

DISSENY I IMPLEMENTACIÓ SISTEMA DE GESTIÓ D'AMONESTACIONS I SANCIONS EN CENTRES EDUCATIUS

Pla de treball

Antoni Mercader Gimeno
ETIG

Manel Rella Ruiz

20 de març de 2011

Dedicatòria

A la meva dona Berta, has estat sempre al meu cantó, suportant el meu mal humor dels últims dies. Donant-me suport i ajudant-me en els moments mes durs.

A la Família que sempre ha estat quan la necessitava.

I en especial a una persona que ja no hi es, i que somiava amb veure el seu "nen" amb la carrera acabada. Gracies "laia" això va per tu.

ÍNDIX

1. Introducció	4
1.1 Descripció del TFC.	4
1.2 Objectius del projecte.	4
1.3 Recursos emprats.	5
1.4 Gestió de riscos.	6
1.5 Planificació.	6
1.5.1 Tasques.	6
1.5.2 Fites.	8
1.5.3 Planificació temporal.	9
2. Anàlisi de requeriments.	11
2.1 Requeriments funcionals.	11
3. Disseny del sistema de base de dades.	13
3.1 Disseny conceptual	13
3.2 Disseny lògic.	14
3.2.1 Model relacional.	14
3.2.2 Restriccions d'integritat.	17
3.3. Disseny físic.	18
4. Implementació.	18
4.1 Creació de taules.	19
4.2 Disparadors i procediments relacionats.	20
4.3 Índexs.	23
4.4 Procediments.	25
4.5 Casos d'us.	27
4.6 Mòdul Estadístic.	52
5. Proves.	56
6. Valoració econòmica.	59
7. Conclusions.	60
8. Bibliografia.	61

Descripció del projecte.

En aquest projecte es proporcionarà al client, la Conselleria d'Ensenyament de la Generalitat de Catalunya, el sistema de base de dades necessari per a la implementació d'una Base de Dades per a l'aplicació de gestió d'amonestacions i sancions en centres educatius d'ensenyament secundari. Caldrà cobrir totes les etapes del cicle de vida: planificació, anàlisi, disseny i prova.

El context del projecte és el canvi de Govern de la Generalitat de Catalunya, que preveu la creació de una nova aplicació. De moment ens centrarem només en el disseny de la Base de dades, ja que l'aplicació de la gestió es definirà en una segona fase del pla de sistemes d'informació de la Generalitat de Catalunya.

Del sistema es requereix per una banda que proporcioni l'estructura de dades necessària per a la realització del projecte, i per altra un mòdul estadístic que li permeti fer un seguiment de les amonestacions.

Objectius del projecte.

El treball final de carrera (TFC) té com a objectiu materialitzar els coneixements adquirits durant la carrera, en aquest cas l'Enginyeria Tècnica en Informàtica de Sistemes, en la consecució d'un projecte real en totes les seves fases, des de l'Anàlisi previ i presa de requeriments fins a la prova i posada en marxa.

Aquest TFC se situa en l'àmbit de les bases de dades relacionals, de manera que, a més dels conceptes propis de les assignatures d'Enginyeria del Programari i Competències Comunicatives per a Professionals de la Informàtica, s'aprofundirà en allò après a Bases de Dades I i Bases de Dades II: anàlisi i disseny de bases de dades, llenguatge SQL, programació de procediments incloent el control d'errors i excepcions, i utilització de jocs de proves.

La consecució dels objectius es materialitzarà en l'entrega del producte, juntament amb la memòria i una presentació que resumiran el treball realitzat. Previ a l'entrega final es realitzaran tres entregues parcials:

- PAC1: el pla de treball.
- PAC2: l'Anàlisi i Disseny de la base de dades.
- PAC3: Implementació del mòdul estadístic i fase de proves.

Descripció de les tasques

1. **Instal·lació del programari.** S'instal·la Oracle Express i Microsoft Office Project en les màquines que utilitzarem.
2. **Descripció del TFC.** Es descriu el projecte que es durà a terme.
3. **Objectius del projecte.** Es detallen els objectius a assolir.
4. **Identificació de tasques.** Es divideix el projecte en una sèrie de tasques.
5. **Determinació de Fites del projecte, Riscos i Recursos.** S'avaluen aquests conceptes, necessaris per a la PAC1.
6. **Planificació temporal.** S'estableix el calendari d'execució de cada tasca.
7. **Redacció del Pla de Treball.** Es sintetitzen tots els resultats anteriors per a l'entrega de la PAC1.
8. **Validació del Pla de Treball.** S'adapta el PT a les correccions del Consultor.
9. **Anàlisi de requeriments.** Detall de les funcions que ha de realitzar el sistema.
 - 9a. **Requeriments funcionals.** Es recullen les necessitats d'informació del client.
 - 9b. **Requeriments no funcionals.** Es defineixen criteris per avaluar l'operació del sistema.
10. **Disseny del sistema de base de dades.** Inclou el que a Enginyeria del Programari anomenàvem Anàlisi (independent de la tecnologia) i Disseny (especificació de les solucions tecnològiques als requeriments d'informació).
 - 10a. **Disseny conceptual.** Estructura de la BD independent de la tecnologia. Es determina el model Entitat-Interrelació (ER) i es llisten entitats i atributs, incloent les claus primàries.
 - 10b. **Disseny lògic.** Es transforma el model ER a model relacional
 - 10b1. **Model relacional.** Del model ER es deriven les relacions i claus foranes.
 - 10b2. **Restriccions d'integritat.** Es determinen les regles d'integritat del model i les definides per l'usuari.
 - 10c. **Disseny físic.** Transformació del model lògic per raons d'eficiència (per exemple, dividint o combinat relacions), i aspectes d'implementació física (per exemple, tamany dels buffers). Definició d'índexs.
11. **Implementació.** Generació del codi que donarà lloc a la base de dades.
 - 11a. **Creació de taules.** Scripts de generació de les taules.
 - 11b. **Disparadors i procediments relacionats.** Scripts que definiran els disparadors i els procediments necessaris per a implementar-los.
 - 11c. **Procediments.** Els procediments emmagatzemats que es facilitaran a l'usuari per satisfer les funcionalitats demanades.
12. **PAC2.** redacció de la PAC2 a partir dels resultats anteriors.
13. **Mòdul estadístic.** Implementació del mòdul estadístic a partir dels requeriments de l'usuari.
 - 13a. **Definició de consultes.** Elaboració de la consulta SQL que satisfà cada requeriment estadístic.

- 13b. Definició de taules.** A partir de les consultes, creem les taules on es guardaran les dades.
- 13c. Modificació de procediments.** Es revisen els procediments creats anteriorment per a que alimentin el mòdul estadístic.
- 13d. Creació de procediments.** Es creen els procediments que permetran extreure informació del mòdul estadístic.
- 14. Proves.** Crear i executar el joc de proves. Depurar errors.
- 15. PAC3.** Es redacta la PAC3 amb els resultats anteriors.
- 16. Redacció de la Memòria.** Es completa el redactat de la memòria final.
- 17. Presentació virtual.** Es prepara la presentació PowerPoint que resumirà el projecte.

Fites del projecte.

Les dates clau que caldrà tenir en compte per al desenvolupament del projecte són les següents:

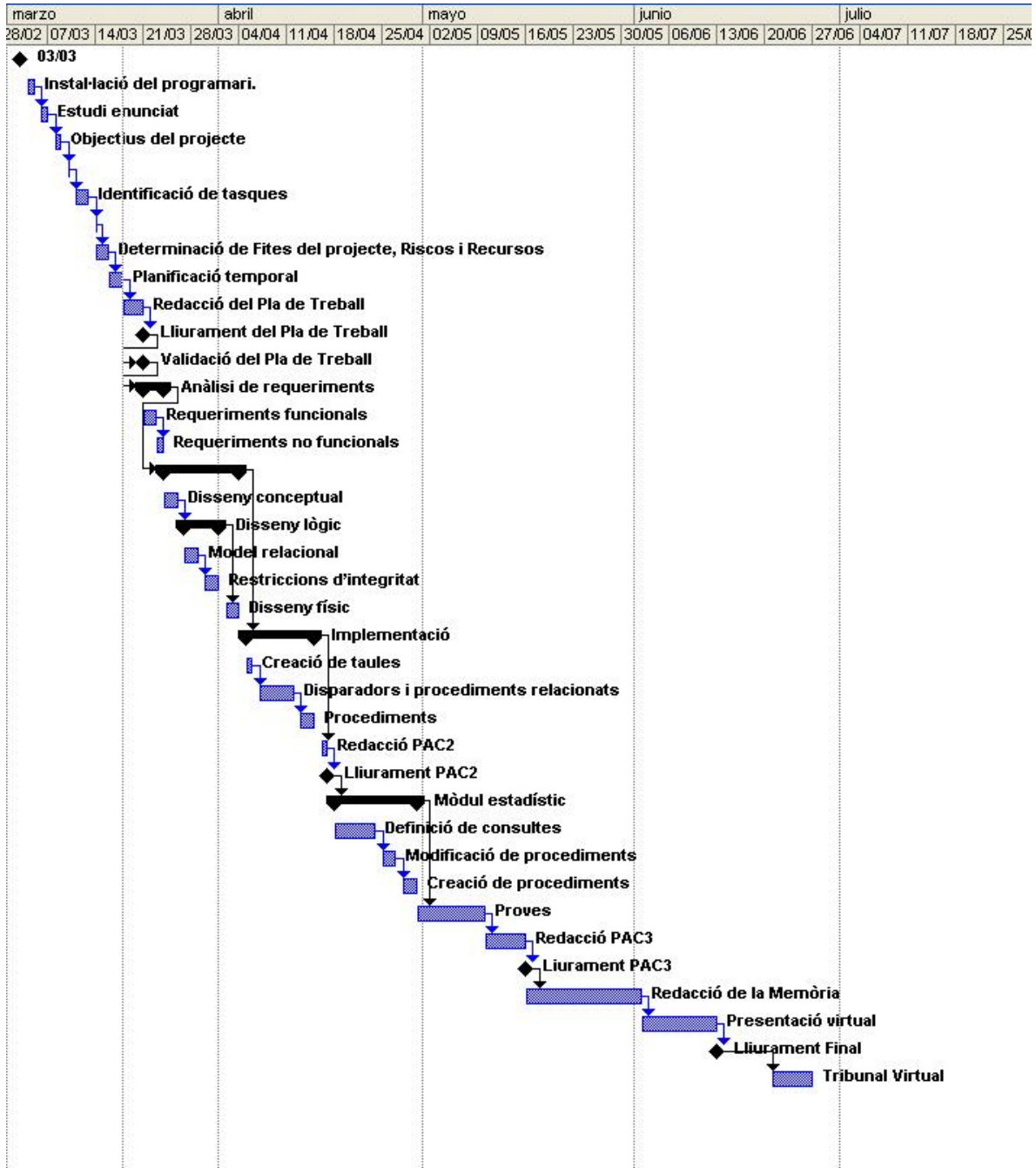
DATA	NOM DE LA FITA	DESCRIPCIÓ
03/03/11	Inici del curs	Data de començament del semestre acadèmic.
12/03/11	Trobada presencial	Reunió per discutir aspectes generals i requeriments del projecte.
20/03/11	PAC1	Lliurament del Pla de Treball.
17/04/11	PAC2	Lliurament de l'Anàlisi i Disseny de la base de dades, i la implementació sense mòdul estadístic.
15/05/11	PAC3	Lliurament de la base de dades completa amb el joc de proves.
12/06/11	Entrega final	Lliurament del producte, la memòria i la presentació.
20/06/09	Inici Tribunal Virtual	S'inicia el període de preguntes per part del tribunal.
26/06/09	Fi Tribunal Virtual	Es tanca el període de preguntes del Tribunal.

Planificació temporal.

Calendari.

	Nombre de tarea	Duración	Comienzo	Fin
1	Inici semestre	0 horas	jue 03/03/11	jue 03/03/11
2	Instal·lació del programari.	4 horas	jue 03/03/11	vie 04/03/11
3	Estudi enunciat	4 horas	sáb 05/03/11	dom 06/03/11
4	Objectius del projecte	4 horas	lun 07/03/11	mar 08/03/11
5	Descripció del TFC	2 horas	mié 09/03/11	mié 09/03/11
6	Identificació de tasques	6 horas	jue 10/03/11	sáb 12/03/11
7	Trobada Presencial	1,5 horas	dom 13/03/11	dom 13/03/11
8	Determinació de Fites del projecte, Riscos i Recursos	3 horas	dom 13/03/11	mar 15/03/11
9	Planificació temporal	5 horas	mar 15/03/11	jue 17/03/11
10	Redacció del Pla de Treball	6 horas	jue 17/03/11	dom 20/03/11
11	Lliurament del Pla de Treball	0 horas	dom 20/03/11	dom 20/03/11
12	Validació del Pla de Treball	0 horas	dom 20/03/11	dom 20/03/11
13	<input type="checkbox"/> Anàlisi de requeriments	3 días	dom 20/03/11	mié 23/03/11
14	Requeriments funcionals	4 horas	dom 20/03/11	mar 22/03/11
15	Requeriments no funcionals	2 horas	mar 22/03/11	mié 23/03/11
16	<input type="checkbox"/> Disseny del sistema de base de dades	11,25 días	mié 23/03/11	dom 03/04/11
17	Disseny conceptual	4,5 horas	mié 23/03/11	vie 25/03/11
18	<input type="checkbox"/> Disseny lògic	6 días	sáb 26/03/11	jue 31/03/11
19	Model relacional	6 horas	sáb 26/03/11	lun 28/03/11
20	Restriccions d'integritat	6 horas	mar 29/03/11	jue 31/03/11
21	Disseny físic	6 horas	vie 01/04/11	dom 03/04/11
22	<input type="checkbox"/> Implementació	11 días	lun 04/04/11	jue 14/04/11
23	Creació de taules	4 horas	lun 04/04/11	mar 05/04/11
24	Disparadors i procediments relacionats	12 horas	mié 06/04/11	lun 11/04/11
25	Procediments	6 horas	mar 12/04/11	jue 14/04/11
26	Redacció PAC2	4 horas	vie 15/04/11	sáb 16/04/11
27	Lliurament PAC2	0 horas	sáb 16/04/11	sáb 16/04/11
28	<input type="checkbox"/> Mòdul estadístic	13 días	dom 17/04/11	vie 29/04/11
29	Definició de consultes	14 horas	dom 17/04/11	sáb 23/04/11
30	Modificació de procediments	6 horas	dom 24/04/11	mar 26/04/11
31	Creació de procediments	6 horas	mié 27/04/11	vie 29/04/11
32	Proves	20 horas	sáb 30/04/11	lun 09/05/11
33	Redacció PAC3	12 horas	mar 10/05/11	dom 15/05/11
34	Lliurament PAC3	0 horas	dom 15/05/11	dom 15/05/11
35	Redacció de la Memòria	34 horas	lun 16/05/11	mié 01/06/11
36	Presentació virtual	22 horas	jue 02/06/11	dom 12/06/11
37	Lliurament Final	0 horas	dom 12/06/11	dom 12/06/11
38	Tribunal Virtual	7 días	lun 20/06/11	dom 26/06/11

Diagrama de Gantt.



Valoració de riscos.

RISC	SOLUCIÓ DE CONTINGÈNCIA	COSTOS DERIVATS
Fallada del maquinari principal	Còpia diària en el servidor secundari	Pèrdua de fins a un dia de feina. Cost de reposició del maquinari en temps i diners.
Nou projecte Laboral. Engedada d'un DataWarehouse a l'empresa que pot provocar una major dedicació a la feina	Demandar un període de vacances a la feina per a la realització del projecte.	Increment del cost del projecte.

Recursos disponibles.**Recursos de maquinari.**

MÀQUINA	TIPUS	CPU	RAM	SISTEMA OPERATIU
Servidor Principal	Fujitsu Amilo Pi-2540	Intel Core™2 Duo T5450 @ 1.66 Ghz	2 GB	Windows XP Professional SP3
Servidor Secundari	PC	Intel® Core(TM)2 6600 @2.40 Ghz	4 GB	Windows XP Professional SP3

Recursos de programari.

PROGRAMARI	SUPORT
Oracle Express v10.2.0.1	CD-Rom
MS Office 2003, Visio i Project	DVD-Rom
DIA (dibuix de diagrames)	Web

Recursos bibliogràfics.

RECURS	SUPPORT
Material de l'assignatura	Web
Enllaços del pla docent	Web
Fòrums d'Oracle	Web
Manual d'Oracle 9i	Paper

Recursos humans.

RECURS	CATEGORIA	TIPUS	DISPONIBILITAT (hores/setmana)
Antoni Mercader Gimeno	Analista/Programador	Ordinari	14

2. Anàlisi de requeriments.

2.1 Requeriments funcionals.

Del sistema es demanen una sèrie de funcionalitats que s'hauràn de satisfer mitjançant procediments emmagatzemats, la qual cosa proporcionarà una millor eficiència i seguretat que l'accés directe a les taules. Són les següents:

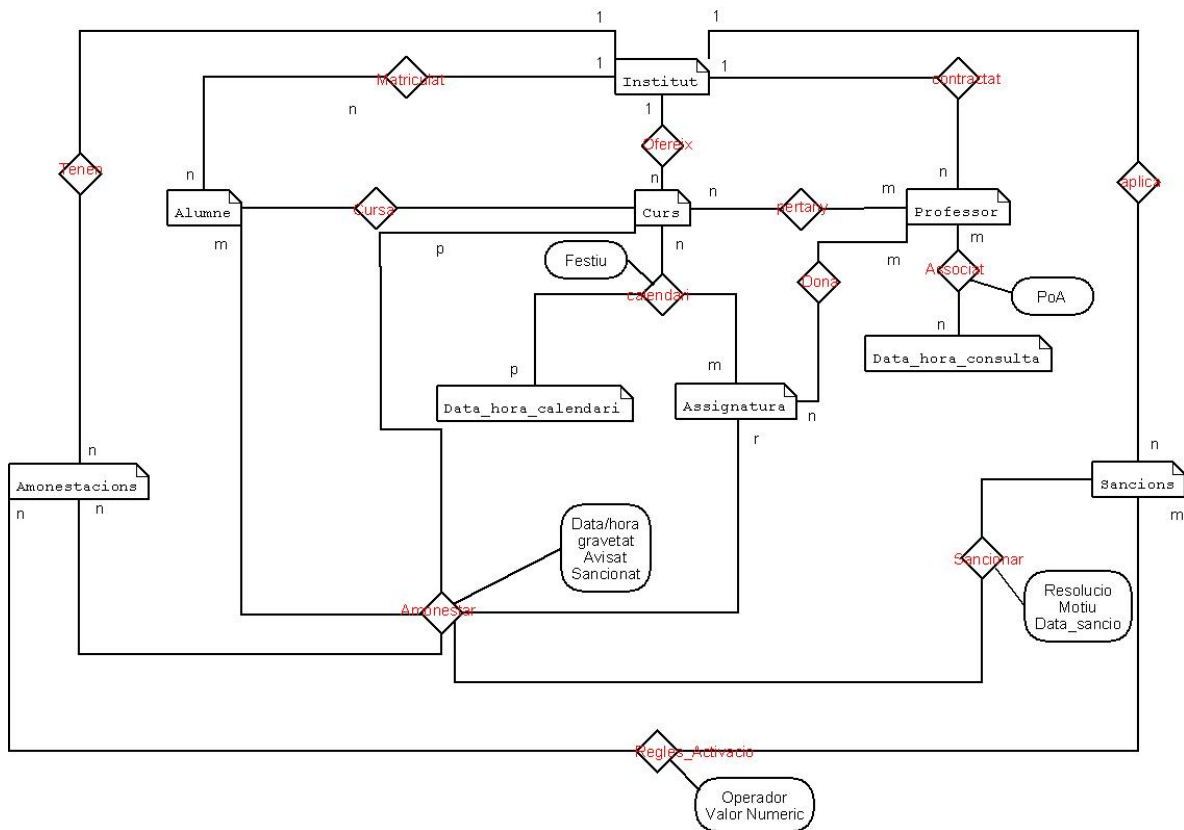
1. Gestió de la base de dades. Caldran funcions per mantenir les taules, entre elles les taules d'alumnes, professors, cursos, assignatures, amonestacions i sancions.
2. S'hauran de fer una sèrie de procediments de consulta per obtenir uns llistats en concret.
3. Mòdul estadístic. Es requereix un cert nombre de dades estadístiques que estaran disponibles a temps constant. No servirà doncs de fer consultes a la base de dades en el moment de que es demana la dada sinó que hauran d'estar disponibles en tot moment, actualitzades cada vegada que es realitzin modificacions rellevants sobre la base de dades.

3. Disseny del sistema de base de dades.

3.1 Disseny conceptual.

En aquesta primera fase es començarà a modelar l'univers del problema sense pensar encara en la implementació. S'utilitzarà el model Entitat-interrelació (ER), en el que partint de l'especificació es determinen les entitats de l'univers del problema, els seus atributs i les interrelacions entre elles.

El diagrama d'Entitats i Interrelacions queda com segueix:



La llista d'entitats amb els seus atributs és la següent (s'ha subratllat la clau primària).

INSTITUT

codi_institut, descripció

ALUMNE

Num_expedient, codi_institut, nom_alumne, adreça, e-mail, telefon, any_curs

PROFESSOR

Nif_professor, nom_professor, codi_institut.

CURS

Codi_curs, nom_curs, tutor, professor, codi_institut

ASSIGNATURA

Codi_assignatura, nom_assignatura, codi_curs, nif_professor

AMONESTACIONS

Tipus, descripcio, codi_institut

SANCIONS

ID, sanció, codi_institut

DATA_HORA_CALENDARI

data_hora

DATA_HORA_CONSULTA

data_hora

Decisions de disseny:

Veiem que Amonestar és una interrelació entre un Alumne, un curs, un professor, una assignatura i una amonestació i no pas una entitat, tot i que té tres atributs, la data i hora, la gravetat i Avisat. La cardinalitat és M-N-P-R ja que res no impedeix que un alumne tingui amonestacions per diversos cursos/assignatures/professors, o que un professor hagi amonestat a diversos alumnes per curs/assignatura. També es pot donar que en un curs s'hagin produït diverses amonestacions en assignatures diferents, fetes per professors diferents a alumnes diferents. Un altre cas que es pot donar es que per una assignatura que estigui a diferents cursos hi hagin diversos alumnes amonestats per professors diferents. I finalment es pot donar que una amonestació estigui assignada a diversos alumnes que estan a cursos i assignatures diferents i que estiguin imposades per diferents professors,

També veiem que Sancionar es una interrelació entre Amonestar i Sancions i no es una entitat, també veiem que te dos atributs que són resolució i motiu. La cardinalitat en aquest cas es M-N perquè per a una Sanció poden haver varies amonestacions i per a una amonestació es podrien aplicar varies sancions en funció de la gravetat d'aquesta.

Pel que fa a les claus primàries, només professor amb el nif, Amonestació amb tipus, tenen un atribut amb unicitat prou sòlida com per ser utilitzat com a clau primària. Per les altres entitats definirem claus primàries numèriques.

3.2 Disseny lògic.

En aquesta etapa haurem d'adaptar el disseny conceptual anterior al SGBD que utilitzarem, Oracle, un sistema de bases de dades relacional. Es tractarà doncs de fer la transformació del model Entitat Interrelació a un model relacional.

3.2.1 Model relacional.

1. Les entitats es transformen en relacions:

INSTITUT(codi_institut, descripció)

ALUMNE(num_expedient, codi_institut, nom_alumne, adreça, e-mail, telefon, any_curs)

PROFESSOR(nif_professor, nom_professor, codi_institut)

CURS(codi_curs, nom_curs, tutor, professor, codi_institut)

ASSIGNATURA(codi_assignatura, nom_assignatura, codi_curs, nif_professor)

AMONESTACIONS(tipus, descripcio, codi_institut)

SANCIONS(ID, sanció, codi_institut)

DATA_HORA_CALENDARI(data_hora)

DATA_HORA_CONSULTA(data_hora)

2. Les interrelacions binàries 1:1 i 1:N es transformen en claus foranes:

INSTITUT(codi_institut,descripció)

ALUMNE (Codi_institut)

On { Codi_institut } referencia INSTITUT

CURS (Codi_institut)

On { Codi_institut } referència INSTITUT

PROFESSOR(Codi_institut)

On { Codi_institut } referència INSTITUT

AMONESTACIONS(Codi_institut)

On { Codi_institut } referència INSTITUT

SANCIONS(Codi_institut)

On { Codi_institut } referència INSTITUT

DATA_HORA_CALENDARI(data_hora)

DATA_HORA_CONSULTA(data_hora)

3. Les interrelacions binàries N:M es transformen en relacions:

PERTANY (nif_professor,codi_curs)

On { nif_professor } referencia PROFESSOR { nif_professor }
i { codi_curs } referència CURS {nif}

ASSOCIAT (nif_professor,data_hora)

On { nif_professor } referència PROFESSOR {cif}
i { data_hora } referència DATA_HORA_CONSULTA

SANCIONAR (num_expedient, tipus_amonestacio,sancio)

On {num_expedient} referència AMONESTAR{num_expedient}

On { tipus_amonestacio } referència AMONESTAR{tipus_amonestacio}

i {sancio} referència SANCIONS{sancio}

REGLES_ACTIVACIO (codi_amonestacio,tipus_sancio)

On {codi_amonestacio} referència AMONESTACIONS{ tipus_amonestacio}

On {tipus_sancio} referència SANCIONS{sancio}

4. Les interrelacions n-àries, $n > 2$, es transformen en relacions:

AMONESTAR(num_expedient, codi_curs, codi_assignatura, tipus_amonestacio,
tutor,data_hora,gravetat,avisat)
 On {num_expedient} referència ALUMNE{num_expedient}
 i {codi_curs} referència CURS{codi_curs}
 i {tutor} referència CURS
 i {codi_assignatura} referència ASSIGNATURA{codi_assignatura}
 i {tipus_amonestacio} referència AMONESTACIONS{tipus_amonestacio}

El resultat final és, doncs:

AMONESTAR(num_expedient, codi_curs, codi_assignatura, tipus_amonestacio,
tutor,data_hora,gravetat,avisat)
 On {num_expedient} referència ALUMNE{num_expedient}
 i {codi_curs} referència CURS{codi_curs}
 i {tutor} referència CURS
 i {codi_assignatura} referència ASSIGNATURA{codi_assignatura}
 i {tipus_amonestacio} referència AMONESTACIONS{tipus_amonestacio}

PERTANY (nif_professor,nif)
 On { nif_professor } referència CURS { nif_professor }
 i { nif } referència PROFESSOR {nif}

ASSOCIAT (nif,data_hora)
 On { nif } referència PROFESSOR {cif}
 i { data_hora } referència DATA_HORA_CONSULTA
 INSTITUT(codi_institut,descripció)

SANCIONAR (num_expedient, tipus_amonestacio,sancio)
 On {num_expedient} referència AMONESTAR{num_expedient}
 On { tipus_amonestacio} referència AMONESTAR{tipus_amonestacio}
 i {sancio} referència SANCIONS{sancio}

REGLES_ACTIVACIO (codi_amonestacio,tipus_sancio)
 On {codi_amonestacio} referència AMONESTACIONS{ tipus_amonestacio}
 On {tipus_sancio} referència SANCIONS{sancio}

ALUMNE (Codi_institut)
 On { Codi_institut} referència INSTITUT

CURS (Codi_institut)

On { Codi_institut} referència INSTITUT

PROFESSOR(Codi_institut)

On { Codi_institut} referència INSTITUT

AMONESTACIONS(Codi_institut)

On { Codi_institut} referència INSTITUT

SANCIONS(Codi_institut)

On { Codi_institut} referència INSTITUT

DATA_HORA_CALENDARI(data_hora)

DATA_HORA_CONSULTA(data_hora)

3.2.2 Restriccions d'integritat:

-Restriccions d'integritat d'entitat: les definides per les claus primàries de l'apartat anterior.

