

# Ús de Bases de Dades

## PRÀCTICA 1: Disseny de Bases de Dades i implementació de vistes

### Proposta de solució

### Presentació

En aquesta Pràctica s'exerciten els aspectes que convé tenir en compte en el disseny d'una base de dades i la seva posterior implementació en un SGBD PostgreSQL.

### Competències

En aquesta Pràctica es desenvolupen les següents competències del Grau en Multimèdia:

- Capacitat d'analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per abordar-lo i resoldre'l.
- Capacitat per aplicar les tècniques específiques de tractament, emmagatzematge i administració de dades.

### Objectius

Els objectius concrets d'aquesta Pràctica són:

- Saber fer el disseny conceptual de les dades d'un sistema d'informació mitjançant el model ER.
- Saber fer el disseny lògic d'una base de dades relacional partint d'un disseny conceptual expressat amb el model ER.
- Conèixer el gestor de bases de dades relacionals amb suport per a objectes PostgreSQL.
- Completar l'estudi dels components lògics de dades, afegint el component base de dades i les vistes al component lògic de dades fonamental, les taules.

## Enunciat de la Pràctica:

Es vol dissenyar una base de dades per tal de realitzar la gestió de targetes de fidelitat d'un eix comercial.

Els establiments d'un eix comercial, per tal de d'atraure i consolidar la seva clientela habitual han decidit crear una targeta de fidelitat que permetrà als seus clients accedir a descomptes i regals en funció de les seves compres. Per a implantar aquest nou servei s'han posat en contacte amb una empresa especialitzada que s'encarregarà de facilitar les targetes i de crear el software necessari per a la gestió de la informació. Aquest software estarà compost per una base de dades de nova creació, a més d'un portal web per tal de facilitar a l'usuari l'accés a les dades, ofertes i promocions de les botigues i establiments.

La informació que es vol guardar tindrà en compte tots els establiments que formen l'eix comercial, indicant el nom, adreça, telèfon, latitud i longitud que conformen la seva posició geogràfica .

A més, es vol guardar la informació de la categoria a la que pertany l'establiment (cafeteria, forn de pa, ferreteria, fruiteria, ...). Cal tenir en compte que hi ha categories que englobaran d'altres més especialitzades (per exemple un forn pot ser de tipus artesanal o bio, una perruqueria pot ser unisex, d'homes, de dones, etc. i un taller mecànic pot estar especialitzat en motos, reparació de punxades, etc.). Cada categoria s'identifica per un valor autoincremental, tot i que el nom de la categoria és únic. Volem emmagatzemar una descripció i la categoria genèrica a la que pertany (si s'escau).

L'altra branca principal del projecte són els clients. Els actuals requeriments per a cada client de l'eix comercial inclouen guardar el nom i cognoms, l'adreça física, telèfon i una adreça de correu electrònic de contacte (que servirà d'identificador al portal), a més d'una contrasenya d'accés, la data en que es va donar d'alta i els punts de que disposa, que anirà guanyant a mesura que vagi realitzant compres i gastant al bescanviar-los pers regals o descomptes que s'ofereixen en les promocions . Un client no pot tenir un saldo negatiu de punts. Inicialment desconeixem com s'identificaran les targetes i com serà el software pel qual s'anirà afegint punts a cada client.

Els clients podran realitzar comentaris dels establiments a través del portal web. La informació que s'haurà de guardar per cada comentari és el seu text i la hora i la data en que es va realitzar. Cal considerar que un client pot realitzar diferents comentaris referents a més d'un establiment i que aquests també es poden realitzar en diferents instants de temps.

De forma opcional cada botiga pot facilitar el catàleg de productes o serveis que tenen a la venda, per tal de que els usuaris puguin tenir accés des del portal i planificar les seves compres. Per cada producte es guardarà només el codi de referència, el nom i una petita descripció i el seu preu. El codi de referència identifica unívocament un producte. El valor del codi de referència d'un producte

serà diferent en tots els catàlegs de les diferents botigues. No interessa conèixer els diferents preus que ha tingut el producte.

