

**Creació d'un sistema
d'informació per a la
gestió i ús de dades
d'interferometria.**

Autor:
Albert Calabria i Ferrer

Tutors:
Laura Sala i Martín (LIGIT)
Neus Querol i Vidal (IGC)
Jordi Marturià i Alavedra (IGC)

Febrer 2011

UAB

Universitat Autònoma de Barcelona
Departament de Geografia

12 mtig 2010

Professionals per a la Societat de la Informació



IGC
Institut Geològic
de Catalunya



Generalitat de Catalunya
**Departament de Política Territorial
i Obres Públiques**

CONTINGUTS

	<u>pàg</u>
1 Introducció.....	6
1.1 Presentació.....	6
1.2 Marc institucional.....	6
1.3 Marc teòric.....	6
2 Descripció del projecte.....	8
2.1 Situació i necessitats actuals.....	8
2.2 Usuaris.....	9
2.3 Definició dels objectius.....	9
2.4 Planificació inicial dels treballs.....	10
3 Requeriments.....	11
3.1 Requeriments funcionals.....	11
3.2 Requeriments de programari.....	11
4 Disseny i implementació de la BBDD.....	12
4.1 Requeriments i objectius.....	12
4.2 Aproximació tecnològica.....	12
4.2.1 Programari.....	12
4.2.2 Llenguatges de programació.....	13
4.3 Model conceptual.....	13
4.4 Model lògic.....	14
4.4.1 Aspectes formals del model lògic.....	15

4.4.2	Relacions entre entitats.....	16
4.4.3	Disparadors (Triggers).....	16
4.5	Implementació de la Base de Dades.....	17
4.6	Càrrega inicial de dades.....	19
5	Aplicació de càrrega de dades.....	23
5.1	Requeriments i objectius.....	23
5.2	Aproximació tecnològica.....	23
5.3	Programació de l'aplicació de càrrega de dades.....	23
5.3.1	Connexió a la Base de dades.....	24
5.3.2	Modificació de la taula satèl·lit.....	27
5.3.3	Càrrega de noves imatges.....	29
5.3.4	Càrrega d'una nova zona i selecció de la que en volem veure les dades.....	31
5.3.5	Càrrega de les dades a la taula punts.....	34
5.3.6	Exportació de les taules.....	36
5.4	Creació d'arxiu instal·lable.....	38
6	Conclusions.....	39
7	Referències.....	40
8	Annexes.....	41
8.1	Model lògic de la Base de dades.....	41
8.2	Codi d'implementació de la Base de dades.....	76
8.3	Esquema de l'aplicació.....	91

Resum

El present projecte representa un conveni de col·laboració entre l'Institut Geològic de Catalunya (IGC) i el Departament de Geografia de la Universitat Autònoma de Barcelona (UAB) per a realitzar el Projecte final del Màster en Tecnologies de la Informació Geogràfica, 12a edició, a través de pràctiques professionals.

L'objectiu per fer aquest projecte és crear una Base de Dades amb Oracle 10g per tal d'emmagatzemar les dades d'interferometria d'una manera ordenada i fàcil de consultar. A més a més de la base de dades, també s'ha de crear un Aplicatiu de càrrega que permeti a l'usuari guardar noves dades a la base de dades de forma senzilla. Per a fer aquest aplicatiu de càrrega s'ha utilitzat el llenguatge .NET a través de Visual Basic 2010.

Primer de tot s'ha hagut de dissenyar un model lògic de dades amb el programa Dezign que ens ha permès estructurar la base de dades segons les necessitats que tenia l'Institut Geològic. Un cop realitzat el model lògic s'ha creat la base de dades amb Oracle a través de sentències SQL. Tot seguit s'ha fet la càrrega inicial de totes les dades i finalment l'aplicatiu de càrrega que ens permet incorporar noves dades a la base.

Per concloure, s'ha de dir que s'han complert els objectius que s'havien marcat tan pel que fa a la Base de Dades, com per l'aplicatiu de càrrega.

Resumen

El presente proyecto representa un convenio de colaboración entre el Institut Geològic de Catalunya (IGC) y el Departament de Geografia de la Universitat Autònoma de Barcelona (UAB) para realizar el Proyecto final del Màster en Tecnologies de la Informació Geogràfica, 12a edició, a través de prácticas profesionales.

El objetivo para este proyecto es crear una Base de Datos con Oracle 10g para almacenar los datos de interferometría de una manera ordenada y fácil de consultar. Además de la base de datos, también se creará un Aplicativo de carga que permita al usuario guardar nuevos datos en la base de datos de forma sencilla. Para hacer esta aplicación de carga se ha utilizado el lenguaje .NET a través de Visual Basic 2010.

En primer lugar se ha tenido que diseñar un modelo lógico de datos con el programa DeSign que nos ha permitido estructurar la base de datos según las necesidades que tenía el Instituto Geológico. Una vez realizado el modelo lógico se ha creado la base de datos con Oracle a través de sentencias SQL. A continuación se ha hecho la carga inicial de todos los datos y finalmente la aplicación de carga que nos permite incorporar nuevos datos a la base.

Para concluir, hay que decir que se han cumplido los objetivos que se habían marcado respecto a la Base de Datos y al aplicativo de carga.

1. Introducció

1.1. Presentació

Aquest informe està emmarcat dins del Màster en Tecnologies de la Informació Geogràfica, 12a edició (MTIG) que es cursa al Laboratori d'Informació Geogràfica i Teledetecció (LIGIT) de la Universitat Autònoma de Barcelona (UAB) amb un conveni de col·laboració amb l'Institut Geològic de Catalunya (IGC), i es tracta del projecte final del present Màster.

El document és el resum del projecte realitzat al llarg de quatre mesos a l'IGC i descriu tots els passos que s'han seguit per tal d'assolir els objectius que s'han fixat. Per arribar a aquests objectius s'ha dividit el projecte en diferents fases. Al llarg d'aquest document explicatiu, es detalla cada una de les fases amb tots els processos i mètodes que s'han utilitzat per tal de desenvolupar-lo.

1.2. Marc institucional

Aquest projecte es realitza gràcies a un acord entre l'IGC i el LIGIT per tal de realitzar el projecte final del 12è MTIG, gràcies a un conveni de col·laboració entre ambdues parts. De fet, en edicions anteriors de l'MTIG, ja hi havia hagut col·laboracions entre aquestes dues entitats.

La durada d'aquest projecte és d'uns quatre mesos, des del Setembre al Desembre del 2010 amb un conveni de mitja jornada.

1.3. Marc Teòric

Les dades que hi ha per fer el projecte, les recull l'ICC i les processa abans de lliurar-les a l'IGC. El processat consisteix en fer una interferometria de radar diferencial. Aquesta tècnica ens permet detectar desplaçaments de la superfície terrestre de l'ordre d'uns pocs centímetres a partir de diferents imatges de satèl·lit (captures seriades en el temps), on es comprova el moviment a partir d'uns punts coneguts de referència al llarg del temps. A partir d'aquestes dades, es pot calcular la velocitat del moviment d'aquests punts d'estudi. Per tant, aquest tipus de tècniques ens permeten l'anàlisi temporal de processos dinàmics, al poder establir amb una gran precisió desplaçaments.

Tal i com s'acaba d'explicar, l'IGC té interès per conèixer quins són el moviment que hi ha la superfícies de Catalunya, ja que l'Institut Geològic es centra fonamentalment en l'estudi, l'assessorament, la investigació i la prevenció del sòl i subsòl de Catalunya.

2. Descripció del projecte

En aquest apartat es troba explicat en que consisteix el projecte.

Es tracta de crear un Sistema d'Informació per tal d'organitzar les dades d'interferometria que té l'IGC sobre el moviment de diferents punts específics de diverses zones de Catalunya. Les tasques a realitzar són les següents:

- Crear un model lògic
- Implementar aquest model en una base de dades
- Crear un aplicatiu de càrrega de dades

A continuació es descriuen més profundament aquestes tasques:

S'ha de crear un model lògic coherent i eficaç per tal de construir la base de dades dels diferents punts d'estudi . Un cop creada la base de dades, es farà l'aplicatiu de càrrega que servirà per quan algú de l'IGC necessiti introduir-ne de noves a partir de les dades d'interferometria en format *.txt que els serveix l'ICC. L'aplicatiu també permetrà descarregar-les.

2.1. Situació i necessitats actuals

L'IGC es va crear l'any 2005 gràcies a la llei 19/2005 i va quedar adscrit al Departament de Política Territorial i Obres Públiques (DPTOP). Des d'aquell moment van quedar ben definides quines són les seves funcions, que es centren fonamentalment en l'estudi, l'assessorament, la investigació i la prevenció del sòl i el subsòl de Catalunya. Les activitats principals de l'Institut són, segons la seva pàgina web:

“elaborar el mapa geològic de Catalunya, i portar a terme programes de cartografia, bases de dades i sistemes d'informació sobre el sòl i el subsòl; desenvolupar i mantenir la xarxa sísmica; estudiar i avaluar els riscos geològics, inclòs el sísmic i el d'allaus; elaborar i fomentar estudis, treballs i avaluacions en el camp de la geologia i de les ciències que s'hi relacionen; assessorar i prestar assistència tècnica en el camp de la geologia i de les disciplines afins al DPTOP, altres departaments i altres administracions; supervisar, si se sol·licita, estudis geotècnics de terrenys; elaborar estudis sobre el sòl i el subsòl; establir protocols a seguir en l'elaboració dels estudis geològics, geofísics i geotècnics, i facilitar informació aplegada en les bases de dades.”

És amb la finalitat d'estudiar els moviments de la superfície de Catalunya on s'ubica aquest projecte.

Actualment, l'IGC, ja té un projecte similar que data del Setembre 2006. Es tracta d'un projecte més ampli que entre d'altres dades, utilitza les dades interferomètriques que també s'utilitzaran en aquest cas. Es tracta simplement d'un aplicatiu de càrrega de dades a un repositori centralitzat. El problema és que aquest aplicatiu no és gaire funcional, i per això es vol crear un nou sistema de càrrega més eficient, començant per les dades d'interferometria. També, es necessari que el programa d'introducció de noves dades sigui més senzill i còmode per tal que sigui una eina funcional pels tècnics de l'IGC.

2.2. Usuaris

En un principi, l'aplicatiu de càrrega de dades, es crea per tal de ser utilitzats per diferents tècnics de l'Institut. Professionals de l'Àrea de Geotècnica i Riscos Geològics, l'Àrea de Geofísica i de l'Àrea de Teledetecció, són dels que més utilitzaran el programa.

2.3. Definició dels objectius

Els objectius que es volen assumir amb aquest projecte seran especificats a continuació.

L'objectiu principal del projecte és la creació d'un Sistema d'Informació Geogràfica que permeti a l'IGC emmagatzemar i visualitzar les dades que obté de l'ICC a través de la interferometria de imatges i així se capaç d'analitzar els moviments que hi ha al llarg del temps en unes zones determinades. Per tal de complir aquest objectiu principal, s'han d'assolir uns altres objectius previs. A continuació s'explicaran quins.

El primer que s'ha de fer per tal de construir un SIG, és tenir una bona base de dades relacionada entre si de la forma més coherent possible. Per tant, el primer objectiu serà crear un model lògic que ens permeti relacionar atributs iguals de taules diferents i implementar aquest model a la nostra base de dades. És un dels objectius més importants del projecte, ja que tenir un bon model lògic permet que la programació sigui més senzilla a l'hora de fer qualsevol consulta. També serveix perquè si s'ha de modificar i/o ampliar més endavant, no hi hagi problemes. A més a més, una bona base de dades permetrà donar resposta a les consultes que es facin en el futur.

El següent objectiu, és crear una aplicació per carregar les dades que tenim actualment, i per afegir-hi dades en el futur. Aquest aplicatiu de dades és la part més important del projecte, ja que ha de ser senzill i pràctic. El programa ha de tenir les finestres justes per tal d'entrar noves dades a la base de dades, canviar dades ja existents o eliminar les dades que ja no es necessitin. Un entorn aparentment simple, però que ens permeti incorporar totes aquestes noves dades que necessitem de forma ràpida i efectiva. S'ha d'utilitzar una aparença harmonitzada amb Windows.

2.4. Planificació inicial dels treballs

La planificació dels treballs al llarg dels quatre mesos que durarà el projecte, és la que es veu a la taula següent. Les primeres setmanes serviran per familiaritzar-se amb les dades i planificar el projecte. A principis de Novembre s'ha d'acabar el model lògic i tot seguit, començar amb l'aplicació de càrrega de dades per acabar-lo a al Desembre. També és important recordar, que al llarg d'aquests quatre mesos s'anirà recopilant informació per tal de poder escriure l'informe final.

FASE	NOM DE LA FASE	TASCA ESPECÍFICA	SETEMBRE				OCTUBRE				NOVEMBRE				DESEMBRE			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Fase 1	Planificació i Documentació	Estructuració del projecte																
		Veure precedents i programari a utilitzar																
Fase 2	Creació del Model de Dades	Familiarització i ordenació de les dades																
		Creació model lògic i conceptual de la BD.																
Fase 3	Creació de l'Aplicatiu de Càrrega	Creació i càrrega de dades amb Oracle																
		Aplicatiu de Càrrega amb Visual Studio																
Fase 4	Redacció del Projecte	Recopilació d'informació pel redactat																

Figura 1: Esquema de la planificació dels treballs a realitzar durant el projecte

3. Requeriments

3.1. Requeriments funcionals

Els requeriments que l'IGC fa són diversos. Per començar, el més important per a l'Institut és l'aplicatiu de càrrega de dades, ja que actualment ja n'hi ha un, però és poc funcional. Les funcions que ha de tenir aquest aplicatiu són les següents:

- Per tal que sigui segur i que no qualsevol pugui accedir-hi, ha de tenir una connexió a la base de dades a través de contrasenya
- Introducció o eliminació de noves dades, i edició o eliminació de taules
- Opció d'indicar en quina taula es volen introduir les noves dades o quina es vol editar
- La inserció de dades ha de ser simple, ràpida i canviant el mínim les dades rebudes per l'ICC
- Ha de ser un programa intuïtiu que permeti fer-lo servir a un usuari no experimentat sense problemes

3.2. Requeriments de programari

Per a fer el model lògic s'ha donat total llibertat a l'hora d'escollir el programa més adient, i en aquest cas s'ha escollit DeZign for Databases.

En canvi, per l'aplicatiu de càrrega sí que hi ha el requeriment de fer-lo amb Microsoft Visual Studio 2010 per tal que sigui un entorn d'estil Windows. L'entorn de desenvolupament que t'ofereix és la programació amb .NET. Es tria aquest entorn i llenguatge perquè tant l'alumne com els tècnics de l'IGC estan familiaritzats amb aquest entorn.

