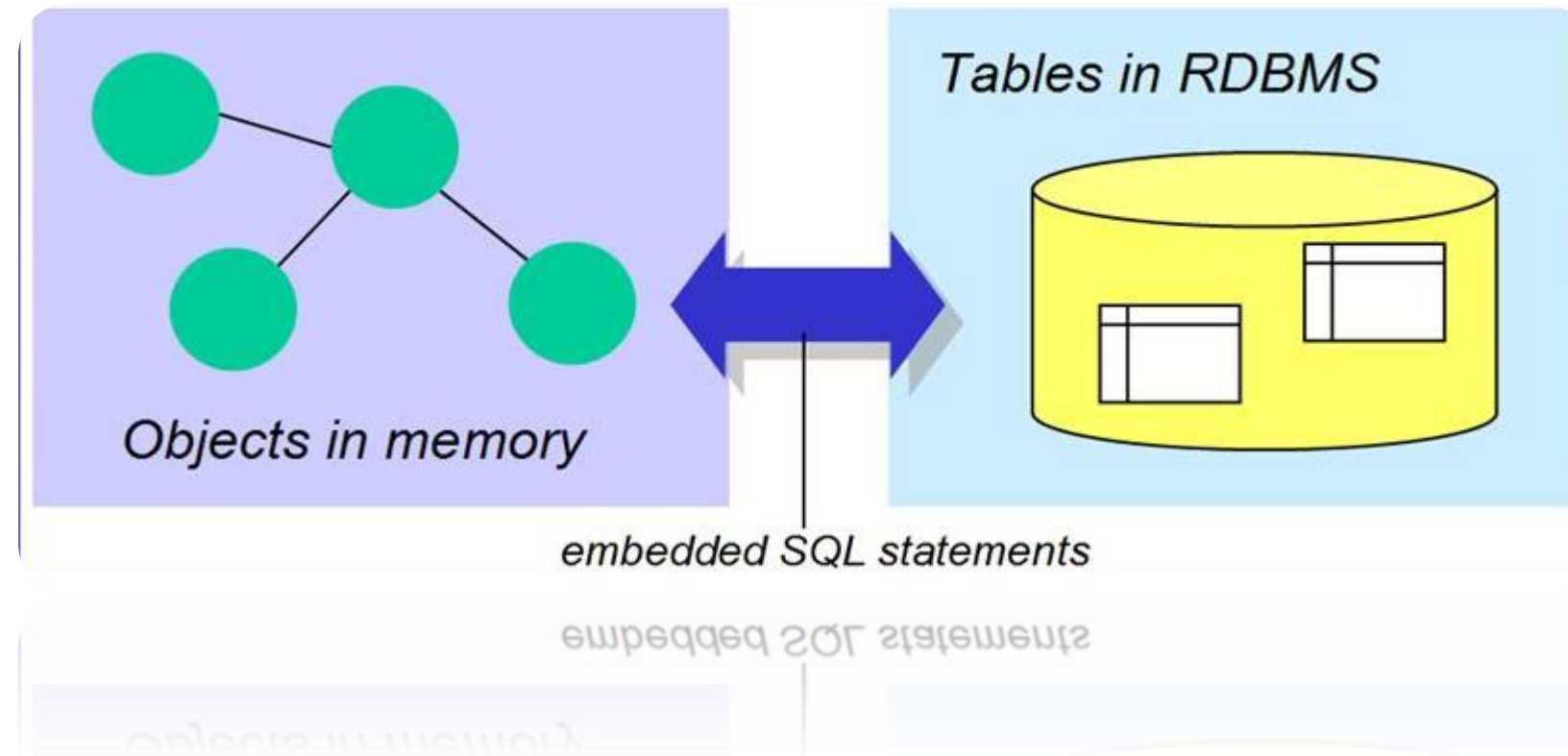


# BDOR

## Exemples



**Institut Rafael  
Campalans**

Plaça del Remei, 1  
17160 Anglès

**Cicle:** DAM

**Curs:** 2022/2023

**Mòdul:** 02 Bases de Dades

# Objectius

---



- Treball amb BDOR
  - Tipus compostos
  - Taula d'objectes
  - Herència
  - References