Finalment cada botiga podrà oferir una sèrie de promocions a les que els usuaris tindran dret depenent de la quantitat de punts que hagin anat acumulant a través de les seves compres. Per simplificar el sistema es suposarà que les promocions tant sols permetran oferir descomptes sobre un dels productes oferts. Per a cada promoció es vol guardar l'establiment que l'ofereix, la referència del producte que s'ofereix, la descripció de la promoció, el seu cost en punts, sempre superior a zero, i un atribut que ens permeti conèixer si la promoció és vigent o no. Un usuari tant sols podrà aprofitar una determinada promoció un cop al dia i en cas de que tingui suficients punts acumulats.

La funcionalitat corresponent a l'actualització del nombre de punts d'un client en funció dels punts de les promocions que va aprofitant aquest client s'ha deixat pendent de definició.

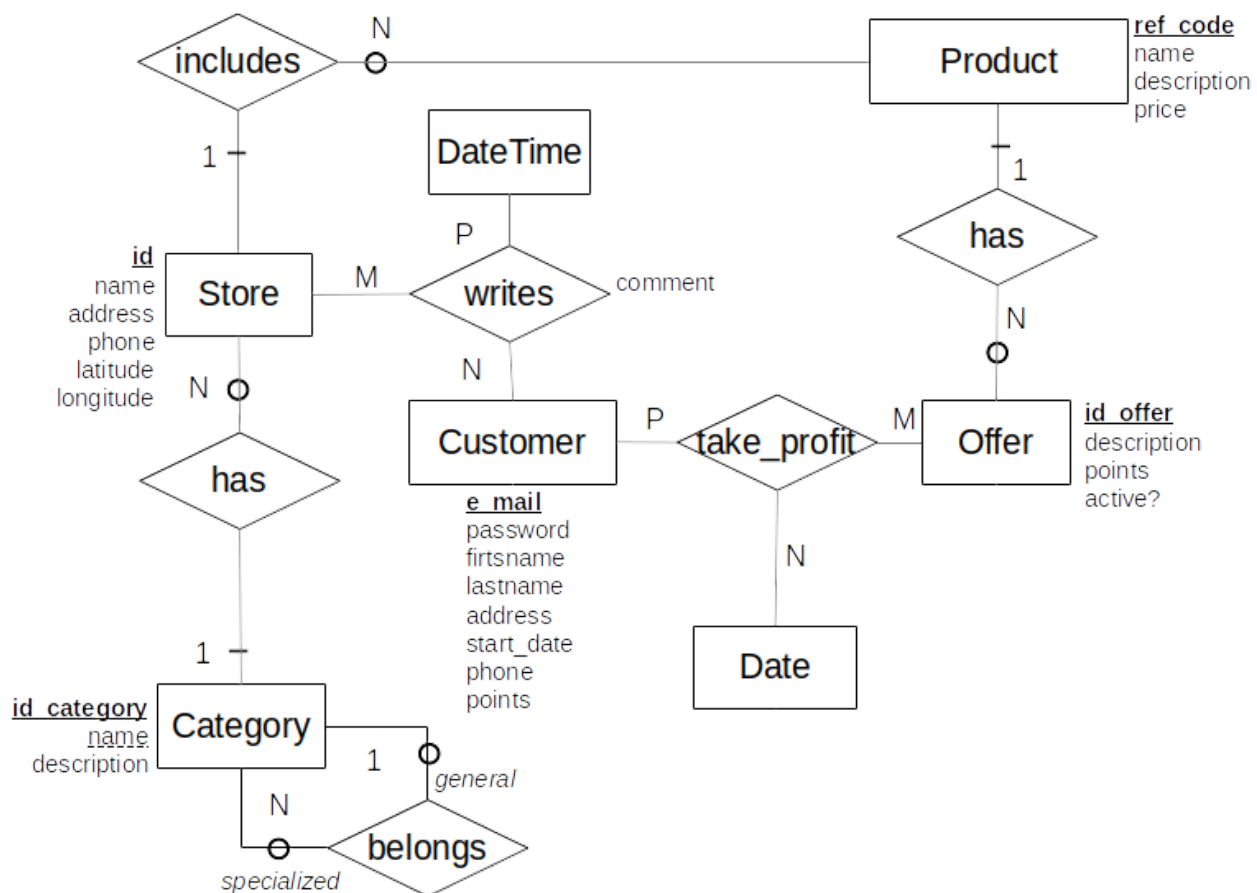
A partir d'aquesta descripció, cal respondre els següents exercicis de forma incremental.