També la base de dades té un requeriment de programari i s'ha de fer amb Oracle – ODACHome1 que és el sistema utilitzat a l'IGC. Dins del programa Oracle, es fa servir la fulla de treball SQL Plus que ens permet crear la base de dades a partir del model creat pel DeZign i introduir noves dades segons ens convingui.

4. Disseny i implementació de la BBDD

4.1 Requeriments i objectius

En aquest apartat es definirà com serà el disseny i quina funcionalitats tindrà el model de dades. Per tant s'han de saber quines són les exigències del client abans de començar.

Al model de dades creat és on s'emmagatzemarà tota la informació que actualment tenim i és on s'hauran d'introduir les noves dades que arribin a partir d'ara.

El sistema escollit per contenir aquest model de dades i totes les mesures és el Sistema Gestor de Bases de Dades (SGBD) Oracle, en concret la versió 10.2.0.2.20. S'ha escollit aquest sistema perquè l'usuari ja està familiaritzat amb aquest programari i és una bona eina de treball.

S'hauran de crear diversos procediments en el model de dades que permetran actualitzar totes les taules quan s'introdueixin noves dades, alhora que es creen taules amb els històrics per si en el futur es necessita conèixer quines dades hi havia amb anterioritat.

4.2 Aproximació tecnològica

A continuació es veurà quin és el programari utilitzat i quins llenguatges específics es faran servir per tal de crear el model.

4.2.1 Programari

- Dezignn for Databases Versió: 6.3.0: Disseny de la Base de dades
- Microsoft Office Excel 2007: Ordenació de les dades inicials.
- Oracle Provider for OLE DB 10.2.0.2.20: Creació i implementació de la base de dades. Es farà servir la 'Hoja de trabajo SQL Plus' que permet executar sentències SQL.
- Oracle SQL Developer Release 2.1 Release Notes: També ens permet la creació, la modificació i l'organització de bases de dades.

4.2.2 Llenguatges de programació

Per crear la base de dades a Oracles, s'utilitzarà diferents llenguatges:

- SQL (Structures Query Language). Es tracta d'un llenguatge estàndard de comunicació amb bases de dades relacionals universal. L'SQL ens permetrà crear la BD i fer la primera càrrega massiva de les dades.
- PL/SQL (Procedural Language / Structured Query Language). És un llenguatge propietat d'Oracle que ens permet emmagatzemar procediments. Amb aquest llenguatge podem guardar els triggers, altrament dits disparadors, que ens permeten crear camps autonumèrics de forma automàtica.

4.3. Model conceptual

El primer pas per a crear el model lògic és crear un model conceptual.

El model conceptual és un peça fonamental on s'hi ha de col·locar tots els actors que intervindran en la BD. Es tracta de fer un esquema amb els principals elements de la BD. És el pas previ per crear el model lògic que ens permetrà crea la nova BD, per tant, si ja cometem errors en el model conceptual, els anirem arrossegant al llarg de tot el projecte.

A l'hora de crear aquest model conceptual és imprescindible treballar conjuntament amb el client, ja que entendre la forma com s'emmagatzemen les dades permetrà actualitzar-les i modificar-les en el futur.

Per generar el model conceptual tindrem en compte tres passos definits per Marqués l'any 2001:

- Primer identificarem les entitats que conformen les diverses taules
- En segon lloc s'ha d'identificar les relacions que hi ha entre les entitats creades amb anterioritat.
- Per últim, hem d'identificar els atributs, que són les característiques de cada entitat.

La presentació d'aquest model conceptual es fa a través d'un esquema dibuixat on es poden veure els tres elements acabats de definir. En el nostre cas, ja que el temps és limitat s'ha optat per fer al model conceptual a l'hora del model lògic. Així que s'ha revisat l'esquema del model lògic.

Les diferents entitats i atributs que trobem en el nostre model conceptual són les següents:

Satèl·lit: Ens permet saber amb quins satèl·lits s'han pres les dades. Els seus atributs són: el nom del satèl·lit i el sensor que porta.

Imatge: Són les fotografies que fa el satèl·lit i de les quals s'extreuen les dades d'interferometria que s'utilitzen. Els atributs són: la data de la fotografia, la seva resolució, el seu track i el seu frame.

Zona: Són les àrees d'estudi en les quals treballa l'IGC. Els seus atributs són: el nom de la zona, la orbita que ha fet el satèl·lit, la data d'inici de recollida de dades, la data final de la recollida de dades, la geometria del polígon de l'àrea d'estudi i les observacions que s'hi vulgui fer.

Punt: Es tracta dels punts exactes on es fa l'estudi del moviment. Els atribut principal és: la geometria de cada punt a partir de les seves coordenades.

Velocitat: Aquesta taula en permet veure la velocitat a que es mouen aquests punts. Els atributs són: la velocitat i la coherència d'aquesta.

Mesura: és la distància que es mou cada punt al llarg del temps. El seu atribut principal és: la distància del moviment.

Com que les relacions són difícils de comprendre descrites simplement, s'explicaran més endavant, a través de l'esquem del model lògic.

4.4. Model lògic

Quan ja tenim el model conceptual és crea el model lògic amb el programa DeSign for Databases. Es decideix per aquest programa, ja que permet escollir quin SGBD s'utilitzarà posteriorment, i ell mateix ja et crea les seqüències SQL per implementar directament a l'Oracle.

El resultat final del model lògic és el següent i continuació s'explicarà com i perquè s'ha fet així.

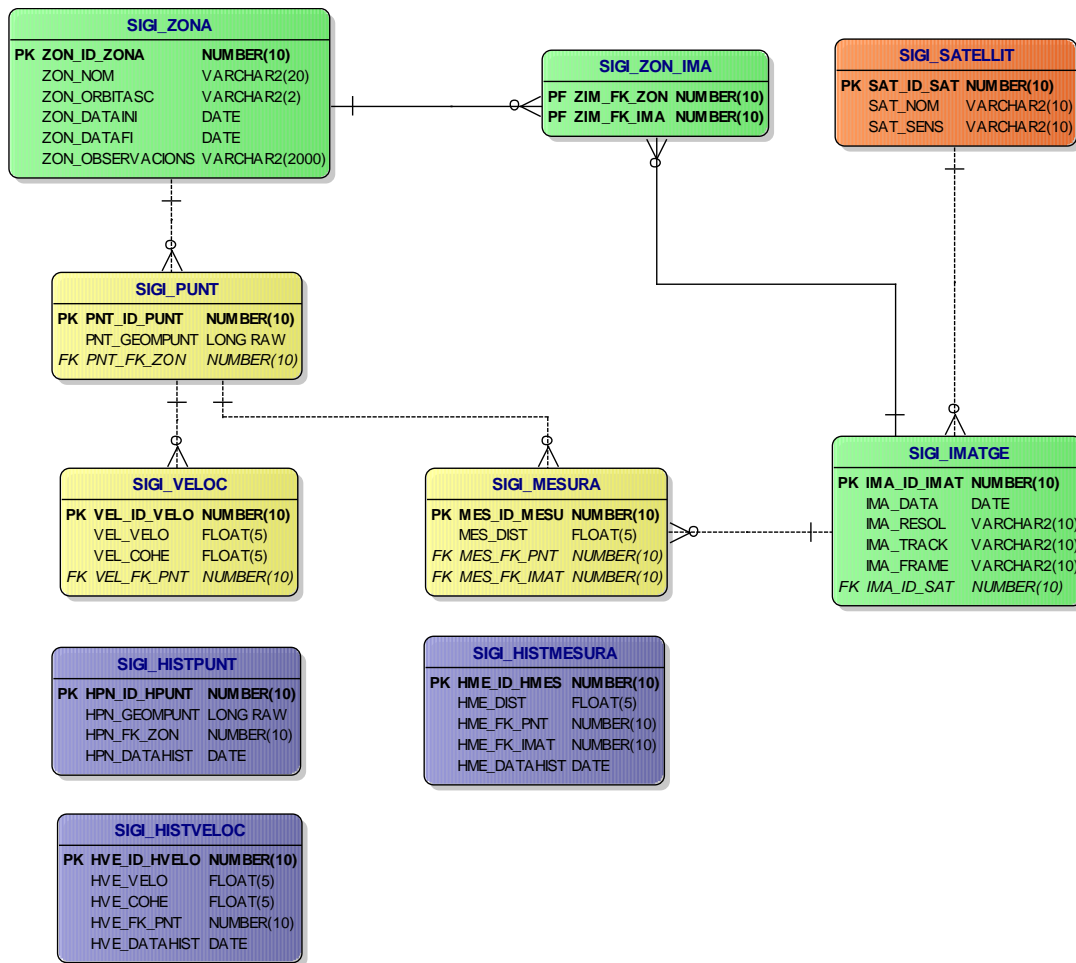


Figura 2: Model lògic de la Base de Dades

4.4.1. Aspectes formals del model lògic

Per tal d'identificar les taules que fan referència al nostre estudi, s'ha decidit posar unes sigles davant de cada una. Així, s'ha decidit que si posi SIGI que és una abreviatura de Sistema d'Informació Geogràfica d'Interferometria.

El mateix raonament s'ha utilitzat amb els atributs. Davant de cada un d'ells s'hi ha posat un codi de tres lletres que ens identifica a quina entitat pertany.

Aquesta decisió permet que qualsevol persona que no estigui familiaritzada amb el model, ràpidament pugui relacionar cada atribut amb la seva entitat.

Un altre aspecte formal és la codificació de cada taula. A cada entitat se li associa una clau primària que és un codi autonumèric que fa d'identificador dels valors que té la taula.

4.4.2. Relacions entre entitats

En aquest apartat es descriuran quines són les relacions que hi ha entre les diferents entitats per tal de justificar el model lògic presentat.

La primera relació és la que hi ha entre el Satèl·lit i la Imatge. Es tracta d'una relació amb una cardinalitat de un o molts. Això significa que cada satèl·lit té unes imatges determinades, i que per tant, una imatge pertany a un únic satèl·lit.

La segona relació és la que ens ajunta les imatges amb les zones d'estudi. Aquesta relació és de molts a molts, i per això s'ha de crear una taula intermèdia que ens permeti fer-la. En aquest cas, una imatge pot donar informació de varies zones i una zona pot agafar informació de diferents imatges.

La relació entre zones i punts, és de una molts, ja que cada zona conté molts punts, però cada punt pertany a una zona concreta.

Seguidament hi ha la relació entre punt i velocitats, i punt i mesures. Les dues són una relació de un a molts. S'explica fàcilment, ja que cada punt té varies velocitats i també varies mesures.

Finalment, tenim la relació entre imatges i mesures que és una relació de un a molts. L'explicació és la mateixa que anteriorment, és a dir, una imatge ens permet obtenir diverses mesures.

4.4.3 Disparadors (*Triggers*)

A l'hora de crear el model lògic pot sorgir la necessitat de crea *triggers* (disparadors). Un *trigger* (o disparador) en una base de dades, és un procediment que s'executa quan es compleix una condició establerta en realitzar una operació d'inserció (*insert*), actualització (*update*) o esborrament (*delete*).

En la creació de la base de dades s'han creat *triggers* per tal d'automatitzar diferents procediments, entre ells, la creació dels atributs autonumèrics. A continuació s'especifica el procediment i el codi implementat per a fer-ho. Aquests *triggers* es fan al mateix programa on hem creat el model lògic, és a dir, el DeZign. Com a exemple de la utilització dels *triggers* veurem com s'han fet els que ens permeten autonumerar els atributs.

1. Primer de tot creem una seqüència.
2. Una cop creada cal editar-ne les seves propietats . Com es tracta d'un valor autonumèric es determina 0 com el valor mínim amb un increment de 1 per a cada valor. D'aquesta forma, cada vegada que s'executi el *trigger* augmentarà el seu valor en una unitat.
3. Seguidament s'editen les propietats de l'entitat on es vol crear el *trigger*. Es selecciona la pestanya de *triggers* i se li dóna el nom desitjat. A continuació, i mitjançant l'editor de codi, s'escriu el codi *PL/SQL* corresponent.
4. El codi dels diferents *triggers* autonumèrics implementats és el següent:

```
create or replace  
TRIGGER SIGI_SAT_TR_SEQ  
BEFORE INSERT ON SIGI_SATELLIT  
FOR EACH ROW  
BEGIN  
IF :NEW.SAT_ID_SAT IS NULL THEN  
SELECT SIGI_SAT_SEQ.NEXTVAL  
INTO :NEW.SAT_ID_SAT  
FROM DUAL;  
END IF;  
END;
```

El *trigger* s'executa cada vegada que es vol introduir un nou valor a la taula, en aquest cas SIGI_SATELLIT. Es selecciona el codi de satellit i s'hi insereix un nou valor segons la seqüència especificada.

Aquest tipus de *trigger* es troba a totes les entitats que contenen camps autonumèrics.

4.5 Implementació de la Base de Dades

El procés d'implementació de la base de dades consisteix en construir físicament la base de dades en el SGBD escollit, aquest cas oracle. Aquest procés d'implementació conté un doble vessant:

1. En primer terme cal, mitjançant el programa DeSign, crear el codi que permeti la implementació de la base de dades. El gran avantatge d'aquest programari és que aquest pas és senzill i ràpid. Tan sols cal prémer a la barra d'eines *database* i tot

seguit *generate database*. El mateix programa crea les seqüències SQL que ja es poden executar directament en oracle.

Pel que fa a les *constraints* cal evitar que el *Dezign* les anomeni automàticament ja que aquest pas provoca que existeixi una duplicitat en algunes d'elles i això provoca errors a l'hora d'executar el SQL. Indicant al programa, quan es genera el SQL, que no assigni de forma automàtica un nom als diferents *constraints* evitem aquest problema.

2. Una vegada generat el SQL ja es pot executar en oracle. En aquest cas es fa servir SQL **Plus Worksheet*. Simplement es copien les ordres SQL i s'executen a la finestra del programa.

