

## Restriccions

Les restriccions són regles que es vol imposar per protegir la base de dades, de manera que no es pugui arribar a un estat inconsistent en el qual les dades no reflectisquen la realitat o siguin contradictòries.

Perquè una **base de dades** siga **vàlida**, s'han de complir totes les restriccions d'integritat que tinga definides.

Recordem que hi ha diferents **tipus de restriccions**:

- **Inherents** al model (formen part de forma implícita).

- Cada taula té un nom diferent.
- Cada atribut de la taula pren un únic valor a cada fila.
- Cada atribut té un nom diferent a cada taula (encara que pot coincidir en taules diferents)
- Cada fila és única (no hi ha tuples duplicades).
- L'ordre dels atributs no importa.
- L'ordre de les files no importa

- **Restriccions semàntiques** i regles de negoci:

Qualsevol operació que es realitzi sobre les dades ha de complir les restriccions i les regles que imposa el funcionament de l'empresa. Parlem de restriccions quan es donen certes condicions que no han de violar i parlem de regles de negoci quan es requereix l'execució automàtica de certes accions davant determinats esdeveniments.

--> **Restriccions de domini**: Tots els atributs tenen un domini associat, que és el conjunt de valors que cada atribut pot prendre.

Exemple: "El tipus de publicació únicament pot ser Novel·la, Conte, Teatre o Poesia".

Dom\_tipus : {Novel·la, Conte, Teatre, Poesia, ...}

Les restriccions dels dominis són la forma més simple de la restricció d'integritat.

S'especifica un domini de valors possibles per a cada atribut.

Una definició adequada de les restriccions dels dominis no només permet verificar els valors introduïts a la base de dades sinó també examinar les consultes per assegurar-se que tinguen sentit les comparacions que facen

• **Restricció de clau principal (PRIMARY KEY)** : Serveix per indicar les columnes que serveixen per identificar cada fila. Ho indiquem subratllant el nom de la/les columna/es. Obliga a omplir valors obligatòriament a cada fila. No permet repetir valors en aquests atributs.

» **Restricció d'entitat**: L'identificador d'una entitat no pot ser nul, per tant, les claus primàries de les taules no admeten nuls. Això és perquè aquesta clau s'empra per identificar tuples individuals en una relació. Si es permetera aquest valor, no seria possible diferenciar-les → no es podrien identificar certes tuples

• **Unicitat (UNIQUE)**: Els atributs marcats així no poden repetir valors. Ho indiquem amb un asterisc sobre el nom de la columna.(\*)

- **Obligatorietat (NOT NULL):** Alguns atributs han de contindre valors en tot moment, és a dir, **no admeten nuls**.

» Clau alternativa (Alternate Key): A gairebé cap gestor de base de dades existeix com a tal. Els atributs que són claus alternatives es marquen amb restricció Unique i Not Null.

- **Integritat referencial (FOREIGN KEY):** Marca les claus secundàries. Restringeix els valors de manera que les claus secundàries no poden prendre valors que no existisquen a la clau principal, per exemple, no podem marcar una nota a un núm. d'alumne que no existisca.

» Una clau secundària o aliena enllaça cada fila de la taula filla o dependent amb la fila de la taula mare que té el mateix valor en la seua clau primària. La integritat referencial diu que si una clau aliena té valor (si és no nul·la), aquest valor ha de ser un dels valors de la clau primària a la qual referència.

### Problemes:

- Si esborrem o modifiquem claus principals, les seues secundàries relacionades quedarien sense complir la restricció
- Per gestionar aquest problema les bases de dades tenen **polítiques especials** en **eliminar** i al **modificar** (les polítiques poden ser diferents)

Per aconseguir que es complisca la integritat referencial, hem de tindre en compte:

**1. Admet nuls la clau aliena?** Cada clau aliena expressa una relació. Si la participació de l'entitat filla en la relació és obligatòria (cardinalitat mínima 1), llavors la clau aliena no admet nuls; si és opcional (cardinalitat mínima 0), la clau aliena ha d'acceptar nuls.

**2. Què fer quan es vol esborrar o modificar una ocurrència (fila) de l'entitat mare que té alguna filla?** Això és el mateix que preguntar-se què fer quan es vol esborrar una fila que està sent referenciada per una altra fila a través d'una clau aliena. Hi ha diverses respostes possibles (polítiques):

- **Restringir (no deixar fer res, RESTRICT):** no es poden esborrar/modificar files (en concret la clau principal) si estan relacionades amb altres files.
- **Propagar (en cascada, CASCADE):** s'esborra/modifica la fila desitjada (clau principal) i es propaga l'esborrat/modificació a totes les files que li fan referència (amb les claus secundàries relacionades).
- **Anul·lar (posar nuls, SET NULL):** s'esborra/modifica la fila desitjada (clau principal) i totes les referències que tenia (claus secundàries) es posen, automàticament, a nul (**aquesta opció només és vàlida si la clau aliena accepta nuls**).
- **Valor per defecte (SET DEFAULT):** s'esborra/modifica la fila desitjada (clau principal) i totes les referències (claus secundàries) prenen, automàticament, el valor per defecte (**aquesta opció només és vàlida si s'ha especificat un valor per defecte per a la clau aliena**).

El més habitual en gairebé tots els sistemes per solucionar el problema de modificar o eliminar claus primàries són els canvis en cascada o bé no fer res.

