ysalcedo@xtec.cat – Abril 2017

Respecte de la unitat 8, UF4, bases de dades objecte-relacionals

Al temari es veu una introducció i aplicació de l'Orientació a Objectes a les bases de dades originàriament relacionals, es defineixen i practiquen els conceptes fonamentals.

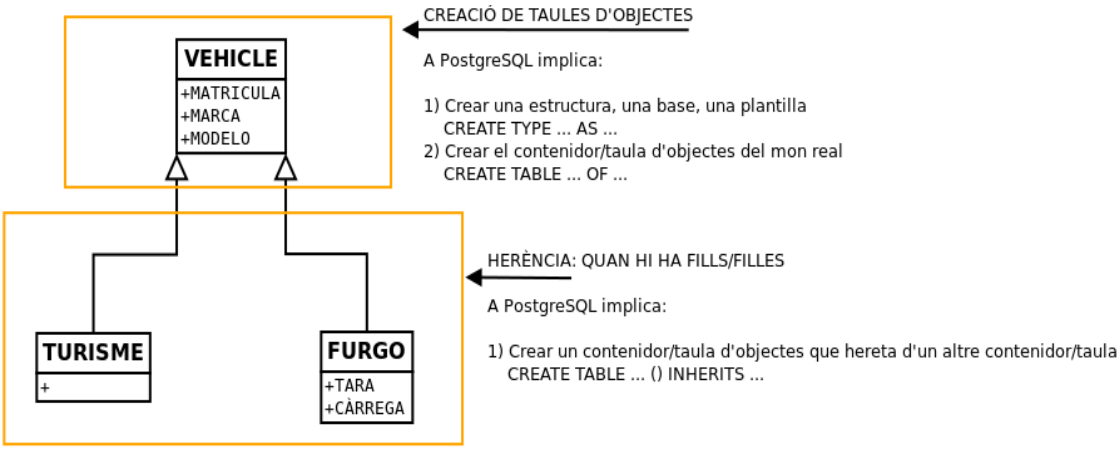
PostgreSQL no es defineix com a SGBD OO sinó que ha estat adaptat (a partir de la versió 9.2) per poder donar alternatives al paradigma de definició d'objectes. Per això es diu que és (pot ser) un SGBD Objecte-Relacional. Aquí les adaptacions bàsiques per fer això i que es veuen la temari:

- la creació de taules basades en un tipus de dada, és a dir, CREATE TABLE … **OF** ….
- i la creació de taules basades en altres (herència), és a dir, CREATE TABLE … **INHERITS**...

Asumits i entesos els conceptes de l'OO, aplicables amb diversos entorns de programació, centrats a PostgreSQL es pot fer un resum de les comandes associades a l'OO:

- creació de tipus de dades (de tipus d'objectes): CREATE TYPE ... AS … (no és comanda O-R, però es fa servir junt amb les dues anteriors);
- creació de taules mares (de contenidors/taules d'objectes): CREATE TABLE ... OF ... en aquestes s'especifiquen las claus primàries; i
- creació de les taules filles (de contenidors/taules d'objectes basades en altres taules d'objectes): CREATE TABLE ...( ) INHERITS … són les que hereten de les mares.
  Cal tenir en compte les claus primaries i les foràinies en aquestes definicions d'herencia, doncs s'hereten els atributs, els camps, però no les restriccions de claus. És per això que han de ser repetides les restriccions de claus primàries i foràinies quan es defineixen les taules filles. Per la cas de les foràinies recordar que l'estructura seria: CONSTRAINT ... FOREIGN KEY (…) REFERENCES … MATCH SIMPLE ON UPDATE ... ON DELETE ...

Un diagrama a mode d'exemple (petit diagrama basat en UML):



Per aquest esquema, a PostgreSQL:
* es crearia un tipus d'objecte VEHICLE amb els tres atributs indicats. CREATE TYPE ... AS ..., es l'eina del llenguatge PostgreSQL que es fa servir per defin aquest tipus d'objectes. En realitat, és crear un tipus de dada, però és el recurs que tenim a una base de dades objecte-relacional per poder fer-ho.
* es crearia una taula d'objectes. CREATE TABLE ... OF ... (no CREATE TABLE sinò CREATE TABLE ... OF ....) és una eina creada expressament a PostgreSQL que permet crear una taula en base a un tipus, amb l'estructura d'un tipus de dada. Aquesta és la manera de crear un contenidor d'objectes, un recurs per emmagatzemar objectes (realitats), ocurrències del món real.
- El registres de la taula d'objectes creada serán els objectes de món real, les ocurrències del món real del tipus d'objecte definit.
* es crearien, finalment, dues taules més amb la sintax CREATE TABLE ... INHERITS basades en la taula d'objectes ja creada (la taula mare, es pot dit) tot afegint a cada taula filla els trets (els camps) que la distingeixen de la mare, creante-se una taula amb els tres camps heretats de la mare més els propis cada filla, cap pels TURISME i els dos de la FURGO. L'herència permet aprofitar la definició d'un tipus d'objecte per altres que en tenen atributs en comú.
- De nou, els registres d'aquestes taules filles serian objectes (realitats), ocurrències del món real, però amb les particularitats definides.