# Exemple de tipus compostos 1

```
postgres=# CREATE TYPE persona AS (  
postgres(# dni text,  
postgres(# nom text,  
postgres(# cognoms text  
postgres(# );  
CREATE TYPE
```

```
postgres=# CREATE TABLE parella (  
postgres(# p_a persona,  
postgres(# p_b persona  
postgres(# );  
CREATE TABLE
```

```
postgres=# SELECT * FROM parella; p_a | p_b  
-----+-----  
(0 filas)
```

```
postgres=# INSERT INTO parella VALUES  
postgres=# (('42123232x','marc','garcia'),('34523846w','ernest','sanchez')),  
postgres=# (('37123211m','albert','garcia'),('27523117n','joan','molina')),  
postgres=# (('17197411r','roger','mendez'),('23423234j','eva','sanchez'));  
INSERT 0 3
```

# Exemple de tipus compostos 1

```
postgres=# SELECT * FROM parella;
```

p_a	p_b
(42123232x,marc,garcia)	(34523846w,ernest,sanchez)
(37123211m,albert,garcia)	(27523117n,joan,molina)
(17197411r,roger,mendez)	(23423234j,eva,sanchez)

(3 filas)

```
postgres=# SELECT (p_a).dni FROM parella;
```

dni

42123232x
37123211m
17197411r

(3 filas)

```
postgres=# SELECT (p_b).cognoms FROM parella;
```

cognoms

sanchez
molina
sanchez

(3 filas)

# Exemple de tipus compostos 1

```
postgres=# SELECT * FROM parella;
```

p_a	p_b
(42123232x,marc,garcia)	(34523846w,ernest,sanchez)
(37123211m,albert,garcia)	(27523117n,joan,molina)
(17197411r,roger,mendez)	(23423234j,eva,sanchez)

(3 filas)

```
postgres=# SELECT (p_a).nom, (p_b).nom FROM parella WHERE (p_a).nom like '%er%' OR (p_b).nom like '%ev%';
```

nom	nom
albert	joan
roger	eva

(2 filas)

```
postgres=# SELECT (p_a).nom, (p_b).nom FROM parella WHERE (p_a).nom like '%er%' AND (p_b).nom like '%ev%';
```

nom	nom
roger	eva

(1 fila)

---

# EXEMPLE TIPUS COMPOSTOS 2

# Exemple de tipus compostos 2

```
postgres=# CREATE TYPE t_persona AS (  
postgres(# dni varchar(9),  
postgres(# nom text,  
postgres(# cognoms text  
postgres(# );
```

**CREATE TYPE**

```
postgres=# CREATE TYPE t_telefon AS (  
postgres(# prefix_internacional varchar(4),  
postgres(# numero varchar(9)  
postgres(# );
```

**CREATE TABLE**

```
postgres=# CREATE TABLE persona (  
postgres(# id serial PRIMARY KEY,  
postgres(# dades t_persona,  
postgres(# nums_tel integer,  
postgres(# telefons t_telefon[3]  
postgres(# );
```

**CREATE TABLE**

# Exemple de tipus compostos 2

```
postgres=# CREATE OR REPLACE PROCEDURE afegir_persona(d varchar(9), n text, c text)
postgres-# LANGUAGE SQL AS $$
postgres$# INSERT INTO persona(dades.dni,dades.nom,dades.cognoms,nums_tel) VALUES (d,n,c,0);
postgres$# $$;
CREATE PROCEDURE
```

```
postgres=# SELECT * FROM persona;
```

id	dades	nums_tel	telefonos
-----+-----+-----+-----			
(0 filas)			

```
postgres=# CALL afegir_persona('23128437i','pere','marti vidal');
CALL
```

```
postgres=# SELECT * FROM persona;
```

id	dades	nums_tel	telefonos
-----+-----+-----+-----			
1	(23128437i,pere,"marti vidal")		0

(1 fila)



# Exemple de tipus compostos 2

```
postgres=# CREATE OR REPLACE PROCEDURE afegir_num_tel(d varchar(9), p varchar(4), t varchar(9))
postgres=# LANGUAGE plpgsql AS $$
postgres$# DECLARE
postgres$#     n integer;
postgres$# BEGIN
postgres$#     IF EXISTS (SELECT * FROM persona WHERE (dades).dni LIKE d) THEN
postgres$#         SELECT nums_tel INTO n FROM persona WHERE (dades).dni = d;
postgres$#         IF (n < 3) THEN
postgres$#             UPDATE persona SET telefons[n].prefix_internacional = p WHERE (dades).dni = d;
postgres$#             UPDATE persona SET telefons[n].numero = t WHERE (dades).dni = d;
postgres$#             UPDATE persona SET nums_tel = n+1;
postgres$#         END IF;
postgres$#     END IF;
postgres$# END;
postgres$# $$;
CREATE PROCEDURE
postgres=#
```