Tot i això cal destacar les principals sentències SQL generades:

Creació d'una taula:

```
CREATE TABLE SIGI_PUNT (
PNT_ID_PUNT NUMBER(10) CONSTRAINT NN_SIGI_PUNT_PNT_ID_PUNT
NOT NULL,
PNT_GEOMPUNT SDO_GEOMETRY CONSTRAINT
NN_SIGI_PUNT_PNT_GEOMPUNT NOT NULL,
PNT_FK_ZON NUMBER(10) CONSTRAINT NN_SIGI_PUNT_PNT_FK_ZON NOT
NULL,
CONSTRAINT PK_SIGI_PUNT PRIMARY KEY (PNT_ID_PUNT)
);
CREATE INDEX IDX_SIGI_PUNT_1_FK ON SIGI_PUNT (PNT_FK_ZON);
CREATE UNIQUE INDEX IDX_SIGI_PUNT_PK ON SIGI_PUNT (PNT_ID_PUNT);
COMMENT ON TABLE SIGI_PUNT IS 'Aquesta taula tindrà tots els punts que
s"estudïen a Catalunya.';
COMMENT ON COLUMN SIGI_PUNT.PNT_ID_PUNT IS 'Identificador únic de
cada punt';
COMMENT ON COLUMN SIGI_PUNT.PNT_GEOMPUNT IS 'Permet
l'emagatzematge de la geometria de cada punt';
COMMENT ON COLUMN SIGI_PUNT.PNT_FK_ZON IS 'FK de la zona, és a dir
que cada punt mostra l"identificador de la zona a la qual pertany'
```

Creació dels constraints:

```
ALTER TABLE SIGI_PUNT ADD CONSTRAINT SIGI_ZONA_SIGI_PUNT
FOREIGN KEY (PNT_FK_ZON) REFERENCES SIGI_ZONA (ZON_ID_ZONA);
```

La resta d'elements (*triggers* i seqüències) ja han quedat explicats anteriorment. Per veure totes les sentències, veure annex 8.2.

4.6 Càrrega inicial de dades

La càrrega inicial de dades a la base de dades té per finalitat omplir les diferents taules amb les dades reals.

No ha estat una tasca fàcil degut al gran nombre de dades i l'escassa sistematització que aquestes presentaven. Degut a això es decideix que la càrrega inicial de dades es farà amb sentències *SQL* generades en *Excel* a partir de les dades originals. És a dir, que es crea un *Excel* amb totes les dades a emplenar i mitjançant la concatenació de dades i text es creen les diverses sentències *SQL* que després s'executaran al *SQL *plus*.

A partir de les dades de les taules de Excel es crea una fórmula que concatena totes les dades creant al mateix temps una seqüència *INSERT* en llenguatge *SQL*. El resultat final de la concatenació és el següent:

```
INSERT INTO SIGI_SATELLIT (SAT_NOM, SAT_SENS, SAT_RESOL,
SAT_ANGLE) VALUES ('ERS', 'AMI', '30', '23');
```

Una vegada aquest procés s'efectua per al conjunt de les dades, sempre tenint en comte les regles d'integritat, s'obté un conjunt de sentències *SQL* que executades al *SQL *plus* permeten la càrrega de dades a la BBDD.

L'aspecte més important a considerar en aquest sentit és el següent:

La importància de l'ordre d'inserció de les dades a cada una de les teules. Per evitar errors en la inserció de les dades degut a les relacions establertes entre les diverses entitats cal seguir l'ordre que s'especifica a continuació:

Satèl·lit → Imatge → Imazon → Zona → Punt → Velocitat → Mesura

Tot i això, per fer la inserció dels punts, s'ha hagut de fer d'una forma completament diferent. Això és degut a que es vol guardar la geometria dels punts. Per tal de poder fer-ho, s'ha hagut d'utilitzar unes petites aplicacions (*shp2sdo* i

sqlldr) que ens permeten executar les comandes necessàries per a guardar aquesta geometria. A continuació s'explica detalladament com es fa:

1. Primer s'han de tenir els punts en un shapefile.
2. Obrim el símbol de sistema i executem aplicatiu shp2sdo. Aquest programet ens permetrà crear tres arxius, un .sql per crear la taula, un .ctl per controlar la càrrega de les dades i un arxiu .dat que conté totes les dades dels punts i la seva geometria. Aquí es veu els paràmetres que es necessiten per utilitzar el programa:

```
D:\PFMTIG12\Albert\oracle_geom>shp2sdo.exe
```

```
shp2sdo - Shapefile(r) To Oracle Spatial Converter
```

```
Version 2.15 21-May-2004
```

```
Copyright 1997,2004 Oracle Corporation
```

```
For use with Oracle Spatial.
```

```
Input shapefile (no extension): sigi_punt
```

```
Shape file sigi_punt.shp contains 32134 points
```

```
Output table [sigi_punt]:
```

```
Output data model [O]:
```

```
Geometry column [GEOM]: pnt_geompunt
```

```
ID column []: pnt_id_punt
```

```
Starting number [1]: 1
```

```
Points stored in SDO_POINT_TYPE ? [Y]: y
```

```
Use a spatial reference system ID (SRID) ? [N]: 23031
```

```
Change tolerance value from the default (0.00000005) ? [N]:
```

```
Generate data inside control files ? [N]:
```

```
Target database Oracle8i? [N]:
```

```
Spatial Data requires more than 6 digits precision? [N]: y
```

```
Bounds: X=[298397.500000,491256.500000]
```

```
Y=[4546764.500000,4664367.500000]
```

```
Override ? [N]:
```

Processing shapefile sigi_punt into spatial table SIGI_PUNT

Data model is object-relational

Geometry column is PNT_GEOMPUNT

Id column is PNT_ID_PUNT

Numbered from 1

Points stored in SDO_POINT attributes

Data is in a separate file(s)

Control file generation for Oracle9i or higher

Spatial data loaded with 10 digits of precision

Conversion complete : 32134 points processed

The following files have been created:

sigi_punt.sql : SQL script to create the table

sigi_punt.ctl : Control file for loading the table

sigi_punt.dat : Data file

3. Tot seguit es crea la taula a partir del fitxer sql que s'ha obtingut:

D:\PFMTIG12\Albert\oracle_geom>sqlplus

*SQL*Plus: Release 10.2.0.3.0 - Production on Tue Nov 16 17:28:45
2010*

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Enter user-name: sigi

Enter password:

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production

With the Partitioning, OLAP and Data Mining options

SQL> @sigi_punt.sql

DROP TABLE SIGI_PUNT

ERROR at line 1:

ORA-00942: table or view does not exist

Table created.

0 rows deleted.

1 row created.

Commit complete.

SQL> quit

- Per finalitzar, es fa la càrrega de les dades amb l'aplicatiu sqlldr de la forma següent:

```
D:\PFMTIG12\Albert\oracle_geom>sqlldr          userid=signi/signi,
control=signi_pnt.ctl, log=import.log, bad=errors.bad, data=signi_punt.dat,
discard=discard
```

Un cop es tenen totes les dades carregades es comprova mitjança SQL Developer que tot s'hagi emmagatzemat correctament. A la figura següent s'està comprovant que les dades de punts s'hagin carregat com es volia. Amb aquest programa podem modificar varies coses de la base de dades sense necessitat d'escriure-ho amb sentències SQL, i per tant és de gran ajuda per retocar petits errors que haguem pogut cometre.

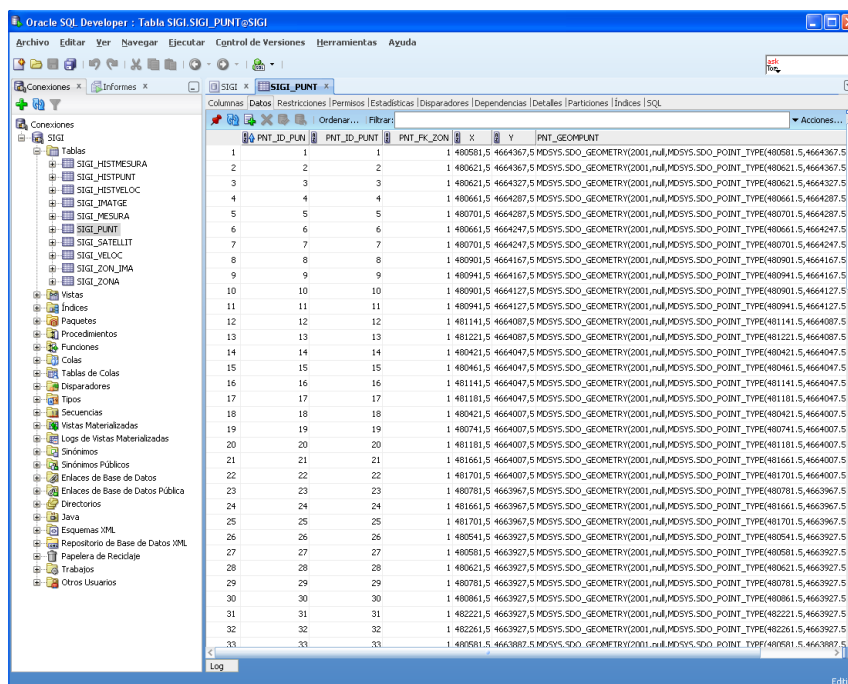


Figura 3: Interfície del programa Oracle SQL Developer Release 2.1 Release Notes

5 Aplicació de càrrega de dades

5.1 Requeriments i objectius

Els requeriments per part de l'empresa, i per tant, els objectius a assolir, són els següents:

Requeriments funcionals:

El programa ha de permetre la inserció de noves dades d'interferometria.

S'ha de poder carregar noves dades a partir de fitxers .txt.

El programa a de ser intuïtiu. Un usuari no experimentat ha de poder fer servir el programa sense grans dificultats.

La inserció de dades ha de ser segura, per tant es necessitarà una clau d'accés.

Requeriments no funcionals:

La inserció de les dades ha de ser ràpida, i senzilla a partir dels documents inicials que té l'IGC.

En aquest cas s'ha acordat fer el programa amb Visual Studio 2010 amb el llenguatge de programació .NET.

5.2 Aproximació tecnològica

Tal i com s'especifica a l'últim apartat dels requeriments en aquesta etapa del projecte el programari a utilitzar és Visual Studio 2010 i el llenguatge de programació .NET. Diverses són les raons que porten a aquesta elecció:

a) El software VS2010 és el programari que té l'IGC.

b) L'estudiant es troba familiaritzat amb l'entorn i llenguatge de programació *Visual* i per tant, l'adaptació a l'entorn .NET és més ràpida, augmentant així la productivitat.

c) Els tècnics del IGC es troben familiaritzats amb la programació en .NET amb el que es facilita en un futur les millores i adaptacions que puguin sorgir.

d) Existeix abundant documentació sobre oracle i .NET el que facilitarà el desenvolupament del programa.

5.3 Programació de l'aplicació de càrrega de dades

En el present apartat s'explicarà el procés que s'ha seguit per la creació de l'aplicació de càrrega de dades. Es tracta d'un apartat tècnic més centrat en com s'implementen les diverses funcionalitats de l'aplicació i, per tant, no està dirigit als usuaris.

Per tal de facilitar la comprensió a més de les explicacions i el codi es mostra la visió que té l'usuari, és a dir les pantalles visualitzades per l'usuari.

5.3.1 Connexió a la Base de dades

En iniciar-se l'aplicació l'usuari ha d'introduir les dades referents a la connexió a la base de dades, l'usuari i la contrasenya, tal i com es pot veure a la següent figura.

Figura 4: Formulari de connexió

Una vegada l'usuari introdueix l'usuari i la contrasenya i fa clic sobre el botó *connectar* l'aplicació intenta connectar-se a la base de dades a través de la llibreria *Oracle.DataAccess.Client*.

El codi de connexió és el següent:

```
PublicSub creaDataset()
'definir la conexio a la base de dades
Dim uname AsString
Dim slength AsLong
    uname = Space(10) 'ens dona suficient espai per posar els valors
' llegim l'arxiu INI //a implementació LIGIT no cal ho poso jo a pel
    slength = GetPrivateProfileStringKey("variables", "instancia",
"anonymous", uname, 10,
"\\montsec\treball\egeol\BBDD_SIG\1_Aplicacions\2_Aplicacions_GIS\SIA\config.ini"
)
    uname = Strings.Left(uname, slength) 'extreu l'string que ens dona l'INI
    MsgBox("Instancia: "& uname)

Dim usr AsString = txtUsuari.Text
Dim pwd AsString = txtContrasenya.Text
```



```
        dataSource = uname
Dim ora_db AsString = "Data Source=" & dataSource & " ;User Id=" & usr & "
;Password=" & pwd & ";"

'establim la connexió a la base de dades
Dim conn AsNewOracleConnection(ora_db)
    conn.ConnectionString = ora_db

'definirem cadenes sql per recuperar la info
Dim sqlSatelit AsString = "SELECT * FROM SIGI_SATELLIT ORDER BY SAT_ID_SAT"
Dim sqlImatge AsString = "SELECT * FROM SIGI_IMATGE ORDER BY IMA_ID_IMAT"
Dim sqlImaZon AsString = "SELECT * FROM SIGI_ZON_IMA"
Dim sqlZona AsString = "SELECT * FROM SIGI_ZONA ORDER BY ZON_ID_ZON"
Dim sqlPunt AsString = "SELECT PNT_ID_PUNT, PNT_FK_ZON, X, Y FROM SIGI_PUNT ORDER
BY PNT_ID_PUNT"
Dim sqlVeloc AsString = "SELECT SIGI_VELOC.*, SIGI_ZONA.ZON_ID_ZON FROM
SIGI_VELOC, SIGI_PUNT, SIGI_ZONA where SIGI_ZONA.zon_id_zon =
SIGI_PUNT.pnt_fk_zon and SIGI_PUNT.pnt_id_punt = SIGI_VELOC.vel_fk_pnt ORDER BY
VEL_ID_VELO"
Dim sqlMesura AsString = "SELECT SIGI_MESURA.*, SIGI_ZONA.ZON_ID_ZON FROM
SIGI_MESURA, SIGI_PUNT, SIGI_ZONA where SIGI_ZONA.zon_id_zon =
SIGI_PUNT.pnt_fk_zon and SIGI_PUNT.pnt_id_punt = SIGI_MESURA.mes_fk_pnt ORDER BY
MES_ID_MESU"
Dim sqlHistpunt AsString = "SELECT HPN_ID_HPUNT, HPN_FK_ZON, HPN_DATAHIST FROM
SIGI_HISTPUNT ORDER BY HPN_ID_HPUNT"
Dim sqlHistmesura AsString = "SELECT * FROM SIGI_HISTMESURA ORDER BY HME_ID_HMES"
Dim sqlHistveloc AsString = "SELECT * FROM SIGI_HISTVELOC ORDER BY HVE_ID_HVELO"

'creem adaptadors de dades per executar les cadenes sql definides
    daSatelit = NewOracleDataAdapter(sqlSatelit, conn)
    daSatelit.AcceptChangesDuringUpdate = True
    daImatge = NewOracleDataAdapter(sqlImatge, conn)
    daImatge.AcceptChangesDuringUpdate = True
    daImaZon = NewOracleDataAdapter(sqlImaZon, conn)
    daImaZon.AcceptChangesDuringUpdate = True
    daZona = NewOracleDataAdapter(sqlZona, conn)
    daZona.AcceptChangesDuringUpdate = True
    daPunt = NewOracleDataAdapter(sqlPunt, conn)
    daPunt.AcceptChangesDuringUpdate = True
    daVeloc = NewOracleDataAdapter(sqlVeloc, conn)
    daVeloc.AcceptChangesDuringUpdate = True
    daMesura = NewOracleDataAdapter(sqlMesura, conn)
    daMesura.AcceptChangesDuringUpdate = True
    daHistpunt = NewOracleDataAdapter(sqlHistpunt, conn)
    daHistpunt.AcceptChangesDuringUpdate = True
    daHistmesura = NewOracleDataAdapter(sqlHistmesura, conn)
    daHistmesura.AcceptChangesDuringUpdate = True
    daHistveloc = NewOracleDataAdapter(sqlHistveloc, conn)
    daHistveloc.AcceptChangesDuringUpdate = True
Dim cbSatelit AsNewOracleCommandBuilder(daSatelit)
Dim cbImatge AsNewOracleCommandBuilder(daImatge)
```

```

Dim cbImaZon AsNewOracleCommandBuilder(daImaZon)
Dim cbZona AsNewOracleCommandBuilder(daZona)
Dim cbPunt AsNewOracleCommandBuilder(daPunt)
Dim cbVeloc AsNewOracleCommandBuilder(daVeloc)
Dim cbMesura AsNewOracleCommandBuilder(daMesura)
Dim cbHistpunt AsNewOracleCommandBuilder(daHistpunt)
Dim cbHistmesura AsNewOracleCommandBuilder(daHistmesura)
Dim cbHistveloc AsNewOracleCommandBuilder(daHistveloc)

Try
'obrim connexió i omplim el dataset
    conn.Open()
    daSatelit.Fill(DataSet, "Satelit")
    daImatge.Fill(DataSet, "Imatge")
    daImaZon.Fill(DataSet, "ImaZon")
    daZona.Fill(DataSet, "Zona")
    daPunt.Fill(DataSet, "Punt")
    daVeloc.Fill(DataSet, "Veloc")
    daMesura.Fill(DataSet, "Mesura")
    daHistpunt.Fill(DataSet, "Histpunt")
    daHistmesura.Fill(DataSet, "Histmesura")
    daHistveloc.Fill(DataSet, "Histveloc")

'tancar connexió
    conn.Close()

'Obrim el següent formulari
frmTipus.Show()

'tanquem el formulari d'accés a la BBDD
Me.Hide()

'treballem amb els possibles errors i li donem un missatge que entengui l'usuari
Catch ex AsOracleException' només amb els errors d'oracle
SelectCase ex.Number
Case 1017
        MsgBox("Nom d'usuari o contrassenya incorrectes. No es pot
establir la connexió")
        txtUsuari.Text = ""
        txtContrasenya.Text = ""

Case Else
        MsgBox("Database error: " + ex.Message.ToString() & vbNewLine
&"Contacta amb l'administrador", MsgBoxStyle.Exclamation)
        txtUsuari.Text = ""
        txtContrasenya.Text = ""

EndSelect
EndTry
EndSub

