redisseny, per tal de que funcionin seqüencialment. Així proposem la creació d'una nova classe que s'encarregarà de rebre els paràmetres de l'usuari.



3. Diagrama de classes complet.

Com es pot comprovar, hem integrat els diagrames de classes corresponents al mòdul de càrrega de dades i al mòdul d’anàlisi connectant-los mitjançant la classe “RetroManager” que s’encarregarà de realitzar la gestió del procés.

Manual d'usuari.

La aplicació de Redisseny està pensada per a usuaris que tenen un grau de coneixement i d'experiència elevat en el món de les bases de dades relacionals.

Conceptes com SQL, taula, entitat o relació no tenen que ser estranys, si aquest fos el cas, desaconsellem que s'utilitzi aquesta aplicació ja que les sortides que ofereix serien del tot incomprensibles. A més, caldrà escriure patrons de consultes SQL per obtenir informació del catàleg del SGBDR, per la qual cosa es requereix una certa experiència per obtenir els resultats desitjats.

La aplicació està realitzada en llenguatge Java i utilitza la tecnologia JDBC per establir les necessàries connexions al Sistema Gestor de Base de Dades Relacionals (SGBDR en endavant) per obtenir informació sobre les estructures físiques on s'emmagatzemen les dades.

Així doncs, és necessari proveir a l'aplicació del driver JDBC adient per al SGBDR que utilitzarem. Aquest driver és un fitxer amb extensió ".jar" que podem trobar (probablement) a la pàgina web del fabricant del nostre SGBDR. El fitxer obtingut cal que es copiï al mateix directori on tenim l'aplicació.

Igualment cal aconseguir dos paràmetres per al funcionament de la connexió. La cadena de declaració del driver i la cadena d'establiment de connexió. Aquestes dues cadenes es trobaran, probablement al mateix lloc on obtenim el fitxer .jar esmentat abans.

L'aplicació incorpora el driver i les definicions necessàries per funcionar contra el SGBDR MySql 5.

Amb la informació obtinguda cal crear un fitxer de configuració per la connexió al SGBDR, aquest fitxer consta de dos paràmetres (declaració de driver i connexió) i set seccions que corresponen, cada una d'elles, a una consulta SQL que caldrà reescriure per diferents SGBDRs.

El fitxer creat serà de tipus text, amb el següent contingut:

Secció "general", on inclourem la informació obtinguda per a la cadena de declaració de driver i per la cadena d'obtenció de connexió que ja hem esmentat.

```
[general]
driver="cadena de declaració de driver"
connection="cadena d'obtenció de connexió"
```

Secció "tableQuery", on escriurem una consulta SQL contra el catàleg del SGBDR que actua com a hoste de la base de dades que volem analitzar. La sortida

d'aquesta consulta cal que sigui d'una única columna amb el nom de les taules que formen part de la base de dades que volem analitzar.

```
[tableQuery]
SELECT table_name
FROM tables
WHERE table_schema=' #DATABASE# ' ;
```

Secció “columnQuery”, novament escriurem una consulta SQL contra el catàleg del SGBDR, ara, el que cerquem és el nom dels atributs de cadascuna de les taules, el tipus de dades i si admeten nuls o no,

```
[columnQuery]
SELECT column_name, data_type, is_nullable
FROM columns
WHERE table_schema=' #DATABASE# '
AND table_name=' #TABLENAME# ' ;
```

Secció “PKQuery”, ara cerquem informació sobre les claus primàries de cadascuna de les taules, necessitem saber el nom de la columna i la posició dins de la restricció de tipus “clau primària”.

```
[PKQuery]
SELECT kcu.column_name, ordinal_position
FROM table_constraints tc, key_column_usage kcu
WHERE tc.table_schema=' #DATABASE# '
AND kcu.table_schema=' #DATABASE# '
AND tc.table_name=' #TABLENAME# '
AND tc.constraint_type='PRIMARY KEY'
AND tc.constraint_name=kcu.constraint_name
AND tc.table_name=kcu.table_name
ORDER BY kcu.ordinal_position;
```

