CFGS Desenvolupament d'Aplicacions Multiplataforma
Mòdul 2 – Gestió de bases de dades

UF4 Gestió de la informació en bases de dades objecte-relacionals

Unitat 8 – Bases de dades objecte-relacionals
EAC7
(Curs 2018-19 / 2on semestre)

Presentació i resultats d'aprenentatge

Aquest exercici d'avaluació continuada (EAC) es correspon amb els continguts treballats a la unitat **"U8 Bases de dades objecte-relacionals"** de la **"UF4 Gestió de la informació en bases de dades objecte-relacionals"**

1. Gestiona la informació emmagatzemada en bases de dades objecte-relacionals, avaluant i utilitzant les possibilitats que proporciona el sistema gestor.
    - Descriu les característiques de les bases de dades objecte-relacionals.
    - Crea tipus de dades objecte, els seus atributs i mètodes.
    - Crea taules d'objectes i taules de columnes tipus objecte.
    - Crea tipus de dades col·lecció.
    - Realitza consultes.
    - Modifica la informació emmagatzemada mantenint la integritat i consistència de les dades.

Criteris d'avaluació

La puntuació màxima assolible a cada activitat s'indica a l'enunciat respectiu.

Els criteris que es tindran en compte per avaluar el treball de l'alumnat són els següents:

- La correcció i la completesa de les respostes
- La coherència i la bona estructuració de les respostes, així com la seva pulcritud

Forma i data de lliurament

Un cop finalitzat l'exercici d'avaluació continuada s'ha d'enviar el document a la bústia de **"Lliurament EAC7"** de l'aula, dins del termini establert. Tingueu en compte que el sistema no permetrà fer lliuraments després de la data i hora indicades.

El nom del fitxer tindrà el següent format:
"DA2_M02B2_EAC7_Cognom1_InicialDelCognom2.odt". Per tant, heu de lliurar l'arxiu en el format original: **document de text OpenOffice** (*.odt*).

Els cognoms s'escriuran sense accents. Per exemple, l'estudiant Joan García Santos posaria el següent nom al seu fitxer de l'EAC3: **"DA2_M02B2_EAC7_Garcia_S.odt"**.
Substituïu també **"Nom i cognoms"** de la capçalera per les vostres dades personals.

Podeu fer servir captures de pantalla i qualsevol altre recurs per documentar el vostre treball i aclarir les vostres proves, però és imprescindible que afegiu la **transcripció literal de les vostres respostes**, sigui codi PosgreSQL o llenguatge natural.

El termini de lliurament finalitzarà a les **23:55 h** del dia **10/05/2019.** La proposta de solució de l'EAC és publicarà el dia 13/05/2019 i les qualificacions el dia 16/05/2019.

Enunciat

Per a realitzar aquest EAC8 necessiteu tenir correctament instal·lats PostgreSQL 9.2 (o posterior) i l'eina pgAdmin III. La versió de PostgreSQL no ha de ser exactament aquesta, que podeu descarregar de:

http://www.enterprisedb.com/products-services-training/pgdownload#windows

però ha de ser una versió on funcioni la creació d'una taula a partir d'un tipus, és a dir, si creem el tipus CREATE TYPE xxx... hem de poder crear una taula amb la sentència CREATE TABLE yyy OF xxx...

Activitat 1. Característiques de bases de dades objecte-relacionals

1.1. **(1 punt)** Responeu al qüestionari que teniu a l'aula anomenat «Qüestionari UF4».

La puntuació de les respostes és la següent:

- Cada resposta correcta suma 0,10 punt.
- Cada resposta incorrecta resta 0,05 punts.
- Les respostes en blanc no puntuen.

1.2. Amb les vostres paraules, definiu i descriviu les característiques principals dels següents conceptes del model orientat a objectes **(0,5 punts)**

- Objectes
- Propietats
- Herència
- Polimorfisme

Resposta:

Objectes

Un objecte és un tipus de dades que encapsula les dades necessàries i les funcions per accedir-hi.

