

BD Primer Parcial

1. Introducció

Els tres mons

Podem dividir-ho tot en 3 mons ben diferenciats:

Ordre	Nom	Descripció	Exemples
1	Món real	Objectes, persones i organitzacions reals	UPC, professors...
2	Món conceptual	Coneixements, informacions	UML, esquemes
3	Món de les representacions	Dades	Taules en bases de dades

Parts món conceptual

- **Classe d'objectes:** Descriu un conjunt d'objectes.
- **Atribut:** Propietat compartida pels objectes d'una classe.
- **Associació:** Interrelació entre classes d'objectes.

1.1 BD i SGBD

Definició de base de dades (BD):

Una base de dades és una col·lecció de dades que permet representar classes d'objectes, els seus atributs i les seves associacions i que està gestionat per un SGBD

1.1.1 Els SGBD ens proporcionen

- Persistència
- Eficiència en l'accés
- Emmagatzematge de quantitats massives de dades
- Accés multi-usuari
 - Diversos usuaris accedint concurrentment a una mateixa DB.

- Per gestionar aquests accesos als SGBD utilitzen **transaccions**.
- Una transacció és un conjunt d'operacions que s'intenten executar, però només ho acaben fent si **totes** les operacions se poden executar, sino es torna al estat anterior.
- Seguretat
 - Identificació i autenticació dels usuaris.
 - Possibilitat de definir autoritzacions o drets d'accés.
- Fiabilitat
 - **Regles d'integritat**: Condicions que les dades han de complir. Son definides per el SGBD.
 - **Regles d'integritat dels usuaris**: Condicions que han de complir les dades definides pels usuaris, hi han els dos següents típus:
 - Estàtiques: La seva avaluació depèn d'un sol estat de la BD.
 - Dinàmiques: La seva avaluació depèn de diversos estats de la BD.
 - Redundàncies controlades
 - Transaccions
 - No pèrdua de dades
- Conveniència
 - Flexibilitat als canvis
 - Canvis conceptuals
 - Canvis a l'enregistrament físic de les dades
 - **Arquitectura ANSI/SPARC**: Està orientada a l'obtenció de la independència física i lògica de les dades. Tres nivells:
 - Nivell Extern
 - Nivell Conceptual
 - Nivell Intern

1.1.2 Usuaris de BD

- **Usuaris informàtics:** Dissenyadors de DB, Programadors, Administradors...
- **Usuaris no informàtics:**
 - **Usuaris paramètrics:**
 - Usen programes o aplicacions creats pels informàtics.
 - No han de conèixer la BD ni cap llenguatge de BD.
 - **Usuaris finals:**
 - Necessiten realitzar ocasionalment consultes (no previstes)
 - Usen un "llenguatge" de consulta d'alt nivell per a programar-les
 - Han de conèixer part de l'estructura de la BD i el llenguatge de consulta.

2. El Model Relacional

Objectius:

1. Facilitar que la BD sigui percebuda de manera independent a la implementació.
2. Proporcionar una estructura lògica simple i uniforme: Conjunt relacions atòmiques

2.1 Les Relacions

Algunes definicions

Una **tupla** es una "entrada" en una relació. Una "Fila".

Els **atributs** són els paràmetres que contenen cada tupla d'una relació.

La **cardinalitat** d'una relació és el nombre de tuples que conté.

El **grau** d'una relació és el nombre d'atributs.

La **superclau** d'una relació és el subconjunt d'atributs que identifiquen una tupla.

La **clau** d'una relació és aquella superclau que no té superclaus dins del subconjunt.

També se la pot anomenar **clau candidata**.

Una i només una clau es designa com a **clau primària**, la resta s'anomenaran **claus alternatives**.

Una **clau forana** és una clau que fa **referència** a una **clau primària**.

2.2 Les operacions

Manipulació de dades:

- Actualització
 - **Inserció** de tuples a una relació.
 - **Esborrat** de tuples d'una relació.
 - **Modificació** de tuples d'una relació.
- Consulta: Obtenció de dades deduïbles a partir de les relacions.

