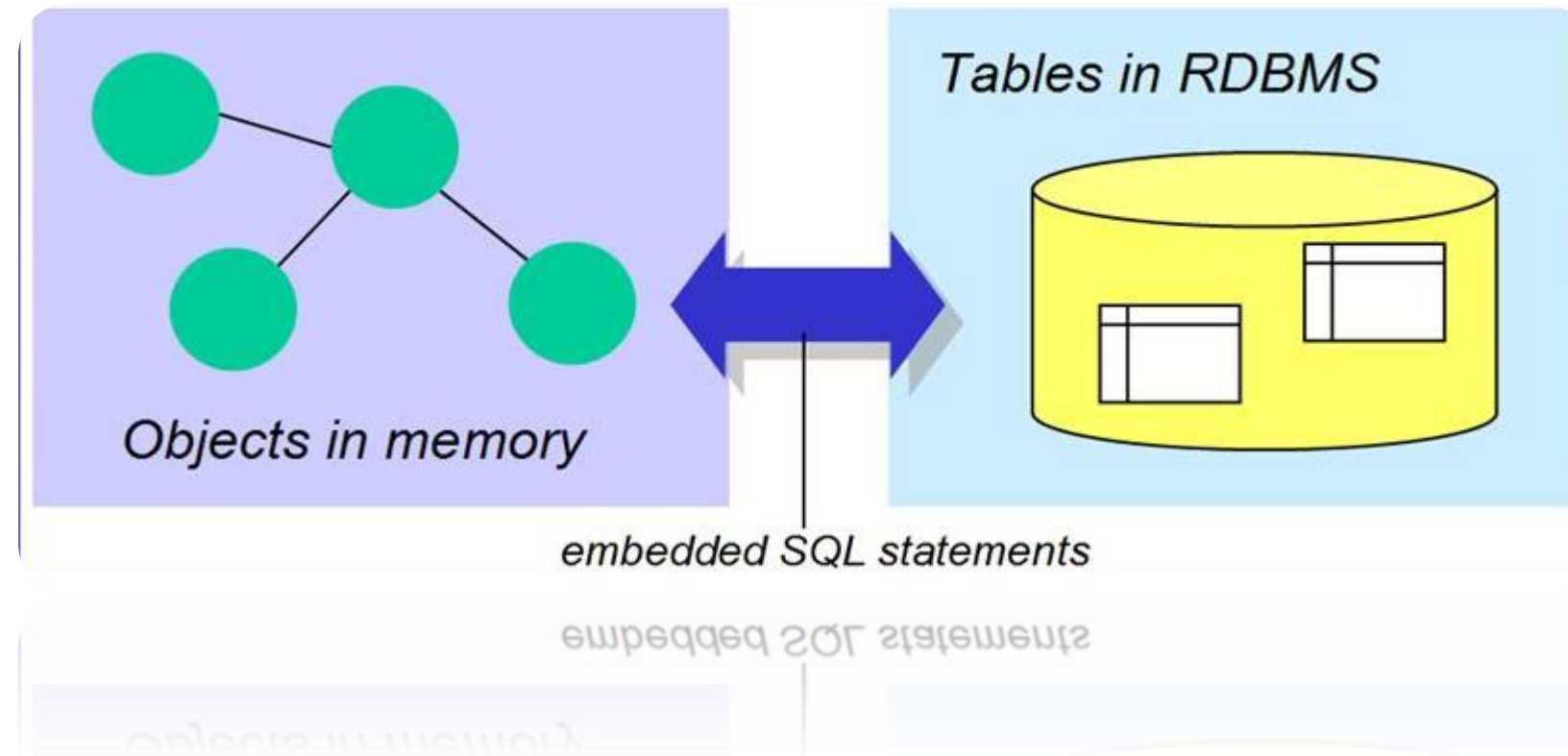


Bases de dades Objecte Relacional (BDOR)



**Institut Rafael
Campalans**

Plaça del Remei, 1
17160 Anglès

Cicle: DAM

Curs: 2022/2023

Mòdul: 02 Bases de Dades

Objectius



- Bases de dades Objecte Relacionals
- Repàs P.O.O.
- Tipus compostos (objectes)
- DDL i DML amb tipus compostos

BD Objecte-Relacionals (ORDBMS)

- Les bases de dades relacionals han esdevingut amb el temps un dels sistemes més utilitzats i efectius per a la gestió de les dades a nivell empresarial.
- Amb l'aparició del paradigma de programació orientada a objectes apareix la necessitat d'emmagatzemar les dades com objectes complets.
- Alguns SGBD Relacionals estenen els llenguatges per treballar amb tipus objecte esdevenint SGBD Objecte-relacional, i per tant combinen l'orientació a objectes i el model relacional.