# Exemple de tipus compostos 2

```
postgres=# SELECT * FROM persona WHERE (dades).dni = '23128437i';
```

id	dades	nums_tel	telefons
+	+	+	
4	(23128437i,pere,"marti vidal")	0	

(1 fila)

```
postgres=# CALL afegir_num_tel('23128437i','34','650573881');
```

```
postgres=# SELECT * FROM persona WHERE (dades).dni = '23128437i';
```

id	dades	nums_tel	telefons
+	+	+	
4	(23128437i,pere,"marti vidal")	1	[0:0]={ "(34,650573881)" }

(1 fila)

```
postgres=# CALL afegir_num_tel('23128437i','35','643256211');
```

```
postgres=# SELECT * FROM persona WHERE (dades).dni = '23128437i';
```

id	dades	nums_tel	telefons
+	+	+	
4	(23128437i,pere,"marti vidal")	2	[0:1]={ "(34,650573881)", "(35,643256211)" }

(1 fila)

```
postgres=# CALL afegir_num_tel('23128437i','36','723451133');
```

```
postgres=# SELECT * FROM persona WHERE (dades).dni = '23128437i';
```

id	dades	nums_tel	telefons
+	+	+	
4	(23128437i,pere,"marti vidal")	3	[0:2]={ "(34,650573881)", "(35,643256211)", "(36,723451133)" }

(1 fila)

```
postgres=#
```

---

# EXEMPLE TIPUS COMPOSTOS 3

# Exemples de tipus compostos 3

```
postgres=# CREATE TYPE coordenada AS (  
postgres(# latitud float,  
postgres(# longitud float  
postgres(# );
```

**CREATE TYPE**

```
postgres=# CREATE TYPE adreca AS (  
postgres(# carrer text,  
postgres(# numero text,  
postgres(# poblacio text,  
postgres(# codi_postal varchar(5),  
postgres(# posicio coordenada  
postgres(# );
```

**CREATE TYPE**

```
postgres=# CREATE TABLE usuari (  
postgres(# id serial not null primary key,  
postgres(# nom text,  
postgres(# cognoms text,  
postgres(# data_naixement date,  
postgres(# telefon varchar(11),  
postgres(# email text,  
postgres(# domicili adreca  
postgres(# );
```

**CREATE TABLE**

# Exemples de tipus compostos 3

```
postgres=# INSERT INTO usuari(nom,cognoms,data_naixement,telefon,email,domicili) VALUES
postgres-# (
postgres(# 'laura',
postgres(# 'garcia perez',
postgres(# '1979-09-15',
postgres(# '650435647',
postgres(# 'lgarcia@dfre.cat',
postgres(# ROW('sant pere','10','lloret de mar','17310',ROW(41.699904021726496, 2.846679613059704))
postgres(# );
```

INSERT 0 1

```
postgres=# INSERT INTO usuari(nom,cognoms,data_naixement,telefon,email,domicili) VALUES
postgres-# (
postgres(# 'albert',
postgres(# 'pibernat menendez',
postgres(# '1975-04-23',
postgres(# '654321298',
postgres(# 'rpibernat@fest.org',
postgres(# ROW('ample','20','blanes','17300',ROW(41.67476909919283, 2.7918434747287906))
postgres(# );
```

INSERT 0 1

# Exemples de tipus compostos 3

```
postgres=# SELECT * FROM usuari;
```

id	nom	cognoms	data_naixement	telefon	email	domicili
1	laura	garcia perez	1979-09-15	650435647	<a href="mailto:lgarcia@dfre.cat">lgarcia@dfre.cat</a>	("sant pere",10,"lloret de mar",17310,"(41.6999040217265,2.8466796130597)")
2	albert	pibernat menendez	1975-04-23	654321298	<a href="mailto:rpibernat@fest.org">rpibernat@fest.org</a>	(ample,20,blanes,17300,"(41.6747690991928,2.79184347472879)")