Un tipus objecte o classe defineix les característiques (atributs) i els comportaments (mètodes) que són comuns per a tots els objectes d'un cert tipus. Tots els objectes que comparteixen les mateixes propietats pertanyen al mateix tipus objecte. Per mitjà d'aquest es defineixen les propietats dels objectes i s'utilitza com a plantilla per crear-los. Anomenem instància de classe la utilització de la definició d'un tipus objecte per obtenir un determinat objecte. Com a concepte, la instància és equivalent a un tuple (en català n-pla) concret en una taula d'una base de dades.

En el món que ens envolta hi ha diferents objectes d'un mateix tipus -també podríem dir d'una mateixa classe. Per exemple, hi pot haver més d'un cotxe del mateix tipus. Si utilitzem la terminologia d'orientació a objectes, direm que el nostre cotxe és una instància del tipus objecte *cotxes*.

Propietats

Les propietats dels objectes poden ser dinàmiques o estàtiques. Els **atributs** representen una propietat estàtica d'un objecte. Els **mètodes** representen una propietat dinàmica, és a dir, una modificació sobre un atribut o una acció que es pot efectuar.

El conjunt de valors dels atributs en un moment donat es coneix com a **estat de l'objecte**. Els operadors actuen sobre l'objecte canviant-ne l'estat. La seqüència d'estats pels quals passa un objecte en executar els mètodes en defineixen el **comportament**.

Les propietats dels objectes són el seu estat i el seu comportament.

Herència

L'herència és la relació entre un tipus objecte general i un altre de més especific. Un tipus objecte es deriva de l'altre estenent la seva funcionalitat, essent el mecanisme fonamental per implementar el polimorfisme i la reutilització.

L'herència es defineix com el mecanisme pel qual s'utilitza la definició d'una classe anomenada *pare* per definir un nova classe anomenada *fill* que en pot heretar els atributs i les operacions.

Polimorfisme

El polimorfisme fa referència a la sobreescriptura de mètodes a l'hora d'implementar un tipus objecte utilitzant l'herència. No s'ha de confondre amb la sobrecàrrega de funcions que permeten altres llenguatges de programació.

1.3. **(0,5 punts)** Indiqueu quines diferències hi ha entre:

- Encapsulació i herència
- Polimorfisme i sobrecàrrega.

Resposta:

L'encapsulació simplifica la gestió d'un tipus objecte tot ocultant els detalls del seu funcionament, augmentant així el nivell d'abstracció i permetent modificacions internes de la implementació sense afectar el desenvolupament extern.

En canvi, l'herència apareix en la relació entre un tipus objecte general i un altre de més especific. Aquests últims hereten els atributs i mètodes del tipus objecte general, permetent reutilitzar un codi ja existent.

El polimorfisme fa referència a la sobreescriptura de mètodes a l'hora d'implementar un tipus objecte utilitzant l'herència. **En canvi, la sobrecàrrega** permet tenir diferents mètodes amb el mateix nom en un mateix tipus objecte.

1.4. L'objectiu d'aquesta activitat és conèixer les diferències entre **objectes, tipus d'objecte, atributs i mètodes. (1 punt)**

Identifiqueu els objectes, els tipus d'objecte, els atributs i els mètodes de la descripció següent:

Marina vol apuntar a la seva filla a algunes activitats extraescolars de l'escola de la seva filla. Es dirigeix a l'AMPA de la seva escola per tal que li informin de les activitats que hi pot fer, horaris, aules.... La persona que l'atén l'informa que per a poder accedir als serveis que ofereix l'AMPA ha de donar-se d'alta com a soci i fer-se el seu carnet. Li explica les diferents modalitats de carnet en funció del tipus de servei que li interessi:

- El "**activitats migdia**", que li permet accedir a les activitats extraescolars que es fan al migdia, per als nens que es queden a dinar a l'escola.
- El "**activitats global**", que li permet l'accés a totes les activitats extraescolars que es fan a l'escola, siguin al migdia o siguin a la tarda després de l'horari escolar. A més a més, amb aquesta modalitat de carnet també té accés al casal d'estiu que es fa al mes de juliol.

Li expliquen que, independentment del tipus de modalitat que es doni d'alta, per Internet pot consultar el seu número de soci, nom, adreça, relació d'activitats en les que la seva filla participi, consulta d'activitats que realitza.... També li explica que podrà fer inscripcions on-line de les diferents activitats a les que es vulgui apuntar.