Característiques (ORDBMS)

- Les bases de dades objecte-relacionals permeten als usuaris crear els seus propis tipus, així com mètodes o funcions associades a aquests tipus, permetent la representació d'objectes.
- Les bases de dades objecte-relacional proporcionen:

L'encapsulació

L'herència

El polimorfisme

Creació de tipus

Sintaxi:

```
CREATE TYPE name AS  
    ( [ attribute_name data_type [ COLLATE collation ]  
      [, ... ] ] )
```

Exemple:

```
create type persona as  
( dni text,  
  nom text,  
  cognoms text);
```

Creació de tipus

Exemples:

```
create table parella    (p_a persona, p_b persona );

insert into parella  values
(('40332042','lluis','lopez'),('40321043','maria','pinyol')),
(('39203034','raul','rius'),('40099024','eloi','camps'));
```

```
select * from parella;
```

p_a

p_b

(40332042,lluis,lopez)

(40321043,maria,pinyol)

(39203034,raul,rius)

(40099024,eloi,camps)

Creació de tipus

Exemples:

```
select (p_a).dni, (p_a).nom, (p_a).cognoms, (p_b).dni  
from parella;
```

Dni	nom	cognoms	dni
40332042	lluis	lopez	40321043
39203034	raul	rius	40099024

Creació de tipus

Exemples:

```
CREATE TYPE inventari_element AS ( nom text,  
proveidor_id integer,  
preu numeric );
```

```
CREATE TABLE a_ma (  
element inventari_element,  
quantitat integer );
```

```
insert into a_ma  
values (('roda',1,120), 10),  
      (('volant',2,50), 5);
```


Creació de tipus

Exemples:

```
select (element).nom, quantitat  
from a_mà
```

nom	quantitat
Roda	10
Volant	5

Podem definir funcions associades a un determinat tipus:

```
CREATE FUNCTION preu_extensio (ie inventari_element, q integer)  
RETURNS numeric AS . . .
```

Tipus (Constructors)

Sintaxi:

```
' ( val1 , val2 , ... ) '  
O be  
ROW(val1, val2, . . . )
```

Exemple:

```
' ("fuzzy dice",42,1.99) '  
O  
ROW('fuzzy dice', 42, 1.99)
```

Tipus (Accés)

Sintaxi:

```
(nom_tipus).columna
```

Exemple:

```
SELECT (element).nom FROM a_ma WHERE (element).preu >9.99;
```

Nom

Roda

Volant

Exemple: Taules d'objectes

```
CREATE TYPE inventari_element AS (  
  nom text,  
  proveidor_id integer,  
  preu numeric );  
  
CREATE TABLE a_ma OF inventari_element (  
  PRIMARY KEY (nom),  
  preu DEFAULT 10);
```

IMPORTANT

En aquests tipus de taules no s'admeten atributs tipus Serial, per tant l'objecte **inventari_element** no pot tenir atributs **Serial**.

Exemple: DML amb Objectes

```
CREATE TYPE inventory_element AS (  
    nom            text,  
    proveidor_id   integer,  
    preu           numeric );
```

```
CREATE TABLE a_ma (  
    element        inventory_element,  
    quantitat      integer);
```

```
UPDATE mytab SET complex_col = ROW(1.1,2.2) WHERE ...;
```

```
INSERT INTO mytab (complex_col.r, complex_col.i) VALUES(1.1, 2.2);
```

```
SELECT * FROM a_ma  
ORDER BY (element).nom, (element).proveidor_id, (element).preu;
```

Exemple: Herència entre taules

```
CREATE TABLE persona (  
  id serial,  
  nom varchar (30),  
  adreça varchar (30)  
);  
CREATE TABLE estudiant  
  ( carrera varchar  
    (50), grup char,  
    grau int ) INHERITS (persona);
```

```
select * from estudiant;
```

nom	adreça	carrera	grup	grau
-----	--------	---------	------	------

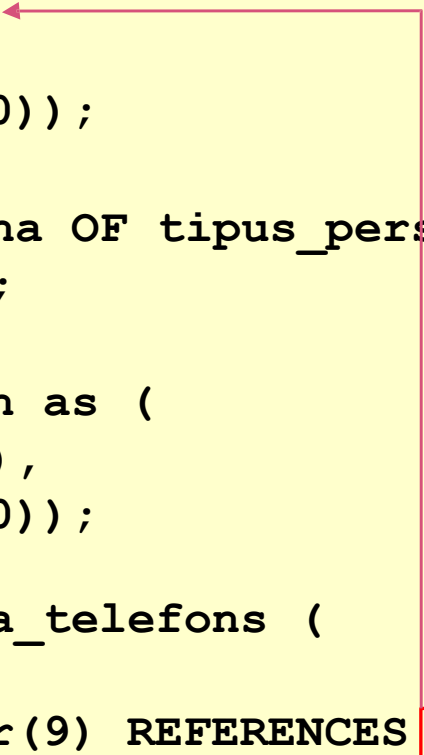
Referències

- Els identificadors únics assignats als objectes que s'emmagatzemen en una taula permeten que aquests es puguin referenciar des dels atributs d'altres objectes.
- Si volem que un atribut emmagatzemi una referencia a un objecte del tipus definit, l'ordre que utilitzem es **REFERENCES**, que implementa una relació d'associació entre els dos tipus objecte.