Segons la manera de classificar les consultes el llenguatge relacional es poden dividir:

- Llenguatges basats en l'àlgebra relacional
- Llenguatges basats en el càlcul relacional

2.3 Regles d'integritat

La regla d'integritat d'identitat: les **claus primàries** han de ser **úniques** i no **nules**.

La regla d'integritat referencial: les **claus foranes** han de contenir valors de la **clau primària referenciada** o **valors nuls**. Per a mantenir-la es realitza el següent:

Dos opcions:

- Es **rebutja l'operació**: Restricció o anul·lació.
- S'accepta i per tant es realitzen **accions compensatòries**: Cascada

Es poden produir en dos contextes:

- **Esborrat d'una clau primària**
- **Modificació d'una clau primària**

Aplicació en **restricció**:

No es pot esborrar o modificar una clau prim. referenciada per una clau forana

Aplicació en **cascada**:

En cas d'esborrar o modificar una clau primària referenciada per alguna clau forana, s'esborren o es modifiquen totes les referències.

Aplicació en anul·lació:

En cas d'esborrar o modificar una clau primària referenciada en alguna clau forana, s'assignen valors nuls a totes les referències.

3. Llenguatges

3.1 Àlgebra relacional

Unió

$R = \text{PERSONAL-ADM} \cup \text{PERSONAL-LAB}$

$R = T \cup S$

- Serveix per fer una unió de taules amb tuples similars

Reanomenament

$R = \text{PERSONAL-DOC} \{ \text{modul-de} \rightarrow \text{modul}, \text{num-de} \rightarrow \text{num} \}$

$R = T \{ \text{atr11} \rightarrow \text{atr21}, \dots \}$

- Serveix per a reanomenar les columnes per a fer un select/projecció

Intersecció

$R = \text{PERSONAL-ADM} \cap \text{PERSONAL-LAB}$

$R = T \cap S$

- Serveix per a quedar-se amb les files que siguin iguals en dos taules

Diferència

$R = \text{PERSONAL-ADM} - \text{PERSONAL-LAB}$

$R = T - S$

- Serveix per a quedar-se amb els elements de T que no son a S

Producte Cartesià

R =MODUL-CN × OFICINA

$R = T_x_S$

- Us **molt limitat**, serveix per a crear totes les *combinacions*

Selecció

R =OFICINA(modul-de='B2' AND superfície>16)

$R = T$ (comparacions separades per operadors lògics)

- Serveix per seleccionar una serie de files en concret segons certs operadors lògics
- Es pot usar: <, <=, >, >=, <>, =, !=, AND, OR

Projecció

R =PERSONAL-ADM[modul, num]

$R = T$ [atr11, atr12,...]

- Serveix per a "seleccionar" unes certes columnes d'una determinat conjunt

Combinació (join)

R =MODUL-CN[modul=modul-de, sup-promig-of<=superfície]OFICINA

$R = T$ [comparacions separades per comes] S

- Serveix per a fer combinacions de dos conjunts amb certes limitacions lògiques.

Natural Join implícita i explícita

- Implícita

R =PERSONAL-ADM * MODUL-CN

$R = T * S$

- Serveix per a combinar igualant els atributs que tinguin el mateix nom

- Explícita

R =PERSONAL-ADM[modul*modul-de. num*num-de]OFICINA

R = T [combinacions separades per comes] S

- Serveix per a combinar igualant els atributs seleccionats amb el mateix nom
- Tots els operadors han de ser *

3.2 SQL

Create Table

CREATE TABLE

([], ...