Una resposta possible seria:

- **Tipus objecte:** carnet de soci AMPA
- **Objectes**: carnet modalitat activitats migdia i carnet modalitat activitats global
- **Atributs**: número de soci, nom del soci, adreça, entrades i sortides, activitats...
- **Mètodes**: consultar dades, fer inscripcions on-line...

Activitat 2. Taules d'objectes	**2 punts**

En aquesta activitat se us demana adaptar les dades de l'entitat «*Barcelona Enjoy*» amb la que heu treballat al llarg de la UF3. En aquest cas:

Els usuaris que es volen inscriure en alguna activitat de les que ofereix l'entitat, tenen una fitxa de **usuaris** amb la següent informació:

- identificador de l'usuari, DNI, nom, cognoms, adreça i correu electrònic.

De les **activitats** guarden la següent informació:

- Identificador de l'activitat, nom, descripció i preu.

Quan un usuari demana **inscriure's** a una activitat, li enregistren les següent dades:

- Identificador de la inscripció, identificador de l'usuari, identificador de l'activitat, data d'inici de l'activitat i la data de finalització de la mateixa.

Heu de crear els tipus d'objectes i taules d'objectes adients als requeriments que s'han marcat de l'entitat *«Barcelona Enjoy»*. Fixeu-vos que haureu de relacionar les taules (no fem servir oid)

A tal situació, és una condició necessària que les taules es derivin dels tipus d'objectes (create table … of) i que hi hagi la relació entre:

- fitxa de l'usuari i les inscripcions.
- inscripcions i activitats.

Afegiu també que no deixi eliminar un usuari si en té inscripcions associades.

No cal fer el disseny Entitat Relació, ni el relacional, heu de contestar amb els create type i create table adients.

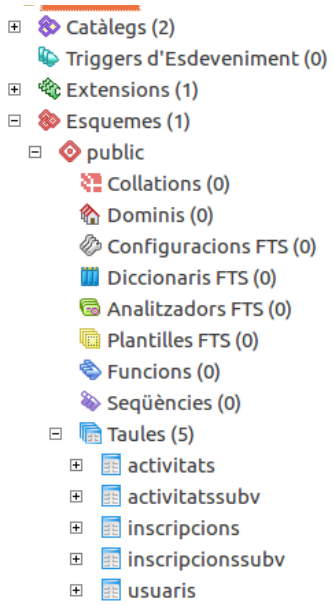
Feu-ho tot dins d'una nova base de dades amb nom «enjoy2**elvostrenom**», on «ElVostreNom» ha de ser el vostre nom de pila, per exemple: «xarxa2Yolanda».

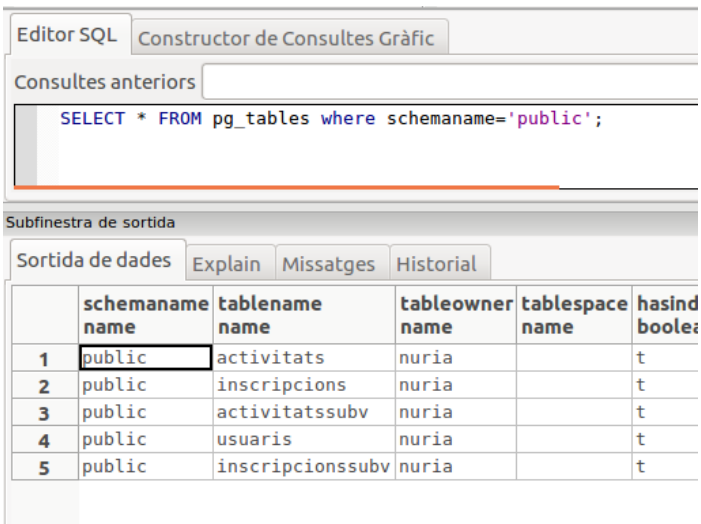
Resposta:

Feu dues captures de pantalla per cadascuna de les accions:

1. Comandes utilitzades per crear els tipus i les taules.
2. Captura de pantalla on es vegin els objectes creats a la base de dades, ja sigui per entorn gràfic, o per llistat de comandes.

```
CREATE TYPE usuari AS(
id_usuari integer,
dni varchar(15),
nom varchar(30),
cognoms varchar(30),
adreca varchar(50),
email varchar(30)
);

CREATE TYPE activitat AS(
id_activitat integer,
nom_act varchar(20),
descripcio text,
preu numeric (6,2)
);

CREATE TYPE inscripcio AS(
id_inscripcio integer,
id_user integer,
id_act integer,
data1 date,
data2 date
);


CREATE TABLE usuaris OF usuari (PRIMARY KEY (id_usuari));

CREATE TABLE activitats OF activitat (PRIMARY KEY (id_activitat));

CREATE TABLE inscripcions OF inscripcio(PRIMARY KEY (id_inscripcio),
CONSTRAINT FKAct FOREIGN KEY (id_act) REFERENCES activitats(id_activitat),
CONSTRAINT FkUser FOREIGN KEY (id_user) REFERENCES usuaris(id_usuari) MATCH
SIMPLE ON DELETE RESTRICT
);
```





	schemaname name	tablename name	tableowner name	tablespace name	hasind boolea
1	public	activitats	nuria		t
2	public	inscripcions	nuria		t
3	public	activitatssubv	nuria		t
4	public	usuaris	nuria		t
5	public	inscripcionssubv	nuria		t

Editor SQL | Constructor de Consultes Gràfic

Consultes anteriors | Esborra | Esborra-ho tot

```
select pg_type.typname,pg_attribute.attname from pg_type left join pg_namespace on
(pg_type.typnamespace = pg_namespace.oid)
                    join pg_class on (pg_type.typrelid=pg_class.oid)
                    join pg_attribute on
(pg_attribute.attrelid=pg_class.oid)
                    join pg_type otro on (otro.oid=pg_attribute.atttypid)
                    where pg_type.typcategory='C' and relkind='c' order
by pg_namespace.nspname
```

Subfinestra de sortida

Sortida de dades | Explain | Missatges | Historial

	typname name	attname name
1	usuari	id_usuari
2	usuari	dni
3	usuari	nom
4	usuari	cognoms
5	usuari	adreca
6	usuari	email
7	activitat	id_activitat
8	activitat	nom_act
9	activitat	descripcio
10	activitat	preu
11	inscripcio	id_inscripcio
12	inscripcio	id_user
13	inscripcio	id_act
14	inscripcio	data1
15	inscripcio	data2

Activitat 3. Herència	1 punts

Ara, el responsable de l'entitat *«Barcelona Enjoy»,* us ha comentat que volen diferenciar entre les **activitats no subvencionades** i les **activitats subvencionades** (venen subvencionades i avalades per l'àrea de turisme de l'ajuntament de Barcelona), per tal d'enregistrar la informació de més a més en aquests darrers.

A part de la informació general que guarden per totes les activitats, de les activitats subvencionades necessiten guardar:

- codi facilitat per l'àrea de turisme de l'ajuntament de Barcelona, de tipus *varchar*.
- nivell (dada numèrica que determina el nivell de subvenció atorgat a l'activitat)

A més a més, quan un usuari es vol inscriure en una activitat subvencionada, a part de les dades que s'enregistren en una inscripció normal, s'enregistra un camp més: «**punts**» (un camp de tipus numèric que indica quants punts ha obtingut l'usuari fent aquella activitat).

A tal situació, teniu **activitats**, **inscripcions**, **activitatsSubv** i **inscripcionsSubv.**

Heu de dissenyar una arquitectura de taules amb herència, per tal d'afegir aquest nou requeriment que se us demana, aprofitant l'estructura creada en l'anterior activitat.

A més d'heretar, haureu d'afegir també les particularitats de cadascuna. Finalment, heu de considerar que una *primary key* no s'hereta i llavors necessitareu repetir-les a les taules heretades. I el mateix succeeix amb les *claus foranes*.

