

DISSENY I ADMINISTRACIÓ DE BASES DE DADES

CONTROL 1

Data: 1 d'abril de 2019

Temps: 2 hores

Problema 1 [3 punts; màxim: 45 minuts]

Ens han encarregat programar una nova versió del sistema operatiu UNIX anomenada UNIX-RELATIONAL on la informació del sistema es guardarà en una base de dades relacional.

En aquesta primera versió caldrà gestionar usuaris, grups d'usuaris i elements del sistema de fitxers (fitxers, enllaços i directoris).

Dels usuaris guardem el seu username que és únic, la password, el grup principal al que pertany, el fitxer que conté la shell de l'usuari i el directori del seu home (directori personal). Dos usuaris no poden compartir el mateix directori personal. Dels grups d'usuaris només guardem el groupname que és únic. Un usuari pot pertànyer a varis grups d'usuaris.

Els elements del sistema de fitxers de l'UNIX-RELATIONAL com ja hem dit són de tres tipus: fitxers, enllaços i directoris. Tots ells tenen en comú el número d'inode que els identifica, el nom de l'element, la data del darrer accés, la data de la darrera modificació, l'usuari i el grup d'usuari al que pertany i el directori a on està situat (tots els elements estan situats en un directori; l'element directori arrel considerarem que està situat en el directori arrel). Els noms dels elements d'un mateix directori no poden estar repetits.

Els elements del tipus fitxer a més a més tenen un contingut (que pot ser text, binari, ..., això en aquesta versió no importa). Els elements de tipus enllaç guarden si l'enllaç és simbòlic o no i l'element al que enllaça (podem fer enllaços de fitxers, d'altres enllaços i de directoris). I els elements de tipus directori guarden el nombre d'elements que contenen.

Per últim, aprofitant que fem un sistema operatiu UNIX nou, es vol implementar els permisos de lectura, escriptura i execució dels elements a l'estil de les ACL (Acces Control Lists). O sigui volem definir els permisos dels elements segons el grup d'usuaris, indicant per cada grup d'usuaris si pot o no llegir-lo, si pot o no modificar-lo i si pot o no executar-lo.

Proposa un model de classes, enumerant aquelles ambigüitats que puguis detectar i indicant com les has resolt, i indicant també, si és el cas, quines condicions de l'enunciat no es reflecteixen de manera directa en el teu model.

DISSENY I ADMINISTRACIÓ DE BASES DE DADES

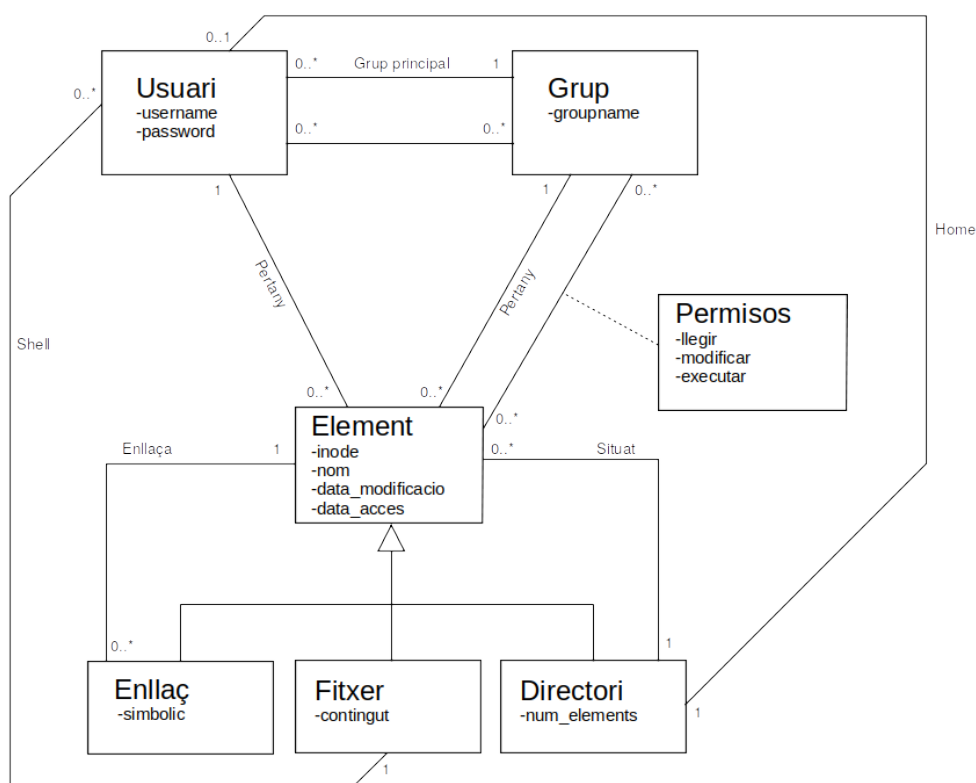
CONTROL 1

Data: 1 d'abril de 2019

Temps: 2 hores

Problema 2 [7 punts; màxim: 75 minuts]