(2 filas)

```
postgres=# SELECT nom,cognoms,(domicili).poblacio FROM usuari;
```

nom	cognoms	poblacio
laura	garcia perez	lloret de mar
albert	pibernat menendez	blanes

(2 filas)

```
postgres=# SELECT nom,cognoms,(domicili).poblacio,(domicili).posicio.latitud,(domicili).posicio.longitud FROM usuari;
```

nom	cognoms	poblacio	latitud	longitud
laura	garcia perez	lloret de mar	41.6999040217265	2.8466796130597
albert	pibernat menendez	blanes	41.6747690991928	2.79184347472879

(2 filas)

# Exemples de tipus compostos 3

```
postgres=# SELECT
postgres=# id, nom, cognoms, (domicili).carrer, (domicili).numero, (domicili).posicio.latitud, (domicili).posicio.longitud
postgres=# FROM usuari;
```

id	nom	cognoms	carrer	numero	latitud	longitud
1	laura	garcia perez	sant pere	10	41.6999040217265	2.8466796130597
2	albert	pibernat menendez	ample	20	41.6747690991928	2.79184347472879

2 filas)

```
postgres=# UPDATE usuari SET
postgres=# domicili.carrer = 'llarg',
postgres=# domicili.numero = '15',
postgres=# domicili.posicio = ROW(41.67454217779667, 2.7896852133632413)
postgres=# WHERE id = 2;
UPDATE 1
```

```
postgres=# SELECT
postgres=# id, nom, cognoms, (domicili).carrer, (domicili).numero, (domicili).posicio.latitud, (domicili).posicio.longitud
postgres=# FROM usuari;
```

id	nom	cognoms	carrer	numero	latitud	longitud
1	laura	garcia perez	sant pere	10	41.6999040217265	2.8466796130597
2	albert	pibernat menendez	llarg	15	41.6745421777967	2.78968521336324

(2 filas)

# Exemple de tipus compostos 3

```
postgres=# CREATE PROCEDURE inserir_dades_usuari(n text, c text, d date, t varchar(11), e text)
postgres=# LANGUAGE SQL AS $$
postgres$#      INSERT INTO usuari(nom,cognoms,data_naixement,telefon,email) VALUES (n,c,d,t,e);
postgres$# $$;
CREATE PROCEDURE
```

```
postgres=# CALL inserir_dades_usuari('genis','torroella lopez','1943-02-27','972362123','getorlo@linosa.org');
CALL
```

```
postgres=# SELECT id,nom,cognoms,data_naixement,telefon,email FROM usuari;
 id |  nom  |      cognoms      | data_naixement | telefon |      email
----+-----+-----+-----+-----+-----
  1 | laura | garcia perez      | 1979-09-15     | 650435647 | lgarcia@dfre.cat
  2 | albert | pibernat menendez | 1975-04-23     | 654321298 | rpibernat@fest.org
  3 | genis | torroella lopez   | 1943-02-27     | 972362123 | getorlo@linosa.org
(3 filas)
```

```
postgres=# SELECT id,domicili FROM usuari;
 id |      domicili
----+-----
  1 | ("sant pere",10,"lloret de mar",17310,"(41.6999040217265,2.8466796130597)")
  2 | (ample,20,blanes,17300,"(41.6747690991928,2.79184347472879)")
  3 |
(3 filas)
```



# Exemple de tipus compostos 3

```
postgres=# CREATE PROCEDURE afegir_domicili_usuari(i integer, c text, n text, p text, cp varchar(5), la float, lo float)
postgres=# LANGUAGE SQL AS $$
postgres$# UPDATE usuari SET
postgres$#     domicili.carrer = c,
postgres$#     domicili.numero = n,
postgres$#     domicili.poblacio = p,
postgres$#     domicili.codi_postal = cp,
postgres$#     domicili.posicio = ROW(la,lo)
postgres$# WHERE id = i;
postgres$# $$;
CREATE PROCEDURE
```

```
postgres=# SELECT id,nom,cognoms,domicili FROM usuari WHERE id = 3;
 id | nom  | cognoms      | domicili
----+-----+-----+-----
  3 | genis | torroella lopez |
(1 fila)
```

```
postgres=# CALL afegir_domicili_usuari(3,'enric granados','9','lloret de mar','17310',41.70060029780517,2.8440005952061647);
CALL
postgres=# SELECT id,nom,cognoms,domicili FROM usuari WHERE id = 3;
 id | nom  | cognoms      | domicili
----+-----+-----+-----
  3 | genis | torroella lopez | ("enric granados",9,"lloret de mar",17310,"(41.7006002978052,2.84400059520616)")
(1 fila)
postgres=#
```