Secció “FKQuery”, recollirà la consulta necessària per obtenir la llista de claus foranes que hi ha definida per cadascuna de les taules, cal obtenir el nom de la restricció de tipus clau forana.

```
[FKQuery]
SELECT constraint_name
FROM table_constraints
WHERE table_schema=' #DATABASE# '
AND table_name=' #TABLENAME# '
AND constraint_type='FOREIGN KEY' ;
```

Secció “FKQuery2”, complementaria a l'anterior i obté informació sobre els components de cadascuna de les restriccions de tipus “clau forana” que hem trobat amb la consulta anterior. Cal obtenir el nom de columna de la taula que

referencia, la posició a la restricció de tipus clau forana, la taula referenciada i la columna referenciada, per aquest ordre.

```
[FKQuery2]
SELECT column_name, ordinal_position,
       referenced_table_name,
       referenced_column_name
FROM   key_column_usage
WHERE  table_schema='#DATABASE#'
AND    table_name='#TABLENAME#'
AND    constraint_name='#FKNAME#';
```

Secció “Uquery”, al igual que amb les claus foranes, l’obtenció de la definició completa de les restriccions d’unicitat definides a una taula, la realitzarem en dos etapes, en la primera, aquesta, obtindrem el nom d’aquest tipus de restriccions.

```
[UQuery]
SELECT constraint_name
FROM   table_constraints
WHERE  table_schema='#DATABASE#'
AND    table_name='#TABLENAME#'
AND    constraint_type='UNIQUE';
```

Secció “Uquery2”, finalment, obtindrem la composició de les restriccions d’unicitat que hem obtingut amb la consulta anterior. Cal obtenir el nom de la columna participant de la restricció i la seva posició a la restricció.

```
[UQuery2]
SELECT column_name, ordinal_position
FROM   key_column_usage
WHERE  table_schema='#DATABASE#'
AND    table_name='#TABLENAME#'
AND    constraint_name='#UKNAME#';
```

Amb el fitxer de configuració generat, tant sols cal que coneguem un nom d’usuari per accedir al SGBDR i el seu mot de pas.

Per executar l’aplicació cal que emprem la sintàxi:

```
Java -cp .:fitxerDriverJDBC RetroManager fitxerConfiguracio [baseDades] [Usuari]
```

On les opcions són:

```
-cp .:fitxerDriverJDBC
```

Indica el classpath on Java cercarà les classes, cal indicar-hi en aquest paràmetre el nom del fitxer amb el controlador JDBC adient per al SGBDR al que volem connectar-nos-hi.

fitxerConfiguracio

Indica el fitxer de configuració, amb les cadenes de declaració, connexió y les consultes d'SQL, adient per al SGBDR al que volem connectar-nos-hi.

[baseDades]

Indica quina base de dades dins del SGBDR volem analitzar. El paràmetre te entrada opcional a la línia del punt indicatiu, si no ho passem com paràmetre el programa ens ho sollicitarà.

[Usuari]

Indica un nom d'usuari amb suficients permisos com per poder realitzar consultes al catàleg. El paràmetre te entrada opcional a la línia del punt indicatiu, si no ho passem com paràmetre el programa ens ho sollicitarà.

Un exemple de sintaxi per al SGBDR mySql 5 és:

```
java -cp .; mysql-connector-java-3.1.12-bin.jar RetroManager mySql.txt proves root
```

Una vegada executada l'ordre, l'aplicació ens demanarà els paràmetres optatius que pot ser no hem inclòs a la instrucció per executar l'aplicació, si hem especificat tots els paràmetres, com a mínim, ens sol·licitarà el mot de pas per l'usuari declarat.

Una vegada introduït el mot de pas, el programa no demana més entrades per pantalla y finalitza la seva execució presentant un informe per pantalla.

Conclusions i què és el que ha quedat per fer.

Amb la determinació de mecanismes estrictes per a les etapes de transformació d'esquemes, dins del procés de disseny d'una base de dades relacional, podem assegurar el manteniment semàntic de la base de dades, sigui quin sigui el nivell en que es trobem.

He pogut comprovar, malgrat les idees inicials, que el procés d'enginyeria inversa aplicat sobre bases de dades relacionals és completament determinista.