## Exemple – Actualització en cascada o propagació

Alumno	Cod Alumno
Juanjo	1
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
1	1	9
1	2	9
1	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

### Actualizar en cascada

Alumno	Cod Alumno
Juanjo	7
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
7	1	9
7	2	9
7	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

## Exemple – Eliminació en cascada o propagació

Alumno	Cod Alumno
Juanjo	1
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
1	1	9
1	2	9
1	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

Alumno	Cod Alumno
Juanjo	1
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
1	1	9
1	2	9
1	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

### Exemple – Eliminació a nuls

Alumno	Cod Alumno
Juanjo	1
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
1	1	9
1	2	9
1	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

Alumno	Cod Alumno
Juanjo	1
Ana	2
Sonia	3

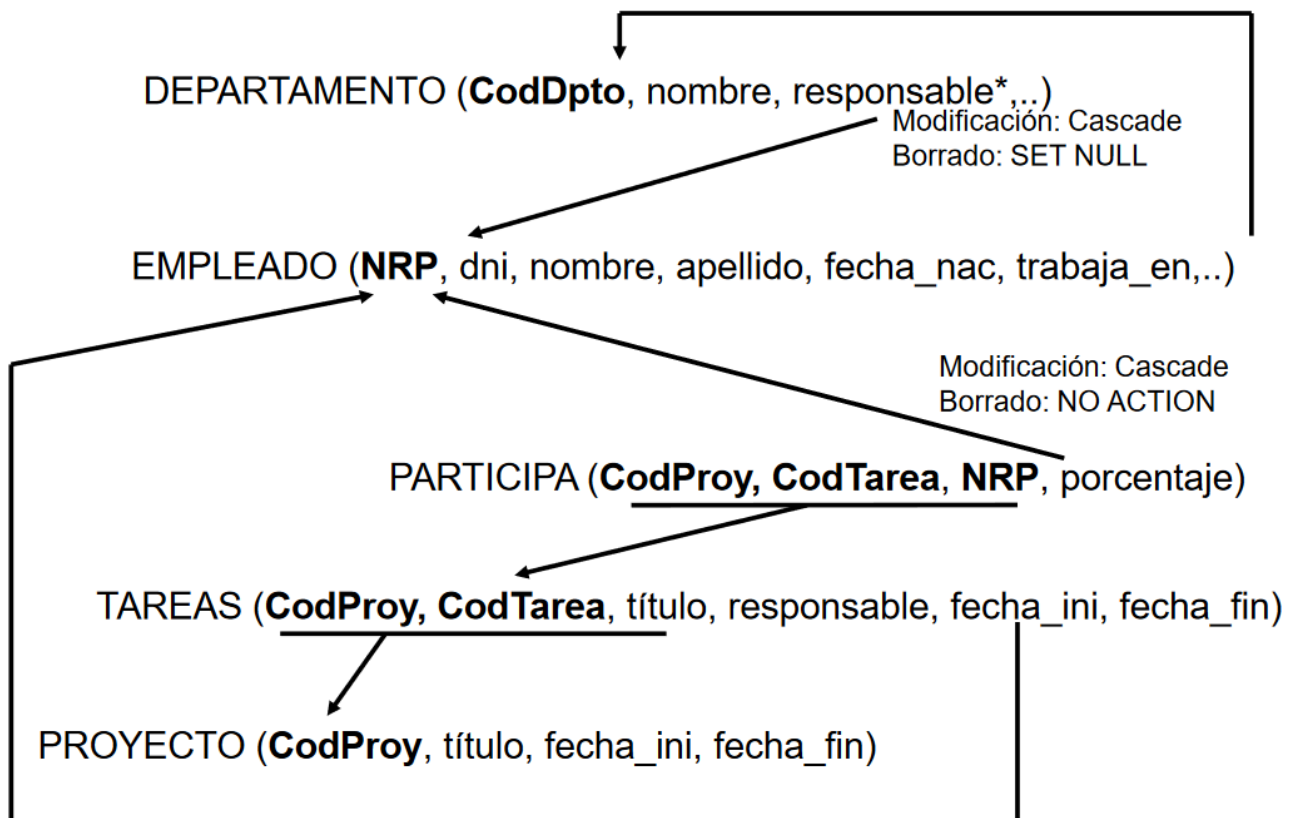
Cod alumno	Cod asignatura	Nota
(null)	1	9
(null)	2	9
(null)	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

### Exemple – Eliminació valor per defecte

Alumno	Cod Alumno
Juanjo	1
Ana	2
Sonia	3
Desconocido	999

Cod alumno	Cod asignatura	Nota
999	1	9
999	2	9
999	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

Indicarem en el model relacional la integritat referencial indicant l'operació (modificat o esborrat i la política). Exemple:



- **Validació o verificació (CHECK):** Prohibeix que s'afegisquen a la base de dades, valors que incomplisquen una determinada condició. Per exemple que l'edat siga menor de 18 anys.

- **Asserts i Triggers:** els asserts i els disparadors (triggers) permeten implementar també restriccions d'integritat. Aquestes poden ser:

- Restriccions d'integritat estàtiques (CREATE ASSERTION...). És semblant a els CHECKs , però es diferencia en què pot afectar diversos elements.

- Restriccions d'integritat de transició (disparador o TRIGGER): s'executa quan passa un determinat esdeveniment a la base de dades. Permet imposar condicions molt elaborades i complicades (però útils). Per exemple:

- No deixar afegir un DNI la lletra del qual no es corresponga amb els números (després d'aplicar la complexa fórmula del DNI)

- No permetre afegir dades entre les 3 i les 6 de la vesprada.

Totes les restriccions d'integritat establertes en aquest pas s'han de reflectir en la documentació de l'esquema lògic perquè puguin ser tingudes en compte durant la fase de el disseny físic.