---

# EXEMPLE TAULES D'OBJECTES

# Exemple de taules d'objectes

```
postgres=# CREATE TYPE article AS (  
postgres(# id_article integer,  
postgres(# nom text,  
postgres(# preu float,  
postgres(# iva float,  
postgres(# proveidor  
text  
postgres(# );
```

**CREATE TYPE**

AQUEST TIPUS DE CREATE TYPE  
NO POT CONTENIR CAMPS TIPUS  
SERIAL

```
postgres=# CREATE TABLE inventari OF article (  
postgres(# PRIMARY KEY (id_article),  
postgres(# iva WITH OPTIONS DEFAULT 21.0  
postgres(# );
```

**CREATE TABLE**

# Exemple de taules d'objectes

```
postgres=# CREATE PROCEDURE afegir_article_inventari(i integer, n text, e float, o text)
postgres-# LANGUAGE SQL AS $$
postgres$#         INSERT INTO inventari (id_article,nom,preu,proveidor)
postgres$#         VALUES (i,n,e,o);
postgres$# $$;
CREATE PROCEDURE
```

# Exemple de taules d'objectes

```
postgres=# CREATE PROCEDURE modificar_iva_article(i integer, t text)
postgres-# LANGUAGE plpgsql AS $$
postgres$# DECLARE
postgres$#     codi ALIAS FOR $1;
postgres$#     tipus ALIAS FOR $2;
postgres$#     nou float;
postgres$# BEGIN
postgres$#     IF tipus = 'general' THEN
postgres$#         nou = 21.0;
postgres$#     ELSEIF tipus = 'reduit' THEN
postgres$#         nou = 10.0;
postgres$#     ELSEIF tipus = 'superreduit' THEN
postgres$#         nou = 4.0;
postgres$#     ELSE
postgres$#         nou = 21.0;
postgres$#         RAISE NOTICE 'IVA general.';
postgres$#     END IF;
postgres$#     UPDATE inventari SET iva = nou WHERE id_article = codi;
postgres$# END;
postgres$# $$;
CREATE PROCEDURE
```

# Exemple de taules d'objectes

```
postgres=# SELECT * FROM inventari;
 id_article | nom | preu | iva | proveidor
-----+-----+-----+-----+-----
(0 filas)
```

```
postgres=# CALL afegir_article_inventari(1,'caixa',3.5,'caixasa');
CALL
postgres=# CALL afegir_article_inventari(2,'paper',1.5,'papelera sl');
CALL
postgres=# CALL afegir_article_inventari(3,'cordill',0.25,'safa sa');
CALL
postgres=# SELECT * FROM inventari;
 id_article |  nom  | preu | iva | proveidor
-----+-----+-----+-----+-----
          1 | caixa |  3.5 |  21 | caixasa
          2 | paper |  1.5 |  21 | papelera sl
          3 | cordill | 0.25 |  21 | safa sa
(3 filas)
```

# Exemple de taules d'objectes

```
postgres=# CALL modificar_iva_article(2,'superreduit');  
CALL
```

```
postgres=# SELECT * FROM inventari WHERE id_article = 2;  
 id_article |  nom  | preu | iva | proveidor  
-----+-----+-----+-----+-----  
          2 | paper |  1.5 |   4 | papelera sl  
(1 fila)
```

---

# EXEMPLE HERÈNCIA



# Exemple d'herència

```
postgres=# CREATE TABLE persona (  
postgres(# id serial,  
postgres(# nom varchar(30),  
postgres(# cognoms varchar(60),  
postgres(# d_naixement date  
postgres(# );  
CREATE TABLE
```

AQUEST TIPUS D'HERENCIA  
REQUEREIX UN CAMP  
TIPUS SERIAL

```
postgres=# CREATE TABLE estudiant (  
postgres(# nom_cf varchar(50),  
postgres(# curs char,  
postgres(# notes_mp int[15]  
postgres(# ) INHERITS (persona);  
CREATE TABLE
```

```
postgres=# SELECT * FROM estudiant;  
id | nom | cognoms | d_naixement | nom_cf | curs | notes_mp  
----+-----+-----+-----+-----+-----+-----  
(0filas)
```

