

# M2 UF4 Pt2.1 PostgreSQL Objectes

## M2 UF4 Pt2.1 PostgreSQL Objectes

0. Preparació entorn
1. Crear un camp compost
2. Crear taula amb camp compost
3. Inserts a una taula amb camp compost
4. Restriccions als camps
5. Restriccions a les taules

## 0. Preparació entorn

- Crea la base de dades DESCOMPOS\_PCS a partir de l'script arxiu.psql seguint les instruccions indicades al peu del mateix.

El primer pas serà la creació de la BD amb el nom desitjat:

```
CREATE DATABASE DESCOMPOS_PCS;
```

```
postgres=# \l
```

Name	Owner	Encoding	Collate	Ctype	Access privileges
albums	albums	UTF8	es_ES.UTF-8	es_ES.UTF-8	
postgres	postgres	UTF8	es_ES.UTF-8	es_ES.UTF-8	
template0	postgres	UTF8	es_ES.UTF-8	es_ES.UTF-8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	es_ES.UTF-8	es_ES.UTF-8	=c/postgres + postgres=CTc/postgres

```
(4 rows)
```

```
postgres=# CREATE DATABASE DESCOMPOS_PCS;
```

```
CREATE DATABASE
```

```
postgres=# \l
```

Name	Owner	Encoding	Collate	Ctype	Access privileges
albums	albums	UTF8	es_ES.UTF-8	es_ES.UTF-8	
descompos_pcs	postgres	UTF8	es_ES.UTF-8	es_ES.UTF-8	
postgres	postgres	UTF8	es_ES.UTF-8	es_ES.UTF-8	
template0	postgres	UTF8	es_ES.UTF-8	es_ES.UTF-8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	es_ES.UTF-8	es_ES.UTF-8	=c/postgres + postgres=CTc/postgres

```
(5 rows)
```

```
postgres=#
```

Seguidament imпорarem l'arxiu sql per emplenar la BD amb totes les comandes desitjades:

```
psql -d descompos_pcs -a -f arxiu.psql
```

```
carlos@berserker:/home$ su postgres
Contraseña:
postgres@berserker:/home$ ls
arxiu.psql carlos
postgres@berserker:/home$ psql -d DESCOMPON_PCS -a -f arxiu.psql
psql: FATAL: database "DESCOMPON_PCS" does not exist
postgres@berserker:/home$ psql -d descompon_pcs -a -f arxiu.psql
drop table productes cascade;
psql:arxiu.psql:1: ERROR: table "productes" does not exist
drop table proveidors cascade;
psql:arxiu.psql:2: ERROR: table "proveidors" does not exist
drop table preus cascade;
psql:arxiu.psql:3: ERROR: table "preus" does not exist
drop table guanys cascade;
psql:arxiu.psql:4: ERROR: table "guanys" does not exist
CREATE TABLE productes (
part varchar(20),
tipus varchar(20),
especificacio varchar(20),
psuggerit float(6),
clau serial,
PRIMARY KEY(clau)
);
CREATE TABLE
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('Processador','2 GHz','32 bits',NULL);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('Processador','2.4 GHz','32 bits',35);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('Processador','1.7 GHz','64 bits',205);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('Processador','3 GHz','64 bits',560);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('RAM','128MB','333 MHz',10);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('RAM','256MB','400 MHz',35);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('Disc Dur','80 GB','7200 rpm',60);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('Disc Dur','120 GB','7200 rpm',78);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('Disc Dur','200 GB','7200 rpm',110);
INSERT 0 1
INSERT INTO productes (part,tipus,especificacio,psuggerit) VALUES ('Disc Dur','40 GB','4200 rpm',NULL);
```

```
INSERT INTO preus VALUES ('Nacional',001,30.82);
INSERT 0 1
INSERT INTO preus VALUES ('Nacional',002,32.73);
INSERT 0 1
INSERT INTO preus VALUES ('Nacional',003,202.25);
INSERT 0 1
INSERT INTO preus VALUES ('Nacional',005,9.76);
INSERT 0 1
INSERT INTO preus VALUES ('Nacional',006,31.52);
INSERT 0 1
INSERT INTO preus VALUES ('Nacional',007,58.41);
INSERT 0 1
INSERT INTO preus VALUES ('Nacional',010,64.38);
INSERT 0 1
INSERT INTO preus VALUES ('Patito',001,30.40);
INSERT 0 1
INSERT INTO preus VALUES ('Patito',002,33.63);
INSERT 0 1
INSERT INTO preus VALUES ('Patito',003,195.59);
INSERT 0 1
INSERT INTO preus VALUES ('Patito',005,9.78);
INSERT 0 1
INSERT INTO preus VALUES ('Patito',006,32.44);
INSERT 0 1
INSERT INTO preus VALUES ('Patito',007,59.99);
INSERT 0 1
INSERT INTO preus VALUES ('Patito',010,62.02);
INSERT 0 1
INSERT INTO preus VALUES ('Tecno-k',003,198.34);
INSERT 0 1
INSERT INTO preus VALUES ('Tecno-k',005,9.27);
INSERT 0 1
INSERT INTO preus VALUES ('Tecno-k',006,34.85);
INSERT 0 1
INSERT INTO preus VALUES ('Tecno-k',007,59.95);
INSERT 0 1
INSERT INTO preus VALUES ('Tecno-k',010,61.22);
INSERT 0 1
INSERT INTO preus VALUES ('Tecno-k',012,62.29);
INSERT 0 1
postgres@berserker:/home$
```

```

postgres@berserker:/home$ psql
psql (10.10 (Ubuntu 10.10-0ubuntu0.18.04.1))
Type "help" for help.

postgres=# \c descompon_pcs
You are now connected to database "descompon_pcs" as user "postgres".
descompon_pcs=# \dt
               List of relations
 Schema |      Name      | Type  | Owner
-----+-----+-----+-----
 public | guanys         | table | postgres
 public | preus          | table | postgres
 public | productes      | table | postgres
 public | proveidors     | table | postgres
(4 rows)

descompon_pcs=#

