

Disseny i implementació d'una base de dades relacional

Gerardo Aceituna Ibañez
E.T.Informàtica de Gestió

Juan Martínez Bolaños
Consultor TFC

11 de gener de 2015

Agraïments

En primer lloc, vull agrair als meus pares, en Julio i l'Olga, les lliçons de vida que m'han permès aconseguir arribar a la darrera fase dels meus estudis universitaris a la UOC. En segon punt, però no menys important, a l'Àngela que amb la seva paciència, constància i suport ha fet possible que no llences la tovallola quan o veia tot negre, quan no podíem fer certa activitat un cap de setmana perquè tenia una entrega que finalitzar, en fi, per moltíssimes circumstàncies que feien que aquest viatge tingues moltes pedres pel camí.

Finalment, per a tots els amics i companys de feina que m'han donat ànims quan tenia la moral baixa després d'un mal examen, quan ells finalitzaven i em donaven el copet per jo ser el següent.

Resum

Una important cadena de roba, ha obert un concurs per a rebre diferents propostes sobre el disseny d'una base de dades i que a la mateixa vegada, actuï com a *data warehouse* centralitzat, donant a la direcció de la cadena un accés senzill i ràpid a tot un seguit de consultes que permetin millorar el funcionament del negoci.

A partir d'aquest punt de partida, s'ha suposat que la nostra empresa ha guanyat aquest concurs i per tant, ha de fer tot el disseny de la base de dades (emprant per això diferents eines de modelatge d'entitats), una implementació física pensant en què no es tindrà accés directe als servidors i tot un joc de proves per validar amb l'empresa client el correcte funcionament del projecte.

Altres característiques que es demanen en el plec de condicions és que l'aplicació tingui un sistema de procediments d'alta, baixa i modificació de les dades, un seguit de consultes prèviament definides i el sistema de *data warehouse* fent servir consultes sense funcions agregades. Tot això en un entorn escalable i de fàcil manteniment.

Índex de contingut

1.	Introducció	6
1.1.	Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC	6
1.2.	Objectius del TFC.....	6
1.3.	Enfocament i mètode seguit	7
1.4.	Planificació del treball	8
1.4.1.	Descomposició en activitats	8
1.4.2.	Fites	9
1.4.3.	Diagrama de Gantt previst	9
1.4.4.	Diagrama de Gantt final	10
1.4.5.	Avaluació de Riscos	11
1.4.6.	Avaluació del material necessari.....	12
1.4.6.1.	Recursos humans	12
1.4.6.2.	Maquinari.....	12
1.4.6.3.	Programari.....	12
1.5.	Productes obtinguts	13
1.6.	Breu descripció de altres capítuls de la memòria	13
1.7.	Valoració econòmica del projecte.....	13
2.	Definició i anàlisi de requisits.....	16
2.1.	Regles de negoci.....	16
2.2.	Requisits funcionals.....	16
3.	Disseny	17
3.1.	Diagrama entitat-relació en UML.....	17
3.2.	Definició de les entitats.....	17
4.	Implementació	22
4.1.	Creació de la base de dades	22
4.2.	Creació de les taules.....	23
4.3.	Creació del disparador	27
4.4.	Creació dels procediments emmagatzemats	38
4.4.1.	Procediment Transversal.....	39
4.4.2.	Procediments ABM.....	39
4.4.3.	Procediments consulta	42
4.4.4.	Prodediments estadístics	42

5.	Joc de proves	43
6.	Conclusions	50
7.	Glossari.....	51
8.	Bibliografia	52
9.	Annexos.....	53
9.1.	Annex 1 – Instal·lació i joc de proves.	53

1. Introducció

1.1. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC

Les bases de dades són una eina imprescindible per emmagatzemar la informació, gestionar-la i explotar les dades per tal de millorar els processos i/o mecanismes de les empreses i particulars.

A partir d'aquesta idea, ens trobem en el cas on s'ha de presentar un projecte (que d'altra banda resulta guanyador en un concurs públic) per desenvolupar el *data warehouse* i tot l'entorn de base de dades d'una important cadena de botigues de roba.

Per tant, el projecte motiu del TFC haurà de gestionar tota la informació d'aquesta cadena i després proveir als directius d'informació veraç, compactada i en temps real per tal de què puguin prendre les decisions adients per al correcte funcionament de l'empresa.

En aquest sentit, la meua aportació va en el sentit de proveir aquest sistema d'emmagatzematge, unificant en una única persona les diferents tasques que cal portar a terme per desenvolupar aquesta eina.

1.2. Objectius del TFC

El present treball de fi de carrera permet demostrar la consolidació dels coneixements adquirits al llarg de tota la carrera i en particular, les assignatures relacionades amb base de dades.

Per això, cal dissenyar e implementar una base de dades en un entorn relacional que ha de poder gestionar una cadena de botigues, amb un catàleg de productes amb els quals treballen aquestes botigues, la relació de vendes que tenen i finalment, poder realitzar una sèrie d'anàlisis de les dades introduïdes a partir d'un mòdul estadístic, que vindrà prefixat pels requeriments previs indicats per l'empresa client.

Per aquest motiu, la conclusió que podem extreure és que la nota treta al final del projecte ens indicarà el grau d'assoliment dels objectius prefixats a l'assignatura, sent aquests:

- Proporcionar un entorn on emmagatzemar tota la informació generada a l'empresa mitjançant procediments emmagatzemats.
- Consolidació de les dades per tal de proveir als directius d'informació estadística en temps real.
- Altra informació relativa al bon funcionament de l'empresa i les seves botigues.

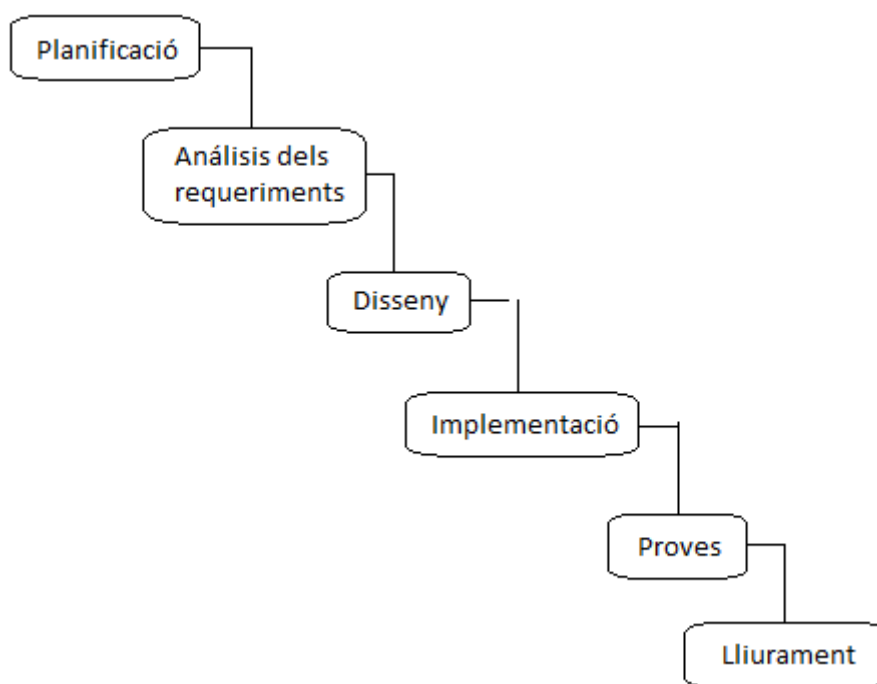
1.3. Enfocament i mètode seguit

La metodologia de treball que s'utilitzarà al llarg del projecte és el cicle de vida en cascada donat que la magnitud del treball a realitzar no és massa gran i només comptem amb una persona, per tant les avantatges que tenim són: ens facilita la gestió del projecte, assegura completar totes les tasques previstes en una etapa abans de saltar a la següent i permet controlar fàcilment si hi han desviacions en el temps previst per cadascuna de les activitats planificades.

Aquestes activitats són:

- Planificació, correspon a l'estudi previ de l'abast i contingut del projecte.
- Anàlisi dels requeriments, s'analitzen els requeriments del sistema per determinar els objectius a cobrir i que per tant, marcarà el desenvolupament de la resta de les activitats.
- Disseny, es realitza el disseny que ha de cobrir els requeriments especificats en la fase prèvia. Aquest disseny és en l'àmbit lògic i físic de la base de dades.
- Implementació, consisteix a crear tot el sistema de consultes a partir del disseny especificat en les fases anteriors.
- Proves, conjunt de verificacions del sistema. S'han de provar totes les funcionalitats incloses per tal de comprovar que es cobreixen cadascun dels requeriments especificats.
- Lliurament, recopilació dels jocs de proves i creació de la documentació final a presentar al client. A més, aquesta documentació serveix com a resum i recordatori de tots els aspectes que abasta el projecte.

La metodologia de treball que s'utilitzarà al llarg del projecte és el cicle de vida en cascada donat que la magnitud del treball a realitzar no és massa gran i només contem amb una persona, per tant les ventatges que tenim són: ens facilita la gestió del projecte, assegura completar totes les tasques previstes en una etapa abans de saltar a la següent i permet controlar fàcilment si hi han desviacions en el temps previst per cadascuna de les activitats planificades.



1 Cicle de vida en cascada

1.4. Planificació del treball

A l'hora de planificar el treball a realitzar, hem de veure l'abast total del projecte, atomitzar el més possible cadascuna de les parts en què es divideix i calcular el temps necessari per fer-hi cadascuna d'aquestes parts.

En atomitzar les activitats en unitats més petites ens permet fer-nos una idea més ajustada de la feina a fer i per tant, ajustar el càlcul de temps necessari.

Per tant, en primer lloc hem de veure quines tasques hem de fer, calcular els dies necessaris per portar-ho a terme, ajustar el temps de treball amb les fites a presentar i finalment, calendaritzar la feina.

1.4.1. Descomposició en activitats

Activitat	Dies
PAC1-Pla de treball	
Lectura de l'enunciat	7
Definició de riscos i contingències	2
Recerca d'informació, estudi i selecció del software necessari	3
Especificació de les tasques a realitzar	3
Definició dels recursos a utilitzar	1
Redacció de la descripció del projecte	4
Valoració de la feina del projecte	1
Valoració econòmica del projecte	1
Revisió del document	1
Entrega	1
PAC2-Disseny de la base de dades	

Identificació i definició d'entitats de negoci	7
Definició de les restriccions dels triggers	2
Creació de l'script per creació de la base de dades	1
Creació de l'script per creació de taules	3
Creació de l'script per creació de triggers	2
Creació de dades de prova	6
Test dels scripts i restriccions dels triggers	8
Entrega	1
PAC3-Lògica del negoci	
Creació dels procediments enmagatzemats	17
Creació joc de prova	3
Execució joc de proves i correccions	10
Entrega	1
Lliurament final	
Redacció memòria	22
Redacció presentació	7
Recopilació fitxers	1
Revisió ortogràfica i semàntica general	3
Entrega	1
Tribunal virtual	
Resolució dubtes tribunal	3

1.4.2. Fites

El lliurament final del TFC es durà a terme l'11 de gener del 2016, sent el dia 27 de gener el darrer dia per resoldre els dubtes del Tribunal virtual.

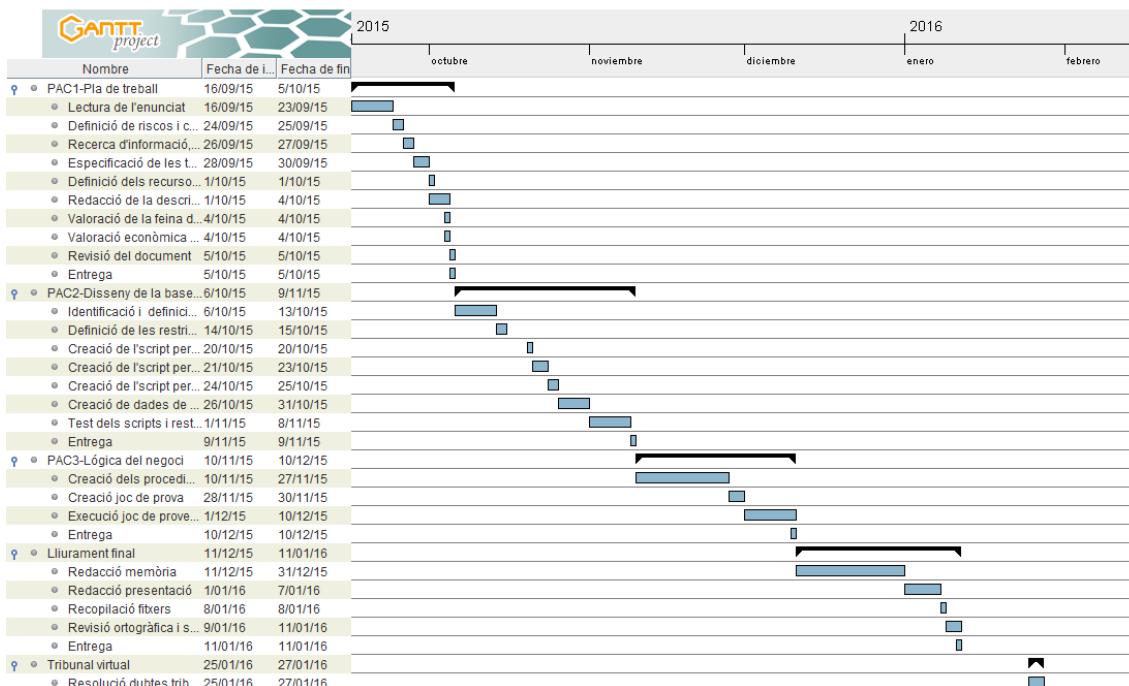
A més del lliurament final, al llarg del quadrimestre es realitzaran altres tres lliuraments més que permetran marcar uns objectius parcials per a la correcta consecució del treball.

A continuació, es mostra un quadre de resum de les diferents fites, el contingut d'aquestes i la data límit per lliurar-les:

Nom lliurament	Contingut	Data de lliurament
PAC 1	Pla de treball	05/10/2015
PAC 2	Disseny de la base de dades	09/11/2015
PAC 3	Lògica del negoci	10/12/2015
Lliurament final	Redacció de la memòria, presentació final i agrupament dels fitxers per l'entrega del càs pràctic	11/01/2016
Tribunal virtual	Resolució de dubtes del Tribunal virtual	27/01/2016

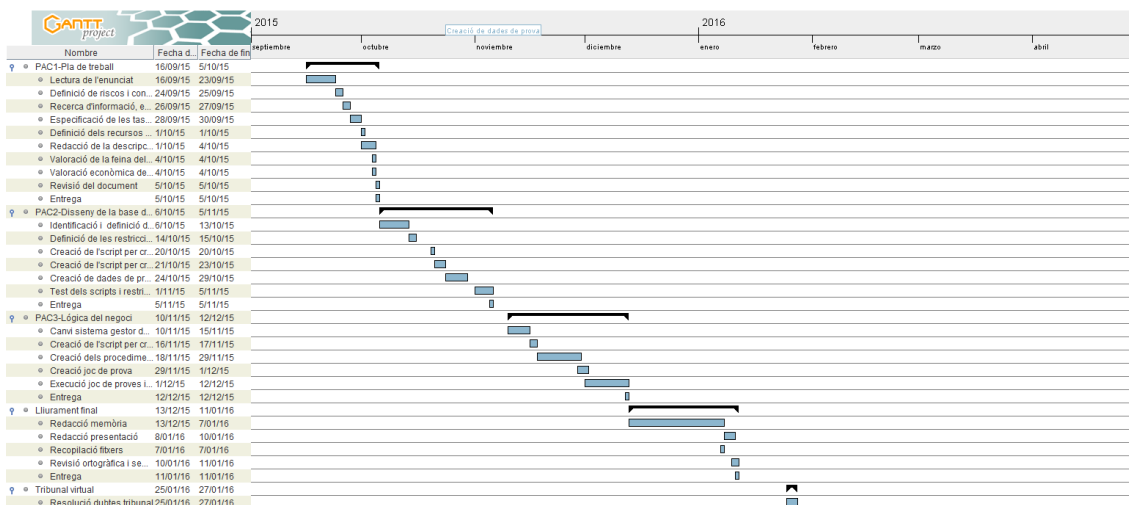
1.4.3. Diagrama de Gantt previst

A continuació es mostra el diagrama de Gantt amb les diferents activitats (entregues) atomitzades al llarg del període establert que s'ha previst al inici del projecte.



2 Diagrama de Gantt

1.4.4. Diagrama de Gantt final



3 Diagrama de Gantt resultant

Un cop finalitzat el projecte, reflectim en el diagrama de Gantt tots els canvis i veiem els desviaments generats durant la realització de l'aplicació.

En primer lloc, podem observar que la segona fita finalitza abans d'hora donat que per motius del meu aniversari i diferents celebracions van provocar que no estigués disponible l'últim cap de setmana i la feina prevista per aquells dies s'endarrerís.

També, cal destacar un altre endarreriment provocat pel canvi de gestor de base de dades. Va provocar sumar un cert nombre d'hores per adaptar els fitxers de creació al nou sistema.

En quant l'última fita, podem veure que la redacció de la memòria s'allarga més enllà del temps definit inicialment. Això bé donat a la manca de temps per les festes nadalenques (no previst a l'avaluació de riscos del projecte) i en el període de vacances al gener s'ha de fer un esforç suplementari per arribar a data.

1.4.5. Avaluació de Riscos

Durant tota la vida del projecte, ens podem trobar amb algun imprevist o complicació (improbable però que pot succeir) que ens enredereixin en el desenvolupament del projecte. Per aquest motiu és bo tindre un pla d'actuació per tal de mitigar en el possible les afectacions que és puguin produir.

Durant l'anàlisi dels possibles riscos hem agafat dos criteris:

- Percentatge de probabilitat, sent una escala que va de menor a major probabilitat (des de l'1% fins al 99%).
- Impacte que té en el projecte (1 = poca afectació fins a 5 = cancel·lació del projecte).

A continuació, tenim una taula que ens mostra aquests possibles riscos i el pla de contingència previst

Risc	Descripció	Prob.	Imp.	Pla de contingència
Baix	Malaltia durant el transcurs del projecte que impedeixi seguir la planificació establerta	10%	1	S'ha de mitigar el retard ampliant la dedicació després d'haver superat la malaltia. També, és necessari avisar al consultor si coincideix amb alguna fita del projecte.
Baix	Pèrdua de la feina feta donat alguna incidència en el sistema informàtic.	5%	4	Es realitzaran còpies periòdiques en altres dispositius del material produït.
Baix	Escapada pel pont del Pilar	20%	1	Augment de la dedicació en els dies posteriors a l'escapada
Mig	Desconeixement d'algunes característiques de les eines empleades	20%	1	Cerca d'informació a Internet o llibres especialitzats
Mig	Hores extres a la feina	10%	2	Augment de la dedicació quan retorni a l'horari habitual i utilització dels dies extres posteriors per posar-me al dia.
Alt	Mala planificació temporal donada la falta d'experiència en aquest camp	25%	3	Augment de la dedicació inclòs, si això implica una sobre dedicació en època nadalenca.
Alt	Correccions de les PAC, donat més incidències de les previstes en el càlcul de temps per correccions.	40%	2	Augment de la dedicació

1.4.6. Avaluació del material necessari

A continuació es descriu els recursos humans necessaris per al correcte desenvolupament del projecte, els dispositius emprats i el programari utilitzat per l'execució del mateix.

1.4.6.1. Recursos humans

Encara que solament realitza aquest projecte una única persona: l'alumna del TFC, assessorada i guiada pel consultor de l'àrea; hem de tindre en compte els perfils i rols que intervenen en la realització de les diferents tasques a desenvolupar durant la vida del projecte.

Aquests perfils vénen marcats per les tasques i responsabilitats que tenen, i això va lligat a un preu hora unitari diferent, que marcarà la valoració econòmica final del projecte.

Els rols utilitzats són:

- Cap de projecte (CP): responsable d'analitzar els requeriments rebuts per part de l'empresa client, proposar el model de dades inicials i supervisar el correcte avanç del desenvolupament del treball per confirmar que es va en temps.
- Programador PL/SQL (PR): persona encarregada d'implementar el model de dades proposat i generar tots els scripts necessaris per al correcte funcionament de l'aplicació. Amb aquests scripts, haurà d'incloure les explicacions del que fa cada cosa per facilitar el manteniment futur.
- Tècnic de proves i documentador (PD): per un costat, el tècnic de proves haurà de generar el joc de proves per validar el correcte funcionament de la base de dades i per un altre, el documentador n'haurà de recopilar tot el treball fet i generar la memòria final, la presentació i preparar el material per l'entrega final.

1.4.6.2. Maquinari

Es farà servir un ordinador portàtil on s'instal·laran tots els elements de programari necessaris. Les característiques són:

- Processador: Intel(R) Core(TM) i3-2350M CPU 2.30GHz
- Memòria: 4GB
- Sistema operatiu: Windows 7 Ultimate SP1

A més, es farà servir un disc dur extraïble (Wester Digital de 500GB) on es realitzaran còpies periòdiques del material produït i un compte en Google Drive per emmagatzemar el material entregat a les PAC i documents auxiliars generats durant la confecció d'aquest material.

1.4.6.3. Programari

A continuació, es llista tots aquells elements de programari necessaris per a la correcta execució del projecte:

- Base de dades: SQL Server 2008 R2
- Gestió de la base de dades: SQL Server 2008 R2 Client
- Tractament de textos: Microsoft Office Word 2007

- Diagrama UML: Microsoft Office Visio 2007
- Planificació i seguiment del projecte: GanttProject
- Presentació del TFC: Microsoft Office PowerPoint 2007

El programari utilitzat durant el desenvolupament d'aquest treball compleix la legalitat, donat que algunes eines són gratuïtes i d'altres estan registrades amb llicència d'estudiant a la UOC.

1.5. Productes obtinguts

Els productes obtinguts durant l'execució d'aquest projecte són els següents:

- Memòria: document de text que conté tota la informació relacionada amb el projecte. Per exemple la metodologia portada a terme, els objectius del treball, relació de la feina realitzada...
- Presentació: document dinàmic que conté un resum de tota la feina realitzada.
- Scripts: conjunt de documents de text que contenen les sentències per a crear la base de dades, les taules on se n'emmagatzemaran les dades, els procediments emmagatzemats, el disparador i les dades de proves.

1.6. Breu descripció de altres capítuls de la memòria

En els propers quatre punts ens trobarem resumit el procés de definició i construcció de la base de dades amb tota la lògica de negoci necessària per al correcte funcionament del projecte. Estant en ordre d'aparició:

- Definició i anàlisi de requisits: En aquest primer capítol, es defineixen les regles que s'hi ha de tenir en compte a l'hora de construir la base de dades. Les regles de funcionament vénen determinades pels requeriments del client.
- Disseny: En aquest altre capítol, s'indica la definició de taules que cal tindre per abastar les diferents necessitats que ens marquen les regles de funcionament.
- Implementació: Aquí tenim la construcció de consultes que contenen la lògica del funcionament i que per tant, són les encarregades de complir els requeriments del client.
- Joc de proves: Per últim, aquest capítol s'encarrega de validar el disseny i implementació de la base de dades.

1.7. Valoració econòmica del projecte

La valoració econòmica inclou la valoració de totes les hores emprades en els diferents processos en què consta el desenvolupament del projecte, això inclou l'anàlisi de requeriments, el disseny de la base de dades, la implementació del model, les proves del producte i la creació de la documentació necessària.

Així, que la distribució de rols davant cadascuna de les activitats és la següent:

Activitat	Rol	Dies	Hores emplades
PAC1-Pla de treball			
Lectura de l'enunciat	CP	7	20
Definició de riscos i contingències	CP	2	5
Recerca d'informació, estudi i selecció del software necessari	CP	3	10
Especificació de les tasques a realitzar	CP	3	8
Definició dels recursos a utilitzar	CP	1	2
Redacció de la descripció del projecte	CP	4	17
Valoració de la feina del projecte	CP	1	1
Valoració econòmica del projecte	CP	1	1
Revisió del document	CP	1	2
Entrega	CP	1	1
PAC2-Disseny de la base de dades			
Identificació i definició d'entitats de negoci	CP	7	18
Definició de les restriccions dels triggers	CP	2	4
Creació de l'script per creació de la base de dades	PR	1	3
Creació de l'script per creació de taules	PR	3	8
Creació de l'script per creació de triggers	PR	2	5
Creació de dades de prova	PR	6	10
Test dels scripts i restriccions dels triggers	PD	8	14
Entrega	CP	1	1
PAC3-Lògica del negoci			
Creació dels procediments enmagatzemats	PR	17	50
Creació joc de prova	PD	3	12
Execució joc de proves i correccions	PD	10	20
Entrega	CP	1	1
Lliurament final			
Redacció memòria	PD	22	50
Redacció presentació	PD	7	15
Recopilació fitxers	PD	1	3
Revisió ortogràfica i semàntica general	PD	3	6
Entrega	CP	1	1
Tribunal virtual			
Resolució dubtes tribunal	CP	3	5

Tal com s'ha explicat abans, a l'hora de calcular el cost del projecte hem de tenir en compte els rols de cadascuna de les persones que intervindran en l'execució del projecte, donat que cadascuna tindrà un cost diferent segons la feina ha de realitzar. El cost unitari d'aquestes persones vénen especificats a un preu aproximat de mercat, sent els següents:

- Cap de projecte: 60,00€/h
- Programador PL-SQL: 43,00€/h
- Tècnic de proves i documentador: 30,50€/h

D'aquesta manera, si convertim el cost unitari pel temps dedicat, ens surt un cost total del projecte de:

Perfil	Q.	P. Unitari	Import
Cap de projecte	97	60,00€	5.820€
Programador PL-SQL	76	43,00€	3.268€
Tècnic de proves i documentador	136	30,50€	4.148€
Total:			13.236€

2. Definició i anàlisi de requisits

2.1. Regles de negoci

Les regles de negoci que podem extreure dels requeriments del client són les següents:

Identificador	Descripció
RN1	Emmagatzemar el nombre de vendes realitzades diàriament de cadascun dels productes disponibles al catàleg.
RN2	Emmagatzemar informació estadística consultable en temps real per la direcció de la cadena que permetrà prendre decisions sobre el negoci.
RN3	Emmagatzemar informació diversa sobre les botigues de la cadena.
RN4	El producte esta centrat només en el disseny de la base de dades.

2.2. Requisits funcionals

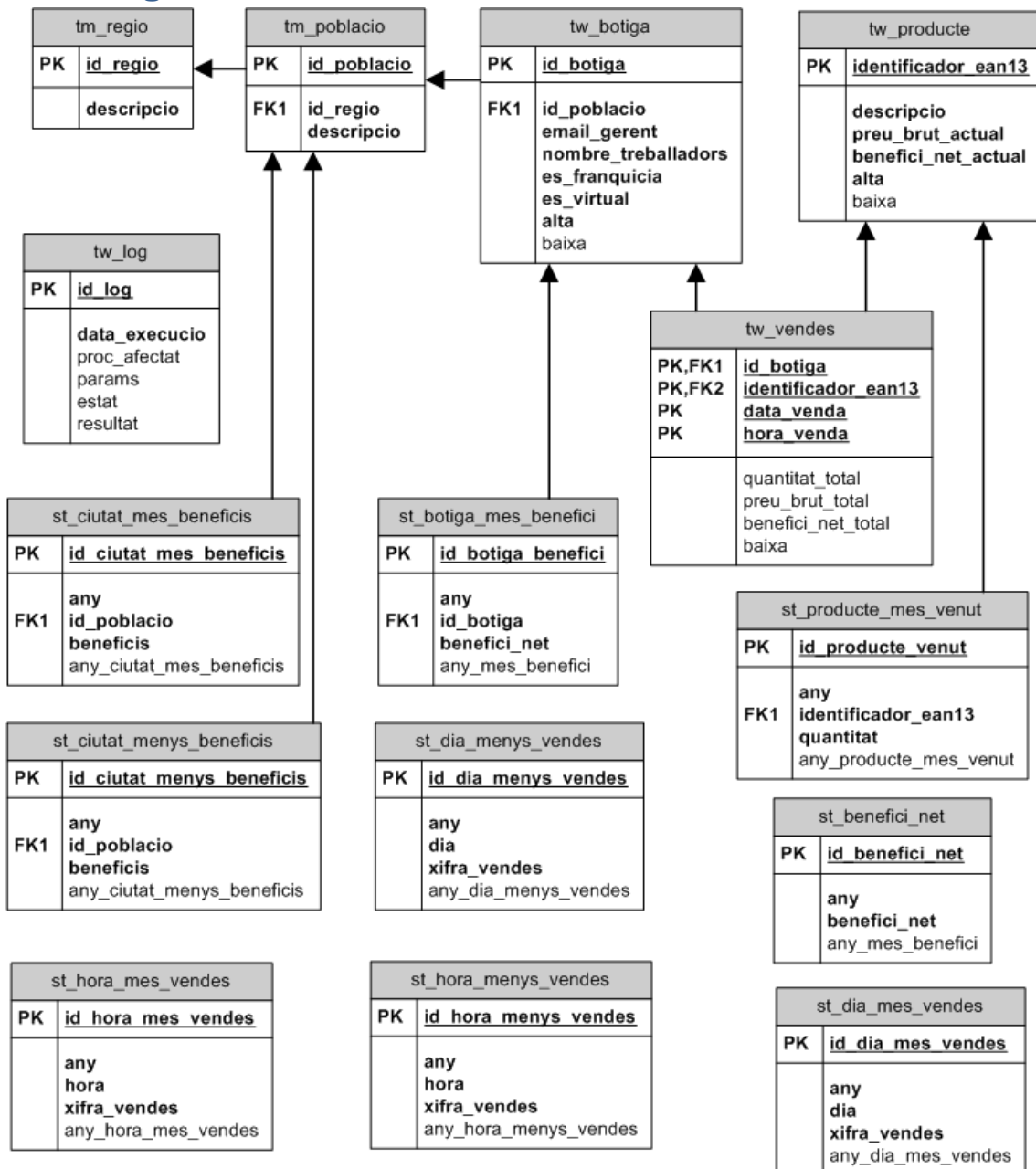
També, tenim uns requisits funcionals extrets dels requeriments demanats pel client:

Identificador	Descripció
RF1	Emmagatzemar dades bàsiques de cada botiga de la cadena.
RF2	Emmagatzemar el catàleg de productes que pot vendre cada botiga, sent aquest únic i comú per a totes les botigues.
RF3	Creació de taula de fets de les vendes realitzades.
RF4	Procediments d'ABM de les botigues, dels productes i de la taula de fets.
RF5	Procediments de consulta que permetin obtenir la informació següent: Llistat de les botigues de la cadena incloent per cada botiga informació variada en relació de les vendes en un moment donat. Productes del catàleg i informació variada de cadascun dels productes en un moment donat. Relació de vendes i beneficis per botigues en cadascun dels dies del mes indicat.
RF6	Mòdul estadístic amb consultes sense funció agregades.

3. Disseny

A partir de l'anàlisi de requisits s'ha de definir una estructura de dades per implementar el model de base de dades independent al sistema gestor empleat.

3.1. Diagrama entitat-relació en UML



4 - Diagrama E/R

3.2. Definició de les entitats

A l'hora de definir el nom de les entitats, s'hi ha escollit donar a cadascuna de les taules un prefix que faciliti a la llarga el manteniment de les entitats i de la informació que emmagatzemen. Per aquesta raó, s'han definit tres prefixos, que són els següents:

- TM, indica que és una taula mestra.
- TW, indica que és una taula d'aplicació, és a dir, rep el manteniment des de l'aplicació que farà un tercer.
- ST, indica que és una taula del mòdul estadístic.

A continuació, apareixen les diferents entitats amb un resum de les característiques que les conformen.

Nom taula	tm_regio					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_regio	int		No	Si		
descripcio	nvarchar	50	No	No		

Nom taula	tm_poblacio					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_poblacio	int		No	Si		
id_regio	int		No	No	tm_regio	id_regio
descripcio	nvarchar	50	No	No		

Nom taula	tw_botiga					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_botiga	int		No	Si		
id_poblacio	int		No	No	tm_poblacio	id_poblacio
email_gerent	nvarchar	100	No	No		
nombre_treballadors	int		No	No		
es_franquicia	bit		No	No		
es_virtual	bit		No	No		
Alta	datetime		No	No		
baixa	datetime		Si	No		

Nom taula	tw_producte					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
identificador_ean13	nvarchar	13	No	Si		
descripcio	nvarchar	200	No	No		
preu_brut_actual	decimal	18,2	No	No		
benefici_net_actual	decimal	18,2	No	No		
Alta	datetime		No	No		
baixa	datetime		Si	No		

Nom taula	tw_vendes					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_botiga	int		No	Si	tw_botiga	id_botiga
identificador_ean13	nvarchar	13	No	Si	tw_producte	identificador_ean13
data_venta	date		No	Si		
hora_venta	int		No	Si		
quantitat_total	int		Si	No		
preu_brut_total	decimal	18,2	Si	No		
benefici_net_total	decimal	18,2	Si	No		
baixa	datetime		Si	No		

Nom taula	tw_log					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_log	int		No	Si		
data_execucio	datetime		No	No		
proc_afectat	nvarchar	50	Si	No		
params	nvarchar	Màxim	Si	No		
estat	bit		Si	No		
resultat	nvarchar	200	Si	No		

Nom taula	st_benefici_net					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_benefici_net	int		No	Si		
Any	int		No	No		
benefici_net	decimal	18,2	No	No		
any_mes_benefici	bit		Si	No		

Nom taula	st_botiga_mes_benefici					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_botiga_benefici	int		No	Si		
Any	int		No	No		
id_botiga	int		No	No	tw_botiga	id_botiga
benefici_net	decimal	18,2	No	No		
any_mes_benefici	bit		Si	No		

Nom taula	st_producte_mes_venut					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_producte_venut	int		No	Si		
any	int		No	No		
identificador_ean13	nvarchar	13	No	No	tw_producte	identificador_ean13
quantitat	int		No	No		
any_producte_mes_venut	bit		Si	No		

Nom taula	st_hora_mes_vendes					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_hora_mes_vendes	int		No	Si		
any	int		No	No		
hora	int		No	No		
xifra_vendes	int		No	No		
any_hora_mes_vendes	bit		Si	No		

Nom taula	st_hora_menys_vendes					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_hora_menys_vendes	int		No	Si		
any	int		No	No		
hora	int		No	No		
xifra_vendes	int		No	No		
any_hora_menys_vendes	bit		Si	No		

Nom taula	st_dia_mes_vendes					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_dia_mes_vendes	int		No	Si		
any	int		No	No		
dia	int		No	No		
xifra_vendes	int		No	No		
any_dia_mes_vendes	bit		Si	No		

Nom taula	st_dia_menys_vendes					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_dia_menys_vendes	int		No	Si		
any	int		No	No		
dia	int		No	No		
xifra_vendes	int		No	No		
any_dia_menys_vendes	bit		Si	No		

Nom taula	st_ciutat_mes_beneficis					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_ciutat_mes_beneficis	int		No	Si		
any	int		No	No		
id_poblacio	int		No	No	tm_poblacio	id_poblacio
beneficis	decimal	18,2	No	No		
any_ciutat_mes_beneficis	bit		Si	No		

Nom taula	st_ciutat_menys_beneficis					
Nom columna	Tipus	Mida	Admet nuls	Clau primària	Clau externa	
					Taula	Columna
id_ciutat_menys_beneficis	int		No	Si		
any	int		No	No		
id_poblacio	int		No	No	tm_poblacio	id_poblacio
beneficis	decimal	18,2	No	No		
any_ciutat_menys_beneficis	bit		Si	No		

4. Implementació

A l'hora d'implementar la base de dades del projecte, hem de fer un seguit de passos. En primer lloc, construir la base de dades on hostatjarem la informació.

Un cop creada la base de dades, hem de crear les taules necessàries per emmagatzemar les dades segons el disseny fet prèviament.

Finalment, crearem els procediments emmagatzemats i els disparadors necessaris per donar la lògica de negoci a l'aplicació tercera.

En base aquestes tres passes, els fitxers de creació estan enumerats en l'ordre adient per construir l'entorn sense cap problema. Cal comentar, que també apareixen els fitxers 4 i 9 que contenen les instruccions pels diferents jocs de proves.

4.1. Creació de la base de dades

A l'hora de construir una base de dades amb el SQL Server tenim dos possibilitats, una amb l'assistent gràfic i un altre amb un script.

En aquest cas, s'ha optat per fer servir un *script* de creació donat la impossibilitat de tenir accés directament al servidor a on s'ha de fer la instal·lació.

En aquest *script* es configura el nom de la base de dades i la resta de característiques (com pot ser la cultura de l'idioma, si distingeix minúscules de majúscules...).

A continuació és mostra la instrucció de creació:

```
USE [master] GO

CREATE DATABASE [TFC] ON PRIMARY
( NAME = N'TFC', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\TFC.mdf' , SIZE = 3072KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = N'TFC_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\TFC_log.ldf' , SIZE = 1024KB ,
MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO

ALTER DATABASE [TFC] SET COMPATIBILITY_LEVEL = 100 GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [TFC].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO

ALTER DATABASE [TFC] SET ANSI_NULL_DEFAULT OFF GO
ALTER DATABASE [TFC] SET ANSI_NULLS OFF GO
ALTER DATABASE [TFC] SET ANSI_PADDING OFF GO
ALTER DATABASE [TFC] SET ANSI_WARNINGS OFF GO
ALTER DATABASE [TFC] SET ARITHABORT OFF GO
ALTER DATABASE [TFC] SET AUTO_CLOSE OFF GO
```

```

ALTER DATABASE [TFC] SET AUTO_CREATE_STATISTICS ON GO
ALTER DATABASE [TFC] SET AUTO_SHRINK OFF GO
ALTER DATABASE [TFC] SET AUTO_UPDATE_STATISTICS ON GO
ALTER DATABASE [TFC] SET CURSOR_CLOSE_ON_COMMIT OFF GO
ALTER DATABASE [TFC] SET CURSOR_DEFAULT GLOBAL GO
ALTER DATABASE [TFC] SET CONCAT_NULL_YIELDS_NULL OFF GO
ALTER DATABASE [TFC] SET NUMERIC_ROUNDABORT OFF GO
ALTER DATABASE [TFC] SET QUOTED_IDENTIFIER OFF GO
ALTER DATABASE [TFC] SET RECURSIVE_TRIGGERS OFF GO
ALTER DATABASE [TFC] SET DISABLE_BROKER GO
ALTER DATABASE [TFC] SET AUTO_UPDATE_STATISTICS_ASYNC OFF GO
ALTER DATABASE [TFC] SET DATE_CORRELATION_OPTIMIZATION OFF GO
ALTER DATABASE [TFC] SET TRUSTWORTHY OFF GO
ALTER DATABASE [TFC] SET ALLOW_SNAPSHOT_ISOLATION OFF GO
ALTER DATABASE [TFC] SET PARAMETERIZATION SIMPLE GO
ALTER DATABASE [TFC] SET READ_COMMITTED_SNAPSHOT OFF GO
ALTER DATABASE [TFC] SET HONOR_BROKER_PRIORITY OFF GO
ALTER DATABASE [TFC] SET READ_WRITE GO
ALTER DATABASE [TFC] SET RECOVERY SIMPLE GO
ALTER DATABASE [TFC] SET MULTI_USER GO
ALTER DATABASE [TFC] SET PAGE_VERIFY CHECKSUM GO
ALTER DATABASE [TFC] SET DB_CHAINING OFF GO

```

4.2. Creació de les taules

El següent pas és crear les taules necessàries per emmagatzemar les dades generades durant la vida de l'aplicació.

El primer pas és definir la base de dades que es farà servir amb la instrucció "USE".

Un cop definida la base de dades, validem que per proves no existeixin ja les taules. Si existeixen, les esborrem per evitar problemes posteriors.

El tercer pas és crear cadascuna de les taules amb totes les restriccions necessàries, com són les claus primàries, claus foranes, possibilitat d'inserir valors nuls, tipus dels atributs de cadascuna de les entitats, etc.

```

use TFC

set nocount on

/* esborrat de taules existents */
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_benefici_net]')))
    drop table [dbo].[st_benefici_net]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_botiga_mes_benefici]')))
    drop table [dbo].[st_botiga_mes_benefici]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_producte_mes_venut]')))
    drop table [dbo].[st_producte_mes_venut]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_hora_mes_vendes]')))
    drop table [dbo].[st_hora_mes_vendes]

```

```

if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_hora_menys_vendes]')))
    drop table [dbo].[st_hora_menys_vendes]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_dia_mes_vendes]')))
    drop table [dbo].[st_dia_mes_vendes]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_dia_menys_vendes]')))
    drop table [dbo].[st_dia_menys_vendes]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_ciutat_mes_beneficis]')))
    drop table [dbo].[st_ciutat_mes_beneficis]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[st_ciutat_menys_beneficis]')))
    drop table [dbo].[st_ciutat_menys_beneficis]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[tw_log]')))
    drop table [dbo].[tw_log]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[tw_vendes]')))
    drop table [dbo].[tw_vendes]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[tw_producte]')))
    drop table [dbo].[tw_producte]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[tw_botiga]')))
    drop table [dbo].[tw_botiga]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[tm_poblacio]')))
    drop table [dbo].[tm_poblacio]
if(exists(select * from sys.objects where object_id =
object_id(N'[dbo].[tm_regio]')))
    drop table [dbo].[tm_regio]

/* taula de creació de les regions on es troben localitzades les
botigues */
create table tm_regio (
    id_regio int identity not null,
    descripcio nvarchar(50) not null,
    constraint pk_regio primary key (id_regio)
);

/* taula de creació de les poblacions on es troben localitzades les
botigues, pertanyen a una única regió */
create table tm_poblacio (
    id_poblacio int identity not null,
    id_regio int not null,
    descripcio nvarchar(50) not null,
    constraint pk_poblacio primary key (id_poblacio),
    constraint fk_regio_poblacio foreign key (id_regio) references
tm_regio (id_regio)
);

```



```

/* taula de botigues */
create table tw_botiga (
    id_botiga int identity not null,
    id_poblacio int not null,
    email_gerent nvarchar(100) not null,
    nombre_treballadors int not null,
    es_franquicia bit not null,
    es_virtual bit not null,
    alta datetime not null,
    baixa datetime null,
    constraint pk_botiga primary key (id_botiga),
    constraint fk_poblacio_botiga foreign key (id_poblacio)
references tm_poblacio (id_poblacio)
);

/* taula de productes disponibles per les botigues */
create table tw_producte (
    identificador_ean13 nvarchar(13) not null,
    descripcio nvarchar(200) not null,
    preu_brut_actual decimal(18,2) not null,
    benefici_net_actual decimal(18,2) not null,
    alta datetime not null,
    baixa datetime null,
    constraint pk_producte primary key (identificador_ean13)
);

/* taula de fets, per registrar les vendes realitzades per una
botiga en un moment determinat en format 24h */
create table tw_vendes (
    id_botiga int identity not null,
    identificador_ean13 nvarchar(13) not null,
    data_venda date not null,
    hora_venda int not null,
    quantitat_total int,
    preu_brut_total decimal(18,2),
    benefici_net_total decimal(18,2),
    baixa datetime null,
    constraint pk_vendes primary key (id_botiga,
identificador_ean13, data_venda, hora_venda),
    constraint fk_botiga_vendes foreign key (id_botiga) references
tw_botiga(id_botiga),
    constraint fk_producte_vendes foreign key (identificador_ean13)
references tw_producte (identificador_ean13)
);

/* registre de les accions portades a terme */
create table tw_log (
    id_log int identity not null,
    data_execucio datetime not null,
    proc_afectat nvarchar(50),
    params nvarchar(max),
    estat bit,
    resultat nvarchar(200),
    constraint pk_log primary key (id_log)
);

```

```

/*****
***** modul estadistic *****/
*****/
/* El benefici net total de tota la cadena */
create table [dbo].[st_benefici_net] (
    id_benefici_net int identity not null,
    [any] int not null,
    benefici_net decimal(18,2) not null,
    any_mes_benefici bit,
    constraint pk_benefici_net primary key (id_benefici_net)
);

/* L'identificador de la botiga que més beneficis nets ha
aconseguit, així com la xifra total d'aquest benefici en euros. */
create table [dbo].[st_botiga_mes_benefici] (
    id_botiga_benefici int identity not null,
    [any] int not null,
    id_botiga int not null,
    benefici_net decimal(18,2) not null,
    any_mes_benefici bit,
    constraint pk_botiga_benefici primary key
(id_botiga_benefici),
    constraint fk_botiga_botiga_benefici foreign key (id_botiga)
references tw_botiga(id_botiga)
);

/* L'identificador EAN13 del producte més venut */
create table [dbo].[st_producte_mes_venut] (
    id_producte_venut int identity not null,
    [any] int not null,
    identificador_ean13 nvarchar(13) not null,
    quantitat int not null,
    any_producte_mes_venut bit,
    constraint pk_producte_venut primary key (id_producte_venut),
    constraint fk_producte_producte_mes_venut foreign key
(identificador_ean13) references tw_producte(identificador_ean13)
);

/* L'hora del dia on més productes s'han venut i la xifra de total
de productes venuts. */
create table [dbo].[st_hora_mes_vendes] (
    id_hora_mes_vendes int identity not null,
    [any] int not null,
    [hora] int not null,
    xifra_vendes int not null,
    any_hora_mes_vendes bit,
    constraint pk_hora_mes_vendes primary key (id_hora_mes_vendes)
);

/* L'hora del dia on menys productes s'han venut i la xifra total de
productes venuts. */
create table [dbo].[st_hora_menys_vendes] (
    id_hora_menys_vendes int identity not null,
    [any] int not null,
    [hora] int not null,
    xifra_vendes int not null,
    any_hora_menys_vendes bit,
    constraint pk_hora_menys_vendes primary key
(id_hora_menys_vendes)
);

```

```

/* El dia del mes on més vendes s'han realitzat i la xifra total de
productes venuts. */
create table [dbo].[st_dia_mes_vendes] (
    id_dia_mes_vendes int identity not null,
    [any] int not null,
    [dia] int not null,
    [xifra_vendes] int not null,
    any_dia_mes_vendes bit,
    constraint pk_dia_mes_vendes primary key (id_dia_mes_vendes)
);

/* El dia del mes on menys vendes s'han realitzat i la xifra total
de productes venuts. */
create table [dbo].[st_dia_menys_vendes] (
    id_dia_menys_vendes int identity not null,
    [any] int not null,
    [dia] int not null,
    [xifra_vendes] int not null,
    any_dia_menys_vendes bit,
    constraint pk_dia_menys_vendes primary key
(id_dia_menys_vendes)
);

/* Ciutat on més beneficis nets s'han obtingut juntament amb aquest
benefici. */
create table [dbo].[st_ciutat_mes_beneficis] (
    id_ciutat_mes_beneficis int identity not null,
    [any] int not null,
    id_poblacio int not null,
    beneficis decimal(18,2) not null,
    any_ciutat_mes_beneficis bit,
    constraint pk_ciutat_mes_beneficis primary key
(id_ciutat_mes_beneficis),
    constraint fk_poblacio_ciutat_mes_beneficis foreign key
(id_poblacio) references tm_poblacio(id_poblacio)
);

/* Percentatge de beneficis obtinguts per tendes virtuals respecte
al total de beneficis de la cadena. */
create table [dbo].[st_ciutat_menys_beneficis] (
    id_ciutat_menys_beneficis int identity not null,
    [any] int not null,
    id_poblacio int not null,
    beneficis decimal(18,2) not null,
    any_ciutat_menys_beneficis bit,
    constraint pk_ciutat_menys_beneficis primary key
(id_ciutat_menys_beneficis),
    constraint fk_poblacio_ciutat_menys_beneficis foreign key
(id_poblacio) references tm_poblacio(id_poblacio)
);

```

4.3. Creació del disparador

El fitxer número tres conté la definició del disparador que afecta la taula de fets, és a dir, la taula anomenada `tw_vendes`.

S'ha decidit que totes les consultes es trobin dintre d'un únic disparador de forma que se sintetitzi tota la lògica en un únic punt.

Com en els punts anteriors, el primer que cal fer, és revisar la instrucció "USE" per validar la base de dades que es fa servir.

A continuació es revisa que no existeixi ja el disparador i en el cas que si existeix, l'esborrem.

A continuació es mostra la creació del disparador amb la determinació del tipus d'acció que ha sofert la taula i la lògica enrere a la càrrega de dades de les taules del mòdul estadístic.

```
use TFC

if(exists(select * from sys.triggers where name = 'trg_vendes'))
    drop trigger [dbo].[trg_vendes]
go

create trigger [dbo].[trg_vendes]
on tw_vendes
after insert, update, delete
as
begin
    set nocount on

    -- comprova tipus d'acció
    declare @accio as char(1)
    set @accio = (case when exists(select * from inserted) and
exists(select * from deleted) then 'U' --actualització
        when exists(select * from inserted) then 'I' -- inserció
        when exists(select * from deleted) then 'D' -- eliminació
        else null -- acció no vàlida
        end)

    --definim les variables auxiliars
    declare @id_botiga int,
            @identificador_ean13 nvarchar(13),
            @id_poblacio int,
            @data date,
            @hora int,
            @quantitat_total int,
            @preu_brut_total decimal(18,2),
            @benefici_net_total decimal(18,2),
            @baixa datetime

    -- carreguem les dades a les variables auxiliars
    if(@accio = 'I' or @accio = 'U')
    begin
select @id_botiga = id_botiga, @identificador_ean13 =
identificador_ean13, @data = data_venta, @hora = hora_venta,
@quantitat_total = quantitat_total, @preu_brut_total =
preu_brut_total, @benefici_net_total = benefici_net_total, @baixa =
baixa
        from INSERTED
    end
```

```

else
  begin
    select @id_botiga = id_botiga, @identificador_ean13 =
    identificador_ean13, @data = data_venta, @hora = hora_venta,
    @quantitat_total = quantitat_total, @preu_brut_total =
    preu_brut_total, @benefici_net_total = benefici_net_total, @baixa =
    baixa
    from DELETED
  end

  --cerquem la poblacio arran de la botiga
  set @id_poblacio = (select id_poblacio from tw_botiga where
  id_botiga = @id_botiga)

  declare @id_st int -- auxiliar per agafar l'identificador de
  la taula
  declare @any int = year(@data) --agafem l'any de la venda

  -----
  -- st_benefici_net
  -----
  -- reiniciem el historic
  update st_benefici_net set any_mes_benefici = 0

  -- cerquem si n'hi ha una entrada pel any especificat
  select @id_st = id_benefici_net from st_benefici_net where
  [any] = @any
  if(@id_st is null)
  begin
    -- inserim un nou any
    insert into st_benefici_net ([any], benefici_net,
  any_mes_benefici)
    select @any, isnull(sum(benefici_net_total), 0), 0
    from tw_vendes
    where YEAR(data_venta) = @any
    and baixa is null
  end
  else
  begin
    update st_benefici_net set benefici_net = isnull((select
  SUM(benefici_net_total) from tw_vendes where YEAR(data_venta) =
  @any), 0) where [any] = @any
  end

  -- cerquem quin es l'any amb més vendes
  declare @any_historic int = (select top 1 [any] from
  st_benefici_net t inner join (select max(benefici_net) benefici_net
  from st_benefici_net) tt on tt.benefici_net = t.benefici_net)
  update st_benefici_net set any_mes_benefici = 1 where [any] =
  @any_historic

  -----
  -- st_botiga_mes_benefici
  -----
  -- reiniciem el historic
  update st_botiga_mes_benefici set any_mes_benefici = 0

```

```

-- cerquem si n'hi ha una entrada pel any especificat
select @id_st = id_botiga_benefici from st_botiga_mes_benefici
where [any] = @any
if(@id_st is null)
begin
-- inserim un nou any
insert into st_botiga_mes_benefici ([any], id_botiga,
benefici_net, any_mes_benefici)
select @any, @id_botiga, isnull(sum(benefici_net_total),
0), 0
from tw_vendes
where YEAR(data_venda) = @any
and id_botiga = @id_botiga
and baixa is null
end
else
begin
-- obtenim el benefici actual i el calculat per la botiga
declare @aux_ben decimal(18,2), @aux_bot int
select @aux_ben = benefici_net, @aux_bot = id_botiga from
st_botiga_mes_benefici where [any] = @any
declare @aux2_ben decimal(18,2)
select @aux2_ben = sum(benefici_net_total) from tw_vendes
where year(data_venda) = @any and id_botiga = @id_botiga and baixa
is null

-- si el total del benefici de la botiga es més alt que
el actual, substituim
if @aux2_ben > @aux_ben
update st_botiga_mes_benefici set id_botiga =
@id_botiga, benefici_net = @aux2_ben where id_botiga_benefici =
@id_st
else
begin
-- si es la mateixa botiga i el benefici es menor,
calculem quin botiga té el benefici més alt i substituim
if @aux_bot = @id_botiga and @aux2_ben < @aux_ben
begin
update st_botiga_mes_benefici set id_botiga =
v.id_botiga, benefici_net = v.benefici_net
from st_botiga_mes_benefici st, (
select top 1 id_botiga,
sum(benefici_net_total) benefici_net
from tw_vendes
where year(data_venda) = @any
and baixa is null
group by id_botiga
order by benefici_net desc
) v
where st.id_botiga_benefici = @id_st
end
end
end

-- cerquem quin es l'any amb més vendes
set @any_historic = (select top 1 [any] from
st_botiga_mes_benefici t inner join (select max(benefici_net)
benefici_net from st_botiga_mes_benefici) tt on tt.benefici_net =
t.benefici_net)
update st_botiga_mes_benefici set any_mes_benefici = 1 where
[any] = @any_historic

```

```

-----
-- st_producte_mes_venut
-----
-- reiniciem el historic
update st_producte_mes_venut set any_producte_mes_venut = 0

-- cerquem si n'hi ha una entrada pel any especificat
select @id_st = id_producte_venut from st_producte_mes_venut
where [any] = @any
if(@id_st is null)
begin
    -- inserim un nou any
    insert into st_producte_mes_venut ([any],
identificador_ean13, quantitat, any_producte_mes_venut)
select @any, @identificador_ean13,
isnull(sum(quantitat_total), 0), 0
from tw_vendes
where YEAR(data_venda) = @any
and identificador_ean13 = @identificador_ean13
and baixa is null

end
else
begin
    declare @aux_quan decimal(18,2), @aux_prod nvarchar(13)
    -- obtenim la quantitat de vendes actual i el calculat
per el producte
    select @aux_quan = quantitat, @aux_prod =
identificador_ean13 from st_producte_mes_venut where [any] = @any
    declare @aux2_quan decimal(18,2)
    select @aux2_quan = sum(quantitat_total) from tw_vendes
where year(data_venda) = @any and identificador_ean13 =
@identificador_ean13 and baixa is null

    -- si el total de la quantitat de vendes del producte es
més alt que el actual, substituïm
    if @aux2_quan > @aux_quan
        update st_producte_mes_venut set
identificador_ean13 = @identificador_ean13, quantitat = @aux2_ben
where id_producte_venut = @id_st
    else
    begin
        -- si es el mateix producte i la quantitat es
menor, calculem quin producte té més vendes i substituïm
        if @aux_prod = @identificador_ean13 and @aux2_quan
< @aux_quan
            begin
                update st_producte_mes_venut set
identificador_ean13 = v.identificador_ean13, quantitat = v.quantitat
                from st_producte_mes_venut st, (
                    select top 1 identificador_ean13,
sum(quantitat_total) quantitat
                    from tw_vendes
                    where year(data_venda) = @any
                    and baixa is null
                    group by identificador_ean13
                    order by quantitat desc
                ) v
                where st.id_producte_venut = @id_st
            end
        end
    end
end
end

```

```

-- cerquem quin es l'any amb més vendes
set @any_historic = (select top 1 [any] from
st_producte_mes_venut t inner join (select max(quantitat) quantitat
from st_producte_mes_venut) tt on tt.quantitat = t.quantitat)
update st_producte_mes_venut set any_producte_mes_venut = 1
where [any] = @any_historic

-----
-- st_hora_mes_vendes
-----

-- reiniciem el historic
update st_hora_mes_vendes set any_hora_mes_vendes = 0

-- cerquem si n'hi ha una entrada pel any especificat
select @id_st = id_hora_mes_vendes from st_hora_mes_vendes
where [any] = @any
if(@id_st is null)
begin
-- inserim un nou any
insert into st_hora_mes_vendes ([any], hora,
xifra_vendes, any_hora_mes_vendes)
select @any, @hora, isnull(sum(quantitat_total), 0), 0
from tw_vendes
where YEAR(data_venda) = @any
and hora_venda = @hora
and baixa is null
end
else
begin
declare @aux_vendes_mes int, @aux_hora_mes int
-- obtenim la quantitat de vendes actual i el calculat
per el producte
select @aux_vendes_mes = xifra_vendes, @aux_hora_mes =
hora from st_hora_mes_vendes where [any] = @any
declare @aux2_vendes_mes int
select @aux2_vendes_mes = sum(quantitat_total) from
tw_vendes where year(data_venda) = @any and hora_venda = @hora and
baixa is null

-- si el total de la quantitat de vendes en una hora es
més alt que el actual, substituïm
if @aux2_vendes_mes > @aux_vendes_mes
update st_hora_mes_vendes set hora = @hora,
xifra_vendes = @aux2_vendes_mes where id_hora_mes_vendes = @id_st
else
begin
-- si es la mateixa hora i la quantitat es menor,
calculem quina hora té més vendes i substituïm
if @aux_hora_mes = @hora and @aux2_vendes_mes <
@aux_vendes_mes
begin
update st_hora_mes_vendes set hora =
v.hora_venda, xifra_vendes = v.quantitat
from st_hora_mes_vendes st, (
select top 1 hora_venda,
sum(quantitat_total) quantitat
from tw_vendes
where year(data_venda) = @any
and baixa is null
group by hora_venda
order by quantitat desc
) v

```



```

                                where st.id_hora_mes_vendes = @id_st
                                end
                                end
                                end
                                end

                                -- cerquem quin es l'any amb més vendes
                                set @any_historic = (select top 1 [any] from
st_hora_mes_vendes t inner join (select max(xifra_vendes)
xifra_vendes from st_hora_mes_vendes) tt on tt.xifra_vendes =
t.xifra_vendes)
                                update st_hora_mes_vendes set any_hora_mes_vendes = 1 where
[any] = @any_historic

                                -----
                                -- st_hora_menys_vendes
                                -----
                                -- reiniciem el historic
                                update st_hora_menys_vendes set any_hora_menys_vendes = 0

                                -- cerquem si n'hi ha una entrada pel any especificat
                                select @id_st = id_hora_menys_vendes from st_hora_menys_vendes
where [any] = @any
                                if(@id_st is null)
                                begin
                                        -- inserim un nou any
                                        insert into st_hora_menys_vendes ([any], hora,
xifra_vendes, any_hora_menys_vendes)
                                        select @any, @hora, isnull(sum(quantitat_total), 0), 0
                                        from tw_vendes
                                        where YEAR(data_venda) = @any
                                        and hora_venda = @hora
                                        and baixa is null
                                end
                                else
                                begin
                                        declare @aux_vendes_menys int, @aux_hora_menys int
                                        -- obtenim la quantitat de vendes actual i el calculat
per el producte
                                        select @aux_vendes_menys = xifra_vendes, @aux_hora_menys
= hora from st_hora_menys_vendes where [any] = @any
                                        declare @aux2_vendes_menys int
                                        select @aux2_vendes_menys = sum(quantitat_total) from
tw_vendes where year(data_venda) = @any and hora_venda = @hora and
baixa is null
                                        -- si el total de la quantitat de vendes en una hora es
més baix que el actual, substituïm
                                        if @aux2_vendes_menys < @aux_vendes_menys
                                        update st_hora_menys_vendes set hora = @hora,
xifra_vendes = @aux2_vendes_menys where id_hora_menys_vendes =
@id_st
                                        else
                                        begin
                                                -- si es la mateixa hora i la quantitat es més
gran, calculem quina hora té menys vendes i substituïm
                                                if @aux_hora_menys = @hora and @aux2_vendes_menys >
@aux_vendes_menys
                                                begin
                                                        update st_hora_menys_vendes set hora =
v.hora_venda, xifra_vendes = v.quantitat
                                                        from st_hora_menys_vendes st, (
                                                                select top 1 hora_venda,
sum(quantitat_total) quantitat

```

```

        from tw_vendes
        where year(data_venda) = @any
              and baixa is null
        group by hora_venda
        order by quantitat asc
    ) v
    where st.id_hora_menys_vendes = @id_st
end
end
end

-- cerquem quin es l'any amb menys vendes
set @any_historic = (select top 1 [any] from
st_hora_menys_vendes t inner join (select min(xifra_vendes)
xifra_vendes from st_hora_menys_vendes) tt on tt.xifra_vendes =
t.xifra_vendes)
update st_hora_menys_vendes set any_hora_menys_vendes = 1
where [any] = @any_historic

-----
-- st_dia_mes_vendes
-----

-- reiniciem el historic
update st_dia_mes_vendes set any_dia_mes_vendes = 0

-- cerquem si n'hi ha una entrada pel any especificat
select @id_st = id_dia_mes_vendes from st_dia_mes_vendes where
[any] = @any
if(@id_st is null)
begin
    -- inserim un nou any
    insert into st_dia_mes_vendes ([any], dia, xifra_vendes,
any_dia_mes_vendes)
select @any, DAY(@data), isnull(sum(quantitat_total), 0),
0
    from tw_vendes
    where YEAR(data_venda) = @any
          and DAY(data_venda) = DAY(@data)
          and baixa is null
end
else
begin
    declare @aux_vendesdia_mes int, @aux_dia_mes int
    -- obtenim la quantitat de vendes actual i el calculat
per el dia
    select @aux_vendesdia_mes = xifra_vendes, @aux_dia_mes =
dia from st_dia_mes_vendes where [any] = @any
    declare @aux2_vendesdia_mes int
    select @aux2_vendesdia_mes = sum(quantitat_total) from
tw_vendes where year(data_venda) = @any and DAY(data_venda) =
DAY(@data) and baixa is null

    -- si el total de la quantitat de vendes en un dia es més
alt que el actual, substituïm
    if @aux2_vendesdia_mes > @aux_vendesdia_mes
        update st_dia_mes_vendes set dia = DAY(@data),
xifra_vendes = @aux2_vendesdia_mes where id_dia_mes_vendes = @id_st
    else
    begin

```

```

-- si es el mateix dia i la quantitat es menor,
calculem quin dia té més vendes i substituim
    if @aux_dia_mes = DAY(@data) and
@aux2_vendesdia_mes < @aux_vendesdia_mes
        begin
            update st_dia_mes_vendes set dia =
v.data_venta, xifra_vendes = v.quantitat
            from st_dia_mes_vendes st, (
                select top 1 DAY(data_venta)
data_venta, sum(quantitat_total) quantitat
                from tw_vendes
                where year(data_venta) = @any
                    and baixa is null
                group by DAY(data_venta)
                order by quantitat desc
            ) v
            where st.id_dia_mes_vendes = @id_st
        end
    end
end

-- cerquem quin es l'any amb més vendes
set @any_historic = (select top 1 [any] from st_dia_mes_vendes
t inner join (select max(xifra_vendes) xifra_vendes from
st_dia_mes_vendes) tt on tt.xifra_vendes = t.xifra_vendes)
update st_dia_mes_vendes set any_dia_mes_vendes = 1 where
[any] = @any_historic

-----
-- st_dia_menys_vendes
-----

-- reiniciem el historic
update st_dia_menys_vendes set any_dia_menys_vendes = 0

-- cerquem si n'hi ha una entrada pel any especificat
select @id_st = id_dia_menys_vendes from st_dia_menys_vendes
where [any] = @any
if(@id_st is null)
begin
    -- inserim un nou any
    insert into st_dia_menys_vendes ([any], dia,
xifra_vendes, any_dia_menys_vendes)
    select @any, DAY(@data), isnull(sum(quantitat_total), 0),
0
    from tw_vendes
    where YEAR(data_venta) = @any
        and DAY(data_venta) = DAY(@data)
        and baixa is null
end
else
begin
    declare @aux_vendesdia_menys int, @aux_dia_menys int
-- obtenim la quantitat de vendes actual i el calculat
per el dia
    select @aux_vendesdia_menys = xifra_vendes,
@aux_dia_menys = dia from st_dia_menys_vendes where [any] = @any
    declare @aux2_vendesdia_menys int select
@aux2_vendesdia_menys = sum(quantitat_total) from tw_vendes where
year(data_venta) = @any and DAY(data_venta) = DAY(@data) and baixa
is null

```

```

-- si el total de la quantitat de vendes en un dia es més baix que
el actual, substituïm
    if @aux2_vendesdia_mes > @aux_vendesdia_mes
        update st_dia_menys_vendes set dia = DAY(@data),
xifra_vendes = @aux2_vendesdia_menys where id_dia_menys_vendes =
@id_st
    else
        begin
            -- si es el mateix dia i la quantitat es més gran,
calculem quin dia té menys vendes i substituïm
            if @aux_dia_menys = DAY(@data) and
@aux2_vendesdia_menys > @aux_vendesdia_menys
                begin
                    update st_dia_menys_vendes set dia =
v.data_venda, xifra_vendes = v.quantitat
                    from st_dia_menys_vendes st, (
                        select top 1 DAY(data_venda)
data_venda, sum(quantitat_total) quantitat
                        from tw_vendes
                        where year(data_venda) = @any
                            and baixa is null
                        group by DAY(data_venda)
                        order by quantitat asc
                    ) v
                    where st.id_dia_menys_vendes = @id_st
                end
            end
        end
    end

    -- cerquem quin es l'any amb menys vendes
    set @any_historic = (select top 1 [any] from
st_dia_menys_vendes t inner join (select min(xifra_vendes)
xifra_vendes from st_dia_menys_vendes) tt on tt.xifra_vendes =
t.xifra_vendes)
    update st_dia_menys_vendes set any_dia_menys_vendes = 1 where
[any] = @any_historic

-----
-- st_ciutat_mes_beneficis
-----

-- reiniciem el historic
update st_ciutat_mes_beneficis set any_ciutat_mes_beneficis =
0

-- cerquem si n'hi ha una entrada pel any especificat
select @id_st = id_ciutat_mes_beneficis from
st_ciutat_mes_beneficis where [any] = @any
if(@id_st is null)
begin
    -- inserim un nou any
    insert into st_ciutat_mes_beneficis ([any], id_poblacio,
beneficis, any_ciutat_mes_beneficis)
    select @any, @id_poblacio,
isnull(sum(benefici_net_total), 0), 0
    from tw_vendes v
        inner join tw_botiga b on b.id_botiga = v.id_botiga
    where YEAR(data_venda) = @any
        and id_poblacio = @id_poblacio
        and v.baixa is null

    end
else
begin

```

```

-- cerquem quin es l'any amb més benefici
    set @any_historic = (select top 1 [any] from
st_ciutat_mes_beneficis t inner join (select max(beneficis)
beneficis from st_ciutat_mes_beneficis) tt on tt.beneficis =
t.beneficis)
    update st_ciutat_mes_beneficis set any_ciutat_mes_beneficis =
1 where [any] = @any_historic

-----
-- st_ciutat_menys_beneficis
-----

-- reiniciem el historic
update st_ciutat_menys_beneficis set
any_ciutat_menys_beneficis = 0

-- cerquem si n'hi ha una entrada pel any especificat
select @id_st = id_ciutat_menys_beneficis from
st_ciutat_menys_beneficis where [any] = @any
if(@id_st is null)
begin
    -- inserim un nou any
    insert into st_ciutat_menys_beneficis ([any],
id_poblacio, beneficis, any_ciutat_menys_beneficis)
select @any, @id_poblacio,
isnull(sum(benefici_net_total), 0), 0
from tw_vendes v
    inner join tw_botiga b on b.id_botiga = v.id_botiga
where YEAR(data_venda) = @any
and id_poblacio = @id_poblacio
and v.baixa is null

end
else
begin
    declare @aux_benepobla_menys decimal(18,2),
@aux_poblacio_menys int
    -- obtenim el benefici actual i el calculat per botiga
select @aux_benepobla_menys = beneficis,
@aux_poblacio_menys = id_poblacio from st_ciutat_menys_beneficis
where [any] = @any
    declare @aux2_benepobla_menys decimal(18,2)
select @aux2_benepobla_menys = sum(benefici_net_total)
from tw_vendes v inner join tw_botiga b on b.id_botiga = v.id_botiga
where year(data_venda) = @any and id_poblacio = @id_poblacio and
v.baixa is null

    -- si el benefici es més baix que el actual, substituïm
if @aux2_benepobla_menys < @aux_benepobla_menys
    update st_ciutat_menys_beneficis set id_poblacio =
@id_poblacio, beneficis = @aux2_benepobla_menys where
id_ciutat_menys_beneficis = @id_st
    else
    begin
        -- si es el mateixa poblacio i el benefici és més
gran, calculem quina poblacio té menys benfici i substituïm
if @aux_poblacio_menys = @id_poblacio and
@aux2_benepobla_menys > @aux_benepobla_menys
    begin
        update st_ciutat_menys_beneficis set
id_poblacio = v.id_poblacio, beneficis = v.benefici_net_total

```

```

from st_ciutat_menys_beneficis st, (
                                select top 1 id_poblacio,
sum(benefici_net_total) benefici_net_total
                                from tw_vendes v
                                inner join tw_botiga b on
b.id_botiga = v.id_botiga
                                where year(data_venda) = @any
                                and v.baixa is null
                                group by id_poblacio
                                order by benefici_net_total asc
) v
                                where st.id_ciutat_menys_beneficis = @id_st
                                end
                                end
                                end
                                end

-- cerquem quin es l'any amb menys benefici
set @any_historic = (select top 1 [any] from
st_ciutat_menys_beneficis t inner join (select min(beneficis)
beneficis from st_ciutat_menys_beneficis) tt on tt.beneficis =
t.beneficis)
update st_ciutat_menys_beneficis set
any_ciutat_menys_beneficis = 1 where [any] = @any_historic

end
go

```

4.4. Creació dels procediments emmagatzemats

El següent pas és crear els procediments emmagatzemats, s'han definit en quatre tipus diferents i cada tipus en un fitxer diferent en funció de la funcionalitat que tenen.

En primer lloc, tenim el fitxer número cinc que correspon a les funcionalitats transversals. En la versió actual del projecte només tenim una única funció transversal que és la traça d'execucions.

El següent fitxer conté els procediments d'alta, baixa i modificació de les taules de l'aplicació (requisit funcional RF4).

A continuació, disposem dels procediments de consulta indicats al requisit funcional RF5.

Per últim, el procediment que consulta les dades estadístiques del projecte (requisit funcional RF6).

Amb aquesta breu introducció, començarem a definir per cadascun dels tipus de procediments el nom, els paràmetres d'entrada i de sortida que tenen i una breu descripció del que fa.

4.4.1. Procediment Transversal

Procediment	trace			
Taules afectades	tw_log			
Descripció	Inserta el resultat d'una execució en la taula de registres			
Nom	Tipus	Mida	Es sortida	Descripció
proc_afectat	nvarchar	50	No	Nom del procediment executat
params	nvarchar	max	No	Nom i valors dels paràmetres fets servits en l'entrada del procediment
estat	bit		No	Indica si el resultat ha anat bé o amb error
resultat	nvarchar	200	No	Descripció del resultat
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

4.4.2. Procediments ABM

Procediment	pa_botigues			
Taules afectades	tw_botiga			
Descripció	Inserta una botiga			
Nom	Tipus	Mida	Es sortida	Descripció
id_poblacio	int		No	Identificador de la població
email_gerent	nvarchar	100	No	Email del gerent de la botiga
nombre_treballadors	int		No	Nombre de treballadors de la botiga
es_franquicia	bit		No	Indica si és franquicia
es_virtual	bit		No	Indica si és virtual
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	pm_botigues			
Taules afectades	tw_botiga			
Descripció	Edita una botiga			
Nom	Tipus	Mida	Es sortida	Descripció
id_botiga	int		No	Identificador de la botiga
id_poblacio	int		No	Identificador de la població
email_gerent	nvarchar	100	No	Email del gerent de la botiga
nombre_treballadors	int		No	Nombre de treballadors de la botiga
es_franquicia	bit		No	Indica si és franquicia
es_virtual	bit		No	Indica si és virtual
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	pb_botigues			
Taules afectades	tw_botiga			
Descripció	Esborra una botiga			
Nom	Tipus	Mida	Es sortida	Descripció
id_botiga	int		No	Identificador de la botiga
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	pa_productes			
Taules afectades	tw_producte			
Descripció	Crea un nou producte			
Nom	Tipus	Mida	Es sortida	Descripció
identificador_ean13	nvarchar	13	No	Identificador del producte
descripcio	nvarchar	200	No	Descripció del producte
preu_brut_actual	decimal	18,2	No	Preu brut que té actualment el producte
benefici_net_actual	decimal	18,2	No	Benefici que aporta el producte
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	pm_productes			
Taules afectades	tw_producte			
Descripció	Edita un producte			
Nom	Tipus	Mida	Es sortida	Descripció
identificador_ean13	nvarchar	13	No	Identificador del producte
descripcio	nvarchar	200	No	Descripció del producte
preu_brut_actual	decimal	18,2	No	Preu brut que té actualment el producte
benefici_net_actual	decimal	18,2	No	Benefici que aporta el producte
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	pb_productes			
Taules afectades	tw_producte			
Descripció	Esborra un producte			
Nom	Tipus	Mida	Es sortida	Descripció
identificador_ean13	nvarchar	13	No	Identificador del producte
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	pa_vendes			
Taules afectades	tw_vendes			
Descripció	Inserta una venda			
Nom	Tipus	Mida	Es sortida	Descripció
id_botiga	int		No	Identificador de la botiga
identificador_ean13	nvarchar	13	No	Identificador del producte
data_venda	date		No	Data de la venda
hora_venda	int		No	Hora en què s'ha produït la venda
quantitat_total	int		No	Quantitat venuda
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	pm_vendes			
Taules afectades	tw_vendes			
Descripció	Edita una venda			
Nom	Tipus	Mida	Es sortida	Descripció
id_botiga	int		No	Identificador de la botiga
identificador_ean13	nvarchar	13	No	Identificador del producte
data_venda	date		No	Data de la venda
hora_venda	int		No	Hora en què s'ha produït la venda
quantitat_total	int		No	Quantitat venuda
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	pb_vendes			
Taules afectades	tw_vendes			
Descripció	Esborra una venda			
Nom	Tipus	Mida	Es sortida	Descripció
id_botiga	int		No	Identificador de la botiga
identificador_ean13	nvarchar	13	No	Identificador del producte
data_venda	date		No	Data de la venda
hora_venda	int		No	Hora en què s'ha produït la venda
quantitat_total	int		No	Quantitat venuda
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

4.4.3. Procediments consulta

Procediment	ps_botigues_llistat			
Taules afectades	tw_vendes, tw_botiga			
Descripció	Opte les botigues que disposa la cadena en un mes i any indicat			
Nom	Tipus	Mida	Es sortida	Descripció
any	int		No	Any a comprovar
mes	int		No	Mes a comprovar
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	ps_productes_catalog			
Taules afectades	tw_productes, tw_vendes			
Descripció	Opte el catàleg de productes que disposa la cadena en un mes i any indicat			
Nom	Tipus	Mida	Es sortida	Descripció
any	int		No	Any a comprovar
mes	int		No	Mes a comprovar
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

Procediment	ps_benefici_per_dia			
Taules afectades	tw_vendes			
Descripció	Opte el benefici obtingut per dia la cadena, el producte més venut i el nombre d'unitats venudes, i la botiga que ha proporcionat més benefici i quin és el valor d'aquest benefici, tot filtrat per mes i any			
Nom	Tipus	Mida	Es sortida	Descripció
any	int		No	Any a comprovar
mes	int		No	Mes a comprovar
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

4.4.4. Prodediments estadístics

Procediment	ps_modul_estadistic			
Taules afectades	st_benefici_net, st_botiga_mes_benefici, st_producte_mes_venut, st_hora_mes_vendes, st_hora_menys_vendes, st_dia_mes_vendes, st_dia_menys_vendes, st_ciutat_mes_beneficis, st_ciutat_menys_beneficis			
Descripció	Opte les estadístiques anuals definides. Si no s'informa l'any, agafa el registre considerat com el més alt de l'històric			
Nom	Tipus	Mida	Es sortida	Descripció
any	Int		No	Any a comprovar
rsp	nvarchar	50	Si	Conté la resposta d'aquesta execució

5. Joc de proves

L'última part del procés, correspon a la part de proves de l'aplicació. S'ha dividit en dues parts diferenciades i que es troben en ordre per comprovar el correcte funcionament de cadascuna de les execucions anteriors.

El primer fitxer que trobem ens permet comprovar que les restriccions de les taules són correctes, així que es fa una primera inserció de dades en cadascuna de les taules. En cas, que alterem alguna dada que estigui relacionada amb un altre (veure definició d'entitats) possiblement ens trobaríem amb un error en executar el fitxer.

```
use TFC
set nocount on

insert into tm_regio (descripcio) values ('Tarraconensis');
insert into tm_regio (descripcio) values ('Gallaecia');
insert into tm_regio (descripcio) values ('Cartaginensis');
insert into tm_regio (descripcio) values ('Lusitania');
insert into tm_regio (descripcio) values ('Betica');

insert into tm_poblacio (id_regio, descripcio) values (1, 'Barcino');
insert into tm_poblacio (id_regio, descripcio) values (1, 'Tarraco');
insert into tm_poblacio (id_regio, descripcio) values (1, 'Valentia');
insert into tm_poblacio (id_regio, descripcio) values (2,
'Brigantium');
insert into tm_poblacio (id_regio, descripcio) values (2, 'Bracara
Augusta');
insert into tm_poblacio (id_regio, descripcio) values (2, 'Asturica
Augusta');
insert into tm_poblacio (id_regio, descripcio) values (3, 'Carthago
Nova');
insert into tm_poblacio (id_regio, descripcio) values (3, 'Toletum');
insert into tm_poblacio (id_regio, descripcio) values (3, 'Castulo');
insert into tm_poblacio (id_regio, descripcio) values (4,
'Salmantica');
insert into tm_poblacio (id_regio, descripcio) values (4, 'Augusta
Emerita');
insert into tm_poblacio (id_regio, descripcio) values (4, 'Olissipo');
insert into tm_poblacio (id_regio, descripcio) values (5, 'Sulpicia');
insert into tm_poblacio (id_regio, descripcio) values (5, 'Hispalis');
insert into tm_poblacio (id_regio, descripcio) values (5, 'Aurgi');

insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (1,
'gerente.barcino.1@mitienda.es', 10, 0, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (1,
'gerente.barcino.2@mitienda.es', 16, 1, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (1,
'gerente.barcino.3@mitienda.es', 3, 0, 1, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (2,
'gerente.tarraco.1@mitienda.es', 12, 0, 0, CAST('01/11/2015' as
datetime));
```

```

insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (3,
'gerente.valentia.1@mitienda.es', 5, 0, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (3,
'gerente.valentia.2@mitienda.es', 13, 1, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (4,
'gerente.brigantium.1@mitienda.es', 3, 0, 1, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (5,
'gerente.bracara.1@mitienda.es', 10, 0, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (6,
'gerente.asturica.1@mitienda.es', 4, 0, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (7,
'gerente.carthago.1@mitienda.es', 4, 1, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (7,
'gerente.carthago.2@mitienda.es', 8, 1, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (8,
'gerente.toletum.1@mitienda.es', 5, 0, 1, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (9,
'gerente.castulo.1@mitienda.es', 6, 1, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (10,
'gerente.salmantica.1@mitienda.es', 16, 0, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (11,
'gerente.emerita.1@mitienda.es', 12, 0, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (12,
'gerente.olissipo.1@mitienda.es', 2, 0, 1, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (13,
'gerente.sulspicia.1@mitienda.es', 4, 1, 0, CAST('01/11/2015' as
datetime));
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta, baixa) values (14,
'gerente.hispalis.1@mitienda.es', 8, 0, 0, CAST('01/11/2015' as
datetime), GETDATE());
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (14,
'gerente.hispalis.2@mitienda.es', 7, 0, 0, CAST('01/11/2015' as
datetime));

```

```
insert into tw_botiga (id_poblacio, email_gerent, nombre_treballadors,
es_franquicia, es_virtual, alta) values (15,
'gerente.aurgi.1@mitienda.es', 5, 0, 0, CAST('01/11/2015' as
datetime));
```

En aquest altre fitxer validem que els procediments emmagatzemats s'executen correctament. Podem observar com a capçalera dels resultats s'indica el tipus de valor que ha de resultar i que en executar-lo ens corrobora'n aquesta informació.

```
use TFC

set nocount on

/*
 * NOTA:
 * - Valor booleans
 *       0 - False
 *       1 - True
 */

declare @rsp nvarchar(50)

-----
-- execució errònea, població no definida
exec pa_botigues 8888, 'test@tes.es', 1, 1, 0, @rsp output
select @rsp 'pa_botigues - con error'

-- execució errònea, email no definit
exec pa_botigues 1, null, 1, 1, 0, @rsp output
select @rsp 'pa_botigues - con error'

-- execució errònea, nombre de treballador igual a 0
exec pa_botigues 1, 'test@test.es', 0, 1, 1, @rsp output
select @rsp 'pa_botigues - con error'

-- execució correcta
exec pa_botigues 1, 'test@test.es', 4, 1, 1, @rsp output
select @rsp 'pa_botigues - sense error'

-----
-- execució errònea, botiga no definida
exec pm_botigues 9999, 14, 'test@tes.es', 1, 1, 0, @rsp output
select @rsp 'pm_botigues - con error'

-- execució errònea, botiga donada de baixa
exec pm_botigues 18, 14, 'test@tes.es', 1, 1, 0, @rsp output
select @rsp 'pm_botigues - con error'

-- execució errònea, població no definida
exec pm_botigues 21, 8888, 'test@tes.es', 1, 1, 0, @rsp output
select @rsp 'pm_botigues - con error'

-- execució errònea, email no definit
exec pm_botigues 21, 1, null, 1, 1, 0, @rsp output
select @rsp 'pm_botigues - con error'
```

```

-- execució errònea, nombre de treballador igual a 0
exec pm_botigues 21, 1, 'test@test.es', 0, 1, 1, @rsp output
select @rsp 'pm_botigues - con error'

-- execució correcta
exec pm_botigues 21, 1, 'test@test.es', 4, 1, 1, @rsp output
select @rsp 'pm_botigues - sense error'

-----

-- execució errònea, botiga no definida
exec pb_botigues 9999, @rsp output
select @rsp 'pb_botigues - con error'

-- execució errònea, botiga donada de baixa
exec pb_botigues 18, @rsp output
select @rsp 'pb_botigues - con error'

-- execució correcta
exec pb_botigues 21, @rsp output
select @rsp 'pb_botigues - sense error'

-----

-- execució errònea, identificador no informat
exec pa_productes null, 'descripcio prova 2', 10.0, 1.5, @rsp output
select @rsp 'pa_productes - con error'

-- execució errònea, identificador existeix
exec pa_productes 8412345678905, 'descripcio prova 2', 10.0, 1.5, @rsp
output
select @rsp 'pa_productes - con error'

-- execució errònea, sense descripcio
exec pa_productes 4402906660009, '', 10.0, 1.5, @rsp output
select @rsp 'pa_productes - con error'

-- execució errònea, preu brut negatiu
exec pa_productes 4402906660009, 'descripcio prova', -10.0, 1.5, @rsp
output
select @rsp 'pa_productes - con error'

-- execució errònea, benefici negatiu
exec pa_productes 4402906660009, 'descripcio prova', 10.0, -1.5, @rsp
output
select @rsp 'pa_productes - con error'

-- execució correcta,
exec pa_productes 4402906660009, 'descripcio prova', 10.0, 1.5, @rsp
output
select @rsp 'pa_productes - sense error'

-----

-- execució errònea, producte no existeix
exec pm_productes 9999999999999, 'descripcio prova 2', 10.0, 1.5, @rsp
output
select @rsp 'pm_productes - con error'

```

```

-- execució errònea, producte donat de baixa
exec pm_productes 8412345678905, 'descripcio prova 2', 10.0, 1.5, @rsp
output
select @rsp 'pm_productes - con error'

-- execució errònea, sense descripcio
exec pm_productes 4402906660009, '', 10.0, 1.5, @rsp output
select @rsp 'pm_productes - con error'

-- execució errònea, preu brut negatiu
exec pm_productes 4402906660009, 'descripcio prova 2', -10.0, 1.5,
@rsp output
select @rsp 'pm_productes - con error'

-- execució errònea, benefici negatiu
exec pm_productes 4402906660009, 'descripcio prova 2', 10.0, -1.5,
@rsp output
select @rsp 'pm_productes - con error'

-- execució correcta,
exec pm_productes 4402906660009, 'descripcio prova 2', 10.0, 1.5, @rsp
output
select @rsp 'pm_productes - sense error'

-----

-- execució errònea, producte no existeix
exec pb_productes 9999999999999, @rsp output
select @rsp 'pb_productes - con error'

-- execució errònea, producte donat de baixa
exec pb_productes 8412345678905, @rsp output
select @rsp 'pb_productes - con error'

-- execució errònea, sense descripcio
exec pb_productes 4402906660009, @rsp output
select @rsp 'pb_productes - con error'

-----

-- execució errònea, botiga no informada
exec pa_vendes null, '8400000000192', '01/12/2015', 12, 1, @rsp output
select @rsp 'pa_vendes - con error'

-- execució errònea, botiga donada de baixa
exec pa_vendes 18, '8400000000192', '01/12/2015', 12, 1, @rsp output
select @rsp 'pa_vendes - con error'

-- execució errònea, producte no existeix
exec pa_vendes 20, '9999999999999', '01/12/2015', 12, 1, @rsp output
select @rsp 'pa_vendes - con error'

-- execució errònea, producte donat de baixa
exec pa_vendes 20, '8412345678905', '01/12/2015', 12, 1, @rsp output
select @rsp 'pa_vendes - con error'

-- execució errònea, data venda no vàlida
exec pa_vendes 20, '8400000000192', NULL, 12, 1, @rsp output
select @rsp 'pa_vendes - con error'

```

```

-- execució errònea, hora venda no vàlida
exec pa_vendes 20, '8400000000192', '01/12/2015', -1, 1, @rsp output
select @rsp 'pa_vendes - con error'

-- execució errònea, quantitat venuda no vàlida
exec pa_vendes 20, '8400000000192', '01/12/2015', 11, -1, @rsp output
select @rsp 'pa_vendes - con error'

-- execució correcte
exec pa_vendes 20, '8400000000192', '01/12/2015', 11, 3, @rsp output
select @rsp 'pa_vendes - sense error'

-----

-- execució errònea, botiga no informada
exec pm_vendes null, '8400000000192', '01/12/2015', 12, 1, @rsp output
select @rsp 'pm_vendes - con error'

-- execució errònea, botiga donada de baixa
exec pm_vendes 18, '8400000000192', '01/12/2015', 12, 1, @rsp output
select @rsp 'pm_vendes - con error'

-- execució errònea, producte no existeix
exec pm_vendes 20, '9999999999999', '01/12/2015', 12, 1, @rsp output
select @rsp 'pm_vendes - con error'

-- execució errònea, producte donat de baixa
exec pm_vendes 20, '8412345678905', '01/12/2015', 12, 1, @rsp output
select @rsp 'pm_vendes - con error'

-- execució errònea, data venda no vàlida
exec pm_vendes 20, '8400000000192', NULL, 12, 1, @rsp output
select @rsp 'pm_vendes - con error'

-- execució errònea, hora venda no vàlida
exec pm_vendes 20, '8400000000192', '01/12/2015', -1, 1, @rsp output
select @rsp 'pm_vendes - con error'

-- execució errònea, quantitat venuda no vàlida
exec pm_vendes 20, '8400000000192', '01/12/2015', 11, -1, @rsp output
select @rsp 'pm_vendes - con error'

-- execució errònea, no es troba ninguna venda amb aquestes
característiques
exec pm_vendes 1, '8400000000192', '01/12/2015', 11, 4, @rsp output
select @rsp 'pm_vendes - con error'

-- execució correcte
exec pm_vendes 20, '8400000000192', '01/12/2015', 11, 3, @rsp output
select @rsp 'pm_vendes - sense error'

-----

-- execució errònea, botiga no informada
exec pb_vendes null, '8400000000192', '01/12/2015', 12, @rsp output
select @rsp 'pb_vendes - con error'

-- execució errònea, botiga donada de baixa
exec pb_vendes 18, '8400000000192', '01/12/2015', 12, @rsp output
select @rsp 'pb_vendes - con error'

```



```

-- execució errònea, producte no existeix
exec pb_vendes 20, '9999999999999', '01/12/2015', 12, @rsp output
select @rsp 'pb_vendes - con error'

-- execució errònea, producte donat de baixa
exec pb_vendes 20, '8412345678905', '01/12/2015', 12, @rsp output
select @rsp 'pb_vendes - con error'

-- execució errònea, data venda no vàlida
exec pb_vendes 20, '8400000000192', NULL, 12, @rsp output
select @rsp 'pb_vendes - con error'

-- execució errònea, hora venda no vàlida
exec pb_vendes 20, '8400000000192', '01/12/2015', -1, @rsp output
select @rsp 'pb_vendes - con error'

-- execució errònea, no es troba ninguna venda amb aquestes
característiques
exec pb_vendes 1, '8400000000192', '01/12/2015', 11, @rsp output
select @rsp 'pb_vendes - con error'

-- execució correcte
exec pb_vendes 20, '8400000000192', '01/12/2015', 11, @rsp output
select @rsp 'pb_vendes - sense error'

-- execució errònea, la venda ja esta donada de baixa
exec pb_vendes 20, '8400000000192', '01/12/2015', 11, @rsp output
select @rsp 'pb_vendes - con error'

-----

exec ps_botigues_llistat 2015, 11, @rsp output
select @rsp 'ps_botigues_llistat'

exec ps_productes_cataleg 2015, 11, @rsp output
select @rsp 'ps_productes_cataleg'

exec ps_benefici_per_dia 2015, 11, @rsp output
select @rsp 'ps_benefici_per_dia'

-----

exec ps_modul_estadistic 2015, @rsp output
select @rsp 'mòdul estadístic any 2015'

exec ps_modul_estadistic null, @rsp output
select @rsp 'mòdul estadístic any 2015'

```

6. Conclusions

En el moment de donar per finalitzada la redacció dels documents d'entrega i el projecte, estic en la posició per realitzar un seguit de conclusions i valoracions de tota la feina feta.

En primer lloc, des de la meua posició crec que he assolit tots els objectius fixats prèviament encara que una petita millora proposada (casos d'ús) pel meu tutor Juan, no l'he arribat a tindre a temps. Però com diu Mies van der Roe, "menys es més".

També, cal dir que tenir una sèrie de fites al llarg del quadrimestre on cal fer l'entrega de cert material que s'ha avançat i que prèviament ha tingut el vistiplau del client (en el nostre cas el tutor), possibilita que tinguem més en compte que cada dia hem d'avançar en el projecte, no ens encantem amb altres tasques també importants del dia a dia i que d'aquesta manera, arribem a la data final amb tota la feina feta i sense el neguit de "no arribar a dia".

En aquest sentit, crec que he anat bastant bé en l'evolució del projecte, arribant a temps, encara que cal dir que en aquesta última fita he tingut diversos endarreriments que han fet que no arribi massa sobrat de temps.

Com experiència, tot aquest projecte ens aporta una visió completa important pel nostre futur laboral, constància, seriositat, són dos adjectius que han d'estar presents en el nostre diccionari personal i en la nostra forma de treballar. Si no, la feina no surt i si surt, no tindrà la qualitat que mereix.

El fet de crear un data warehouse, cal dir que ha sigut una experiència totalment diferent, i per tant, ha suposat un gran repte personal.

Per finalitzar, m'agradaria reiterar-me en la idea que aquesta experiència ha sigut molt enriquidora, però que també ha sigut llarga, n'ha necessitat molta dedicació i que per tant, s'ha de treballar intensament per arribar als objectius.

7. Glossari

Base de dades, conjunt estructurat de dades.

Base de dades relacional, conjunt estructurat de dades que a més, compleix el model relacional d'organització de dades.

Clau primària, atribut únic d'una entitat que ens permet identificar el registre inequívocament respecte a la resta de dades.

Clau forana, atribut d'una entitat que fa referència a l'atribut d'una altra entitat.

Diagrama entitat-relació (diagrama E/R), diagrama gràfic que permet visualitzar les relacions existents entre diferents entitats.

Disparadors (Triggers), funció autònoma que s'executa a conseqüència d'una acció externa i sense activitat de l'usuari.

Entitat, representació d'un objecte o concepte abstracte per definir-ho en una base de dades.

Magatzem de dades (Data Warehouse), grup de dades atomitzades amb una variant purament empresarial que permet fer una toma de decisions de manera ràpida i senzilla.

Procediments ABM, procediments emmagatzemats que permeten fer insercions, edicions o eliminacions (físiques o lògiques) de dades en taules d'una base de dades.

Procediment emmagatzemat, conjunt d'execucions que permeten abstraure a l'usuari de l'arquitectura de la base de dades i d'altra banda, consultar, editar, inserir o eliminar dades.

SQL, llenguatge estructurat que ens permet interactuar amb les bases de dades.

Taula de fets, taula que conté la informació generada donat un fet rellevant al nostre ecosistema i propi del negoci del projecte.

8. Bibliografia

Institut d'Estudis Catalans. (s.f.). Recuperat el novembre de 2015, de <http://pdl.iec.es/entrada/diec.asp>

Microsoft Corporation. (s.f.). *MSDN*. Recuperat el novembre de 2015, de [https://msdn.microsoft.com/es-es/library/bb510741\(v=sql.120\).aspx](https://msdn.microsoft.com/es-es/library/bb510741(v=sql.120).aspx)

Sánchez, J. P. (14 de enero de 2013). *Univeristat Oberta de Catalunya*. Recuperat el diciembre de 2015, de <http://hdl.handle.net/10609/18686>

Soloriio, M. (16 de abril de 2013). *Metodologia en cascada*. Recuperat l'octubre de 2015, de <http://metodologiaencascada.blogspot.com.es/>

w3schools.com. (s.f.). Recuperat el novembre de 2015, de <http://www.w3schools.com/sql/>

Wikipedia. (s.f.). Recuperat el gener de 2016, de <https://es.wikipedia.org/>

9. Annexos

9.1. Annex 1 – Instal·lació i joc de proves.

Dins del document comprimit adjunt al lliurament ens trobarem una carpeta anomenada "Scripts". Dintre d'aquesta carpeta n'hi han un seguit de fitxers enumerats de l'u al deu que ens permet fer la instal·lació i joc de proves de l'aplicació.

L'execució dels diferents fitxers ha de fer-se en l'ordre definit al número.

Tal com s'ha comentat en un capítol anterior, els fitxers quatre i deu contenen les proves de funcionament, per tant, en cas d'instal·lació en un entorn real no s'hauran d'executar i en tot cas, fer-hi una adaptació del fitxer número quatre per inserir els valors reals de la companyia client.