Referències

Exemple:

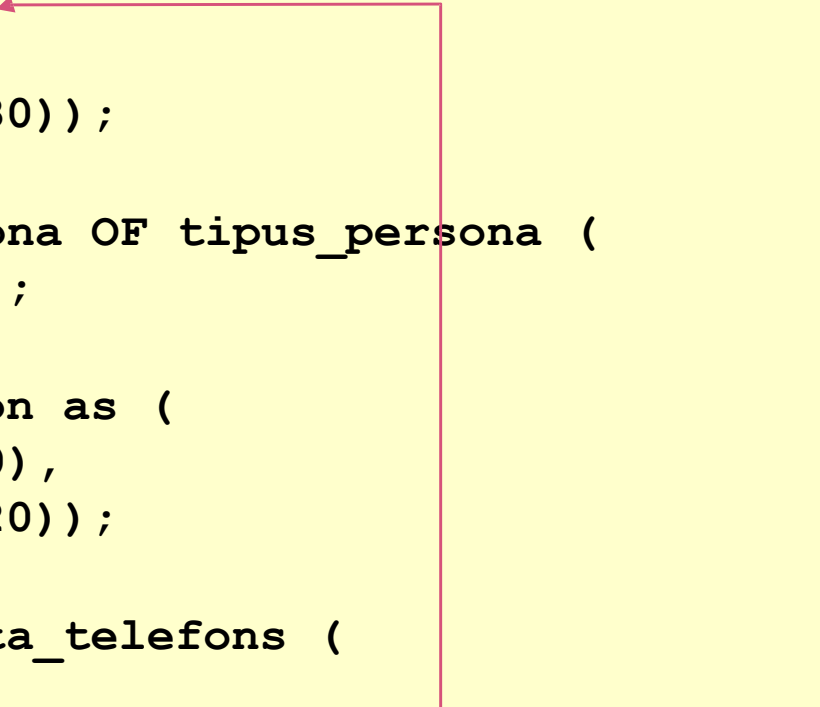
```
CREATE TYPE tipus_persona AS (  
  dni varchar (9),  
  nom varchar (30),  
  adreça varchar (30));  
  
CREATE TABLE persona OF tipus_persona (  
  PRIMARY KEY (dni));  
  
CREATE TYPE telefon as (  
  tipus varchar (10),  
  numero varchar (20));  
  
CREATE TABLE llista_telefons (  
  tlf telefon,  
  propietari varchar(9) REFERENCES persona ) ;
```



Referències

Exemple: podem arribar a indicar quina és la columna a la que fem referència.

```
CREATE TYPE tipus_persona AS (  
  dni varchar (9),  
  nom varchar (30),  
  adreça varchar (30));  
  
CREATE TABLE persona OF tipus_persona (  
  PRIMARY KEY (dni));  
  
CREATE TYPE telefon as (  
  tipus varchar (10),  
  numero varchar (20));  
  
CREATE TABLE llista_telefons (  
  tlf telefon,  
  propietari varchar(9) REFERENCES persona(dni) );
```



Col·leccions (ARRAY)

- El tipus de dades array es una de les característiques especials del PostgreSQL.
- Permet l'emmagatzematge de més d'un valor del mateix tipus en la mateixa columna.

```
CREATE TYPE telefon AS (  
    tipus varchar (10),  
    numero varchar (20));  
  
CREATE TABLE llista_telefons (  
    nom varchar (30),  
    tlf telefon[10] );
```

Col·leccions (ARRAY)

```
INSERT INTO llista_telefons  
VALUES ('Agenda feina', ARRAY[ROW('mobil','623533123')::telefon]);
```

```
SELECT nom, tlf[1] FROM llista_telefons;
```

Nom	tlf

Agenda feina	(mobil,623533123)

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
- PostgreSQL Exercices, Practice,Solution, W3resource,Setembre 2022, <https://www.w3resource.com/postgresql-exercises/>
- PostgreSQL Documentation, PostgreSQL, Setembre 2022, <https://www.postgresql.org/docs/>