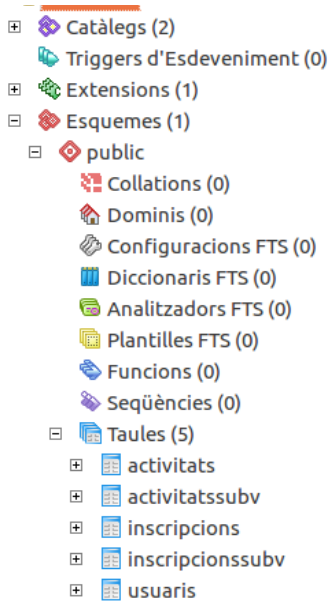
Resposta:

Feu dues captures de pantalla per cadascuna de les accions:

1. Comandes utilitzades per crear els tipus i les taules heretades.
2. Captura de pantalla on es vegin els objectes creats a la base de dades, ja sigui per entorn gràfic, o per llistat de comandes.

```
CREATE TABLE activitatsSubv(
codi_oficial varchar(10),
nivell integer,
PRIMARY KEY (id_activitat)
)INHERITS (activitats);
```

```
CREATE TABLE inscripcionsSubv(
 punts integer,
 PRIMARY KEY (id_inscripcio),
 CONSTRAINT FKActsub FOREIGN KEY (id_act) REFERENCES
 activitatsSubv(id_activitat),
 CONSTRAINT FkUsersub FOREIGN KEY (id_user) REFERENCES usuaris(id_usuari)
 MATCH  SIMPLE ON DELETE RESTRICT
)INHERITS (inscripcions);
```


Catàlegs (2)
Triggers d'Esdeveniment (0)
Extensions (1)
Esquemes (1)
public
Collations (0)
Dominis (0)
Configuracions FTS (0)
Diccionaris FTS (0)
Analitzadors FTS (0)
Plantilles FTS (0)
Funcions (0)
Seqüències (0)
Taules (5)
activitats
activitatssubv
inscripcions
inscripcionssubv
usuaris

Activitat 4. Sentències DML

Amb les taules anteriors, heu d'escriure les sentències que donen solució a les qüestions que es plantegen als següents apartats.

***Adjunteu una captura de pantalla amb el resultat de les sentències i amb un exemple que demostri que heu provat que el que feu funciona, amb diferents casuístiques.**

4.1. Inserció (1 punt)

a) Inseriu un parell d'usuaris a la vostra base de dades, un d'ells amb els vostre nom i cognoms. La resta de dades lliure.

b) Inseriu un parell d'activitats. Una no subvencionada i una altra de tipus «subvencionada».

c) Inseriu un parell d'inscripcions, una d'elles de tipus «subvencionada».

Resposta:

```
INSERT INTO xxx VALUES ( yyyy );

*caldrà tenir en compte que les dades de tipus alfanumèriques hauran
d'anar entre cometa simple.
```

100 registre

	id_activitat [PK] integer	nom_act character varying(20)	descripcio text	preu numeric(6,2)
1	10	activitat 10	paracaigudes	40.00
2	20	activitat 20	barranquisme	26.00
*				

100 registre

	id_activitat [PK] integer	nom_act character varying(20)	descripcio text	preu numeric(6,2)	codi_oficial character varying(10)	nivell integer
1	100	activitat subv 100	paracaigu	30.00	CE10	2
2	200	activitat subv 200	barranqui	15.00	BA32	1
*						

100 registre

	id_usuari [PK] integer	dni character varying(15)	nom character varying(30)	cognoms character varying(30)	adreca character varying(50)	email character varying(30)
1	1	22222x	Pep	Garcia	C/Mig	pgarcia@gmail.com
2	2	3333y	Laura	Lopez	C/Llarg	llopez@gmail.com
3	3	4444z	Sergi	Marti	Plaça 1	smarti@gmail.com

100 registre

	id_inscripcio [PK] integer	id_user integer	id_act integer	data1 date	data2 date
1	1	1	10	2017-03-22	2017-03-24
2	2	1	20	2017-03-25	2017-03-27
3	101	3	100	2017-04-10	2017-04-12
4	102	3	200	2017-05-05	2017-05-07
*					

100 registre

	id_inscripcio [PK] integer	id_user integer	id_act integer	data1 date	data2 date	estrelles integer
1	101	3	100	2017-04-10	2017-04-12	2
2	102	3	200	2017-05-05	2017-05-07	3
*						

4.2. Modificació (0,5 punts)

1. Modifiqueu el codi de l'activitat subvencionada que heu inserit en l'exercici anterior. El nou codi serà: 'C3EF'.
2. Modifiqueu la descripció de l'activitat no subvencionada que heu inserit en l'exercici anterior. La nova descripció serà: 'Ruta gastronòmica port vell'.