);

El valor per defecte pot ser un o NULL

- restriccions_col

]

Nom restricció	Descripció
UNIQUE	La columna no pot tenir valors repetits
PRIMARY KEY	La columna és clau primària de la taula
REFERENCES [
La columna és clau forana que referencia la taula idicada	
CHECK ()	La columna ha de complir les condicions . Només es pot referir a la columna per la que es declara .
NOT NULL	La columna no pot tenir valors nuls

- restriccions_taula

Nom restricció	Descripció
UNIQUE ()	Les columnes no poden tenir valors repetits
PRIMARY KEY ()	Les columnes son la clau primària de la taula
FOREIGN KEY () REFERENCES []	Les columnes son una clau forana que referencia la taula idicada
CHECK ()	La taula ha de complir les condicions . Només es pot referir a la taula per la que es declara.

Insert into

INSERT INTO [()]

(VALUES { | NULL, ...}) | ;

- Els valors s'han de posar en el mateix ordre que a la taula a la que s'insertaran

Delete from

DELETE FROM

WHERE ;

- S'eliminen les files de la taula que compleixin les condicions del WHERE

UPDATE

UPDATE

SET = {expressió|NULL}[, ...]

WHERE ;

- Es modifiquen les columnes de les files que compleixen les condicions

Consultes bàsiques

SELECT num_empl, nom_empl, sou # Selecciona certes columnes de les taules

FROM empleats # Selecciona les taules sobre les que es faran les operacions

WHERE num_dpt = 3; # Selecciona les files que compleixin la condicio

- Operadors, els evidents de tot llenguatge de programació (!= es <>) +:

- BETWEEN AND
- IN (, [...,])
- LIKE <característica>
- IS [NOT] NULL

Order by

ORDER BY [DESC|ASC],...;

- Simplement ordena la sortida descendentment o ascendentmet segons columna

Evitar Repeticions

SELECT [DISTINCT|ALL]

- Si posem DISTINCT només seleccionarà "una de cada"

Duncions d'agregació

S'especifiquen a la clausula WHERE

- **COUNT:**

COUNT (*): compta totes les files que compleixen la condició

COUNT(DISTINCT): Compta els valors diferents de la columna sense nul

COUNT(): Compta els valors de la columna sense comptar nuls.

- **SUM** (expressió), **MIN** (expressió), **MAX** (expressió), **AVG** (expressió):

expressió pot ser un, càcul o simplement una columna.

SUM dona la suma dels valors resultants de la condició where

MIN dona el mínim dels valors resultants de la condició where

MAX dona el mínim dels valors resultants de la condició where

AVG dona el promitj dels valors resultants de la condició where

- Compta el nombre de files que compleixen la condició del where

Group By

GROUP BY ;

- Divideix en grups les files que compleixen les condicions del where

Having

HAVING ;

- Només apareixeran els grups que compleixin la condició del having

Consultes a mes d'una taula

FROM ,

- Aixó fa una natural join implícita

Union

SELECT

...

UNION

SELECT

...

- El resultat es la unió dels dos resultats del select

NOT IN | NOT EXISTS

WHERE [NOT IN | NOT EXISTS] (...);

- NOT IN no és cert si en el resultat del where no hi es la columna_taula.
- NOT EXISTS és cert si la subconsulta no dona cap resultat.

4. Components lògics d'una base de dades

Una BD es pot dividir en quatre parts:

1. **Servidor:** pot contenir 0 o mes catàlegs.
2. **Catàleg/s:** És un grup d'esquemes, un dels quals es el **information schema**.
3. **Esquema/s:** És un conjunt de vistes que conté, taules, vistes, etc.

- L'**information schema** es un exemple concret d'esquema, que conté vistes de totes les dades SQL que pertanyen al catàleg corresponent. O sigui es un esquema d'esquemes (meta-dades)
- Podem crear esquemes amb la sentència: CREATE SCHEMA.
- Podem borrar esquemes amb la sentència: DROP SCHEMA
 - Si la executem amb RESTRICT només borrarà el schema si és buit.
 - Si la executem amb CASCADE borrarà el schema encara que no ho sigui.

4. Taules, vistes, procediments, etc...

Conceptes importants

Una **connexió** es pot definir com l'associació que es crea entre un client i un servidor.

- Usem CONNECT TO per connectar-nos
- Usem DISCONNECT per desconectar-nos

Una **sessió** son el conjunt de sentències que s'executen mentre hi ha una connexió.

- Usem SET SESSION CHARACTERISTICS AS per configurar la sessió.

Una **transacció** és el conjunt de sentències que volem que s'executin alhora.

- Usem SET TRANSACTION per a definir les seves característiques.
- Les transaccions poden acabar amb un COMMIT (si no hi ha hagut cap error).

O poden acabar amb un ROLLBACK (si hi ha hagut algún error).

Un **domini** és un conjunt de valors vàlids definits per l'usuari.

- Usem CREATE DOMAIN per ha crear un domini.
- Usem CONSTRAINT "nom restricció" CHECK "condició" per a restringir el domini.

Una **taula** és el component base d'una BD.

- Es pot definir amb CREATE TABLE "nom_taula" i un seguit de columnes.
- Les restriccions de taula son les mateixes que les de columna, llevat del NOT NULL, i només s'han d'usar quan hi han restriccions de columna que afecten a varies.

Una **columna** es pot definir amb: "nom_columna" "tipus_dades" [restricció_col]

- Les restriccions de columna poden ser:

NOT NULL: La columna no pot tenir valors nuls.

UNIQUE: La columna no pot tenir valors repetits.

PRIMARY KEY: La columna és clau primària de la taula.

REFERENCES "nom_taula" ["nom_columna"]: La columna és clau forana.

[CONSTRAINT "nom_restricció"] CHECK (condicions)

- La columna ha de complir les condicions especificades.
- Típicament són de rang del domini
- És opcional el donar nom a aquestes restriccions

Una **assertion** és una restricció d'integritat que afecta a diverses taules

- No estan definides en la majoria de sistemes actuals.

Una **vista** és una relació derivada el seu esquema i contingut es deriva d'altres.

- **CREATE VIEW** "nom_vista" ["nom_col1, ..."] **AS** "SELECT..." [**WITH CHECK OPTION**]
- **WITH CHECK OPTION** fa que les files es comprovin si són vàlides abans de insertar.
- Són actualitzables les següents vistes:
 - Les que fan **SELECT** sobre una única relació sense func. agregades ni **DISTINCT**.
 - Els atributs del **SELECT** han d'incloure tots els **NOT NULL** i que no tinguin valor
 - No pot haver-hi **GROUP BY**

Privilegis

Els privilegis es donen a un usuari/ grup d'usuaris per a fer una operació sobre un objecte en concret.

Es donen amb **GRANT** i **REVOKE**.

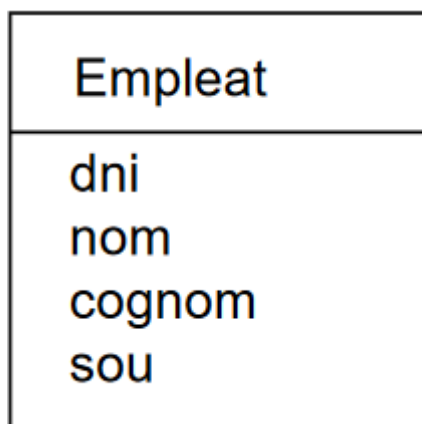
- **GRANT** "privilegis" ON "objectes" TO "usuaris" [**WITH GRANT OPTION**]
- **REVOKE** [**GRANT OPTION FOR**] "privilegis" ON "objectes" FROM "usuaris" {C|R}

- Es pot treure en CASCADE (C) revocant tots els privilegis concedits per l'usuari.
- Se li pot treure en RESTRICT (R) no es permet revocar si es treies algun altre.