Nota: Si es considera que manca alguna informació per dissenyar el model, l'estudiant pot realitzar les suposicions que consideri oportunes, sempre que no es contradiguin amb l'enunciat i que es documentin adequadament.

## Pregunta 1 [ 25 % ]

Crea el diagrama entitat-interrelació (E-R) a partir de les especificacions anteriors. Descriu els atributs de cada tipus d'entitats i dels tipus d'interrelacions que hi ha entre elles, així com els requisits que no han quedat reflectits en el supòsit proposat. Comenta i justifica les decisions preses en aquest punt, així com les possibles alternatives del disseny. Si s'ha realitzat alguna suposició semàntica addicional també s'ha d'indicar.

*Proposta de solució:*



## Pregunta 2 [ 25 % ]

Crea una base de dades en un SGBD PostgreSQL (el nom de la base de dades serà el teu *identificador d'usuari*) basada en el disseny anterior, que compleixi els requeriments esmentats. Comenta i justifica les decisions preses en aquest pas i inclou les sentències SQL que permeten la creació dels components de la base de dades, dins d'un esquema implementat a propòsit anomenat *pra1*. Identifica si hi ha alguna especificació corresponent a la presa de requisits que no s'hagi pogut especificar completament en el model lògic.

**Proposta de solució:**

```
CREATE DATABASE jperezbr;

CREATE SCHEMA pra1;

CREATE table pra1.category(
    id_category serial,
    name varchar(50) not null,
    description text,
    general_category integer,
    CONSTRAINT category_pk PRIMARY KEY (id_category),
    CONSTRAINT general_category_fk FOREIGN KEY (general_category) REFERENCES
pra1.category(id_category),
    CONSTRAINT category_uq UNIQUE (name)
);

CREATE table pra1.store(
    id integer,
    name varchar(50) not null,
    address varchar(100) not null,
    phone char(9),
    latitude float,
    longitude float,
    category integer not null,
    CONSTRAINT store_pk PRIMARY KEY (id),
    CONSTRAINT store2category_fk FOREIGN KEY (category) REFERENCES
pra1.category(id_category)
);

CREATE table pra1.product(
    ref_code integer,