# Exemple d'herència

```
postgres=# CREATE PROCEDURE afegir_estudiant(i integer, n varchar(30), c varchar(60), d date)
postgres=# LANGUAGE SQL AS $$
postgres$#      INSERT INTO estudiant (id,nom,cognoms,d_naixement) VALUES (i,n,c,d);
postgres$# $$;
CREATE PROCEDURE
```

```
postgres=# CALL afegir_estudiant(1,'monica','garcia lopez','2005-10-21'); CALL
postgres=# CALL afegir_estudiant(2,'laura','lopez perez','2005-04-12');
CALL
```

```
postgres=# CALL afegir_estudiant(3,'manel','perez garcia','2004-12-31'); CALL
postgres=# CALL afegir_estudiant(4,'oscar','garcia menendez','2005-05-17');
CALL
```

```
postgres=# SELECT * FROM estudiant;
```

id	nom	cognoms	d_naixement	nom_cf	curs	notes_mp
1	monica	garcia Lopez	2005-10-21			
2	laura	lopez perez	2005-04-12			
3	manel	perez garcia	2004-12-31			
4	oscar	garcia menendez	2005-05-17			

(4 filas)

# Exemple d'herència

```
postgres=# CREATE PROCEDURE afegir_estudiant_cf(i integer, n varchar(50), c char)
postgres=# LANGUAGE SQL AS $$
postgres$#      UPDATE estudiant SET nom_cf = n WHERE id = i;
postgres$#      UPDATE estudiant SET curs = c WHERE id = i;
postgres$# $$;
```

```
CREATE PROCEDURE
```

```
postgres=# CALL afegir_estudiant_cf(1,'dam','1');
```

```
CALL
```

```
postgres=# CALL afegir_estudiant_cf(2,'daw','1');
```

```
CALL
```

```
postgres=# CALL afegir_estudiant_cf(3,'asix','1');
```

```
CALL
```

```
postgres=# CALL afegir_estudiant_cf(4,'asix','2');
```

```
CALL
```

```
postgres=# SELECT * FROM estudiant;
```

id	nom	cognoms	d_naixement	nom_cf	curs	notes_mp
1	monica	garcia lopez	2005-10-21	dam	1	
2	laura	lopez perez	2005-04-12	daw	1	
3	manel	perez garcia	2004-12-31	asix	1	
4	oscar menendez	garcia	2005-05-17	asix	2	

```
(4 filas)
```

# Exemple d'herència

---

```
postgres=# CREATE PROCEDURE afegir_nota_estudiant_cf(i integer, m int, n int)
postgres=# LANGUAGE SQL AS $$
postgres$#      UPDATE estudiant SET notes_mp[m] = n WHERE id = i;
postgres$# $$;
CREATE PROCEDURE
postgres=# CALL afegir_nota_estudiant_cf(2,1,4);
CALL
postgres=# CALL afegir_nota_estudiant_cf(2,2,9);
CALL
postgres=# CALL afegir_nota_estudiant_cf(3,1,5);
CALL
postgres=# CALL afegir_nota_estudiant_cf(3,2,7);
CALL
postgres=# CALL afegir_nota_estudiant_cf(3,3,2);
CALL
postgres=# CALL afegir_nota_estudiant_cf(4,3,1);
CALL
postgres=# CALL afegir_nota_estudiant_cf(4,1,4);
CALL
```

# Exemple d'herència

---

```
postgres=# SELECT * FROM estudiant;
```

id	nom	cognoms	d_naixement	nom_cf	curs	notes_mp
1	monica	garcia lopez	2005-10-21	dam	1	
2	laura	lopez perez	2005-04-12	daw	1	{4,9}
3	manel	perez garcia	2004-12-31	asix	1	{5,7,2}
4	oscar menendez	garcia	2005-05-17	asix	2	{4,NULL,1}