```

Ens connectem a la base de dades i agafem totes les dades que necessitem i les guardem en un Dataset. D'aquesta manera, ja podem tancar la connexió i treballar sobre el aquest Dataset. Gràcies a això, ja no farà falta connectar-se gairebé en cap altre part de l'aplicatiu.

Primer de tot definim la connexió a la base de dades. En aquest cas, accedirem a un arxiu .ini de l'IGC on hi ha guardat el Tablespace on hi ha la base de dades. Tot seguit establim la connexió a la base de dades. Per fer-ho, primer definirem les cadenes SQL que ens permetran recuperar la informació que necessitem. Després creem els adaptadors de dades per tal d'executar les cadenes sql definides anteriorment i llavors ja podem obrir la connexió i omplir el Dataset amb el que acabem de definir. Ara ja es pot tancar la connexió i no es tornarà a obrir per gairebé cap altre procediment. Per últim hi posem un filtre d'errors per tal que si hi ha algun problema durant aquest procés aparegui un missatge d'avís.

Les sentències SQL que es fan servir són Selects que ens permeten agafar la informació que necessitem de cada taula. En el cas de la taula de punts, hem d'especificar quines columnes volem, ja que el recordset no pot guardar les dades referents a la geometria. A més a més, també relacionem les taules Velocitat i Mesura, amb l'ID de la taula Zona per tal de poder fer consultes més endavant.

A continuació ens apareix un formulari on hem d'escollir si volem introduir noves dades a la base de dades o consultar i exportar les taules ja existents.

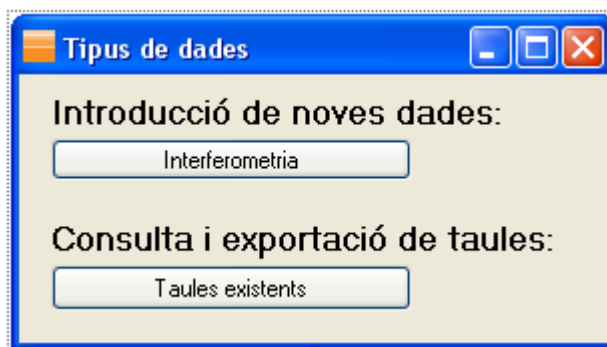


Figura 5: Formulari per escollir què volem fer

5.3.2 Modificar la taula satèl·lit i de la taula zona

Per tal de poder afegir nous satèl·lits a la taula, s'ha escollit que es pugui fer directament a la taula que s'està observant de forma manual. Així, tenim un formulari com el següent que ens permet veure la taula que tenim actualment:

Taula de satèl·lits

	SAT_NOM	SAT_SENS	SAT_RESOL	SAT_ANGLE
▶	ERS	AMI	30	23
	JERS-1	SAR	18	35
	ENVISAT	ASAR	30	42,5
	ALOS	PALSAR	10	38,7
	TerraSAR-X	-	3	45
	COSMO-SkyMed	-	3	45

Afegir nou satèl·lit
(directament a la taula)

Anar al Menú
Principal

Figura 6: Formulari per afegir nous satèl·lits

Per tal que funcioni aquest formulari, es fa servir el codi de programació següent:

```
PublicSub RejectChanges()
'ens permet que no es guardin els canvis fets a la taula
  dtsat.RejectChanges()
EndSub

PrivateSub btnGuardar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGuardar.Click
  dtsat = DGridSatellit.DataSource
  frmConnexio.daSatelit.Update(dtsat)

  dtsat.AcceptChanges()

Call GridFalse()
EndSub

PrivateSub frmSatellit_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
  dtsat = frmConnexio.DataSet.Tables("Satelit")
Call GridFalse()
EndSub

PrivateSub btnAfegir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAfegir.Click
Call GridTrue()
EndSub
```

```
PrivateSub btnMenu_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMenu.Click
Me.Hide()
frmTaulesDisponibles.Show()

Call RejectChanges()
Call GridFalse()
EndSub
```

En un primer moment no es pot editar la taula. Per tant, si es vol editar s'ha de fer un clic sobre el botó “*Afegir nou satèl·lit*” i llavors s'obre el mode d'edició de la taula que ens permet escriure directament a la taula. Un cop fets el canvis corresponents, si es pitja el botó guardar, representat amb el símbol d'un disquet, aquests canvis es guardaran a la base de dades. Això es possible ja que es fa un Update a daSatellit, que és un OracleDataAdapter, a partir del DataTable que hi ha en aquell moment en el DataGridView que s'està visualitzant.

Aquest procediment d'actualització de la base de dades es fa servir en gairebé totes les altres taules.

A més a més, s'ha d'afegir que el procés d'introduir nous satèl·lits és el mateix que es fa servir per introduir noves zones a la taula de Zona. A la resta és completament diferent. A continuació veurem com es fa.

5.3.3 Càrrega de noves imatges

El formulari per carregar noves imatges és semblant al de satèl·lit, però té algunes diferències. Visualment són molt semblants:

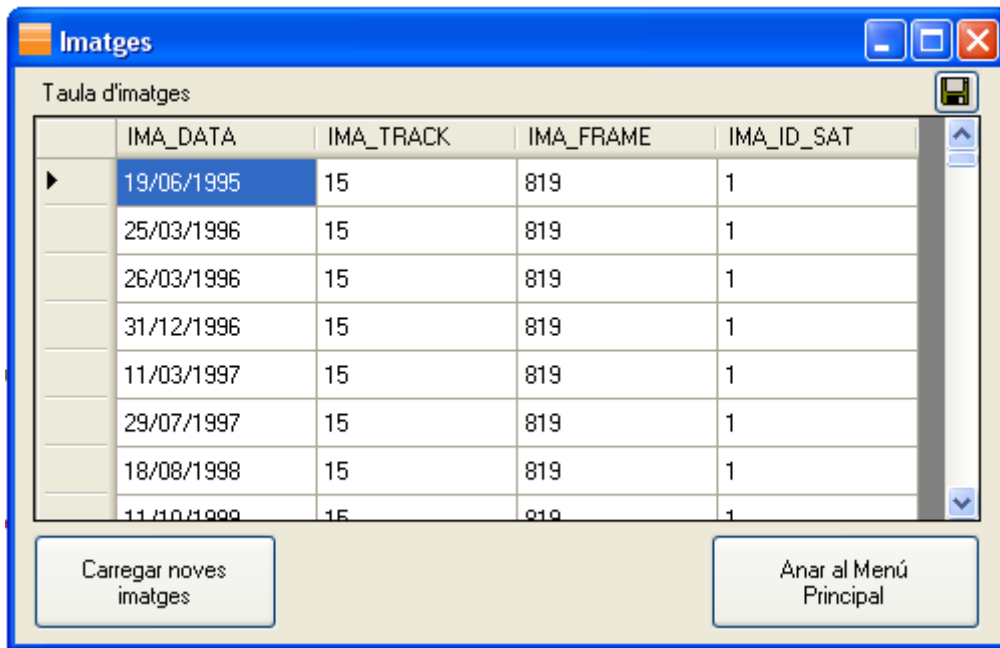


Figura 7: Formulari per carregar noves imatges

La diferència principal és que no es pot editar la taula, ja que el que s'ha de fer és carregar les noves imatges mitjançant un arxiu .txt. El codi per tal de fer-ho és el següent:

```
PrivateSub btnCarreg_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCarreg.Click
Dim taula AsNew OpenFileDialog
Dim dirubicacio AsString
    taula.Filter = "Tots els arxius txt (*.txt) | *.txt"
    taula.ShowDialog()
If Len(taula.FileName()) > 0 Then
    dirubicacio = taula.FileName()
    ruta_cargar = Trim(dirubicacio)
Else
    MsgBox("No s'ha seleccionat un arxiu Text", MsgBoxStyle.Critical)
EndIf

'ens assegurem que l'arxiu existeix
If System.IO.File.Exists(ruta_cargar) = TrueThen
'fem un try per si el fitxer es troba obert o hi ha algun error de lectura
Try
Dim stream_reader AsNew IO.StreamReader(ruta_cargar)
Catch ex AsException
    MsgBox("Error en obrir el fitxer seleccionat" + vbNewLine +
"Assegureu-vos que el fitxer no es troba obert", MsgBoxStyle.Critical, "SIGI")
Exit Sub
EndTry
```

```

'PER OMLIR EL DATAGRID AMB LES DADES DEL txt
Dim TextLine AsString = ""
Dim SplitLine() AsString
Dim objReader AsNew System.IO.StreamReader(ruta_cargar)
'omplim el datagrid amb els valors del txt
    DGridImatge.DataSource = Nothing
    DGridImatge.ColumnCount = 5
DoWhile objReader.Peek() <> -1
'fins que s'acaba l'arxiu
    TextLine = objReader.ReadLine() 'llegeix la línia
    SplitLine = Split(TextLine, ",") 'tipus de separació
    DGridImatge.Rows.Add(SplitLine) 'afegim els registres al datagrid
    DGridImatge.Sort(DGridImatge.Columns(0),
System.ComponentModel.ListSortDirection.Ascending)
Loop
Else
    MsgBox("El fitxer no existeix", MsgBoxStyle.Critical, "SIGI")
EndIf
EndSub

```

Primer obrim un OpenFileDialog on es pot veure les carpetes i els arxius .txt que hi ha a l'ordinador. Per fer això és fa un filtre que només ens deixi veure els arxius .txt. Un cop seleccionat el fitxer de noves imatges, ens assegurem que l'arxiu existeix i que no estigui obert. Tot seguit omplim el Datagrid amb les dades de l'arxiu. Ho fem llegint línia a línia i indicant quin és el separador que hi ha entre cada dada, en el nostre cas un punt i coma. D'aquesta manera afegim els registres al Datagrid.

5.3.4 Càrrega d'una nova zona i selecció de la que en volem veure les dades

Per crear i carregar una nova zona s'ha de seguir el mateix procediment que per un Satèl·lit, per tant, com que ja està explicat anteriorment, no es tornarà a repetir. La principal diferència amb el Satèl·lit, és que el formulari de Zona també s'escull de quina zona volem veure les dades de Punts, Velocitats i Mesures. Això es fa fent un clic sobre la graella del Datagrid i tot seguit al botó corresponent. A la figura següent ho podem observar:

	ZON_NOM	ZON_ORBITA	ZON_DATAIN	ZON_DATAFI
▶	Banyoles	No	20021201	20071125
	Barcelona	Sí	20030306	20080403
	Cambrils	Sí	20030114	20070508
	Cambrils	No	20030108	20060621
	Cardona	Sí	20021017	20080925
	Cardona	No	20030108	20071128
	Girona	No	20021201	20071125

Figura 8: Formulari per afegir i seleccionar zones

A més a més, a continuació s'explica el codi que s'ha utilitzat per tal de poder veure els Punts, les Velocitats i les Mesures de la Zona seleccionada.

```
PrivateSub btnPunts_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPunts.Click
```

```
Me.Hide()
```

```
Call GridFalse()
```

```
Dim oDataView AsNewDataView
```

```
oDataView.Table = frmConnexio.DataSet.Tables("Punt")
```

```
oDataView.RowFilter = "PNT_FK_ZON = "& ZonaPunt
```

```
frmPunts.DGridPunts.DataSource = oDataView
```

```
frmPunts.Show()
```

```
EndSub
```

```
PrivateSub btnVeloc_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnVeloc.Click
```

```
Me.Hide()
```

```
Call GridFalse()
```

```
Dim oDataView AsNewDataView
```

```
oDataView.Table = frmConnexio.DataSet.Tables("Veloc")
```

```
oDataView.RowFilter = "ZON_ID_ZON ="& ZonaPunt
```



```

        frmVelocitat.DGridVelo.DataSource = oDataView

frmVelocitat.Show()

EndSub

PrivateSub btnMesures_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMesures.Click

Me.Hide()

Call GridFalse()

Dim oDataView AsNewDataView

oDataView.Table = frmConnexio.DataSet.Tables("Mesura")

        oDataView.RowFilter = "ZON_ID_ZON = "& ZonaPunt

frmMesures.DGridMesura.DataSource = oDataView

frmMesures.Show()

EndSub

```

El codi que s'acaba de mostrar, ens permet veure quins són primer els Punts, tot seguit les Velocitats i finalment les Mesures de la Zona escollida al formulari de Zona.