Junt amb la imatge ja s'explica el procés per tal de fer l'aplicació del model OO amb les eines de la base de dades O-R PostgreSQL.

Afegir, per la seua importància, i a propòsit de CREATE TABLE ... *OF type_name* remarcar les especificacions següents, que indiquen què fa aquesta sentència (tret del manual de PostgreSQL):

> *Creates a typed table, which takes its structure from the specified composite type (name optionally schema-qualified). A typed table is tied to its type; for*

example the table will be dropped if the type is dropped (with DROP TYPE ... CASCADE).
When a typed table is created, then the data types of the columns are determined by the underlying composite type and are not specified by the CREATE TABLE command. But the CREATE TABLE command can add defaults and constraints to the table and can specify storage parameters.
...

En resum, que una taula creada en base a un tipus depen de l'estructura del tipus, és a dir, els camps no els crea realment la comanda *CREATE TABLE* sino la part de on l' diu en base a quin tipus es crea (***OF** <tipus de dada>*). Per això:
1. No es poden definir camps afegits perquè els ha de tenir el tipus
2. La taula depen del tipus, i si s'esborra el tipus s'esborra la taula. I aquí está el lligam amb l'OO. Es veu, oi?. Que el tipus d'objecte, va abans de la taula d'objectes i en té dependència.

Seguint amb l'exemple de la imatge anterior, una possible implementació a PostgreSQL podria ser així:

```
-- creació de l'estructura de la base de dades
CREATE TYPE vehicle AS (
        matricula varchar(8),
        marca varchar(25),
        model varchar(30));

CREATE TABLE vehicles OF vehicle (
        PRIMARY KEY (matricula));

CREATE TABLE turismes (
     PRIMARY KEY (matricula)
)INHERITS(vehicles);

CREATE TABLE furgos (
        tara float,
        carrega float,
        PRIMARY KEY (matricula)
)INHERITS(vehicles);
-- a les taules filles es repeteixen les restriccions de clau primària

-- dades de prova i consultes
insert into vehicles values ('CSK-1234','Peugeot','205');
insert into vehicles values ('BKM-4321','Seat','León');
select * from vehicles;
insert into turismes values ('LPT-9876','Renault','Clio');
select * from vehicles;
select * from turismes;
insert into furgos values ('ZHG-9876','Peugeot','Vito', 1500, 1300);
select * from vehicles;
select * from furgos;
```

A un SGBD objecte-relacional, com ho pot ser PostgreSQL, es poden barrejar les defincions de taules d'objectes amb taules “normals”, es poden fer servir les comandes fer definició de taules sense restriccions,PostgreSQL no avisa si s'està seguint el model relacional i/o el model O-R. No obstant, la idea de tot això és determinar si la nostra base de dades serà Objecte-Relacional o relacional i, segons el que sigui, seguir l'esquema de construcció de taules d'objectes o de taules “normals”.

Quan es fa servir l'esquema O-R, de la manera que s'ha estat explicant, els recursos de la base de dades (tipus i taules) queden lligats amb dependències, com és lògic. Una manera de veure aquest lligam, entre taules mares i filles, per exemple, és mirar l'atribut *tableoid*, que permet veure com PostgreSQL lliga cada registre de cada taula mare amb cada registre associat de la taula filla, justament amb una dada clau, que és la manera que es fa servir als models relacionals i que és en la que, en realitat, està organitzat/estructurat aquest SGBD originàriament relacional.

Seguint amb l'exemple de la imatge anterior, amb les següents consultes es podria veure això:

```
-- consultes del tableoid de les taules
select tableoid,* from vehicles order by tableoid;
select tableoid,* from turismes;
select tableoid,* from furgos;
```

Per la resta d'accions, i pel motiu explicat anterioment, la base de dades es comporta com una base de dades relacional, tot tenint en compte els lligams entre mares i filles.

Seguint amb l'exemple de la imatge anterior, amb les següents consultes es podria veure això:

```
-- esborrats
delete from vehicles;
select * from vehicles
-- ...tot esborrat: sense mare no hi ha filles

delete from turismes where matricula='LPT-9876';
select * from turismes;
select * from vehicles;

-- actualitzacions
...
```