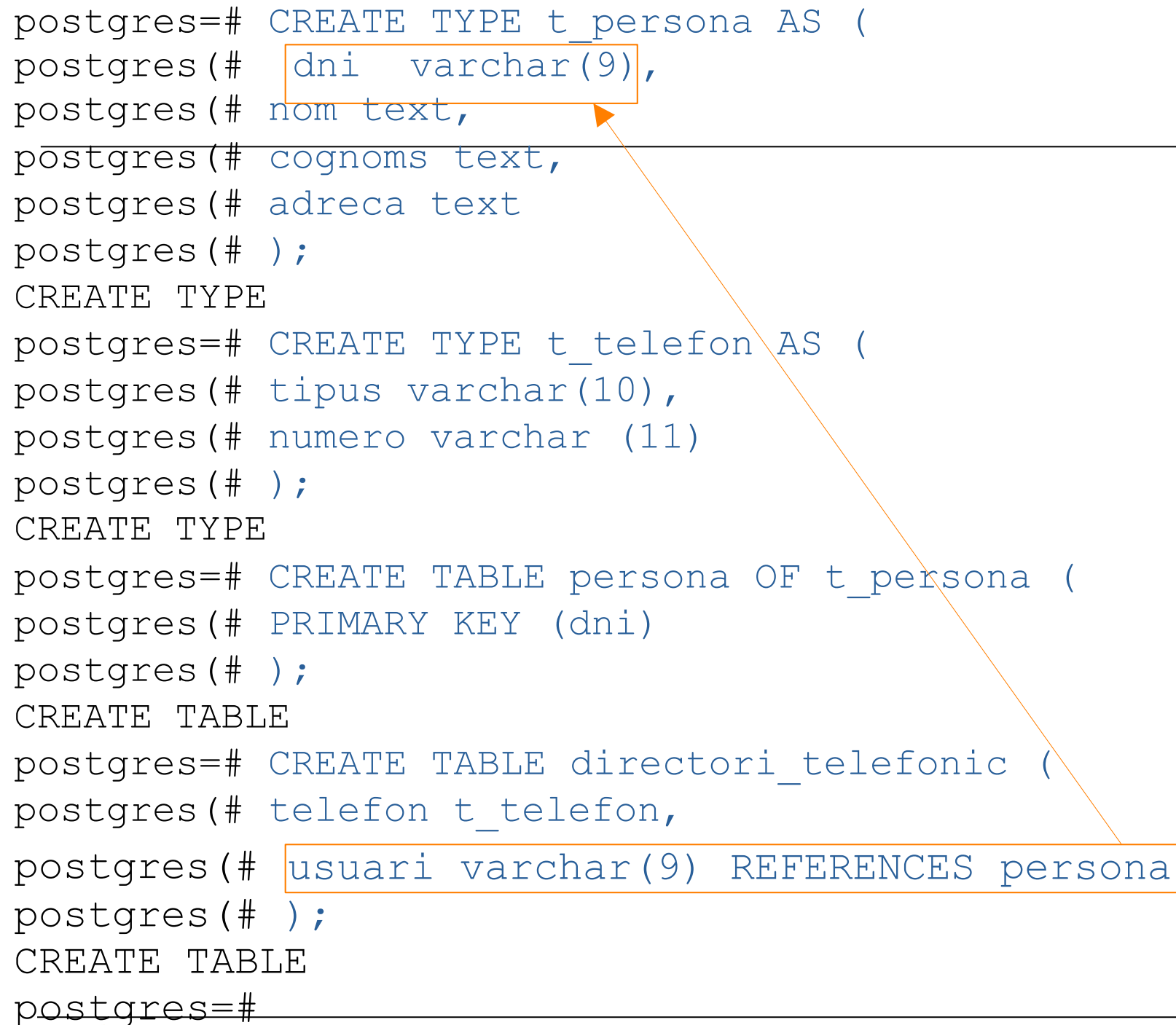
(4 filas)

```
postgres=#
```

---

# EXEMPLE REFERÈNCIES

```
postgres=# CREATE TYPE t_persona AS (  
postgres=#   dni varchar(9),  
postgres=#   nom text,  
postgres=#   cognoms text,  
postgres=#   adreca text  
postgres=# );  
CREATE TYPE  
postgres=# CREATE TYPE t_telefon AS (  
postgres=#   tipus varchar(10),  
postgres=#   numero varchar (11)  
postgres=# );  
CREATE TYPE  
postgres=# CREATE TABLE persona OF t_persona (  
postgres=#   PRIMARY KEY (dni)  
postgres=# );  
CREATE TABLE  
postgres=# CREATE TABLE directori_telefonic (  
postgres=#   telefon t_telefon,  
postgres=#   usuari varchar(9) REFERENCES persona  
postgres=# );  
CREATE TABLE  
postgres=#
```