    name varchar(50) not null,
    description text not null,
    price numeric(10,2),
    store integer not null,
    CONSTRAINT product_pk PRIMARY KEY (ref_code),
    CONSTRAINT product2store_fk FOREIGN KEY (store) REFERENCES pra1.store(id),
    CONSTRAINT valid_price CHECK (price > 0.0)
```

```

);

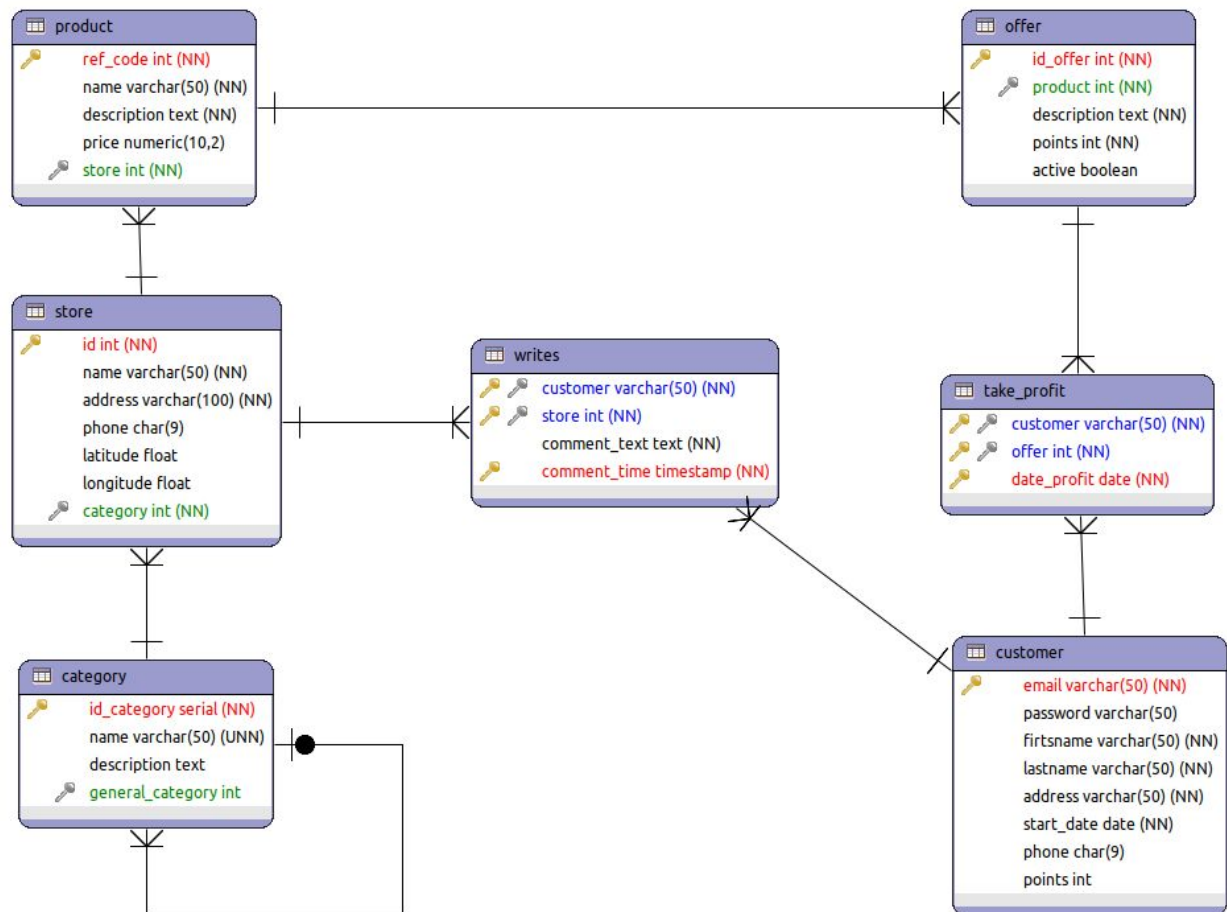
CREATE table pral.offer(
    id_offer integer,
    product integer not null,
    description text not null,
    points integer not null,
    active boolean DEFAULT true,
    CONSTRAINT offer_pk PRIMARY KEY (id_offer),
    CONSTRAINT offer2product_fk FOREIGN KEY (product) REFERENCES
pral.product(ref_code),
    CONSTRAINT valid_points_offer CHECK (points > 0)
);

CREATE table pral.customer(
    email varchar(50),
    password varchar(50),
    firtsname varchar(50) not null,
    lastname varchar(50) not null,
    address varchar(50) not null,
    start_date date not null,
    phone char(9),
    points integer,
    CONSTRAINT customer_pk PRIMARY KEY (email),
    CONSTRAINT valid_points_customer CHECK (points >= 0)
);

CREATE table pral.writes(
    customer varchar(50) not null,
    store integer not null,
    comment_text text not null,
    comment_time timestamp not null,
    CONSTRAINT writes_pk PRIMARY KEY (customer,store,comment_time),
    CONSTRAINT writes2store_fk FOREIGN KEY (store) REFERENCES pral.store(id),
    CONSTRAINT writes2customer_fk FOREIGN KEY (customer) REFERENCES
pral.customer(email)
);

CREATE TABLE pral.take_profit(
    customer varchar(50) not null,
    offer integer not null,
    date_profit date not null,
    CONSTRAINT take_profit_pk PRIMARY KEY (customer,offer,date_profit),
    CONSTRAINT profit2offer_fk FOREIGN KEY (offer) REFERENCES pral.offer(id_offer),
    CONSTRAINT profit2customer_fk FOREIGN KEY (customer) REFERENCES
pral.customer(email)
);

```



## Pregunta 3 [ 10 % ]

Identifica la clau primària i claus alternatives de cadascuna de les taules. Justifica la teva elecció i comenta possibles alternatives en aquest sentit, incloent la possibilitat de crear claus primàries artificials.

