



Bases de dades relacionals

Daniel Villegas González
Grau en Enginyeria Informàtica
Bases de dades

Jordi Ferrer Duran
María Isabel Guitart Hormigo

13/06/2016



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Bases de dades relacionals</i>
Nom de l'autor:	<i>Daniel Villegas González</i>
Nom del consultor/a:	<i>Jordi Ferrer Duran</i>
Nom del PRA:	<i>María Isabel Guitart Hormigo</i>
Data de lliurament (mm/aaaa):	<i>06/2016</i>
Titulació o programa:	<i>Grau en Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Bases de dades</i>
Idioma del treball:	<i>Català</i>
Paraules clau	Postgresql, SQL, bases de dades.

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

Les empreses de reclutament de personal han d'afrontar una sèrie de reptes per les seves tasques diàries, pel gran volum de candidats per cobrir les necessitats del mercat de llocs de treball. RecruitCAT és una empresa de reclutament de personal que necessita gestionar la seva base de dades.

El objectiu és dissenyar i implementar una base de dades per gestionar una empresa de reclutament de personal, assolint les tasques d'estudi, plantejament teòric i els scripts de generació de la base de dades.

Com a gestor de base de dades s'usarà PostgreSQL; el llenguatge PL/PGSQL; PGAdmin III com gestor visual de base de dades; Open Project com gestor de projectes i Gimp com editor gràfic. Microsoft Office per taules i documents de ofimàtica; com eina CASE Magic Draw.

S'obté una base de dades amb la capacitat de gestionar les seues de l'empresa, els comercials, els clients i els candidats, així com les relacions entre cada una de les entitats. Per tal de verificar el correcte funcionament del sistema ABM es realitzen 183 proves dels quals 30 han donat error (83.61% correcte).

La creació d'una base de dades per gestionar una empresa de contractació proporciona una millor gestió de dades.

Abstract (in English, 250 words or less):

Recruitment companies are facing a number of challenges in their daily tasks, due to the large volume of candidates to cover the needs of the job market. RecruitCAT is a recruitment company that needs to manage its database.

The objective is to design and implement a database to manage a recruitment company, managing the work of study, theoretical plan and the scripts for generating the database.

PostgreSQL will be used as a database manager; the language PL/pgSQL; PGAdmin III as visual database manager; Open Project as project manager and as graphic editor Gimp. Microsoft Office for office tables and documents and Magic Draw as CASE tool.

You obtain a database with the ability to manage their business centre, sales person, clients and candidates, as well as the relationships between each of the entities. To verify the correct operation of the ABM system performed 183 tests of which 30 have error (83.61% correct).

Creating a database to manage a recruitment company provides better data management.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball	1
1.3 Enfocament i mètode seguit.....	2
1.4 Planificació del Treball	2
1.5 Breu sumari de productes obtinguts.....	5
1.6 Breu descripció dels altres capítols de la memòria	5
2. Desenvolupament	7
2.1 Model UML.....	7
2.2 Tipus	12
2.3 Taules	12
2.4 Funcions i disparadors	15
2.5 Estadístiques.....	19
2.5.1 Estadística a (est_a)	21
2.5.2 Estadística b (est_b)	21
2.5.3 Estadística c (est_c).....	22
2.5.4 Estadística d (est_d)	22
2.5.5 Estadística e (est_e, est_e_temp).....	22
2.5.6 Estadística f (est_f)	22
2.5.7 Estadística g (est_g)	23
2.5.8 Estadística h (est_h)	23
2.5.9 Estadística i (est_i).....	23
2.5.10 Estadística j (est_j).....	23
2.5.11 Estadística k (est_k).....	23
2.5.12 Estadística l (est_l).....	23
2.5.13 Estadística m (est_m).....	24
2.5.14 Estadística n (est_n).....	24
2.6 Log	24
2.7 Proves	25
3. Conclusions.....	36
4. Glossari	37
5. Bibliografia.....	38
6. Annexos	39

Llista de figures

Figura 1. Diagrama de Gantt inicial.	3
Figura 2. Diagrama de Gantt final.	4
Figura 3. Diagrama UML	7
Figura 4. Secció seus empresa.	7
Figura 5. Secció candidats.	8
Figura 6. Secció clients.	9
Figura 7. Secció comercials.	9

1. Introducció

1.1 Context i justificació del Treball

Les empreses de reclutament de personal ofereixen els seus serveis amb la finalitat d'oferir el millor candidat per les vacants que altres empreses tenen. S'ha de crear la base de dades que gestiona la empresa de reclutament de personal per tal de donar suport als usuaris de l'empresa RecruitCAT.

Actualment les empreses de reclutament de personal fan servir aplicacions per gestionar els candidats on es poden donar de alta i de baixa a les diferents ofertes de treball i són els candidats els que executen les accions.

Es pretén realitzar l'estudi de la base de dades necessària per tal d'abordar la necessitat presentada per l'empresa de selecció de personal RecruitCAT.

Per una banda es necessari tenir una base de dades de perfils dels candidats per tal de donar una resposta ràpida als clients; per altra banda, mantenir les interaccions mantingudes amb els clients. De la mateixa forma ha de donar el suport necessari als usuaris de la base de dades per generar informes i consultes, amb la finalitat de portar un control del bon funcionament de l'empresa, i poder així oferir el millor candidat abans que altres empreses de reclutament de personal.

1.2 Objectius del Treball

El principal objectiu és dissenyar i implementar una base de dades per tal de gestionar una empresa de reclutament de personal. Es volen assolir les tasques d'estudi, plantejament teòric i els scripts de generació de la base de dades.

Es portarà a terme l'estudi de totes les necessitats que la empresa ens ha transmès i per tal de donar solució:

- Es generarà els models conceptuals en format UML per iniciar l'estudi.
- Es descriuran i s'implementarà els procediments d'Alta, Baixa y Modificació que es faran servir per tal efecte.
- S'implementarà el repositori estadístic amb el propòsit de poder obtenir les dades estadístiques.
- Es generaran els scripts de generació de la base de dades així com la inicialització del conjunt de dades.
- Es generarà un conjunt de proves per tal de donar suport al manteniment del sistema així com el control de errors i excepcions.

1.3 Enfocament i mètode seguit

Com a estratègia a seguir es pot buscar una solució preparada i adaptar-la a les nostres necessitats però donada la envergadura del projecte no sembla la més adient. Com punt inicial sembla que desenvolupar un producte nou pot aportar una solució òptima.

Es desenvoluparà un producte nou generant les taules, disparadors i procediments emmagatzemats necessaris per tal de assolir els objectius plantejats adaptant així la solució a les particularitats de la aplicació demanda.

D'aquesta forma s'assegura que s'oferiran totes les funcionalitats demandades pel client sense haver de testejar solucions preparades per tercers havent de fer així adaptacions que ens ocuparà el mateix temps que preparar una aplicació des de zero.

Com a pla de contingència es mantindrà una còpia actualitzada de la carpeta que contindrà les dades del projecte al núvol i es treballarà amb la còpia local, fent còpies cada dia de treball. D'aquesta forma es podrà recuperar la última còpia en cas de desastre.

Es té disponible dues connexions a Internet, una de FTTH i una altra de banda ampla de mòbil. Podent fer servir indistintament qualsevol de les connexions en qualsevol dels dispositius, oferint així mobilitat en cas que fos necessari.

Referent al maquinari es compta amb dos ordinadors, un de sobretaula i un portàtil podent fer servir indistintament qualsevol dels dos. Donat que es té instal·lat Postgresql i PGAdmin III als dos, i tots dos poden accedir a les dades al núvol. En cas que fos necessari desplaçar-se per treball o altra motiu es podrà continuar treballant amb la mateixa assiduitat.

Si per malaltia o qualsevol altre motiu que no permetés continuar amb el treball amb el mateix ritme que s'ha proposat a la planificació del treball es recuperarà el temps els caps de setmana o dies festius quan sigui necessari.

1.4 Planificació del Treball

S'ha procurat utilitzar el major nombre de eines de software lliure per tal de donar solució i portar a terme el projecte i per aquest motiu s'han utilitzat les eines següents, un ordinador amb Postgresql que és un SGBD i aquest mantindrà la base de dades i PGAdmin III per tal de facilitar la creació de scripts de la base de dades. Per tal de generar el diagrama UML i com eina CASE es farà servir Magic Draw donat que permet generar les classes necessàries per tal de planificar la distribució

de la base de dades, per el diagrama de GANTT es farà servir Open Project que és una eina de programari sota la llicència Common Public Attribution License, amb GIMP es generaran les imatges necessàries i amb el paquet de ofimàtica de Microsoft es generarà les taules i documents necessaris.

Primerament s'abordarà la creació del diagrama en UML per tal de poder continuar amb el desenvolupament del projecte. Una vegada s'hagi arribat al plantejament lògic de la aplicació es començarà a implementar les taules al SGBD, a partir d'aquest moment s'abordarà el procés de creació del sistema ABM sense tindre en compte el sistema de excepcions. Seguidament el sistema de estadístiques amb els disparadors, taules i taules temporals necessàries. A continuació s'inicialitzarà les dades de la base de dades per tal de comprovar el correcte funcionament de la base de dades i per últim s'implementarà el control de errors i excepcions.

