

Treball fi de carrera: Bases de dades

Disseny i implementació de la base de dades d'un sistema de gestió automatitzat de carteres d'inversió

Josep Unyó i Batlles

Enginyeria tècnica informàtica de gestió

Data: 15/06/15

Manel Rella Ruiz

Tutor

Agraïments

Primerament voldria agrair a la Gemma, per la seva infinita paciència i per haver-me permès dedicar un munt d'hores tots aquests anys per realitzar la enginyeria.

També voldria donar gràcies als meus fills Joan Marc i Georgina, als que, durant aquests anys, no he dedicat tot el temps de joc que haguessin volgut. La seva felicitat i energia m'il·luminen el camí. A ells els dedico el projecte.

Finalment, voldria agrair a l'equip de la UOC el suport i l'oportunitat de poder realitzar aquests estudis.

Resum del projecte

Per a finalitzar la carrera, l'estudiant ha de realitzar un projecte que posi a prova els aprenentatges adquirits en les diferents assignatures superades al llarg dels estudis d'Enginyeria Informàtica de Gestió, i que la planificació i gestió depengui del propi estudiant.

Concretament, en aquest projecte es desenvoluparà un sistema de Base de Dades (BD) adaptat a les necessitats d'un broker, dedicat a la gestió dels actius financers dels seus clients. Aquesta BD ha de permetre al broker disposar de totes les dades necessàries dels clients i la seva cartera d'inversió, per tal de poder-los assessorar correctament en les inversions.

Aquesta BD ha de ser capaç d'emmagatzemar tota la informació dels clients, de les seves accions i de les ordres de compra/venda realitzades.

També ha de permetre la realització de consultes al mòdul operacional, dels paràmetres i llistes que consideren primordials pel desenvolupament del negoci.

Finalment, ha de disposar d'un mòdul estadístic que permeti extreure'n un seguit d'informació útil pel negoci.

La programació de la la BD s'ha de fer de manera que deslliuri al client de qualsevol tasca relacionada amb el manteniment o gestió de les dades dins la BD. Entenem com a client qualsevol persona o programari que interactui amb la BD.

El projecte només se centrarà en el disseny de la BD, i no implementarà cap interfície gràfica visual, ni aplicació d'alt nivell d'accés a les dades.

La implementació es realitzarà sobre el programari Oracle Database Express Edition R11g R2.

Un cop entregat el projecte, s'hauria de continuar amb l'etapa de manteniment, ja que tant de temps com es faci servir el programari caldrà mantenir-lo, és a dir, fer canvis –petits o grans– per a corregir errors, millorar les funcions o l'eficiència, o adaptar-lo a un nou maquinari o a canvis en les necessitats d'informació.

Índex de continguts

1. - Introducció.....	9
1.1 - Objectius del projecte.....	9
1.2 - Metodologia de disseny.....	10
1.3 – Cronologia del projecte.....	11
1.4- Planificació de les tasques del projecte.....	11
1.5 – Diagrama de Gantt.....	12
1.6 - Anàlisi de riscos.....	13
1.7 – Recursos.....	14
1.7.1 - Materials.....	14
1.7.2 - Humans.....	14
1.8 – Productes obtinguts.....	14
2.- Definició i anàlisi de requisits.....	16
2.1. - Anàlisi prèvia.....	16
2.2.- Anàlisi de requisits.....	16
2.2.1- Regles de negoci.....	16
2.2.2- Funcionalitats del sistema.....	17
3 – Disseny conceptual.....	19
3.1- Model estàtic.....	19
3.1.1 – Anàlisi d'entitats: mòdul operacional.....	22
3.1.2 – Anàlisi d'entitats: mòdul data warehouse.....	23
3.1.3 – Anàlisi d'entitats: mòdul de control.....	24
3.2- Model dinàmic.....	24
3.2.1 - Actors.....	25
3.2.2 – Casos d'ús.....	25
3.2.1 – Anàlisi dels casos d'ús: mòdul operacional.....	27
3.2.2 – Anàlisi dels casos d'ús: mòdul data warehouse.....	29
3.2.3 – Anàlisi dels casos d'ús: mòdul de control.....	30
4 – Disseny lògic.....	31
4.1 – Disseny lògic del mòdul operacional.....	32
4.2 – Disseny lògic del mòdul Data Warehouse.....	33
4.2.1 – Taules de fets.....	33
4.2.2 – Taules de dimensions.....	33
4.3 – Disseny lògic del mòdul de Control.....	33
4.4 – Model ER del mòdul operacional.....	34
4.5 – Model ER del mòdul estadístic.....	35
5 – Disseny físic.....	36
5.1 – Creació de la base de dades.....	36
5.2 – Creació dels espais virtuals.....	36
5.3 – Creació dels usuaris.....	37
5.4 - Creació de les taules.....	38
5.5 - Creació de les seqüències.....	38
5.6 - Creació dels índex.....	38
6 - Implementació.....	40
6.1 – Package clients.....	40
6.1.1 - prc_client_alta.....	40
6.1.2 - prc_client_modificacio.....	41

6.1.3 - prc_client_baixa.....	41
6.1.4 - prc_client_baixa.....	42
6.2 – Package accions.....	42
6.2.1 - prc_accio_alta.....	42
6.2.2 - prc_accio_modificacio.....	42
6.2.3 - prc_accio_baixa.....	43
6.2.4 - prc_accio_baixa.....	43
6.2.5 - prc_accio_preu.....	43
6.2.6 - prc_accio_preu.....	44
6.3 – Package cartera.....	44
6.3.1 - prc_cartera_adjudicar.....	44
6.3.2 - prc_cartera_adjudicar.....	45
6.3.3 - prc_cartera_retirar.....	45
6.3.4 - prc_cartera_retirar.....	45
6.3.5 - prc_cartera_actualitzar.....	46
6.3.6 - prc_cartera_actualitzar.....	46
6.4 – Package ordres.....	46
6.4.1 - prc_ordre_alta.....	47
6.4.2 - prc_ordre_modificacio.....	47
6.4.3 - prc_ordre_cancelacio.....	48
6.4.4 - prc_ordre_execucio.....	48
6.4.5 - prc_ordre_execucio_auto.....	49
6.4.6 - prc_ordre_execucio_auto.....	49
6.5 – Package DWH.....	49
6.5.1 - fun_insDim_tdTemps.....	50
6.5.2 - prc_insDim_tdAccio.....	50
6.5.3 - prc_insDim_tdClient.....	50
6.5.4 – prc_ins_tfAccionsDisponibles.....	50
6.5.5 – prc_ins_tfAccMesNegociada.....	51
6.5.6 – prc_ins_tfCarteraValors.....	51
6.5.7 – prc_ins_tfBeneficis.....	51
6.6 – Package consultes.....	52
6.6.1 - prc_cns_operacionsClient.....	52
6.6.2 - prc_cns_operacionsAccio.....	52
6.6.3 - prc_cns_10accionsMesAdquirides.....	52
6.6.4 - prc_cns_carteraVigent.....	53
6.7 – Package consultes DWH.....	53
6.7.1 - prc_cns_beneficiMesClient.....	53
6.7.2 - prc_cns_accDisponibles.....	53
6.7.3 - prc_cns_clientCarteraMax.....	54
6.7.4 - prc_cns_accioMesNegociada.....	54
6.8 – Package control.....	54
6.8.1 - prc_logProcediment.....	54
6.8.2 - fun_Error.....	55
6.8.3 - prc_mercat_alta.....	55
6.9 – Triggers o disparadors.....	55
6.9.1 - trigHistoricAccions.....	55
6.9.2 - trigHistoricCartera.....	56
6.9.3 - trigAltaClients.....	56

6.9.4 - trigAltaAccions.....	56
7 - Fase de proves.....	57
7.1 - Test ABM clients i accions.....	57
7.2 - Test ABM ordres i cartera.....	58
7.3 – Anàlisi de les proves.....	59
8 - Costos d'accés del mòdul estadístic.....	60
8.1 - Costos prc_cns_beneficimesClient.....	60
8.2 - Costos prc_cns_accDisponibles.....	61
8.3 - Costos prc_cns_clientCarteraMax.....	61
8.4 - Costos prc_cns_accioMesNegociada.....	62
9 – Seguretat.....	64
10 – Valoració econòmica del projecte.....	66
11 – Milliores futures.....	67
12 - Conclusions.....	68
13 - Glossari.....	69
14 - Bibliografia.....	70
14.1 - Enllaços a internet.....	70
15 - ANNEXES.....	71
15.1 - Instal·lació i proves.....	71
15.1.1 – Instal·lació manual.....	71
15.2 - Anàlisi d'entitats.....	72
15.2.1 – Entitats del mòdul d'explotació.....	72
15.2.2 – Entitats del mòdul d'estadístic (Data warehouse).....	73
15.2.3 – Entitats del mòdul de control.....	75
15.3 – Anàlisi de casos d'ús.....	76
15.3.1 - Anàlisi dels casos d'ús: mòdul operacional.....	76
15.3.2 - Anàlisi dels casos d'ús: mòdul estadístic (DWH).....	80
15.3.3 - Anàlisi dels casos d'ús: mòdul de control.....	81

Índex de figures

Figura 1: Taula de cronologia, fites principals del projecte.....	11
Figura 2: Diagrames de Gantt de les diferents fases del projecte.....	13
Figura 3: Anàlisi de riscos.....	13
Figura 4: Diagrama del model estàtic conceptual del mòdul operacional.....	20
Figura 5: Diagrama del model estàtic conceptual del mòdul estadístic.....	21
Figura 6: Diagrama del model estàtic conceptual del mòdul de control.....	21
Figura 7: Fitxa anàlisi de l'entitat Ordres.....	22
Figura 8: Fitxa anàlisi de l'entitat LiniaOrdre.....	23
Figura 9: Fitxa anàlisi de l'entitat cartera.....	23
Figura 10: Diagrama casos d'ús mòdul operacional.....	26
Figura 11: Diagrama casos d'ús relacionats amb les consultes.....	26
Figura 12: Diagrama casos d'ús determinats pel mòdul de control.....	27
Figura 13: Fitxa cas d'ús actualització preu de l'acció.....	27
Figura 14: Fitxa cas d'ús adjudicar o retirar accions de la cartera.....	28
Figura 15: Fitxa cas d'ús execució manual d'una ordre de compra/venda.....	28
Figura 16: Fitxa cas d'ús execució automàtica de les ordres de compra/venda.....	29
Figura 17: Fitxa cas d'ús càlcul beneficis nets.....	30
Figura 18: Fitxa cas d'ús control d'errors.....	30
Figura 19: Diagrama del model lògic del mòdul operacional.....	31
Figura 20: Diagrama del model lògic sobre Oracle del mòdul operacional.....	34
Figura 21: Diagrama del model lògic sobre Oracle del mòdul estadístic.....	35
Figura 22: Costos planificador prc_cns_beneficimesClient.....	60
Figura 23: Costos planificador prc_cns_accionsDisponibles.....	61
Figura 24: Costos planificador prc_cns_clientcarteramax.....	61
Figura 25: Costos planificador prc_cns_clientcarteramax amb índex.....	62
Figura 26: Costos planificador prc_cns_acciomesnegociada.....	62
Figura 27: Costos planificador prc_cns_acciomesnegociada amb índex.....	63
Figura 28: Quadre valoració econòmica del projecte.....	65

1. - Introducció

Un important broker, dedicat a la gestió d'actius financers, ha obert un concurs públic en el que accepta propostes per un desenvolupament a mida d'una BD que li permeti controlar les carteres d'inversió dels seus clients.

1.1 - Objectius del projecte

Els objectius generals plantejats pel client en l'aspecte d'emmagatzemament d'informació, són els següents:

1. Ha d'emmagatzemar les dades associades als clients, incloent la composició de la seva cartera d'inversió, tant actual com històrica.
2. Ha d'emmagatzemar les ordres de compra/venda d'accions, així com el seu estat d'execució.
3. Ha d'emmagatzemar les dades associades a les accions, així com el seu històric de preus.
4. Ha d'emmagatzemar informació estadística consultable pels mateixos clients i pels analistes del broker.

Els objectius en relació a les funcionalitats són el següents:

1. Procediments d'alta, baixa i modificació (ABM) de les ordres de compra/venda, dels clients, de les carteres d'inversió.
2. Procediment d'actualització del valor de tancament diari de les accions.
3. Ha de disposar de procediments automatitzats per alimentar de dades al Data WareHouse
4. Els procediments emmagatzemats es dissenyaran de la manera següent:
 - 4.1. Han de retornar un paràmetre de sortida, que indicarà si l'execució ha finalitzat amb èxit o ha fracassat.
 - 4.2. Han de disposar de tractaments d'excepcions.
 - 4.3. S'han d'emmagatzemar totes les crides a procediments en una taula de log.

Els objectius en relació a la capacitat de consultes a la base de dades operacional són:

1. Llistar la cartera de valors d'un client, incloent el valor en euros de les accions que hi té, introduint com a paràmetres una data i un identificador de client.
2. Llistar tots els clients que hagin realitzat operacions en un any concret, introduint com a paràmetre l'any que es vol consultar.
3. Llistar totes les accions sobre les que s'hagi realitzat alguna operació (compra o venda) en un any concret, entrant com a paràmetre l'any que es vol consultar.
4. Llistar les 10 accions més adquirides per sistemes automàtics, de forma històrica, ordenades pel volum de forma decreixent.

Els objectius en relació a la capacitat de consultes al mòdul estadístic, que permeti obtenir la informació a temps constant 1, són:

1. El benefici net, en euros, obtingut per un client en un mes, introduint com a paràmetres d'entrada l'identificador del client i el mes.

Benefici net = valor cartera final de mes – valor cartera inici de mes

2. La evolució en percentatge de la cartera de valors d'un client al llarg d'un mes, introduint com a paràmetres d'entrada l'identificador del client i el mes.
Evolució mensual Cartera % = $(\text{valor final} - \text{valor inicial}) / \text{valor inicial} * 100$
3. El nombre d'accions disponibles en el moment de la consulta. S'entén com a acció disponible a les accions que estan en ordres de venda no executades ni cancel·lades.
4. El nombre de clients que tenen, en la seva cartera de valors, alguna acció que està disponible en el moment de la consulta.
5. El client amb la cartera de valors amb un valor més gran, especificant el seu valor.
6. L'acció més negociada de forma històrica i el seu volum.

Els objectius particulars que es pretenen aconseguir són els següents:

1. Dissenyar una BD robusta, que tingui les taules necessàries per contenir la informació que demana el client, evitant caure en errors de disseny.
2. Implementar els mecanismes de seguretat necessaris per evitar dades inconsistents i la realització d'operacions no permeses.
3. Dotar a la BD de les consultes necessàries, tals que retornin les dades sol·licitades.
4. Obtenir un mòdul estadístic amb dades fiables, obtingudes en temps real a partir de les taules del mòdul d'explotació.
5. Optimitzar els costos de les operacions realitzades en la BD, per tal d'obtenir el millor rendiment possible.

1.2 - Metodologia de disseny

Per a realitzar aquest projecte se seguirà un cicle de vida del programari clàssic, o altrament dit en cascada. Aquest sistema de disseny està caracteritzat per què una etapa sempre va després d'una altra, i no comença fins que l'anterior ha acabat. És a dir, que les successives etapes del desenvolupament és fan de manera consecutiva i lineal.

S'escull aquesta metodologia per què els requisits ja estan ben definits en la proposta de projecte, i no hi ha usuaris 'reals' que vagin afegint o canviant d'opinió sobre el que es vol, i per tant no hi ha necessitat de fragmentar una part dels requisits, estudiar-los, fer el disseny, provar-lo i, un cop donat per bo, fer el mateix amb un altre conjunt de requisit; és el que seria una metodologia iterativa i incremental. Un altre motiu és que disposem de molt poc personal per afrontar el projecte, i resulta més senzill de planificar.

Seguint la metodologia en cascada al final de cada etapa del disseny s'obtindrà un document resultant, que serà el punt de partida per la fase següent. Concretament tindrem els següents documents:

1. Especificació del sistema, com a resultat de l'anàlisi prèvia realitzat amb l'anàlisi de sistemes.
2. Especificació del requisits, com a resultat de l'etapa de l'anàlisi de requisits
3. Especificació del disseny, com a resultat de l'etapa de disseny
4. Especificació de les proves, que s'iniciarà en la fase de disseny i s'anirà completant en la fase de programació fins la fase de proves.
5. Manual del programari, que servirà per al manteniment futur de l'aplicació.

Per arribar al final del projecte amb èxit passarem per les següents fases o etapes del disseny:

1. Anàlisi prèvia
2. Anàlisi de requisits
3. Disseny conceptual
4. Disseny lògic
5. Disseny físic
6. Implementació
7. Prova
8. Preparació del projecte
9. Revisió del projecte

1.3 – Cronologia del projecte

Per mantenir una regularitat en el desenvolupament del projecte es proposa fer un lliurament parcial cada mes. En aquestes dates es lliurarà la documentació generada fins al moment.

TASCA	DATA INICI	DATA LLIURAMENT	DIES	HORES	CONTINGUT
PAC1	26/02/15	09/03/15	11	22	Pla de treball
PAC2	10/03/15	13/04/15	34	68	Anàlisi i disseny
PAC3	14/04/15	11/05/15	27	54	Implementació i proves
Lliurament final	12/05/15	15/06/15	34	68	Memòria, presentació i treball pràctic
Inici del Tribunal		22/06/15			
Fi del Tribunal		26/06/15			

Figura 1: Taula de cronologia, fites principals del projecte

El projecte es desenvoluparà dins el període del 26/2/15 al 15/6/15, que en total són 106 dies.

Es planteja d'inici dedicar al projecte 2 hores al dia, incloent els caps de setmana.

1.4- Planificació de les tasques del projecte

El detall de les tasques del projecte i la seva durada prevista és la següent:

PAC1- Pla de treball

- Lectura de l'enunciat del projecte: 2 hores
- Definició dels objectius: 2 hores
- Definició de les tasques per realitzar: 6 hores
- Planificació de les tasques per realitzar: 6 hores
- Concreció del diagrama de Gantt: 2 hores
- Redacció del document PAC1: 4 hores

PAC2 – Anàlisi i disseny

- Anàlisi de requisits: 2 hores
- Disseny conceptual: 15 hores
- Disseny lògic: 15 hores
- Disseny físic: 20 hores
- Recerca informació: 10 hores
- Redacció del document PAC2: 6 hores

PAC3 – Implementació i proves

- Instal·lació Oracle Express 11g R2: 8 hores
- Instal·lació Oracle SQLDeveloper: 2 hores
- Creació d'esquemes i taules: 8 hores
- Implementació de procediments i disparadors ABM: 7 hores
- Implementació de procediments de consultes: 7 hores
- Implementació de procediments i disparadors d'estadístiques: 7 hores
- Creació dels jocs de proves: 8 hores
- Documentació dels procediments: 4 hores
- Redacció del document PAC3: 6 hores

LLIURAMENT FINAL – Memòria, presentació i producte

- Realització de les proves finals: 16 hores
- Preparació instal·lació i redacció manual instal·lació producte: 16 hores
- Redacció de la memòria: 16 hores
- Redacció de la presentació: 8 hores
- Revisió global del projecte: 8 hores

1.5 – Diagrama de Gantt

La representació gràfica de la seqüenciació de les tasques del projecte és la següent:

PAC1- Pla de treball	26/02	27/02	28/02	1/03	2/03	3/03	4/03	5/03	6/03	7/03	8/03	9/03
Lectura de l'enunciat del projecte	■	■										
Definició dels objectius		■	■									
Definició de les tasques per realitzar				■	■	■						
Planificació de les tasques per realitzar							■	■	■	■		
Concreció del diagrama de Gantt										■		
Redacció del document PAC1										■	■	

PAC2 – Anàlisi i Disseny	nº setmana				
	11	12	13	14	15
Anàlisi de requisits	■				
Disseny conceptual		■	■		
Disseny lògic			■	■	
Disseny físic				■	■
Recerca informació		■	■		
Redacció del document PAC2					■

PAC3 – Implementació i proves	nº setmana				
	16	17	18	19	20
Instal·lació Oracle Express 11g R2	■				
Instal·lació Oracle SQLDeveloper	■				
Creació de taules		■	■		
Creació de procediments i disparadors			■	■	
Creació dels jocs de proves				■	
Documentació dels procediments					■
Redacció del document PAC3					■

LLIURAMENT FINAL – Memòria, presentació i producte	rº setmana				
	20	21	22	23	24
Realització de les proves finals	■	■			
Preparació instal·lació i redacció manual instal·lació producte			■	■	
Redacció de la memòria				■	
Redacció de la presentació					■
Revisió global del projecte					■

Figura 2: Diagrames de Gantt de les diferents fases del projecte

En el transcurs del projecte, s'aniran realitzant controls periòdics de l'evolució del projecte, que es contrastaran amb aquesta planificació inicial. D'aquesta forma es podran detectar ràpidament les desviacions produïdes.

1.6 - Anàlisi de riscos

En aquest apartat explicarem com podem minimitzar l'afectació en el calendari del projecte si es dona algun dels imprevistos que considerem que tenen una certa possibilitat de donar-se. Per tant, ja descartem, i no fem aparèixer, els riscos que tenen molt poques possibilitats que passin, o si passen, malauradament, no hi haurà pla de contingència vàlid.

Incidència	Risc	Impacte	Conseqüència	Pla de contingència
Avaria en el maquinari	Baix	Alt	Inhabilitació de treballar en la BD durant el temps d'avaría. Pèrdua de les dades entrades.	Es disposa d'un equip portàtil auxiliar, al que s'hi podria instal·lar la BD
Malaltia	Mig	Baix	Penalització en el calendari	Augmentar les hores destinades al projecte el cap de setmana
Avaria en el disc dur	Baix	Alt	Pèrdua de la documentació generada fins al moment i la BD	Realitzar còpies periòdiques a un repositori extern de la documentació i dels fitxers de la BD
Viatge	Baix	Mig	Inhabilitació de treballar en la BD durant el viatge	Instal·lar la BD en l'equip portàtil, per poder continuar, encara que sigui sota mínims, durant el temps del viatge

Figura 3: Anàlisi de riscos

1.7 – Recursos

Per realitzar el projecte amb garanties d'èxit, s'ha considerat necessari poder disposar dels següents recursos:

1.7.1 - Materials

Per al desenvolupament del projecte s'utilitzarà el següent maquinari i programari:

- Ordinador personal de sobretaula amb processador Intel core i3, 8Gb de Ram
- Sistema operatiu Ubuntu 12.04 LTS de 64 bits
- LibreOffice 3, per realitzar la documentació del projecte i la presentació
- La BD Oracle Express 11g R2
- Oracle SQLDeveloper per fer la implementació de la BD.
- MagicDraw per realitzar els diagrames UML

En el cas de tenir problemes amb aquest equip, es disposa d'un portàtil auxiliar amb les característiques:

- Ordinador portàtil Toshiba, amb processador Intel core I3, 8Gb de Ram
- Sistema operatiu Windows 8.1 de 64 bits

1.7.2 - Humans

Tot i que el projecte el realitzarà una mateixa persona, les tasques que s'han de realitzar les podem classificar segons els perfils professionals diferents:

- Cap de projecte: Serà el responsable de coordinar els integrants de l'equip, supervisar que el projecte es desenvolupi seguint els requisits establerts, i que es segueixi el timing definit. Serà el màxim responsable del projecte, i com a tal es responsabilitzarà de la qualitat final del mateix.
- Analista: Serà l'encarregat de realitzar els anàlisis i els dissenys, seguint els requeriments presentats pel client
- Programador: Serà l'encarregat d'implementar el disseny físic. A l'hora, també serà el responsable de realitzar les proves per validar el funcionament.
- Documentalista: Serà l'encarregat de redactar la memòria i la presentació del TFC. També s'ocuparà de realitzar el manual d'instal·lació i de funcionament de la BD.

1.8 – Productes obtinguts

Al finalitzar el projecte es lliurarà el següent contingut i informació:

Producte: Els fitxers que contenen els scripts de creació de la BD, dels usuaris, de càrrega de dades i els de realització de les proves de validació.

Memòria: Document amb la justificació del projecte, detallant els requisits, el disseny la implementació i les proves.

Annexes: Document que inclou informació de referència del projecte, que complementa i ajuda a entendre la memòria.

Presentació: Resum dels principals aspectes del projecte.

2.- Definició i anàlisi de requisits

2.1. - Anàlisi prèvia

A partir de l'anàlisi del document de requisits funcionals, que ens ha proporcionat el mateix client, veiem que els tipus de dades que haurà d'emmagatzemar són dels tipus simples (date, smallint, integer, float, character), i no caldrà emmagatzemar dades del tipus objecte gran (LOBS), com imatge, vídeo o so.

També veiem que la BD s'integrarà en un sistema OLTP, i realitzarà la funció de servidor de dades als treballadors del broker i als diferents clients. Aleshores, podem pensar que s'accedirà a les dades a través d'una aplicació connectada a la xarxa. Molt possiblement aquesta aplicació tindrà una estructura client/servidor, que encaixarà amb el tipus ab/cde o amb el tipus abc/de, segons el grau de participació que tingui el servidor o l'aplicació en l'execució de la lògica i en l'accés a les dades. Es pot dir que l'arquitectura encaixarà amb la de client lleuger i servidor pesant.

Tal com es dissenyarà la BD, encaixarà amb aplicacions amb arquitectura de 3 capes, en les que el client fa peticions d'execució de procediments emmagatzemats en el servidor, i aquests accedeixen a les dades. Aquests procediments emmagatzemats poden rebre paràmetres i retornar resultats, actuant com una capa intermèdia entre el client i les estructures de dades. La utilització dels procediments emmagatzemats millora el rendiment, incrementa la seguretat d'accés a les dades, protegeix la integritat de les dades i facilita a fer canvis de l'aplicació client.

Així mateix, de l'anàlisi del document de requisits funcionals, veiem que la base de dades no cal que sigui distribuïda, i per tant les dades podran estar emmagatzemades en un únic servidor. Això simplificarà considerablement l'estructura de la BD i els recursos de xarxa necessaris perquè funcioni correctament.

La BD estarà dividida en dos blocs funcionals. Un bloc destinat a l'explotació, que contindrà les dades vigents, i un altre bloc destinat a l'emmagatzematge de les dades històriques.

2.2.- Anàlisi de requisits

De l'anàlisi del document de requisits funcionals, proporcionat pel mateix client, extraïem informació que la classifiquem en dos grans grups: regles de negoci i funcionalitats del sistema. Aquests 2 grans blocs corresponen a la definició de les estructures de dades i a les funcionalitats per a la gestió de les estructures.

S'han plantejat al client tots els dubtes que han aparegut en l'anàlisi de requisits, i d'aquesta forma s'ha pogut definir les regles de negoci i funcionalitats sense cap ambigüitat.

2.2.1- Regles de negoci

S'han definit les següents regles de negoci, en les que ens basarem pel disseny:

R1: Dades dels clients

1. ha d'incloure les dades personals dels clients: nom, cognoms, NIF, adreça, telèfon, correu electrònic
2. ha d'incloure la cartera d'accions actual i històrica del client.

3. Un client només pot tenir una cartera d'accions, que s'anirà modificant, en funció de les ordres de compra/venda

R2: Ordres de compra venda

1. per cada ordre de compra/venda s'informarà: identificador únic, identificador del client, identificador de l'acció, el tipus de l'ordre (compra o venda), l'origen de l'ordre (manual o automàtica), la data d'entrada, l'estat de l'ordre (pendent, executada o cancel·lada), la data d'execució, quantitat d'accions afectades, preu mínim d'execució, preu obtingut.
2. Un cop una ordre s'ha executat o cancel·lat no es podrà ni modificar ni esborrar.
3. Les ordres de compra/venda podran ser gestionades (alta, baixa, modificació) pels mateixos clients de forma manual.
4. Les ordres de compra/venda podran ser gestionades (alta, baixa, modificació) per sistemes de senyals de trading automàtics. Si l'ordre ha estat executada per un sistema automàtic, cal desar-ne l'identificador.
5. Un client només pot tenir pendent una ordre per a cada acció. És a dir, no es permet que un mateix client tingui actives dues ordres (tant li fa compra o venda) associades a la mateixa acció.

R3: Accions

1. Per les accions s'informarà de: identificador (o ticker), identificador del mercat financer, nom de l'acció, preu de tancament diari.
2. Es tindrà registrat l'històric de preus de l'acció.
3. Si no afecta al pressupost del projecte, es considerarà deixar el sistema preparat per tractar amb altres tipus de productes financers.

2.2.2- Funcionalitats del sistema

També hem definit els següents requisits funcionals del sistema:

R4: Funcionalitats

1. Procediments ABM (alta, baixa i modificació) de clients, ordres de compra/venda i d'accions.
2. Procediments per actualitzar el valor de tancament diari de les accions. La càrrega dels valors de les accions es realitzarà des d'un fitxer de proves SQL, que realitzarà les crides a un procediment emmagatzemat.
3. No cal implementar procediments ABM dels identificadors de mercat, ni dels senyals de trading.
4. Procediment de consulta, que donat una data i un identificador de client, com a paràmetres d'entrada, retorni un llistat amb la cartera d'accions del client, vigent en la data indicada, incloent-hi el valor total en euros de les accions. També es demana que retorni el sumatori de tots els totals.
5. Procediment de consulta, que donat un any, com a paràmetre d'entrada, retorni el llistat de tots els clients que hagin realitzat operacions en aquell any. Per a cada client s'informarà: identificador client, nom client, nombre ordres compra carregades durant l'any, nombre d'ordres de compra executades durant l'any, sumatori – en euros- de totes les ordres de compra executades, valor mig – en euros- de cada operació de compra executada, nombre

d'ordres de venda carregades durant l'any, nombre d'ordres de venda executades durant l'any, sumatori -en euros- de les ordres de venda executades, valor mig -en euros- de cada operació de venda executada, nombre d'ordres de compra cancel·lades durant l'any i nombre d'ordres de venda cancel·lades durant l'any.

6. Procediment de consulta, que donat un any com a paràmetre d'entrada, retorni el llistat de totes les accions sobre les que s'hagi realitzat alguna operació (compra o venda). Per a cada acció s'informarà del identificador de l'acció, del seu nom, de nombre d'operacions de compra executades durant l'any, del sumatori – en euros- de les ordres de compra realitzades sobre l'acció, del valor mitjà -en euros- de les ordres de compra executades sobre l'acció, del nombre d'operacions de venda executades durant l'any, del sumatori – en euros- de les ordres de venda executades sobre l'acció, del valor mitjà – en euros- de les ordres de venda executades sobre l'acció, de la revalorització de l'acció en % des del 1 de gener de l'any indicat.
7. Procediment de consulta que llisti les 10 accions més adquirides pels sistemes automàtics, ordenades de forma decreixent pel seu volum.

R5: Funcionalitats del mòdul estadístic

1. El mòdul estadístic s'ha d'alimentar de les dades del mòdul d'explotació.
2. Les consultes a les dades del mòdul estadístic hauran de donar resposta immediata, amb un un cost de procés igual a 1.
3. Procediment de consulta, que donat un identificador de client i un mes concret (anterior a l'actual), retorni el benefici net, en euros, i el % de la cartera d'accions durant aquell mes.
 1. Definim benefici net:
$$\text{Benefici net} = \text{valor cartera inici mes} - \text{valor cartera a final de mes}$$
 2. Definim benefici en %
$$\% \text{ Benefici} = \text{valor inici} / \text{Valor final} * 100$$
4. Procediment de consulta, que donat un identificador d'una acció, retorni el nombre d'accions disponibles (lligades a ordres de venda en situació pendent) dins el broker en aquell moment i el nombre de clients que la tenen en la cartera d'accions de forma activa.
5. Procediment de consulta que retorni el client amb la cartera d'accions mes gran, i el valor en euros de la cartera.
6. Procediment de consulta que retorni l'acció més negociada (comprada i venuda) en el broker de forma històrica, i el seu volum

R6: Funcionalitats transversals

1. Ha de disposar de procediments automatitzats per alimentar de dades el Data Warehouse
2. Els procediments sempre han de retornar un paràmetre, de tipus string, anomenat RSP, que indicarà si el procediment ha finalitzat amb èxit o amb error.
3. Els procediments disposaran de tractaments d'excepcions que finalitzin de forma controlada els procediments en cas de donar-se un error.
4. Totes les crides als procediments s'emmagatzemaran en un registre Log, que permeti anar a consultar quin s'ha executat, amb quins paràmetres, què ha retornat, i si ha finalitzat o no amb èxit.

3 – Disseny conceptual

A partir de l'anàlisi de requisits obtenim la informació necessària per descriure l'estructura de les dades emmagatzemades en la BD, així com les interaccions entre els objectes dins la BD. El primer punt es refereix al model estàtic, i el segon punt, al dinàmic.

Aquests esquemes conceptuals són independents de la tecnologia d'implementació, i donen una visió clara i simple a l'analista, però propera també al usuari, de les relacions entre les diferents dades.

3.1- Model estàtic

Representem, utilitzant diagrames UML, el model estàtic dels diferents mòduls que tindrà la BD: el mòdul operacional, el mòdul data warehouse i el mòdul de control.

El mòdul operacional defineix les relacions necessàries per emmagatzemar la informació viva del negoci. En aquest mòdul es permet que les consultes no siguin a temps constant, així com la realització d'agrupament i càlculs al vol. Aquesta part de la base de dades tindrà un clar perfil transaccional, i es pot considerar com un sistema OLTP (On-Line Transactional Processing). Es realitzarà un accés freqüent a les dades, tant per tasques de lectura com d'escriptura. Per tant, aquesta part de la base de dades tindrà un funcionament típic de les BD operacionals. Es pot veure el diagrama conceptual d'aquest mòdul en la figura 4.

El mòdul del Data Warehouse defineix les relacions necessàries per obtenir la informació estadística que s'ha decidit, i en temps constant. No es permetrà realitzar càlculs ni agrupaments al vol. Consta de les taules de dimensió i les taules de fets. Aquesta part de la BD té una clara orientació al processament analític, i es pot considerar com un sistema OLAP (On-Line Analytical Processing). S'accedirà a grans quantitats de dades, habitualment només en mode lectura, per tal d'obtenir informació útil pel negoci. Les dades s'emmagatzemen en forma històrica, i s'obtenen dels sistemes operacionals. La figura 5 representa el model conceptual del mòdul estadístic.

El mòdul de control defineix les relacions necessàries per gestionar les dades de funcionament, i de comandament de la BD. Es pot veure el model conceptual del mòdul de control en la figura 6.

S'ha realitzat una anàlisi detallada de les entitats indicades en els diferents diagrames conceptuals, en el que s'ha especificat les característiques principals de cada una. Però per qüestions de claredat, en aquest apartat només s'indiquen les que s'ha considerat més significatives. L'anàlisi de la resta d'entitats es pot trobar en el corresponent apartat de l'annexe.

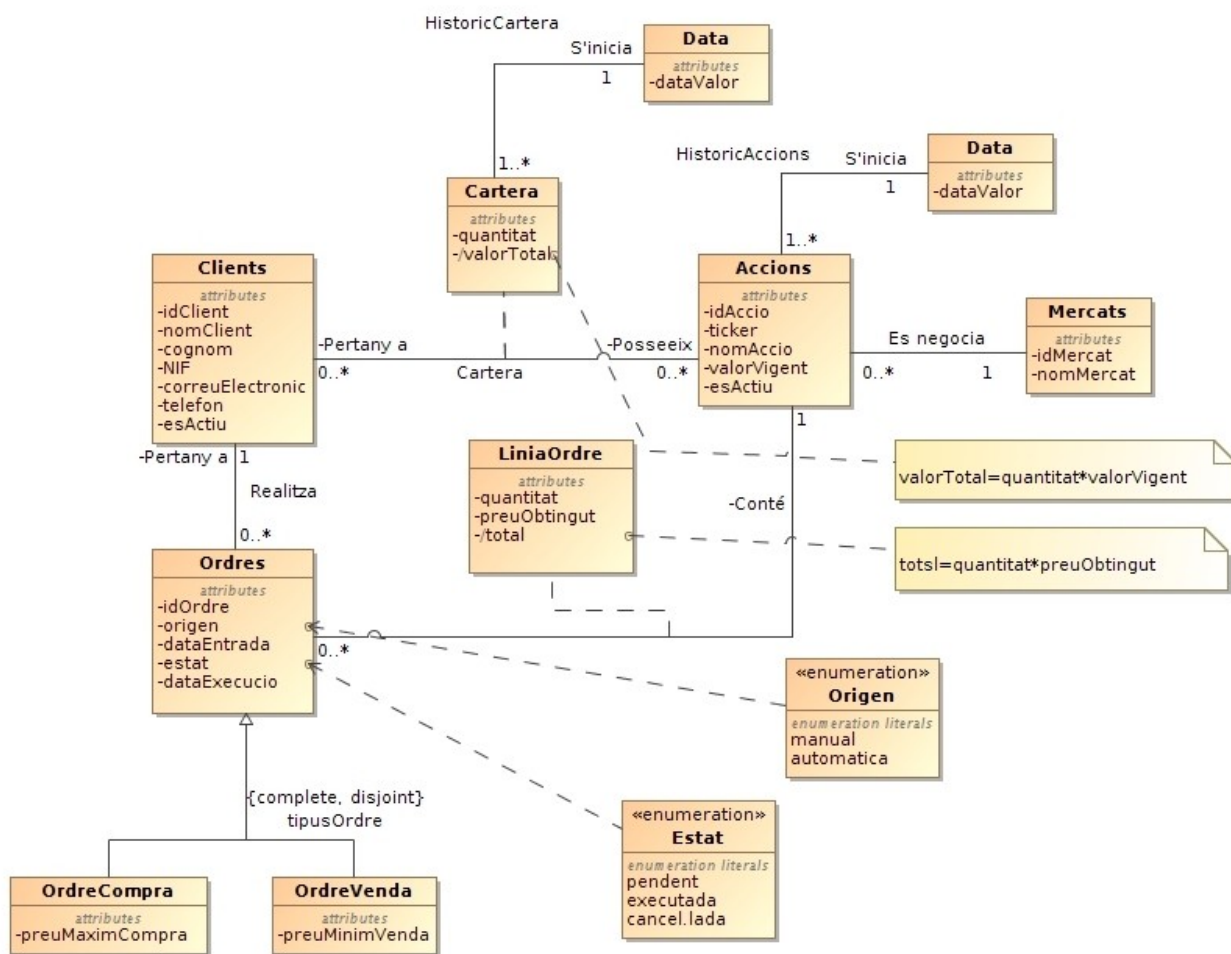


Figura 4: Diagrama del model estàtic conceptual del mòdul operacional

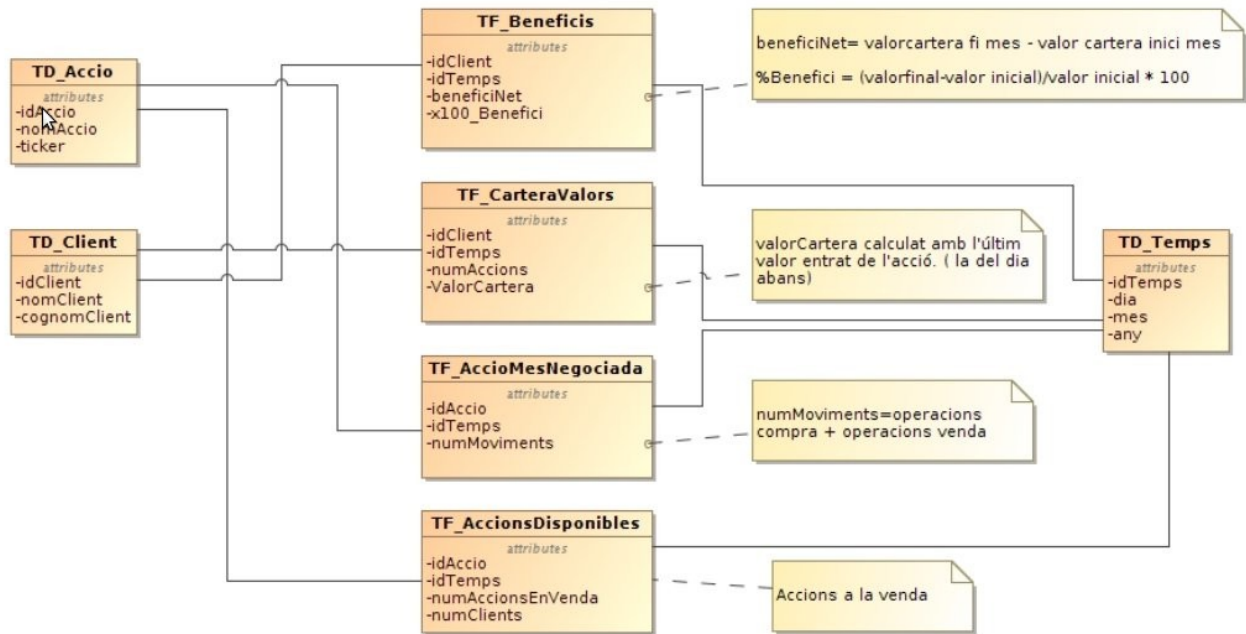


Figura 5: Diagrama del model estàtic conceptual del mòdul estadístic



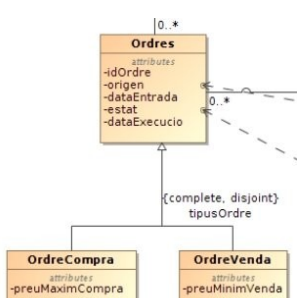
Figura 6: Diagrama del model estàtic conceptual del mòdul de control

3.1.1 – Anàlisi d'entitats: mòdul operacional

Les entitats del mòdul operacional representen els objectes que participen en el funcionament diari de la BD. Les transaccions habitualment es realitzaran sobre aquestes entitats. Les taules que representen aquestes entitats contindran informació de les altes, baixes i modificacions, i al tractar-se d'una BD transaccional, sofriran canvis constants.

Classe	Ordres
Descripció de la classe	Classe que representa la informació associada a cada ordre de compra/venda d'accions
Responsabilitats	Col·laboracions/col·laboradors
Contenir la informació que caracteritza a les ordres	Col·labora amb la classe d'enumeració Tipus, Origen i Estat, per definir els valor possibles dels camps corresponents.
Realitzar ordres	Col·laboració amb les classes Clients
Conté accions	Col·laboració amb la classe Accions per definir quines accions es compren/venen
Quantifica	Col·laboració amb la classe LiniaOrdre per definir la quantitat d'accions associades a cada ordre.

Figura 7: Fitxa anàlisi de l'entitat Ordres

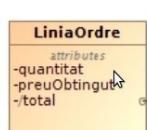


No es permet valor nuls en el camp dataEntrada.
 No es permet valor nuls en els camps tipusOrdre, origen i estat.
 No es permet valor nuls en els camps dataExecució si l'estat és executada
 tipusOrdre només admet valors: compra/venda
 origen només admet valors: manual/auto
 estat només admet valors: pendent/executat/cancel.lat
 En la transformació al model lògic, es tindrà en compte que no es permet que un client tingui en estat 'pendent', més d'una ordre amb la mateixa acció.

També es considerarà transformar la relació de generalització/especialització en una sola relació que inclogui tots els atributs de les subclasses. Això evitarà haver de fer operacions de combinació entre la taula superclasse i les subclasses (join), i això ens donarà un millor rendiment a les consultes realitzades sobre la taula. Com que és una relació disjunta, transformarem els 2 camps en un de sol anomenat PreuObjectiu, que servirà per compres i per vendes.

Classe	LiniaOrdre
Descripció de la classe	Classe que representa el número d'accions que es volen comprar/vendre
Responsabilitats	Col·laboracions/col·laboradors
Informar de la quantitat d'accions que intervindran en l'operació	Col·labora amb la classe ordre i accions per definir la quantitat
Informa d'altres atributs dependents de l'acció	Col·laboració amb les classes Clients i Accions

Figura 8: Fitxa anàlisi de l'entitat LiniaOrdre



Classe associativa que afegeix informació a la relació entre Ordres i Accions. Quantitat no pot ser null, ja que no es permetrà una ordre sense cap acció associada

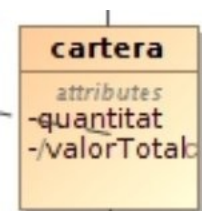
El preu obtingut serà el valor final de venda o de compra a la que es negociarà l'ordre.

L'atribut total serà un camp calculat, resultat del preuObtingut * quantitat.

En la transformació al model lògic es fusionarà amb la taula Ordres, ja que la relació és 1:1.

Classe	Cartera
Descripció de la classe	Classe que representa el grup d'accions que té en propietat un client.
Responsabilitats	Col·laboracions/col·laboradors
Informar de la quantitat d'accions que disposa el client en cartera	Col·labora amb les classes Client i Accions, per definir la cartera
Registra la cartera de valors de forma històrica	Col·laboració amb la classe Data per determinar en quin període de temps una acció pertany al client

Figura 9: Fitxa anàlisi de l'entitat cartera



És una classe associativa que afegeix informació a la relació entre clients i accions. Quantitat pot ser null, ja que es permet tenir clients que no tinguin accions en cartera.

ValorTotal és un camp calculat que conté el valor total vigent del grup d'accions. $valorTotal = valor\ diari\ acció * quantitat$

3.1.2 – Anàlisi d'entitats: mòdul data warehouse

Les classes del data warehouse emmagatzemen informació que serà d'interès per a l'anàlisi del negoci, i serà utilitzada en la presa de decisions. És informació referida a moments puntuals, que ens permetran reconstruir els fets històrics amb consultes simples. Són com instantànies de la realitat agafades en un moment concret, que un cop agafades no es modifiquen més.

Aquestes classes es divideixen en taules de fets (TF), que contenen les dades principals, quantitatives, que utilitzaran els analistes en la presa de decisions, i en taules de dimensions (TD), que proporcionen informació addicional que descriu alguna de les característiques de les taules de fets. És a dir, les taules de dimensions contenen els atributs que s'utilitzaran per restringir o agrupar les dades emmagatzemades en les taules de fets quan es realitzen les consultes.

El model de dades que s'ha plantejat pel mòdul del data warehouse és del tipus anomenat esquema en estrella, en el que cada taula de fets té una relació directa amb una o més taules de dimensions, a través de les seves claus. És un esquema que procura eliminar al màxim les join entre taules a l'hora de realitzar les consultes, per tal d'obtenir un millor temps de resposta. Aquesta estructura proporciona un esquema senzill d'interpretar i de fer consultes, que el fa àmpliament utilitzat.

En aquest projecte s'han definit les següents taules de dimensions:

TD_Temps, TD_client, TD_accio;

i hem definit les següents taules de fets:

TF_beneficis, TF_CarteraValors, TF_AccióMesNegociada i TF_AccionsDisponibles.

Les taules de fets s'han definit amb 2 dimensions, una sempre és la dimensió temps, i l'altre serà la dimensió clients o la dimensió accions, segons correspongui.

És important remarcar que la taula que representa la dimensió temps s'ha definit de manera que la clau primària és un valor numèric, amb el format 'yyyymmdd', ja que els valors numèrics tenen un millor rendiment a les consultes i ocupen menys espai. A més, per completar la perspectiva de temps, s'han definit els atributs any, mes i dia, representats també en valors numèrics. Aquesta definició farà que les consultes referides als atributs anys, mes o dia tinguin un rendiment molt bo, ja que no caldrà realitzar cap conversió. També és important dir, que s'ha considerat que no calia definir més atributs qualificadors a la data ja que l'enunciat no ho demanava, però es podria haver afegit fàcilment a la relació atributs del tipus: número de trimestre, setmana, dia de la setmana, etc.

3.1.3 – Anàlisi d'entitats: mòdul de control

Les entitats del mòdul de control donaran suport al funcionament global de l'aplicació. Ens serviran per registrar informació comuna a la resta d'entitats, i podran ser utilitzades per totes elles.

S'han definit les entitats LogProcediments i ErrorsReconeguts.

L'entitat LogProcediments respon al requisit plantejat per l'enunciat del projecte de registrar les crides als procediments. En canvi l'entitat ErrorsReconeguts no respon a cap requisit de l'enunciat, sinó que es defineix com a solució per obtenir unes respostes centralitzades en els missatges d'error generats. Aquesta forma presenta diverses avantatges com: unificació dels missatges, coherència de missatges, facilitat d'implementar nous missatges, facilitat de modificar o traduir els missatges.

3.2- Model dinàmic

En aquest apartat es representen els diagrames de casos d'us del model del negoci (BD operacional), del bloc de consultes, del data Warehouse i del bloc de control. Identificant aquests casos d'us, s'ha obtingut gran part de la informació necessària per determinar les funcionalitats que s'hauran d'implementar-se en la BD., ja que els casos d'ús ens indicaran les interaccions entre els actors externs i el sistema. Amb l'anàlisi d'aquestes interaccions es podrà fer un planteig de les funcions que ha de realitzar el sistema.

3.2.1 - Actors

A partir de l'anàlisi de requisits, es poden distingir 5 actors que actuaran amb el sistema. Aquests actors s'autenticaran directament contra el SGBD. Es definiran un conjunt de rols, amb els seus permisos corresponents, als que s'hi assignaran els diferents actors d'acord a les seves necessitats, en vista a facilitar el manteniment de permisos de la BD.

Els actors identificats són:

- **Broker:** serà l'encarregat del manteniment i gestió de les dades de clients i de les accions. També podrà crear noves ordres de venda i podrà executar o cancel·lar les ordres de compra/venda que estiguin pendents. Actuarà com a especialització d'un agent borsari, amb permisos per actualitzar el valor de les accions.
- **Agent borsari:** és un actor que només tindrà la missió d'actualitzar els valors de les accions.
- **Client:** serà l'encarregat del manteniment i gestió de les ordres de compra (altes, baixes, modificacions i execucions).
- **Sistema de Trading:** és un actor que apareix com a especialització del client, i que podrà fer les mateixes accions que els clients.
- **Sistema:** Aquest rol el realitzarà el mateix sistema. S'encarregarà de mantenir la integritat de les dades, amb l'ajuda de disparadors i procediments, i preservar la seguretat de les dades. També s'encarregarà de realitzar els processos necessaris per alimentar de dades i mantenir el mòdul estadístic. Una altra funció serà la de registrar totes les operacions i execucions dels diferents processos que modifiquen les dades, a dins les taules Log.

3.2.2 – Casos d'ús

Els diagrames de casos d'ús ens resumeixen les interaccions del sistema amb l'exterior. Després d'un anàlisi preliminar s'ha vist que els casos d'ús es podien agrupar en 3 grans blocs: bloc de gestió (o operatiu), bloc de consultes (que englobi les consultes a la part operacional i a la part estadística) i un bloc de control del sistema.

La figura 10 ens mostra el diagrama dels casos d'ús del bloc de gestió. Principalment està format per les interaccions de ABM.

La figura 11 ens mostra els casos d'ús determinats pel mòdul de consultes, que engloba les realitzades dins el mòdul d'explotació i les realitzades dins el mòdul del data warehouse.

Per finalitzar, en el diagrama de la figura 12 veiem els casos d'ús determinats pel mòdul de control.

S'ha fet una anàlisi en detall dels casos d'ús dels mòduls principals: operacional i estadístic.

No s'han analitzat els casos d'ús corresponents a les consultes, per considerar que no aporten valor, ja que en tots els casos es tractarà d'accedir al mòdul de consultes i executar un procediment que retornarà un valor o conjunt de registres.

Per facilitar la comprensió, només s'han deixat en aquest apartat les fitxes dels casos d'ús més significatius. La resta es poden trobar en l'apartat corresponent de l'annexe.

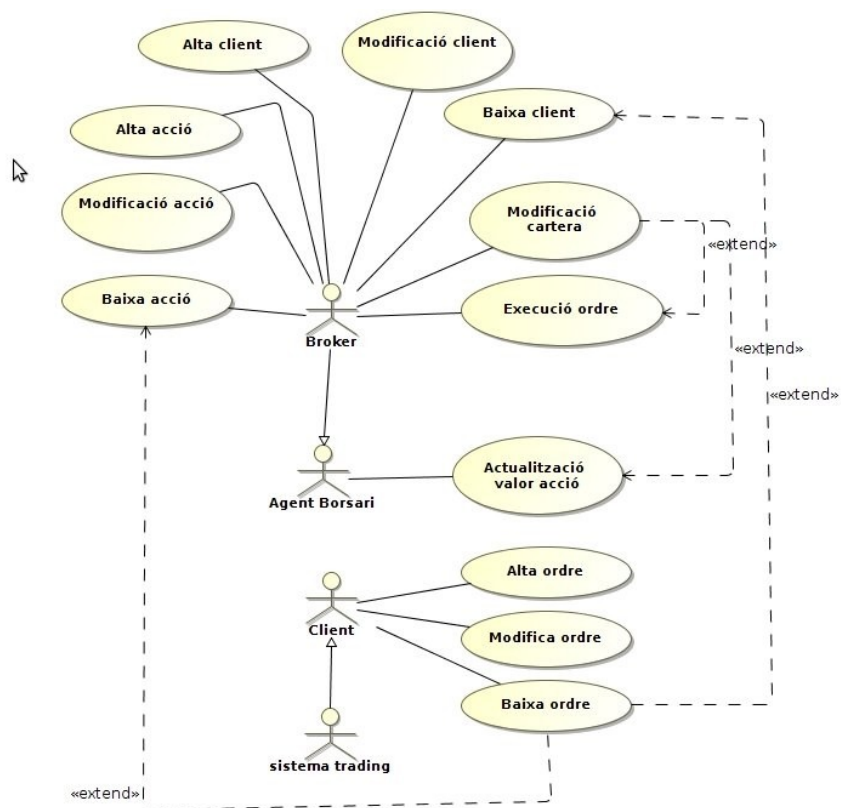


Figura 10: Diagrama casos d'ús mòdul operacional



Figura 11: Diagrama casos d'ús relacionats amb les consultes

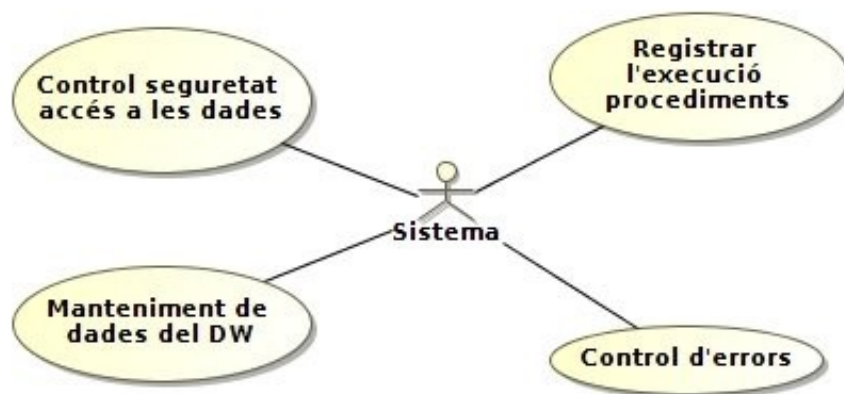


Figura 12: Diagrama casos d'ús determinats pel mòdul de control

3.2.1 – Anàlisi dels casos d'ús: mòdul operacional

En aquest apartat s'analitzen les interaccions que fan els diferents actors amb el sistema d'explotació. Aquest anàlisi donarà una visió clara de quines entrades té el sistema, i quines són les sortides esperades.

Cas d'us	007 – Actualització del preu de l'acció
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol canviar el preu d'una acció
Precondició	L'usuari ha de tenir obligatòriament el rol de broker. L'acció ha d'estar activa.
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema actualitzarà el preu de l'acció, i ho registrarà a la taula log
Escenari principal d'èxit	1- s'introdueixen el identificador o ticker de l'acció que es vol actualitzar, i el nou preu 2- actualitza el preu de la Taula Accions 3- Insereix un registre a la taula Històric accions 4- S'executen les ordres de compra/venda que compleixen preu Objectiu 5- S'actualitza el valor de la cartera 6- S'insereix un registre a la taula Històric Cartera 7- el sistema registra el moviment a la taula de log
Extensions	--

Figura 13: Fitxa cas d'ús actualització preu de l'acció

Cas d'us	008 – Adjudicar o retirar accions de la cartera
Actor	Sistema
Àmbit	BD operativa
Nivell d'objectiu	Es vol actualitzar, modificar la cartera d'accions assignada a un client
Precondició	S'ha d'haver executat una ordre de compra/venda
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema registra les accions i la quantitat al client, i ho registrarà a la taula log
Escenari principal d'èxit	<ol style="list-style-type: none"> 1- s'introdueix el identificador del client, l'identificador de l'acció i la quantitat que es vol associar a la cartera 2- el sistema verifica si el client ja té accions del tipus entrat <ol style="list-style-type: none"> 2a – si es volen adjudicar accions d'un tipus que el client ja té, se li sumen. 2b- si es volen treure accions d'un tipus que el client no té, o se'n volen treure més de les disponibles, ha de retornar un missatge d'error 3- el sistema actualitza la cartera d'accions 4- el sistema registra el moviment a la taula de log 5- S'insereix un nou registre en la taula Històric Cartera
Extensions	

Figura 14: Fitxa cas d'ús adjudicar o retirar accions de la cartera

Cas d'us	012 – Execució manual d'una ordre
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol executar una ordre que està en situació PD.
Precondició	L'ordre ha d'estar PD
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema canviarà l'estat de l'ordre, actualitzarà la cartera de valors del client , i ho registrarà a la taula log
Escenari principal d'èxit	<ol style="list-style-type: none"> 1- s'introdueix el identificador de l'ordre que es vol executar. 2- el sistema canvia d'estat l'ordre a executada 3- el sistema actualitza la cartera de valors del client 4- s'actualitza el històric cartera 5- si l'ordre era de venda s'actualitza la TF_accions disponibles 6- S'actualitza a TF_cartera de valors 7- S'actualitza la TF_beneficis 8-S'actualitza TF_accioMesNegociada 9- el sistema registra el moviment a la taula de log
Extensions	

Figura 15: Fitxa cas d'ús execució manual d'una ordre de compra/venda

Cas d'us	013 – Execució automàtica de les ordres
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol executar un conjunt d'ordres pendents que compleixen la condició de preu objectiu
Precondició	
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema canviarà l'estat de l'ordre, actualitzarà la cartera de valors del client , i ho registrarà a la taula log
Escenari principal d'èxit	1- el sistema fa una cerca de les ordres que compleixen el requisit del preu mínim entrat, comparant-lo amb el valor de les accions. 2- per a cada ordre que compleix el requisit s'aplica el cas d'ús 012
Extensions	

Figura 16: Fitxa cas d'ús execució automàtica de les ordres de compra/venda

3.2.2 – Anàlisi dels casos d'ús: mòdul data warehouse

S'interpreta que el mòdul DWH és un mòdul extern al mòdul operacional. Per això s'analitzen els casos d'ús que actuen contra el DWH, i es considerarà al sistema de procediments del mòdul operacional com un actor extern al DWH.

Es configura el sistema per què al passar determinats esdeveniments s'activin els procediments encarregats de mantenir el data warehouse actualitzat. La responsabilitat d'execució d'aquests casos d'ús recau en el propi sistema.

Cas d'us	014 – Actualització TF_Beneficis
Actor	Sistema
Àmbit	BD estadística
Nivell d'objectiu	Es vol actualitzar la taula de fets de beneficis, que conté els beneficis mensuals acumulatius de cada client.
Precondició	S'ha actualitzat la cartera de valors d'un client o s'ha actualitzat el valor de les accions. Això passa al executar una ordre i al actualitzar preus.
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema calcula els beneficis nets des de l'inici de mes fins a la data del moviment, i l'increment en %, actualitzarà la taula TF_Beneficis, i ho registrarà a la taula log del DWH. S'agafa la primera data del mes en la que hi ha valor entrat per fer el càlcul.
Escenari principal d'èxit	1- Per a cada client que té accions associades a la seva cartera, el sistema recupera el valor de la cartera en la primera data del mes en la que hi ha valor entrat, i el valor en la última data entrada del mes. Amb aquestes dades en calcula la diferència, i l'increment en %.

	2- Actualitza la TF_Beneficis pel client i data concrets 3- El sistema registra el moviment a la taula de log
Extensions	

Figura 17: Fitxa cas d'ús càlcul beneficis nets

3.2.3 – Anàlisi dels casos d'ús: mòdul de control

El mòdul de control també s'interpretarà com un mòdul extern al mòdul d'explotació i al mòdul DWH. Per això, des d'aquest punt de vista, el mòdul d'explotació i el DWH seran actors externs que interactuaran, i executaran els seus processos.

Cas d'us	020 – Control d'errors del sistema
Actor	Sistema
Àmbit	BD, sistema de control
Nivell d'objectiu	Centralitzar els missatges d'error produïts en l'execució dels procediments, per donar resposta ordenada, coherent i unificada.
Precondició	
Garanties mínimes	El sistema donarà una resposta centralitzada i unificada als diferents errors produïts.
Garanties cas d'èxit	S'haurà llançat el missatge d'error esperat.
Escenari principal d'èxit	1- s'executa un procediment que dona error 2- es captura l'error 3- es consulta a la taula d'errors reconeguts quina resposta s'ha de donar. 4- es llança el missatge d'error corresponent.
Extensions	

Figura 18: Fitxa cas d'ús control d'errors

4 – Disseny lògic

A partir de l'anàlisi del disseny conceptual, format pel diagrama estàtic i per les fitxes de cada classe, s'ha traduït en un nou diagrama estàtic que representa el model lògic de les relacions entre les dades. En aquest model, ja es té en compte les particularitats del SGBD que s'utilitzarà per implementar. És un model més proper al programador i al sistema. També s'ha realitzat la definició de forma textual de totes les taules. Per a la descripció textual del model lògic s'utilitza la següent notació:

- Es subratlla les claus primàries amb línia contínua.
- Les claus alternatives es descriuen de forma textual.
- Es posa en negreta els atributs que no admeten valors nuls.
- Les claus foranes es descriuen de forma textual.

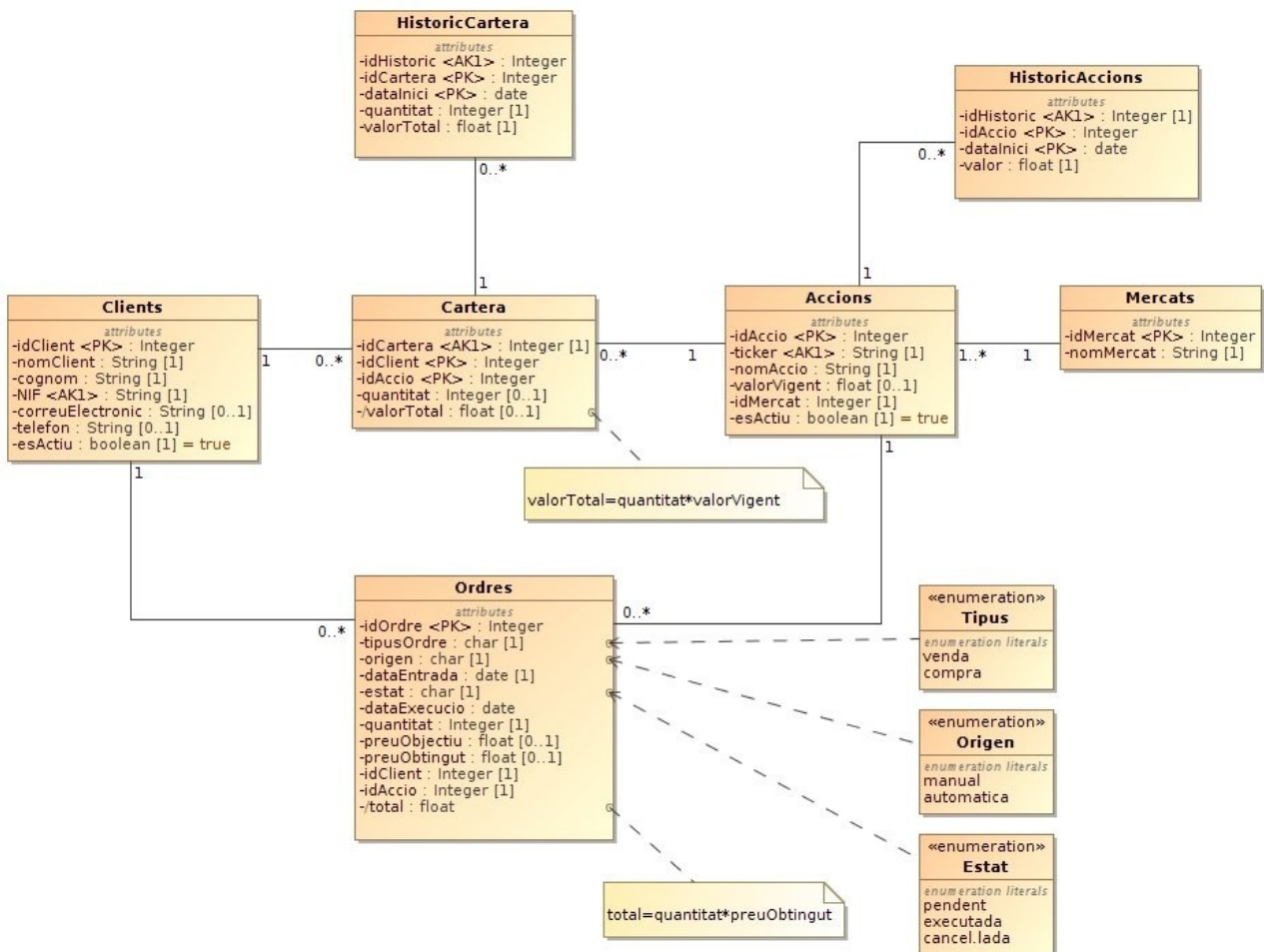


Figura 19: Diagrama del model lògic del mòdul operacional

4.1 – Disseny lògic del mòdul operacional

Clients (idClient, **nomClient**, cognom, **nif**, correuElectronic, telefon, esActiu)
on {nif}és clau alternativa

Accions (idAccio, **ticker**, **nomAccio**, valorVigent, Mercat, esActiu)
on {ticker}és clau alternativa
on {Mercat} és clau forana de Mercats(idMercat)

Mercats (idMercat, **nomMercat**)
on {nomMercat}és clau alternativa

Cartera (idCartera, Client, Accio, **quantitat**, /valorTotal)
on {idCartera} és clau alternativa
on {/valorTotal} és un atribut derivat

HistoricCartera (idHistoricCartera, **Cartera**, **quantitat**, **valor**, **dataInici**, **operacio**)
on {Cartera, dataInici} és clau alternativa
on {Cartera} és clau forana de Cartera(idCartera)
on {operacio} pot pendre els valors Alta, Modificació i Baixa

HistoricAccions (idHistoricAccio, **Accio**, **valor**, **dataInici**)
on {Accio, dataInici} és clau alternativa
on {Accio} és clau forana de Accions(idAccio)

Ordres (idOrdre, **Client**, **Accio**, **tipusOrdre**, **origen**, **dataEntrada**, **estat**, dataExecucio, **quantitat**, preuObjectiu, preuObtingut, esActiu)
on {tipusOrdre} pot pendre els valors de COMPRA, VENDA
on {origen} pot pendre els valors de MAN (manual) o AUT (automàtic)
on {estat} pot pendre els valors de PD (pendent), EX (executat) o CA (cancel.lat)
on {Client} és clau forana de Clients(idClient)
on {Accio} és clau forana de Accions(idAccio)

4.2 – Disseny lògic del mòdul Data Warehouse

El disseny lògic del magatzem de dades està definit per les taules de fets i per les taules de dimensions. Les taules de fets registren els valors que voldrem analitzar. Les taules de dimensions registren els valors que donaran sentit i profunditat als fets.

4.2.1 – Taules de fets

TF_Beneficis (idClient, idTemps, **beneficiNet**, **beneficix100**)

on {idClient} és clau forana de TD_Client(idClient)

on {idTemps} és clau forana de TD_Temps(idTemps)

TF_CarteraValors (idClient, idTemps, **numAccions**, **valorCartera**)

on {idClient} és clau forana de TD_Client(idClient)

on {idTemps} és clau forana de TD_Temps(idTemps)

TF_AccioMesNegociada (idAccio, idTemps, **numMoviments**)

on {idAcció} és clau forana de TD_Accio(idAccio)

on {idTemps} és clau forana de TD_Temps(idTemps)

TF_AccionsDisponibles (idAccio, idTemps, **numAccionsEnVenda**, **numClients**)

on {idAcció} és clau forana de TD_Accio(idAccio)

on {idTemps} és clau forana de TD_Temps(idTemps)

4.2.2 – Taules de dimensions

TD_Accio (idAccio, **nomAccio**)

TD_Client (idClient, **nomClient**, **cognom**, **NIF**)

TD_Temps (idTemps, **dia**, **mes**, **any**)

4.3 – Disseny lògic del mòdul de Control

LogProcediments (id, **procedimentExecutat**, parametresEntrada, parametresSortida, **data**)

on {procedimentExecutat, data} és clau alternativa

Errors_reconeguts (idError, **descripcio**)

4.4 – Model ER del mòdul operacional

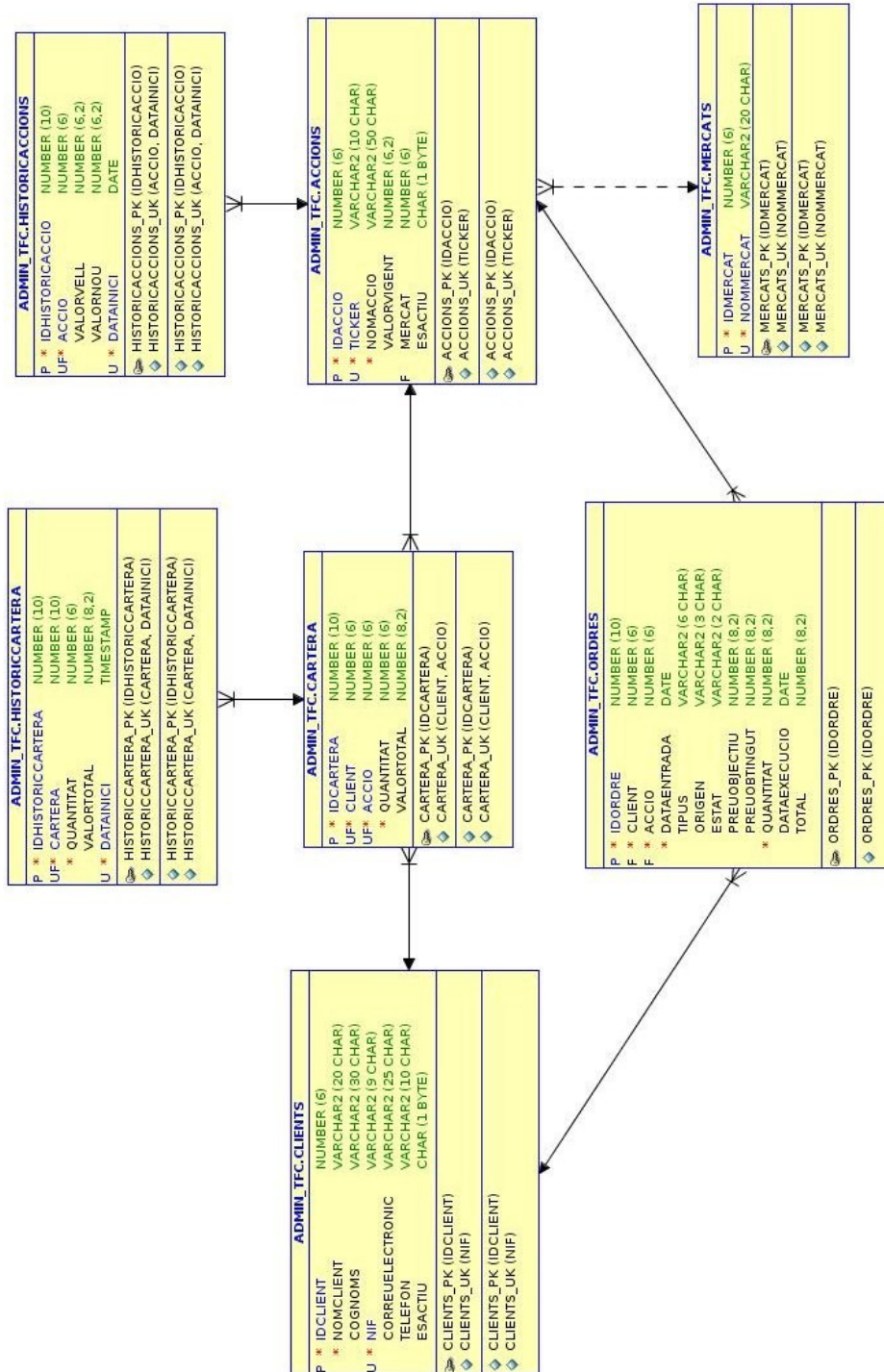


Figura 20: Diagrama del model lògic sobre Oracle del mòdul operacional

4.5 – Model ER del mòdul estadístic

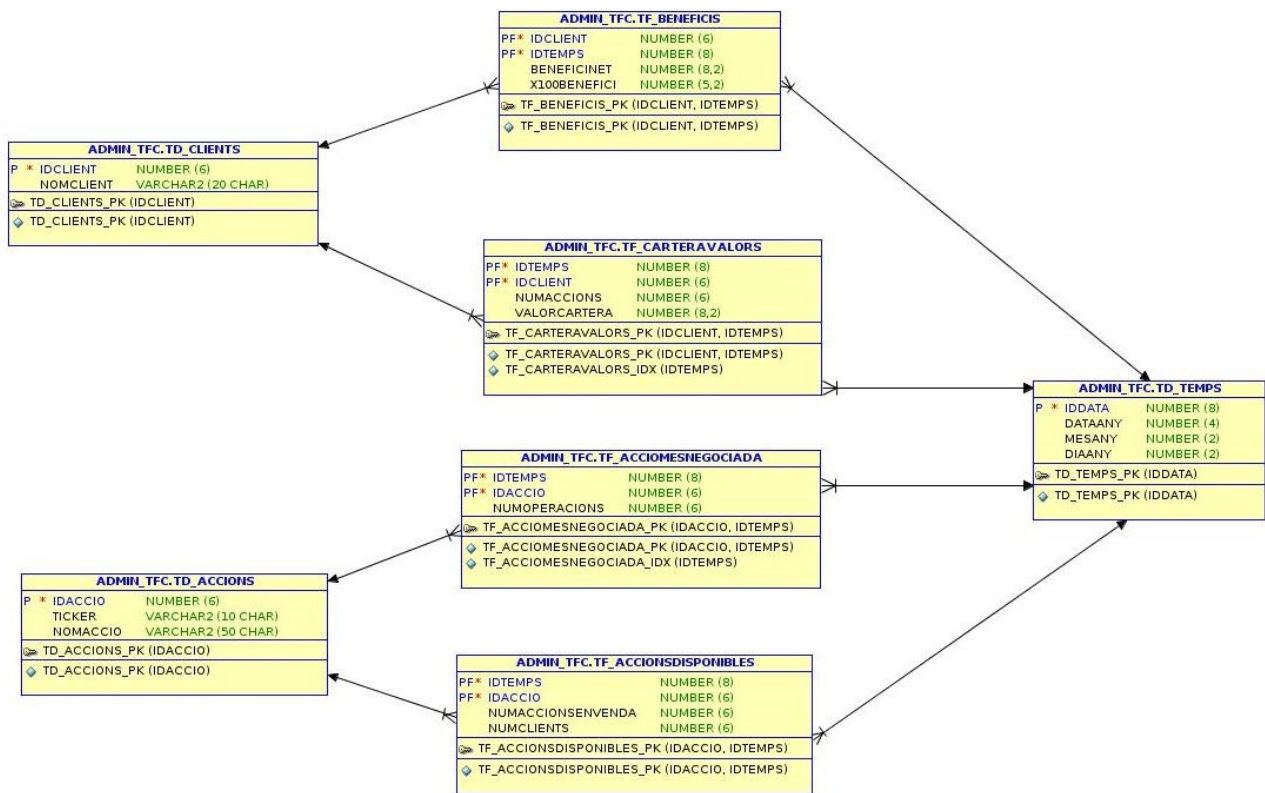


Figura 21: Diagrama del model lògic sobre Oracle del mòdul estadístic

5 – Disseny físic

En aquesta etapa, es concreta el sistema de gestió de bases de dades (SGBD) que s'utilitzarà per implementar la base de dades, en aquest cas ORACLE. La implementació dependrà de les característiques concretes del SGBD escollit.

Es defineixen les taules amb les claus primàries, foranes i alternatives corresponents, que es relacionen amb els fitxers físics a través de l'espai virtual. També es creen els índex necessaris, que es relacionaran amb el seu propi espai virtual d'índexs, que alhora estarà associat a un fitxer físic. Aquesta implementació ens proporcionarà flexibilitat en la relació lògica – física.

5.1 – Creació de la base de dades

S'instal·larà la versió 11g R2 Express Edition (XE) d'Oracle sobre una màquina virtual Oracle VM VirtualBox amb el SO Ubuntu 12.04 LTS.

Per a gestionar la BD, s'ha instal·lat el Oracle SQL Developer.

S'utilitzarà la base de dades que s'instal·la per defecte, anomenada XE.

5.2 – Creació dels espais virtuals

Les dades s'han de guardar en un suport no volàtil, que permeti accedir-hi quan sigui necessari: els fitxers. Però no hi ha una relació unívoca i directe entre la taula i el fitxer, ja que la relació entre les taules i els fitxers pot ser de molts a molts.

Per simplificar aquest escenari, s'introdueix l'espai virtual, que connectarà els nivells lògic i físic, que proporcionarà al SGBD una visió simplificada del nivell físic. Els espais virtuals a Oracle s'anomenen Tablespace, i cada Tablespace estarà connectat a un o més datafiles.

Per aquest projecte ens interessarà crear els següents espais virtuals:

- Espai virtual per les dades operacionals
- Espai virtual per a les dades del Data Warehouse
- Espai d'índex
- Espai pels temporals

Cada un d'aquests espais virtuals s'hauria de configurar de forma optimitzada per tractar amb els tipus de dades que s'hi relacionen i la forma com s'hi executen les transaccions. Definir la capacitat dels datafiles, la forma en què es van expandint a mesura que es van emplenat de dades, la forma com es va reservant espai de disc i la dimensió del bloc mínim de lectura.

En un primer moment es configuren els tablespaces amb uns valors normals i vàlids per a qualsevol implementació. Però queda pendent la tasca d'afinar la seva configuració i així poder augmentar el rendiment de la base de dades, tenint en compte que la part operacional és del tipus transaccional, i la part estadística és del tipus analítica.

Per exemple, per la creació del tablespace destinat a les taules del mòdul operacional s'ha fet el següent:

```
CREATE TABLESPACE TFC_BD
DATAFILE '/home/vmpep/Documents/tfc/bd_tfc/TFC_DADES_01.dbf' SIZE 50M
AUTOEXTEND ON NEXT 500K MAXSIZE 200M
```

```
BLOCKSIZE 8K  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE  
SEGMENT SPACE MANAGEMENT AUTO;
```

Hi definim el nom del fitxer que contindrà les dades, TFC_DADES_01.DBF, amb una capacitat inicial de 50Mb.

Indiquem que vagi fent extensions del fitxer, augmentant la capacitat, en salts de 500Kb. Però li posem un límit de 200Mb de capacitat màxima. Això últim ens ajudarà a detectar si tenim un problema de funcionament en la BD, que per exemple faci augmentar molt ràpidament mida de les taules, abans no esgotem l'espai físic real d'emmagatzemament.

També li indiquem que la mida del block sigui de 8Kb, que és la mida per defecte. Més endavant serà recomanable ajustar aquesta dimensió, ja que es recomana que les BD transaccionals, amb moltes lectures/escriptures es configuren amb un blocksize petit (4Kb per exemple), ja que això evita que els blocks quedin fragmentats molt ràpidament. En canvi, en una BD del tipus data warehouse, en què sobretot els fan lectures massives de dades, és més interessant tenir un blocksize gran (32 Kb).

També definim que el sistema gestioni automàticament les extensions de la capa virtual. El sistema determinarà la dimensió òptima de l'extensió amb un mínim de 64Kb, per a cada nou extent.

Com a últim pas, es configura la gestió de la memòria dels segments, que s'ocuparà de com es gestionaran els espais lliures dins de cada bloc de memòria. És habitual definir-ho automàtic, que fa la gestió de l'espai dels blocs es faci mitjançant bitmaps, i és més eficient que no el mètode manual, que es basa en free-lists.

El fitxer 01-CreacioTablespace.sql conté el codi de creació del Tablespace.

5.3 – Creació dels usuaris

Es crearan 2 usuaris que representen els perfils d'actor que més interactuaran amb la BD. Hi haurà un perfil d'usuari amb privilegis sobre objectes, que inclouen la consulta, modificació i inserció de dades, i un perfil d'administrador, amb privilegis de sistema, amb capacitat també, de crear, esborrar i modificar qualsevol dels objectes de la BD (com taules, vistes, procediments, seqüències...).

Aquest privilegis s'assignaran als usuaris com a privilegis de rol. Així facilitarem el manteniment per propagar amb facilitat els privilegis a nous usuaris que es puguin donar d'alta en el futur.

Preferim crear els nostres propis rols, més que no pas utilitzar els rols que hi ha predefinits a Oracle. Això permet un major control dels privilegis assignats, i per tant una major seguretat. Aleshores creem els rols: bàsic, usuari, i administrador, i els associem als usuaris adm_tfc, user_tfc.

Aquests usuaris es defineixen amb el tablespace de dades per defecte de TFC_BD, i se'ls assigna el tablespace temporal TFC_temp.

Concretament els privilegis de sistema que es donen al rol d'administrador són els de crear i esborrar taules, vistes, procediments, seqüències, disparadors, tipus i sinònims.

Els privilegis d'objecte assignats rol d'usuari són els de seleccionar en les taules propietat de l'usuari admin_tfc: clients, accions, mercats, cartera, ordres, errors_reconeguts, TF_beneficis, TF_CarteraValors, TF_AccioMesNegociada, TF_AccionsDisponibles.

També tindrà permisos per poder executar els procediments i funcions definits en els paquets, propietat de admin_tfc: pkg_accions, pkg_clients, pkg_ordre, pkg_consultes, pkg_consultes_DWH i pkg_control. Els permisos de select sobre les taules assignats al rol d'usuari no són necessaris, ja que la interacció d'aquest perfil d'usuari amb la BD ha de ser sempre a través dels procediments emmagatzemats definits.

El fitxer 02_CreacioUsuaris.sql conté el codi de creació dels usuaris. El fitxer 06_permisos_user.sql conté el codi d'assignació dels permisos a objectes.

5.4 - Creació de les taules

El següent pas és crear les taules que contindran les dades. En aquest pas es creen les taules del mòdul operatiu, del mòdul estadístic (DWH) i les del mòdul de control.

Les taules del mòdul operatiu i les del mòdul de control creen en el tablespace TFC_BD, i les taules del mòdul DWH es creen en el tablespace TFC_DWH.

En aquest moment no es té en consideració els paràmetres d'emplenat de les pàgines, que controlaran el % d'espai lliure que s'ha de destinar en cada pàgina per permetre operacions d'alta, modificació i baixa, i evitar haver de crear enllaços a altre pàgines. Per tant, la configuració dels paràmetres PCTFREE i PCTUSED per a cada taula es deixarà com a tasca pendent de refinat.

De forma orientativa, el valor de PCTFREE per una taula transaccional ha de ser de 50%, mentre que per una taula estadística, en la que que ni es modifiquen ni s'esborren els registres, ha de ser de l'ordre del 0%.

El fitxer 03_CreacioTaules.sql conté el codi de creació de les taules.

5.5 - Creació de les seqüències

Es crea un identificador numèric seqüencial per a cada taula del mòdul d'explotació, que identificarà cada registre, i permetrà simplificar la selecció dels registres.

El fitxer 04_CreacioSequencies.sql conté el codi de creació de les seqüències.

5.6 - Creació dels índex

A mesura que es van definint les claus primàries i les restriccions d'unicitat, oracle crea automàticament els índex associats a aquelles columnes, per tal de millorar el rendiment quan s'accedeix a aquelles dades i evitar la presència de valors nuls.

En la fase de proves s'ha vist necessari crear un índex per la taula TF_CARTERAVALORS i un per la taula TF_ACCIOMESNEGOCIADA, ja que els accessos no es realitzaven amb el rendiment sol·licitat en els requeriments. En l'apartat 8 de la memòria es descriu amb més detall.

Es deixa oberta la possibilitat de, més endavant, crear més índexs per millorar el rendiment d'altres consultes.

Els índex es creen en un Tablespace propi, el TFC_IDX, ja que d'aquesta forma es pot configurar i adaptar a les seves característiques per millorar-ne l'eficiència. Aquest punt, també es deixa per les tasques pendents.

El fitxer 045_index.sql conté el codi de creació dels índexs.

6 - Implementació

El codi que aporta la funcionalitat al sistema s'ha repartit en paquets, i s'ha procurat que tot el codi que s'inclou en cada paquet faci referència a una sola taula. D'aquesta forma ordenada és fàcil accedir a qualsevol procediment o funció.

S'han creat els següents paquets :

pkg_control.sql

pkg_dwh.sql

pkg_cartera.sql

pkg_ordres.sql

pkg_accions.sql

pkg_clients.sql

pkg_consultes.sql

pkg_consultes_DWH.sql

pkg_utils.sql

A continuació es descriuran els diferents paquets i els procediments i funcions que contenen. Cal tenir en compte que tots els procediments implementen la funcionalitat que a l'executar-se emmagatzemen la seva crida en una taula Log. Per això ja s'obviarà aquest detall en la descripció de cadascun d'ells.

També és important comentar que hi ha procediments que estan pensats per a ser executats per un actor, extern a la BD, però altres procediments estan pensats per ser executats pel mateix sistema. Tots dos grups estan implementats de forma que es poden executar externament, però en un d'aquests no serà la forma habitual.

Com a norma general, els procediments dels paquets pkg_accions, pkg_clients, pkg_ordres, pkg_consultes i pkg_consultes_DWH estan pensats per ser executats per un actor extern. En canvi els altres estan més destinats a ser executats per crides internes del mateix sistema.

En la implementació dels procediments s'ha fet servir en diverses ocasions la propietat de sobrecàrrega de procediments, que consisteix en tenir més d'un procediment amb el mateix nom, però al que se li han de passar paràmetres diferents. Principalment això ens servirà per executar un procediment passant-li diferents identificadors clau.

6.1 – Package clients

Aquest paquet inclou els procediments que són necessaris per actuar amb la taula Clients.

Conté els procediments d'alta, modificació i baixa

6.1.1 - prc_client_alta

Procediment que permet donar d'alta un nou client en el sistema

Els paràmetres que se li han de passar són:

- pNomclient VARCHAR2 : Nom del client. Not null.

- pCognoms VARCHAR2 : Cognoms del client
- pNif VARCHAR2 : Nif del client. Not null.
- pCorreuelectronic VARCHAR2 : Correu electrònic del client
- pTelefon VARCHAR2 : Telèfon del client
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error registre duplicat: Error retornat quan s'ha intentat duplicar un camp clau
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.1.2 - prc_client_modificacio

Procediment per modificar les dades principals d'un client. Es permet modificar el nom, cognoms, nif, correu electrònic i telèfon. En el cas de no passar algun dels paràmetres, es mantindrà els valors vigents dins la taula.

Els paràmetres que se li han de passar són:

- pIdclient NUMBER : Identificador del client. Not null.
- Nomclient VARCHAR2 : Nou nom del client
- pCognoms VARCHAR2 : Nous cognoms del client
- pNif VARCHAR2 : Nou nif del client
- pCorreuelectronic VARCHAR2 : Nou correu electrònic del client
- pTelefon VARCHAR2 : Nou telèfon del client
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan l'identificador passat no es correspon a cap identificador vàlid de client.
- Error registre duplicat: Error retornat quan s'intenta canviar el NIF per un que ja existeix en la taula.
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.1.3 - prc_client_baixa

Procediment per donar la baixa lògica a un client, identificat amb el seu idClient. Aquest procediment verifica si el client té ordres de compra/venda actives (pendants), i si és el cas, les cancel·la.

La cancel·lació de ordres de venda afecta a la taula del DWH TF_accionsdisponibles.

Els paràmetres que se li han de passar són:

- pIdclient NUMBER, : Identificador de client. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap client associat amb l'identificador passat per paràmetre.
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.1.4 - prc_client_baixa

Procediment per donar de baixa lògica un client identificat amb el seu NIF. És una sobrecàrrega del procediment anterior, i té les mateixes funcionalitats.

Els paràmetres que se li han de passar són:

- pNif VARCHAR2, : NIF del client. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap client associat amb l'identificador passat per paràmetre.
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.2 – Package accions

Aquest paquet inclou els procediments necessaris per actuar amb la taula Accions. Conté els procediments d'alta, modificació, baixa i canvi de preu de les accions.

6.2.1 - prc_accio_alta

Procediment per donar d'alta una nova acció en el sistema.

Els paràmetres que se li han de passar són:

- pTicker VARCHAR2 : És un codi identificador únic assignat a una acció. Not null.
- pNomAccio VARCHAR2 : Nom de l'acció. Not null.
- pMercat NUMBER : Mercat on es negocia l'acció
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error registre duplicat: Error retornat quan s'ha intentat duplicar un camp clau
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.2.2 - prc_accio_modificacio

Procediment per modificar les dades principals d'una acció. Es permet modificar el Ticker, el nom i el mercat. En el cas de no passar algun dels paràmetres, es mantindrà els valors vigents dins la taula.

Els paràmetres que se li han de passar són:

- pIdAccio NUMBER : Identificador de l'acció. Not null.
- pTicker VARCHAR2 : Nou Ticker de l'acció
- pNomAccio VARCHAR2 : Nou nom de l'acció

- pMercat NUMBER : Nou mercat on es negocia l'acció
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan l'identificador passat no es correspon a cap identificador vàlid d'acció.
- Error registre duplicat: Error retornat quan s'intenta canviar el Ticker per un que ja existeix en la taula.
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.2.3 - prc_accio_baixa

Procediment per donar la baixa lògica a una acció, identificada amb el seu id. Aquest procediment verifica si l'acció està associada a alguna ordre de compra/venda activa (pendent), i si és el cas, les cancel·la.

La cancel·lació de ordres de venda afecta a la taula del DWH TF_accionsdisponibles.

Els paràmetres que se li han de passar són:

- pIdAccio NUMBER : identificador de l'acció. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap acció associada a l'identificador passat per paràmetre.
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.2.4 - prc_accio_baixa

Procediment per donar la baixa lògica a una acció identificada amb el seu Ticker. És una sobrecàrrega del procediment anterior i té les mateixes funcionalitats.

Els paràmetres que se li han de passar són:

- pTicker VARCHAR2 : Ticker de l'acció. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap acció associada a l'identificador passat per paràmetre.
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.2.5 - prc_accio_preu

Procediment per modificar el preu d'una acció, identificada amb el seu id. Només es permet actualitzar el preu de les accions actives.

L'actualització del preu d'una acció afecta a moltes taules. En la part de la BD operacional, afecta a la taula cartera, ja que modifica el valor de la cartera dels clients que tenen l'acció modificada. Afecta a la taula ordres, ja que s'analitza quines ordres es poden executar automàticament per què el

seu preu objectiu encaixa amb el nou preu de l'acció. En la part del DWH, afecta a la taula TF_Beneficis i a la taula TF_CarteraValors.

Els paràmetres que se li han de passar són:

- pIdAccio NUMBER : Identificador de l'acció. Not null.
- pPreu NUMBER : Nou preu de l'acció. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap acció associada a l'identificador passat per paràmetre que compleixi les condicions.
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.2.6 - prc_accio_preu

Procediment per modificar el preu d'una acció. És una sobrecàrrega del mètode anterior i té les mateixes funcionalitats.

Els paràmetres que se li han de passar són:

- pTicker VARCHAR2 : Ticker de l'acció. Not null.
- pPreu NUMBER : Nou preu de l'acció. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap acció associada a l'identificador passat per paràmetre que compleixi les condicions.
- Error no controlat: Error retornat quan s'han passat algun paràmetre incorrecte.

6.3 – Package cartera

Aquest paquet inclou els procediments necessaris per actuar amb la taula Cartera. Conté els procediments d'adjudicació i de retirada d'accions, i el d'actualització del valor de la cartera. Els procediments d'aquest paquet estan pensats per ser cridats pel mateix sistema, des d'altres procediments, més que no per un actor extern. Concretament s'ha plantejat que sigui el procediment d'execució d'ordres el que faci la crida a aquests procediments.

6.3.1 - prc_cartera_adjudicar

Procediment per adjudicar accions a un client. Crearà una cartera nova per un determinat client i acció, o si es dona el cas que aquell client ja té cartera per l'acció concreta, incrementarà el nombre d'accions amb la quantitat passada.

Els paràmetres que se li han de passar són:

- pIdClient NUMBER : Identificador del client al que se li vol adjudicar noves accions. Not null.
- pIdAccio NUMBER : identificador de l'acció que es vol adjudicar. Not null.
- pQuantitat NUMBER : Quantitat d'accions que es volen adjudicar. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.3.2 - prc_cartera_adjudicar

Procediment per adjudicar accions a un client. És una sobrecàrrega del procediment anterior i té la mateixa funcionalitat. En aquest cas també es passa el valor vigent de l'acció, per facilitar el càlcul de valor de la cartera.

Els paràmetres que se li han de passar són:

- pIdClient NUMBER : Identificador del client al que se li vol adjudicar noves accions. Not null
- pIdAccio NUMBER : identificador de l'acció que es vol adjudicar. Not null.
- pQuantitat NUMBER : Quantitat d'accions que es volen adjudicar. Not null.
- pValorVigent NUMBER : Valor vigent de l'acció que es vol adjudicar. Not null
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.3.3 - prc_cartera_retirar

Procediment per retirar accions a un client. No es permet retirar més accions que les que disposa el client, ja que una cartera no pot quedar amb quantitat d'accions negativa.

Els paràmetres que se li han de passar són:

- pIdClient NUMBER : Identificador del client al que se li vol retirar unes accions. Not null
- pIdAccio NUMBER : identificador de l'acció que es vol retirar. Not null.
- pQuantitat NUMBER : Quantitat d'accions que es volen retirar. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error poques accions disponibles: Error retornat quan es volen retirar més accions de les que disposa el client.
- Error no s'ha trobat el registre: Error retornat quan no hi ha cap client o cap acció que tingui associats els identificadors passats.
- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap acció associada a l'identificador de client o acció.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.3.4 - prc_cartera_retirar

Procediment per retirar accions a un client. És una sobrecàrrega del procediment anterior amb les mateixes funcionalitats. En aquest cas també es passa el valor vigent de l'acció, per facilitar el càlcul de valor de la cartera.

Els paràmetres que se li han de passar són:

- pIdClient NUMBER : Identificador del client al que se li vol retirar noves accions. Not null.
- pIdAccio NUMBER : identificador de l'acció que es vol retirar. Not null.
- pQuantitat NUMBER : Quantitat d'accions que es volen retirar. Not null.

- pValorVigent NUMBER : Valor vigent de l'acció que es vol retirar. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error poques accions disponibles: Error retornat quan es volen retirar més accions de les que disposa el client.
- Error no s'ha trobat el registre: Error retornat quan no hi ha cap client o cap acció que tingui associats els identificadors passats.
- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap cartera associada a l'identificador de client o acció.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.3.5 - prc_cartera_actualitzar

Procediment per actualitzar la cartera d'accions d'un client, d'acord a una variació del preu de l'acció. Actualitzarà el valor de les carteres associades a l'acció passada, al preu vigent de l'acció, i considerant només les carteres actives, que són les que estan associades a clients i accions actius.

Els paràmetres que se li han de passar són:

- pIdAccio NUMBER : Identificador de l'acció de la que es vol actualitzar les carteres a les que està associada. Not null.
- pPreuVigent NUMBER : Preu de l'acció que es passa. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no hi ha cap client o cap acció que tingui associats els identificadors passats.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.3.6 - prc_cartera_actualitzar

Procediment per actualitzar totes les carteres d'accions de tots els clients. Actualitza tots els registres de la taula cartera d'acord al valor vigent de les accions. És una sobrecàrrega del procediment anterior. Només es consideren les carteres associades a clients actius i accions actives.

Els paràmetres que se li han de passar són:

- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no hi ha cap client o cap acció que tingui associats els identificadors passats.
- Error no s'ha actualitzat cap registre: Error retornat quan no hi ha cap cartera associada a l'identificador de les accions.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.4 – Package ordres

Aquest paquet inclou els procediments necessaris per crear, modificar, cancel·lar i executar ordres de compra/venda. Cal tenir present que aquest paquet representa el cor del negoci de l'empresa, ja que és a través de la gestió de les ordres de compra/venda que es va definint la cartera d'un client. És a

dir, per què un client acabi essent propietari d'unes accions, s'ha de fer els passos previs de crear una ordre de compra, després executar-la, i això aportarà al client les accions comprades.

6.4.1 - prc_ordre_alta

Procediment per donar d'alta una nova ordre. No es permet crear ordres amb accions no vàlides (no actives). Si l'ordre és de venda es verifica que no es posen a la venda més accions de que té disponibles aquell client. No es permet tenir més d'una ordre de compra activa per client i acció.

L'alta d'una nova ordre del tipus Venda, afecta a la taula TF_Accions disponibles.

Els paràmetres que se li han de passar són:

- pIdClient NUMBER : Identificador del client que crea l'ordre. Not null.
- pIdAccio NUMBER : Identificador de l'acció associada a l'ordre. Not null.
- pQuantitat NUMBER : Quantitat d'accions que es volen comprar/vendre. Not null.
- pTipus VARCHAR2 : Classificació de l'ordre. Pot ser COMPRA o VENDA
- pOrigen VARCHAR2: Definició de l'origen de l'ordre. Pot ser MAN o AUT.
- pPreuObjectiu NUMBER: Preu que es considera correcte per executar automàticament l'ordre. Default 0.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no es poden vendre accions inactives: Error retornat quan s'ha intentat generar una ordre amb una acció inactiva.
- Error poques accions disponibles: Error retornat quan s'intenta crear una ordre de venda en la que la quantitat d'accions que es posen a la venda supera a la quantitat d'accions disponibles del client.
- Error alta ordre ja existent: Error retornat quan s'intenta donar d'alta una ordre per un client i acció, i ja en té una d'activa.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.4.2 - prc_ordre_modificacio

Procediment per modificar les dades principals d'una ordre. Es permet modificar la quantitat d'accions que es volen comprar/vendre i el preu objectiu de l'ordre. Només es poden modificar ordres que estiguin en estat PD. Si l'ordre és del tipus Venda es verifica que no es posen a la venda més accions de que té disponibles aquell client.

Si l'ordre és del tipus Venda, un canvi de la quantitat afecta a la taula TF_Accions disponibles.

Els paràmetres que se li han de passar són:

- pIdOrdre NUMBER : Identificador de l'ordre que es vol modificar. Not null.
- pQuantitat NUMBER : Nova quantitat d'accions associades a l'ordre.
- pPreuObjectiu number : Nou preu objectiu.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error poques accions disponibles: Error retornat quan s'intenta crear una ordre de venda en la que la quantitat d'accions que es posen a la venda supera a la quantitat d'accions disponibles del client.

- Error no s'ha actualitzat cap registre: Error retornat quan s'intenta modificar una ordre que no està en situació pendent 'PD'.
- Error no s'ha trobat el registre: Error retornat quan no hi ha cap ordre associada al identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.4.3 - prc_ordre_cancelacio

Procediment per cancel·lar una ordre que està en estat PD. Només es poden cancel·lar les ordres que estiguin en estat PD.

Si l'ordre és del tipus Venda, una cancel·lació afecta a la taula TF_Accions disponibles.

Els paràmetres que se li han de passar són:

- pIdOrdre NUMBER : Identificador de l'ordre que es vol cancel·lar. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan s'intenta modificar una ordre que no està en situació pendent 'PD'.
- Error no s'ha trobat el registre: Error retornat quan no hi ha cap ordre associada al identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.4.4 - prc_ordre_execucio

Procediment que permet executar una ordre en curs que està en estat PD. Es realitza l'operació de compra o de venda independentment del preu objectiu que hi hagi marcat. La compra o la venda es realitza al preu vigent de l'acció.

A l'executar una ordre, s'adjudica o es retira la quantitat d'accions de l'ordre al client. Per tant, s'actualitza la cartera del client.

L'execució d'una ordre de Venda afecta a la TF_Accions disponibles. La execució d'una ordre, de qualsevol tipus, afecta a les taules TF_AccioMesNegociada, TF_CarteraValors i TF_Beneficis.

Els paràmetres que se li han de passar són:

- pIdOrdre NUMBER : Identificador de l'ordre que es vol executar. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no s'ha trobat cap ordre que compleixi la condició de preu objectiu.
- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap ordre en situació PD, lligada a clients i accions actives.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.4.5 - prc_ordre_execucio_auto

Procediment que permet executar automàticament les ordres en curs, que estan en estat PD, basant-se en el preu objectiu i en el valor vigent de les accions a les que estan associades. No se li passa cap paràmetre, i el procediment fa un recorregut per a totes les ordres i executa les que compleixen la condició de preu objectiu.

A l'executar una ordre, s'adjudica o es retira la quantitat d'accions de l'ordre al client. Per tant, s'actualitza la cartera del client.

L'execució d'una ordre de VENDA afecta a la TF_Accions disponibles. La execució d'una ordre, de qualsevol tipus, afecta a les taules TF_AccioMesNegociada, TF_CarteraValors i TF_Beneficis.

Els paràmetres que se li han de passar són:

- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no s'ha trobat cap ordre que compleixi la condició de preu objectiu.
- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap ordre en situació PD, lligada a clients i accions actives.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.4.6 - prc_ordre_execucio_auto

Procediment que permet executar automàticament les ordres en curs, que estan associades a l'acció que es passa per paràmetre, i compleixen el criteri del preu objectiu. Aquest procediment és una sobrecàrrega del procediment anterior, i té les mateixes funcionalitats. En aquest cas, però, se li passa el identificador de l'acció i el seu preu vigent per agilitzar l'execució de l'ordre.

Els paràmetres que se li han de passar són:

- pIdAccio NUMBER . Identificador de l'acció de la que es vol revisar les ordres que la tenen vinculada. Not null.
- pValorVigent NUMBER : Valor de l'acció. Not null.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha actualitzat cap registre: Error retornat quan no s'ha trobat cap ordre que compleixi la condició de preu objectiu.
- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap ordre en situació PD, lligada a clients i accions actives.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.5 – Package DWH

Aquest paquet inclou els procediments necessaris per realitzar el manteniment de les taules del mòdul estadístic (o Data Warehouse), tant les taules de dimensions com les taules de fets. Tots aquests procediments estan pensats per a ser cridats per altres procediments, i l'actor exterior no els ha d'utilitzar si no és per administració. A cada procediment qui fa la crida i quan la fa.

6.5.1 - fun_insDim_tdTemps

Funció que insereix un nou registre a la taula de dimensions TD_temps, i el retorna per poder ser utilitzat pels altres procediments. Converteix la data que es passa per paràmetre, en un identificador de la data en el format numèric YYYYMMDD.

Prèviament a cada inserció en una taula de fets, es genera un identificador pel dia en que es realitza la transacció.

Els paràmetres que se li han de passar són:

- pData DATE : data que es vol inserir en la taula de dimensió. Not null.
- RETURN NUMBER : retorna el identificador de la data inserida.

Informació de retorn:

- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.5.2 - prc_insDim_tdAccio

Procediment que insereix un nou registre a la taula de dimensions TD_Accio, corresponent a l'identificador de l'acció passat. Recupera els valors idAccio, ticker i el nomAccio de la taula Accions, i els insereix en la taula de dimensions.

Els paràmetres que se li han de passar són:

- pIdAccio NUMBER : Identificador de l'acció que es vol inserir en la taula de dimensió. Not null.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap acció que tingui associada el identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.5.3 - prc_insDim_tdClient

Procediment que insereix un nou registre a la taula de dimensions TD_Client. Corresponent a l'identificador de client passat. Recupera els valors idClient, nomclient de la taula Clients, i els insereix en la taula de dimensions.

Els paràmetres que se li han de passar són:

- pIdClient NUMBER : Identificador del client que es vol inserir en la taula de dimensió.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap client que tingui associat el identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.5.4 – prc_ins_tfAccionsDisponibles

Procediment que insereix un nou registre a la taula de fets TF_AccionsDisponibles. El procediment registra la quantitat d'accions que estan associades a ordres de venda, i els clients que les tenen en la cartera.

No cal passar-li cap paràmetre.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap acció que tingui associat el identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

Aquest procediment es crida quan:

- Es crea una nova ordre, si és de venda
- S'executa una ordre, si és de venda
- Es cancel·la una ordre, si és de venda

6.5.5 – prc_ins_tfAccMesNegociada

Procediment que insereix un nou registre a la taula de fets TF_AccioMesNegociada. El procediment registra l'acció o accions que han participat en més operacions de compra/venda.

No cal passar-li cap paràmetre.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap acció que tingui associat el identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

Aquest procediment es crida quan:

- S'executa una ordre, ja sigui de compra o de venda.

6.5.6 – prc_ins_tfCarteraValors

Procediment que insereix un nou registre a la taula de fets TF_CarteraValors. El procediment registra el client amb la cartera d'accions amb un valor més elevat

No cal passar-li cap paràmetre.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap client que tingui associat el identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

Aquest procediment es crida quan:

- S'executa una nova ordre (ja que és una operació que actualitza la cartera)
- Es canvia el preu de les accions (ja que també actualitza la cartera)

6.5.7 – prc_ins_tfBeneficis

Procediment que insereix un nou registre a la taula de fets TF_Beneficis. El procediment registra el diferencial del valor de la cartera entre el principi de mes i la data en la que s'executa.

No cal passar-li cap paràmetre.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap client que tingui associat l'identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

Aquest procediment es crida quan:

- S'executa una nova ordre (ja que és una operació que actualitza la cartera)
- Es canvia el preu de les accions (ja que també actualitza la cartera)

6.6 – Package consultes

Aquest paquet inclou els procediments necessaris per realitzar les consultes al mòdul d'explotació definides en els requisits.

6.6.1 - prc_cns_operacionsClient

Procediment que executa la consulta que retorna les dades dels clients que han realitzat operacions durant l'any. Es considera operació a qualsevol ordre creada, ja sigui de compra o de venda, i en qualsevol estat.

Els paràmetres que se li han de passar són:

- pAny VARCHAR2 : Any sobre el que es vol realitzar la consulta
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap client que hagi realitzat operacions l'any indicat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.6.2 - prc_cns_operacionsAccio

Procediment que executa la consulta que retorna les dades de les accions que han participat en operacions de compra/venda que s'han executat durant l'any.

Els paràmetres que se li han de passar són:

- pAny VARCHAR2 : Any sobre el que es vol realitzar la consulta
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap accio que s'hagi executat en l'any indicat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.6.3 - prc_cns_10accionsMesAdquirides

Procediment que executa la consulta que retorna un llistat amb les 10 accions més adquirides pels sistemes automàtics. Aquestes accions provenen de les ordres que tenen l'especificació origen de tipus Automàtic.

Els paràmetres que se li han de passar són:

- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap accio.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.6.4 - prc_cns_carteraVigent

Procediment que executa la consulta que retorna un llistat amb la composició de la cartera de valors d'un client en una data concreta.

Els paràmetres que se li han de passar són:

- pIdClient NUMBER : Identificador del client sobre el que es vol realitzar la consulta.
- pData DATE : Data en la que interessa realitzar la consulta.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap històric de la cartera vàlid.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.7 – Package consultes DWH

Aquest paquet inclou els procediments necessaris per realitzar les consultes al mòdul estadístic que s'han definit en els requisits. Són consultes que han de tenir un cost constant, ja que la resposta ha de ser ràpida i independent del nombre de files que puguin contenir les taules. Per aconseguir aquesta resposta es faran consultes referenciades a un camp clau.

6.7.1 - prc_cns_beneficiMesClient

Procediment que executa la consulta que retorna el benefici net, en €, i el % de la cartera d'accions del mes anterior al passat per paràmetre.

Els paràmetres que se li han de passar són:

- pIdclient NUMBER : Identificador del client sobre el que es vol realitzar la consulta.
- pMes DATE : Mes posterior a que es vol consultar.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap client que tingui associat l'identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.7.2 - prc_cns_accDisponibles

Procediment que executa la consulta que retorna el nombre d'accions disponibles (associades a ordres de venda No executades) en el broker en aquell moment i el nombre de clients que les tenen en cartera.

Els paràmetres que se li han de passar són:

- pIdAccio NUMBER : Identificador de l'acció sobre la que es vol realitzar la consulta.
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap acció que tingui associada l'identificador passat.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.7.3 - prc_cns_clientCarteraMax

Procediment que executa la consulta que retorna el client amb la cartera d'accions més gran, i el valor en euros de la cartera.

No cal passar-li cap paràmetre.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap client.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.7.4 - prc_cns_accioMesNegociada

Procediment que executa la consulta que retorna l'acció més negociada (comprada + venuda) en el broker de forma històrica.

No cal passar-li cap paràmetre.

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap acció.
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.8 – Package control

Aquest paquet inclou els procediments de suport que podran ser utilitzats des d'altres paquets, Inclou el procediment per registrar l'execució dels procediments a la taula Log, i la funció que retorna els errors personalitzats.

6.8.1 - prc_logProcediment

Procediment que insereix la informació del procediment que s'està executant dins la taula Log.

Els paràmetres que se li han de passar són:

- pNomprocediment VARCHAR2 : Nom del procediment que s'executa
- pParametresin VARCHAR2 : Cadena amb els paràmetres d'entrada del procediment que s'executa.
- pFinalitzacioTxt VARCHAR2 : Cadena que informa de com finalitza l'execució. Pot ser Ok, o informar de l'error produït.

Informació de retorn:

- Error no s'ha registrat l'error correctament: Error retornat quan el procediment ha fallat, i no ha insertat cap registre a la taula Log.

6.8.2 - fun_Error

Funció que consulta a la taula d'errors definits i retorna al missatge d'error establert.

Els paràmetres que se li han de passar són:

- pIdError NUMBER : Identificador de l'error que es vol tractar

Informació de retorn:

- Error no s'ha trobat el registre: Error retornat quan no s'ha trobat cap error associat a l'identificador passat..
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.8.3 - prc_mercat_alta

Procediment que serveix per donar d'alta els mercats. S'ha plantejat com una utilitat ja que els procediments ABM dels la taula Mercats quedaven fora de l'abast del projecte.

Els paràmetres que se li han de passar són:

- pNomMercat VARCHAR2: Nom del nou mercat. Not null
- rsp OUT VARCHAR2 : Variable que conté el missatge de sortida.

Informació de retorn:

- Error registre duplicat: Error retornat quan s'ha intentat duplicar un camp clau
- Error null no admès: Error retornat quan s'intenta passar un valor null
- Error no controlat: Error retornat quan s'ha passat algun paràmetre incorrecte.

6.9 – Triggers o disparadors

Aquest paquet inclou la definició dels disparadors que s'han considerat necessaris pel manteniment automàtic de les taules. S'ha considerat que les taules que s'han de mantenir amb disparadors són les taules d'auditoria, que en aquest projecte hi ha definides la taula HistoricAccions i la taula HistoricCartera. Però també s'ha cregut interessant fer el manteniment mitjançant disparadors, de les taules de dimensió del mòdul estadístic TD_clients i TD_accions, ja que d'aquesta forma contindran sempre tots els clients i accions registrats en la BD operacional.

6.9.1 - trigHistoricAccions

Aquest trigger es dispara en cada actualització de la columna valor vigent de la taula Accions, i afegeix un registre en la taula històric accions, que manté l'històric dels seus preus.

Informació de retorn:

- Error no controlat: Error retornat quan no s'ha pogut fer correctament la inserció en la taula historicAccions.

6.9.2 - trigHistoricCartera

Aquest trigger es dispara en cada actualització de les columnes quantitat i valor total, de la taula Cartera, i afegeix un registre en la taula històric Cartera, que manté l'històric.

Informació de retorn:

- Error no controlat: Error retornat quan no s'ha pogut fer correctament la inserció en la taula historicCartera.

6.9.3 - trigAltaClients

Aquest trigger es dispara en cada nova inserció en la taula Clients, i fa el manteniment de la taula TF_Clients

Informació de retorn:

- Error no controlat: Error retornat quan no s'ha pogut fer correctament la inserció en la taula TF_Clients.

6.9.4 - trigAltaAccions

Aquest trigger es dispara en cada nova inserció en la taula Accions, i fa el manteniment de la taula TF_Accions.

Informació de retorn:

- Error no controlat: Error retornat quan no s'ha pogut fer correctament la inserció en la taula TF_Accions.

7 - Fase de proves

S'ha planificat la fase de proves per poder provar el màxim nombre de situacions d'error que es poden donar a l'executar els procediments definits.

Dins les situacions d'error possibles, es poden diferenciar els errors propis del sistema i els errors propis del desenvolupament, que s'han definit per informar que algun dels requisits establerts no es compleix. Dins el primer grup s'hi pot trobar errors del tipus: valors duplicats, registres no trobats, valors null no admesos, etc. En el segon grup s'hi pot trobar errors dels tipus: poques accions disponibles, no es poden posar a la venda accions no actives, ordre ja existent, etc.

Per fer el test, farem una crida als procediments passant-los paràmetres incorrectes, i mostrant el detall de l'error capturat. En el cas del test dels procediments de consulta, abans de realitzar el test, farem una càrrega massiva de dades per tenir dades suficients que facin fiables els resultats de les consultes. En aquesta càrrega inserirem dades en diferents dates, simulant el funcionament diari de la BD.

El script 07_insercions_TauFix.sql insereix les dades a les taules fixes, que no tenen implementats procediments ABM.

Els scripts 08_test_AMB_1.sql i 09_test_AMB_2.sql realitzen les crides als diferents procediments d'alta, modificació i baixa de registres, amb paràmetres correctes i incorrectes. Les crides realitzades amb paràmetres incorrectes hauran de retornar l'error esperat.

El script 10_insercionsMass.sql realitzarà la inserció massiva de dades, i deixarà les taules preparades per poder realitzar-hi consultes

El script 11_test_consultes.sql realitzarà les crides als procediments de consulta del mòdul operacional i del mòdul estadístic. Es faran crides passant paràmetres correctes, i també passant paràmetres incorrectes, i es mostrarà l'error capturat.

7.1 - Test ABM clients i accions

En aquest apartat farem el test dels procediments inclosos en els paquets clients i accions.

pkg_clients

prc_client_alta: intent d'alta de client ja existents, per provocar error unicitat ORA-00001.

prc_client_alta: intent alta passant camp clau amb valor null, per provocar error ORA-01400

prc_client_baixa: intent de baixa de client inexistent, per provocar error ORA-1403

prc_client_baixa: intent de baixa passant identificador null, per ORA-01400

prc_client_modificacio: intent de modificar client inexistent, per provocar error ORA-01403.

prc_client_modificacio: intent de modificació, passant id null, per provocar ORA-01400.

pkg_accions

prc_accio_alta: intent alta acció ja existent, per provocar ORA-00001.

prc_accio_alta: intent alta, passant camp clau amb valor null, per provocar error ORA-01400

prc_accio_modificacio: intent modificació acció inexistent, per provocar ORA-01403.

prc_accio_modificacio: intent de modificació, passant id null, per provocar ORA-01400.
prc_accio_modificacio: intent modificació acció inactiva, per provocar error personalitzat.
prc_accio_baixa: intent baixa acció inexistent, per provocar error ORA-01403.
prc_accio_baixa: intent de baixa passant id null, per provocar error ORA-01400.
prc_accio_preu: intent assignació de preu a acció inactiva, per provocar error personalitzat.
prc_accio_preu: intent assignació de preu a acció inexistent, per provocar error ORA-01403.
prc_accio_preu: intent assignació de preu passant id null, per provocar error ORA-01400.

7.2 - Test ABM ordres i cartera

En aquest apartat farem el test dels procediments inclosos en els paquets ordres i cartera. Els procediments del paquet cartera habitualment no els executarà cap actor extern, sinó que seran executats per altres procediments, principalment procediments del paquet ordres. Tot i així, es planifica una sèrie de proves per validar-ne el funcionament.

pkg_ordres

prc_ordre_alta: intent d'alta nova ordre de client que ja en té una de pendent, per provocar error personalitzat.
prc_ordre_alta: intent alta nova ordre passant identificadors d'accio/client inexistent, per provocar error ORA-01403.
prc_ordre_alta: intent alta nova ordre passant identificadors null, per provocar error ORA-01400.
prc_ordre_execucio: intent execució ordre inexistent, per provocar error ORA-01403.
prc_ordre_execucio: intent execució ordre passant id null, per provocar error ORA-01400.
prc_ordre_execucio: intent execució d'una ordre en estat diferent al PD, per provocar error personalitzat.
prc_ordre_modificacio: intent de modificar una ordre amb estat diferent al PD, per provocar error personalitzat
prc_ordre_modificacio: intent de modificar ordre passant id inexistent, per provocar error ORA-01403.
prc_ordre_modificacio: intent de modificar ordre passant id null, per provocar error ORA-01400
prc_ordre_modificacio: intent de modificar nº accions d'una ordre de venda per sobre de la quantitat permesa, per provocar error personalitzat.
prc_ordre_cancelacio: intent de cancel·lar una ordre en un estat diferent al PD, per provocar error personalitzat.
prc_ordre_cancelacio: intent de cancel·lar una ordre passant un id inexistent, per provocar error ORA-01403
prc_ordre_cancelacio: intent de cancel·lar una ordre passant un id null, per provocar error ORA-01400

pkg_cartera

prc_cartera_adjudicar: intent de adjudicar accions passant identificadors inexistents, per provocar error ORA-01403

prc_cartera_adjudicar: intent de adjudicar accions passant identificadors null, per provocar error ORA-01400

prc_cartera_retirar: intent de retirar accions passant identificadors inexistents, per provocar error ORA-01403

prc_cartera_retirar: intent de retirar accions passant identificadors null, per provocar error ORA-01400

prc_cartera_retirar: Intent de retirar més accions que les que disposa el client, per provocar error personalitzat.

prc_cartera_actualitzar: Intent d'actualitzar la cartera passant identificador d'acció no vàlid, per provocar ORA-01403

prc_cartera_actualitzar: Intent d'actualitzar la cartera passant paràmetres null, per provocar ORA-01400

7.3 – Anàlisi de les proves

Amb la realització de les proves planificades, es pot assegurar que el funcionament normal de la BD està garantit, ja que s'han assajat tots els procediments davant les situacions més habituals.

Aquesta fase de realització de les proves ha servit per detectar diversos errors en els procediments, depurar-los i, un cop solucionats, ha convertit el disseny molt més robust.

Malgrat que es pot considerar que l'assaig de qualitat s'ha superat, no es pot garantir que en la primera fase d'exploració de la BD pugui sorgir algun altre dels menys freqüents.

8 - Costos d'accés del mòdul estadístic

Un dels requisits del projecte és que les consultes realitzades al mòdul estadístic (DWH) tinguin un cost 1, és a dir que no depenguin del nombre de files de les taules. Es demana això, per què és habitual que les taules dels DWH tinguin moltes files, i això aniria penalitzant la consulta de la informació. L'accés als valors en una columna no indexada requereix que es faci un nombre indeterminat de lectures, fins a un màxim igual al nombre de files, per trobar el valor cercat, ja que és molt possible que el valor que estem cercant no estigui emmagatzemat de forma ordenada en l'espai físic. La utilització dels índex millora el rendiment dels accessos per valor.

8.1 - Costos prc_cns_beneficimesClient

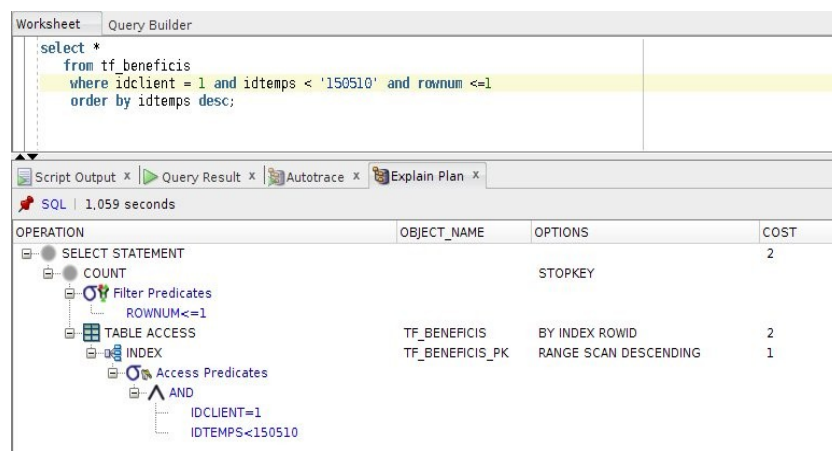


Figura 22: Costos planificador prc_cns_beneficimesClient

En la figura 22 es pot veure el pla d'execució de la consulta associada al procediment prc_benefici_mesClient, que ha establert el planificador de consultes de la BD. Com es pot veure s'accedeix a les dades utilitzant l'índex associat a la clau primària de la taula TF_beneficis.

8.2 - Costos prc_cns_accDisponibles

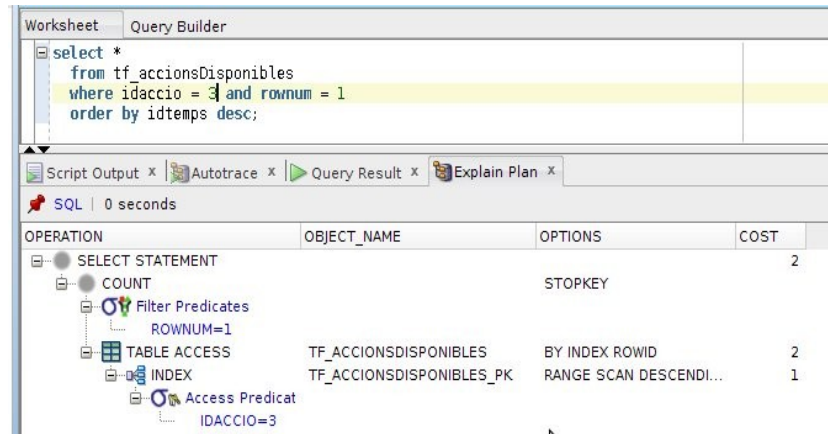


Figura 23: Costos planificador prc_cns_accionsDisponibles

En la figura 23 es pot veure el pla d'execució de la consulta associada al procediment prc_cns_accDisponibles, que defineix el planificador. En aquest cas també s'accedeix a la fila utilitzant l'índex associat a la clau primària.

8.3 - Costos prc_cns_clientCarteraMax

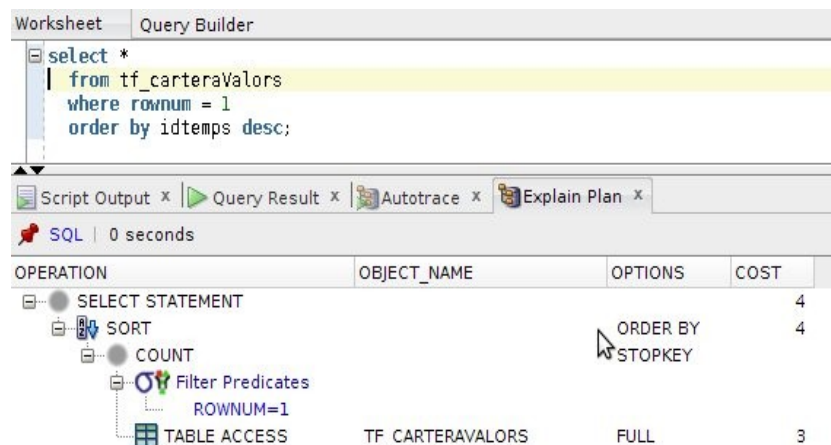


Figura 24: Costos planificador prc_cns_clientcarteramax

En la figura 24 imatge es pot veure la consulta associada al procediment prc_cns_clientCarteraMax. En aquest cas es fa un recorregut complet per la taula TF_carteravalors. És a dir, que el cost depèn del nombre de files. Per millorar el cost, definirem un índex sobre la columna idtemps.

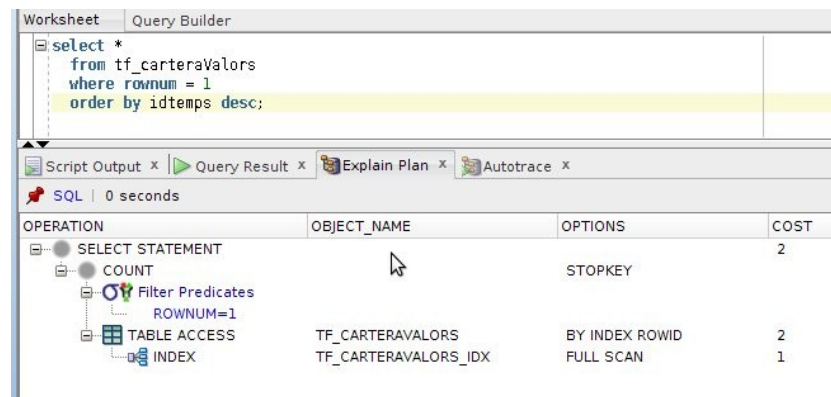


Figura 25: Costos planificador prc_cns_clientcarteramax amb índex

En la figura 25, es veu que un cop definit l'índex, el planificador ja l'està utilitzant per millorar el rendiment de la consulta.

8.4 - Costos prc_cns_accioMesNegociada

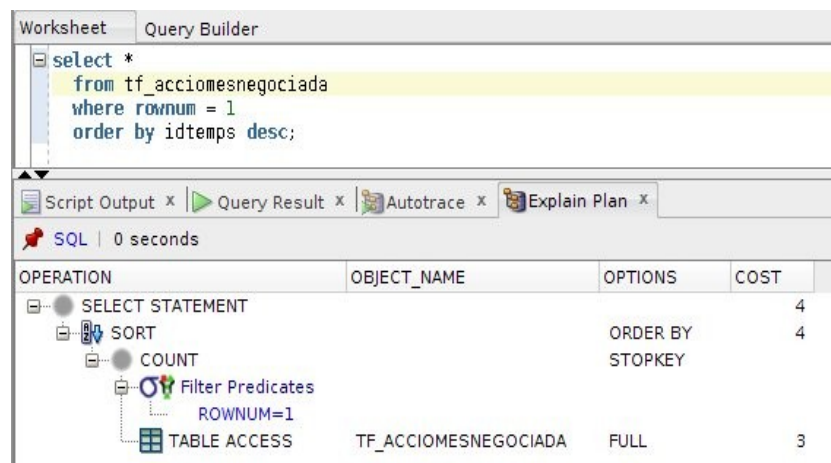


Figura 26: Costos planificador prc_cns_acciomesnegociada

En la figura 26 es pot veure la consulta associada al procediment prc_cns_accioMesNegociada. En aquest cas també es fa un recorregut complet per la taula TF_acciomesnegociada. Per millorar el rendiment, hem de definir un índex sobre la columna idtemps.

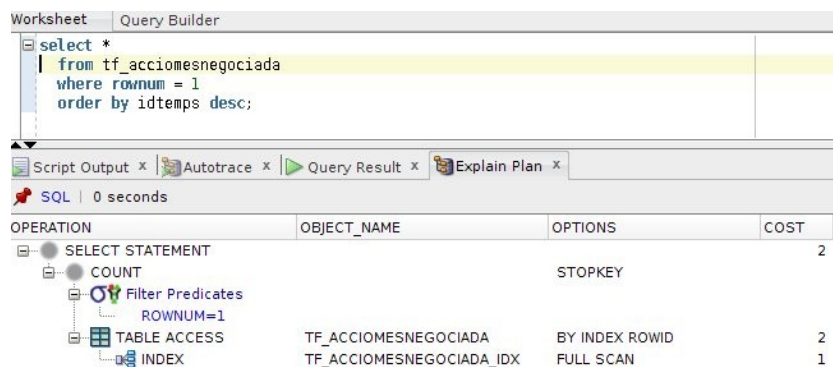


Figura 27: Costos planificador *prc_cns_acciomesnegociada* amb índex

En la figura 27, es pot veure que un cop creat l'índex, el planificador l'utilitza per millorar el rendiment de la consulta.

9 – Seguretat

En l'aspecte de la seguretat s'ha de considerar diferents conceptes. El primer és la confidencialitat, després la disponibilitat i finalment la integritat. És a dir, hem de protegir les dades d'accessos no autoritzats, les hem de mantenir a disposició dels usuaris i garantir que no es perdin, i finalment hem d'evitar que es facin modificacions no autoritzades.

En l'apartat de confidencialitat, com que recollim dades personals i financeres dels clients, estarem obligats per llei a definir un nivell de seguretat mitjà, i les mesures hauran de quedar especificades en el document de seguretat. El RD 1720/2007, en el capítol II del títol VIII, defineix com ha de ser el document de seguretat, i en el capítol III del títol VIII, defineix les mesures de seguretat aplicables.

Per tant, hem d'assegurar que cada client només pugui veure les seves dades, donat que són dades confidencials del client i crítiques pel negoci. Aquest nivell de seguretat implica que hem de restringir l'accés a les dades a nivell de files. Aquesta forma d'actuar s'anomena 'fine-grained access control'.

Per implementar aquest punt considerem 2 alternatives possibles.

La primera es basa en crear una vista parametrizada de les taules que contenen dades sensibles, que mostri les dades filtrades per client. Després es donarà accés als clients només a aquestes vistes.

Per exemple, per què el client1, que s'identifica a la BD amb el nif = 77999888A pugui accedir a la taula ordres, que conté ordres de tots els clients, i per tant la informació és sensible, farem:

```
CREATE or REPLACE VIEW ordresClient1_view AS
select * from ordres o
join clients cl
on o.client = cl.idclient
where cl.nif = user;
```

```
CREATE USER "77999888A"
IDENTIFIED BY abc1234
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp
ACCOUNT unlock;
```

```
grant create session to "77999888A";
grant select on admin_tfc. ordresClient1_view to "77999888A";
```

Per facilitar l'accés, també es podria crear un sinònim públic i que el client accedeixi a la vista definida a través del sinònim.

Amb aquesta opció assegurem que el client "77999888A" només vegi les dades que la vista li permet veure. La solució és extensible a la resta de clients ja que les filtra per user.

La segona opció, que potser és més elaborada, passa per utilitzar la funcionalitat de Virtual private database (VPD) que ofereix la Enterprise Edition de Oracle. Aquesta eina permet un control d'accés fi a files i columnes, i permet definir atributs de sessió. Quan un usuari accedeix a una taula, vista o sinònim protegit amb VPD, qualsevol ordre DML se li afegeix una clàusula where que implementa les polítiques de seguretat definides.

Aquesta solució és molt més simple de mantenir i més flexible que la primera, ja que les polítiques poden definir-se de forma global. Malauradament, aquesta funcionalitat de VPD no està disponible

en la versió de BD Oracle SE, ja que només ho està en la versió de pagament Enterprise Edition (EE).

Per implementar-ho d'aquesta segona forma caldria crear una funció que defineixi la variable de contexte i una altra que defineixi les polítiques de seguretat. Finalment caldria activar la política.

```
-- definició del procediment que configura la variable del contexte de seguretat
PROCEDURE ctx_seguretat
IS
    numClient NUMBER( 6);

BEGIN

    SELECT idClient into numClient
    from clients
    where nif = SYS_CONTEXT('USERENV', 'SESSION_USER');

    -- defineix la variable del contexte
    DBMS_SESSION.SET_CONTEXT ('ADMIN_TFC.CTX_USUARI', 'VC_ID_CLIENT', numClient);

END ctx_seguretat;

-- funció que controla les polítiques de seguretat de la BD
FUNCTION pcy_seguretat (
    esquema_nom VARCHAR2,
    taula_nom VARCHAR2
)
RETURN VARCHAR2
IS
-- declaració de variables locals del procediment
    sentència VARCHAR2 (500 char);
BEGIN
    sentència := 'client = SYS_CONTEXT(''CTX_USUARI'', ''VC_ID_CLIENT'')';
    RETURN sentència;

END pcy_seguretat;

-- activació de la política de seguretat
BEGIN
    DBMS_RLS.ADD_POLICY (
        object_schema => 'ADMIN_TFC', -- esquema que es vol protegir
        object_name    => 'ORDRES', -- objecte que es vol protegir
        policy_name     => 'POLITICA_ACCES', -- nom de la política de seguretat
        function_schema => 'ADMIN_TFC', -- esquema on s'ha creat la funció de seguretat
        policy_function => 'pkg_control.pcy_seguretat', -- funció que implementa la política
        statement_types => 'select, insert, update, delete' -- operacions que aplica la política
    );
END;
```

Amb això, aconseguirem que al final de cada sentència DML s'hi afegixi, per exemple, la clàusula `where client = 1` (on 1 és el id del client). Aquesta solució és flexible ja que funciona per tots els clients.

Es podria definir una política diferent només modificant el codi de la funció que la defineix.

Un altre element que ajudarà en l'aspecte de la seguretat, és el programari que farà la interfície visual. Aquest programari, convenientment implementat, aportarà un altre grau de seguretat al sistema, ja que a partir de variables d'entorn podrà filtrar els resultats mostrats per pantalla.

10 – Valoració econòmica del projecte

El pressupost realitzat només té en consideració les hores destinades al desenvolupament del projecte. No s'inclou cap apartat referent a les inversions necessàries per comprar el software i el hardware sobre el que correrà el projecte. Conceptes com servidors, dispositius de xarxa, encaminadors, dispositius d'emmagatzemament, llicències d'Oracle, etc, no s'han valorat.

Per a realitzar el càlcul de pressupost, s'han comptabilitzat les hores destinades a les diferents tasques que inclou el projecte, i s'han facturat segons el cost establert per la qualificació de la persona que ha executat aquella part del treball.

Hem dividir les feines en els rols explicats en el punt anterior: Cap de projecte, analista, programador i documentalista.

	PAC1	PAC2	PAC3	Lliurament Final	Total Hores	Total
cap de projecte	16	15		20	51	3.060,00 €
analista		25		10	35	1.750,00 €
programador		20	44	18	82	3.280,00 €
documentalista	6	8	10	20	44	1.100,00 €
						9.190,00 €
					IVA 21,00%	1.929,90 €
					TOTAL	11.119,90 €

Figura 28: Quadre valoració econòmica del projecte

Aleshores, segons els càlculs realitzats el cost de projecte és de 9.190€, sense IVA.

11 – Milllores futures

Ha quedat fora de l'abast del projecte realitzar una configuració dels tablespaces més ajustada a les necessitats reals de les taules i les dades que contenen. Per exemple, incrementant el blocksize del tablespace assignat al DWH milloraria el rendiment, ja que es reduiria el nombre d'accessos al disc. En canvi els tablespaces destinats a taules petites o taules molt transaccionals, seria convenient treballar amb un blocksize més petit, que ajudaria a evitar la fragmentació de la informació dins del block. També queda pendent definir una dimensió inicial de la taula i dels seus extends ajustat a les dades que contindran.

També s'hauria de treballar l'aspecte del ratio d'emplenat de cada taula (PCTFREE i PCTUSED). Per taules en les que no hi ha ni modificacions ni baixes de registres, és indicat fer emplenat del 100% (no deixar espai lliure), en canvi en taules amb moltes altes, baixes i modificacions, és més convenient definir ratio d'espai lliure proper al 50%.

També caldria analitzar les diferents consultes que es realitzen en el mòdul operatiu per veure si podem millorar-ne el rendiment creant índexs específics.

Ha quedat pendent l'anàlisi i implementació de taules que permetin gestionar altres productes a més de les accions.

En l'aspecte de la seguretat, queda pendent confeccionat el Document de Seguretat. També caldria treballar en la seguretat de les dades per permetre només accessos autoritzats a les dades, segons el que s'ha comentat en l'apartat 9.

12 - Conclusions

Una de les conclusions que he arribat és que cal un recull detallat de les necessitats del client per poder desenvolupar amb èxit un projecte. No hi poden haver dubtes en els requisits. Per aconseguir-ho és fonamental establir una bona comunicació amb el client.

També és primordial tenir una bona planificació de les tasques que s'han de realitzar. Però a més, és absolutament necessari realitzar controls periòdics de la situació del projecte i comparar-la amb la planificada, per tal d'anar corregint les desviacions detectades el més ràpid possible. Considero que una desviació no detectada a temps pot provocar entregar el projecte fora de termini.

També considero que un dels punts importants del projecte és que s'hagi fet una bona definició i estigui molt ben documentada. Això inclou detallar clarament les classes, els casos d'ús, els actors. Aquesta definició clara serà imprescindible a l'hora de començar a desenvolupar, ja que defineix el camí que ha de seguir el programador per obtenir el producte encarregat. Amb això voldria destacar el paper de l'analista, que el considero fonamental, i que sense la seva participació en el projecte el programador no podria avançar, o ho faria a cegues.

També vull destacar la importància de la fase de proves i depuració. Una acurada selecció dels jocs de proves ha permès detectar el major nombre d'errors possibles, que han pogut ser corregits i documentats. I això ens garanteix la qualitat del producte.

Finalment, voldria dir que la realització del projecte m'ha servit per aprofundir en coneixements dins el perfil de DBA, malgrat que he vist clarament que per dominar tots els aspectes de configuració del SGBD Oracle cal una dedicació molt més important, però per sobre de tot, m'ha servit per tenir una experiència pràctica de realització d'un projecte d'inici a final, amb aportació d'altres coneixements transversals, com la gestió, planificació o documentació.

13 - Glossari

- **ABM:** Es refereix als procediments d'alta, baixa i modificació de registres.
- **BD:** Es refereix a Base de Dades.
- **model ER:** Model entitat-interrelació de dades d'alt nivell que permet modelitzar els requisits, les especificacions i les restriccions. De l'anglès entity-relationship model.
- **Disseny conceptual:** Etapa del disseny d'una base de dades que obté una estructura de la informació de la futura base de dades independent de la tecnologia que es vol emprar.
- **Disseny lògic:** Etapa del disseny d'una base de dades que parteix del resultat del disseny conceptual i el transforma de manera que s'adapti al model de sistema gestor de bases de dades amb el qual es vol implementar la base de dades.
- **Data WareHouse:** Magatzem de dades. Normalment es refereix a una base de dades amb la informació històrica d'una organització, dissenyada i estructurada per a realitzar-hi consultes eficientment. També s'anomena BD estadística o mòdul estadístic.
- **DWH:** Vegeu Data WareHouse.
- **Cartera d'inversió:** accions que estan relacionades a un client, i tenen un preu determinat.
- **OLTP:** Sistema que facilita i administra aplicacions transaccionals, habitualment entrada de dades, recuperació i procés de transaccions.
- **OLAP:** Sistema utilitzat en el camp de la intel·ligència de negoci (Business Intelligence) dirigit a agilitzar les consultes a grans quantitats de dades, amb l'objectiu d'analitzar-les i poder treure'n conclusions.
- **Procediment ABM:** Es refereix a alta, baixa i modificació de registres.
- **SGBD :** Vegeu sistema gestor de bases de dades.
- **Sistema gestor de bases de dades:** Tipus de programari específic que serveix d'interfície entre la base de dades, l'usuari i les aplicacions que la utilitzen.
- **SO:** es refereix a sistema operatiu.
- **Ticker:** Identificador únic de l'acció.
- **Script:** És un arxiu d'ordres, arxiu de processament per lots o guió. És un programa emmagatzemat en un arxiu de text pla, que casi sempre és interpretat pel shell (o interfície amb el SO).

14 - Bibliografia

Antolín Muñoz, Chaparro, Oracle 11g Pl/SQL, Rc Libros, Madrid, 2012

Alan Beaulieu, Aprende SQL, Ediciones Anaya Multimedia, Madrid, 2009

Xavier Burgués Illa, Disseny de bases de dades, FUOC, Barcelona, 2012

Benet Campderrich Falgueras, Enginyeria del programari, FUOC, Barcelona, 2004

14.1 - Enllaços a internet

Oracle.com: https://docs.oracle.com/cd/E11882_01/index.htm

TechOnTheNet: <http://www.techonthenet.com/oracle/index.php>

BC Burleson Consulting: <http://www.dba-oracle.com/>

Me and my Ubuntu: <http://meandmyubuntulinux.blogspot.com.es/2012/05/installing-oracle-11g-r2-express.html>

DataPrix: <http://www.dataprix.com/data-warehousing-y-metodologia-hefesto>

Oracle certificación Blog 1Z-052: <http://oracle11gnotes.blogspot.com.es/>

15 - ANNEXES

15.1 - Instal·lació i proves

1. Instal·lar el SGBD ORACLE Express 11gR2, i anomenar a la BD el nom per defecte: XE . En aquest projecte s'ha instal·lat sobre una màquina virtual VirtualBox amb el SO Ubuntu 12 LTS, seguint la guia Me and my Ubuntu.
2. Instal·lar el SQLDeveloper, per tenir un entorn gràfic més còmode per treballar que el sql*plus, d'entorn text.
3. Executar el script install_tot.sh, que crearà els tablespaces, els usuaris, les taules, les seqüències, els índexs, els disparadors. En aquest punt el sistema està llest però està buit, sense cap dada.

Per executar el script cal passar com a paràmetres el nom i la clau d'un usuari administrador SYSDBA de la BD XE. Per exemple, install_tot.sh SYS oracle.

4. Executar el script inici_test.sh, que farà la càrrega de les dades necessàries i anirà executant els diferents procediments.

No cal passar cap paràmetre al script.

Els scripts generen uns fitxers de sortides *.log que registren els resultats de l'execució. Els fitxers .log principals són els que s'anomenen out...

15.1.1 – Instal·lació manual

En el cas de voler instal·lar manualment la BD, per exemple des de dins l'entorn de sql*plus, s'ha de procedir de la següent forma:

1. Amb un usuari DBA, per exemple SYS/oracle, carregar per ordre els fitxers, per crear els tablespaces i els usuaris amb els permisos
01_CreacioTablespace.sql i 02_CreacioUsuaris.sql
2. Amb l'usuari admin_tfc/tfc, carregar per ordre els fitxers, per crear els objectes i els procediments de la BD
03_CreacioTaules.sql , 04_CreacioSequencies.sql , 05_triggers.sql i 045_index.sql
pkg_control.sql , pkg_dwh.sql , pkg_cartera.sql , pkg_ordres.sql , pkg_accions.sql ,
pkg_clients.sql , pkg_consultes.sql , pkg_consultes_DWH.sql , pkg_utils.sql i
06_permisos_user.sql

En aquest punt ja tenim la BD creada i està apunt per carregar dades i fer els tests.

3. Amb l'usuari admin_tfc/tfc carregar el fitxer, per fer la càrrega les dades de les taules fixes
07_insercions_tauFix.sql
4. Amb l'usuari user_tfc/tfc carregar per ordre els fitxers, per realitzar els primers tests
08_test_ABM1.sql i 09_test_ABM2.sql
5. Amb l'usuari admin_tfc/tfc carregar el fitxer, per realitzar una càrrega massiva
10_insercionsMass.sql

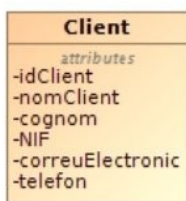
6. Amb l'usuari user_tfc/tfc carregar per ordre els fitxers, per realitzar els tests finals
11_test_consultes.sql

15.2 - Anàlisi d'entitats

Aquest annexe complementa a l'apartat 3.1 de la memòria.

15.2.1 – Entitats del mòdul d'explotació

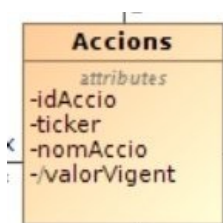
Classe	Clients
Descripció de la classe	Classe que representa la informació directament associada al client
Responsabilitats	Col·laboracions/col·laboradors
Contenir les dades dels clients	Ella mateixa
Contenir informació de la cartera de valors	Col·laboració amb la classe Accions i amb la classe associativa Cartera
Realitzar ordres	Col·laboració amb les classes Ordres i Accions



NIF: serà un identificador únic. Serà clau alternativa. Ja que no es permetrà repeticions en altres clients.

-P
0.
NomClient, cognom, NIF: no admetran valor nuls
correuElectronic, telefon:admeten valors nuls

Classe	Accions
Descripció de la classe	Classe que representa la informació directament associada les accions
Responsabilitats	Col·laboracions/col·laboradors
Contenir les dades de les accions	Col·labora amb la classe Mercat per determinar el mercat de negociació de l'acció
Formar part de la cartera de valors	Col·laboració amb la classe Clients i amb la classe associativa Cartera
Participar en les ordres	Col·laboració amb les classes Ordres i Clients
Mantenir històric de valors de les accions	Col·labora amb la classe HistòricAccions per mantenir el històric



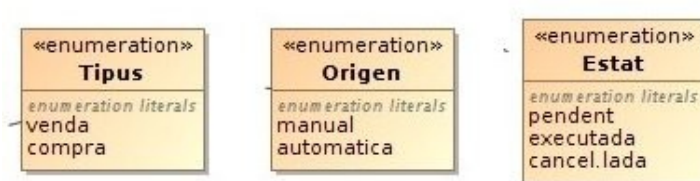
El identificador ticker és un identificador únic. Serà clau alternativa.
El valorVigent és el camp que contindrà el valor vàlid de l'acció en la data actual.

NomAccio no admet valors nuls

Classe	HistoricAccions
Descripció de la classe	Classe que representa el històric del valor de les accions
Responsabilitats	Col·laboracions/col·laboradors
Contenir les dades històriques del valor de les accions	Col·labora amb la classe Accions per determinar el valor

Classe	HistoricCartera
Descripció de la classe	Classe que representa el històric del valor i la quantitat de la cartera
Responsabilitats	Col·laboracions/col·laboradors
Contenir les dades històriques del valor i quantitat de les carteres.	Col·labora amb la classe cartera per determinar el valor

Classes Enumeració



Classes que contindran els valor possibles que es poden assignar als paràmetres tipus de venda, Origen i estat. No tindran procediments ABM.

En la implementació s'optarà per transformar-les en una restricció de tipus domini de la variable, ja que en els 3 casos el nombre de valors és molt reduït, i no es preveu que s'hagi d'augmentar.

15.2.2 – Entitats del mòdul d'estadístic (Data warehouse)

Classe	TF_Beneficis
Descripció de la classe	Classe que representa els beneficis nets mensuals i l'increment de valor mensual de la cartera d'accions, en %, dels clients.
Responsabilitats	Col·laboracions/col·laboradors
Mantenir el registre de beneficis mensuals, en €, de cada client.	Col·labora amb les classes TD_Temps i TD_Client per fer el registre.
Mantenir l'increment de valor mensual de la cartera de valors, en %, de cada client.	Col·labora amb les classes TD_Temps i TD_Client per fer el registre.

Classe	TF_CarteraValors
Descripció de la classe	Classe que representa el nombre d'accions i el valor de la cartera del client en un moment determinat.
Responsabilitats	Col·laboracions/col·laboradors
Mantenir el registre del nombre d'accions en propietat del client.	Col·labora amb les classes TD_Temps i TD_Client per fer el registre.
Mantenir el registre del valor de les accions en propietat del client.	Col·labora amb les classes TD_Temps i TD_Client per fer el registre.

Classe	TF_AccioMesNegociada
Descripció de la classe	Classe que representa el nombre de moviments que ha tingut cada acció de forma històrica.
Responsabilitats	Col·laboracions/col·laboradors
Mantenir el registre del nombre de moviments que ha tingut cada acció.	Col·labora amb les classes TD_Temps i TD_Accio per fer el registre.

Classe	TF_AccionsDisponibles
Descripció de la classe	Classe que representa el nombre d'accions que hi ha posades a la venda en un moment concret i els client que les posseeixen
Responsabilitats	Col·laboracions/col·laboradors
Mantenir el registre del nombre d'accions que hi ha posades a la venda en cada moment.	Col·labora amb les classes TD_Temps i TD_Accio per fer el registre.

Classe	TD_Accio
Descripció de la classe	Classe que representa a les accions existents. Ha de contenir tants valors com accions hi hagi en el sistema.
Responsabilitats	Col·laboracions/col·laboradors
Mantenir el registre de les accions existents en la BD operativa.	

Classe	TD_Client
Descripció de la classe	Classe que representa als clients existents. Ha de contenir tants valors com clients hi hagi en el sistema.
Responsabilitats	Col·laboracions/col·laboradors
Mantenir el registre dels clients existents en la BD operativa.	

Classe	TD_Temps
Descripció de la classe	Classe que representa a la dimensió temps, que caracteritzarà a cada esdeveniment de les taules de fets. Hi haurà tants valors com entrades hi hagi en les taules de fets.
Responsabilitats	Col·laboracions/col·laboradors
Mantenir el registre dels moments o instants en els que s'ha emmagatzemat alguna informació en una de les taules de fets.	

15.2.3 – Entitats del mòdul de control

Classe	LogProcediments
Descripció de la classe	Classe que registra l'execució dels procediments
Responsabilitats	Col·laboracions/col·laboradors
Mantenir el registre d'execució dels procediments, la data i els paràmetres de la crida al procediment.	

Classe	Errors reconeguts
Descripció de la classe	Classe que conté els tipus d'error i els seus missatges associats.
Responsabilitats	Col·laboracions/col·laboradors
Centralitzar la informació relacionada als missatges d'error llançats pels diferents procediments.	

15.3 – Anàlisi de casos d'ús

Aquest annexe complementa a l'apartat 3.3.2 de la memòria.

15.3.1 - Anàlisi dels casos d'ús: mòdul operacional

Cas d'us	001 – alta client
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol donar d'alta un nou client
Precondició	L'usuari ha de tenir obligatòriament el rol de broker
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema crearà el nou client, i ho registrarà a la taula log
Escenari principal d'èxit	1- s'introdueixen les dades del nou client 2- el sistema crea el nou client 3- el sistema registra el moviment a la taula de log
Extensions	S'actualitza la TD_clients del mòdul estadístic

Cas d'us	002 – Modificació client
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol modificar alguna de les dades modificables del client. Es consideren modificables nom, cognoms, nif, correu-e i telèfon
Precondició	L'usuari ha de tenir obligatòriament el rol de broker
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema modificarà les dades del client, i ho registrarà a la taula log
Escenari principal d'èxit	1- s'introdueixen les noves dades dades del client 2- el sistema modifica les dades del client 3- el sistema registra el moviment a la taula de log
Extensions	

Cas d'us	003 – Baixa client
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol donar la baixa lògica a un client
Precondició	L'usuari ha de tenir obligatòriament el rol de broker

Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema desactivarà el client, i ho registrarà a la taula log
Escenari principal d'èxit	1- s'introdueix l'identificador o el nif del client que es vol donar de baixa 2- el sistema verifica si té ordres associades. Si en té les cancel·la. 3- el sistema dona de baixa (desactiva) el client 4- el sistema registra el moviment a la taula de log
Extensions	--

Donat que es tracta d'una baixa lògica, en cap cas s'ha de borrar el registre físic.

Cas d'us	004 – alta acció
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol donar d'alta una nova acció
Precondició	L'usuari ha de tenir obligatòriament el rol de broker
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema crearà la nova acció, i ho registrarà a la taula log
Escenari principal d'èxit	1- s'introdueixen les dades de la nova acció 2- el sistema crea la nova acció 3- el sistema registra el moviment a la taula de log
Extensions	--

Cas d'us	005 – Modificació acció
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol modificar alguna de les dades de les accions. Es consideren modificables el Ticker, el nom de l'acció i el mercat
Precondició	L'usuari ha de tenir obligatòriament el rol de broker. L'acció ha d'estar activa.
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema modificarà les dades de l'acció, i ho registrarà a la taula log
Escenari principal d'èxit	1- S'introdueix el ID de l'acció que es vol modificar i les noves dades de l'acció 2- el sistema modifica les dades de l'acció 3- el sistema registra el moviment a la taula de log
Extensions	--

Cas d'us	006 – Baixa acció
Actor	Broker
Àmbit	BD operativa
Nivell d'objectiu	Es vol donar la baixa lògica a una acció
Precondició	L'usuari ha de tenir obligatòriament el rol de broker
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema donarà de baixa (desactivarà) a l'acció, i ho registrarà a la taula log
Escenari principal d'èxit	<ul style="list-style-type: none"> 1- s'introdueixen el identificador de l'acció o ticker que es vol donar de baixa 2- el sistema verifica si associada a un client o a una ordre 2,1 – Si està associada a ordres: cancel·la l'ordre de compra/venda 3- el sistema dona de baixa a l'acció 4- el sistema registra el moviment a la taula de log
Extensions	--

Donat que es tracta d'una baixa lògica, en cap cas s'ha de borrar el registre físic.

Cas d'us	009 – Alta ordre compra/venda
Actor	client
Àmbit	BD operativa
Nivell d'objectiu	Es vol donar donar d'alta una ordre de compra/venda d'accions
Precondició	El clients i l'acció han d'estar actius.
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema crea l'ordre, i ho registrarà a la taula log
Escenari principal d'èxit	<ul style="list-style-type: none"> 1- s'introdueixen les dades de la nova ordre 2- el sistema crea la nova ordre 3- Si la nova ordre és de venda, s'actualitza la taula TF_accions Disponibles 4- el sistema registra el moviment a la taula de log
Extensions	

Cas d'us	010 – Modificació ordre compra/venda
Actor	client
Àmbit	BD operativa
Nivell d'objectiu	Es vol modificar alguna dada d'una ordre de compra/venda d'accions. Es permet modificar els valors de quantitat i preu Objectiu. Només es permeten modificar ordres en situació PD
Precondició	L'ordre ha d'estar PD
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema modifica l'ordre amb els nous valors, i ho registrarà a la taula log
Escenari principal d'èxit	1- s'introdueix l'identificador de l'ordre que es vol actualitzar, amb la quantitat i el preu objectiu 2- Si l'ordre està en situació pendent, el sistema modifica l'ordre 3- Si es modifica el preu objectiu, es revisa si es pot executar l'ordre. 3,1- Si la ordre és de venda, s'actualitza la taula TF_accions Disponibles 4- el sistema registra el moviment a la taula de log
Extensions	3b- si l'ordre ja està executada es retorna un missatge d'error

Cas d'us	011 – Cancel·lació ordre compra/venda
Actor	client
Àmbit	BD operativa
Nivell d'objectiu	Es vol anular una ordre de compra/venda d'accions. Només es permet anul·lar ordres que estan en situació PD
Precondició	L'ordre ha d'estar PD
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema cancel·larà l'ordre especificada, i ho registrarà a la taula log
Escenari principal d'èxit	1- s'introdueix l'identificador de l'ordre que es vol anular. 2- si l'ordre està en situació PD, el sistema cancel·la l'ordre 3- Si l'ordre és de Venda, s'actualitza la taula TF_accions disponibles 4- el sistema registra el moviment a la taula de log
Extensions	3b- si l'ordre ja està executada es retorna un missatge d'error

15.3.2 - Anàlisi dels casos d'ús: mòdul estadístic (DWH)

Cas d'us	015 – Actualització TF_CarteraValors
Actor	Sistema
Àmbit	BD estadística
Nivell d'objectiu	Es vol actualitzar la taula de fets de cartera de valors, que conté els valor màxims de les carteres dels clients.
Precondició	S'ha actualitzat la cartera de valors d'un client o s'ha actualitzat el valor de les accions. Es considera el valor màxim, i no la màxima quantitat.
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema calcula el valor de la cartera d'accions del client, actualitzarà la taula TF_CarteraValors, i ho registrarà a la taula log del DW
Escenari principal d'èxit	1- El sistema fa la consulta del client o clients que tenen la cartera amb el valor màxim fins aquell moment. 2- S'actualitza la TF_cartera valors, amb els clients, nombre accions, i valor de la cartera 3- El sistema registra el moviment a la taula de log
Extensions	

Cas d'us	016 – Actualització TF_AccioMesNegociada
Actor	Sistema
Àmbit	BD estadística
Nivell d'objectiu	Es vol actualitzar la taula de fets de les accions més negociades de forma històrica
Precondició	S'ha executat una ordre de compra/venda. Es consideren les accions associades a ordres ja executades.
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema afegeix al registre el nombre d'accions que s'han executat, actualitzant la TF_AccioMesNegociada, i ho registrarà a la taula log del DW
Escenari principal d'èxit	1- El sistema fa la consulta de l'acció o accions que s'han negociat més vegades fins aquell moment 2- S'actualitza la TF_accioMes Negociada amb l'acció i el nombre de moviments 3- El sistema registra el moviment a la taula de log
Extensions	

Cas d'us	017 – Actualització TF_AccionsDisponibles
Actor	Sistema
Àmbit	BD estadística
Nivell d'objectiu	Es vol actualitzar la taula de fets de les accions que estan disponibles
Precondició	S'ha actualitzat la taula d'ordres de compra/venda; hi ha hagut un moviment, s'ha executat, creat, modificat o cancel·lat una ordre.
Garanties mínimes	El sistema retornarà un paràmetre de sortida informant de l'error
Garanties cas d'èxit	El sistema fa el càlcul de les accions que estan en ordres de venda no executades, actualitza la TF_AccionsDisponibles, i ho registrarà a la taula log del DW
Escenari principal d'èxit	1- Per a cada acció que participa en una ordre de venda, el sistema fa el recompte de la quantitat i dels clients que la tenen en la cartera 2- Actualitza la TF_AccionsDisponibles , entrant l'acció, el nombre que hi ha a la venda i el nombre de clients que la tenen associada 3- El sistema registra el moviment a la taula de log
Extensions	

15.3.3 - Anàlisi dels casos d'ús: mòdul de control

Cas d'us	018 – Control d'accés
Actor	Usuari (en general, qualsevol que accedeix a la BD)
Àmbit	BD, sistema de control
Nivell d'objectiu	Permetre l'accés a les dades, procediments i recursos només als usuaris autoritzats.
Precondició	L'usuari ha d'estar activat en la BD i assignat a un rol
Garanties mínimes	El sistema retornarà un missatge informant que s'ha denegat l'accés.
Garanties cas d'èxit	L'usuari tindrà els permisos assignats al rol que té.
Escenari principal d'èxit	1- L'usuari s'identifica 2- El sistema l'autentifica, i li permet executar les operacions autoritzades. 3- El sistema registra l'accés de l'usuari al registre d'accessos.
Extensions	

Cas d'us	019 – Registre d'activitats
Actor	Sistema
Àmbit	BD, sistema de control
Nivell d'objectiu	Registrar totes les crides realitzades a procediments emmagatzemats en una taula d'auditoria.
Precondició	
Garanties mínimes	El sistema registrarà l'execució del procediment i retornarà un error d'execució
Garanties cas d'èxit	El procediment quedarà registrat, amb la data, i els paràmetres que han intervingut. Es valorarà registrar l'usuari que ha executat el procediment.
Escenari principal d'èxit	1- S'executa un procediment 2- Es registra el nom del procediment, els seus paràmetres d'entrada, el resultat, la data d'execució i l'usuari que l'ha llançat, en la taula de log.
Extensions	