*Proposta de solució:*

Seguint els requeriments de la pràctica és necessari crear claus artificials. Les claus primàries i claus alternatives de cada taula són les següents:

TABLE	PRIMARY KEY	ALTERNATIVE KEY
category	id_category ( <i>artificial</i> )	name
store	id ( <i>artificial</i> )	
product	id_product ( <i>artificial</i> )	
offer	id_offer ( <i>artificial</i> )	
customer	email	
writes	customer,store,comment_time	
take_profit	customer,offer,date_profit	



## Pregunta 4 [ 10 % ]

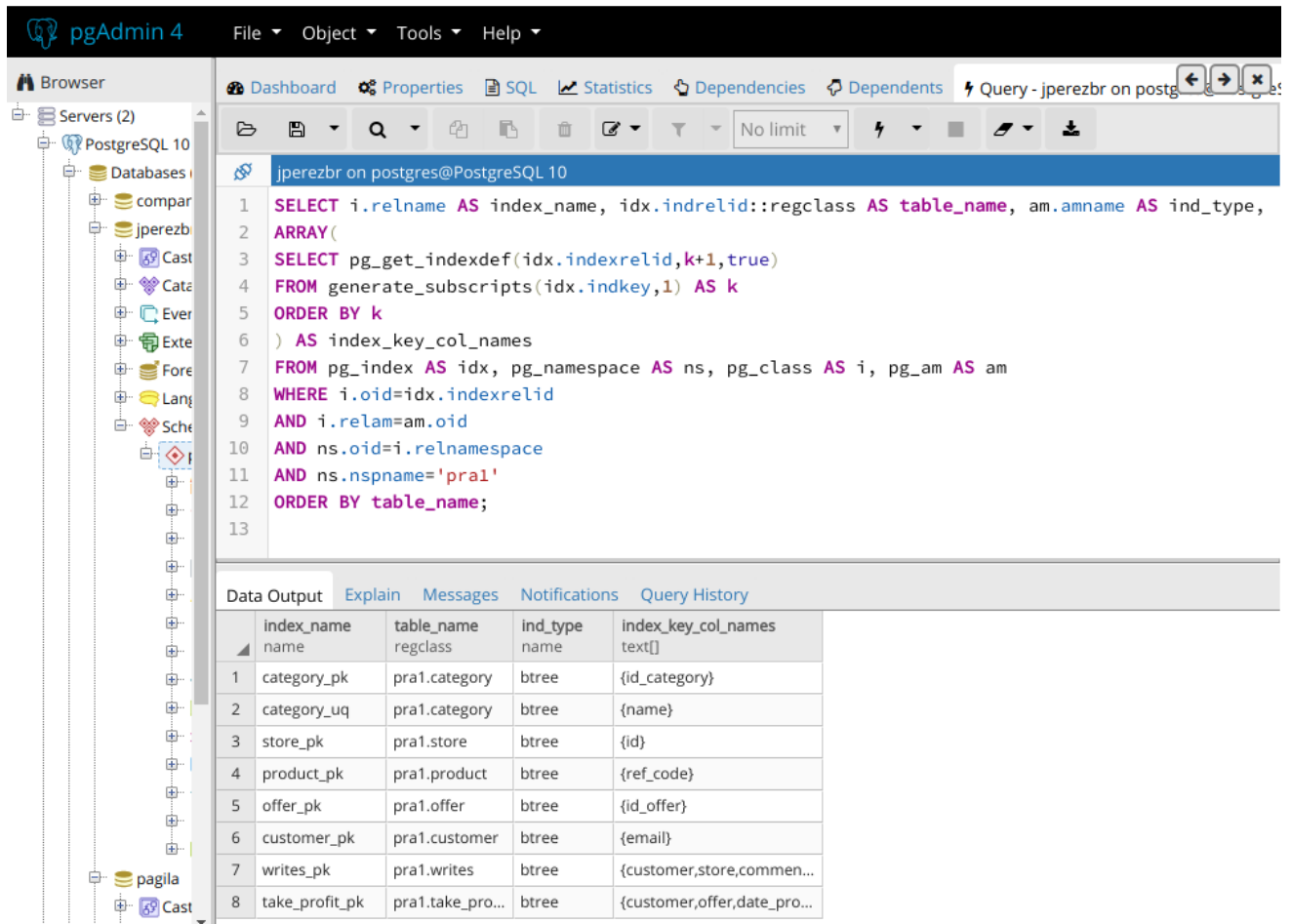
Suposant que el registre de clients serà molt nombrós, crea un índex per tal que la consulta corresponent a obtenir les dades dels clients en funció del nombre de punts que tenen acumulats, tingui el menor cost. A continuació identifica tots els índexs presents en la base de dades. Justifica'ls i comenta la seva funció.