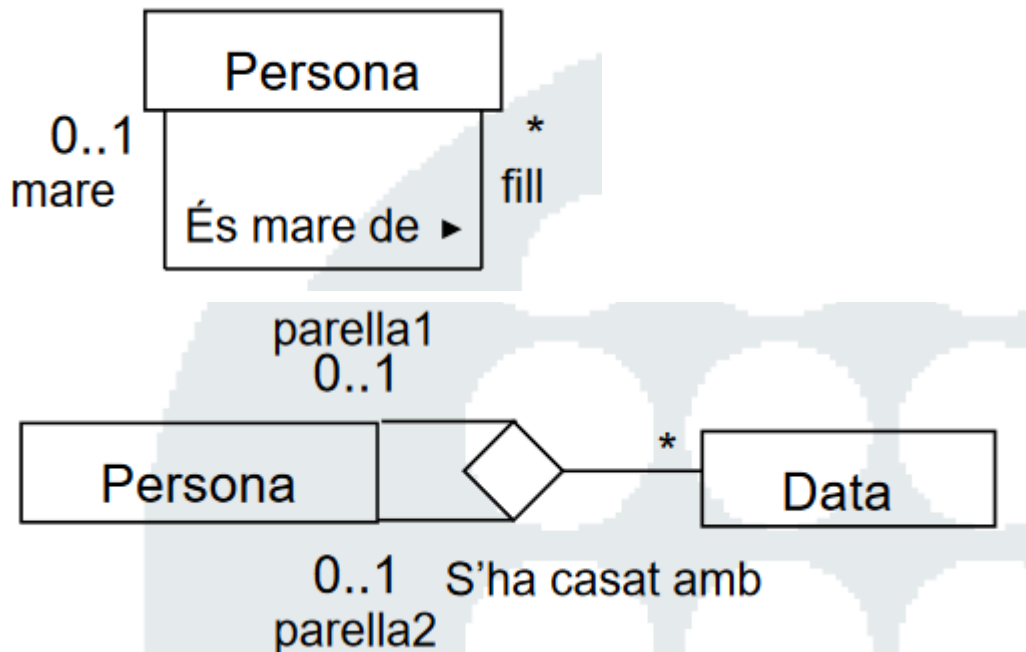
Nosaltres tractem 6 tipus de privilegis:

1. SELECT
2. INSERT
3. UPDATE
4. DELETE
5. REFERENCES
6. ALL

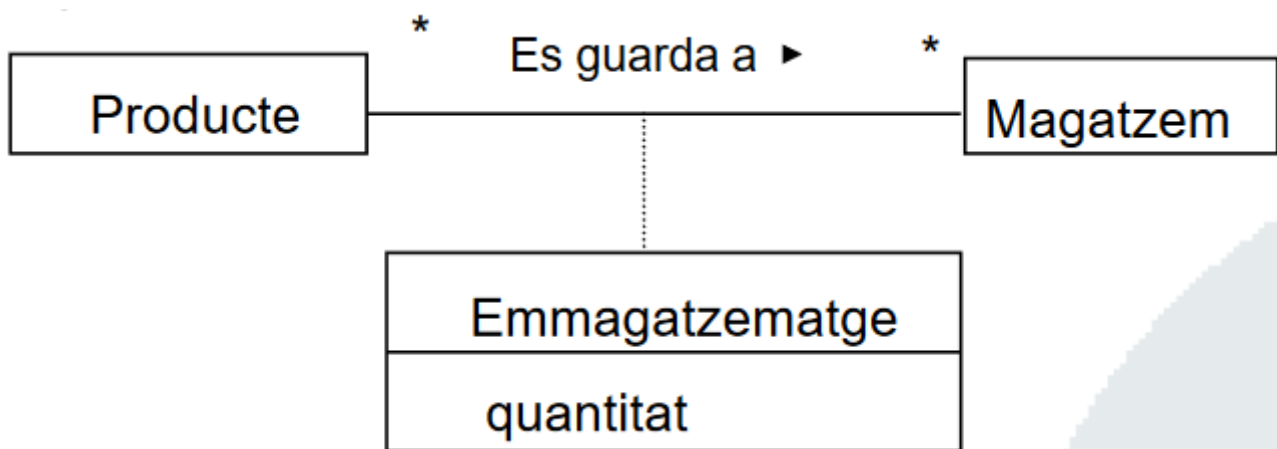
7. Introducció al disseny de bases de dades realcionals