El que es fa és un nou DataView on es carrega la taula que necessitem en cada cas, i que ja hem guardat en el nostre Dataset en el moment que ens hem connectat. Com que a l'hora de fer el Select de cada taula s'ha vinculat l'ID de Zona amb totes aquestes taules, en aquest moment podem fer un RowFilter, és a dir un filtre, al DataView que s'acaba de crear a partir del ID que s'ha guardat al clicar sobre la Zona, en una variable que es diu ZonaPunt. A continuació veiem el codi que s'ha fet servir per guardar aquesta variable:

```

PrivateSub DGridZona_RowEnter(ByVal sender AsObject, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DGridZona.RowEnter

        frmPunts.Label1.Text = DGridZona(2, e.RowIndex).Value.ToString

        frmVelocitat.Label2.Text = DGridZona(2, e.RowIndex).Value.ToString

        frmMesures.Label2.Text = DGridZona(2, e.RowIndex).Value.ToString

        ZonaPunt = DGridZona(0, e.RowIndex).Value.ToString

EndSub

```

En aquest codi es veu que es guarda l'ID de la zona, que és la columna 0, a la variable ZonaPunt quan es fa clic sobre la taula. A més a més, també s'agafa el nom de cada Zona, per tal de posar-lo en els següents formularis. El procediment és el mateix que l'anterior, però en aquest cas s'agafa la columna 2, que és on hi ha el nom de cada zona i es col·loca directament a la capçalera de cada formulari.

5.3.5 Càrrega de les dades a la taula punts

La Taula de punts té una manera molt diferent d'actualitzar la base de dades, ja que volem guardar la geometria i això requereix una escriptura SQL diferent de la emprada per les altres taules.

Així, per començar carregarem els nous punts al DataGrid, tal i com hem fet amb les altres. Les dades que necessitem és la X i la Y dels punts que ens permetran guardar la geometria a partir d'una sentència com la següent:

```
INSERT INTO SIGI_PUNT VALUES ( 1 ,SDO_GEOMETRY(2001,NULL,
SDO_POINT_TYPE(480581.500000,4664367.500000,NULL) ,NULL,NULL) );
```

Com es pot veure en la següent figura, el formulari és molt semblant als anteriors:

PNT_ID_PUNT	PNT_FK_ZON	X	Y
1	1	480581,50	4664367,50
2	1	480621,50	4664367,50
3	1	480621,50	4664327,50
4	1	480661,50	4664287,50
5	1	480701,50	4664287,50
6	1	480661,50	4664247,50
7	1	480701,50	4664247,50

Figura 9: Formulari per carregar nous punts

Quan l'usuari prem el botó de guardar, es vol que s'eliminïn a la base de dades tots els punts que fan referència a la zona que s'ha escollit i els reemplaci pels punts que s'acaben de carregar al DataGrid. Per fer aquest procediment, s'ha decidit que no es pot treballar desconnectat, ja que s'ha de fer una consulta SQL directament a la base de dades. A més a més, en el cas dels punts, s'ha d'utilitzar una seqüència com la que acabem de veure per tal de poder guardar la geometria. A continuació es mostra el codi per guardar els nous punts:

```

Dim sel As String = "DELETE FROM SIGI_PUNT WHERE PNT_FK_ZON = " & frmZona.ZonaPunt

Using connect As New OracleConnection(ora_db)

    connect.ConnectionString = ora_db

    Dim cmd As New OracleCommand(sel, connect)

    connect.Open()

    cmd.ExecuteNonQuery()

    connect.Close()

End Using

Dim conn As New OracleConnection(ora_db)

conn.ConnectionString = ora_db

conn.Open()

Dim SqlString As String = "" ' variable de la sentència

Dim objCmd As OracleCommand

'recorrer el datagrid

For i As Integer = 0 To DGridPunts.Rows.Count - 2

    Dim x As String = DGridPunts.Rows(i).Cells(1).Value

    Dim y As String = DGridPunts.Rows(i).Cells(2).Value

    SqlString = "INSERT INTO sigi_punt (PNT_FK_ZON,X,Y,PNT_GEOMPUNT)
VALUES (" & DGridPunts.Rows(i).Cells(0).Value.ToString() & "," &
DGridPunts.Rows(i).Cells(1).Value.ToString() & "," &
DGridPunts.Rows(i).Cells(2).Value.ToString() & ", MDSYS.SDO_GEOMETRY
(2001,null,MDSYS.SDO_POINT_TYPE(" & x & "," & y & ",null),null,null))"

    objCmd = New OracleCommand(SqlString, conn)

    'agreguem la sentència a la comanda

    objCmd.ExecuteNonQuery()

Next

conn.Close()

```

Primer de tot hi ha la variable *sel* que és una sentència SQL per eliminar tots els punts que siguin de la zona que hem escollit. A continuació es connecta a la base de dades i s'executa la sentència de la variable *sel*.

Un cop eliminats tots els punts d'aquesta zona, el programa carregarà els nous punts mitjançant una Insert Into. Per fer-ho primer es connectarà un altre cop a la base de dades i s'executarà un bucle que recorrerà tot el DataGrid fila a fila i omplirà, també, fila a fila la taula de punts de la base de dades. Un cop carregats tots els punts es tornarà a tancar la connexió i es continuarà treballant desconnectat. Aquest procediment, és molt semblant a l'utilitzat per emmagatzemar les noves velocitats i les noves zones.

5.3.6 Exportació de les taules

L'aplicació, també té una funció per tal de poder exportar les taules que hi ha guardades a la Base de Dades a un arxiu .csv. El disseny d'aquesta part de l'aplicació és molt semblant a la de càrrega per tal de guardar un equilibri complet entre ambdues parts.

Un exemple de la pantalla que ens permet exportar, és la de Satèl·lit. A la figura que hi ha a continuació, es pot comprovar que la línia de disseny és molt semblant a la de càrrega, tot i que el codi és molt diferent:

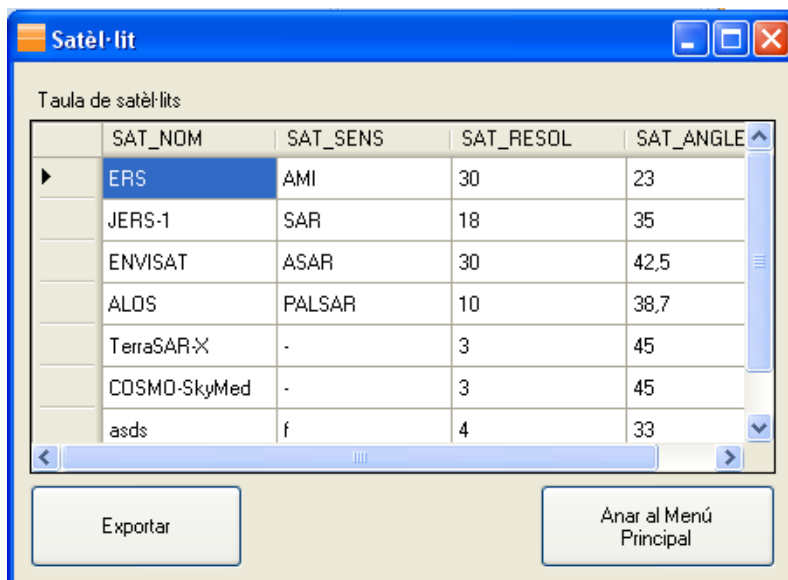


Figura 10: Formulari per exportar la taula de satèl·lit

```
PrivateSub btnExportar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExportar.Click
```

```
Const DELIMITADOR AsString = ";"
```

```
Dim arxiu_csv AsString

SaveFileDialog1.InitialDirectory = "C:"

SaveFileDialog1.Filter = "CSV (*.csv)|*.csv"

SaveFileDialog1.FileName = "SIGI_SATELLIT"

SaveFileDialog1.ShowDialog()

arxiu_csv = SaveFileDialog1.FileName

Try

Using arxiu AsStreamWriter = NewStreamWriter(arxiu_csv)

Dim linea AsString = String.Empty

With DGridSatellit

For fila AsInteger = 0 To DGridSatellit.RowCount - 1

linea = String.Empty

For col AsInteger = 0 To .Columns.Count - 1

linea = linea & .Item(col, fila).Value & DELIMITADOR

Next

Next

With arxiu

linea = linea.Remove(linea.Length - 1).ToString

arxiu.WriteLine(linea)

EndWith

Next

EndUsing

Catch ex AsException

MsgBox(ex.Message.ToString, MsgBoxStyle.Critical)

EndTry

EndSub
```

El que s'ha de fer és recórrer el DataGridView per guardar en un arxiu. Per tal de poder indicar on volem guardar aquest nou fitxer que crearem, s'ha d'obrir un Common Dialog per veure les diferents direccions on podem guardar-lo. Un cop tenim el

Common Dialog obert s'ha definit quina és la ruta predeterminada per guardar el fitxer, en el nostre cas C:/; quin és el format, *.cvs; i el nom amb que es vol guardar. Com que es tracta de la taula Satèl·lit, el nom serà Sigi_Satellit.

A continuació crearem un StreamWriter per tal d'accedir a l'arxiu i poder guardar les diferents línies. Un cop fet això, comencem a fer el bucle que ens permet anar guardant cada línia recorrent la taula fila a fila. Finalment, es controla que tot hagi funcionat bé utilitzant un MessageBox d'error.

5.4 Creació d'arxiu instal·lable

Quan tenim l'aplicació acabada, s'ha de crear un petit programa d'instal·lació per tal que tots els usuaris puguin instal·lar aquesta aplicació al seu ordinador. Els passos a seguir per tal de fer-ho són els següents:

Primer s'ha d'obrir la solució del projecte que s'ha creat.

A continuació es crea un nou projecte del tipus instal·lació, és a dir, un Setup Project.

Un cop el projecte Setup ha estat creat, s'ha d'afegir la opció Primary Output del Project Output.

Això permet que el programa mateix detecti que estem treballant amb l'OracleDataAcces.

Finalment, s'ha d'indicar on volem guardar el projecte d'instal·lació que s'acaba de crear.

6 Conclusions

El primer que s'ha de fer per començar les conclusions és analitzar si s'han assolit els propòsits que s'havien marcat al començar el projecte. En el meu cas es pot dir que a grans trets sí que he aconseguit l'objectiu principal, tot i que no ha estat una feina fàcil i per la qual he hagut de preparar-me a fons.

El primer objectiu era crear un Base de Dades per tal d'emmagatzemar les dades d'Interferometria que té l'Institut Geològic de Catalunya i a l'hora crear un aplicatiu de càrrega per actualitzar aquestes dades en el futur. Aquest s'ha assolit perfectament. El segon objectiu, era aprofundir en la programació .Net a partir del programa Visual Basic, i es pot concloure que el nivell actual de programació amb el llenguatge .Net és bastant millor que quan es vaig començar el projecte i he pogut veure les grans oportunitats que dona un programa com el Visual Basic.

El disseny d'una Base de Dades des de zero, és un autèntic repte, ja que es tracta d'una estructura bastant rígida en la qual s'han de fer el mínim de retocs possibles en el futur per tal que funcioni bé. Per tant, gràcies a l'ajuda dels meus tutors hem pogut configurar un disseny que complau a l'IGC i que a l'hora és suficientment senzill com per fer-lo en tant poc temps.

Pel que fa a l'Aplicatiu de càrrega, he de dir que té les funcionalitats que es necessiten i a part, està dissenyat de forma que sigui molt intuïtiu. En un programa com aquest una de les coses més importants és que la funcionalitat estigui per davant de tot i crec que s'ha aconseguit.

La realització d'un projecte d'aquestes característiques ha estat un autèntic repte, ja que mai havia fet una feina de recerca com la que he hagut de fer a l'hora d'elaborar-lo. He hagut d'aprendre com funciona tota una Base de Dades, un programa amb tantes possibilitats com l'Oracle i una llenguatge de programació amb moltes possibilitats com és .NET.

Per acabar aquestes conclusions, s'ha de dir que per tal d'acabar el projecte estaria molt bé fer un visor on poder visualitzar totes les dades que hi ha a la Base en diferents mapes temàtics. Malauradament, no hi ha hagut suficientment temps per fer-lo. També està clar que l'Aplicatiu es pot millorar, i això vol dir que en el futur segur que s'hi poden afegir millores i més aplicacions.

7 Referències

Referències bibliogràfiques:

Pérez López, César (2002). *Oracle 9i: administració y anàlisis de bases de datos*. Paracuellos de Jarama: RA – MA, 718p..

Gabillaud, Jérôme (2005). *Oracle 10g: SQL, PL/SQL, SQL *Plus*. Ediciones ENI, 496p..

Gómez Jiménez, Enrique (2002). *Visual Basic.NET: Microsoft*, Anaya Multimedia, Madrid, 384p..

Halvorson, Michael (2009). *Visual Basic 2008 paso a paso*. Anaya Multimedia, Madrid, 655p..

Groussard, Thierry (2009). *Visual Basic.NET (VB.NET) – Programe con Visual Studio 2008*. Ediciones ENI, 502p..

MacKenzie, Duncan; Sharkey, Kent (2004). *Aprendiendo Visual Basic.Net en 21 Lecciones Avanzadas*. Pearson Educación, 660p..