```

## 1. Crear un camp compost

- Crea un tipus de dades compost que inclogui més d'un camp (`CREATE TYPE ... AS ...`) que pugui servir per una columna d'una nova taula que hauràs de crear a la BD (no es pot usar el tipus de l'exemple dels apunts).

Per crear un camp compost ho farem de la següent manera, la taula tindrà la següent estructura:

Empresa:

- id
- Nom
- Adreça
- Localització
- Representant [ ]

```

CREATE TYPE portaveu AS
(
    nom VARCHAR(50),
    cognom VARCHAR(50),
    dni VARCHAR(10)
);

```

```

descompon_pcs=# CREATE TYPE portaveu AS
descompon_pcs=# (
descompon_pcs(#   nom VARCHAR(50),
descompon_pcs(#   cognom VARCHAR(50),
descompon_pcs(#   dni VARCHAR(10)
descompon_pcs(# );
CREATE TYPE
descompon_pcs=#

```

## 2. Crear taula amb camp compost

- Crea una nova taula que inclogue algun camp del tipus de dades compost anterior.

Avans hem esmentat com seria la taula, ara només cal crear-la de forma que inclogui el camp compost:

```
CREATE TABLE empreses
(
  id SERIAL NOT NULL PRIMARY KEY,
  nom varchar(50),
  adreca varchar(100),
  localitzacio varchar(100),
  representant portaveu
);
```

```
descompon_pcs=# CREATE TABLE empreses
descompon_pcs=# (
descompon_pcs(# id SERIAL NOT NULL PRIMARY KEY,
descompon_pcs(# nom varchar(50),
descompon_pcs(# adreca varchar(100),
descompon_pcs(# localitzacio varchar(100),
descompon_pcs(# representant portaveu
descompon_pcs(# );
CREATE TABLE
descompon_pcs=#
```

## 3. Inserts a una taula amb camp compost

- Mostra exemples d'inserció de dades a la taula (INSERT INTO...), actualització i consulta de les mateixes (SELECT...), accedint en tots els casos al camp del tipus de dades compost.