Malgrat que aquest àmbit de treball m'ha deixat prou satisfet, altres aspectes que hi eren a la planificació, com l'ús de DresdenOCL per extreure un diagrama UML amb definicions d'OCL incloses, no han pogut ser abastats.

Igualment, l'aplicació té aspectes força millorables, particularment en tot allò que te a veure amb la independència de plataforma, amb l'extensibilitat del programa. La definició d'un fitxer de paràmetres que serveixi per emmagatzemar també les consultes que proveiran d'informació a l'aplicació, és una solució millor que la plantejada inicialment que tenia nou fitxers amb paràmetres, però, el mantenir un fitxer auxiliar de l'aplicació, actualitzable per l'usuari amb l'ús dels paràmetres adjunts a la línia de comandes (o fins i tot, a través d'Internet) hagués millorat molt la qualitat de l'aplicació.

Valoració personal.

Circumstàncies imprevistes, alienes al projecte, van requerir molt del temps que estava planificat per dedicar al treball. Aquests imprevistos van posar de manifest que són el punt feble de tota planificació.

Amb menys temps del previst i, sobretot, amb jornades intensives per tal de dedicar-hi tant de temps com fos possible el desenvolupament del treball s'ha ressentit força.

Malgrat tot, l'estudi realitzat sobre les etapes de transformació d'esquemes, i el trencaclosques que va suposar esbrinar quines eren les preguntes adjunts que calia fer per determinar el caràcter primigeni d'una taula en una base de dades, va resultar, finalment molt satisfactori.

Glossari.

Atribut d'una entitat

Propietat que interessa d'una entitat.

Atribut d'una interrelació.

Propietat que interessa d'una interrelació.

Connectivitat d'una interrelació.

Expressió del tipus de correspondència entre les ocurrències d'entitats associades amb la interrelació.

Disseny conceptual.

Etapa del disseny d'una base de dades que obté una estructura de la informació de la futura BD independent de la tecnologia que es vol emprar.

Disseny físic.

Etapa del disseny d'una base de dades que transforma l'estructura obtinguda a l'etapa del disseny lògic amb l'objectiu d'aconseguir una major eficiència i que, a més, la completa amb aspectes d'implementació física que dependran de l'SGBDR que s'ha d'utilitzar.

Disseny lògic.

Etapa del disseny d'una base de dades que parteix del resultat del disseny conceptual i el transforma de manera que s'adapti al model de l'SGBDR amb el qual es desitja implementar la base de dades.

Enginyeria inversa.

Procés mitjançant el qual, obtenim els condicionants i especificacions que van determinar les característiques que té un producte dissenyat.

Entitat.

Objecte del món real que podem distingir de la resta d'objectes i del qual ens interessin algunes propietats.

Entitat associativa.

Entitat resultat de considerar una interrelació entre entitats com una nova entitat.

Entitat dèbil.

Entitat els atributs de la qual no la identifiquen completament, sinó que no més la identifiquen de manera parcial.

Entitat obligatòria en una interrelació binària.

Entitat tal que una ocurrència de l'altra entitat que intervé en la interrelació només pot existir si es dóna com a mínim una ocurrència de l'entitat obligatòria a que hi està associada.

Entitat opcional en una interrelació binària.

Entitat tal que una ocurrència de l'altra entitat que intervé en la interrelació pot existir encara que no hi hagi cap ocurrència de l'entitat opcional a que hi està associada.

Generalització/especialització.

Construcció que permet reflectir que existeix una entitat general, que anomenem *entitat superclasse*, que es pot especialitzar en entitats subclasse. L'entitat superclasse ens permet modelitzar les característiques comunes de l'entitat vista a un nivell genèric, i amb les entitats subclasse podem modelitzar les característiques pròpies de les seves especialitzacions.

Grau d'una interrelació.

Nombre d'entitats que associa la interrelació.

Interrelació.

Associació entre entitats.

Interrelació recursiva.

Interrelació a la qual alguna entitat està associada més d'una vegada.

Bibliografía.

C.J.Date. Introducción a los Sistemas de Base de Datos. Volumen 1. Quinta edición. Editorial Addison, Wesley, Longman.

Adoración de Miguel y Mario Piattini. Fundamentos y Modelos de Bases de Datos. Segunda edición. Editorial Ra-ma.