Referències electròniques:

Pàgina d'Oracle

<http://www.oracle.com/pls/db112/homepage>

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

Pàgina de Microsoft

<http://msdn.microsoft.com/en-us/vstudio/bb984878.aspx>

<http://msdn.microsoft.com/es-es/library/ms235317.aspx>

Altres recursos

<http://www.elguille.info/default.aspx>

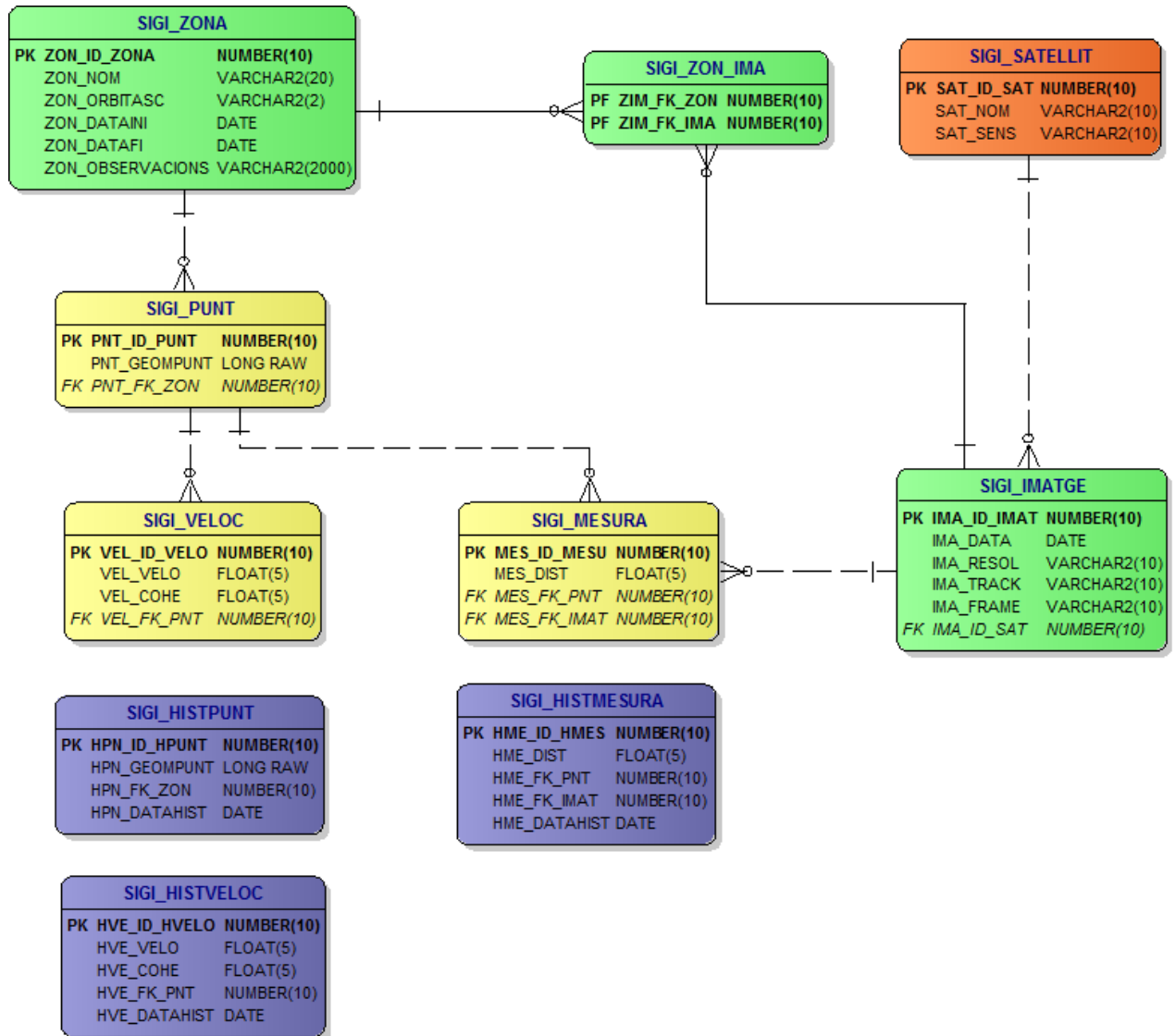
8 Annexes

8.1 Model lògic de la Base de dades

SISTEMA D'INFORMACIÓ GEOGRÀFICA D'INTERFEROMETRIA

DatabaseType	Oracle 10g
Date	11/11/2010
Trial version	This report has been generated with a TRIAL version.

ER diagram



List of entities

Name	Primary key constraint name	Number of columns	Comment
SIGI_HISTMESURA	PK_SIGI_HISTMESURA	5	Conté els registres històrics de les mesures referents a cada punt
SIGI_HISTPUNT	PK_SIGI_HISTPUNT	4	Conté els registres històrics dels punts referents a cada zona
SIGI_HISTVELOC	PK_SIGI_HISTVELOC	5	Conté els registres històrics de la velocitat a la que es mou cada punt i la coherència d'aquesta velocitat, la informació d'aquesta taula s'obté a partir de les dades de la taula de mesures associades a cada punt
SIGI_IMATGE	PK_SIGI_IMATGE	6	És un recull de totes les imatges de satèl·lit que hi ha
SIGI_MESURA	PK_SIGI_MESURA	4	Conté les mesures referents a cada punt
SIGI_PUNT	PK_SIGI_PUNT	3	Aquesta taula tindrà tots els punts que s'estudien a Catalunya.
SIGI_SATELLIT	PK_SIGI_SATELLIT	3	Taula de les propietats del satèl·lit
SIGI_VELOC	PK_SIGI_VELOC	4	Ens permet veure la velocitat a la que es mou cada punt i la coherència d'aquesta velocitat, la informació d'aquesta taula s'obté a partir de les dades de la taula de mesures associades a cada punt
SIGI_ZONA		6	Proporciona informació sobre les zones d'estudi
SIGI_ZON_IMA	PK_SIGI_ZON_IMA	2	Permet la implementació de la relació N a N entre imatges i zones. Una imatge ens pot donar

			informació de moltes zones i una zona pot estar informada per més d'una imatge
--	--	--	--

Entity details

Entity: SIGI_HISTMESURA	
Primary key constraint name	PK_SIGI_HISTMESURA
Comment	Conté els registres històrics de les mesures referents a cada punt
Table options	

Attributes:

Column name	Primary key	Data type	Not NULL	Comment
HME_ID_HMES	Yes	NUMBER(10)	Yes	Identificador mesures
HME_DIST	No	FLOAT(5)	Yes	Valor d'interferometria, mesura
HME_FK_PNT	No	NUMBER(10)	Yes	És la FK del punt, que ens diu de quin punt és cada mesura
HME_FK_IMAT	No	NUMBER(10)	Yes	És la FK de imatge, que ens diu quina imatge té cada mesura
HME_DATAHIST	No	DATE	No	Data d'inserció del registre a la taula d'històrics. Permetrà obtenir la vigència temporal de la mesura, que serà la diferència entre HME_DATAHIST I ZON_DATAINI

Constraints:

Name	Type	Level	Constraint
NN_SIGI_HISTMESURA_HME_DIST	Not Null	Column Constraint	NOT NULL (HME_DIST)
NN_SIGI_HISTMESURA_HME_FK_IMAT	Not Null	Column Constraint	NOT NULL (HME_FK_IMAT)
NN_SIGI_HISTMESURA_HME_FK_PNT	Not Null	Column Constraint	NOT NULL (HME_FK_PNT)
NN_SIGI_HISTMESURA_HME_ID_HMES	Not Null	Column Constraint	NOT NULL (HME_ID_HMES)
PK_SIGI_HISTMESURA	Primary Key	Table Constraint	PRIMARY KEY (HME_ID_HMES)

Triggers:

Trigger name	Code

SIGI_HME_TR_SEQ	<pre> CREATE OR REPLACE TRIGGER SIGI_HME_TR_SEQ BEFORE INSERT ON SIGI_HISTMESURA FOR EACH ROW BEGIN IF :NEW.HME_ID_HMES IS NULL THEN SELECT SIGI_HME_SEQ.NEXTVAL INTO :NEW.HME_ID_HMES FROM DUAL END IF; END;</pre>
-----------------	---

Entity: SIGI_HISTPUNT	
Primary key constraint name	PK_SIGI_HISTPUNT
Comment	Conté els registres històrics dels punts referents a cada zona
Table options	

Attributes:

Column name	Primary key	Data type	Not NULL	Comment
HPN_ID_HPUNT	Yes	NUMBER(10)	Yes	Identificador únic de cada punt
HPN_GEOMPUNT	No	LONG RAW	Yes	Permet l'emagatzematge de la geometria de cada punt
HPN_FK_ZON	No	NUMBER(10)	Yes	FK de la zona, és a dir que cada punt mostra l'identificador de la zona a la qual pertany
HPN_DATAHIST	No	DATE	No	Data d'inserció del registre a la taula d'històrics. Permetrà obtenir la vigència temporal dels punts, que serà la diferència entre HPN_DATAHIST i ZON_DATAINI

Constraints:

Name	Type	Level	Constraint
NN_SIGI_HISTPUNT_HP_NFK_ZON	Not Null	Column Constraint	NOT NULL (HPN_FK_ZON)
NN_SIGI_HISTPUNT_HP_NGEOMPUNT	Not Null	Column Constraint	NOT NULL (HPN_GEOMPUNT)
NN_SIGI_HISTPUNT_HP_NID_HPUNT	Not Null	Column Constraint	NOT NULL (HPN_ID_HPUNT)

PK_SIGI_HISTPUNT	Primary Key	Table Constraint	PRIMARY KEY (HPN_ID_HPUNT)
------------------	-------------	------------------	----------------------------

Triggers:

Trigger name	Code
SIGI_HPNT_TR_SEQ	<pre> CREATE OR REPLACE TRIGGER SIGI_HPNT_TR_SEQ BEFORE INSERT ON SIGI_HISTPUNT FOR EACH ROW BEGIN IF :NEW.HPN_ID_HPUNT IS NULL THEN SELECT SIGI_HPNT_SEQ.NEXTVAL INTO :NEW.HPN_ID_HPUNT FROM DUAL END IF; END;</pre>

Entity: SIGI_HISTVELOC	
Primary key constraint name	PK_SIGI_HISTVELOC
Comment	Conté els registres històrics de la velocitat a la que es mou cada punt i la coherència d'aquesta velocitat, la informació d'aquesta taula s'obté a partir de les dades de la taula de mesures associades a cada punt
Table options	

Attributes:

Column name	Primary key	Data type	Not NULL	Comment
HVE_ID_HVELO	Yes	NUMBER(10)	Yes	Identificador de velocitat
HVE_VELO	No	FLOAT(5)	Yes	Velocitat del moviment per cada punt
HVE_COHE	No	FLOAT(5)	Yes	Coherència de les dades de velocitat
HVE_FK_PNT	No	NUMBER(10)	Yes	Punt al que es refereixen les dades de velocitat
HVE_DATAHIST	No	DATE	No	Data d'inserció del registre a la taula d'històrics. Serà la data en la que s'elimina el registre de la taula veloc. Ens permetrà conèixer la vigència temporal de la informació de velocitat, que serà la diferència entre HVE_DATAHIST i ZON_DATAINI

Constraints:

Name	Type	Level	Constraint
NN_SIGI_HISTVELOC_HVE_COHE	Not Null	Column Constraint	NOT NULL (HVE_COHE)
NN_SIGI_HISTVELOC_HVE_FK_PNT	Not Null	Column Constraint	NOT NULL (HVE_FK_PNT)
NN_SIGI_HISTVELOC_HVE_ID_HVELO	Not Null	Column Constraint	NOT NULL (HVE_ID_HVELO)
NN_SIGI_HISTVELOC_HVE_VELO	Not Null	Column Constraint	NOT NULL (HVE_VELO)
PK_SIGI_HISTVELOC	Primary Key	Table Constraint	PRIMARY KEY (HVE_ID_HVELO)

Triggers:

Trigger name	Code
SIGI_HVE_TR_SEQ	<pre> CREATE OR REPLACE TRIGGER SIGI_HVE_TR_SEQ BEFORE INSERT ON SIGI_HISTVELOC FOR EACH ROW BEGIN IF :NEW.HVE_ID_HVELO IS NULL THEN SELECT SIGI_HVE_SEQ.NEXTVAL INTO :NEW.HVE_ID_HVELO FROM DUAL END IF; END;</pre>

Entity: SIGI_IMATGE	
Primary key constraint name	PK_SIGI_IMATGE
Comment	És un recull de totes les imatges de satèl·lit que hi ha
Table options	

Attributes:

Column name	Primary key	Data type	Not NULL	Comment
IMA_ID_IMAT	Yes	NUMBER(10)	Yes	Codi identificador de la imatge
IMA_DATA	No	DATE	Yes	Data de cada imatge
IMA_RESOL	No	VARCHAR2(10)	Yes	Resolució de cada imatge
IMA_TRACK	No	VARCHAR2(10)	Yes	Track de cada orbita
IMA_FRAME	No	VARCHAR2(10)	Yes	Frame de cada òrbita
IMA_ID_SAT	No	NUMBER(10)	Yes	Identificador de satèl·lit

Relationships:

Name	Relationship type	Parent	Child	Cardinality
SIGI_IMATGE_SIGI_MESURA	Non Identifying	SIGI_IMATGE	SIGI_MESURA	Zero Or More
SIGI_IMATGE_SIGI_ZON_IMA	Identifying	SIGI_IMATGE	SIGI_ZON_IMA	Zero Or More
SIGI_SATELLIT_SIGI_IMATGE	Non Identifying	SIGI_SATELLIT	SIGI_IMATGE	Zero Or More

Indexes:

Name	Index columns	Index type
IDX_SIGI_IMATGE_1_FK	IMA_ID_SAT	

Constraints:

Name	Type	Level	Constraint
NN_SIGI_IMATGE_IMA_DATA	Not Null	Column Constraint	NOT NULL (IMA_DATA)
NN_SIGI_IMATGE_IMA_FRAME	Not Null	Column Constraint	NOT NULL (IMA_FRAME)
NN_SIGI_IMATGE_IMA_ID_IMAT	Not Null	Column Constraint	NOT NULL (IMA_ID_IMAT)
NN_SIGI_IMATGE_IMA_ID_SAT	Not Null	Column Constraint	NOT NULL (IMA_ID_SAT)
NN_SIGI_IMATGE_IMA_RESOL	Not Null	Column Constraint	NOT NULL (IMA_RESOL)
NN_SIGI_IMATGE_IMA_TRACK	Not Null	Column Constraint	NOT NULL (IMA_TRACK)
PK_SIGI_IMATGE	Primary Key	Table Constraint	PRIMARY KEY (IMA_ID_IMAT)
SIGI_SATELLIT_SIGI_IMATGE	Foreign Key	Table Constraint	FOREIGN KEY (IMA_ID_SAT) REFERENCES SIGI_SATELLIT(SAT_ID_SAT)

Triggers:

Trigger name	Code
SIGI_IMA_TR_SEQ	<pre>CREATE OR REPLACE TRIGGER SIGI_IMA_TR_SEQ BEFORE INSERT ON SIGI_IMATGE FOR EACH ROW BEGIN IF :NEW.IMA_ID_IMAT IS NULL THEN SELECT SIGI_IMA_SEQ.NEXTVAL INTO :NEW.IMA_ID_IMAT FROM DUAL END IF; END;</pre>