qawrtgasrgargerg

```
INSERT INTO empreses (nom, adreca, localitzacio, representant)
VALUES
('PCComponentes', 'c/Algun lloc de almeria', 'Espanya',
ROW
('Carlos', 'Jimenez Fabregat', '47478945-E')
);
```

```
descompon_pcs=# INSERT INTO empreses (nom, adreca, localitzacio, representant)
descompon_pcs=# VALUES
descompon_pcs=# ('PCComponentes', 'c/Algun lloc de almeria', 'Espanya',
descompon_pcs(# ROW
descompon_pcs(# ('Carlos', 'Jimenez Fabregat', '47478945-E')
descompon_pcs(# );
INSERT 0 1
descompon_pcs=#
```

## 4. Restriccions als camps

- Afegiu a alguna taula ja creada, o creeu-ne alguna de nova, que tingue diferents tipus de restriccions (`PRIMARY KEY`, `UNIQUE`, `FOREIGN KEY/REFERENCES`, `NULL`, `CHECK`, ...) als atributs. Demostreu la utilitat de les restriccions insertant dades a les taules.

Jo he triat demostrar el tipus `UNIQUE` que és molt útil per a tipus de camps que volem que actuen com una clau primària a nivell de que no es poden repetir i han de ser únics per exemple un DNI

```
CREATE TABLE treballadors
(
  id SERIAL NOT NULL PRIMARY KEY,
  dni varchar(10),
  nom varchar(50),
  cognom varchar(100)
);
```

```
ALTER TABLE treballadors ADD CONSTRAINT dni_treballador UNIQUE (dni);
```

```
descompon_pcs=# CREATE TABLE treballadors
descompon_pcs=# (
descompon_pcs=# id SERIAL NOT NULL PRIMARY KEY,
descompon_pcs=# dni varchar(10),
descompon_pcs=# nom varchar(50),
descompon_pcs=# cognom varchar(100)
descompon_pcs=# );
descompon_pcs=# CREATE TABLE
descompon_pcs=# ALTER TABLE treballadors ADD CONSTRAINT dni_treballador UNIQUE (dni);
descompon_pcs=# ALTER TABLE
descompon_pcs=#
```

Ara només caldrà realitzar les pertinents comprovacions

```
INSERT INTO treballadors (dni, nom, cognom)
VALUES
('47478945-E', 'Carlos', 'Jimenez Fabregat');
```

- Com podem veure al intentar inserir dos treballadors amb el mateix DNI ens diu que no poden haver-hi duplicats

```
(carlos) 192.168.204.193 — Konsole
descompon_pcs=# INSERT INTO treballadors (dni, nom, cognom)
descompon_pcs=# VALUES
descompon_pcs=# ('47478945-E', 'Carlos', 'Jimenez Fabregat');
INSERT 0 1
descompon_pcs=# INSERT INTO treballadors (dni, nom, cognom)
descompon_pcs=# VALUES
descompon_pcs=# ('47478945-E', 'Carlos', 'Jimenez Fabregat');
ERROR: duplicate key value violates unique constraint "dni_treballador"
DETAIL: Key (dni)=(47478945-E) already exists.
descompon_pcs=# select * from treballadors
;
 id |      dni      |  nom  |  cognom
----+-----+-----+-----
  1 | 47478945-E | Carlos | Jimenez Fabregat
(1 row)
descompon_pcs=#
```



## 5. Restriccions a les taules

- Feu el mateix que al punt 4 però en restriccions de taula.

Ara simplement caldrà crear la taula nova directament amb els camps dels que volm fer el unique

```
CREATE TABLE treballadors2
(
  id SERIAL NOT NULL,
  dni varchar(10) NOT NULL,
  nom varchar(50),
  cognom varchar(100),
  CONSTRAINT pk_treballadors2 PRIMARY KEY(id),
  CONSTRAINT id_dni_treballadors2 UNIQUE(id, dni)
);
```

```
descompon_pcs=# CREATE TABLE treballadors2
descompon_pcs=# (
descompon_pcs(# id SERIAL NOT NULL,
descompon_pcs(# dni varchar(10) NOT NULL,
descompon_pcs(# nom varchar(50),
descompon_pcs(# cognom varchar(100),
descompon_pcs(# CONSTRAINT pk_treballadors2 PRIMARY KEY(id),
descompon_pcs(# CONSTRAINT id_dni_treballadors2 UNIQUE(id, dni)
descompon_pcs(# );
CREATE TABLE
descompon_pcs=# █
```