Resposta:

```
UPDATE activitatssubv set codi_oficial='C3EF' WHERE id_activitat=100;
```

Editor SQL | Constructor de Consultes Gràfic

Consultes anteriors

```
UPDATE activitatssubv set codi_oficial='C3EF' WHERE id_activitat=100;
```

Subfinestra de sortida

Sortida de dades | Explain | Missatges | Historial

```
Query returned successfully: one row affected, 12 msec execution time.
```

100 registre

	id_activitat [PK] integer	nom_act character varying(20)	descripcio text	preu numeric(6,2)	codi_oficial character varying(10)	nivell integer
1	100	activitat subv 100	paracaigudes subvencionat	30.00	C3EF	2
2	200	activitat subv 200	barranquisme subvencionat	15.00	BA32	1
*						

```
UPDATE activitats set descripcio='Ruta gastronòmica port vell' WHERE
id_activitat=10;
```

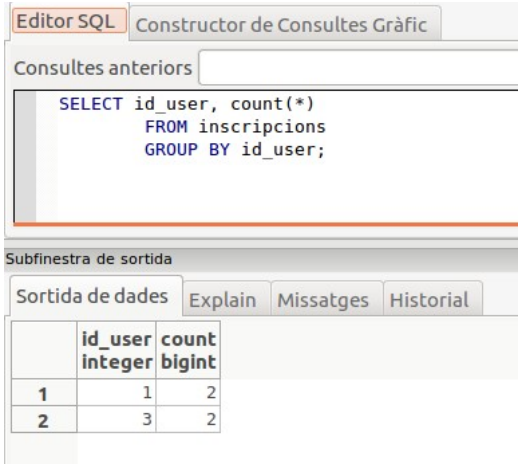
4.3. Selecció (1 punt).

a) Dissenyeu una sentència SELECT per mostrar el número d'activitats, siguin subvencionades o no, als que s'ha inscrit cada usuari. Heu de seleccionar l'identificador de l'usuari i el número d'inscripcions que ha fet.

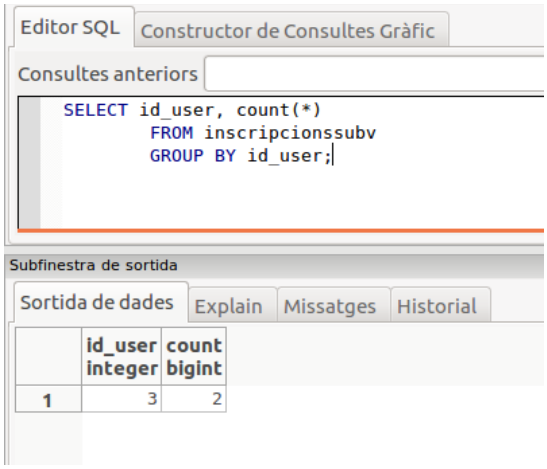
b) Dissenyeu ara, una sentència SELECT per mostrar el número d'activitats als que s'ha inscrit cada usuari però només els referents a activitats subvencionades. Heu de seleccionar l'identificador de l'usuari i el número d'inscripcions que ha fet.

Resposta:

```
SELECT id_user, count(*)
      FROM inscripcions
      GROUP BY id_user;
```




```
SELECT id_user, count(*)
      FROM inscripcionssubv
      GROUP BY id_user;
```



4.4 (1,5 punts) Implementeu una funció, anomenada **mostrar_inscripcions**, tal que, donat l'identificador d'un usuari, mostri totes les inscripcions que ha realitzat, mostrant la data de l'inici de l'activitat i l'identificador de l'activitat.

A més a més, si l'usuari s'ha inscrit a més de 5 activitats, també ha de mostrar, amb un RAISE, el nombre d'inscripcions que ha fet.

També heu de verificar que l'usuari existeixi (l'identificador que us passen és correcte), i en cas contrari, donar un missatge d'error. Aneu en compte perquè pot existir i no haver fet cap inscripció encara.