Es seguirà el guió marcat inicialment pel diagrama de GANTT següent de forma que:

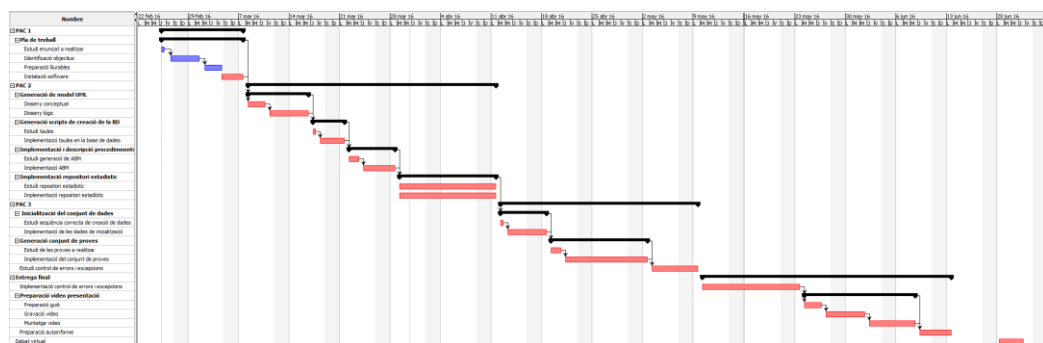


Figura 1. Diagrama de Gantt inicial.

En el transcurs del projecte s'introdueixen canvis en quant a la realització de les tasques de control de errors es refereixen; s'ha trigat menys temps de l'estipulat inicialment ja que s'ha simplificat el procés de implementació de control de errors del que inicialment es va plantejar, aconseguint així arribar a qualsevol error que es donés canviant el següent codi on només es tenia en compte 4 errors:

```
EXCEPTION
  WHEN not_null_violation THEN
    RSP = 'ERROR TOTS ELS CAMPS SON REQUERITS.';
  WHEN foreign_key_violation THEN
    RSP = 'ERROR NO ES POT ELIMINAR EXISTEIXEN DEPENDENCIES SOBRE
EL REGISTRE.';
  WHEN string_data_right_truncation THEN
    RSP = 'ERROR S'HA SUPERAT EL NOMBRE MAXIM DE CARACTERS
PERMESOS.';
  WHEN unique_violation THEN
    RSP = 'ERROR YA EXISTEIX EL REGISTRE.';
```

Per:

```
EXCEPTION
  WHEN OTHERS THEN
```

```
RSP = SQLERRM;
INSERT INTO uoc_recrutcat.log
VALUES('nom_funció', [CAMPS DE LA FUNCIO] , 'ERROR '||RSP);
```

Aconseguint d'aquesta forma capturar qualsevol error que es doni. El temps s'ha aprofitat per a millorar el detall de les explicacions dels lliurables. Donant com a resultat el següent diagrama de Gantt.

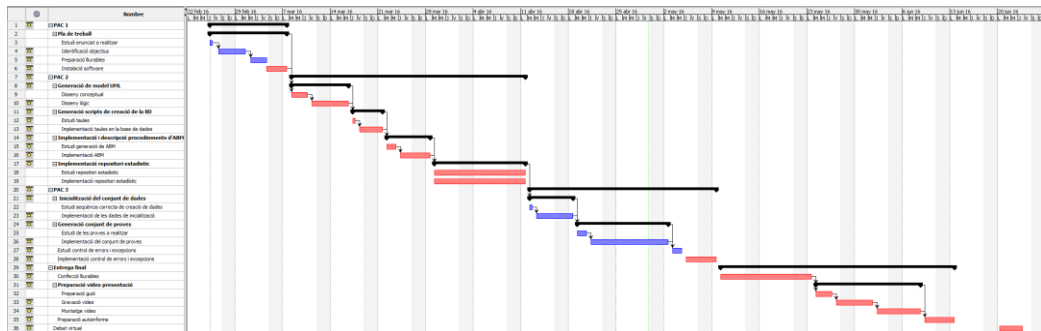


Figura 2. Diagrama de Gantt final.

Si desglossem els dies en hores de treball ens dona un total de 232 h de treball:

PAC 1	7 dies	21 h
Pla de treball	7 dies	21 h
Estudi enunciat a realitzar	1 dia	3 h
Identificació objectius	2 dies	6 h
Preparació lliurables	3 dies	9 h
Instal·lació software	1 dia	3 h
PAC 2	25 dies	75 h
Generació de model UML	7 dies	21 h
Disseny conceptual	3 dies	9 h
Disseny lògic	4 dies	12 h
Generació scripts de creació de la BD	3 dies	9 h
Estudi taules	1 dia	3 h
Implementació taules en la base de dades	2 dies	6 h
Implementació i descripció procediments d'ABM	5 dies	15 h
Estudi generació de ABM	2 dies	6 h
Implementació ABM	3 dies	9 h
Implementació repositori estadístic	10 dies	30 h
Estudi repositori estadístic	10 dies	15 h
Implementació repositori estadístic	10 dies	15 h
PAC 3	20 dies	60 h
Inicialització del conjunt de dades	5 dies	15h
Estudi seqüència correcta de creació de dades	1 dia	3 h
Implementació de les dades de inicialització	4 dies	12 h
Generació conjunt de proves	10 dies	30 h
Estudi de les proves a realitzar	2 dies	6 h
Implementació del conjunt de proves	8 dies	24 h
Estudi control de errors i excepcions	2 dies	6 h

Implementació control de errors i excepcions	3 dies	9 h
Entrega final	25 dies	75 h
Confecció lliurables	10 dies	30 h
Preparació vídeo presentació	12 dies	36 h
Preparació guió	3 dies	9 h
Gravació vídeo	4 dies	12 h
Muntatge vídeo	5 dies	15 h
Preparació autoinforme	3 dies	9 h
Debat virtual	4 dies	1 h
TOTAL HORES		232 h

1.5 Breu sumari de productes obtinguts

S'obté la memòria amb els procediments seguits per tal de finalitzar el projecte. A més els arxius de generació de la base de dades:

1. esquema i taules: On es troben els scripts necessaris per generar l'esquema i les taules de la base de dades.
2. funcions ABM: On es troben tots els scripts per tal de generar tots els procediments emmagatzemats que s'encarreguen de donar de alta, modificar i donar de baixa dades a les taules.
3. funcions i disparadors d'estadístiques: On es troben tots els procediments emmagatzemats i disparadors associats a taules per tal de generar el repositori estadístic.
4. dades inicials: On es troben les dades inicials de prova per comprovar el correcte funcionament de la base de dades
5. proves: On es troben tots els scripts per testejar el sistema de alta, modificació i baixa o sistema ABM.

1.6 Breu descripció dels altres capítols de la memòria

Al capítol 2.1 es pot fer el seguiment del disseny de la aplicació on es desenvolupa el diagrama UML junt amb el disseny lògic per comprovar les restriccions i checks necessaris.

Al capítol 2.2 s'expliquen els diferents tipus que s'han fet servir, aprofitant que Postgresql ofereix la possibilitat evitem crear taules per albergar dades com l'estat, o els diferents nivells d'estudis.

Al capítol 2.3 es defineixen les taules i el seu sentit dintre de la aplicació.

Al capítol 2.4 es poden trobar les funcions i disparadors que s'han dissenyat per tal de donar solució al sistema ABM de la aplicació.

Al capítol 2.5 es defineix cada una de les funcions que donen solució a les estadístiques del sistema.

Al capítol 2.6 s'explica el funcionament del log de dades.

Al capítol 2.7 es pot consultar les proves fetes al sistema per tal de poder verificar el correcte funcionament del sistema ABM de la aplicació.

2. Desenvolupament

2.1 Model UML

Per tal d'arribar a la confecció de les taules de la base de dades cal inicialment fer un model en llenguatge UML així com un disseny lògic, S'ha fet servir Magic Draw per tal de realitzar el model conceptual de les taules obtenint el següent diagrama UML.

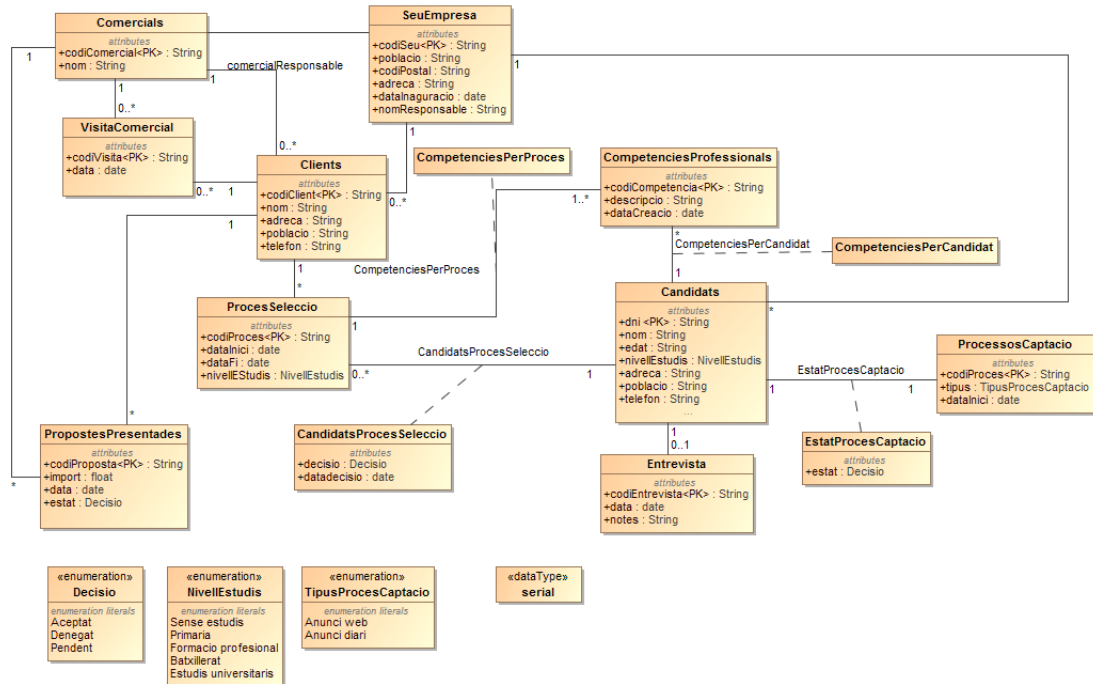


Figura 3. Diagrama UML

El diagrama es pot dividir en 4 seccions, més els tipus enumeratius que tenen el seu propi apartat:

- Secció on es descriu les seus de l'empresa:

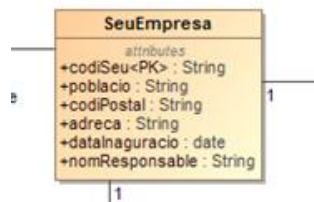


Figura 4. Secció seus empresa.

Aquesta secció està composta per la taula seuempresa on es guardarà les dades referents a les diferents seus que componen la empresa de reclutament de personal RecruitCAT. La taula seuempresa es relaciona amb els clients, els candidats i els comercials. Totes les relacions són 1:N donat que els clients, els comercials i els candidats pertanyen a una sola seu i no pas a una altra d'aquesta forma s'indica aquest fet.

- Secció on es descriu totes les dades relacionades amb els candidats:

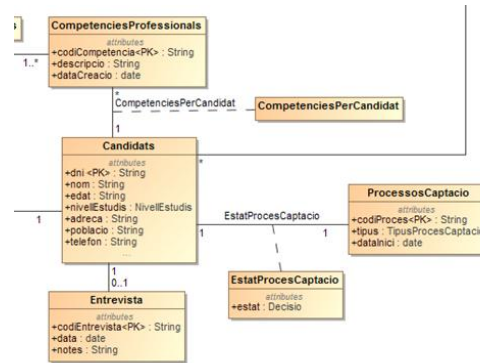


Figura 5. Secció candidats.

Aquesta secció recull totes les dades que envolten a la entitat Candidats, així com la entrevista que es correspon al primer contacte que té RecruitCAT amb el candidat, el procés de captació que guarda les dades referents a com el candidat s'ha posat en contacte amb la empresa, i per últim les seves capacitats i habilitats com a professional indicat per la taula competenciesprofessionals, on quedarà relacionada després amb els processos de selecció que els clients demanaran a la empresa.

Un candidat a més de les dades pròpies que el defineixen com el DNI, el nom, la edat, etc..., també el defineixen, per una banda les competències professionals que té. Cada competència s'emmagatzema en una taula i aquesta es relaciona amb el Candidat mitjançant la taula competenciespercandidat, es crea una taula i no un camp donat que un mateix usuari pot tindre diferents competències professionals i donat que el nombre de competències és indefinit (*) s'ha de guardar en una taula diferent.

Per altra banda també el defineix quin procés de captació és el que s'ha fet servir per captar al candidat i donat que diferents candidats poden tenir el mateix procés de captació es crea una taula per tal de guardar l'estat del procés de captació ja que es pot rebutjar el candidat per no complir una sèrie de requisits i aquest fet es guardarà a la taula EstatProcesCaptacio. La relació 1:1 marca aquest fet que un candidat tindrà el seu procés de captació, però el mateix procés de captació tindrà diferents resultats en funció del candidat.

Finalment es guardarà la entrevista en una taula diferent donat que cada usuari tindrà la seva entrevista i per motius de claredat del codi es crea una taula. Es podria haver creat els camps de l'entrevista a la taula candidat degut que un usuari solament tindrà una entrevista, però d'aquesta forma si cal modificar les especificacions i decidir que un usuari ha de passar varies entrevistes ja són preparades les taules per aquest futur canvi.

- Secció on es descriu totes les dades relacionades amb els clients:

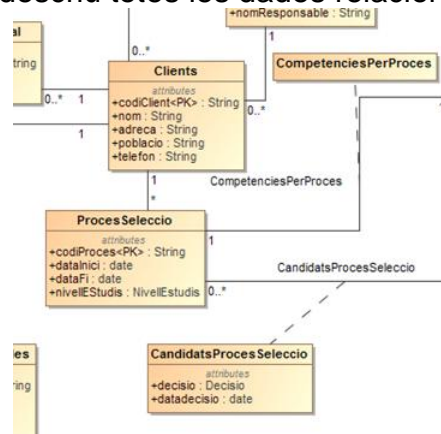


Figura 6. Secció clients.

Aquesta secció recull les dades referents als clients i les seves ofertes, on els candidats han de passar un procés de selecció per tal de que passin a ser acceptats i optin al lloc de treball.

Dels clients es necessitarà amés del CIF, el nom, l'adreça, etc..., els processos de selecció que oferta, aquest procés de selecció necessita relacionar-se amb la taula CompetènciesPerProcés i donat aquest fet existeix la relació amb la taula CompetènciesProfessionals on es guarden les competències professionals. Ja que poden tenir 1 o varies competències professionals es guarden en una taula diferent i no es guarden les competències professionals amb el procés de selecció.

Igualment es guarda en una taula diferent els candidats per cada un dels processos de selecció ja que poden ser 1 o varis els candidats que s'inscriuen. En aquesta taula es guardarà la decisió i la data de decisió per tal de tenir constància que s'ha decidit i quan. El camp datadecisio ha de ser un camp automàtic i dins del procediment emmagatzemat que modificarà la taula es generarà automàticament.

- Secció on es descriuen les dades relacionades amb els comercials:

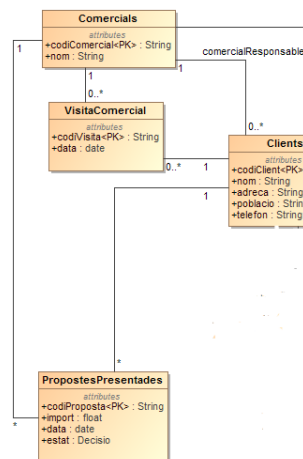


Figura 7. Secció comercials.

Aquest últim apartat recull les dades referents als contactes que té el client amb la empresa com ara les propostes comercials presentades, les visites comercials i els comercials, que són els encarregats de comunicar-se amb els clients.

Els comercials tenen relació amb la taula Clients donat que aquesta necessita especificar quin és el comercial responsable i està referit en un camp dins la taula Clients donat que solament tindrà un comercial responsable.

Igualment es necessita tenir un control de les visites que un comercial fa a un client i donat aquest fet es generarà la taula VisitaComercial. Aquesta relació és pot donar que no existeixin visites comercials o que hi hagi moltes, donat aquest fet la relació és 1-* ja que una visita comercial solament la podrà fer un comercial.

Finalment es necessari guardar les ofertes que l'empresa RecruitCAT fa als seus clients i donat aquest fet es generarà la taula PropostesPresentades. De la mateixa forma que al paràgraf anterior un comercial pot tindre cap o moltes ofertes i donat a quest fet la relació es 1-*.

Al disseny inicial no es tenia en compte el camp codiComercial a la taula PropostesPresentades, es va decidir obtenir-lo de la relació amb la taula client, però per tal de facilitar la obtenció de estadístiques es va afegir com a camp propi de la taula PropostesPresentades.

De la mateixa forma inicialment no es va indicar la relació CompetenciesPerProces donat que no es va detectar com una necessitat, però al realitzar les estadístiques si va aparèixer i així es va fer constar.

Els camps PK (Primary Key) es varen definir com auto incrementals però per tal de tenir un millor control en les insercions es va decidir canviar de tipus serial a tipus varchar(10) fent així que les claus primàries seran de la forma:

- CS000000 per les seus,
- CM000000 per a els comercials,
- el CIF per a els clients,
- VC000000 per les visites comercials,
- PS000000 per als processos de selecció,
- el DNI per als candidats,
- E000000 per les entrevistes,
- CP000000 per les competències professionals,
- PC000000 pels processos de captació i
- PP000000 per les propostes presentades.

Seguidament s'ha realitzat el disseny lògic de la mateixa base de dades obtenint les relacions i les diferents restriccions que més tard s'implementaran al SGBD. Les claus primeres es marquen amb el nom

del camp subratllat, en cas de tenir claus foranes s'indica amb el nom del camp entre { } seguit del nom de la taula on apunta aquesta clau forana.

SeuEmpresa(codiSeu,adreca,poblacio,codiPostal, dataInauguracio,nomResponsable)

Comercials(codiComercial, nom, codiSeu)
{ codiSeu } is foreign key to SeuEmpresa

Clients(codiClient, nom ,adreca, poblacio, telefon,codiSeu, comercialResponsable)
{ codiSeu } is foreign key to SeuEmpresa
{ comercialResponsable } is foreign key to Comercials

VisitaComercial(codiVisita, data, codiComercial, codiClient)
{ codiComercial } is foreign key to Comercials
{ codiClient } is foreign key to Clients

ProcesSeleccio(codiProces, dataInici, dataFi, nivellEstudis, codiClient)
{ codiClient } is foreign key to Clients
Check dataInici<datafi

Candidats(dni, nom, edat, nivellEstudis,adreca,telefon,codiSeu,codiEntrevista)
{ codiSeu } is foreign key to SeuEmpresa
{ codiEntrevista } is foreign key to Entrevista

Entrevista(codiEntrevista, data, notes, codiCandidat)
{ codiCandidat } as foreign key to Candidats

CandidatsProcesSeleccio(codiProces, codiCandidat, decisio)
{codiProces} is foreign key to ProcesSeleccio
{codiCandidat} is foreign key to Candidats

CompetenciesProfesionals(codiCompetencia, descripcio, dataCreacio)

CompetenciesPerCandidat(codiCandidat, codiCompetencia)
{ codiCandidat } is foreign key to Candidats
{codiCompetencia} is foreign key to CompetenciesProfesional

ProcessosCaptacio(codiProces, tipus, dataInici)

EstatProcesCaptacio(codiProces, codiCandidat,estat)
{codiProces} is foreign key to ProcessosCaptacio
{codiCandidat } is foreign key to Candidats

PropostesPresentades(codiProposta, import, data, estat, codiClient,codiComercial)
{ codiClient } is foreign key to Clients
{codiComercial} is foreign key to Comercials

CompetenciesPerProces(codiProces, codiCompetencia)
{ codiProces } is foreign key to ProcesSeleccio
{codiCompetencia} is foreign key to CompetenciesProfesional

S'ha aprofitat la capacitat de revisar mitjançant les clàusules CHECK per tal de fer comprovacions a les dades introduïdes abans de guardar al SGBD.

Com a camps a comprovar amb la clàusules CHECK, s'ha tingut en compte que la data fi d'un procés de selecció no pot ser inferior a la data de inici per tal d'aprofitar els mecanismes de filtratge de errors de la base de dades.

http://csrg.inf.utfsm.cl/~ifuentes/_build/html/lectures/week5/lecture20.html

2.2 Tipus

Els tipus permeten definir a cada camp de la base de dades la longitud en bytes que haurà de reservar el SGBD per tal de satisfer les necessitats de les dades que s'introduiran en un futur en aquest camp. Aquests solen ser character, integer, text, etc...

Per tal de satisfer les necessitat de la base de dades plantejada a l'exercici, es crearan tres tipus per la correcta gestió de les taules els quals son Decisió, NivellEstudis, TipusProcesCaptacio:

Per al tipus Decisió el camp podrà variar entre:

- Acceptat,
- Pendent i
- Denegat.

Per al tipus NivellEstudis es proposen els següent tipus:

- Sense estudis
- Infantil
- Primària
- ESO
- Batxillerat
- Formació professional
- Grau superior
- Estudis universitaris
- Màster
- Doctorat

Per al tipus TipusProcesCaptacio es proposen els següents tipus:

- Anunci radio
- Anunci web

Es podria haver solucionat amb taules addicionals però PostgreSQL ofereix la possibilitat de crear els tipus, de forma que automàticament els camps definits amb aquest tipus donaran error en cas de introduir dades que no corresponguin als tipus donats de alta.

Aquests permeten afegir nous elements a cada tipus una vegada que s'han definit fent servir la instrucció ALTER TYPE Nom Acció [, ...]. Els valors guardats es poden consultar a la taula pg_enum de la base de dades postgres al catàleg PostgreSQL (pg_catalog).

<http://www.postgresql.org/docs/9.5/static/datatype-enum.html>

2.3 Taules

El sistema està compost per 14 taules on es mantindran les dades de cada una de les entitats. En aquest apartat es mostren les taules que tenen relacions entre elles, no es mostren les taules corresponents a las de les estadístiques ni la taula corresponent al log de dades. Aquestes corresponen a les taules [est_a, est_n] i la taula log.

SeuEmpresa: Manté les dades referents a les seus de les empreses del sistema.

- *codiseu:* Clau primària, representa el codi de la seu.
- *poblacio:* Població de on és la seu.
- *codipostal:* Codi postal de la seu
- *adreça:* Adreça de la seu.
- *Datainauguracio:* Data de inauguració de la seu.
- *Nomresponsable:* Nom del responsable de la seu.

Comercials: Manté les dades referents als comercials que efectuaran les visites comercials.

- *codicomercial:* Clau primària, codi del comercial.
- *Nom:* Guarda el nom del comercial.

Clients: Manté les dades referents als clients que rebran les propostes presentades dels comercials.

- *codiclient:* Clau primària, codi del client. Correspon al CIF.
- *nom:* Nom o raó social del client.
- *adreça:* Adreça del client.
- *poblacio:* Població del client.
- *telefon:* Telèfon del client.
- *codiseu:* Codi de la seu a la que pertany el client.
- *comercialresponsable:* Codi del comercial responsable del client.

VisitaComercial: Manté les dades referents a les visites comercials mantingudes amb els clients.

- *codivisita:* Clau primària, codi de la visita.
- *data:* Data de la visita comercial.
- *codicomercial:* Codi del comercial que ha fet la visita.
- *codiclient:* Codi del client que rep la visita.

ProcesSeleccio: Manté les dades referents als diferents processos de selecció que els clients demanen.

- *codiproces:* Clau primària, codi del procés de selecció.
- *datainici:* Data de inici del procés de selecció.
- *datafi:* Data fi del procés de selecció.
- *nivellestudis:* Nivell d'estudis.
- *codiclient:* Codi del client que fa el procés de selecció.

Candidats: Manté les dades dels candidats a nous llocs de treball que es posaran en contacte amb els nostres clients.

- *dni:* Clau primària, identificador de l'usuari.
- *nom:* Nom del candidat.
- *edat:* Edat del candidat
- *nivellestudis:* Nivell d'estudis del candidat
- *adreça:* Adreça del candidat.
- *poblacio:* Població del candidat

- *telefon*: Telèfon de contacte del candidat.
- *codiseu*: Codi de la seu del candidat.

Entrevista: Manté les dades referents a la entrevista interna que s'ha de fer a un candidat.

- *codientrevista*: Clau primària, codi de la entrevista.
- *data*: Data de la entrevista.
- *notes*: Notes que s'han pres a la entrevista
- *codicandidat*: Codi del candidat.

CandidatsProcesSeleccio: Manté la llista de candidats que te els diferents processos de selecció demanat per els clients.

- *codiproces*: Clau primària, codi del procés de selecció.
- *codicandidat*: Codi del candidat.
- *decisio*: Decisió presa pel client.
- *datadecisio*: Data de la decisió.

CompetenciesProfesionals: Manté les diferents competències professionals.

- *codicompetencia*: Clau primària, codi de la competència.
- *descripcio*: Descripció de la competència.
- *datacreacio*: Data de la creació de la competència.

CompetenciesPerCandidat: Manté la llista de les diferents competències que els candidats tenen.

- *codicandidat*: Clau primària junt amb codicomeptència.
- *codicompetencia*: Clau primària junt amb codicandidat.

ProcessosCaptacio: Manté els diferents processos de captació que la empresa ha d'efectuar per obtenir nous candidats.

- *codiproces*: Clau primària, codi del procés.
- *tipus*: Tipus del procés de captació.
- *datainici*: Data de inici del procés de captació.

EstatProcesCaptacio: Guarda les dades de l'estat del procés de captació amb els candidats.

- *codiproces*: Clau primària junt amb codicandidat
- *codicandidat*: Clau primària junt amb codiproces
- *estat*: Estat del procés de captació.

PropostesPresentades: Manté les dades referents a les propostes comercials presentades als clients.

- *codiproposta*: Clau primària, codi de la proposta.
- *import*: Import de la proposta
- *data*: Data de la proposta
- *estat*: Estat de la proposta.
- *codiclient*: Codi del client.
- *codicomercial*: Codi del comercial.

CompetènciesPerProces: Manté la llista de les diferents competències professionals que té un procés de selecció.

- *codiproces:* Clau primària unt amb codicomeptencia.
- *codicomeptencia:* Clau primària unt amb codiproces

http://www.w3schools.com/sql/sql_primarykey.asp

2.4 Funcions i disparadors

Les funcions corresponen a procediments emmagatzemats, aquests executaran les accions necessàries per tal de complir la funcionalitat demanada i pot ser executada manualment o mitjançant un disparador associat a una taula.

Es decideix utilitzar disparadors donat que aquest executarà un procediment emmagatzemat cada vegada que esdevinguin accions com INSERT, UPDATE ó DELETE, d'aquesta forma cada canvi que es produeixi a la taula quedarà emmagatzemat a la estadística.

Per executar un procediment emmagatzemat manualment cal cridar la sentència SELECT i el nom del procediment emmagatzemat (nom_funció()), d'aquesta forma es produiran els canvis i/o s'executaran les accions descrites internament al procediment emmagatzemat.

Per executar un procediment emmagatzemat mitjançant un disparador cal definir que el procediment emmagatzemat retorna com a tipus un trigger indicant a la primera línia:

```
CREATE FUNCTION nom_funció () RETURNS trigger AS ...
```

Els disparadors executaran un procediment emmagatzemat cada vegada que es produeixi un esdeveniment de tipus INSERT, UPDATE i/o DELETE a una taula associada.

Les funcions corresponen al següent esquema:

```
CREATE FUNCTION nom_funció() RETURNS tipus AS $$
DECLARE
  --Variables
BEGIN
  --Executa les funcions necessàries per complir el propòsit

  RETURN valor;
END;
$$ LANGUAGE plpgsql;
```

Els disparadors corresponen al següent esquema:

```
CREATE TRIGGER nom_trigger BEFORE INSERT OR UPDATE
FOR EACH ROW EXECUTE PROCEDURE nom_funció()
```

Les funcions de ABM (Alta, Baixa i Modificació) es creen per tal de unificar la entrada de dades al sistema seguint el patró de Alta_[nom_taula], Baixa_[nom_taula], Modificacio_[nom_taula]. Totes les funcions retornen un camp RSP amb el resultat de la funció, en cas de que la funció hagi finalitzat correctament retorna el valor OK, en cas contrari retorna el error o excepció.

Els procediments emmagatzemats s'estructuren en general seguint el mateix patró:

- Inicialment es comprova si existeixen les dades a la base de dades per tal de no introduir dades duplicades,
- si es troben dades duplicades es mostrarà un error al camp RSP
- si no, es procedeix a afegir o modificar les dades al sistema.
- En qualsevol cas es guardarà el resultat al log de dades.
- Si es produeix un error o una excepció es cancel·la el procés i es guardarà l'error a la taula log.

En cas de funcions de eliminació (baixa_[nom_funció]) es comprova si existeixen les dades a esborrar, en cas de no existir es mostra un error al camp RSP i es guarda aquest junt amb les dades de entrada al sistema de log, per tal de indicar que s'ha produït un intent d'esborrar dades que no existeixen.

Com exemple i per tal de veure amb detall la estructuració interna de una de les funcions de ABM, es desgranarà el procediment emmagatzemat que s'encarrega de donar de alta les seus de l'empresa. Aquest es pot trobar als lliurables entregats conjuntament amb aquesta memòria:

```
create or replace function alta_seuempresa
```

La funció es diu alta_seuempresa([camps]), donat que és la funció que donarà de *alta* a la taula *seuempresa*,

```
(abm_codiseu varchar(10),  
abm_adreca varchar(100),  
abm_poblacio varchar(50),  
abm_codipostal varchar(5),  
abm_datainauguracio date,  
abm_nomresponsable varchar(50),  
OUT RSP text)
```

Els camps de la funció corresponen a cada un dels camps d'entrada de la taula es nombren amb el prefix *abm_* per distingir el camp de la taula del camp de la funció i son:

- abm_codiseu: Camp codiseu de la taula.
- abm_adreca: Camp adreca de la taula.
- abm_poblacio: Camp poblacio de la taula.
- abm_codipostal: Camp codipostal de la taula.
- abm_datainauguracio: Camp datainauguracio de la taula.

- abm_nomresponsable: Camp nomresponsable de la taula.
- OUT RSP: Camp de sortida per al log de dades.

returns TEXT as

S'indica que el tipus de retorn és text donat que existeix un camp de tipus OUT i aquest contindrà text de sortida.

```
declare id varchar(10);
```

Declarem la variable *id* on després guardarem el resultat de una consulta.

```
begin
```

Inicia les accions que esdevindran al executar el procediment emmagatzemat

```
--Es comprova la duplicitat de les dades introduïdes
select codiseu into id from uoc_recrutcat.seuempresa
where upper(adreca) = upper(abm_adreca)
and upper(poblacio) = upper(abm_poblacio)
and upper(codipostal) = upper(abm_codipostal);
```

Es selecciona de la taula seuempresa el codi de la seu i es guarda en la variable que s'ha declarat *id* per comprovar si existeixen dades duplicades.

```
-- Si es troben dades duplicades es retorna un error
if found then
    RSP = 'ERROR LA SEU AMB ADREÇA '||abm_adreca||' '||abm_codipostal||' -
' ||abm_poblacio||' JA ESTA REGISTRAT AMB EL ID '|| id;
```

Si es troben dades (if found then) vol dir que existeixen les mateixes dades amb un altre *id*, es guarda un error indicant el fet al camp RSP.

```
--Si NO, s'insereixen les dades a la base de dades
else
    INSERT INTO uoc_recrutcat.seuempresa
    VALUES (abm_codiseu ,abm_adreca, abm_poblacio, abm_codipostal,
abm_datainaguracio, abm_nomresponsable);
    RSP = 'OK';
end if;
```

Si no es troben dades es procedeix a guardar les dades introduïdes als camps del procediment emmagatzemat en una nova fila i es guarda el valor OK al camp RSP.

```
-- S'insereix el resultat de la consulta en qualsevol cas al log.
INSERT INTO uoc_recrutcat.log
VALUES('alta_seuempresa', abm_codiseu ||', '||abm_adreca||', '||
abm_poblacio||', '|| abm_codipostal||', '|| abm_datainaguracio||', '||
abm_nomresponsable ,RSP);
```

En qualsevol dels casos es guarda al log el resultat de la consulta, indicant el nom del procediment, els camps d'entrada i el resultat de la consulta mitjançant el camp RSP.

```

-- Catch errors
EXCEPTION
  WHEN OTHERS THEN
    --Es captura l'error de l'SQL al camp RSP.
    RSP = 'ERROR '||SQLERRM;
    --Es guarda al log.
    INSERT INTO uoc_recrUITcat.log
    VALUES('alta_seuempresa', abm_codiseu ||', '||abm_adreca||', '||
    abm_poblacio||', '|| abm_codipostal||', '|| abm_datainaguracio||', '||
    abm_nomresponsable , 'ERROR '||RSP);

```

En tots els procediments emmagatzemats del sistema ABM s'inclou el bloc de detecció d'errors d'entrada de dades on es captura qualsevol error mitjançant:

```

EXCEPTION
  WHEN OTHERS THEN

```

S'obté el error causat al servidor amb la sentència:

```
SQLERRM
```

Y es guarda al log mitjançant:

```

--Es guarda al log.
INSERT INTO uoc_recrUITcat.log
VALUES('alta_seuempresa', abm_codiseu ||', '||abm_adreca||', '||
abm_poblacio||', '|| abm_codipostal||', '|| abm_datainaguracio||', '||
abm_nomresponsable , 'ERROR '||RSP);

```

Finalment es tanca el procediment emmagatzemat mitjançant `end $$` i indicant el tipus de llenguatge `language plpgsql`;

Totes les funcions comproven els camps nuls, les claus foranies, la unicitat de les claus úniques, el llarg de les dades introduïdes, etc... De la mateixa forma es comproven que no s'insereixin dades duplicades al sistema retornant un error en aquest cas.

Les funcions del sistema de estadístiques es nombren seguint el patró `fn_[acció]_est_[lletra estadística]_[nom taula afectada]()`. Els disparadors es nombren seguint el patró `trg_[acció]_est_[lletra estadística]_[nom taula afectada]()`. D'aquesta forma es facilita el seguiment en cas de mal funcionament de les funcions o disparadors.

<http://www.postgresql.org/docs/current/static/plpgsql.html>

<http://www.postgresql.org/docs/9.1/static/sql-createtrigger.html>

2.5 Estadístiques

Les estadístiques es guardaran en taules amb els resultats esperats a taules anomenades de la forma est_[lletra estadística] i en el cas de necessitar taules temporals per la mateixa estadística est_[lletra estadística]_temp. Aquestes taules no tenen relació entre elles, i no es mostren al document UML. Per tal d'obtenir les dades s'haurà de cridar la sentència SELECT * FROM est_[lletra estadística].

De forma general cada estadística està estructurada de la següent forma:

- S'obtenen les dades de la estadística mitjançant una consulta.
- Es comprova si existeixen dades a la taula de la estadística.
- Es guarden les noves dades.

Per tal de guardar les dades i poder obtenir resultats en temps constant es necessari:

- Crear una taula per guardar les dades de la estadística (est_[lletra_estadística]).
- Crear el procediment emmagatzemat que executarà les accions necessàries per guardar les dades a la taula.
- Crear un disparador (trigger) associat a una taula per executar el procediment emmagatzemat.

Per tal de veure un exemple d'estadística i tots els passos necessaris es desgranarà la estadística i:

```
--Es crea la taula de la estadística I
CREATE TABLE uoc_recrutcat.est_i
(
  codicclient character(10)
);
```

Es crea la taula per tal de guardar les dades de la estadística.

```
--Es crea la funció que executarà el disparador
trg_update_est_i_candidatsprocesseleccio
CREATE OR REPLACE FUNCTION fn_update_est_i_candidatsprocesseleccio() RETURNS
TRIGGER AS $insertar$
```

Es crea la funció que actualitzarà les dades des de la taula i s'indica que l'executarà un disparador (trigger) candidatsprocesseleccio

```
DECLARE
fn_codi_client varchar(10);
fn_codi_client_temp varchar(10);
```

Es declaren la variable per tal de obtenir les dades de la consulta que calcula la estadística fn_codi_client i la variable que guardarà el valor previ de la taula est_i per tal de comprovar que existeixen dades o no prèvies.

```
BEGIN
```

Comença el càlcul de la estadística.

```
--Es calculen les dades del client que més candidats ens ha acceptat
SELECT
    processeleccio.codiclient INTO fn_codi_client
FROM
    uoc_recrutcat.candidatsprocesseleccio,
    uoc_recrutcat.processeleccio
WHERE
    processeleccio.codiproces = candidatsprocesseleccio.codiproces
    AND candidatsprocesseleccio.decisio = 'Acceptat'
GROUP BY processeleccio.codiclient
ORDER BY COUNT(processeleccio.codiclient) DESC
LIMIT 1;
```

Aquesta sentència calcula en codi del client que més candidats ha acceptat i el guarda a la variable `fn_codi_client`

```
--Es comprova si existeixen dades a la taula est_i
SELECT codiclient INTO fn_codi_client_temp FROM uoc_recrutcat.est_i;
```

Es comproven si existeixen dades a la taula guardant a la variable `fn_codi_client_temp` el possible codi de client de la estadística.

```
IF FOUND THEN
    --Si existeixen s'actualitzen
    UPDATE uoc_recrutcat.est_i SET codiclient = fn_codi_client;
ELSE
    --Si NO es creen de noves
    INSERT INTO uoc_recrutcat.est_i VALUES (fn_codi_client);
END IF;
```

Si existeixen dades s'actualitza amb les dades obtingudes a la consulta inicial amb les dades guardades a la variable `fn_codi_client`, si no es guarda amb les noves dades a la taula `est_i`

```
RETURN NULL;

END;
$insertar$ LANGUAGE plpgsql;
```

Es finalitza el procediment emmagatzemat.

```
--Es crea el disparador que executarà cada canvi de la taula
candidatsprocesseleccio la funció fn_update_est_i_candidatsprocesseleccio();
CREATE TRIGGER trg_update_est_i_candidatsprocesseleccio
```

Seguidament es procedeix a crear el disparador que executarà el procediment emmagatzemat.

```
AFTER INSERT OR UPDATE
```

S'indica que després de cada INSERT o UPDATE

```
ON uoc_recrutcat.candidatsprocesseleccio
```

de la taula `candidatsprocesseleccio`

FOR EACH ROW

Per cada línia de entrada a la taula

```
EXECUTE PROCEDURE fn_update_est_i_candidatsprocesseleccio();
```

s'executarà el procediment emmagatzemat
fn_update_est_i_candidatsprocesseleccio()

D'aquesta forma es finalitza els passos necessaris per crear una estadística on a mode de resum es necessari realitzar tres accions: Crear la taula, crear el procediment emmagatzemat i el disparador que executarà el procediment

Es decideix facilitar la identificació de les estadístiques mitjançant el sistema de lletres com apartats de l'enunciat.

2.5.1 Estadística a (est_a)

Aquesta retornarà donat un any qualsevol, número d'entrevistes realitzades internament.

Per una banda es crearà un disparador per tal d'obtenir cada vegada que es inserida una nova entrevista a la taula entrevista, aquest si ja existeix executa una funció que actualitza les dades o crea una nova entrada per l'any de la entrevista.

Per altra banda s'ha creat un disparador que cada vegada que s'esborra una entrada a la taula entrevista resta en una unitat la quantitat de entrevistes.

Cal destacar l'ús de la funció EXTRACT(YEAR FROM data) per tal de obtenir l'any d'un camp on es guarda una data en format yyyy-mm-dd.

<http://www.postgresql.org/docs/9.5/static/functions-datetime.html>

2.5.2 Estadística b (est_b)

Donat un any concret percentatge de processos de selecció que han acabat amb resultat positiu.

Inicialment es va crear un disparador per tal de obtenir la quantitat de processos de selecció per cada inserció a la taula processeleccio. Es va crear un altre disparador per la taula candidatsprocesseleccio per tal de obtenir els resultats positius, el resultat s'obtenia de la divisió dels dos disparadors, però es produïa un resultat erroni donat que no es calculava cada vegada el total.

Actualment s'ha creat un procediment emmagatzemat que calcula constantment el resultat desitjat i un disparador per la taula candidatsprocesseleccio que executa aquesta funció.

Cal destacar l'ús de la funció CAST(integer AS FLOAT) per tal de obtenir

valors decimals a les divisions si no s'obtenen resultats enters en les divisions fetes amb valors enters.

<http://gustavopeiretti.com/postgresql-division-de-enteros-incluyendo-cociente-con-decimales/>

2.5.3 Estadística c (est_c)

Percentatge de candidats registrats al sistema que en el moment de fer la consulta no han estat seleccionats en cap procés de selecció.

Per una banda, mitjançant un disparador per cada nou candidat inserit a la taula candidats es consulta la quantitat de candidats.

Per altra banda es consulta de la taula candidatsprocesseleccio la quantitat de candidats acceptats, s'obté la diferència entre candidats i candidats acceptats i d'aquesta forma la quantitat de no acceptats, es fa la divisió i s'obté el resultat desitjat.

Cal destacar l'ús dels camps NEW.nomCamp i OLD.nomCamp per obtenir dades de les noves insercions o les dades existents a la base de dades abans de fer canvis.

<http://www.postgresql.org/docs/current/static/plpgsql.html>

2.5.4 Estadística d (est_d)

Donat un any concret, import total de les propostes presentades i que han estat acceptades per algun dels nostres clients.

Inicialment es va proposar que per cada proposta acceptada es guarda el import sumant al total per l'any de la proposta, però el resultat no era el correcte i s'ha canviat la funció per una altra que fa la suma total cada vegada que es produeix un canvi a la taula propostespresentades.

2.5.5 Estadística e (est_e, est_e_temp)

Comercial que, en el moment de fer la consulta, té un percentatge d'èxit millor. Considerem percentatge d'èxit al ràtio propostes acceptades entre propostes presentades.

Per tal de donar solució a la estadística e, ha estat necessari utilitzar una taula temporal. D'aquesta forma es guarda en aquesta taula temporal mitjançant dos disparadors un per la taula propostespresentades per cada INSERT i un altre per la mateixa taula per cada INSERT o UPDATE en cas que estigui Acceptada. Cada vegada que es produeix un canvi es consulta quin és el millor rati i s'actualitza la taula est_e.

2.5.6 Estadística f (est_f)

Quin és el millor procés de captació de candidats. Considerem com a millor el que ens hagi portat un major número de candidats acceptats en processos de selecció.

Es comprovarà cada INSERT o UPDATE de la taula candidatsprocesseleccio quin és el procés de selecció que té els candidats seleccionats com Acceptats de la taula. Fent un JOIN amb la

taula estatprocescaptació obtenim el resultat desitjat actualitzant d'aquesta forma amb les dades obtingudes la taula est_f.

2.5.7 Estadística g (est_g)

Top 10 dels nivells d'estudis dels candidats més demanats en els processos de selecció en un determinat any.

S'obtenen els nivells d'estudi de la taula processeleccio i mitjançant un loop es guarden a la taula est_g cada una de les dades.

<http://www.postgresql.org/docs/current/static/plpgsql-control-structures.html#PLPGSQL-RECORDS-ITERATING>

2.5.8 Estadística h (est_h)

Top 5 de les competències professionals més sol·licitades en un determinat any.

Es comprovarà cada INSERT o UPDATE la quantitat de competències professionals, actualitzant la taula est_h amb les dades de l'any de la introducció de les dades.

2.5.9 Estadística i (est_i)

Client que fins al moment actual i tenint en compte totes les dades enregistrades, que més candidats ens ha acceptat.

Es comprovarà cada INSERT o UPDATE de la taula candidatsprocesseleccio mitjançant un disparador, fent un JOIN amb la taula processeleccio i obtenint quin és el client que més Acceptats té contant els Acceptats i ordenant per aquest camp de forma descendent.

2.5.10 Estadística j (est_j)

Client que, en un any determinat, més propostes presentades ens ha denegat.

Per tal d'obtenir la estadística j es crea un disparador per cada INSERT o UPDATE de la taula propostespresentades, es comprova per l'any de la proposta introduïda quin és el client que més propostes ha denegat i es guarda a la taula j.

2.5.11 Estadística k (est_k)

Percentatge de candidats que, en el moment actual, mai han estat seleccionats per un client.

Per tal de obtenir la estadística es crea un disparador per la taula candidatsprocesseleccio, es divideix el nombre de candidats que no estan a la taula candidatsprocesseleccio entre els que estan a la taula de candidats.

2.5.12 Estadística l (est_l)

Comercial que més visites ha realitzat en el any en curs.

Per tal d'obtenir la estadística es crea un disparador per la taula visitacomercial per cada INSERT o UPDATE per tal de que executi la

funció que calcula quin és el comercial que més visites a realitzat s'obté el COUNT de visites de l'any de la data actual.

2.5.13 Estadística m (est_m)

Procés de captació de candidats que, en l'any en curs, ens ha proporcionat més candidats seleccionats per algunes de les empreses clients enregistrades.

Per tal d'obtenir la estadística es crea un disparador per cada INSERT o UPDATE de la taula candidatsprocesseleccio i s'executa la funció que relaciona per DNI de candidats les taules candidats, estatprocescaptacio i candidatsprocesseleccio i s'obté el codiproces de captació dels candidats en funció de la taula candidatsprocesseleccio, es filtra per l'any en curs i per els que estan acceptats seguidament es guarda el resultat a la taula est_m

2.5.14 Estadística n (est_n)

Seu de l'empresa que , en l'any en curs, ha realitzat menys entrevistes a candidats.

Per tal de obtenir la estadística es crea un disparador a la taula entrevista. Es creuen les taules candidats i entrevista per tal de obtenir quina seu té relacionat amb el codi de entrevista obtenint així la data i filtrar per l'any en curs. Una vegada obtingut les dades es guarden a la taula est_n.

2.6 Log

El log guarda les accions fetes per cada una de les funcions ABM, així com els camps d'entrada i el resultat del camp RSP dins de la taula log. Tots els procediments emmagatzemats del sistema ABM tenen un INSERT a la taula log dins de la part on s'implementa les accions amb la base de dades, en cas de que hagi anat tot correcte i un altre en cas d'excepció.

A tall d'exemple es pot consultar la funció alta_seuempresa() a la secció del log:

En qualsevol dels casos es guarda al log el resultat de la consulta, indicant el nom del procediment, els camps d'entrada i el resultat de la consulta mitjançant el camp RSP.

```
-- Catch errors
EXCEPTION
  WHEN OTHERS THEN
    --Es captura l'error de l'SQL al camp RSP.
    RSP = 'ERROR '||SQLERRM;
    --Es guarda al log.
    INSERT INTO uoc_recrUITcat.log
    VALUES('alta_seuempresa', abm_codiseu ||', '||abm_adreca||', '||
    abm_poblacio||', '|| abm_codipostal||', '|| abm_datainaguracio||', '||
    abm_nomresponsable , 'ERROR '||RSP);
```


En tots els procediments emmagatzemats del sistema ABM s'inclou el bloc de detecció d'errors d'entrada de dades on es captura qualsevol error mitjançant:

```
EXCEPTION  
    WHEN OTHERS THEN
```

S'obté el error causat al servidor amb la sentència:

```
SQLERRM
```

Y es guarda al log mitjançant:

```
--Es guarda al log.  
INSERT INTO uoc_recrutcat.log  
VALUES('alta_seuempresa', abm_codiseu ||', '||abm_adreca||', '||  
abm_poblacio||', '|| abm_codipostal||', '|| abm_datainaguracio||', '||  
abm_nomresponsable , 'ERROR '||RSP);
```

Inicialment es va optar per fer servir la funció `current_query()` per guardar les dades referents al nom de la funció i els camps de entrada, però quan es fa una entrada de dades massiva guarda com `current query` la primera funció executada per cada una de les entrades del log en lloc de guardar cada entrada corresponent a la acció realitzada, no permetent així veure quina funció i quins camps s'executen.

En front a aquest contratemps s'ha implementat individualment cada `INSERT` a la taula log en lloc de fer servir `INSERT INTO LOG (current_query(), RSP)`; Aquest sistema hauria simplificat el procés de creació del log però no ha estat possible.

Igualment s'ha comprovat que al introduir un camp `NULL` a la funció, en el moment de guardar les dades al log no es guarden el contingut dels camps de entrada, s'espera que en futures versions de Postgresql quedi solucionat.

<http://forums.enterprisedb.com/posts/list/3515.page>

2.7 Proves

Es realitzaran les proves necessàries per tal de verificar que cada una de les funcions de alta, baixa i modificació funcionen correctament. D'aquesta forma es controlarà el correcte funcionament d'aquest sistema d'entrada de dades.

Es pretén verificar que la entrada de dades mitjançant el sistema ABM no introduirà inconsistències a la base de dades, per tal de portar a terme el conjunt de proves es comprovarà la introducció de camps nuls en cada una de les funcions i camps del sistema ABM.

ABM de la taula seuempresa		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció	Es guarda un error al log indicant	OK

alta_seuempresa amb el identificador NULL	que es viola la restricció de NOT NULL	
S'executa la funció alta_seuempresa amb la adreça NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_seuempresa amb la població NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_seuempresa amb el codi postal NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_seuempresa amb la data de inauguració NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_seuempresa amb el nom del responsable NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_seuempresa	Es guarden les dades correctament	OK
S'executa la funció modificacio_seuempresa amb el identificador NULL	Es mostra un error no existeix la relació	ERROR
S'executa la funció modificacio_seuempresa amb la adreça NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_seuempresa amb la població NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_seuempresa amb el codi postal NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_seuempresa amb la data de inauguració NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_seuempresa amb el nom del responsable NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_seuempresa	Es modifiquen les dades correctament	OK
S'executa la funció baixa_seuempresa amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	OK
S'executa la funció baixa_seuempresa	S'esborren les dades correctament	OK

ABM de la taula comercials		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_comercials amb el identificador NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_comercials amb el nom NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_comercials amb la codi seu NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció	Es guarden les dades	OK

alta_comercials	correctament	
S'executa la funció modificacio_comercials amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació.	ERROR
S'executa la funció modificacio_comercials amb el nom NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_comercials amb la codi seu NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_comercials	Es guarden les dades correctament	OK
S'executa la funció baixa_comercials amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_comercials	S'esborren les dades correctament	OK

ABM de la taula clients		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_clients amb el identificador NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_clients amb el nom NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_clients amb la adreça NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_clients amb la població NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_clients amb el telèfon NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_clients amb el codi seu NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_clients amb el codi comercial NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_clients amb un codi seu que no existeix	Es guarda un error al log indicant que es viola la restricció de clau forana.	OK
S'executa la funció alta_clients amb un codi de comercial que no existeix	Es guarda un error al log indicant que es viola la restricció de clau forana.	OK
S'executa la funció alta_clients	Es guarden les dades correctament	OK
S'executa la funció modificacio_clients amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_clients amb el nom NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_clients amb la adreça NULL	Es guarda un error al log indicant que es viola la restricció de NOT	OK

	NULL	
S'executa la funció modificacio_clients amb la població NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_clients amb el telèfon NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_clients amb el codi seu NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_clients amb el codi comercial NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_clients	Es modifiquen les dades correctament	OK
S'executa la funció baixa_clients amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_clients	S'esborren les dades correctament	OK

ABM de la taula visitacomercial		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_visitacomercial amb el identificador NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_visitacomercial amb la data NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_visitacomercial amb la codi comercial NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_visitacomercial amb el codi client NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_visitacomercial	Es guarden les dades correctament	OK
S'executa la funció modificacio_visitacomercial amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_visitacomercial amb la data NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_visitacomercial amb la codi comercial NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_visitacomercial amb el codi client NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_visitacomercial	Es modifiquen les dades correctament	OK
S'executa la funció baixa_visitacomercial amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_visitacomercial	S'esborren les dades correctament	OK

ABM de la taula processeleccio		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_processeleccio amb el identificador NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_processeleccio amb la data de inici NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_processeleccio amb la data fi NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_processeleccio amb el nivell estudis NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_processeleccio amb el codi client NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_processeleccio amb un codi client que no existeix	Es guarda un error al log indicant que es viola la restricció de clau forana	OK
S'executa la funció alta_processeleccio amb la data fi < que la data inici	Es guarda un error al log indicant que es viola la restricció check per les dates de inici i fi	OK
S'executa la funció alta_processeleccio	Es guarden les dades correctament	OK
S'executa la funció modificacio_processeleccio amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_processeleccio amb la data de inici NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_processeleccio amb la data fi NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_processeleccio amb el nivell estudis NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_processeleccio amb el codi client NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_processeleccio amb un codi client que no existeix	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_processeleccio	Es modifiquen les dades correctament	OK
S'executa la funció baixa_processeleccio amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_processeleccio	S'esborren les dades correctament	OK

ABM de la taula candidats		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_candidats amb el identificador NULL	Es guarda un error al log indicant que es viola la restricció de NOT	OK

	NULL	
S'executa la funció alta_candidats amb el nom NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidats amb la edat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidats amb el nivell de estudis NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidats amb la adreça NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidats amb la població NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidats amb el telèfon NULL	Es guarden correctament les dades	OK
S'executa la funció alta_candidats amb el codi seu NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidats amb el codi seu inexistent	Es guarda un error al log indicant que es viola la restricció de clau forana	OK
S'executa la funció alta_candidats	Es guarden correctament les dades	OK
S'executa la funció modificacio_candidats amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_candidats amb el nom NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_candidats amb la edat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_candidats amb el nivell de estudis NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_candidats amb la adreça NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_candidats amb la població NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_candidats amb el telèfon NULL	Es modifiquen les dades correctament	OK
S'executa la funció modificacio_candidats amb el codi seu NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_candidats	Es modifiquen les dades correctament	OK
S'executa la funció baixa_candidats amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_candidats	S'esborren les dades correctament	OK

ABM de la taula entrevista		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_entrevista amb el identificador NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_entrevista amb la data NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_entrevista amb les notes NULL	Es guarden les dades correctament	OK
S'executa la funció alta_entrevista amb el codi candidat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_entrevista amb un codi candidat inexistent	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_entrevista	Es guarden les dades correctament	OK
S'executa la funció modificacio_entrevista amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio _entrevista amb la data NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio _entrevista amb les notes NULL	Es modifiquen les dades correctament	OK
S'executa la funció modificacio _entrevista amb el codi candidat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció baixa_entrevista amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_entrevista	S'eliminen les dades correctament	OK

ABM de la taula candidatsprocesseleccio		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_candidatsprocesseleccio amb el identificador NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidatsprocesseleccio amb el candidat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidatsprocesseleccio amb el candidat inexistent	Es guarda un error al log indicant que es viola la restricció de clau forana	OK
S'executa la funció alta_candidatsprocesseleccio amb l'estat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_candidatsprocesseleccio	Es guarden les dades correctament	OK
S'executa la funció modificacio_candidatsprocesseleccio amb el identificador NULL	Es guarda un error al log indicant que no existeix la entrada sollicitada	OK
S'executa la funció modificacio _candidatsprocesseleccio amb el	Es guarda un error al log indicant que no existeix la	OK

candidat NULL	entrada sol·licitada	
S'executa la funció modificacio_candidatsprocessseleccio amb l'estat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_candidatsprocessseleccio	Es modifiquen les dades correctament	OK
S'executa la funció baixa_candidatsprocessseleccio amb el codi proces NULL	Es guarda un error al log indicant que no existeix la entrada sol·licitada	OK
S'executa la funció baixa_candidatsprocessseleccio amb el candidat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció baixa_candidatsprocessseleccio	Es modifiquen les dades correctament	OK

ABM de la taula competenciesprofesionals		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_competenciesprofesionals amb el identificador NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_competenciesprofesionals amb la descripció NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_competenciesprofesionals	Es guarden les dades correctament	OK
S'executa la funció modificacio_competenciesprofesionals amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_competenciesprofesionals amb la descripció NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_competenciesprofesionals	Es modifiquen les dades correctament	OK
S'executa la funció baixa_competenciesprofesionals amb el identificador NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_competenciesprofesionals	S'esborren les dades correctament	OK

ABM de la taula competenciespercandidat		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_competenciespercandidat amb el codi candidat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_competenciespercandidat amb el codi competencia NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_competenciespercandidat	Es guarden les dades correctament	OK
S'executa la funció modificacio_competenciespercandidat amb el codi candidat NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_competenciespercandidat amb el codi competencia NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	ERROR

S'executa la funció modificacio_competenciespercandidat	Es modifiquen les dades correctament	OK
S'executa la funció baixa_competenciespercandidat amb el codi candidat NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_competenciespercandidat	S'esborren les dades correctament	OK

ABM de la taula processoscaptacio		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_processoscaptacio amb el codi procés captació NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_processoscaptacio amb el codi candidat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_processoscaptacio amb l'estat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_processoscaptacio	Es guarden les dades correctament	OK
S'executa la funció modificacio_processoscaptacio amb el codi procés captació NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_processoscaptacio amb el codi candidat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_processoscaptacio amb l'estat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_processoscaptacio	Es modifiquen les dades correctament	OK
S'executa la funció baixa_processoscaptacio amb el codi procés captació NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_processoscaptacio	S'esborren les dades correctament	OK

ABM de la taula estatprocescaptacio		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_estatprocescaptacio amb el codi procés captació NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_estatprocescaptacio amb el codi candidat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_estatprocescaptacio amb l'estat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_estatprocescaptacio	Es guarden les dades correctament	OK
S'executa la funció modificacio_estatprocescaptacio amb el codi procés captació NULL	Es guarda un error al log indicant que no existeix la relació	ERROR

S'executa la funció modificacio_estatprocescaptacio amb el codi candidat NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_estatprocescaptacio amb l'estat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_estatprocescaptacio	Es modifiquen les dades correctament	OK
S'executa la funció baixa_estatprocescaptacio amb el codi procés captació NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_estatprocescaptacio amb el codi candidat NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_estatprocescaptacio	S'esborren les dades correctament	OK

ABM de la taula propostespresentades		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_propostespresentades amb el codi de proposta NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_propostespresentades amb el preu NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_propostespresentades amb la data NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_propostespresentades amb l'estat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_propostespresentades amb el codi client NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_propostespresentades amb el codi de comercial NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_propostespresentades	Es guarden les dades correctament	OK
S'executa la funció modificacio_propostespresentades amb el codi de proposta NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_propostespresentades amb el preu NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_propostespresentades amb la data NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_propostespresentades amb l'estat NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_propostespresentades amb el codi client NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_propostespresentades	Es guarda un error al log indicant	OK

_propostespresentades amb el codi de comercial NULL	que es viola la restricció de NOT NULL	
S'executa la funció modificacio _propostespresentades	Es modifiquen les dades correctament	OK
S'executa la funció baixa _propostespresentades amb el codi de proposta NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa _propostespresentades	S'esborren les dades correctament	OK

ABM de la taula competenciesperproces		
Prova a realitzar	Resultat desitjat	Estat
S'executa la funció alta_competenciesperproces amb el codi procés selecció NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_competenciesperproces amb el codi competència professional NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció alta_competenciesperproces	Es guarden les dades correctament	OK
S'executa la funció modificacio_competenciesperproces amb el codi procés selecció NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_competenciesperproces amb el codi competència professional NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció modificacio_competenciesperproces amb el nou codi competència professional NULL	Es guarda un error al log indicant que es viola la restricció de NOT NULL	OK
S'executa la funció modificacio_competenciesperproces	Es modifiquen les dades correctament	OK
S'executa la funció baixa_competenciesperproces amb el codi procés selecció NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_competenciesperproces amb el codi competència professional NULL	Es guarda un error al log indicant que no existeix la relació	ERROR
S'executa la funció baixa_competenciesperproces	S'eliminen les dades correctament	OK

S'han fet 183 proves al sistema de ABM dels quals 30 han donat error així les proves donen un resultat del 83.61% correcte.

El principal error es dona al introduir un valor NULL a la funció de modificació dins del camp identificatiu. D'aquesta forma Postgresql no comprova que el camp que s'està introduint a la clàusula WHERE és NULL i que aquest no pot ser NULL segons com s'especifica al script per crear la taula.

3. Conclusions

Amb el desenvolupament del projecte he après a posar en pràctica les diferents fases d'un projecte, des de el disseny fins a la implementació. Així com a sortejar les diferents incidències que es venen donant en el transcurs del desenvolupament d'un projecte. Com és el cas del mòdul estadístic, que ha sigut la part en la que més incidències he tingut a la hora d'implementar-la.

Assolir tots els objectius ha estat tot un repte, donat les diferents incidències a la hora de donar resposta a cada un dels apartats de l'enunciat. Tot i així, si s'han assolit tots els objectius plantejats inicialment les taules, el mòdul ABM, el mòdul estadístic, el control de errors i el log de canvis.

S'ha seguit la planificació correctament, tot i que per una banda s'ha canviat l'ordre d'una de les fites que inicialment es varen interpretar incorrectament. Per altra banda s'ha trigat menys temps de l'estipulat inicialment per tal de implementar el mòdul de control de excepcions i errors.

Queda pendent revisar el funcionament del log amb versions de Postgresql futures per tal de comprovar que al introduir valors NULL a la clàusula WHERE en un camp NOT NULL reconeix aquest fet i es marca com excepció.

4. Glossari

ABM: Alta, Baixa, Modificació.

PLPGSQL: Acrònim en anglès de Procedural Language/PostgreSQL Structured Query Language.

Procediment emmagatzemat: Acció o funció definida per un usuari que proporciona un servei determinat. Un cop creat, es guarda en la base de dades i es tracta com un objecte més d'aquesta.

SGBD: Sistema de gestió de bases de dades. És un conjunt de programes que permeten el magatzematge, la modificació i extracció de la informació en una base de dades, a més de proporcionar eines per afegir, esborrar, modificar i analitzar les dades.

SQL: Acrònim en anglès de Structured Query Language. És un llenguatge declaratiu de accés a bases de dades relacionals que permet especificar diverses operacions en elles.

Trigger: Acció o procediment emmagatzemat que s'executa automàticament quan s'executa una operació d'inserció, d'esborrament o de modificació sobre alguna taula de la base de dades.

5. Bibliografia

- | | | |
|------|---|-----------------------|
| [1] | http://csrq.inf.utfsm.cl/~ifuentes/_build/html/lectures/week5/lecture20.html | Visitat el 02/05/2016 |
| [2] | http://www.postgresql.org/docs/9.5/static/datatype-enum.html | Visitat el 02/05/2016 |
| [3] | http://www.w3schools.com/sql/sql_primarykey.asp | Visitat el 02/05/2016 |
| [4] | http://www.postgresql.org/docs/current/static/plpgsql.html | Visitat el 02/05/2016 |
| [5] | http://www.postgresql.org/docs/9.1/static/sql-createtrigger.html | Visitat el 02/05/2016 |
| [6] | http://www.postgresql.org/docs/9.5/static/functions-datetime.html | Visitat el 02/05/2016 |
| [7] | http://gustavopeiretti.com/postgresql-division-de-enteros-incluyendo-cociente-con-decimales/ | Visitat el 02/05/2016 |
| [8] | http://www.postgresql.org/docs/current/static/plpgsql.html | Visitat el 02/05/2016 |
| [9] | http://forums.enterprisedb.com/posts/list/3515.page | Visitat el 02/05/2016 |
| [10] | http://www.postgresql.org/docs/current/static/plpgsql-control-structures.html | Visitat el 02/05/2016 |

6. Annexos

Sense Annexos