-Restriccions d'integritat referencial: les definides per les claus foranes de l'apartat anterior.

-Cap atribut dels llistats en l'apartat anterior pot prendre valor NULL.

-Claus alternatives:

INSTITUT(descripció)

ALUMNE(nom_alummne, adreça, e-mail, telefon,any_curs)

PROFESSOR(nom_professor, codi_institut)

CURS(nom_curs, tutor, professor, codi_institut)

ASSIGNATURA(nom_assignatura, codi_curs, nif_professor)

AMONESTACIONS(descripcio)

SANCIONS(sanció)

-Restriccions d'usuari:

1) Un ALUMNE no pot pertànyer alhora a dos instituts ni estar matriculat a dos cursos diferents.

2) Un CURS no pot pertànyer a dos instituts.

- 3) Un PROFESSOR no pot pertànyer alhora a dos instituts.
- 4) Un CURS no pot tenir dos tutors.
- 5) Una sèrie concreta d'AMONESTACIONS només poden estar en un institut (són les que han decidit els professors del centre en qüestió)
- 6) Les SANCIONS també son definides per cada centre per tant només poden estar en un institut.

3.3 Disseny físic.

En aquest nivell s'han de fer consideracions orientades a optimitzar el rendiment tenint en compte aspectes de la implementació física del sistema de base de dades concret que utilitzarem. No podem perdre de vista que la realitat ens col·loca utilitzant una versió limitada, personal, del gestor de bases de dades en un maquinari limitat, lluny de les condicions en que treballaria en el món real un producte com el demanat. Tot i amb això, sí que procedirem a intentar optimitzar els recursos disponibles en dos sentits: la organització d'Espais Virtuals (en Oracle, tablespaces) i la utilització d'índexs per optimitzar el rendiment de les consultes.

4. Implementació.

Fora inadequat, per redundat i carregós, d'incloure aquí tot el codi de preparació dels elements lògics de la base de dades. Ens limitarem a donar exemples significatius i a explicar el procediment tots els codis aniran amb les carpetes adjuntes al document.

Tot i que la majoria de la feina s'acabarà realitzant escrivint manualment codi PL/SQL, també tenim a la nostra disposició l'eina web, força simple d'utilitzar, que l'SGBD Oracle 10g Express Edition ens proporciona per a la gestió de la base de dades:


```
ALTER TRIGGER "BI_INSTITUT" ENABLE
/
```

Veiem aquí que el que estem fent és definir una clau primària autonumèrica i una clau alternativa fent que la descripció sigui única.

4.2 Disparadors.

Com ja hem dit tenim una sèrie de disparadors requerits per les claus primàries autonumèriques.

- 1) Un ALUMNE no pot pertànyer alhora a dos instituts ni estar matriculat a dos cursos diferents.
- 2) Un CURS no pot pertànyer a dos instituts i Un CURS no pot tenir dos tutors
- 3) Un PROFESSOR no pot pertànyer alhora a dos instituts.
- 4) Una sèrie concreta d'AMONESTACIONS només poden estar en un institut (són les que han decidit els professors del centre en qüestió)
- 5) Les SANCIONS també son definides per cada centre per tant només poden estar en un institut.

Les altres cinc requereixen de la utilització de disparadors:

1)

Per implementar-la hem de modificar el disparador que va a buscar la clau primària en inserir una nova subhasta:

```
create or replace trigger "bi_alumne"
  before insert on "alumne"
  for each row
declare quants_instituts number;
declare quants_cursos number;
begin
select count(codi_institut) into quants_instituts from "alumnes"
where "alumnes"."num_expedient"= :new."num_expedient";
select count(curs) into quants_cursos from "alumnes"
where "alumnes"."num_expedient"= :new."num_expedient";

if(quants_instituts=1) then
  raise_application_error(-20001,'un alumne només pot estar matriculat a un
institut');
end if;

if(quants_cursos=1) then
  raise_application_error(-20002,'un alumne només pot estar matriculat a un curs
alhora');
```

```
end if;
```

```
if inserting then
```

```
    select "alumne_seq".nextval into :new.num_expedient from dual;
```

```
end if;
```

```
end;
```

2) Definim disparador:

```
create or replace trigger "bi_curs"
```

```
    before insert on "curs"
```

```
    for each row
```

```
declare quants_instituts number;
```

```
declare quants_tutors number;
```

```
begin
```

```
select count(codi_institut) into quants_instituts from "curs"
```

```
where "curs"."codi_curs"= :new."codi_curs";
```

```
select count(tutor) into quants_tutors from "curs"
```

```
where "curs"."tutor"= :new."tutor";
```

```
if(quants_instituts=1) then
```

```
    raise_application_error(-20003,'un curs només es pot fer en un institut');
```

```
end if;
```

```
if(quants_tutors=1) then
```

```
    raise_application_error(-20004,'un curs només pot tenir un tutor assignat');
```

```
end if;
```

```
if inserting then
```

```
    select "curs_seq1".nextval into :new.codi_curs from dual;
```

```
end if;
```

```
end;
```

3) Definim disparador:

```
create or replace trigger "professor_t1"
```

```
    before
```

```
insert on "professor"
```

```
for each row
```

```
begin
```

```
declare quants_instituts number;
```

```
select count(codi_institut) into quants_instituts from "professor"
```

```
where "professor"."nif_professor"= :new."nif_professor";
```

```
if(quants_instituts=1) then
```

```
raise_application_error(-20005,'un professor només pot donar classes a un
institut');
end if;
end;
```

4) Definim disparador:

```
create or replace trigger "amonestacions_t1"
before
insert on "amonestacions"
for each row
begin
declare quants_instituts number;
select count(codi_institut) into quants_instituts from "amonestacions"
where "amonestacions"."tipus_amonestacio"= :new."tipus_amonestacio";

if(quants_instituts=1) then
raise_application_error(-20007,'aquesta amonestació només està definida per
un institut en concret');
end if;
end;
```

5) definim disparador:

```
create or replace trigger "sancions_t1"
before
insert on "sancions"
for each row
begin
declare quants_instituts number;
select count(codi_institut) into quants_instituts from "sancions"
where "sancions"."tipus_amonestacio"= :new."tipus_amonestacio";

if(quants_instituts=1) then
raise_application_error(-20006,'aquesta sanció només està definida per un
institut en concret');
end if;
end;
```

Es pot veure en els disparadors anteriors que hem utilitzat una sèrie d'errors definits per l'usuari. Són els següents:

CODI	Missatge
-20001	'un alumne només pot estar matriculat a un institut'
-20002	'un alumne només pot estar matriculat a un curs alhora'
-20003	'un curs només es pot fer en un institut'
-20004	'un curs només pot tenir un tutor assignat'
-20005	'un professor només pot donar classes a un institut'
-20006	'aquesta sanció només està definida per un institut en concret'
-20007	'aquesta amonestació només està definida per un institut en concret'
-20008	Aquest PROFESOR només dona classes a un institut en concret
-20009	Aquest professor ja ha estat creat
-20010	'No existeix el registre que es demana actualitzar'
-20011	No existeix la amonestació que es vol fer servir
-20012	No existeix la Sancio que es vol fer servir
-20013	Aquest alumne no te sancions
-20014	Aquesta amonestació ja està creada
-20015	Aquesta sancio ja està creada

4.3 Índexs.

Després de repassar el contingut dels procediments, podem comprovar que tots els índexs necessaris per a optimitzar les consultes ja s'havien generat automàticament a partir de la definició de les claus primàries i alternatives.