*Proposta de solució:*

La sentència de creació de l'índex esmentat serà la següent:

```
CREATE INDEX customer_x_points_idx ON pra1.customer (points);
```

Posteriorment realitzem la consulta per tal de conèixer els índexs presents a la base de dades:



The screenshot shows the pgAdmin 4 interface. The SQL query editor contains the following query:

```
1 SELECT i.relname AS index_name, idx.indrelid::regclass AS table_name, am.amname AS ind_type,
2 ARRAY(
3 SELECT pg_get_indexdef(idx.indexrelid,k+1,true)
4 FROM generate_subscripts(idx.indkey,1) AS k
5 ORDER BY k
6 ) AS index_key_col_names
7 FROM pg_index AS idx, pg_namespace AS ns, pg_class AS i, pg_am AS am
8 WHERE i.oid=idx.indexrelid
9 AND i.relam=am.oid
10 AND ns.oid=i.relnamespace
11 AND ns.nspname='pra1'
12 ORDER BY table_name;
```

The query results are displayed in a table with the following columns: index\_name, table\_name, ind\_type, and index\_key\_col\_names. The results are as follows:

index_name	table_name	ind_type	index_key_col_names
category_pk	pra1.category	btree	{id_category}
category_uq	pra1.category	btree	{name}
store_pk	pra1.store	btree	{id}
product_pk	pra1.product	btree	{ref_code}
offer_pk	pra1.offer	btree	{id_offer}
customer_pk	pra1.customer	btree	{email}
writes_pk	pra1.writes	btree	{customer,store,commen...
take_profit_pk	pra1.take_pro...	btree	{customer,offer,date_pro...

Es pot comprovar els índexs creats per a cada taula. Els índexs que finalitzen amb `_pk` corresponen a la clau primària i els que finalitzen amb `_uq` a la clau alternativa.

També apareix l'índex `customer_x_points_idx` que s'ha creat anteriorment.

## Pregunta 5 [ 15 % ]

Crea una vista que permeti obtenir, per a cada oferta activa, el nom de la categoria a la que pertany l'establiment, el nom de l'establiment, el nom del producte i el número de punts necessaris per gaudir de la oferta, ordenant el resultat pel nombre de punts segons un criteri descent. En particular caldrà mostrar la següent informació:

<i>category_name</i>	<i>store_name</i>	<i>product_name</i>	<i>points</i>
----------------------	-------------------	---------------------	---------------

Posteriorment realitza una consulta SQL emprant la vista definida anteriorment per tal d'obtenir per a una categoria determinada, el nom de l'establiment, el nom del producte i els punts, d'aquelles ofertes en que el nombre de punts requerits per gaudir-ne sigui inferior a una quantitat donada.

*Proposta de solució:*

La sentència de creació de la vista seria la següent:

```
CREATE VIEW pral.exercise5 (category_name,store_name,product_name,points)
AS
SELECT cat.name , st.name, p.name, o.points
FROM pral.category cat, pral.store st, pral.product p, pral.offer o
WHERE cat.id_category=st.category
AND st.id=p.store
AND p.ref_code=o.product
AND o.active=true
ORDER BY o.points DESC;
```

Per tal de verificar la funcionalitat de la vista executariem:

```
SELECT * FROM pral.exercise5;
```

Es pot realitzar la consulta plantejada sobre la vista amb la següent sentència:

```
SELECT store_name,product_name,points
FROM pral.exercise5
WHERE category_name LIKE 'Bakery'
AND points < 5;
```

## Pregunta 6 [ 15 % ]

Crea una vista actualitzable que permeti visualitzar la concatenació del nom de l'establiment i el nom del producte, l'identificador de la oferta i l'estat de la oferta. és a dir, si és vigent o no. Cal que permeti actualitzar el valor corresponent a la vigència de la oferta.