# Exemple referències

---

```
postgres=# CREATE PROCEDURE afegir_persona(d varchar(9), n text, c text, a text)
postgres-# LANGUAGE SQL AS $$
postgres$#      INSERT INTO persona VALUES (d,n,c,a);
postgres$# $$;
CREATE PROCEDURE
postgres=# CREATE FUNCTION mostrar_persones() RETURNS table(d text, n text, c text)
postgres-# LANGUAGE plpgsql AS $$
postgres$# BEGIN
postgres$#      RETURN QUERY SELECT dni::text, nom, cognoms FROM persona;
postgres$# END;
postgres$# $$;
CREATE FUNCTION
postgres=# CREATE PROCEDURE afegir_telefon(t varchar(10), n varchar(11), d varchar(9))
postgres-# LANGUAGE SQL AS $$
postgres$#      INSERT INTO directori_telefonic VALUES (ROW(t,n),d);
postgres$# $$;
CREATE PROCEDURE
postgres=#
```



# Exemple referències

---

```
postgres=# CALL afegir_persona('41231659d','anna','clapes serra','carrer ample 10, blanes, 17300, girona');
CALL
postgres=# CALL afegir_persona('42546962j','pere','garcia vila','carrer nou 12, lloret de mar, 17310, girona');
CALL
postgres=# CALL afegir_persona('36496211n','laura','perez lopez','carre issac albeniz 1, tossa, 17320, girona');
CALL
postgres=# SELECT mostrar_persones();
      mostrar_persones
(41231659d,anna,"clapes serra")
(42546962j,pere,"garcia vila")
(36496211n,laura,"perez lopez")
(3 filas)
```

# Exemple referències

---

```
-----
postgres=# CALL afegir_telefon('fix','972335794','41231659d');
CALL
postgres=# CALL afegir_telefon('movil','650334219','42546962j');
CALL
postgres=# CALL afegir_telefon('movil','645321123','11111111a');
```

ERROR: inserción o actualización en la tabla «directori\_telefonic» viola la llave foránea «directori\_telefonic\_usuari\_fkey»

DETALLE: La llave (usuari)=(11111111a) no está presente en la tabla «persona».

CONTEXTO: función SQL «afegir\_telefon» en la sentencia 1

```
postgres=# CALL afegir_telefon('fix','972340100','36496211n');
CALL
```

```
postgres=# SELECT * FROM directori_telefonic;
```

telefon	usuari
(fix,972335794)	41231659d
(movil,650334219)	42546962j
(fix,972340100)	36496211n

(3 filas)

```
postgres=#
```

# Exercicis

---



- Crear una base de dades sobre equips de futbol. Ha de disposar de com a mínim d'una taula d'objectes, un tipus compost i una taula que hereti una altre. Totes les insercions, actualitzacions i consultes s'han de realitzar mitjançant procediments o funcions.
- Crear una base de dades sobre grups musicals. Ha de disposar de com a mínim d'una taula d'objectes, un tipus compost i una taula que hereti una altre. Totes les insercions, actualitzacions i consultes s'han de realitzar mitjançant procediments o funcions.
- Crear una base de dades sobre pel·lícules. Ha de disposar de com a mínim d'una taula d'objectes, un tipus compost i una taula que hereti una altre. Totes les insercions, actualitzacions i consultes s'han de realitzar mitjançant procediments o funcions.

# WEBGRAFIA

---

- SQL Tutorial, W3schools, Setembre 2022, <https://www.w3schools.com/sqL/default.asp>
- PostgreSQL Tutorial from scratch, Setembre 2022, <https://www.postgresqltutorial.com/>
- Exercicis Online de SQL, W3schools, Setembre 2022, [https://www.w3schools.com/SQL/sql\\_exercises.asp](https://www.w3schools.com/SQL/sql_exercises.asp)
- PostgreSQL Exercices, Practice,Solution, W3resource,Setembre 2022, <https://www.w3resource.com/postgresql-exercises/>
- PostgreSQL Documentation, PostgreSQL, Setembre 2022, <https://www.postgresql.org/docs/>