4.4 Procediments

L'especificació del problema ens diu que tot accés de l'usuari a la base de dades es farà a través de procediments emmagatzemats, que haurem de proporcionar. Són els següents:

Alta Alumne
Baixa Alumne
Modificacio Alumne
Alta Amonestacions
Baixa Amonestacions
Modificacio Amonestacions
Alta Amonestar
Baixa Amonestar
Modificacio Amonestar
Alta Assignatura
Baixa Assignatura
Modificacio Assignatura
Alta Calendari
Baixa Calendari
Modificacio Calendari
Alta Curs
Baixa Curs
Modificacio Curs
Alta Professor
Baixa Professor
Modificacio Professor
Alta ReglesActivacio
Baixa ReglesActivacio
Modificacio ReglesActivacio
Alta Sancionar
Baixa Sancionar
Modificacio Sancionar
Alta Sancions
Baixa Sancions
Modificacio Sancions
Consulta_amonestacions
Consulta_amonestacions_i_sancions
Consulta_amonestacions_i_sancions_alumne
Consulta_amonestacions_i_sancions_disponible

Respecte del funcionament dels procediments, s'han de satisfer dos requeriments: Un, que hi hagi un paràmetre de sortida, RSP, que ens retorni la situació d'execució satisfactòria o el missatge d'error, i dos, que es tractin adequadament les excepcions. Ja hem vist en l'apartat anterior com hem utilitzats errors definits per l'usuari per impedir la violació de les regles de negoci.

El tractament que fem de l'excepció al procediment emmagatzemat que en intentar executar alguna modificació prohibida fa actuar el disparador, recollirà la informació de l'error com si en fos un dels habituals d'Oracle.

Hi ha un tercer requeriment, i és que tota crida als procediments quedi registrada en una taula log. A continuació es mostra un exemple senzill de com s'ha respost a aquestes necessitats:

```

create or replace procedure "ALTA_PROFESSOR"
(nif IN VARCHAR2,
nom IN VARCHAR2,
codi IN NUMBER,
data IN DATE,
poa IN VARCHAR2,
rsp OUT VARCHAR2)
is
begin
begin
insert into "PROFESSOR"
("NIF_PROFESSOR","NOM_PROFESSOR","CODI_INSTITUT")
VALUES (nif,nom,codi);
if NOT(nif is null) then
    insert into "ASSOCIAT" ("NIF_PROFESSOR","DATA_HORA",'POA')
    VALUES (nif,data,poa);
end if;
rsp:='OK';
EXCEPTION
    WHEN OTHERS THEN
        rsp:=SQLCODE||'-'||SQLERRM;
end;

insert into "LOG" ("procediment",
"paràmetres_entrada","paràmetres_sortida","data")
VALUES('ALTA_PROFESSOR','NIF_PROFESSOR='||nif||',NOM_PROFESSOR='||
nom||',CODI_INSTITUT='||codi||',DATA='||data||',POA='||poa,'rsp='||rsp,sysdate);
end;

```

4.5 Casos d'us

4.5.1. Alta De Alumne

Funcionalitat

Aquest procediment permet crear un nou alumne en la base de dades.

PreCondicions

L'alumne no està emmagatzemat a la base de dades.

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un insert a la taula Alumne.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

Nom_alumne **IN VARCHAR2**,
Adreça **IN VARCHAR2**,
E-mail **IN VARCHAR2**,
Telefon **IN VARCHAR2**,
Codi_institut **IN NUMBER**,
Curs **IN NUMBER**,
Any_curs **IN NUMBER**,
rsp **OUT VARCHAR2** pot prendre valor ok o error si genera excepció.

Postcondicions

L'alumne està emmagatzemat a la base de dades

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_ALUMNES.

4.5.2 Baixa d' Alumne

Funcionalitat

Aquest procediment permet donar de baixa un alumne en la base de dades.

Precondicions

L'alumne esta gravat a la base de dades.

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetre d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
2. Si totes les dades anteriors son correctes fem un delete de la taula Alumne.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

expedient **IN NUMBER**,
rsp **OUT VARCHAR2** pot prendre valor ok o error si genera excepció.

Postcondicions

L'alumne ja no es troba a la base de dades

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_ALUMNES.

4.5.3 Modificar Alumne

Funcionalitat

Aquest procediment permet canviar les dades de un alumne en la base de dades.

Precondicions

L'alumne esta gravat a la base de dades.

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetre d'entrada.
s'han d'introduir els valors per a fer els canvis en els Camps d'entrada.
2. Si totes les dades anteriors son correctes fem un update de la taula Alumne.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG

4. En el cas de llençar una excepció `rsp` serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

expedient IN NUMBER,
nom_nou IN VARCHAR2,
adreça_nou IN VARCHAR2,
email_nou IN VARCHAR2,
telefon_nou IN VARCHAR2
codi_nou IN NUMBER,
curs_nou IN NUMBER,
any_nou IN NUMBER
`rsp` OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

Les dades del alumne estan canviades a la base de dades.

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_ALUMNES.

4.5.4 Alta Tipus D'amonestacions

Funcionalitat

Aquest procediment permet crear una nova amonestació a la base de dades.

Precondicions

L'amonestació no està donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un insert a la taula Amonestacions.
3. El paràmetre de sortida `rsp` serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció `rsp` serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

tipus **IN VARCHAR2**,
codi_institut **IN NUMBER**,
descripcio **IN VARCHAR2**,
rsp **OUT VARCHAR2 pot prendre valor ok o error si genera excepció**

Postcondicions

L'amonestació esta gravada a la base de dades.

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Amonestacions

4.5.5. Baixa tipus d'amonestacions

Funcionalitat

Aquest procediment permet eliminar una amonestació de la base de dades.

Precondicions

L'amonestació esta gravada a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula Amonestacions.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

tipus **IN VARCHAR2**,
rsp **OUT VARCHAR2 pot prendre valor ok o error si genera excepció**

Postcondicions

L'amonestació està donada de baixa de la base de dades

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Amonestacions

4.5.6 Modificació tipus d'amonestacions

Funcionalitat

Aquest procediment permet modificar les dades a una amonestació de la base de dades.

Precondicions

L'amonestació està gravada a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula Amonestacions.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

tipus **IN VARCHAR2**,
desc_vella **IN VARCHAR2**,
desc_nou **IN VARCHAR2**,
rsp **OUT VARCHAR2** pot prendre valor ok o error si genera excepció.

Postcondicions

Les dades estan canviades a la base de dades

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Amonestacions

4.5.7 Alta D'amonestacions**Funcionalitat**

Aquest procediment permet inserir a la base de dades una amonestació per a un alumne

Precondicions

L'alumne i l'amonestació han d'estar gravats en la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Insert a la taula Amonestar.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
- 5.- En aquest procés hi ha un disparador que verifica que no s'hagi arribat al nombre d'amonestacions necessaris per executar una sancio. En cas de arribar a aquest nombre llença una sanció i la emmagatzema a la taula Sancionar.
- 6.- Al mateix temps a la taula amonestar els camps Sancionat que abans eren no ara passen a ser si per evitar que es produeixin noves amonestacions.
- 7.- Aquest procediment a més és el que omple tota la taula Estadística que fa referència a les dades d'amonestacions.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

numero_expedient IN NUMBER,
tipus_amonestacio IN VARCHAR2,
tutor_sanc IN VARCHAR2,
curs IN NUMBER,
codi_assignatura IN NUMBER,
data_hora IN DATE,
gravetat IN NUMBER,
avisat IN VARCHAR2,
sancionat IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

Es grava a un alumne una amonestació

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Amonestar

4.5.8 Baixa d'amonestacions

Funcionalitat

Aquest procediment permet eliminar de la base de dades una amonestació per a un alumne

Precondicions

L'alumne ha de tenir gravada una amonestació

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula Amonestar.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
- 5.- En aquest procés tenim també tot el codi que actualitza les dades per a realitzar l'estadística.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

numero_expedient IN NUMBER,
tipus_amonest IN VARCHAR2,
tutor_sanc IN VARCHAR2,
curs IN NUMBER,
assignatura IN NUMBER,
data IN DATE,
gravetat IN NUMBER,
avisat IN VARCHAR2,
sancionat IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

Es dona de baixa de la base de dades l'amonestació per l'alumne

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Amonestar

4.5.9 Modificació d'amonestacions

Funcionalitat

Aquest procediment permet canviar les dades de les amonestacions dels alumnes

Precondicions

L'amonestació ha d'existir a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula Amonestar.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
- 5.- En aquest procés tenim també tot el codi que actualitza les dades per a realitzar l'estadística.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

numero_expedient IN NUMBER,
 tipus_amonest IN VARCHAR2,
 tutor_sanc IN VARCHAR2,
 curs IN NUMBER,
 assignatura IN NUMBER,
 data IN DATE,
 gravetat IN NUMBER,
 avisat IN VARCHAR2,
 sancionat IN VARCHAR2,
 rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

Les dades modificades es graben en la base de dades.

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Amonestar

4.5.10 Alta D'assignatures

Funcionalitat

Aquest procediment permet inserir a la base de dades una assignatura

Precondicions

L'assignatura no esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.

2. Si totes les dades anteriors son correctes fem un Insert a la taula Assignatura.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

Nom_assignatura **IN VARCHAR2**,
nif_professor **IN VARCHAR2**,
codi_curs **IN NUMBER**,
rsp **OUT VARCHAR2 pot prendre valor ok o error si genera excepció**

Postcondicions

L'assignatura es troba emmagatzemada en la base de dades.

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Assignatura

4.5.11 Baixa d'assignatures

Funcionalitat

Aquest procediment permet eliminar de la base de dades una assignatura

Precondicions

L'assignatura esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula assignatura.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
- 5.- En aquest cas per la baixa només necessitem el codi autocalculat de l'assignatura ja que és únic.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

assignatura **IN NUMBER**,
rsp **OUT VARCHAR2 pot prendre valor ok o error si genera excepció**

Postcondicions

Es dona de baixa de la base de dades l'assignatura

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Assignatura

4.5.12 Modificació d'assignatures**Funcionalitat**

Aquest procediment permet canviar les dades de les assignatures

Precondicions

L'assignatura esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula assignatura.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

assignatura **IN NUMBER**,
nom_nou **IN VARCHAR2**,
nif_professor_nou **IN VARCHAR2**,
codi_curs_nou **IN NUMBER**,
rsp **OUT VARCHAR2 pot prendre valor ok o error si genera excepció**

Postcondicions

S'emmagatzema a la base de dades els canvis

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_assignatura

4.5.13 Alta de calendari

Funcionalitat

Aquest procediment permet inserir a la base de dades una calendari per a saber quins dies i per quines assignatures es festiu.

Precondicions

El calendari no esta donat d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Insert a la taula Calendari.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

codi_curs IN NUMBER,
codi_assignatura IN NUMBER,
data_hora IN DATE ,
festiu IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els mou calendari

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Calendar

4.5.14 Baixa de calendaris**Funcionalitat**

Aquest procediment permet eliminar de la base de dades un calendari

Precondicions

El Calendari esta donat d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula Calendari.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

data IN DATE ,
curs IN NUMBER,
assignatura IN NUMBER,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

Es borra de la base de dades el calendari

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_calendari

4.5.15 Modificació de calendaris

Funcionalitat

Aquest procediment permet canviar les dades dels calendaris

Precondicions

El Calendari esta donat d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula Calendari.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

data IN DATE,

data_nou IN DATE,
codi_curs_vell IN NUMBER,
codi_curs_nou IN NUMBER,
codi_assignatura_vell IN NUMBER,
codi_assignatura_nou IN NUMBER,
festiu_nou IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els canvis

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_calendari

4.5.16 Alta de curs

Funcionalitat

Aquest procediment permet inserir a la base de dades una curs nou

Precondicions

El curs no esta donat d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Insert a la taula Curs.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

nom_curs IN VARCHAR2,
tutor IN VARCHAR2,
codi IN VARCHAR2,
professor IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els nou curs

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_curs

4.5.17 Baixa de curs**Funcionalitat**

Aquest procediment permet eliminar de la base de dades un curs

Precondicions

El Calendari no esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula Curs.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

curs IN VARCHAR2,
institut IN NUMBER,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

Es dona de baixa a la base de dades els curss

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_curs

4.5.18 Modificació de curs**Funcionalitat**

Aquest procediment permet canviar les dades dels cursos

Precondicions

EL calendari no esta donat d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula curs
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

nom_nou IN VARCHAR2,
nom_vell IN VARCHAR2,
tutor_nou IN VARCHAR2,
tutor_vell IN VARCHAR2,,
codi_institut_nou IN NUMBER,
codi_institut_vell IN NUMBER,
professor_nou IN VARCHAR2,
professor_vell IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els canvis

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_curs

4.5.19 Alta de professor

Funcionalitat

Aquest procediment permet inserir a la base de dades un nou professor

Precondicions

EL professor no esta donat d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Insert a la taula professor.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG

4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

(nif **IN VARCHAR2**,
nom **IN VARCHAR2**,
codi **IN NUMBER**,
data **IN DATE** ,
poa **IN VARCHAR2**, en aquest cas poa són les inicials de (p)are o (a)lumne
rsp **OUT VARCHAR2 pot prendre valor ok o error si genera excepció**

Postcondicions

Es dona d'alta a la base de dades el nou professor

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_professor

4.5.20 Baixa de professors

Funcionalitat

Aquest procediment permet eliminar de la base de dades un professor

Precondicions

El professor esta donat d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula professor.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
5. En aquest cas només ens cal un paràmetre d'entrada per eliminar un professor ja que el camp nif és únic.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

nif **IN NUMBER**,

rsp **OUT VARCHAR2** pot prendre valor ok o error si genera excepció

Postcondicions

Es dona de baixa a la base de dades el professor

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_professor.

4.5.21 Modificació de professors.

Funcionalitat

Aquest procediment permet canviar les dades dels cursos

Precondicions

El professor esta donat d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula professor
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

(nif **IN VARCHAR2**,

nif_nou **IN VARCHAR2**,

data_nou **IN DATE**,

data_vell **IN DATE**,

poa_nou **IN VARCHAR2**,

poa_vell **IN VARCHAR2**,

nom_nou **IN VARCHAR2**,

nom_vell **IN VARCHAR2**,

rsp **OUT VARCHAR2** pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els canvis

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_professor.

4.5.22 Alta de Regles d'activació

Funcionalitat

Aquest procediment permet inserir a la base de dades les regles automàtiques de llançament de sancions, es a dir és on direm quan i com s'han executar les sancions.

Precondicions

La regla no esta donada d'alta a la base de dades i estan definides abans les sancions i amonestacions

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Insert a la taula regles_activacio.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

tipus **IN VARCHAR2**, aquí especifiquem el tipus d'amonestació
operador **IN VARCHAR2**, aquí definim l'operador de l'amonestació **</>=**
numero **IN NUMBER**, el valor en el que s'ha de llençar per exemple 5
id_sancio **IN VARCHAR2**, la sanció que s'aplicarà
rsp **OUT VARCHAR2 pot prendre valor ok o error si genera excepció**

Postcondicions

S'emmagatzema a la base de dades la nova regla

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_regles d'activació

4.5.23 Baixa de Regles d'activació

Funcionalitat

Aquest procediment permet eliminar de la base de dades una regla d'activació

Precondicions

La regla esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula regles_activacio.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especificuem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

(tipus **IN VARCHAR2**,
sancio **IN VARCHAR2**,
rsp **OUT VARCHAR2 pot prendre valor ok o error si genera excepció**)

Postcondicions

Es dona de baixa de la base de dades la regla.

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_ regles d'activació.

4.5.24 Modificació de regles d'activació.

Funcionalitat

Aquest procediment permet canviar les dades de les regles d'activació.

Precondicions

La regla esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.

- s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula regles_activacio
 3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
 4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

tipus **IN VARCHAR2**,
tipus_vell **IN VARCHAR2**,
id_sancio_vell **IN VARCHAR2**,
id_sancio **IN VARCHAR2**,
operador_nou **IN VARCHAR2**,
operador_vell **IN VARCHAR2**,
numero_nou **IN VARCHAR2**,
numero_vell **IN VARCHAR2**,
rsp **OUT VARCHAR2** pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els canvis

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_regles d'activació.

4.5.25 Alta de sancions

Funcionalitat

Aquest procediment permet inserir a la base de dades una sanció per a un alumne i afegir les dades per a fer l'estadística

Precondicions

La sancio per l'alumne no esta donada d'alta a la base de dades i estan definits sancions, alumne i amonestacions

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Insert a la taula Amonestar.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG

4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
- 5.- Aquest procediment a més és el que omple tota la taula Estadística que fa referència a les dades de les sancions.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

numero_expedient IN NUMBER,
tipus_amonest IN VARCHAR2,
sancio IN VARCHAR2,
data IN DATE,
motiu IN VARCHAR2,
resolucio IN VARCHAR2,
curs IN NUMBER,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades la nova sancio.

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_sancionar

4.5.26 Baixa de sancions

Funcionalitat

Aquest procediment permet eliminar de la base de dades una sancio per a un alumne

Precondicions

La sancio esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula Sancionar.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
- 5.- En aquest procés tenim també tot el codi que actualitza les dades de les sancions per a realitzar l'estadística.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

numero_expedient IN NUMBER,
tipus_amonest IN VARCHAR2,
sancio IN VARCHAR2,
data IN DATE,
motiu IN VARCHAR2,
resolucio IN VARCHAR2,
curs IN NUMBER,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

Es dona de baixa a la base de dades la sancio

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Sancionar

4.5.27 Modificació de sancions**Funcionalitat**

Aquest procediment permet canviar les dades de les sancions dels alumnes

Precondicions

La sancio esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula Amonestar.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
- 5.- En aquest procés tenim també tot el codi que actualitza les dades per a realitzar l'estadística.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

(num_expedient_nou IN NUMBER,
num_expedient_vell IN NUMBER,

tipus_amonestacio_nou IN NUMBER,
tipus_amonestacio_vell IN NUMBER,
id_sancio_nou IN NUMBER,
id_sancio_vell IN NUMBER,
data_sancio_nou IN DATE,
data_sancio_vell IN DATE,
motiu_nou IN NUMBER,
motiu_vell IN NUMBER,
resolucio_nou IN NUMBER,
resolucio_vell IN NUMBER,
curs_nou IN NUMBER,
curs_vell IN NUMBER,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els canvis

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_Sancionar

4.5.28 Alta de tipus de sancions

Funcionalitat

Aquest procediment permet inserir a la base de dades els diferents tipus de sancions que hi ha.

Precondicions

El tipus de sancio no esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Insert a la taula sancions.
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

id IN VARCHAR2,
codi_institut IN NUMBER,

sancio IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els Motius de sancio

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_sancions

4.5.29 Baixa de tipus de sancions

Funcionalitat

Aquest procediment permet eliminar de la base de dades un tipus de sancio

Precondicions

El tipus de sancio esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Delete a la taula sancions
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.
5. En aquest cas només ens cal un paràmetre d'entrada per eliminar un tipus de sancio ja que el camp id és únic.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

id IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

Es dona de baixa de la base de dades el tipus de sancio

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_sancions.

4.5.30 Modificació de tipus de sancions.

Funcionalitat

Aquest procediment permet canviar les dades dels tipus de sancions

Precondicions

El tipus de sancio esta donada d'alta a la base de dades

Explicació

El procés que seguirà el procediment serà el següent:

1. comprovem el paràmetres d'entrada.
s'han d'introduir tots els valors per els Camps d'entrada.
en el cas de que no estiguin bé els paràmetres llencem una excepció.
2. Si totes les dades anteriors son correctes fem un Update a la taula sancions
3. El paràmetre de sortida rsp serà igual a OK i farem un insert a LOG
4. En el cas de llençar una excepció rsp serà igual a un missatge on especifiquem l'error i farem insert a LOG.

Paràmetres

Els paràmetres d'entrada i sortida seran els següents:

id IN VARCHAR2,
sancio_vella IN VARCHAR2,
sancio_nou IN VARCHAR2,
rsp OUT VARCHAR2 pot prendre valor ok o error si genera excepció

Postcondicions

S'emmagatzema a la base de dades els canvis

Codi

El codi d'aquest procediment el podem trobar a la carpeta procediments i el fitxer es el ABM_sancions.

En molts dels procediments anteriors veurem que els camps clau no ens permet modificar-los per tractar-se precisament de camps clau. Es preferible borrar i tornar a crear-los.

4.6 Mòdul estadístic.

Com s'ha vist en l'apartat de requeriments del projecte, un dels més destacats és la implementació d'un mòdul estadístic que retorni, a cost 1, tota una sèrie de resultats relatius a informació acumulada en la base de dades.

Per obtenir aquests resultats es proveirà al client de 8 procediments:

```
ESTADISTICA_ALUMNE_MES_SANCIONAT_ANY
ESTADISTICA_ALUMNES_NO_AMONESTATS_PER_ANY
ESTADISTICA_AMONESTACIONS_PER_ALUMNE
ESTADISTICA_MITJANA_AMONESTACIONS_PER_PROFESSOR_I_ANY
ESTADISTICA_MITJANA_SANCIONS_PER_CURS_I_ANY
ESTADISTICA_PROFESSOR_MES_SANCIONADOR_DATA_CURS
ESTADISTICA_SANCIONS_PER_ALUMNE_I_ANY
ESTADISTICA_SANCIONS_PER_CURS_I_ANY
```

que consultaran la taula ESTADISTICA per a obtenir la informació de manera directa.

Per alimentar aquesta taula, cada vegada que un del procediments emmagatzemats que atenen les funcionalitats de la base de dades faci un canvi que afecti a l'estadístic acumulat, n'ha d'actualitzar les taules.

Per il·lustrar el procés mostrarem un dels exemples més senzills, referits a l'estadístic anual:

-Creació de la taula:

```
CREATE TABLE "ESTADISTICA"
(
  "NUM_AMONEST" NUMBER,
  "NUM_EXP" NUMBER,
  "PROF_MITJANA_ANUAL" NUMBER,
  "PROFESSOR" NUMBER,
  "PROFESSOR_MES_SANCIONADOR" NUMBER,
  "DATA" NUMBER,
  "CURS" NUMBER,
  "ALUMNES_NO_AMO" NUMBER,
  "NUM_SANCIONS_ALUMNE" NUMBER,
  "NUM_SANCIONS_ANY" NUMBER,
  "ALUMNE_MES_SANCIONAT" NUMBER,
  "MITJANA_SANCIONS_CURS" NUMBER
)
/
```

Posem com exemple la següent consulta:

```

create or replace procedure "EST_ALUMNE_MES_SANCIONAT"
(data IN NUMBER,
alumne_mes_sancionat IN NUMBER DEFAULT NULL,
sancions OUT SYS_REFCURSOR,
rsp OUT VARCHAR2)
is
BEGIN
BEGIN

    open sancions FOR
    select "ALUMNE_MES_SANCIONAT"
    from "ESTADISTICA"
    where "DATA"=data;

    rsp:='OK';
EXCEPTION
    WHEN OTHERS THEN
        rsp:=SQLCODE||'-'||SQLERRM;
end;
insert into "LOG" ("procediment", "paràmetres_entrada","paràmetres_sortida","data")
VALUES('ESTADISTICA_ALUMNE_MES_SANCIONAT_ANY','ALUMNE_MES_SANCIONAT='||alu
mne_mes_sancionat||',DATA='||data,'rsp='||rsp,sysdate);
END;

```

-Actualització de la taula (mostrem només un extracte amb les parts rellevants a l'estadístic mensual):

```

create or replace procedure "ALTA_AMONESTAR"
...
is
...

begin
begin
...

quin_any:=to_number(to_char(data_hora, 'yyyy'));

/* DADES CONSULTA 1*/
select ("NUM_EXPEDIENT")
    into expedient
    from "AMONESTAR"
    where "NUM_EXPEDIENT"=num_expedient;

```

IF (expedient is not null) THEN

```

        update "ESTADISTICA" set "NUM_AMONEST"="NUM_AMONEST"+1 where
"NUM_EXP"=expedient;
ELSE
        insert into "ESTADISTICA" ("NUM_EXP","NUM_AMONEST") values (expedient,1);
end if;

```

/*DADES CONSULTA 3*/

```

select count("TIPUS_AMONESTACIO")
into quantes_amonestacions
from "AMONESTAR"
where "TUTOR"=tutor;

```

```

select count(distinct (to_number(to_char("DATA_HORA", 'yyyy'))))
into quants_anys
from "AMONESTAR"
where "TUTOR"=tutor;

```

IF (quantes_amonestacions is not null) THEN

```

        update "ESTADISTICA" set
"PROF_MITJANA_ANUAL"=quantes_amonestacions/quants_anys where "PROFESSOR"=tutor;
ELSE

```

```

        insert into "ESTADISTICA" ("PROFESSOR","PROF_MITJANA_ANUAL") values
(tutor,quantes_amonestacions/quants_anys);
end if;

```

/*DADES CONSULTA 6*/

```

select "TUTOR"
into professor_mes
from "AMONESTAR" where to_number(to_char("DATA_HORA", 'yyyy'))=quin_any and
"CODI_CURS"=codi_curs group by "TUTOR" having count(*) = (select
max(count("TIPUS_AMONESTACIO")) from "AMONESTAR"
where to_number(to_char("DATA_HORA", 'yyyy'))=quin_any and "CODI_CURS"=codi_curs group
by "TUTOR");

```

IF (professor_mes is not null) THEN

```

        update "ESTADISTICA" set "PROFESSOR_MES_SANCIONADOR"=professor_mes where
"PROFESSOR_MES_SANCIONADOR" IS NOT NULL and "DATA"=quin_any and
"CODI_CURS"=codi_Curs;
ELSE

```

```

        insert into "ESTADISTICA" ("PROFESSOR_MES_SANCIONADOR","DATA","CURS")
values (professor_mes,quin_any,codi_curs);
end if;

```

/* DADES CONSULTA 8*/

```

select count("NUM_EXPEDIENT")
into alumnes_no_amo
from "ALUMNE"
WHERE NOT EXISTS ( SELECT "NUM_EXPEDIENT" FROM "AMONESTAR" WHERE
"ALUMNE"."NUM_EXPEDIENT" = "AMONESTAR"."NUM_EXPEDIENT" AND
"ANY_CURS"=quin_any);

```

IF (alumnes_no_amo is not null) THEN

TFC - BASES DE DADES RELACIONALS

Prova d'avaluació continuada PAC 2

```
update "ESTADISTICA" set "ALUMNES_NO_AMO"=alumnos_no_amo where
"ALUMNES_NO_AMO" IS NOT NULL and "DATA"=quin_any;
ELSE
insert into "ESTADISTICA" ("ALUMNES_NO_AMO","DATA") values
(alumnos_no_amo,quin_any);
end if;
```

EXCEPTION

```
WHEN OTHERS THEN
rsp:=SQLCODE||'-'||SQLERRM;
end;
```

```
insert into "LOG" ("procediment", "paràmetres_entrada","paràmetres_sortida","data")
VALUES('ALTA_AMONESTAR','NUM_EXPEDIENT='||num_expedient||',TIPUS_AMONESTACIO='|
|tipus_amonestacio||',TUTOR='||tutor||',CODI_CURS='||codi_curs||',CODI_ASSIGNATURA='||codi_
assignatura||',DATA_HORA='||data_hora||',GRAVETAT='||gravetat||',AVISAT='||avisat||',SANCIONA
T='||sancionat,rsp='||rsp,sysdate);
end;
```

5. Proves.

És requisit indispensable de tot projecte informàtic la confecció d'un joc de proves exhaustiu i complet que en garanteixi el bon funcionament, ben entès que en la pràctica mai no serà possible comprovar per endavant totes les possibilitats i combinacions d'ús. S'ha procurat, doncs comprovar totes les funcionalitats i la bona resposta del sistema davant de les restriccions de sistema i d'usuari plantejades.

Procediment principal: (proves_mestre)

/* per executar el joc de proves cal:

Anar al directori on tenim els procediments en concret el procediment Esborrar_sequences.sql i des d'aquest directori obrir sessió sql

```
connect HR /01061998
```

activar la resposta per pantalla:

```
set serveroutput on;
```

reiniciar les seqüències, executant

```
@esborrar_sequences
```

i

executar aquest procediment:

```
execute proves_mestre('');
```

/

*/

```
create or replace procedure PROVES_MESTRE (descripcio IN VARCHAR2)
```

```
is
```

```
resultat VARCHAR2(255);
```

```
resultset SYS_REFCURSOR;
```

```
number1 NUMBER;
```

```
number2 NUMBER;
```

```
number3 NUMBER;
```

```
number4 NUMBER;
```

```
number5 NUMBER;
```

```
number6 NUMBER;
```

```
cadena1 varchar2(255);
```

```
cadena2 varchar2(255);
```

```
cadena3 varchar2(255);
```

```
cadena4 varchar2(255);
```

```
cadena5 varchar2(255);
```

```
cadena6 varchar2(255);
```

```
date1 DATE;
```

```
date2 DATE;
```

```
date3 DATE;
```

```
begin
```


/* provarem totes les funcions de la base de dades mentre a la vegada l'omplim d'informació */

/* buidem totes les taules */

```
delete from "REGLES_ACTIVACIO";
delete from "SANCIONS";
delete from "AMONESTACIONS";
delete from "SANCIONAR";
delete from "ASSOCIAT";
delete from "CURS";
delete from "PROFESSOR";
delete from "PERTANY";
delete from "LOG";
delete from "ESTADISTICA";
delete from "CALENDARI";
delete from "ASSIGNATURA";
delete from "AMONESTAR";
delete from "ALUMNE";
delete from "INSTITUT";
```

/* no hi ha procediments ABM per als INSTITUTS*/

```
insert into "INSTITUT" ("DESCRIPCIO") VALUES ('ESCOLA PIA');
insert into "INSTITUT" ("DESCRIPCIO") VALUES ('MARISTES');
insert into "INSTITUT" ("DESCRIPCIO") VALUES ('SALESIANS');
```

```
PROVES_AMONESTACIONS("");
PROVES_SANCIONS("");
PROVES_PROFESSOR("");
PROVES_CURSOS("");
PROVES_REGLES_VALIDACIO("");
PROVES_ALUMNE("");
PROVES_ASSIGNATURA("");
PROVES_CALENDARI("");
PROVES_AMONESTAR("");
PROVES_SANCIONAR("");
PROVES_CONSULTES("");
PROVES_ESTADISTIQUES("");
end;
```

Mostrem també un dels procediments que proven parts concretes de la base de dades, novament utilitzant ALUMNE com a un dels més senzills:

```
create or replace procedure PROVES_ALUMNE (descripcio IN VARCHAR2)
is
resultat VARCHAR2(255);
resultset SYS_REFCURSOR;
number1 NUMBER;
number2 NUMBER;
number3 NUMBER;
number4 NUMBER;
number5 NUMBER;
```

```

number6 NUMBER;
cadena1 varchar2(255);
cadena2 varchar2(255);
cadena3 varchar2(255);
cadena4 varchar2(255);
cadena5 varchar2(255);
cadena6 varchar2(255);
date1 DATE;
date2 DATE;
date3 DATE;

begin
/* provarem totes les funcions de la base de dades mentre a la vegada l'omplim d'informació */

/* provem ABM ALUMNES */
    /* insertem valors correctes */

dbms_output.put_line('Comanda:          ||'|ALTA_ALUMNE('Enric          Garcia', 'casa
seva', 'aaa@aaa.com', '12343211',1,1,2011)');
"ALTA_ALUMNE"('Enric Garcia','casa seva','aaa@aaa.com','12343211',1,1,2011,resultat);
dbms_output.put_line('Resultat: ||resultat);

dbms_output.put_line('Comanda:          ||'|ALTA_ALUMNE('Pere          Marquez', 'pirineus
14', 'bbb@bbb.com', '937875421',1,1,2011)');
"ALTA_ALUMNE"('Pere Marquez','pirineus 14','bbb@bbb.com','937875421',1,1,2011,resultat);
dbms_output.put_line('Resultat: ||resultat);

dbms_output.put_line('Comanda:          ||'|ALTA_ALUMNE('Toni          Perez', 'Aneto
3', 'ccc@ccc.com', '937521478',1,1,2010)');
"ALTA_ALUMNE"('Toni Perez','Aneto 3','ccc@ccc.com','937521478',1,1,2010,resultat);
dbms_output.put_line('Resultat: ||resultat);

dbms_output.put_line('Comanda:          ||'|ALTA_ALUMNE('Cesar          Martí', 'Francesc
Layret', 'ddd@ddd.com', '936542147',1,1,2009)');
"ALTA_ALUMNE"('Cesar Martí','Francesc Layret','ddd@ddd.com','936542147',1,1,2009,resultat);
dbms_output.put_line('Resultat: ||resultat);

/*Provem afegir un alumne a un altre institut*/

dbms_output.put_line('Comanda:          ||'|ALTA_ALUMNE('Toni          Perez', 'Aneto
3', 'ccc@ccc.com', '937521478',2,1,2010)');
"ALTA_ALUMNE"('Toni Perez','Aneto 3','ccc@ccc.com','937521478',2,1,2010,resultat);
dbms_output.put_line('Resultat: ||resultat);

/* provem de posar un segon curs a un alumne en un mateix any */

dbms_output.put_line('Comanda:          ||'|ALTA_ALUMNE('Toni          Perez', 'Aneto
3', 'ccc@ccc.com', '937521478',1,1,2010)');
"ALTA_ALUMNE"('Toni Perez','Aneto 3','ccc@ccc.com','937521478',1,1,2010,resultat);
dbms_output.put_line('Resultat: ||resultat);

/*Donem de baixa un alumne */
dbms_output.put_line('Comanda: ||'|BAIXA_ALUMNE(2)');
"BAIXA_ALUMNE"(2,resultat);
dbms_output.put_line('Resultat: ||resultat);

```

```
/*MODIFIQUEM un alumne */
```

```
dbms_output.put_line('Comanda:      ||'MODIFICA_ALUMNE(3,'Toni      MERCADER', 'Aneto  
7', '111@ccc.com', '997521478',2,1,2011)');  
"MODIFICA_ALUMNE"(3,'Toni      MERCADER', 'Aneto  
7', '111@ccc.com', '997521478',2,1,2010,resultat);  
dbms_output.put_line('Resultat: ||resultat);  
  
end;
```

6 Valoració Econòmica

En aquest projecte no podem realitzar una valoració econòmica ja que està destinat a ser un projecte final de carrera.

Si es volgués fer una estimació final del cost ho podríem fer de dos maneres:

1.- Fixant un preu tancat decidit en el moment de contractar els serveis dels dissenyadors. Aquesta opció en principi és la millor per a la part que demana la feina perquè ja parteix amb un pressupost tancat i que sempre serà un preu estable.

2.- Fixant el preu segons les hores que es dediquin a fer el projecte, aquesta opció serà sempre la millor per el dissenyador ja que al necessitar moltes hores per als jocs de proves, tests i altres comprovacions es pot dispar en preu.

7 Conclusions

Com a conclusió d'aquest projecte podríem dir que per a mi ha estat una càrrega de feina molt important en aquest període concret de la meua vida. A la feina hem sofert retallades de personal i la feina que abans fèiem a 10 persones ara la fem 5 i això ha suposat una doble càrrega de treball. Amb continuats viatges que han suposat una pèrdua de temps per a la realització d'aquest projecte.

A banda d'això també podríem dir que aquest projecte ha anat molt bé per a refrescar tots els temes que anteriorment havíem estudiat a diverses assignatures i que gracies a això he pogut aplicar conceptes a la meua feina diària.

Un tema que m'ha agradat es poder treballar amb aquesta versió d'oracle de la qual no en tenia coneixement i que a partir d'ara la farà servir per a tots els temes que facin referència a bases de dades que tinc a la feina. Tot i que el que no trobo bé es que ens facin fer un projecte final de carrera amb un sistema de Gestió de Bases de Dades que ni tant sols hem fet servir durant tota la carrera, recordar que tot ho fèiem amb l'Informix.

Espero que aquest projecte signifiqui el final d'un procés d'estudi que vaig començar ara fa uns quants anys i que al principi va resultar molt dur per a mi. Mai havia estudiat per Internet, feia més de 7 anys que havia deixat d'estudiar i tornar a fer-ho m'ha costat molt. Però en definitiva dono per molt bo aquest procés i agraeixo a tots els implicat en ell mateix, tant alumnes com consultors l'esforç que realitzen per a poder dur endavant aquesta feina.

8. Bibliografia

<http://www.oracle.com/technology/documentation/index.html>

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

<http://www.webtaller.com/construccion/lenguajes/index/sql/>