Caldrà mostrar la següent informació:

offer	id	active
Golden Bakery - Hamburger Bun	1	true
Golden Bakery - Apple pie	8	false

Cal definir la vista, quines sentències permetran que la vista sigui actualitzable, realitzar una sentència d'actualització emprant la vista i mostrar el resultat d'aquesta actualització.

*Proposta de solució:*

La sentència de creació de la vista seria la següent:

```
CREATE OR REPLACE VIEW pral.exercise6( offer, id, active)
AS
    SELECT st.name||' - '|| p.name, o.id, o.active
    FROM pral.store st, pral.product p, pral.offer o
    WHERE st.id=p.store
    AND p.ref_code=o.product
    ;
```

La sentència de creació de la regla per tal de poder realitzar l'actualització seria la següent:

```
CREATE RULE exercise6_rule AS ON UPDATE TO pral.exercise6
DO INSTEAD
UPDATE pral.offer SET active=NEW.active WHERE id_offer=NEW.id ;
```

La sentència per tal de verificar el funcionament de la regla:

```
UPDATE pral.exercise6 SET active=true
WHERE id=8901223;
```

## Recursos

Els següents recursos són d'utilitat per la realització de la Pràctica:

### Bàsics

- Document "Bases de dades en PostgreSQL"

### Complementaris

- Document "Disseny de bases de dades".
- <https://www.postgresql.org/docs/10/index.html>
- [http://cv.uoc.edu/app/mediawiki64/wiki/Exercicis\\_amb\\_PostgreSQL](http://cv.uoc.edu/app/mediawiki64/wiki/Exercicis_amb_PostgreSQL)

## Criteris d'avaluació

A continuació podeu consultar el pes de cadascun dels exercicis proposats en la nota final de la pràctica:

Exercici 1	25%
Exercici 2	25%
Exercici 3	10%
Exercici 4	10%
Exercici 5	15%
Exercici 6	15%

El lliurament s'haurà de realitzar en l'apartat de lliuraments en un fitxer de text (pdf o similar) que contingui:

- Les respostes a les preguntes plantejades.
- El diagrama entitat-interrelació corresponent al disseny conceptual
- Les sentències en SQL plantejades i les respostes obtingudes.
- La justificació i criteris de disseny utilitzats en cada resposta.

**IMPORTANT:** Addicionalment caldrà implementar la pràctica al servidor <https://ubd.eimt.uoc.edu/eimtbdl/>. Cada alumne disposarà d'un usuari, que es proporcionarà individualment. És necessari que prèviament es realitzin les proves en local, a la instal·lació feta per vosaltres. A mesura que es vagin completant les tasques encomanades i ja es considerin

definitives, es recomana anar implementant les instruccions corresponents a la base de dades que se us proporcionarà en aquest servidor.

No es considerarà vàlid el codi que no es trobi implementat en el servidor i justificat a la memòria de la pràctica.

És molt important incloure el suficient nombre de registres en les diferents taules creades al servidor per verificar la funcionalitat de les vistes implementades i justificar el test realitzat de forma documentada.

## Notes

Aquesta PRA s'ha de fer de manera estrictament individual. Qualsevol indicatiu de còpia serà penalitzat amb un suspens (D) per a totes les parts implicades i la possible avaluació negativa de l'assignatura en la seva totalitat.

Cal que l'estudiant citi totes les fonts que ha utilitzat per a la realització de la PRA. Si no és així, es considerarà que l'estudiant ha comès plagi, sent penalitzat amb un suspens (D) i la possible avaluació negativa de l'assignatura en la seva totalitat.

La data límit per al lliurament d'aquesta PRA és el dia 29 d'abril de 2019.

La data de lliurament d'aquesta PRA ha de ser estrictament respectada, i no s'acceptarà cap lliurament després de la data establerta. Si es considera per alguna raó justificada que no es va a poder complir amb aquesta data, l'estudiant s'haurà de posar en contacte amb el seu consultor de l'assignatura **AMB SUFICIENT ANTERIORITAT** per poder buscar conjuntament una solució al respecte. Si s'acorda el lliurament amb posterioritat, la nota màxima d'aquesta PRA serà un aprovat (C+).