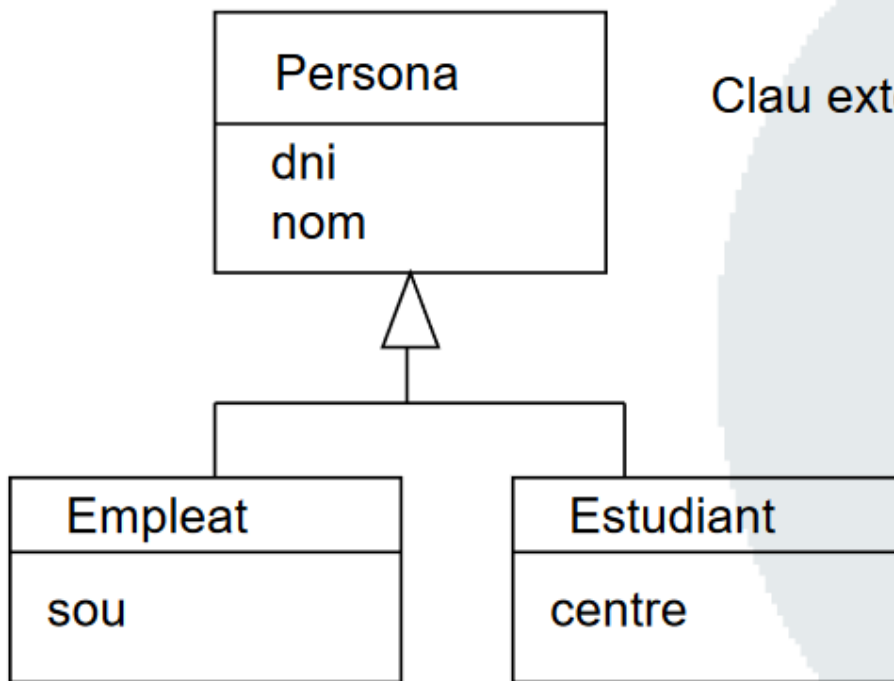
- **Classe d'objectes:** Descriu un conjunt d'objectes similars (Empleat).
- **Atribut:** Propietat compartida pels objectes d'una classe. (dni, nom...)
- **Clau Externa:** Conjunt d'atributs mínims que permeten identificar els objectes d'una classe. En UML no hi ha manera d'explicar-ho, només textualment
- **Les associacions:** Es la representació de la relació entre dos o més objectes.
- **La multiplicitat:** Son dos nombres que identifiquen el mínim i màxim d'elements amb els que s'emparella una opció.
- **Les associacions recursives** son aquelles en les que una mateixa classe d'objectes hi participa més d'una vegada:



- **Classes associatives:** Representa una associació que es pot veure com a classe:



- **La generalització/especialització:** Permet reflectir el fet de les sper/subclases.



Clau externa Persona: dni

Elements del model de partida	Transformació al model relacional
Classe d'objectes	Relació
Associació binària: un a molts	Clau forana al costat molts
Associació binària: un a un	Clau forana a qualsevol costat + Unique al altre
Associació binària: molts a molts	Relació Nova
Associació ternària: m - m - m	Primary Key a les tres bandes
Associació ternària: m - m - u	Primary Key a les dos bandes *
Associació ternària: m - u - u	Primary Key a la banda * + u + unique (* + u)
Associació n-ària	Relació Nova
Associació recursiva	Com les no recursives
Classe associativa	Només s'ha de transformar
Generalització/especialització	Relació per la super i per cadascuna de les sub