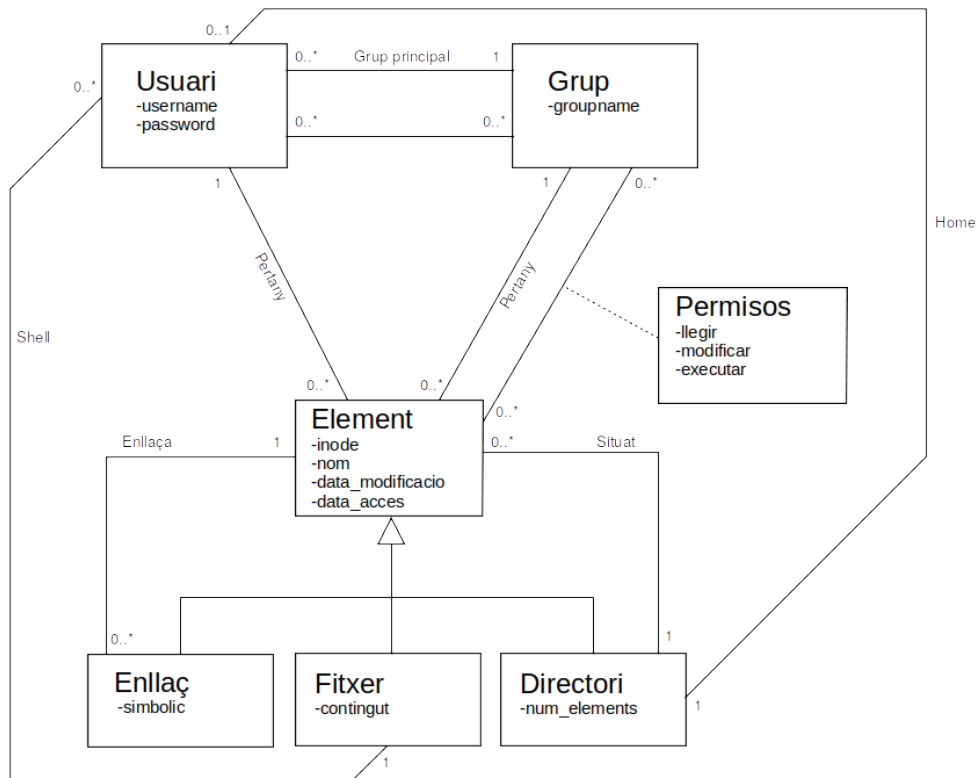
Considerem el diagrama següent:



- (3,5 punts) Descriu informalment les relacions adequades per transformar aquest diagrama en un esquema relacional usant els mateixos noms per les relacions i els atributs. Indica, per a cada relació, quins són els seus atributs, les claus primàries que tries, les claus alternatives si n'hi ha, i les claus foranes que necessita.
- (1,5 punts) Considera la relació que correspon a les classes "Usuari", "Grup", "Element" i "Permisos", tenint en compte només els atributs Nom d'Usuari-U, Nom de Grup-G, Inode de l'element-E i Permisos-P. Analitza les seves dependències funcionals, i indica si aquest esquema de relació amb els atributs (E,U,G,P) està o no en 3NF i si està o no en BCNF. Quan la resposta sigui negativa, proposa diferents formes de normalitzar i explica els avantatges i inconvenients d'aplicar cadascuna d'elles.
- (1 punt) Escriu el codi SQL de creació de les taules corresponents a les classes "Element" i "Directori" amb les polítiques ON DELETE i ON UPDATE adients.
- (1 punt) Escriu la instrucció SQL que permeti actualitzar el camp num_elements amb el nombre actual d'elements que conté cada directori.

Solució Problema 1

Una de les possibles solucions podria ser aquesta:



R.S.: No poden haver dos elements situats al mateix directori amb el mateix nom.

Solució Problema 2a

Llegenda:

- **Clau primària (PK) en negreta**
- *Clau alternativa (UK) en cursiva*
- Clau forana (FK) subratllada

usuari (**username**, password, grup FK a grup, fitxer FK a fitxer , directori FK a directori)
 grup (**groupname**)

usuari_grup (**username** FK a usuari, **groupname** FK a grup)

element (**inode**, nom, data_modificacio, data_access, username FK a usuari, groupname FK a grup, directori FK a directori)

permisos (**inode** FK a element, **groupname** FK a grup, llegir, modificar, executar)

enllaç (**inode** FK a element, simbolic, link FK a element)

fitxer (**inode** FK a element, contingut)