Resposta:

```
CREATE TYPE tinfo AS (
 data date,
 identificador integer
);

CREATE OR REPLACE FUNCTION mostrar_inscripcions (identif integer)
RETURNS SETOF tinfo AS $$

DECLARE
  resultat tinfo;
  quantitat integer;

BEGIN
--Comprovar que l'usuari existeixi
IF (identif NOT IN (SELECT id_usuari FROM usuaris)) THEN
  RAISE NOTICE 'Aquest usuari no existeix en la nostra base de dades';
ELSE
  --L'usuari existeix. Comptabilitzar quantes inscripcions ha fet
  quantitat=0;
  FOR resultat IN SELECT data1, id_act
                   FROM inscripcions
                  WHERE id_user = identif
  LOOP
    RETURN NEXT resultat;
    quantitat=quantitat+1;
  END LOOP;

  --Comprovar si té més de 5 inscripcions
  IF (quantitat > 5) THEN
    RAISE NOTICE 'Aquest usuari consta inscrit a % activitats', quantitat;
  END IF;
END IF;
RETURN;
END;
$$LANGUAGE plpgsql;
```

Taula inscripcions:

100 registre

	id_inscripcio [PK] integer	id_user integer	id_act integer	data1 date	data2 date
1	1	1	10	2017-03-22	2017-03-24
2	2	1	20	2017-03-25	2017-03-27
3	3	1	20	2017-01-01	2017-01-02
4	4	1	10	2017-02-02	2017-02-04
5	5	1	20	2017-03-03	2017-03-05
6	6	1	10	0207-04-15	2017-04-16
7	101	3	100	2017-04-10	2017-04-12
8	102	3	200	2017-05-05	2017-05-07
*					

L'usuari 1 s'ha inscrit a més de 5 activitats:

Editor SQL | Constructor de Consultes Gràfic

Consultes anteriors

```
select * from mostrar_inscripcions (1);
```

Subfinestra de sortida

Sortida de dades | Explain | Missatges | Historial

	data date	identificador integer
1	2017-03-22	10
2	2017-03-25	20
3	2017-01-01	20
4	2017-02-02	10
5	2017-03-03	20
6	0207-04-15	10

Si anem a la pestanya «Missatges»:

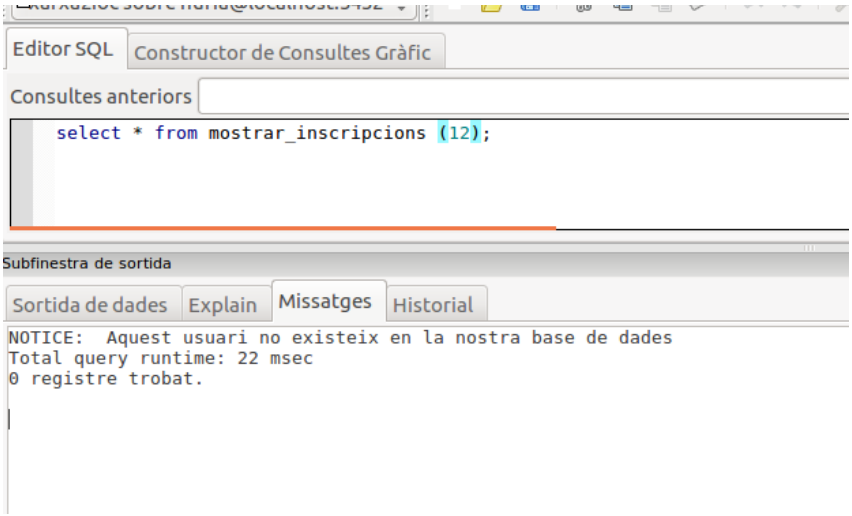
Editor SQL | Constructor de Consultes Gràfic

Consultes anteriors

```
select * from mostrar_inscripcions (1);
```

Subfinestra de sortida

Sortida de dades | Explain | Missatges | Historial

```
NOTICE:  Aquest usuari consta inscrit a 6 activitats
Total query runtime: 22 msec
6 registre trobat.
```

Si posem un identificador d'usuari que no existeix: