



Memòria

Disseny i implementació del data warehouse de Cadena de Roba

TFC ETIG

Tutor: Juan Martínez Bolaños

Rafel Perelló Columbram (rperelloc)

1^r semestre curs 15/16

A les meves nines, a les tres.

Mallorca, desembre de 2015.

Resum.

En aquest document presentam el projecte fi de carrera corresponent a l'Enginyeria Tècnica en Informàtica de Gestió de la UOC realitzat durant el primer semestre del curs 2015/16.

L'objectiu és desenvolupar una aplicació dins un SGBD per a gestionar les dades d'una cadena de botigues de roba que vol registrar les dades de vendes per a poder disposar d'informació sobre aquestes que li permeti conèixer la situació d'aquestes i prendre decisions en funció dels resultats tant de les diferents botigues com de les ciutats on es troben i dels productes que té al catàleg.

Més que disposar d'un producte acabat per a donar al client, l'objectiu del projecte és presentar les diferents etapes del procés de desenvolupament, començant per la fase d'especificació i disseny, passant per la implementació i acabant en la fase de test.

Així presentam aquestes etapes de tot el desenvolupament fet, exposant a cada punt tant les dificultats trobades com les solucions adoptades i les propostes de millora que en cada moment podem veure, tant si aquestes es poden implementar en el moment com si queden pendents per posteriors ampliacions.

Índex.

Índex de continguts

1. Descripció del projecte.....	1
2. Objectius.....	2
3. Enfocament i mètode seguit.....	5
3.1. Metodologia del disseny.....	5
3.1.1. Requisits.....	5
3.1.2. Anàlisi i disseny.....	5
3.1.3. Implementació.....	5
3.1.4. Proves.....	6
3.1.5. Manteniment.....	6
4. Planificació del projecte.....	7
4.1. Projecte complet. TFC.....	7
4.2. Primer lliurament.....	8
4.3. Segon lliurament.....	8
4.4. Lliurament final.....	8
5. Producte obtingut.....	9
6. Estructura de la memòria.....	11
6.1. Fase de disseny.....	11
6.2. Fase d'implementació.....	11
6.3. Test.....	11
7. Disseny.....	12
7.1. Definició i anàlisi dels requisits.....	12
7.1.1. Regles de negoci.....	12
7.1.2. Requisits funcionals.....	14
7.2. Disseny.....	17
7.2.1. Disseny conceptual.....	18
7.3. Disseny lògic.....	23
7.4. Disseny físic.....	24
8. Implementació.....	25
8.1. Taules.....	25
TAULES_CADENA.sql.....	25
TAULA_LOG.sql.....	26
8.2. Procediment de registre de la taula TAULA_LOG: ESCRIU.sql.....	26
8.3. Procediments d'Alta, Baixa i Modificació de dades:ABM_CADENA.sql.....	27
8.4. Procediments d'Alta, Baixa i Modificació de vendes:ABM_VENDES.sql.....	32

8.5. Procediments d'execució de consultes:CONSULTA_VENDES.sql.....	35
8.6. Procediments d'execució del mòdul estadístic: ACT_ESTADISTICA.sql i CONSULTA.sql..	39
9. Test.....	44
9.1. Test dels procediments del package abm_cadena.....	44
9.2. Test dels procediments del package abm_vendes.....	44
9.3. Test dels procediments del package act_estadistica.....	45
9.4. Test dels procediments del package consulta_est.....	46
9.4. Test dels procediments del package consulta_vendes.....	46
10. Recursos i valoració econòmica.....	49
10.1. Recursos humans.....	49
10.2. Maquinari.....	49
10.3. Programari.....	49
10.4. Hores i cost del projecte.....	50
11. Conclusions.....	51
12. Glossari.....	53
13. Bibliografia.....	55
14. Annex 1. (Test del package act_estadistica.).....	56

Índex de figures.

Índex d'il·lustracions

Figura 1. Etapes de desenvolupament del projecte.....	5
Figura 2. Diagrama de Gantt.....	7
Figura 3. Diagrama de Gantt del pla de treball.....	7
Figura 4. Diagrama de Gantt de la fase de disseny.....	8
Figura 5. Diagrama de Gantt de la primera part de la implementació.....	8
Figura 6. Diagrama de Gantt del lliurament final.....	8
Figura 7. Diagrama entitat-interrelació de l'aplicació.....	20
Figura 8. Diagrama UML del grau i la interconnectivitat de les interrelacions.....	21
Figura 9. Diagrama UML del mòdul estadístic.....	22

1. Descripció del projecte.

El projecte consisteix en el disseny i implementació d'una BD per a donar servei a l'empresa **Cadena de Roba**, per a controlar les seves botigues, catàleg i vendes. Es vol, a més, que realitzi la funció de magatzem de dades o *data warehouse*, a fi de poder controlar l'històric de dades i poder-hi fer consultes per analitzar resultats.

Bàsicament l'empresa vol tenir un registre de les seves botigues agrupades per ciutats i aquestes per regions, dels productes que ven i de les vendes que fa cada una de les botigues. El que pretén el producte és guardar aquestes dades i poder fer consultes sobre elles pel que fa a botigues, ciutats i productes en funció de les dates de venda que podem agrupar per anys o per anys i mesos.

A l'hora vol també disposar de dades en temps real de botigues, ciutats i productes amb més vendes i beneficis per a poder prendre decisions sobre la política de l'empresa.

Tot el producte s'implementarà sobre un gestor de bases de dades que gestionarà totes les dades mitjançant procediments que podrem cridar des de l'entorn que vulguem.

2. Objectius.

Per una banda l'empresa vol tenir registrades les botigues que formen la cadena, tant per la seva identificació (nom, ciutat on es troba ubicada, regió) com per la seva gestió (gerent de la botiga, nombre de treballadors, franquícia, botiga virtual).

Les característiques bàsiques que volem guardar són:

- Identificador de la botiga
- Ciutat de la botiga
- Regió de la botiga
- E-mail del gerent de la botiga
- Nombre de treballadors de la botiga
- Booleà que indiqui si és una franquícia o si pertany 100% a la cadena
- Booleà que indiqui si es tracta d'una botiga virtual o bé física

Pel que fa al catàleg, aquest és únic i hi vol tenir registrats els seus productes per a la seva identificació (codi, nom, descripció, preu, etc).

Les característiques bàsiques que volem guardar són:

- Identificador EAN13 del producte
- Descripció del producte
- Data d'incorporació al catàleg

I per les vendes, que és el que finalment donarà sentit a la BD, l'empresa pretén registrar-les associant-les a la botiga que les fa i als productes venuts, així totes les vendes que fa la cadena aniran associades amb la botiga que les fa, així com amb la data i hora i amb els productes venuts, quantitat, preu i beneficis.

Així, a la taula de vendes, taula de fets, es registraran totes les vendes produïdes amb les relacions esmentades amb botigues i catàleg. En aquest cas, degut a la grandària d'algunes botigues i el seu volum de vendes, es vol guardar a la taula de vendes les totals de cada producte fetes cada hora. Així en registrar una venda ho farem per tota l'hora sumant les quantitats venudes en aquest període.

Els valors que voldrem guardar a la taula seran, com a identificatius de cada entrada:

- Identificador de la botiga
- Identificador EAN13 del producte
- Data en format dd/mm/aaaa
- Hora sense minuts en format 24 hores, és a dir, que si es realitza una compra a les 10:24 del matí

desarem al camp el valor 10.

I com a dades:

- Quantitat total del producte venuda aquell dia i hora
- Preu brut total dels productes venuts aquell dia i hora
- Benefici net total dels productes venuts aquell dia i hora

A la base de dades s'hauran d'implementar els procediments per a poder mantenir el registre de botigues, altes, baixes i modificacions, com el catàleg en els mateixos termes.

- Procediments d'ABM (Alta + Baixa + Modificació) de les botigues, dels productes i de la taula de fets.

Així mateix s'hauran d'implementar els procediments per a poder consultar les dades de vendes. L'empresa vol poder consultar dades mensuals de vendes i beneficis de cada botiga, vendes i beneficis de cada producte així com les vendes i beneficis per dies de tot el grup.

- Donat un any i un mes concret el llistat de totes les botigues de la cadena incloent per a cada botiga:
 - El nombre total de productes venuts aquell mes.
 - El nombre de productes diferents venuts aquell mes.
 - El benefici net total obtingut aquell mes.
 - El percentatge de benefici que aporta la botiga en relació al total de beneficis obtinguts per tota la cadena aquell mes.
 - El benefici net dividit pel nombre d'empleats de la botiga.
 - Tot ordenat pel benefici net de cada botiga de forma descendent.
- Donats un any i un mes concret com a paràmetres d'entrada: el llistat de tots els productes del catàleg incloent:
 - Identificador EAN13.
 - Nom del producte.
 - Nombre d'unitats venudes.
 - Benefici net que ha generat el producte.
 - Botiga que n'ha venut més unitats així com el nombre d'unitats que ha venut.
 - Ordenat pel benefici generat de forma descendent.
- Donat un any i un mes concret cal retornar un llistat de tots els dies del mes, incloent-hi per a cada dia:
 - Benefici total net obtingut aquell dia per tota la cadena

- Identificador EAN13 del producte més venut, juntament amb les unitats venudes.
- Identificador de la botiga que més benefici net ha generat juntament amb el valor en euros d'aquest benefici.

Finalment, el mòdul estadístic, pretén que l'empresa pugui conèixer, en temps real i sense haver d'esperar un temps innecessari per a la realització dels càlculs, els beneficis del grup, les botigues, ciutats i productes més productius, així com els dies i hores on més vendes hi ha. També distingir els resultats de les botigues virtuals i les físiques.

El mòdul estadístic haurà de donar resposta immediata a les següents consultes:

- Donat un any concret:
 - El benefici net total de tota la cadena
 - L'identificador de la botiga que més beneficis nets ha aconseguit, així com la xifra total d'aquest benefici en euros.
 - L'identificador EAN13 del producte més venut així com la quantitat total d'unitats venudes.
 - L'hora del dia on més productes s'han venut i la xifra de total de productes venuts.
 - L'hora del dia on menys productes s'han venut i la xifra total de productes venuts.
 - El dia del mes on més vendes s'han realitzat i la xifra total de productes venuts.
 - El dia del mes on menys vendes s'han realitzat i la xifra total de productes venuts.
 - Ciutat on més beneficis nets s'han obtingut juntament amb aquest benefici.
 - Percentatge de beneficis obtinguts per tendes virtuals respecte al total de beneficis de la cadena.
- Les mateixes consultes que abans però de forma històrica, és a dir, tenint en comptes tots els anys d'existència de la cadena.

3. Enfocament i mètode seguit.

3.1. Metodologia del disseny.

A la figura següent¹, podem veure el mètode de desenvolupament de programari que hem escollit per dur a

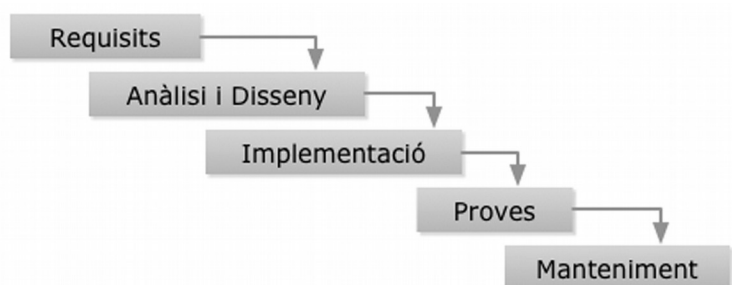


Figura 1. Cicle de vida en cascada.

terme el desenvolupament del projecte. Hem escollit el cicle de vida clàssic o en cascada degut a que és el que millor s'ajusta a les característiques del projecte que desenvolupam. Es tracta d'un projecte que és de curta durada, disposam de tres mesos per a desenvolupar-lo, i amb dues característiques, té un objectiu clar i de solució coneguda. Per tant escollim el mètode per la seva seguretat en el desenvolupament acceptant la poca tolerància als canvis, ja que no n'hi suposam per la mateixa naturalesa del projecte.

Així dividirem el desenvolupament del projecte en quatre etapes:

3.1.1. Requisits.

En aquesta primera etapa, definim quins han de ser els objectius i requisits que ha de complir el producte. Degut al tancament que presenta el mètode, aquesta primera etapa és crítica i ha de definir clarament tots els objectius que ha de dur a terme el producte final, ja que qualsevol canvi suposaria refer etapes anteriors amb un ús de recursos no contemplat.

3.1.2. Anàlisi i disseny.

En aquesta segona etapa hem de definir com ha de ser la base de dades tant internament com externament. A la part d'anàlisi dissenyarem el models conceptual i a la part de disseny, els models lògic i físic de tota l'estructura que implementarem.

3.1.3. Implementació.

En aquesta etapa, crearem totes les taules i consultes que formaran la BD.

3.1.4. Proves.

1 Materials docents de la UOC. Pradel Miquel, Jordi i altres. Enginyeria del programari. Mòdul 1. Introducció a l'enginyeria del programari (PID_00213624).

La darrera etapa correspon a la prova i control del producte final realitzat. Aquesta es durà a terme amb una sèrie de dades fictícies però que serviran tant per a les proves de control com per presentar el producte i mostrar-ne les seves funcionalitats.

3.1.5. Manteniment.

Aquesta fase, contemplada al mètode de desenvolupament del programari, no està contemplada en el projecte, i per tant queda pendent d'un segon contracte un cop presentat el programa. En aquesta etapa es contempla tant la correcció de possibles errors o petites modificacions que no afectin a l'estructura general del producte així com a l'afegit de noves funcions.

4. Planificació del projecte.

El projecte s'ha desenvolupat amb una durada de 15 setmanes i dos dies, amb un càlcul de 21 hores setmanals, que en principi es distribueixen en 7 dies. A més aquest temps s'ha dividit en quatre períodes, que venen marcats pel compromís de contracte del client, que preveu la presentació del pla de treball, dues entregues parcials i el lliurament final.

Les tasques de feina s'han distribuït de la següent manera:

Pla de treball.	26-setembre / 4-octubre	9 dies
Fase de disseny.	5-octubre / 8-novembre	35 dies
Fase d'implementació.	9-novembre / 6-desembre	28 dies
Fase de proves i presentació final.	7-desembre / 10-gener	35 dies

En aquesta distribució hem considerat que la fase d'implementació ha sigut una mica més llarga del temps compromès per a la segona entrega parcial, per tant la darrera part de la implementació, concretament el mòdul estadístic, no s'ha entregat en aquesta data i s'ha desenvolupat durant les dues primeres setmanes del quart període.

A continuació presentam el diagrama de Gantt de tot el projecte, així com els diagrames de les quatre fases amb que l'hem dividit. En ells podem veure la distribució per dies de les etapes i les tasques en que hem dividit el projecte.

4.1. Projecte complet. TFC.

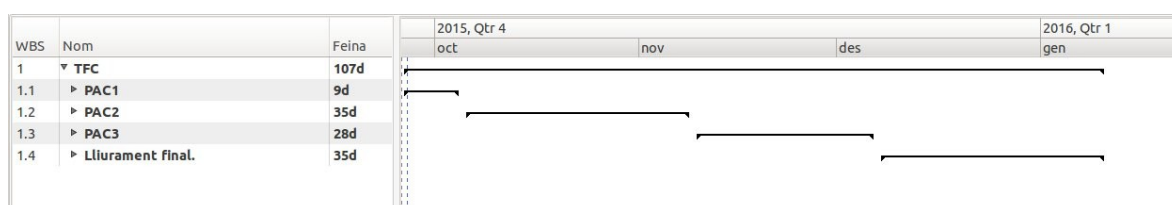


Figura 2. Diagrama de Gantt del projecte TFC.

Pla de treball.

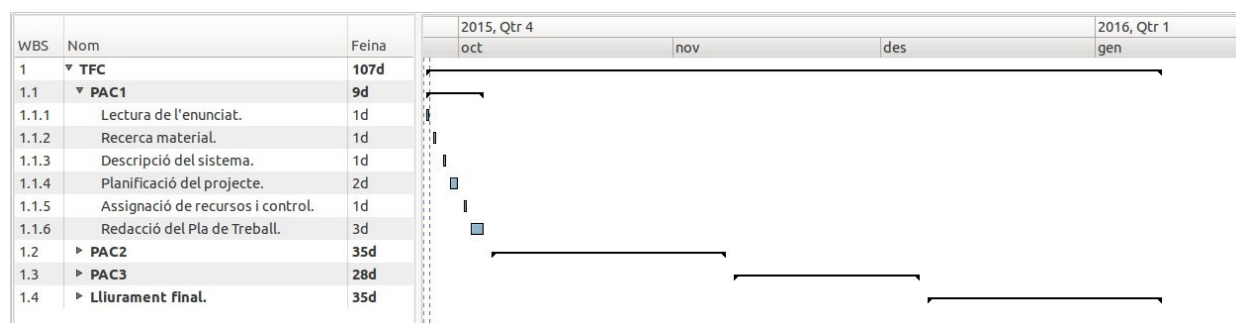


Figura 3. Diagrama de Gantt de l'elaboració del pla de treball.

4.2. Primer lliurament.

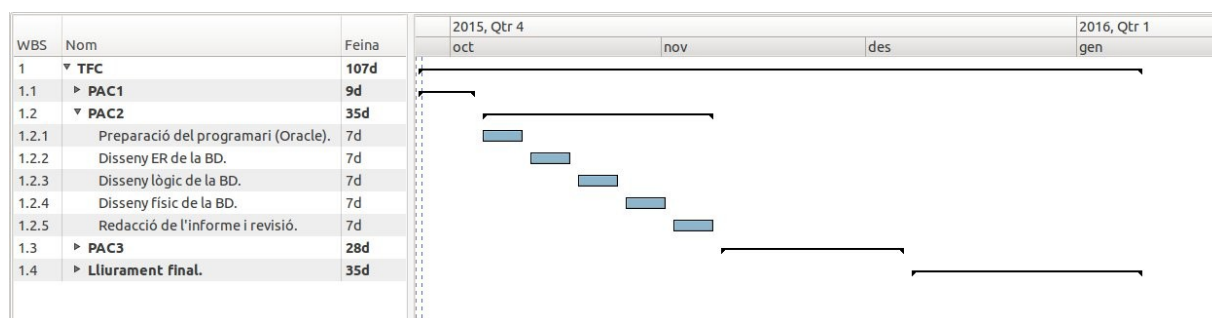


Figura 4. Diagrama de Gantt de l'etapa del primer lliurament.

4.3. Segon lliurament.

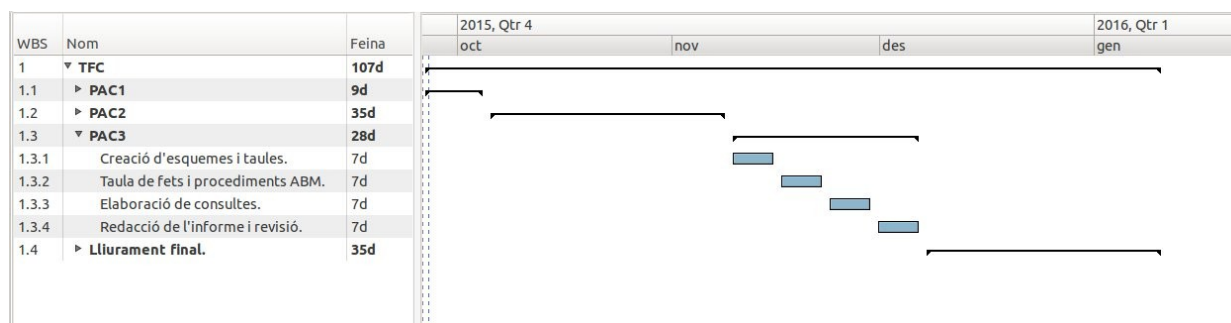


Figura 5. Diagrama de Gantt de l'etapa del segon lliurament.

4.4. Lliurament final.

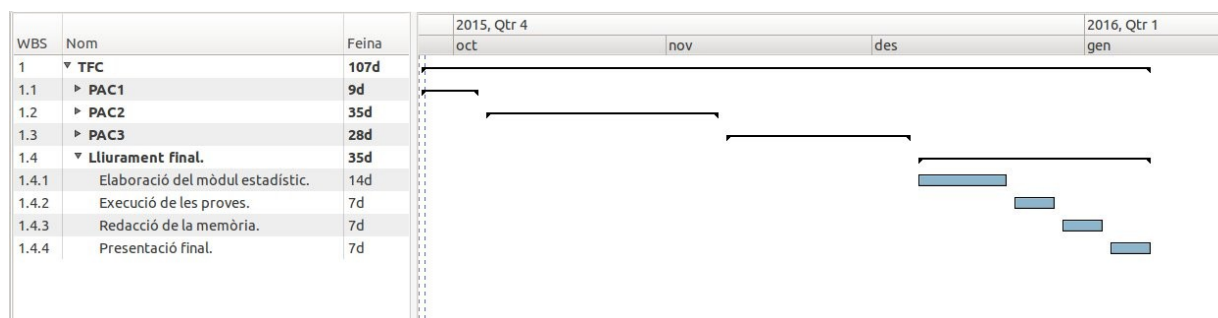


Figura 6. Diagrama de Gantt de l'etapa del lliurament final.

5. Producte obtingut.

El producte final consisteix en:

- una base de dades que definim en el fitxer BD.sql que contindrà tot el producte,
- una col·lecció de taules que definim al fitxer TAULES_CADENA.sql on guardam totes les dades,
- els procediments per a introduir, donar de baixa o actualitzar totes les dades ja siguin de les botigues com dels productes de l'empresa, que presentam al fitxer ABM_CADENA.sql,
- els procediments per a introduir, donar de baixa o modificar les vendes de les diferents botigues, que presentam al fitxer ABM_VENDES.sql,
- el procediment per a actualitzar en temps real la taula de resultats estadístics cada cop que gestionam una venda, que presentam al fitxer ACT_ESTADISTICA.sql,
- un conjunt de procediments per a executar les consultes demanades, que presentam en els fitxers CONSULTA_VENDES.sql i CONSULTA_EST.sql pel que fa a les consultes sobre les vendes d'un any i mes o per les consultes sobre el mòdul estadístic respectivament,
- i un seguit de funcions i procediments auxiliars que presentam en els mateixos fitxers o en els fitxers ESCRIU.sql, INICI_ESTADISTICA.sql i TAULA_LOG.sql.

Per a instal·lar el producte els fitxers 'sql' adjuntats s'han d'executar el l'ordre següent, després d'haver instal·lat l'Oracle Database Express i des d'un usuari amb drets d'administració:

```
BS.sql
TAULES_CADENA.sql
TAULA_LOG.sql
ESCRIU.sql
ABM_CADENA.sql
ABM_VENDES.sql
INICI_ESTADISTICA.sql
ACT_ESTADISTICA.sql
CONSULTA_VENDES.sql
CONSULTA_EST.sql
```

Els procediments que hem d'executar per a utilitzar l'aplicació i que descrivim en els apartats següents són:

Per gestionar l'estructura de l'empresa:

```
PROCEDURE ALTA_REGIO ( IDREGIO IN NUMBER, NOM IN VARCHAR2 );
PROCEDURE ALTA_CIUTAT ( IDCIUTAT IN NUMBER, NOM IN VARCHAR2, IDREGIO IN NUMBER );
PROCEDURE ALTA_BOTIGA ( IDBOTIGA IN NUMBER, NOM IN VARCHAR2, EMAIL IN VARCHAR2,
NTREBALLADORS IN NUMBER, FRANQUICIA IN NUMBER, EBOTIGA IN NUMBER, IDCIUTAT IN
NUMBER );
```

PROCEDURE BAIXA_BOTIGA (IDBOTIGA IN NUMBER);

PROCEDURE MODIFICA_BOTIGA (IDBOTIGA IN NUMBER, NOM IN VARCHAR2, EMAIL IN VARCHAR2, NTREBALLADORS IN NUMBER, FRANQUICIA IN NUMBER, EBOTIGA IN NUMBER, IDCIUTAT IN NUMBER);

PROCEDURE ALTA_PRODUCTE (EAN13 IN NUMBER, NOM IN VARCHAR2, DESCRIPCIO IN VARCHAR2, DATAIN IN DATE, DATAOUT IN DATE, PREU IN NUMBER, BENEFICI IN NUMBER);

PROCEDURE BAIXA_PRODUCTE (EAN13 IN NUMBER);

PROCEDURE MODIFICA_PRODUCTE (EAN13 IN NUMBER, NOM IN VARCHAR2, DESCRIPCIO IN VARCHAR2, DATAIN IN DATE, DATAOUT IN DATE, PREU IN NUMBER, BENEFICI IN NUMBER);

Per a gestionar les vendes:

PROCEDURE ALTA_VENDA (IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER, UNITATS IN NUMBER);

PROCEDURE BAIXA_VENDA (IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER);

PROCEDURE MODIFICA_VENDA (IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER, UNITATS IN NUMBER);

Per les consultes demanades:

PROCEDURE CONSULTA1 (A in number, M in number);

PROCEDURE CONSULTA2 (A in number, M in number);

PROCEDURE CONSULTA3 (A in number, M in number);

Pel mòdul estadístic:

PROCEDURE CONSULTA (A in number);

PROCEDURE CONSULTA_ANY;

6. Estructura de la memòria.

El desenvolupament del projecte el presentam en els tres capítols següents, que descriuen tant el disseny com la implementació i test de tot el projecte.

6.1. Fase de disseny.

En el capítol 7, presentam el disseny de la BD amb els esquemes ER, lògic i físic, on es veuen les taules que formen la BD amb tots els seus camps, els dominis d'aquests i les claus primàries i foranes i les relacions entre aquestes.

De fet al model ER de disseny conceptual s'especifica la informació que ha de guardar la BD, i la seva estructuració, sense cap relació amb les qüestions tecnològiques.

A l'esquema lògic traslladam aquesta estructura al model de BD que volem fer servir, en el nostre cas el model relacional, definint apart dels camps i atributs de les taules, les claus primàries, foranes i relacions.

I a l'esquema físic definim tota l'estructura adaptant-la al programari que utilitzarem, que serà l'Oracle Express Edition 11g Release 2.

6.2. Fase d'implementació.

En el capítol 8, que es correspon amb la fase d'implementació, presentam totes les taules, consultes i procediments que formen el projecte.

6.3. Test.

En el capítol 9, presentam el test fet a la BD amb un conjunt de dades fictícies, però adaptades a l'ús que n'ha de fer l'empresa per a poder provar i mostrar la seva funcionalitat.

7. Disseny.

En aquest capítol presentam l'anàlisi i el disseny funcional de la BD elaborat per a **Cadena de Roba**.

El primer que presentam són les regles de negoci, es a dir el recull d'informació sobre els diferents objectes del món real que representarem a la BD, dels que necessitam conèixer les seves propietats que guardarem com a atributs en la nostra representació.

Seguidament representam els requisits funcionals, això és les característiques que ens donaran les necessitats o restriccions que han de verificar les entitats de la BD desenvolupada.

A partir de les regles de negoci i les regles funcionals, que ens permetran tenir ben especificat el problema a resoldre, donam el disseny de la base de dades, això és l'estructura de les dades que guardarem, els atributs que tendran i les relacions que hi ha entre ells. Per fer-ho seguim les tres etapes més habituals, donam els esquemes del disseny conceptual, del disseny lògic i finalment el disseny físic de la BD ja en el programari que hem triat per a implementar l'aplicació, així com les instruccions en el llenguatge SQL per a definir-les.

7.1. Definició i anàlisi dels requisits.

7.1.1. Regles de negoci.

Botigues.

La primera regla de negoci fa referència a les botigues de l'empresa. Especialment a l'anàlisi de resultats ens interessarà poder relacionar aquests amb les característiques de la botiga en qüestió, per tant són importants les regles de negoci que la caracteritzen.

RN1	Descripció
Botigues	<ol style="list-style-type: none">1. Volem conèixer la gestió de la botiga per a considerar-ho a l'anàlisi de resultats. Guardarem el gerent d'aquesta com a responsable de la gestió i el nombre de treballadors.2. Distingirem entre els dos tipus de propietat, les franquícies i les botigues propietat de la cadena.3. Tendrem, i per tant distingirem, botigues físiques i virtuals, ja que tenen costs i gestió totalment diferents.

Catàleg.

La segona regla de negoci fa referència al catàleg. En voler estudiar les vendes, però especialment els

resultats, hem de considerar també els beneficis. Pel que fa al preu dels productes aquests es guardarà en euros, ja que en aquest moment l'empresa es mou en aquesta zona monetària, però en properes revisions o ampliacions, sempre que l'empresa s'escampi per altres zones, hauríem de considerar la possibilitat d'introduir diferents monedes. Tot i això aquest canvi no s'hauria de guardar en l'estructura del catàleg, ja que el preu de catàleg podria seguir essent únic sinó que s'hauria d'afegir una taula per fer el canvi a altres unitats de moneda.

RN2	Descripció
Catàleg	<ol style="list-style-type: none">1. Aquest catàleg és únic per a totes les botigues.2. Per cada producte guardarem el preu brut de venda però també el benefici que dona la seva venda. En principi aquests són els mateixos per tota la cadena, però en properes revisions o ampliacions s'hauria de poder modificar tant el preu com el benefici, ja que aquest no serà el mateix ni en les botigues de diferent tipus ni en les del mateix tipus situades a diferents llocs.3. Tots els preus es guardaran en euros.

Vendes.

Pel que fa a les vendes, sobre el que gira tota la BD, ja que el seu objectiu principal és servir a l'empresa per analitzar els seus resultats, volem poder classificar aquestes per tots els actors que hi prenen part, això és producte, botiga, moment de venda i benefici.

RN3	Descripció
Vendes	<ol style="list-style-type: none">1. Cada venda ha de identificar la botiga on s'ha produïda i el producte venut. Per tant si un client s'emporta dos o més productes diferents el registrarem com a vendes diferents en la BD.2. De cada venda registrarem a més la data i l'hora en que s'han produït.3. Pel nombre elevat de vendes, guardarem totes les vendes del mateix producte en una mateixa botiga a la mateixa hora natural com a una única venda on registrarem el nombre de productes venuts i l'hora sense minuts.4. A la venda guardada hi constarà el preu brut i el benefici d'aquesta, però també el nombre de productes, ja que s'haurà de tenir en compte a l'hora de fer comparacions (per exemple,

	ens interessarà el benefici per unitat).
--	--

Pel que fa a la regla de negoci de les vendes hem de fer un comentari important. El disseny de l'aplicació que estam fent està enfocat als productes de l'empresa i per tant als beneficis d'aquest producte. Això ens permet veure els beneficis, les botigues i les hores o dies on són majors les vendes o els beneficis. Però al primer punt de la RN3 fem constar que un client pot endur-se més d'un producte o bé més d'una unitat del mateix producte. Cap d'aquests dos aspectes quedarà reflectit a l'aplicació, ja que aquesta guardarà només el nombre de productes venuts, i no possibles coincidències en una mateixa venda. Consideram que aquesta podria ser una ampliació interessant per a l'empresa, ja que apart de l'anàlisi dels productes i botigues, podria analitzar els clients, en aspectes com si compren un sol producte o més d'un o bé si s'emporten més d'una unitat d'un mateix producte. Aquest seria un aspecte molt important a l'hora de poder planificar ofertes, promocions o campanyes específiques que es dirigirien a un tipus de client concret i no centrar-se només en l'especificació dels productes. De totes formes aquesta ampliació no seria senzilla com altres que anirem comentant, com la possibilitat d'acceptar diferents tipus de moneda, ja que aquesta ampliació requeriria el canvi del disseny, hauríem de modificar la interrelació venda i convertir-la en una entitat, ja que apart dels atributs relacionats amb productes, i data i hora de venda, hauria de guardar el nombre de productes diferents venuts a cada una, i aquest aquest atribut no és atribuïble a cap de les altres entitats. Així no es considerarà com una ampliació factible en properes revisions sinó com una ampliació del producte que requeriria un nou contracte i el procés de disseny del nou mòdul.

Mòdul estadístic.

Per acabar veurem les regles de negoci que ha de complir el mòdul estadístic de la BD.

RN4	Descripció
Mòdul estadístic	<ol style="list-style-type: none"> 1. Volem una resposta immediata a la consulta, per tant haurà d'estar constantment actualitzat. 2. Les consultes que li demanarem seran per anys naturals, o bé l'any en curs o qualsevol altre. A més podrem consultar un llistat històric de tots els anys d'activitat de la cadena.

7.1.2. Requisits funcionals.

A continuació veurem les funcionalitats que ha de proporcionar el sistema, com els càlculs que ha de fer, les dades que ha de guardar, com les ha de guardar (no el disseny físic, sinó de quina manera, amb quin format extern,...), com les manipularà, etc.

Així els requisits funcionals seran:

RF1	Descripció
Botigues	Per cada botiga volem guardar: <ul style="list-style-type: none">1. Identificació2. Nom comercial3. Ciutat4. Regió5. E-mail del gerent6. Nombre de treballadors7. Si és una franquícia o propietat de la cadena8. Si és una botiga virtual o física

Consideram la regió de la botiga enlloc del país per les característiques de la cadena, tot i que el canvi seria immediat i no suposaria cap cost per al producte. De totes formes en una expansió per països europeus, degut a la mida i al nombre de regions d'aquests seria convenient seguir amb aquesta classificació. (És més significativa una classificació en regió Mediterrània o de les Illes Canàries que classificar totes les botigues com a espanyoles.

RF2	Descripció
Catàleg	Per cada producte del catàleg volem guardar: <ul style="list-style-type: none">1. Identificador EAN13 del producte2. Nom i descripció del producte3. Data d'incorporació al catàleg4. Data de baixa al catàleg5. Preu brut6. Benefici de la venda

Hem afegit la data de baixa d'un producte al catàleg, que no havíem considerat al document d'especificacions perquè en l'anàlisi de resultats, per exemple anuals, s'hauria de considerar que un producte pot haver-se donat de baixa a mitjan any, tot i que aquest aspecte queda pendent per a posteriors ampliacions del producte.

RF3	Descripció
Vendes	Per cada venda, això és els productes venuts a cada tenda en una hora natural, volem guardar: <ul style="list-style-type: none">1. Identificador de la botiga

	2. Identificador EAN13 del producte 3. Data de la venda 4. Hora de la venda (només l'hora, sense minuts) 5. Nombre de productes 6. Preu brut de la venda 7. Benefici de la venda
--	---

En aquest cas havíem d'optar entre guardar el preu brut i el benefici per producte o per venda, i hem optat pel segon. En primer lloc perquè les quantitats per producte ja són al catàleg i per tant seria redundant guardar-les però a més aquests preus poden variar, i això faria que els llistats trets després d'aquest canvi no reflectissin els preus i beneficis reals de la venda. En segon lloc també hem considerat que les consultes que farem van dirigides a conèixer beneficis o ingressos per producte, i d'aquesta manera només haurem de sumar els valors de les vendes i no fer en cada cas la multiplicació de preu unitari per nombre d'unitats, que en tenir un nombre elevat de vendes podria arribar a ser considerable i repetit molts cops.

Així mateix en aquest cas no hem inclòs un identificador per cada venda ja que la identificació d'aquestes la farem a partir dels quatre primers atributs que hem donat, això és identificador de la botiga, identificador del producte, data i hora.

RF4	Descripció
Ciutats	Per cada ciutat volem guardar: 1. Identificador 2. Nom de la ciutat 3. Identificador de la regió on es troba

RF5	Descripció
Regions	Per cada regió volem guardar: 1. Identificador 2. Nom de la regió

La classificació per ciutats o regions és important ja que tal i com consta a l'apartat d'especificació, demanarem a l'aplicació llistats de vendes o beneficis per ciutats, però que també podrem estendre a regions. En aquest cas, tant pel que fa a la diferència de ciutats com de regions, entenem que el seu coneixement permet avaluar i també dissenyar campanyes publicitàries o adaptacions de les botigues o del catàleg a les diferents zones, ja que les característiques socials, econòmiques, etc. d'aquestes (que no guarda la nostra aplicació) seran determinants per aquestes especificacions.

Així entenem que en properes ampliacions, podríem incloure en les dades de ciutat o regió, camps com població, nivell econòmic, social o altres que permetin contrastar vendes en funció d'algun d'aquests paràmetres.

RF6	Descripció
Altes i baixes	<p>El sistema ens ha de permetre gestionar:</p> <ol style="list-style-type: none"> 1. Procediments d'ABM, (alta, baixa i modificació), de botigues 2. Procediments d'ABM de productes del catàleg
RF7	Descripció
Consultes	<p>El sistema ens ha de permetre dur a terme les consultes donades en els procediments següents:</p> <ol style="list-style-type: none"> 1. Per un any i mes donats, llistar per cada botiga el nombre total de productes venuts, el nombre de productes diferents, el benefici net, el percentatge del benefici respecte al total de la cadena i el benefici per empleat. Ordenant les botigues pel benefici total. 2. Per un any i mes donats, llistar tots els productes del catàleg fent-hi constar l'identificador EAN13, el nom, les unitats venudes, el benefici net de cada producte i la botiga que n'ha venut més, donant-ne el nombre. Ordenant els productes pel benefici net generat. 3. Per una any i mes donats, llistar tots els dies del mes amb el benefici net de tota la cadena, l'identificador del producte més venut i les unitats venudes i la botiga amb més benefici.
RF8	Descripció
Mòdul estadístic	<p>Pel que fa al mòdul estadístic, el sistema ens ha de permetre dur a terme les consultes donades en els procediments següents tant en un any concret donat (que pot ser l'any en curs) com en l'històric d'anys en que la cadena ha estat funcionant:</p> <ol style="list-style-type: none"> 1. El benefici net total de la cadena 2. La botiga amb més beneficis nets i aquesta quantitat 3. El producte més venut així com el nombre total d'unitats d'aquest 4. L'hora del dia amb més vendes i aquesta xifra de productes venuts 5. L'hora del dia amb menys vendes i aquesta xifra de productes venuts 6. El dia amb més vendes i la xifra de productes venuts 7. El dia amb menys vendes i la xifra de productes venuts 8. La ciutat amb més beneficis i el valor d'aquest benefici 9. El percentatge de beneficis obtinguts per les tendes virtuals en front de les de la cadena

7.2. Disseny.

En aquest apartat presentam les tres etapes del disseny de la base de dades a partir de les regles de negoci i funcionals vistes a l'apartat anterior.

7.2.1. Disseny conceptual.

Presentam el disseny conceptual de la base de dades del projecte. En aquesta primera etapa volem representar en un model d'alt nivell, l'estructura de la informació de la BD. Per fer-ho utilitzarem el model entitat-interrelació, que inclou aquests dos elements, els que anomenem entitat i interrelació com a principals de la representació.

En el model representarem les entitats, que representaran objectes amb determinades propietats que volem guardar en la nostra aplicació. Les propietats d'aquests objectes els anomenarem atributs de les relacions.

Hem escollit com a entitats les següents:

REGIO
Atributs
<u>idRegio</u>
nom

Tal i com hem comentat en l'apartat anterior, hem escollit REGIO com una entitat perquè totes les botigues estan localitzades a una ciutat i aquestes ciutats a una regió. Tot i que podríem haver considerat la regió com un atribut de l'entitat CIUTAT, ens pot interessar en un futur estudiar els resultats per regions o bé considerar altres aspectes d'aquestes, com itineraris de repartiment, o bé relacionar-les amb una entitat superior com país i per tant ens convé que aquesta sigui una entitat. De moment només ens interessa conèixer-ne el nom. Distingirem el nom de la clau primària a fi que aquesta sigui més curta i senzilla.

CIUTAT
Atributs
<u>idCiutat</u>
nom

Pel que fa a CIUTAT, l'hem considerada una entitat pels mateixos motius que REGIO, ja que tot i que en aquest moment no ho fem servir, les tenim agrupades per regions. Només ens interessa conèixer-ne el seu nom. També distingirem el nom de la clau primària.

BOTIGA
Atributs
<u>idBotiga</u>
nom
eMail
nTreballadors
franquicia
ebotiga

Pel que fa a les botigues de la cadena és evident la seva consideració com a entitat. En aquest cas ens

interessa conèixer el seu nom, el correu electrònic del seu gerent, (i que per tant l'identificarà si és necessari), el nombre de treballadors i si aquesta és una franquícia o si és una botiga virtual. Aquests atributs *franquícia* i *ebotiga*, només podran prendre dos valors, vertader si ho són i fals en cas contrari, per tant els podrem representar amb un booleà. Com a clau primària agafarem un identificador per cada botiga que no serà el seu nom, que podria ser una clau candidata, però preferim diferenciar-los a fi que el nom pugui coincidir amb el nom comercial que podria duplicar-se en diferents botigues.

PRODUCTE
Atributs
ean13
nom
descripcio
dataIn
dataOut
preu
benefici

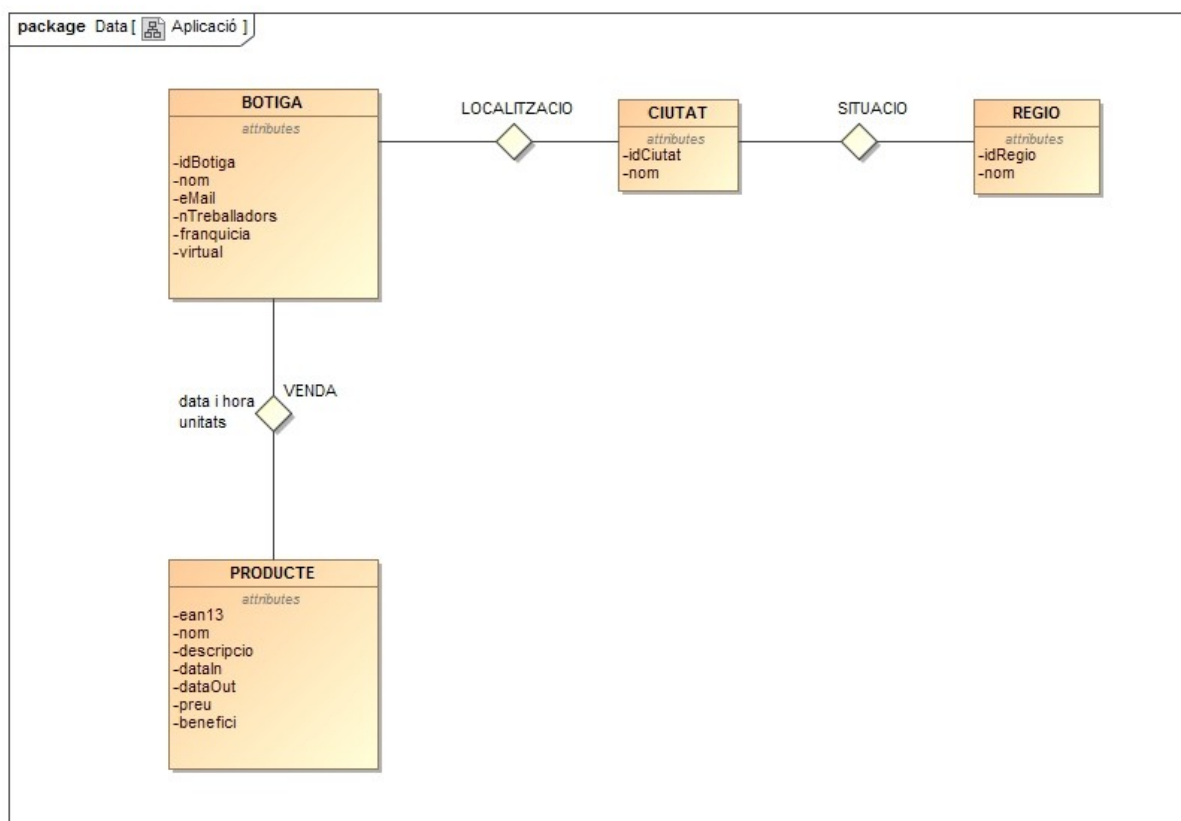
Pel que fa al producte és també clara la seva tria com a entitat i ens interessa conèixer-ne el seu nom, la descripció, les dates en que entren al catàleg i en que en surten, el preu i el benefici net que n'obtenim en la seva venda. La clau primària d'aquesta entitat serà l'EAN13, l'identificador de 13 dígits que porten tots els productes comercials i que avui dia llegeixen amb lectors de codis de barres a les botigues, en particular a les de la nostra cadena.

Per al registre de les vendes, el que seria la taula de fets de l'aplicació hem de fer un parell de consideracions. Podríem haver considerat aquesta com una entitat, anomenada VENDA, i relacionar-la amb BOTIGA i PRODUCTE, però en fer-ho veuríem que aquesta venda està identificada per tres atributs, que en serien la seva clau primària, idBotiga, idProducte i la data de la venda. Pel que fa als dos primers són clau primària de les entitats BOTIGA i PRODUCTE, per tant no aporten cap informació nova i ens indica que VENDA no és més que una interrelació entre aquestes dues, tot i que no està del tot identificada fins a que li afegim la data d'aquesta. Així sembla clar que VENDA serà una interrelació però que per estar identificada haurà de tenir una data i hora. Hem de fer una altra consideració a l'hora de guardar la data i l'hora, és que no volem el format de data i hora que per defecte ens proposen els sistemes de BD sinó que volem guardar només l'hora com un enter per identificar una de les 24 hores del dia. Això mateix ho farem pels dies, mesos i anys, els guardarem com a atributs separats i com a nombres enters. D'aquesta manera ens serà més fàcil a l'hora de fer consultes, ja que no haurem d'extreure de l'atribut data el valor del mes o de l'any cada cop que vulguem fer una consulta amb aquest valor. Com a clau primària agafarem els quatre atributs, tot i que podríem utilitzar un sol atribut que fos un camp de tipus data i hora que ofereixi la BD ens trobaríem que

aquest inclou probablement els minuts i segons i la nostra aplicació no els farà servir. Així hem preferit aquesta tria per evitar confusions en l'ús d'aquesta clau.

Presentam el diagrama UML de la part principal de l'aplicació on es donen les quatre entitats que hem vist i les interrelacions entre aquestes (Figura 7).

Figura 7. Diagrama conceptual entitat-interrelació de l'aplicació.



LOCALITZACIO. Aquesta representa la relació binària entre una botiga i la ciutat on aquesta es troba. No té atributs, ja que només relacionam la botiga amb la ciutat. Cal fer notar que hi haurà botigues virtuals i per tant tenim que aquestes no estan localitzades a cap ciutat, així hem definit una ciutat VIRTUAL per a localitzar-hi aquestes botigues i no mesclar les seves vendes amb les d'una ciutat concreta on les podríem pogut ubicar.

Pel que fa a la connectivitat, ens trobam que una ciutat estarà situada en una única regió i en canvi en una regió i podrem trobar diferents ciutats, així tendrem una connectivitat 1:N.

SITUACIO. En aquest cas representa la interrelació binària entre les ciutats i les regions on es troben les botigues. Tot i que una botiga podia ser virtual i per tant no estar ubicada a cap ciutat, si que totes les ciutats es troben ubicades a una regió i per tant REGIO és una entitat obligatòria.

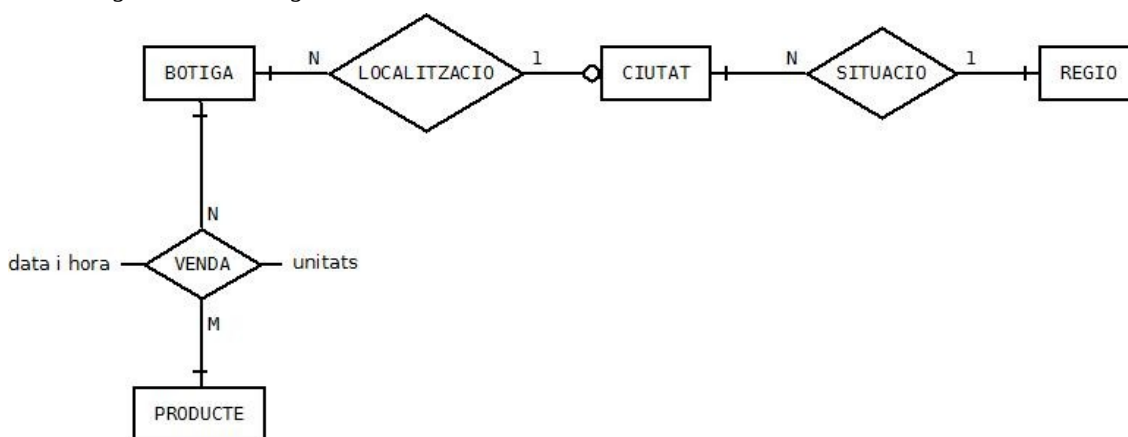
Per la connectivitat ens trobam amb el mateix cas que en les regions i per tant tornam a trobar una connectivitat 1:N.

VENDA. Aquesta és la interrelació més complicada en la seva definició. Hem considerat VENDA com una interrelació amb dos atributs que són la data i hora i el nombre d'unitats venudes del producte.

Si volem estudiar la connectivitat en aquest cas, ens trobam que per una data concreta i un producte concret podrem tenir que hi ha més d'una botiga que faci una venda, així BOTIGA es connecta amb una N a la interrelació. Si ho repetim per una botiga i una data concreta ens trobam que podrem tenir vendes de diferents productes, així PRODUCTE també es connecta amb una M a la interrelació. Es a dir que la connectivitat en aquest cas és M:N.

Hem representat el grau i la connectivitat de les interrelacions que hem comentat en el diagrama de la Figura 8.

Figura 8. Diagrama UML del grau i la interconnectivitat de les interrelacions.



anny: serà l'atribut que identificarà el registre, ja que és el que volem consultar.

benefici: ens donarà el valor del benefici total de la cadena en l'any consultat.

idBotiga: ens donarà l'identificador de la botiga amb més beneficis durant l'any que es consulta.

b_idBotiga: ens donarà el benefici net de la botiga idBotiga en aquest any.

ean13: ens donarà l'identificador del producte més venut durant l'any.

u_ean13: ens donarà les unitats del producte EAN13 venudes durant l'any.

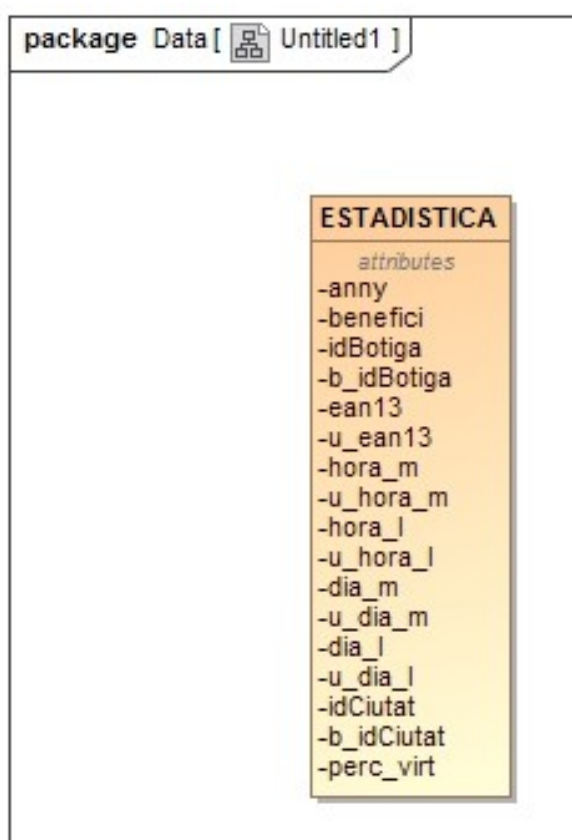
hora_m: de totes les hores del dia, 1...24, ens ha de donar la que ha tingut més vendes en l'any.

u_hora_m: representarà les unitats venudes en l'hora Hora_M guardada com atribut anterior.

hora_l: igual que l'atribut anterior, ens dóna l'hora del dia amb menys vendes durant l'any.

u_hora_l: ens dóna les unitats venudes en l'hora Hora_L.

Figura 9. Diagrama UML del mòdul estadístic.



u_dia_m: ens donarà les unitats venudes en aquest dia.

dia_l: ens dóna el dia del mes amb menys vendes durant l'any.

u_dia_l: ens dóna les unitats venudes durant el dia Dia_L.

id_ciutat: ens dóna l'identificador de la ciutat amb més benefici durant l'any consultat.

b_idCiutat: ens dona el benefici obtingut per les botigues de la ciutat durant l'any consultat.

perc_virt: per cada any guardarà la proporció entre el benefici de les botigues virtuals i les físiques que serà el que llistarà.

7.3. Disseny lògic.

A partir del disseny conceptual presentat al punt 7.2 ara obtindrem el disseny lògic. Per aquest disseny utilitzarem el model relacional, format per relacions² i per claus primàries i foranes. Així ens quedaran les relacions següents, on identifiquem la clau primària amb lletra subratllada i les claus foranes que té cada una de les relacions.

BOTIGA(idBotiga, nom, eMail, nTreballadors, franquicia, ebotiga, idCiutat)

on {idCiutat} referencia CIUTAT.

Tots els atributs que apareixien en l'entitat BOTIGA de l'esquema conceptual s'han heretat i hem afegit l'atribut *idCiutat* que és una clau forana que referencia a la relació CIUTAT. Aquesta clau és derivada de l'interrelació LOCALITZACIO.

CIUTAT(idCiutat, nom, idRegio)

on {idRegio} referencia REGIO.

Hem afegit l'atribut *idRegio* a partir de la interrelació SITUACIO segons l'esquema de la Figura 1, la resta d'atributs pertanyen a l'entitat CIUTAT.

REGIO(idRegio, nom).

PRODUCTE(ean13, nom, descripcio, dataIn, dataOut, preu, benefici).

La relació VENDES no deriva directament de cap entitat del model E/R sinó que ho fa de la interrelació VENDA que en ser de segon grau i connectivitat M:N dona lloc a una relació que té com a clau primària tots els atributs que formen les claus primàries de les relacions amb que està connectada a més de la data i l'hora com ja hem comentat. Apart hi hem afegit l'atribut *unitats*, que és un atribut de la interrelació i els atributs *preuTotal* i *beneficiTotal* que seran calculats a partir de les dades del producte i les unitats.

VENDES(idBotiga, ean13, hora, dia, mes, any, unitats, preuTotal, beneficiTotal)

on {idBotiga} referencia BOTIGA

on {ean13} referencia PRODUCTE

on {hora, dia, mes, any} referencia DATAIHORA.

2 Hem mantingut la nomenclatura feta servir al material didàctic de la UOC de l'assignatura Bases de Dades I per a les relacions del model relacional, que no s'han de confondre amb les interrelacions del model E/R.

Pel que fa al mòdul estadístic tenim la relació següent on hem mantingut tots els atributs presentats en l'apartat anterior i que podem veure a la Figura 9. L'atribut `anny` és la clau primària de la relacions.

ESTADISTICA(anny, benefici, idBotiga, b_idBotiga, ean13, u_ean13, hora_m, u_hora_m, hora_l, u_hora_l, dia_m, u_dia_m, dia_l, u_dia_l, idCiutat, b_idCiutat, perc_virt).

Hem de recordar que aquesta relació és calculada a partir de la relació VENDES i que s'actualitzarà cada cop que aquesta afegeixi una nova entrada.

7.4. Disseny físic.

A partir d'aquí ja tenim tots els components per envestir el disseny físic de la base de dades, es a dir implementar totes les relacions del model lògic com a taules amb columnes per als atributs i els tipus d'aquests definint claus primàries i foranes.

El primer que farem serà definir un usuari en la base de dades Oracle Database 11g Express Edition Release 2 instal·lada sobre Windows 10 Pro 64 bits. El fitxer .sql per executar de de la línia de comandes és el següent (BD.SQL):

```
DROP USER CADENA CASCADE;
CREATE USER CADENA
IDENTIFIED BY cadenaderoba
DEFAULT TABLESPACE USERS
QUOTA UNLIMITED ON USERS
TEMPORARY TABLESPACE TEMP;
GRANT CONNECT, RESOURCE TO CADENA;
```

La definició de les taules la presentam al següent apartat on implementam tota la BD.

Hem de tenir en compte que la taula VENDES apart dels camps que hem obtingut a les etapes anteriors té dos camps calculats que són PREUTOTAL i BENEFICITOTAL que podem calcular a partir de les unitats i dels camps corresponents PREU i BENEFICI de la taula PRODUCTE i del camp UNITATS de la mateixa taula. A l'hora d'introduir-los el procediment d'entrada calcularà els camps.

La taula ESTADISITICA és tota ella calculada a partir de la taula VENDES.

8. Implementació.

En aquest document presentam la implementació de la BD per a l'empresa **Cadena de Roba** que hem dissenyat en l'apartat anterior.

En aquesta fase partim del disseny físic amb que hem acabat l'anterior fase de desenvolupament per a implementar les taules.

Com a segon pas implementarem les consultes necessàries per a utilitzar la BD a l'hora d'entrar, modificar o eliminar dades en procediments que poden ser cridats des de qualsevol gestor. Per a la implementació d'aquests haurem de definir algunes funcions auxiliars que ens facilitaran la gestió d'errors i les comprovacions necessàries.

En tercer lloc definirem les tres consultes demanades com a requisits de l'aplicació que ens permeten consultar dades de la taula de vendes, que és la principal raó de l'aplicació.

I per acabar amb aquesta part farem la implementació del mòdul estadístic que en actualitzar-se en temps real, haurà de modificar la taula ESTADISTICA.

8.1. Taules.

TAULES_CADENA.sql

A la fase de disseny físic hem presentat la implementació de les taules de dades que contempla el disseny de la BD. Presentam la implementació d'aquestes al fitxer adjunt TAULES_CADENA.sql, amb els camps i els atributs que ja donarem en el disseny físic, i són les següents:

TAULES_CADENA.sql	Descripció.
REGIO	Clau primària IDREGIO.
CIUTAT	Clau primària IDCIUTAT i clau forana IDREGIO.
BOTIGA	Clau primària IDBOTIGA i clau forana IDCIUTAT. Els camps FRANQUICIA i EBOTIGA són booleans, i en l'Oracle Express els simulam amb un nombre enter que pot prendre valors 0 i 1, per tant hem afegit aquesta restricció a la definició.
PRODUCTE	Clau primària EAN13.

VENDES	Clau primària IDBOTIGA, EAN13, HORA, DIA, MES i ANNY i clau forana IDBOTIGA i EAN13.
--------	--

Les restriccions comentades de claus primàries i claus foranes les comprovarem en els procediments d'alta, baixa i modificació de dades que anirem definint i en farem la captura i tractament dels errors o excepcions si es produeixen.

TAULA_LOG.sql

Aquesta taula TAULA_LOG no estava definida a la fase de disseny ja que no guardarà dades pròpies de la cadena i de la seva activitat sinó que guardarà un registre de totes les operacions que es duran a terme sobre la BD.

Cada cop que s'executi un procediment escriurem a la taula la data i l'hora en que s'ha executat, la identificació del procediment, els paràmetres d'entrada donats al procediment, la sortida que dona el procediment en executar-se, que pot ser 'OK' si l'execució ha anat bé o 'ERROR' més la descripció d'aquest error segons el tractament que haguem fet de l'excepció. A diferència de les taules de dades anteriors, la clau primària no la utilitzarem en cap altre taula ni la referenciem, per tant hem fet que aquesta sigui un camp numèric que es generarà automàticament mitjançant una seqüència TAULA_LOG_SEQ i llançada per un trigger, TAULA_LOG_TRIG, que definim en el mateix fitxer.

TAULA_LOG.sql	Descripció
TAULA_LOG	Taula. Clau primària ID generada per la seqüència TAULA_LOG_SEQ. Els camps són ID, DATA_I_HORA, PROC, PARAM_IN i PARAM_OUT.
TAULA_LOG_SEQ	Seqüència que dona valors a la clau primària, ID, de la taula TAULA_LOG.
TAULA_LOG_TRIG	Trigger que llança la seqüència TAULA_LOG_SEQ si en haver d'introduir un registre aquest no té valor de la clau primària ID.

8.2. Procediment de registre de la taula TAULA_LOG: ESCRIU.sql

En el fitxer ESCRIU.sql presentam el procediment del mateix nom que fem servir per a escriure en la taula de registre les sortides de la resta de procediments, bé per indicar l'èxit o l'error. Simplement insereix quatre cadenes de caràcters a la taula.

PROCEDURE ESCRIU(D IN VARCHAR, P IN VARCHAR, PIN IN VARCHAR, POUT IN VARCHAR)
Funció
Insereix en la taula TAULA_LOG els quatre camps passats com a paràmetres.
Paràmetres d'entrada
D: cadena de text P: cadena de text PIN: cadena de text POUT: cadena de text
Sortida
Insereix a la taula TAULA_LOG el registre passat com a paràmetre.

8.3. Procediments d'Alta, Baixa i Modificació de dades:ABM_CADENA.sql.

En el fitxer ABM_CADENA.sql presentam el package del mateix nom que consta del procediments d'alta, baixa i modificació de dades de les taules REGIO, CIUTAT, BOTIGA i PRODUCTE. De les dues primeres taules tenim només procediments d'alta i de les dues darreres d'alta, baixa i modificació. Els presentam a continuació així com les funcions auxiliars HI_ES_REGIO(), HI_ES_CIUTAT, HI_ES_BOTIGA i HI_ES_PRODUCTE que utilitzam per fer el tractament d'errors i comprovar si les claus primàries estan utilitzades.

Hem introduït en el fitxer la creació del fitxer CADENA.LOG per a guardar l'execució i poder comprovar errors en la compilació si es produïssin.

Les funcions i procediments del paquet són:

PROCEDURE ALTA_REGIO (IDREGIO IN NUMBER, NOM IN VARCHAR2)
Funció
Insereix en la taula REGIO els dos camps passats com a paràmetres. Comprova si la clau primària existeix a la taula.
Paràmetres d'entrada
IDREGIO (PK): nombre enter positiu NOM: cadena de text
Sortida
Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error de duplictat de la clau primària o d'un error inesperat.

PROCEDURE ALTA_CIUTAT (IDCIUTAT IN NUMBER, NOM IN VARCHAR2, IDREGIO IN NUMBER)
Funció
Insereix en la taula CIUTAT els tres camps passats com a paràmetres. Comprova si la clau primària existeix a la taula i si la clau forana IDREGIO existeix a la taula corresponent.
Paràmetres d'entrada
IDCIUTAT (PK): nombre enter positiu NOM: cadena de text IDREGIO: nombre enter positiu
Sortida
Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error de duplicat de la clau primària, d'inexistència de la clau forana o d'un error inesperat.

PROCEDURE ALTA_BOTIGA (IDBOTIGA IN NUMBER, NOM IN VARCHAR2, EMAIL IN VARCHAR2, NTREBALLADORS IN NUMBER, FRANQUICIA IN NUMBER, EBOTIGA IN NUMBER, IDCIUTAT IN NUMBER)
Funció
Insereix en la taula BOTIGA els set camps passats com a paràmetres. Comprova si la clau primària existeix a la taula i si la clau forana IDCIUTAT existeix a la taula corresponent.
Paràmetres d'entrada
IDBOTIGA (PK): nombre enter positiu NOM: cadena de text EMAIL: cadena de text NTREBALLADORS: nombre enter estrictament positiu FRANQUICIA: nombre enter, 0 o 1, que simula una variable booleana (1=franquícia) EBOTIGA: nombre enter, 0 o 1, que simula si una variable booleana (1=ebotiga) IDCIUTAT: nombre enter positiu
Sortida
Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error de duplicat de la clau primària, d'inexistència de la clau forana o d'un error inesperat.

En aquest cas no hem tractat l'error del tipus de botiga o del nombre de treballadors, ja que suposam que les entrades es faran només una vegada per botiga i les farà el personal de manteniment de l'aplicació sota control. De totes formes la taula té definida la restricció.

PROCEDURE BAIXA_BOTIGA (IDBOTIGA IN NUMBER)
Funció
Esborra de la taula BOTIGA el registre amb clau primària passat com a paràmetre. Comprova si la clau primària existeix a la taula.
Paràmetres d'entrada
IDBOTIGA (PK): nombre enter positiu
Sortida
Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error d'inexistència de la clau primària o d'un error inesperat.

PROCEDURE MODIFICA_BOTIGA (IDBOTIGA IN NUMBER, NOM IN VARCHAR2, EMAIL IN VARCHAR2, NTREBALLADORS IN NUMBER, FRANQUICIA IN NUMBER, EBOTIGA IN NUMBER, IDCIUTAT IN NUMBER);
Funció
Modifica en la taula REGIO els camps passats com a paràmetres del registre identificat com a clau primària. Comprova si la clau primària existeix a la taula i només canvia els valors que no siguin NULL. (L'únic valor no acceptat com a NULL és la clau primària.)
Paràmetres d'entrada
IDBOTIGA (PK): nombre enter positiu NOM: cadena de text EMAIL: cadena de text NTREBALLADORS: nombre enter estrictament positiu FRANQUICIA: nombre enter, 0 o 1, que simula una variable booleana (1=franquícia) EBOTIGA: nombre enter, 0 o 1, que simula si una variable booleana (1=ebotiga) IDCIUTAT: nombre enter positiu
Sortida
Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error d'inexistència de la clau primària o d'un error inesperat.

Igual com hem comentat en el procediment d'alta de les botigues no hem contemplat l'error en el camp de franquícia, botiga virtual i nombre de treballadors.

PROCEDURE ALTA_PRODUCTE (EAN13 IN NUMBER, NOM IN VARCHAR2, DESCRIPCIO IN VARCHAR2, DATAIN IN DATE, DATAOUT IN DATE, PREU IN NUMBER, BENEFICI IN NUMBER)

Funció

Insereix en la taula PRODUCTE els set camps passats com a paràmetres. Comprova si la clau primària existeix a la taula.

Paràmetres d'entrada

EAN13 (PK): nombre enter positiu

NOM: cadena de text

DESCRIPCIO: cadena de text

DATAIN: data (format XX / XX / XXXX)

DATAOUT: data (format XX / XX / XXXX) . Aquest valor pot ser NULL

PREU: nombre real amb dos decimals

BENEFICI: nombre real amb dos decimals

Sortida

Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error de duplicat de la clau primària o d'un error inesperat.

No comprovam que el valor de DATAOUT, en cas d'introduir-se, sigui major que el de DATAIN pels mateixos motius citats en el tractament de botigues.

PROCEDURE BAIXA_PRODUCTE (EAN13 IN NUMBER)

Funció

Esborra de la taula PRODUCTE el registre que coincideixi amb el camp passat com a paràmetre. Comprova si la clau primària existeix a la taula.

Paràmetres d'entrada

EAN13 (PK): nombre enter positiu

Sortida

Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error d'inexistència la clau primària o d'un error inesperat.

PROCEDURE MODIFICA_PRODUCTE (EAN13 IN NUMBER, NOM IN VARCHAR2, DESCRIPCIO IN VARCHAR2, DATAIN IN DATE, DATAOUT IN DATE, PREU IN NUMBER, BENEFICI IN NUMBER)

Funció

Modifica en la taula PRODUCTE els camps passats com a paràmetres del registre identificat com a clau primària. Comprova si la clau primària existeix a la taula i només canvia els valors que no siguin NULL. (L'únic valor no acceptat com a NULL és la clau primària.)

Paràmetres d'entrada

EAN13 (PK): nombre enter positiu

NOM: cadena de text

DESCRIPCIO: cadena de text

DATAIN: data (format XX / XX / XXXX)

DATAOUT: data (format XX / XX / XXXX)

PREU: nombre real amb dos decimals

BENEFICI: nombre real amb dos decimals

Sortida

Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error d'inexistència de la clau primària o d'un error inesperat.

FUNCTION HI_ES_REGIO (codi IN NUMBER) RETURN NUMBER**Funció**

Comprova si l'identificador passat com a paràmetre és a la taula REGIO com a clau primària.

Paràmetres d'entrada

CODI: nombre enter positiu

Sortida

Retorna el codi de la regió, si aquesta existeix o bé -1 si no existeix. En cas de retornar -1, com que la consulta feta retorna buida, ho fem amb la captura de l'error NO_DATA_FOUND.

FUNCTION HI_ES_CIUTAT (codi IN NUMBER) RETURN NUMBER**Funció**

Comprova si l'identificador passat com a paràmetre és a la taula CIUTAT com a clau primària.

Paràmetres d'entrada

CODI: nombre enter positiu

Sortida

Retorna el codi de la ciutat, si aquesta existeix o bé -1 si no existeix. En cas de retornar -1, com que la consulta feta retorna buida, ho fem amb la captura de l'error NO_DATA_FOUND.

FUNCTION HI_ES_BOTIGA(codi IN NUMBER) RETURN NUMBER**Funció**

Comprova si l'identificador passat com a paràmetre és a la taula BOTIGA com a clau primària.

Paràmetres d'entrada

codi: nombre enter positiu

Sortida

Retorna el codi de la botiga, si aquesta existeix o bé -1 si no existeix. En cas de retornar -1, com que la consulta feta retorna buida, ho fem amb la captura de l'error NO_DATA_FOUND.

FUNCTION HI_ES_PRODUCTE(codi IN NUMBER) RETURN NUMBER**Funció**

Comprova si l'identificador passat com a paràmetre és a la taula PRODUCTE com a clau primària.

Paràmetres d'entrada

CODI: nombre enter positiu

Sortida

Retorna el codi del producte, si aquest existeix o bé -1 si no existeix. En cas de retornar -1, com que la consulta feta retorna buida, ho fem amb la captura de l'error NO_DATA_FOUND.

8.4. Procediments d'Alta, Baixa i Modificació de vendes:ABM_VENDES.sql.

En el fitxer ABM_VENDES.sql presentam el package del mateix nom que consta del procediments d'alta, baixa i modificació de vendes que guardam en la taula de fets VENDES. Els presentam a continuació així com la funció auxiliar HI_ES_VENDA(), que utilitzam per fer el tractament d'errors i comprovar si les claus primàries estan utilitzades.

Hem introduït en el fitxer la creació del fitxer VENDES.LOG per a guardar l'execució i poder comprovar errors en la compilació si es produïssin.

La funció i procediments del paquet són:

PROCEDURE ALTA_VENDA(IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER, UNITATS IN NUMBER)
Funció
Insereix en la taula VENDES els camps passats com a paràmetres. Comprova si la clau primària existeix a la taula així com si existeixen les claus IDBOTIGA i EAN13. A l'hora de comprovar la clau primària, formada pels sis primers camps, utilitzam la funció HI_ES_VENDA que retorna -1 si no existeix o bé el codi IDBOTIGA, que és un nombre positiu si ja existeix la venda.
Paràmetres d'entrada
IDBOTIGA (PK): nombre enter positiu EAN13 (PK): nombre enter positiu (format XXXXXXXXXXXXX) HORA (PK): nombre enter positiu (format XX) DIA (PK): nombre enter positiu (format XX) MES (PK): nombre enter positiu (format XX) ANNY (PK): nombre enter positiu (format XXXX) UNITATS: nombre enter positiu
Sortida
Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error de duplicat de la clau primària, de la inexistència dels codis IDBOTIGA o EAN13, o d'un error inesperat.

En aquest procediment no hem tractat l'error en la possible introducció de les dates o bé del nombre d'unitats ja que consideram que aquest l'ha de tractar el gestor que crida a la BD. Si que comprovam que la clau primària no existeix i que el producte i la botiga estan donats d'alta.

PROCEDURE BAIXA_VENDA(IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER)
Funció
Esborra de la taula VENDES el registre que coincideixi amb els camps passats com a paràmetres. Comprova si la clau primària existeix a la taula. (Ho fa igual que el procediment d'alta.)
Paràmetres d'entrada
IDBOTIGA (PK): nombre enter positiu EAN13 (PK): nombre enter positiu (format XXXXXXXXXXXXX) HORA (PK): nombre enter positiu (format XX) DIA (PK): nombre enter positiu (format XX)

MES (PK): nombre enter positiu (format XX)

ANNY (PK): nombre enter positiu (format XXXX)

Sortida

Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error d'inexistència de la clau primària, o d'un error inesperat.

En aquesta execució comprovam que la clau primària existeix per a poder donar de baixa la venda.

PROCEDURE MODIFICA_VENDA(IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER, UNITATS IN NUMBER)

Funció

Modifica en la taula VENDES els camps passats com a paràmetres del registre identificat com a clau primària. Comprova si la clau primària existeix a la taula i només canvia els valors que no siguin NULL. (L'únic valor no acceptat com a NULL és els dels camps de la clau primària.)

Paràmetres d'entrada

IDBOTIGA (PK): nombre enter positiu

EAN13 (PK): nombre enter positiu (format XXXXXXXXXXXXXXX)

HORA (PK): nombre enter positiu (format XX)

DIA (PK): nombre enter positiu (format XX)

MES (PK): nombre enter positiu (format XX)

ANNY (PK): nombre enter positiu (format XXXX)

UNITATS: nombre enter positiu

Sortida

Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant si es tracta d'un error d'inexistència de la clau primària o d'un error inesperat.

En aquest procediment comprovam que la clau primària hi és abans de fer-ne la modificació.

FUNCTION HI_ES_VENDA (IDBOTIG IN NUMBER, EAN IN NUMBER, HOR IN NUMBER, DI IN NUMBER, ME IN NUMBER, ANN IN NUMBER) RETURN NUMBER

Funció

Comprova si l'identificador passat com a paràmetre és a la taula VENDES com a clau primària.

Paràmetres d'entrada

IDBOTIGA (PK): nombre enter positiu

EAN13 (PK): nombre enter positiu (format XXXXXXXXXXXXX)

HORA (PK): nombre enter positiu (format XX)

DIA (PK): nombre enter positiu (format XX)

MES (PK): nombre enter positiu (format XX)

ANNY (PK): nombre enter positiu (format XXXX)

Sortida

Retorna el codi de la botiga, si la clau primària existeix o bé -1 si no existeix. En cas de retornar -1, com que la consulta feta retorna buida, ho fem amb la captura de l'error NO_DATA_FOUND.

En aquest hem de fer notar que el codi de la botiga, IDBOTIGA, que retorna la funció, no identifica la venda, ja que la clau primària està formada per sis camps, però en el retorn de la funció només ens interessa saber si aquest registre és o no a la taula i per tant com que amb un IDBOTIGA que serà positiu o -1 en cas d'inexistència és suficient.

8.5. Procediments d'execució de consultes:CONSULTA_VENDES.sql.

En el fitxer CONSULTA_VENDES.sql presentam el package del mateix nom que consta del procediments que executen les tres consultes demanades com a requisit de l'aplicació, (CONSULTA1, CONSULTA2 i CONSULTA3), així com cinc funcions auxiliars que necessitam per a dur-les a terme (BENEFICI_CADENA, BOTIGA_MES_BENEFICI, BOTIGA_MES_UNITATS, PROD_MES_VENUT i U_BOTIGA_MES_UNITATS).

Hem introduït en el fitxer la creació del fitxer CONSULTA1.LOG per a guardar l'execució i poder comprovar errors en la compilació si es produïssin.

La funció i procediments del paquet són:

PROCEDURE CONSULTA1(A in number, M in number)

Funció

Donats com a paràmetres un any i un mes la consulta retorna per aquest any i mes:

- El codi de totes les botigues de la cadena
- El nombre d'unitats venudes per la botiga
- El nombre de productes diferents venuts per la botiga
- El benefici total de la botiga
- El percentatge del benefici de la botiga respecte al de tota la cadena
- El percentatge de la botiga dividit pel nombre de treballadors.

La consulta utilitza la funció BENEFICI_CADENA que crida amb els paràmetres necessaris.

Paràmetres d'entrada

A: nombre enter (format XXXX)

M: nombre enter (format XX)

Sortida

Dóna la sortida de la consulta per la sortida per defecte.

Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant un error inesperat.

PROCEDURE CONSULTA2(A in number, M in number)

Funció

Donats com a paràmetres un any i un mes la consulta retorna per aquest any i mes:

- El codi de tots els productes del catàleg
- El nom del producte
- El nombre d'unitats venudes del producte
- El benefici que ha suposat per cada producte
- La botiga amb més unitats venudes del producte
- El nombre d'aquestes unitats venudes per la botiga.

La consulta utilitza les funcions BOTIGA_MES_UNITATS i U_BOTIGA_MES_UNITATS que crida amb els paràmetres necessaris.

Paràmetres d'entrada

A: nombre enter (format XXXX)

M: nombre enter (format XX)

Sortida

Dóna la sortida de la consulta per la sortida per defecte.

Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant un error inesperat.

PROCEDURE CONSULTA3(A in number, M in number)

Funció

Donats com a paràmetres un any i un mes la consulta retorna per aquest any i mes:

- Tots els dies del mes en que s'han produït vendes
- El benefici total d'aquest dia obtingut per la cadena
- El producte amb més unitats venudes
- El nombre d'unitats venudes d'aquest producte
- La botiga amb més beneficis aquest dia

<ul style="list-style-type: none"> El benefici obtingut per aquesta botiga. <p>La consulta utilitza les funcions PRODUCTE_MES_VENUT i BOTIGA_MES_BERNEFICI que crida amb els paràmetres necessaris.</p>
Paràmetres d'entrada
<p>A: nombre enter (format XXXX)</p> <p>M: nombre enter (format XX)</p>
Sortida
<p>Dóna la sortida de la consulta per la sortida per defecte.</p> <p>Registra a la taula TAULA_LOG el resultat del procediment com a OK o ERROR especificant un error inesperat.</p>

FUNCTION BENEFICI_CADENA (ANNY IN NUMBER, MES IN NUMBER) RETURN NUMBER
Funció
<p>Rebuta com a paràmetres el valor d'un any i un mes, retorna el valor del benefici de la cadena en aquests. En cas que en l'any i mes donats no hi hagi benefici retorna un -1. Per fer-ho utilitzam el tractament de l'excepció NO_DATA_FOUND.</p>
Paràmetres d'entrada
<p>ANNY: nombre enter (format XXXX)</p> <p>MES: nombre enter (format XX)</p>
Sortida
<p>Retorna el valor del benefici si en té o bé -1 si no hi ha registres en aquest any i mes.</p>

FUNCTION BOTIGA_MES_BENEFICI (ANNY IN NUMBER, MES IN NUMBER, DIA IN NUMBER) RETURN NUMBER
Funció
<p>Rebuta com a paràmetres el valor d'un any, un mes i un dia, retorna el codi de la botiga amb més benefici en aquest interval. En cas que en l'any, mes i dia donats no hi hagi vendes retorna un -1. Per fer-ho utilitzam el tractament de l'excepció NO_DATA_FOUND.</p>
Paràmetres d'entrada
<p>ANNY: nombre enter (format XXXX)</p> <p>MES: nombre enter (format XX)</p> <p>DIA: nombre enter (format XX)</p>

Sortida

Retorna el valor del codi de la botiga o bé -1 si no hi ha registres en aquest any, mes i dia.

FUNCTION BOTIGA_MES_UNITATS (ANNY IN NUMBER, MES IN NUMBER, EAN13 IN NUMBER) RETURN NUMBER**Funció**

Rebuta com a paràmetres el valor d'un any, un mes i el codi d'un producte, retorna el codi de la botiga amb més unitats venudes d'aquest producte en aquest interval. En cas que en l'any i mes donats no hi hagi vendes d'aquest producte retorna un -1. Per fer-ho utilitzam el tractament de l'excepció NO_DATA_FOUND.

Paràmetres d'entrada

ANNY: nombre enter (format XXXX)

MES: nombre enter (format XX)

EAN13: nombre enter (format XXXXXXXXXXXXX)

Sortida

Retorna el valor del codi de la botiga o bé -1 si no hi ha registres en aquest any i mes per aquest producte.

FUNCTION PROD_MES_VENUT (ANNY IN NUMBER, MES IN NUMBER, DIA IN NUMBER) RETURN NUMBER**Funció**

Rebuta com a paràmetres el valor d'un any, un mes i un dia, retorna el codi del producte amb més unitats venudes en aquest interval. En cas que en l'any, mes i dia donats no hi hagi vendes retorna un -1. Per fer-ho utilitzam el tractament de l'excepció NO_DATA_FOUND.

Paràmetres d'entrada

ANNY: nombre enter (format XXXX)

MES: nombre enter (format XX)

DIA: nombre enter (format XX)

Sortida

Retorna el valor del codi del producte o -1 si no hi ha registres en aquest any, mes i dia.

FUNCTION U_BOTIGA_MES_UNITATS (ANNY IN NUMBER, MES IN NUMBER, EAN13 IN NUMBER) RETURN NUMBER

Funció
Rebut com a paràmetres el valor d'un any, un mes i un codi de producte, retorna el codi de la botiga amb més unitats venudes en aquest interval d'aquest producte. En cas que en l'any i mes donats no hi hagi vendes del producte retorna un -1. Per fer-ho utilitzam el tractament de l'excepció NO_DATA_FOUND.
Paràmetres d'entrada
ANNY: nombre enter (format XXXX) MES: nombre enter (format XX) EAN13: nombre enter (format XXXXXXXXXXXXXXX)
Sortida
Retorna el valor del codi de la botiga o -1 si no hi ha registres en aquest any i mes per aquest producte.

En les funcions presentades, en trobar un cas NO_DATA_FOUND hem optat per donar una sortida -1 per a identificar l'error ja que podria donar-se el cas de tenir un benefici zero de la cadena si venguéssim a preu de cost. (Suposam que no tenim beneficis negatius). Pel que fa als codis de botigues i productes aquests són sempre enters positius i per tant el valor -1 indicarà també un codi no trobat.

8.6. Procediments d'execució del mòdul estadístic: ACT_ESTADISTICA.sql i CONSULTA.sql.

En el fitxer ACT_ESTADISTICA.sql presentam el package del mateix nom on es defineix el procediment ACT_ALTA_VENDA que actualitza la taula ESTADISTICA. Aquest procediment és cridat pels procediments d'alta, baixa i modificació de les vendes cada cop que aquests s'executen. També hi definim set funcions auxiliars que realitzen càlculs intermedis per a actualitzar els registres de la taula a fi de clarificar-ne, estructurar-ne i simplificar-ne el disseny.

A l'hora de definir el procediment ACT_ALTA_VENDA, que simplement actualitza el registre corresponent a l'any on es produeix la venda, hauríem d'haver previst que la primera venda de l'any no trobarà registre i per tant no el pot actualitzar sinó que l'ha de definir. Introduir aquesta condició que s'executaria cada cop que es fes una venda, allargaria molt el temps de càlcul i com que només es fa un cop per any, hem creat un fitxer, INICI_ESTADISTICA.sql que introdueix els anys en que treballarà l'aplicació a la taula ESTADISTICA i així resollem el problema.

La funció i procediments del paquet són:

FUNCTION BOTIGA_ANNY (ANNY IN NUMBER) RETURN NUMBER
Funció
Rebut com a paràmetre el valor d'un any, retorna el codi de la botiga amb més benefici en aquest.

Paràmetres d'entrada
ANNY: nombre enter (format XXXX)
Sortida
Retorna el valor del codi de la botiga.

FUNCTION PROD_ANNY (ANNY IN NUMBER) RETURN NUMBER
Funció
Rebut com a paràmetre el valor d'un any, retorna el codi del producte amb més unitats venudes en aquest.
Paràmetres d'entrada
ANNY: nombre enter (format XXXX)
Sortida
Retorna el valor del codi del producte.

FUNCTION CIUTAT_ANNY (ANNY IN NUMBER) RETURN NUMBER
Funció
Rebut com a paràmetre el valor d'un any, retorna el codi de la ciutat amb més unitats venudes en aquest.
Paràmetres d'entrada
ANNY: nombre enter (format XXXX)
Sortida
Retorna el valor del codi de la ciutat.

FUNCTION DIA_M_ANNY (ANNY IN NUMBER) RETURN NUMBER
Funció
Rebut com a paràmetre el valor d'un any, retorna el dia del mes amb més unitats venudes en aquest.
Paràmetres d'entrada
ANNY: nombre enter (format XXXX)

Sortida

Retorna el valor del codi del mes.

FUNCTION DIA_L_ANNY (ANNY IN NUMBER) RETURN NUMBER**Funció**

Rebut com a paràmetre el valor d'un any, retorna el dia del mes amb menys unitats venudes en aquest.

Paràmetres d'entrada

ANNY: nombre enter (format XXXX)

Sortida

Retorna el valor del codi del mes.

FUNCTION HORA_M_ANNY (ANNY IN NUMBER) RETURN NUMBER**Funció**

Rebut com a paràmetre el valor d'un any, retorna l'hora del dia amb més unitats venudes en aquest.

Paràmetres d'entrada

ANNY: nombre enter (format XXXX)

Sortida

Retorna el valor de l'hora del dia.

FUNCTION HORA_L_ANNY (ANNY IN NUMBER) RETURN NUMBER**Funció**

Rebut com a paràmetre el valor d'un any, retorna l'hora del dia amb menys unitats venudes en aquest.

Paràmetres d'entrada

ANNY: nombre enter (format XXXX)

Sortida

Retorna el valor de l'hora del dia.

PROCEDURE ACT_ALTA_VENDA(ANNY IN NUMBER)
Funció
Rebut com a paràmetre el valor d'un any, actualitza el registre de la taula ESTADISTICA amb corresponent a aquest.
Paràmetres d'entrada
ANNY: nombre enter (format XXXX)
Sortida
Actualitza el registre de l'any a la taula ESTADISTICA.

En el fitxer CONSULTA.sql presentam el package del mateix nom que consta del procediments que executen les dues consultes demanades com a requisit del mòdul estadístic, (CONSULTA, CONSULTA_ANY).

Hem introduït en el fitxer la creació del fitxer CONSULTA2.LOG per a guardar l'execució i poder comprovar errors en la compilació si es produïssin.

Els procediments del paquet són:

PROCEDURE CONSULTA(A in number)
Funció
Rebut com a paràmetre el valor d'un any, retorna tots els camps de la taula ESTADISTICA que corresponen a aquest registre.
Paràmetres d'entrada
ANNY: nombre enter (format XXXX)
Sortida
Retorna el registre cercat.

PROCEDURE CONSULTA_ANY;
Funció
Retorna tots els registres de la taula ESTADISTICA (amb tots els seus camps) ordenats per any en forma ascendent.
Paràmetres d'entrada
NULL

[Sortida](#)

Retorna tots els registres de la taula ESTADISTICA.

9. Test.

A continuació presentam el test preparat per a l'aplicació on hem comprovat les seves funcionalitats per etapes, en el mateix ordre en que l'utilitzaríem o instal·laríem en un ús normal.

9.1. Test dels procediments del package abm_cadena.

En aquest paquet hem de testar els procediments d'alta per les taules REGIO, CIUTAT, BOTIGA i PRODUCTE així com els de baixa i modificació de BOTIGA i CIUTAT. A l'hora també comprovarem les funcions i disparadors i els mètodes de tractament d'excepcions que utilitzen.

Presentam el joc de dades al fitxer TEST_DADES_CADENA.sql on hem preparat una entrada de dades i per cada una de les taules així com l'intent d'introduir una dada amb la clau primària repetida, una baixa pels dos procediments que ho fan, una modificació pels dos procediments que ho fan i l'intent de donar de baixa i de modificar una botiga inexistent. Així provarem els procediments següents:

```
PROCEDURE ALTA_REGIO ( IDREGIO IN NUMBER, NOM IN VARCHAR2 );  
PROCEDURE ALTA_CIUTAT ( IDCIUTAT IN NUMBER, NOM IN VARCHAR2, IDREGIO IN NUMBER );  
PROCEDURE ALTA_BOTIGA ( IDBOTIGA IN NUMBER, NOM IN VARCHAR2, EMAIL IN VARCHAR2, NTREBALLADORS IN  
NUMBER, FRANQUICIA IN NUMBER, EBOTIGA IN NUMBER, IDCIUTAT IN NUMBER );  
PROCEDURE BAIXA_BOTIGA ( IDBOTIGA IN NUMBER );  
PROCEDURE MODIFICA_BOTIGA ( IDBOTIGA IN NUMBER, NOM IN VARCHAR2, EMAIL IN VARCHAR2, NTREBALLADORS IN  
NUMBER, FRANQUICIA IN NUMBER, EBOTIGA IN NUMBER, IDCIUTAT IN NUMBER );  
PROCEDURE ALTA_PRODUCTE ( EAN13 IN NUMBER, NOM IN VARCHAR2, DESCRIPCIO IN VARCHAR2, DATAIN IN DATE,  
DATAOUT IN DATE, PREU IN NUMBER, BENEFICI IN NUMBER );  
PROCEDURE BAIXA_PRODUCTE ( EAN13 IN NUMBER );  
PROCEDURE MODIFICA_PRODUCTE ( EAN13 IN NUMBER, NOM IN VARCHAR2, DESCRIPCIO IN VARCHAR2, DATAIN IN  
DATE, DATAOUT IN DATE, PREU IN NUMBER, BENEFICI IN NUMBER );
```

A la taula de registre TAULA_LOG podem comprovar que totes les execucions s'han produït així com esperàvem i a la vegada podem llistar els continguts de les taules i comprovar que tenen els elements esperats, 10 regions, 29 ciutats, 53 botiges i 314 productes. També podem comprovar que la taula LOG té 418 registres que corresponen a les 406 entrades i a les 12 execucions errònies, d'esborrat i modificacions fetes.

9.2. Test dels procediments del package abm_vendes.

En aquest paquet hem de testar els procediments d'alta, de baixa i modificació de la taula VENDES. A l'hora també comprovarem les funcions i disparadors i els mètodes de tractament d'excepcions que utilitzen.

A la vegada que modifiquem la taula VENDES també es modificarà la taula ESTADISTICA, que s'actualitza a temps real amb les vendes, per tant aquesta taula ha d'estar preparada. Com hem comentat a l'apartat d'implementació aquesta taula s'inicialitza amb el fitxer INICI_ESTADISTICA.sql que s'ha d'executar en

primer lloc.

Presentam el joc de dades al fitxer TEST_DADES_VENDA.sql on hem preparat una entrada de dades per la taula així com l'intent de donar de baixa una venda que no existeix, un d'introduir una dada amb la clau primària repetida, una baixa, una altra alta equivocada i una modificació d'aquesta. Així provarem els procediments següents:

```
PROCEDURE ALTA_VENDA ( IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER, UNITATS IN NUMBER );
```

```
PROCEDURE BAIXA_VENDA ( IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER );
```

```
PROCEDURE MODIFICA_VENDA ( IDBOTIGA IN NUMBER, EAN13 IN NUMBER, HORA IN NUMBER, DIA IN NUMBER, MES IN NUMBER, ANNY IN NUMBER, UNITATS IN NUMBER );
```

A la taula de registre TAULA_LOG podem comprovar que totes les execucions s'han produït així com esperàvem i a la vegada podem llistar els continguts de la taula i comprovar que té els elements esperats, 21 vendes. També podem comprovar que la taula LOG té 26 registres (ja que l'haviem buidada per controlar millor el procés), que corresponen a les 21 entrades i a les 5 execucions errònies, d'esborrat i modificacions fetes.

9.3. Test dels procediments del package act_estadistica.

El paquet act_estadistica és el responsable d'actualitzar la taula ESTADISTICA, que conté les dades del modul estadístic i que s'actualitza cada cop que fem una entrada. Per fer-ho executa el procediment PROCEDURE ACT_ALTA_VENDA(ANNY IN NUMBER) que torna a calcular tots els camps cada cop que es produeix un canvi a la taula VENDES.

Així amb el joc de proves que hem executat a l'apartat 9.2, la taula ESTADISTICA ha d'haver actualitzat les dades de les vendes de l'any 2010, que són les úniques que hem entrat i que podem comprovar que són correctes. A la taula observam que només s'ha actualitzat el registre que correspon a 2010 i que els valors calculats són correctes. (En tenir només 21 vendes és fàcil fer la comprovació de tots els camps). A l'annex 1 tenim la taula de vendes amb les ordenacions per botiga, producte, hora i dia i els càlculs de benefici i unitats requerits per comprovar que els resultats coincideixen amb els de la taula ESTADISTICA.

ANY	BENEFICI	BOTIGA	BENEFICI	EAN13	UNITATS	HORA +	UNITATS
2010	2785,4	3	2114,55	9111222000292	46	14	48

HORA -	UNITATS	DIA +	UNITATS	DIA -	UNITATS	CIUTAT	BENEFICI	%
9	3	15	68	10	1	155	2114,55	24,08

Els càlculs per a la ciutat, que no es veuen a l'annex són senzills perquè les dues botigues són de ciutats diferents, per tant el càlcul també el podem veure a la taula on hem ordenat les vendes per botigues i la

botiga 3 està a la ciutat 155. El percentatge també és correcte ja que la botiga 1 és virtual i la 3 és física i així la proporció de beneficis és 670,85 i 2114,55 que correspon amb el calculat.

9.4. Test dels procediments del package consulta_est.

En aquest paquet tenim els dos procediments que llisten les dades de la taula ESTADISTICA, bé per un any demanat o bé donant l'històric de tots els anys. Com que ja hem comprovat que la taula s'omple bé el que farem serà omplir amb més dades per a tenir sis anys plens i així poder comprovar els dos procediments, que són:

```
PROCEDURE CONSULTA(A in number);  
PROCEDURE CONSULTA_ANY;
```

Per fer-ho tenim les dades de vendes del fitxer TEST_CONSULTA_EST.sql que introdueix dades pels anys 2011 fins al 2015 per completar les del 2010 que ja hem entrat. Comprovam al fitxer LOG que totes les dades han entrat correctament, ja que hi ha 168 registres que es corresponen amb les 163 vendes totals que té la taula VENDES i els 5 errors que havíem introduït en l'apartat 9.2.

Ara podem executar els dos procediments:

```
exec consulta_est.consulta(2010);  
exec consulta_est.consulta_any;
```

dels que obtenim els resultats

PL/SQL procedure successfully completed.

Any	Benefici	IDbotiga	Benef.	Botiga	ean13	Unitats	Hora+	Unitats	Hora-	Unitats	Dia+	Unitats	Dia-	Unitats	Ciutat	Benefici	Percentatge
2010	2785,4	3	2114,55	9111222000292	46	14	48	9	3	15	68	10	1	155	2114,55	24,08	

PL/SQL procedure successfully completed.

Any	Benefici	IDbotiga	Benef.	Botiga	ean13	Unitats	Hora+	Unitats	Hora-	Unitats	Dia+	Unitats	Dia-	Unitats	Ciutat	Benefici	Percentatge
2010	2785,4	3	2114,55	9111222000292	46	14	48	9	3	15	68	10	1	155	2114,55	24,08	
2011	3663,65	2	1415,58	9111222000224	24	19	62	7	2	12	37	2	2	0	2166,16	59,13	
2012	2543,35	3	578,96	9111222000147	25	22	62	15	1	10	37	19	1	155	975,68	18,31	
2013	2646,91	2	1751,19	9111222000255	23	22	51	20	1	21	36	8	1	0	2411,73	91,11	
2014	3195,12	3	957,42	9111222000003	24	20	64	9	4	23	36	5	2	155	1129,26	47,47	
2015	4393,25	2	1144,24	9111222000013	23	16	77	0	1	30	77	14	3	0	1728,21	39,34	

que són els esperats pel contingut de la taula ESTADISTICA.

9.4. Test dels procediments del package consulta_vendes.

Per acabar amb els tests hem de provar els tres procediments que executen les consultes del paquet

consulta_vendes, que són les tres consultes que es demanaven. Per fer-ho utilitzarem les mateixes 161 vendes que tenim a la taula després de les proves anteriors.

En l'execució de la consulta 1 amb els paràmetres 2010 i 11 com a any i mes hem d'obtenir per cada botiga, el total de productes, el nombre de productes venuts, el benefici net, el tant per cent que representa cada botiga per la cadena i el benefici per empleat. Al resultat obtingut tenim llistades les 53 botigues, i només dues d'elles tenen vendes aquest mes, donant el resultat següent:

Botiga	Unitats	Productes	Benefici	% Benefici	Benef/Treb
1	23	2	43,06	90,9	3,59
3	1	1	4,31	9,1	0,27

Botiga	EAN13	Hora	Dia	Mes	Any	Unitats	Preu	Benefici
1	9111222000095	10	11	11	2010	14	201,04	30,1
1	9111222000115	23	15	11	2010	9	258,84	12,96
3	9111222000110	9	10	11	2010	1	14,36	4,31

En l'execució de la consulta 2 amb els mateixos paràmetres 2015 i 12 respectivament i afegint dues vendes més tenim el resultat:

9111222000221 PANTALÓN DENIM SLIM 54 224,1 11 20
9111222000310 ZAPATO CASUAL PISO CONTRASTE 15 89,85 4 15
9111222000167 JERSEY HILO ESTAMPADO 11 13,2 9 11

Botiga	EAN13	Hora	Dia	Mes	Any	Unitats	Preu	Benefici
4	9111222000310	13	1	12	2015	15	599,4	89,85
6	9111222000221	18	24	12	2015	17	352,92	70,55
8	9111222000221	21	24	12	2015	17	352,92	70,55
9	9111222000167	13	7	12	2015	11	263,56	13,2
11	9111222000221	22	24	12	2015	20	415,2	83

Podem veure que es venen tres productes diferents, els valors obtinguts coincideixen.

Finalment executam la consulta 3 on amb els paràmetres 2010 i 11 obtenim:

Dia: Benefici total: EAN13: Unitats: Botiga: Benefici botiga:

10 4,31 9111222000110 1 3 4,31

11 30,1 9111222000095 14 1 30,1

15 12,96 9111222000115 9 1 12,96

on veim el dia, el benefici total de la cadena, el producte més venut, i les unitats i la botiga amb més benefici i aquest valor del benefici. Els resultats coincideixen amb els valors de la taula VENDES que hem presentat en el test de la consulta1.

10. Recursos i valoració econòmica.

10.1. Recursos humans.

Per a desenvolupar el projecte hem destinat, tal i com hem projectat, un total de 21 hores setmanals durant 15 setmanes i dos dies, el que fa un total de 321 hores de feina. Aquestes hores s'han distribuït entre les cinc funcions en que ha necessitat el projecte. Aquestes són:

- Direcció del projecte. Les funcions del cap de projecte, que són les de planificar, coordinar i controlar el desenvolupament, a més de ser l'enllaç amb el client. (59 hores).
- Anàlisis. L'analista s'ocupa de la fase de disseny del projecte. (84 hores).
- Programació. El programador és l'encarregat de dur a terme la implementació de tot el programari. (105 hores).
- Test. El desenvolupan entre l'analista i el programador ja que una part correspondrà a l'elaboració de les dades amb que es provarà la BD, i a més, la tria d'aquestes ha de garantir que el resultat sigui extrapolable al funcionament real. (11+10 hores).
- Presentació i redacció de la documentació. La presentació al client la duu a terme el cap de projecte i la redacció de la documentació, tant el pla de treball com els informes i la memòria final, i de la presentació un administratiu. (52 hores).

10.2. Maquinari.

Tant la base de dades com el programari de SGBD s'han treballat sobre un ordinador sobretaula amb les característiques següents:

Ordinador:	Acer Aspire X1900
Processador:	Pentium Dual-Core E5800 @ 3.20 GHz 3.20 GHz x64
Ram:	4,00 GB
Sistema operatiu:	Windows 10 Pro 64 bits

També s'ha utilitzat un ordinador portàtil per a la realització de les tasques de documentació:

Ordinador:	Acer Pavilion dm1
Processador:	AMD E-450 dual core 1.65 GHz x64
Ram:	4,00 GB
Sistema operatiu:	Ubuntu 14.04 LTS 64 bits

10.3. Programari.

El programari utilitzat en l'elaboració del projecte ha sigut el següent:

Oracle Database Express Edition 11g Release 2

SQL Developer 4.1.1 Production

LibreOffice Versión: 4.8.2.2 (sobre Ubuntu)

Planner 0.14.6 (sobre Ubuntu)

10.4. Hores i cost del projecte.

A partir de la distribució d'hores assignades a cada una de les feines, el cost en recursos humans del projecte ha sigut:

Treballador	Hores	Preu per hora (€)	Preu a facturar (€)
Cap de projecte	59	25,00	1.475,00
Analista	95	18,00	1.710,00
Programador	115	15,00	1.725,00
Administratiu	52	12,00	624,00
Total	321	Total	5.534,00

11. Conclusions.

Hem elaborat un programa bàsic per a la gestió d'un magatzem de dades per a Cadena de Roba, que dóna les funcions bàsiques demanades. En aquest mateix document hem descrit les diferents etapes del disseny i com les hem dutes a terme. Voldria destacar com a aspectes bàsics de l'elaboració de l'aplicació els següents:

- L'ajustament del treball a la realitat és important, potser no per un projecte de l'envergadura d'una cadena de roba, però sí per empreses més petites, ja que ens trobam amb un enunciat que no ens dóna més que els trets generals que es demanen i tot el procés d'especificació l'ha de desenvolupar l'equip.
- També el fet de seguir totes les passes del procés de disseny fa que es tengui la sensació d'un disseny a nivell d'empresa.
- Potser seria bo acabar el projecte amb el disseny d'una interfície gràfica amb un llenguatge d'alt nivell per poder tenir una aplicació completa, però som conscient que el volum de feina seria molt més alt i potser no realitzable per una càrrega dels crèdits del treball.
- Destacar les dificultats en l'especificació que es van trobant en les diferents etapes del disseny i que fins i tot obliguen a tornar a etapes anteriors i fer petits canvis en algun pas.
- Com a cas concret del punt anterior citar que en el mòdul estadístic, on demana el dia o hora amb més o menys unitats venudes, no havíem contemplat la possibilitat de que dues hores o dos dies tenguessin les mateixes unitats i tampoc l'enunciat especifica que fer en aquest cas, per tant hem hagut d'agafar una solució de compromís i agafar el dia o l'hora mínima, amb el valor més petit, però no indicam enlloc que hi ha més dies o hores que tenen les mateixes unitats, ja que el disseny de la taula no ho permet. Aquesta millora suposaria incloure dues files per l'any o més ja que podríem tenir més de dos dies o dues hores amb les mateixes unitats o bé treure l'avís en alguna altra taula.
- Això mateix passa a l'hora de cercar el producte o la botiga amb més vendes o amb més benefici, que tampoc queda resolt en cas d'haver-hi un empat.
- Destacar també la dificultat del test, que a més en haver-lo de fer al final i amb el temps d'entrega ja quasi acabat fa que sempre ens quedi el dubte de no haver contemplat totes les possibilitats. Així seria bo poder disposar de més temps i de tenir més jocs de proves per acabar de completar-lo.
- Un aspecte que m'ha costat molt de decidir i crec que és també interessant de comentar és el mòdul estadístic, on el fet de demanar dades en temps real fa que l'eficiència de la solució adoptada sigui

molt dolenta. Vull dir que he optat per un recalculat de les dades en cada actualització de la taula vendes i això fa que els valors siguin correctes i actuals sempre, ha simplificat el disseny, ja que només he necessitat un procediment per actualitzar la taula ESTADISITICA, però fa que el rendiment sigui molt dolent. Com a millores amb el mateix sistema emprat hi hauria la possibilitat de comprovar si l'actualització de la venda afecta a un producte o botiga ja líders i això faria que no calgués recalculat amb totes les vendes sinó només afegir el canvi al producte o botiga. És cert que en aquest cas el càlcul és més ràpid, però suposa fer la comprovació del producte o botiga entrat en cada actualització de les vendes i la condició a cada actualització també consumeix recursos. En el cas de tenir un catàleg amb molts productes o moltes botigues, el temps invertit en la instrucció condicional potser no compensaria el que guanyam amb el temps de càlcul.

- En el mòdul estadístic també podríem haver tractat de forma diferent les altes, les baixes i les modificacions de les vendes, ja que això evitaria recalculat tots els valors amb molts casos, però hagués triplicat o potser més el temps dedicat i la dificultat de les rutines, tot i que el temps de càlcul seria sensiblement inferior, ja que si donam de baixa una venda que no afecta a cap producte ni botiga líder no cal recalculat la taula, per exemple.
- Així ens queda la sensació de que a l'aplicació li queda molta feina i que les possibilitats són molt més grans, però crec que a nivell bàsic el resultat i la sensació general són satisfactoris.
- També tenc la mateixa sensació no en la solució donada a les parts desenvolupades, sinó en les altres possibilitats a desenvolupar, com la comentada interfície gràfica, i també altres possibilitats de tractar més aspectes de les ciutats o regions i de les vendes que podrien millorar les dades que es podrien treure de l'aplicació, que ha hem comentat al llarg del document.
- Considero que és positiu l'ús de l'Oracle com a SGBS ja que és un producte que ocupa una bona posició en l'ús en empreses que empren aquest programari i això fa que el treball s'assembli més al que ens podem trobar en feines futures.
- Així per acabar la sensació final és bona per la feina, el mètode i el tractament donat però deixa moltes portes obertes la sensació de poder millorar i ampliar amb molts aspectes el que hem fet, però suposo que és també part del propòsit del projecte.

12. Glossari.

Atribut: és el nom del paper o la funció que fa un domini en un esquema de relació.

BD (base de dades): conjunt de fitxers entre els quals hi ha lligams o interrelacions.

Camp: cada un dels atributs d'una tupla d'una taula d'una bases de dades relacional.

Data Warehouse (Magatzem de dades): base de dades amb la informació històrica d'una organització dissenyada i estructurada per a realitzar-hi consultes eficientment.

Disseny (conceptual, lògic i físic): etapes del disseny d'una base de dades on es donen l'estructura d'aquesta, l'estructura adaptada a la tecnologia que s'emprarà i l'estructura adaptada a la implementació física, respectivament.

Domini: En el model relacional un domini és un conjunt de valors atòmics o indivisibles es a dir sense estructuració interna.

Entitat: un objecte del món real que podem distingir de la resta d'objectes i del qual ens interessen algunes propietats (en la representació en una base de dades relacional).

Funció (Function): subprograma dins una base de dades que retorna un valor o valors calculats a partir d'uns paràmetres (que poden ser NULL).

Interrelació: associació entre entitats en una base de dades relacional.

Model relacional: model de base de dades basat en l'estructura, manipulació i integritat d'aquestes.

Oracle Database Express Edition 11g Release 2: aplicació de gestió de bases de dades d'Oracle.

Package: conjunt de funcions i procediments amb interrelacions que desenvolupen tasques específiques.

PL/SQL: llenguatge de programació procedimental que completa SQL per a poder implementar programes per manipular les dades.

Procediment (Procedure): subprograma dins una base de dades que realitza una sèrie de tasques a partir d'uns paràmetres (que poden ser NULL).

Regles de negoci: recull d'informació sobre els diferents objectes del món real que representarem a la BD, dels que necessitam conèixer les seves propietats que guardarem com a atributs en la nostra representació.

Relació: nom que reben les taules de dades en el model teòric de BD relacionals.

Requisits funcionals: característiques que ens donaran les necessitats o restriccions que han de verificar les entitats de la BD desenvolupada.

SGBD: Programa que gestiona i controla bases de dades, especialment facilita l'ús a molts usuaris simultanis i n'assegura la seva integritat.

SQL: Structured Query Language, llenguatge per definir, manipular i controlar bases de dades relacionals.

Taula: conjunt de tuples on s'emmagatzemen les dades en una base de dades relacional.

Taula de fets: taula de la BD on guardarem les dades històriques del magatzem de dades (en aquest cas les vendes de l'empresa).

UML: Unified Modeling Language, diagrama estàtic que descriu l'estructura d'un sistema mostrant les seves classes, atributs i les relacions entre ells.

13. Bibliografia.

Materials docents de la UOC.

- Pradel Miquel, Jordi i altres. Enginyeria del programari. (PID_00213624)
- Xhafa, Fatos. Tècniques de desenvolupament de programari. (XP02/05049/00099)
- Sistac Planas, Jaume (coordinador). Bases de dades I. (XP05/05002/00492)
- Sistac Planas, Jaume (coordinador). Bases de dades II. (XP03/05053/02047)

Documentació online Oracle. (http://docs.oracle.com/cd/E17781_01/index.htm).

- Installation Guide for Microsoft Windows
- Getting started
- 2 Dasy BA
- 2 Days Developers Guide
- 2 Day + Application Express Developer's Guide
- Database SQL Language Reference (Oracle Database Online Documentation)
(https://docs.oracle.com/cd/E11882_01/server.112/e41084/toc.htm)

14. Annex 1. (Test del package act_estadistica.)

BENEFICI PER BOTIGA

	Botiga	Producte	Hora	Dia	Mes	Any	Unitats	Preu	Benefici	
1	1	9111222000186	22	15		3	2010	15	239,4	95,7
2	1	9111222000083	16	24		7	2010	21	301,56	75,39
3	1	9111222000233	7	24		9	2010	20	479,2	119,8
4	1	9111222000012	13	28		9	2010	7	2234,4	223,44
5	1	9111222000095	10	11		11	2010	14	201,04	30,1
6	1	91112220000115	23	15		11	2010	9	258,84	12,96
7	1	9111222000049	6	22		12	2010	24	383,76	57,6
8	1	9111222000181	23	28		12	2010	14	223,44	55,86
9	3	9111222000006	14	26		1	2010	22	2270,4	794,64
10	3	9111222000195	18	27		2	2010	4	127,84	44,76
11	3	9111222000232	14	3		3	2010	5	119,8	17,95
12	3	9111222000292	1	7		4	2010	23	422,28	84,41
13	3	9111222000292	0	15		5	2010	23	422,28	84,41
14	3	9111222000174	1	30		6	2010	13	311,48	62,27
15	3	9111222000107	13	15		7	2010	21	217,56	108,78
16	3	9111222000127	14	3		8	2010	21	839,16	335,58
17	3	9111222000124	6	4		8	2010	18	1151,28	345,42
18	3	9111222000001	10	17		9	2010	7	89,32	40,18
19	3	9111222000312	10	28		10	2010	8	447,68	179,04
20	3	9111222000110	9	10		11	2010	1	14,36	4,31
21	3	9111222000020	9	18		12	2010	2	127,92	12,8
										2114,58
										2785,4

2114,55

2785,4

BENEFICI PER PRODUCTE

	Botiga	Producte	Hora	Dia	Mes	Any	Unitats	Preu	Benefici
18	3	9111222000001	10	17	9	2010	7	89,32	40,18
9	3	9111222000006	14	26	1	2010	22	2270,4	794,64
4	1	9111222000012	13	28	9	2010	7	2234,4	223,44
21	3	9111222000020	9	18	12	2010	2	127,92	12,8
7	1	9111222000049	6	22	12	2010	24	383,76	57,624
2	1	9111222000083	16	24	7	2010	21	301,56	75,39
5	1	9111222000095	10	11	11	2010	14	201,04	30,1
15	3	9111222000107	13	15	7	2010	21	217,56	108,78
20	3	9111222000110	9	10	11	2010	1	14,36	4,31
6	1	9111222000115	23	15	11	2010	9	258,84	12,96
17	3	9111222000124	6	4	8	2010	18	1151,28	345,42
16	3	9111222000127	14	3	8	2010	21	839,16	335,58
14	3	9111222000174	1	30	6	2010	13	311,48	62,27
8	1	9111222000181	23	28	12	2010	14	223,44	55,86
1	1	9111222000186	22	15	3	2010	15	239,4	95,7
10	3	9111222000195	18	27	2	2010	4	127,84	44,76
11	3	9111222000232	14	3	3	2010	5	119,8	17,95
3	1	9111222000233	7	24	9	2010	20	479,2	119,8
12	3	9111222000292	1	7	4	2010	23	422,28	84,41
13	3	9111222000292	0	15	5	2010	23	422,28	84,41
19	3	9111222000312	10	28	10	2010	8	447,68	179,04

UNITATS PER HORA

	Botiga	Producte	Hora	Dia	Mes	Any	Unitats	Preu	Benefici	
13	3	9111222000292	0	15	5	2010	23	422,28	84,41	23
12	3	9111222000292	1	7	4	2010	23	422,28	84,41	
14	3	9111222000174	1	30	6	2010	13	311,48	62,27	36
7	1	9111222000049	6	22	12	2010	24	383,76	57,6	
17	3	9111222000124	6	4	8	2010	18	1151,28	345,42	42
3	1	9111222000233	7	24	9	2010	20	479,2	119,8	20
20	3	9111222000110	9	10	11	2010	1	14,36	4,31	
21	3	9111222000020	9	18	12	2010	2	127,92	12,8	3
5	1	9111222000095	10	11	11	2010	14	201,04	30,1	
18	3	9111222000001	10	17	9	2010	7	89,32	40,18	
19	3	9111222000312	10	28	10	2010	8	447,68	179,04	29
4	1	9111222000012	13	28	9	2010	7	2234,4	223,44	
15	3	9111222000107	13	15	7	2010	21	217,56	108,78	28
9	3	9111222000006	14	26	1	2010	22	2270,4	794,64	
11	3	9111222000232	14	3	3	2010	5	119,8	17,95	
16	3	9111222000127	14	3	8	2010	21	839,16	335,58	48
2	1	9111222000083	16	24	7	2010	21	301,56	75,39	21
10	3	9111222000195	18	27	2	2010	4	127,84	44,76	4
1	1	9111222000186	22	15	3	2010	15	239,4	95,7	15
6	1	9111222000115	23	15	11	2010	9	258,84	12,96	6
8	1	9111222000181	23	28	12	2010	14	223,44	55,86	23

UNITATS PER DIA

	Botiga	Producte	Hora	Dia	Mes	Any	Unitats	Preu	Benefici	
16	3	9111222000127	14	3		8	2010	21	839,16	335,58
11	3	9111222000232	14	3		3	2010	5	119,8	17,95
17	3	9111222000124	6	4		8	2010	18	1151,28	345,42
12	3	9111222000292	1	7		4	2010	23	422,28	84,41
20	3	9111222000110	9	10		11	2010	1	14,36	4,31
5	1	9111222000095	10	11		11	2010	14	201,04	30,14
15	3	9111222000107	13	15		7	2010	21	217,56	108,78
6	1	9111222000115	23	15		11	2010	9	258,84	12,96
1	1	9111222000186	22	15		3	2010	15	239,4	95,7
13	3	9111222000292	0	15		5	2010	23	422,28	84,41
18	3	9111222000001	10	17		9	2010	7	89,32	40,18
21	3	9111222000020	9	18		12	2010	2	127,92	12,8
7	1	9111222000049	6	22		12	2010	24	383,76	57,6
2	1	9111222000083	16	24		7	2010	21	301,56	75,39
3	1	9111222000233	7	24		9	2010	20	479,2	119,8
9	3	9111222000006	14	26		1	2010	22	2270,4	794,64
10	3	9111222000195	18	27		2	2010	4	127,84	44,76
4	1	9111222000012	13	28		9	2010	7	2234,4	223,44
8	1	9111222000181	23	28		12	2010	14	223,44	55,86
19	3	9111222000312	10	28		10	2010	8	447,68	179,04
14	3	9111222000174	1	30		6	2010	13	311,48	62,27