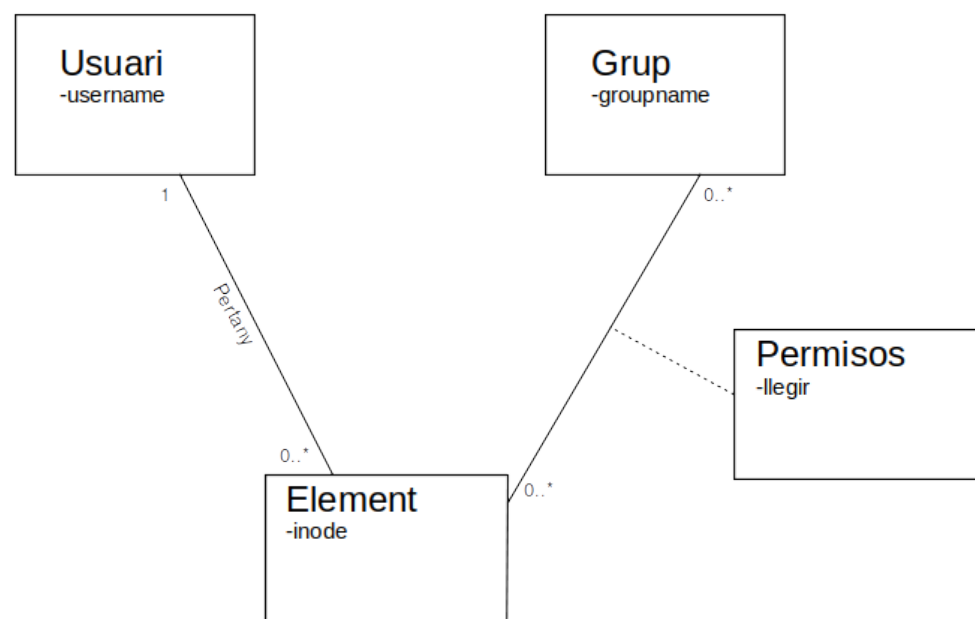
directori (**inode** FK a element, num_elements)

Nota: Si a l'enunciat del problema 2 hi hagués la R.S.: No poden haver dos elements situats al mateix directori amb el mateix nom, llavors caldria afegir una clau alternativa a la relació element formada pels atributs directori i nom.

Solució Problema 2b

Si ens donen la relació $R(E, U, G, P)$, el grup G que hi ha a cada tupla és el relacionat amb l'element E per definir els permissos P , no el grup principal de U o el grup al que pertany E . I l'usuari U és l'usuari al que pertany E .

Per tant la relació $R(E, U, G, P)$ només pot modelitzar aquesta part del diagrama UML:



Les dependències funcionals són:

- $E \rightarrow U$
- $E G \rightarrow P$

La relació $R(E, U, G, P)$ no està ni en 3NF ni en BCNF, doncs la primera dependència funcional no es podria posar com a clau de la relació (ni tampoc U pertany a una clau). Caldria separar-la amb dues relacions:

$R_1(E, U)$

$R_2(E, G, P)$

Ara estan en BCNF i per tant en 3NF.

Solució Problema 2c

```
CREATE TABLE element (  
    inode int NOT NULL PRIMARY KEY,  
    nom varchar(255) NOT NULL,  
    data_access datetime NOT NULL,  
    data_modificacio datetime NOT NULL,  
    username varchar(255) NOT NULL REFERENCES usuari  
        ON DELETE RESTRICT ON UPDATE CASCADE,  
    groupname varchar(255) NOT NULL REFERENCES grup  
        ON DELETE RESTRICT ON UPDATE CASCADE,  
    directori int NOT NULL REFERENCES directori  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    UNIQUE(nom, directori)  
);
```

```
CREATE TABLE directori (  
    inode int NOT NULL PRIMARY KEY REFERENCES element  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    num_elements int NOT NULL  
);
```

Nota: Tal com s'han definit les taules element i directori no seria possibles crear-les en un SGBD ja que cada taula conté una clau forana a l'altra taula, i no es pot definir una clau forana fins que la taula destí està definida. El correcte seria afegir una de les dues claus foranes a posteriori, així:

```
CREATE TABLE element (  
    inode int NOT NULL PRIMARY KEY,  
    nom varchar(255) NOT NULL,  
    data_access datetime NOT NULL,  
    data_modificacio datetime NOT NULL,  
    username varchar(255) NOT NULL REFERENCES usuari  
        ON DELETE RESTRICT ON UPDATE CASCADE,  
    groupname varchar(255) NOT NULL REFERENCES grup  
        ON DELETE RESTRICT ON UPDATE CASCADE,  
);  
CREATE TABLE directori (  
    inode int NOT NULL PRIMARY KEY REFERENCES element  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    num_elements int NOT NULL  
);
```

```
ALTER TABLE element ADD COLUMN directori int NOT NULL REFERENCES directori  
    ON DELETE RESTRICT ON UPDATE CASCADE;
```

Solució Problema 2d

```
UPDATE directori SET num_elements =  
    (SELECT count(*) FROM element WHERE element.directori=directori.inode);
```