Entity: SIGI_MESURA	
Primary key constraint name	PK_SIGI_MESURA
Comment	Conté les mesures referents a cada punt
Table options	

Attributes:

Column name	Primary key	Data type	Not NULL	Comment
MES_ID_MESU	Yes	NUMBER(10)	Yes	Identificador mesures
MES_DIST	No	FLOAT(5)	Yes	Valor d'interferometria, mesura
MES_FK_PNT	No	NUMBER(10)	Yes	És la FK del punt, que ens diu de quin punt és cada mesura
MES_FK_IMAT	No	NUMBER(10)	Yes	És la FK de imatge, que ens diu quina imatge té cada mesura

Relationships:

Name	Relationship type	Parent	Child	Cardinality
SIGI_IMATGE_SIGI_MESURA	Non Identifying	SIGI_IMATGE	SIGI_MESURA	Zero Or More
SIGI_PUNT_SIGI_MESURA	Non Identifying	SIGI_PUNT	SIGI_MESURA	Zero Or More

Indexes:

Name	Index columns	Index type
IDX_SIGI_MESURA_1_FK	MES_FK_PNT	
IDX_SIGI_MESURA_2_FK	MES_FK_IMAT	

Constraints:

Name	Type	Level	Constraint
NN_SIGI_MESURA_MES_DIST	Not Null	Column Constraint	NOT NULL (MES_DIST)
NN_SIGI_MESURA_MES_FK_IMAT	Not Null	Column Constraint	NOT NULL (MES_FK_IMAT)
NN_SIGI_MESURA_MES_FK_PNT	Not Null	Column Constraint	NOT NULL (MES_FK_PNT)
NN_SIGI_MESURA_MES_ID_MES	Not	Column	NOT NULL (MES_ID_MESU)

U	Null	Constrain t	
PK_SIGI_MESURA	Primary Key	Table Constrain t	PRIMARY KEY (MES_ID_MESU)
SIGI_IMATGE_SIGI_MESURA	Foreign Key	Table Constrain t	FOREIGN KEY (MES_FK_IMAT) REFERENCES SIGI_IMATGE(IMA_ID_IMAT)
SIGI_PUNT_SIGI_MESURA	Foreign Key	Table Constrain t	FOREIGN KEY (MES_FK_PNT) REFERENCES SIGI_PUNT(PNT_ID_PUNT)

Triggers:

Trigger name	Code
SIGI_MES_TR_DELETE	create or replace TRIGGER SIGI_MES_TR_DELETE BEFORE DELETE ON SIGI_MESURA FOR EACH ROW BEGIN INSERT INTO SIGI_HISTPUNT(HPN_ID_HPUNT, HPN_GEOMPUNT, HPN_FK_ZON, HPN_DATAHIST) VALUES (:OLD.PNT_ID_PUNT, :OLD.PNT_GEOMPUNT, :OLD.PNT_FK_ZON, SYSDATE); END;
SIGI_MES_TR_SEQ	CREATE OR REPLACE TRIGGER SIGI_MES_TR_SEQ BEFORE INSERT ON SIGI_MESURA FOR EACH ROW BEGIN IF :NEW.MES_ID_MESU IS NULL THEN SELECT SIGI_MES_SEQ.NEXTVAL INTO :NEW.MES_ID_MESU FROM DUAL END IF; END;

Entity: SIGI_PUNT	
Primary key constraint name	PK_SIGI_PUNT
Comment	Aquesta taula tindrà tots els punts que s'estudien a Catalunya.
Table options	

Attributes:

Column name	Primary	Data type	Not	Comment
-------------	---------	-----------	-----	---------

	key		NULL	
PNT_ID_PUNT	Yes	NUMBER(10)	Yes	Identificador únic de cada punt
PNT_GEOMPUNT	No	LONG RAW	Yes	Permet l'emagatzematge de la geometria de cada punt
PNT_FK_ZON	No	NUMBER(10)	Yes	FK de la zona, és a dir que cada punt mostra l'identificador de la zona a la qual pertany

Relationships:

Name	Relationship type	Parent	Child	Cardinality
SIGI_PUNT_SIGI_MESURA	Non Identifying	SIGI_PUNT	SIGI_MESURA	Zero Or More
SIGI_PUNT_SIGI_VELOC	Non Identifying	SIGI_PUNT	SIGI_VELOC	Zero Or More
SIGI_ZONA_SIGI_PUNT	Non Identifying	SIGI_ZONA	SIGI_PUNT	Zero Or More

Indexes:

Name	Index columns	Index type
IDX_SIGI_PUNT_1_FK	PNT_FK_ZON	

Constraints:

Name	Type	Level	Constraint
NN_SIGI_PUNT_PNT_FK_ZON	Not Null	Column Constraint	NOT NULL (PNT_FK_ZON)
NN_SIGI_PUNT_PNT_GEOMPUNT	Not Null	Column Constraint	NOT NULL (PNT_GEOMPUNT)
NN_SIGI_PUNT_PNT_ID_PUNT	Not Null	Column Constraint	NOT NULL (PNT_ID_PUNT)
PK_SIGI_PUNT	Primary Key	Table Constraint	PRIMARY KEY (PNT_ID_PUNT)
SIGI_ZONA_SIGI_PUNT	Foreign Key	Table Constraint	FOREIGN KEY (PNT_FK_ZON) REFERENCES SIGI_ZONA(ZON_ID_ZONA)

Triggers:

Trigger name	Code
SIGI_PNT_TR_DELETE	create or replace TRIGGER SIGI_HPN_TR_DELETE BEFORE DELETE ON SIGI_PUNT FOR EACH ROW BEGIN INSERT INTO SIGI_HISTPUNT(HPN_ID_HPUNT,

	HPN_GEOMPUNT, HPN_FK_ZON, HPN_DATAHIST) VALUES (:OLD.PNT_ID_PUNT, :OLD.PNT_GEOMPUNT, :OLD.PNT_FK_ZON), SYSDATE; END;
SIGI_PNT_TR_SEQ	CREATE OR REPLACE TRIGGER SIGI_PNT_TR_SEQ BEFORE INSERT ON SIGI_PUNT FOR EACH ROW BEGIN IF :NEW.PNT_ID_PUNT IS NULL THEN SELECT SIGI_PNT_SEQ.NEXTVAL INTO :NEW.PNT_ID_PUNT FROM DUAL END IF; END;

Entity: SIGI_SATELLIT	
Primary key constraint name	PK_SIGI_SATELLIT
Comment	Taula de les propietats del satèl·lit
Table options	

Attributes:

Column name	Primary key	Data type	Not NULL	Comment
SAT_ID_SAT	Yes	NUMBER(10)	Yes	Identificador de satèl·lit
SAT_NOM	No	VARCHAR2(10)	Yes	Nom del satèl·lit
SAT_SENS	No	VARCHAR2(10)	Yes	Sensor del satèl·lit

Relationships:

Name	Relationship type	Parent	Child	Cardinality
SIGI_SATELLIT_SIGI_IMATGE	Non Identifying	SIGI_SATELLIT	SIGI_IMATGE	Zero Or More

Constraints:

Name	Type	Level	Constraint
NN_SIGI_SATELLIT_SAT_ID_SAT	Not Null	Column Constraint	NOT NULL (SAT_ID_SAT)
NN_SIGI_SATELLIT_SAT_NOM	Not Null	Column Constraint	NOT NULL (SAT_NOM)
NN_SIGI_SATELLIT_SAT_SENS	Not Null	Column Constraint	NOT NULL (SAT_SENS)
PK_SIGI_SATELLIT	Primary Key	Table Constraint	PRIMARY KEY (SAT_ID_SAT)

Triggers:

Trigger name	Code
SIGI_SAT_TR_SEQ	<pre> CREATE OR REPLACE TRIGGER SIGI_SAT_TR_SEQ BEFORE INSERT ON SIGI_SATELLIT FOR EACH ROW BEGIN IF :NEW.SAT_ID_SAT IS NULL THEN SELECT SIGI_SAT_SEQ.NEXTVAL INTO :NEW.SAT_ID_SAT FROM DUAL END IF; END;</pre>

Entity: SIGI_VELOC	
Primary key constraint name	PK_SIGI_VELOC
Comment	Ens permet veure la velocitat a la que es mou cada punt i la coherència d'aquesta velocitat, la informació d'aquesta taula s'obté a partir de les dades de la taula de mesures associades a cada punt
Table options	

Attributes:

Column name	Primary key	Data type	Not NULL	Comment
VEL_ID_VELO	Yes	NUMBER(10)	Yes	Identificador de velocitat
VEL_VELO	No	FLOAT(5)	Yes	Velocitat del moviment per cada punt
VEL_COHE	No	FLOAT(5)	Yes	Coherència de les dades de velocitat
VEL_FK_PNT	No	NUMBER(10)	Yes	Punt al que es refereixen les dades de velocitat

Relationships:

Name	Relationship type	Parent	Child	Cardinality
SIGI_PUNT_SIGI_VELOC	Non Identifying	SIGI_PUNT	SIGI_VELOC	Zero Or More

Indexes:

Name	Index columns	Index type
IDX_SIGI_VELOC_1_FK	VEL_FK_PNT	

Constraints:

Name	Type	Level	Constraint
------	------	-------	------------

NN_SIGI_VELOC_VEL_COHE	Not Null	Column Constraint	NOT NULL (VEL_COHE)
NN_SIGI_VELOC_VEL_FK_PNT	Not Null	Column Constraint	NOT NULL (VEL_FK_PNT)
NN_SIGI_VELOC_VEL_ID_VELO	Not Null	Column Constraint	NOT NULL (VEL_ID_VELO)
NN_SIGI_VELOC_VEL_VELO	Not Null	Column Constraint	NOT NULL (VEL_VELO)
PK_SIGI_VELOC	Primary Key	Table Constraint	PRIMARY KEY (VEL_ID_VELO)
SIGI_PUNT_SIGI_VELOC	Foreign Key	Table Constraint	FOREIGN KEY (VEL_FK_PNT) REFERENCES SIGI_PUNT(PNT_ID_PUNT)

Triggers:

Trigger name	Code
SIGI_VEL_TR_DELETE	<pre> create or replace TRIGGER SIGI_VEL_TR_DELETE BEFORE DELETE ON SIGI_VELOC FOR EACH ROW BEGIN INSERT INTO SIGI_HISTPUNT(HPN_ID_HPUNT, HPN_GEOMPUNT, HPN_FK_ZON, HPN_DATAHIST) VALUES (:OLD.PNT_ID_PUNT, :OLD.PNT_GEOMPUNT, :OLD.PNT_FK_ZON, SYSDATE); END;</pre>
SIGI_VEL_TR_SEQ	<pre> CREATE OR REPLACE TRIGGER SIGI_VEL_TR_SEQ BEFORE INSERT ON SIGI_VELOC FOR EACH ROW BEGIN IF :NEW.VEL_ID_VELO IS NULL THEN SELECT SIGI_VEL_SEQ.NEXTVAL INTO :NEW.VEL_ID_VELO FROM DUAL END IF; END;</pre>

Entity: SIGI_ZONA	
Primary key constraint name	
Comment	Proporciona informació sobre les zones d'estudi
Table options	

Attributes:

Column name	Primary	Data type	Not	Comment
-------------	---------	-----------	-----	---------

	key		NULL	
ZON_ID_ZONA	Yes	NUMBER(10)	Yes	Identificador únic de cada zona
ZON_NOM	No	VARCHAR2(20)	Yes	Nom de cada zona
ZON_ORBITASC	No	VARCHAR2(2)	Yes	Indica si la òrbita és ascendent (Sí, No)
ZON_DATAINI	No	DATE	Yes	Data d'inici de les dades d'aquesta zona
ZON_DATAFI	No	DATE	Yes	Data de l'última imatge de la zona
ZON_OBSERVACIONS	No	VARCHAR2(2000)	No	Proporciona informació que no és recollida en cap altra atribut de la taula

Relationships:

Name	Relationship type	Parent	Child	Cardinality
SIGI_ZONA_SIGI_PUNT	Non Identifying	SIGI_ZONA	SIGI_PUNT	Zero Or More
SIGI_ZONA_SIGI_ZON_IMA	Identifying	SIGI_ZONA	SIGI_ZON_IMA	Zero Or More

Constraints:

Name	Type	Level	Constraint
	Primary Key	Table Constraint	PRIMARY KEY (ZON_ID_ZONA)
NN_SIGI_ZONA_ZON_DATAFI	Not Null	Column Constraint	NOT NULL (ZON_DATAFI)
NN_SIGI_ZONA_ZON_DATAINI	Not Null	Column Constraint	NOT NULL (ZON_DATAINI)
NN_SIGI_ZONA_ZON_ID_ZONA	Not Null	Column Constraint	NOT NULL (ZON_ID_ZONA)
NN_SIGI_ZONA_ZON_NOM	Not Null	Column Constraint	NOT NULL (ZON_NOM)
NN_SIGI_ZONA_ZON_ORBITASC	Not Null	Column Constraint	NOT NULL (ZON_ORBITASC)

Triggers:

Trigger name	Code
SIGI_ZON_TR_BUUPDATE	CREATE OR REPLACE TRIGGER SIGI_ZON_TR_BUUPDATE BEFORE UPDATE ON SIGI_ZONA FOR EACH ROW BEGIN IF :NEW.ZON_DATAINI <> :OLD.ZON_DATAINI OR

	<pre> :NEW.ZON_DATAFI <> :OLD.ZON_DATAFI THEN DELETE * FROM SIGI_PUNTS WHERE PNT_FK_ZON = :OLD:ZON_ID_ZONA; END IF; END; </pre>
SIGI_ZON_TR_SEQ	<pre> CREATE OR REPLACE TRIGGER SIGI_ZON_TR_SEQ BEFORE INSERT ON SIGI_ZONA FOR EACH ROW BEGIN IF :NEW.ZON_ID_ZONA IS NULL THEN SELECT SIGI_ZON_SEQ.NEXTVAL INTO :NEW.ZON_ID_ZONA FROM DUAL END IF; END; </pre>

Entity: SIGI_ZON_IMA	
Primary key constraint name	PK_SIGI_ZON_IMA
Comment	Permet la implementació de la relació N a N entre imatges i zones. Una imatge ens pot donar informació de moltes zones i una zona pot estar informada per més d'una imatge
Table options	

Attributes:

Column name	Primary key	Data type	Not NULL	Comment
ZIM_FK_ZON	Yes	NUMBER(10)	Yes	FK de la zona
ZIM_FK_IMA	Yes	NUMBER(10)	Yes	Fk de la imatge

Relationships:

Name	Relationship type	Parent	Child	Cardinality
SIGI_IMATGE_SIGI_ZON_IMA	Identifying	SIGI_IMATGE	SIGI_ZON_IMA	Zero Or More
SIGI_ZONA_SIGI_ZON_IMA	Identifying	SIGI_ZONA	SIGI_ZON_IMA	Zero Or More

Indexes:

Name	Index columns	Index type
IDX_SIGI_ZON_IMA_1_FK	ZIM_FK_ZON	
IDX_SIGI_ZON_IMA_2_FK	ZIM_FK_IMA	

Constraints:

Name	Type	Level	Constraint
NN_SIGI_ZON_IMA_ZIM_FK_IMA	Not Null	Column Constraint	NOT NULL (ZIM_FK_IMA)
NN_SIGI_ZON_IMA_ZIM_FK_ZON	Not Null	Column Constraint	NOT NULL (ZIM_FK_ZON)
PK_SIGI_ZON_IMA	Primary Key	Table Constraint	PRIMARY KEY (ZIM_FK_ZON, ZIM_FK_IMA)
SIGI_IMATGE_SIGI_ZON_IMA	Foreign Key	Table Constraint	FOREIGN KEY (ZIM_FK_IMA) REFERENCES SIGI_IMATGE(IMA_ID_IMAT)
SIGI_ZONA_SIGI_ZON_IMA	Foreign Key	Table Constraint	FOREIGN KEY (ZIM_FK_ZON) REFERENCES SIGI_ZONA(ZON_ID_ZONA)

List of attributes

Column name	Table	Key	Data type	Not NULL
HME_DATAHIST	SIGI_HISTMESURA		DATE	No
HME_DIST	SIGI_HISTMESURA		FLOAT(5)	Yes
HME_FK_IMAT	SIGI_HISTMESURA		NUMBER(10)	Yes
HME_FK_PNT	SIGI_HISTMESURA		NUMBER(10)	Yes
HME_ID_HMES	SIGI_HISTMESURA	PK	NUMBER(10)	Yes
HPN_DATAHIST	SIGI_HISTPUNT		DATE	No
HPN_FK_ZON	SIGI_HISTPUNT		NUMBER(10)	Yes
HPN_GEOMPUNT	SIGI_HISTPUNT		LONG RAW	Yes
HPN_ID_HPUNT	SIGI_HISTPUNT	PK	NUMBER(10)	Yes
HVE_COHE	SIGI_HISTVELOC		FLOAT(5)	Yes
HVE_DATAHIST	SIGI_HISTVELOC		DATE	No
HVE_FK_PNT	SIGI_HISTVELOC		NUMBER(10)	Yes
HVE_ID_HVELO	SIGI_HISTVELOC	PK	NUMBER(10)	Yes
HVE_VELO	SIGI_HISTVELOC		FLOAT(5)	Yes
IMA_DATA	SIGI_IMATGE		DATE	Yes
IMA_FRAME	SIGI_IMATGE		VARCHAR2(10)	Yes
IMA_ID_IMAT	SIGI_IMATGE	PK	NUMBER(10)	Yes
IMA_ID_SAT	SIGI_IMATGE	FK	NUMBER(10)	Yes
IMA_RESOL	SIGI_IMATGE		VARCHAR2(10)	Yes
IMA_TRACK	SIGI_IMATGE		VARCHAR2(10)	Yes
MES_DIST	SIGI_MESURA		FLOAT(5)	Yes
MES_FK_IMAT	SIGI_MESURA	FK	NUMBER(10)	Yes
MES_FK_PNT	SIGI_MESURA	FK	NUMBER(10)	Yes
MES_ID_MESU	SIGI_MESURA	PK	NUMBER(10)	Yes
PNT_FK_ZON	SIGI_PUNT	FK	NUMBER(10)	Yes
PNT_GEOMPUNT	SIGI_PUNT		LONG RAW	Yes
PNT_ID_PUNT	SIGI_PUNT	PK	NUMBER(10)	Yes
SAT_ID_SAT	SIGI_SATELLIT	PK	NUMBER(10)	Yes
SAT_NOM	SIGI_SATELLIT		VARCHAR2(10)	Yes
SAT_SENS	SIGI_SATELLIT		VARCHAR2(10)	Yes
VEL_COHE	SIGI_VELOC		FLOAT(5)	Yes
VEL_FK_PNT	SIGI_VELOC	FK	NUMBER(10)	Yes
VEL_ID_VELO	SIGI_VELOC	PK	NUMBER(10)	Yes
VEL_VELO	SIGI_VELOC		FLOAT(5)	Yes
ZIM_FK_IMA	SIGI_ZON_IMA	PK, FK	NUMBER(10)	Yes
ZIM_FK_ZON	SIGI_ZON_IMA	PK, FK	NUMBER(10)	Yes
ZON_DATAFI	SIGI_ZONA		DATE	Yes
ZON_DATAINI	SIGI_ZONA		DATE	Yes
ZON_ID_ZONA	SIGI_ZONA	PK	NUMBER(10)	Yes
ZON_NOM	SIGI_ZONA		VARCHAR2(20)	Yes

ZON_OBSERVACIONS	SIGI_ZONA		VARCHAR2(2000)	No
ZON_ORBITASC	SIGI_ZONA		VARCHAR2(2)	Yes

Attribute details

Attribute: HME_DATAHIST	
Table	SIGI_HISTMESURA
Comment	Data d'inserció del registre a la taula d'històrics. Permetrà obtenir la vigència temporal de la mesura, que serà la diferència entre HME_DATAHIST I ZON_DATAINI
Primary key	No
Data type	DATE
Not NULL	No
Default	

Attribute: HME_DIST	
Table	SIGI_HISTMESURA
Comment	Valor d'interferometria, mesura
Primary key	No
Data type	FLOAT(5)
Not NULL	Yes
Default	

Attribute: HME_FK_IMAT	
Table	SIGI_HISTMESURA
Comment	És la FK de imatge, que ens diu quina imatge té cada mesura
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: HME_FK_PNT	
Table	SIGI_HISTMESURA
Comment	És la FK del punt, que ens diu de quin punt és cada mesura
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: HME_ID_HMES	
Table	SIGI_HISTMESURA
Comment	Identificador mesures
Primary key	Yes

Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: HPN_DATAHIST	
Table	SIGI_HISTPUNT
Comment	Data d'inserció del registre a la taula d'històrics. Permetrà obtenir la vigència temporal dels punts, que serà la diferència entre HPN_DATAHIST i ZON_DATAINI
Primary key	No
Data type	DATE
Not NULL	No
Default	

Attribute: HPN_FK_ZON	
Table	SIGI_HISTPUNT
Comment	FK de la zona, és a dir que cada punt mostra l'identificador de la zona a la qual pertany
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: HPN_GEOMPUNT	
Table	SIGI_HISTPUNT
Comment	Permet l'emagatzematge de la geometria de cada punt
Primary key	No
Data type	LONG RAW
Not NULL	Yes
Default	

Attribute: HPN_ID_HPUNT	
Table	SIGI_HISTPUNT
Comment	Identificador únic de cada punt
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: HVE_COHE	
----------------------------	--

Table	SIGI_HISTVELOC
Comment	Coherència de les dades de velocitat
Primary key	No
Data type	FLOAT(5)
Not NULL	Yes
Default	

Attribute: HVE_DATAHIST	
Table	SIGI_HISTVELOC
Comment	Data d'inserció del registre a la taula d'històrics. Serà la data en la què s'elimina el registre de la taula veloc. Ens permetrà conèixer la vigència temporal de la informació de velocitat, que serà la diferència entre HVE_DATAHIST i ZON_DATAINI
Primary key	No
Data type	DATE
Not NULL	No
Default	

Attribute: HVE_FK_PNT	
Table	SIGI_HISTVELOC
Comment	Punt al que es refereixen les dades de velocitat
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: HVE_ID_HVELO	
Table	SIGI_HISTVELOC
Comment	Identificador de velocitat
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: HVE_VELO	
Table	SIGI_HISTVELOC
Comment	Velocitat del moviment per cada punt
Primary key	No
Data type	FLOAT(5)
Not NULL	Yes
Default	

Attribute: IMA_DATA	
Table	SIGI_IMATGE
Comment	Data de cada imatge
Primary key	No
Data type	DATE
Not NULL	Yes
Default	

Attribute: IMA_FRAME	
Table	SIGI_IMATGE
Comment	Frame de cada òrbita
Primary key	No
Data type	VARCHAR2(10)
Not NULL	Yes
Default	

Attribute: IMA_ID_IMAT	
Table	SIGI_IMATGE
Comment	Codi identificador de la imatge
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: IMA_ID_SAT	
Table	SIGI_IMATGE
Comment	Identificador de satèl·lit
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: IMA_RESOL	
Table	SIGI_IMATGE
Comment	Resolució de cada imatge
Primary key	No
Data type	VARCHAR2(10)
Not NULL	Yes
Default	

Attribute: IMA_TRACK	
Table	SIGI_IMATGE
Comment	Track de cada orbita
Primary key	No
Data type	VARCHAR2(10)
Not NULL	Yes
Default	

Attribute: MES_DIST	
Table	SIGI_MESURA
Comment	Valor d'interferometria, mesura
Primary key	No
Data type	FLOAT(5)
Not NULL	Yes
Default	

Attribute: MES_FK_IMAT	
Table	SIGI_MESURA
Comment	És la FK de imatge, que ens diu quina imatge té cada mesura
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: MES_FK_PNT	
Table	SIGI_MESURA
Comment	És la FK del punt, que ens diu de quin punt és cada mesura
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: MES_ID_MESU	
Table	SIGI_MESURA
Comment	Identificador mesures
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: PNT_FK_ZON	
Table	SIGI_PUNT
Comment	FK de la zona, és a dir que cada punt mostra l'identificador de la zona a la qual pertany
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: PNT_GEOMPUNT	
Table	SIGI_PUNT
Comment	Permet l'emagatzematge de la geometria de cada punt
Primary key	No
Data type	LONG RAW
Not NULL	Yes
Default	

Attribute: PNT_ID_PUNT	
Table	SIGI_PUNT
Comment	Identificador únic de cada punt
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: SAT_ID_SAT	
Table	SIGI_SATELLIT
Comment	Identificador de satèl·lit
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: SAT_NOM	
Table	SIGI_SATELLIT
Comment	Nom del satèl·lit
Primary key	No
Data type	VARCHAR2(10)
Not NULL	Yes

Default	
---------	--

Attribute: SAT_SENS	
Table	SIGI_SATELLIT
Comment	Sensor del satèl·lit
Primary key	No
Data type	VARCHAR2(10)
Not NULL	Yes
Default	

Attribute: VEL_COHE	
Table	SIGI_VELOC
Comment	Coherència de les dades de velocitat
Primary key	No
Data type	FLOAT(5)
Not NULL	Yes
Default	

Attribute: VEL_FK_PNT	
Table	SIGI_VELOC
Comment	Punt al que es refereixen les dades de velocitat
Primary key	No
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: VEL_ID_VELO	
Table	SIGI_VELOC
Comment	Identificador de velocitat
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: VEL_VELO	
Table	SIGI_VELOC
Comment	Velocitat del moviment per cada punt
Primary key	No
Data type	FLOAT(5)
Not NULL	Yes

Default	
---------	--

Attribute: ZIM_FK_IMA	
Table	SIGI_ZON_IMA
Comment	Fk de la imatge
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: ZIM_FK_ZON	
Table	SIGI_ZON_IMA
Comment	FK de la zona
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes
Default	

Attribute: ZON_DATAFI	
Table	SIGI_ZONA
Comment	Data de l'última imatge de la zona
Primary key	No
Data type	DATE
Not NULL	Yes
Default	

Attribute: ZON_DATAINI	
Table	SIGI_ZONA
Comment	Data d'inici de les dades d'aquesta zona
Primary key	No
Data type	DATE
Not NULL	Yes
Default	

Attribute: ZON_ID_ZONA	
Table	SIGI_ZONA
Comment	Identificador únic de cada zona
Primary key	Yes
Data type	NUMBER(10)
Not NULL	Yes

Default	
---------	--

Attribute: ZON_NOM	
Table	SIGI_ZONA
Comment	Nom de cada zona
Primary key	No
Data type	VARCHAR2(20)
Not NULL	Yes
Default	

Attribute: ZON_OBSERVACIONS	
Table	SIGI_ZONA
Comment	Proporciona informació que no és recollida en cap altra atribut de la taula
Primary key	No
Data type	VARCHAR2(2000)
Not NULL	No
Default	

Attribute: ZON_ORBITASC	
Table	SIGI_ZONA
Comment	Indica si la òrbita és ascendent (Sí, No)
Primary key	No
Data type	VARCHAR2(2)
Not NULL	Yes
Default	

List of relationships

Name	Relationship type	Parent	Child	Cardinality
SIGI_IMATGE_SIGI_MESURA	Non Identifying	SIGI_IMATGE	SIGI_MESURA	Zero Or More
SIGI_IMATGE_SIGI_ZON_IMA	Identifying	SIGI_IMATGE	SIGI_ZON_IMA	Zero Or More
SIGI_PUNT_SIGI_MESURA	Non Identifying	SIGI_PUNT	SIGI_MESURA	Zero Or More
SIGI_PUNT_SIGI_VELOC	Non Identifying	SIGI_PUNT	SIGI_VELOC	Zero Or More
SIGI_SATELLIT_SIGI_IMATGE	Non Identifying	SIGI_SATELLIT	SIGI_IMATGE	Zero Or More
SIGI_ZONA_SIGI_PUNT	Non Identifying	SIGI_ZONA	SIGI_PUNT	Zero Or More
SIGI_ZONA_SIGI_ZON_IMA	Identifying	SIGI_ZONA	SIGI_ZON_IMA	Zero Or More

Relationship details

Relationship: SIGI_IMATGE_SIGI_MESURA	
Name	SIGI_IMATGE_SIGI_MESURA
Comment	
Relationship type	Non Identifying
Cardinality	Zero Or More
Parent with columns	SIGI_IMATGE (IMA_ID_IMAT)
Child with columns	SIGI_MESURA (MES_FK_IMAT)
Update rule	None
Delete rule	None

Relationship: SIGI_IMATGE_SIGI_ZON_IMA	
Name	SIGI_IMATGE_SIGI_ZON_IMA
Comment	
Relationship type	Identifying
Cardinality	Zero Or More
Parent with columns	SIGI_IMATGE (IMA_ID_IMAT)
Child with columns	SIGI_ZON_IMA (ZIM_FK_IMA)
Update rule	None
Delete rule	None

Relationship: SIGI_PUNT_SIGI_MESURA	
Name	SIGI_PUNT_SIGI_MESURA
Comment	
Relationship type	Non Identifying
Cardinality	Zero Or More
Parent with columns	SIGI_PUNT (PNT_ID_PUNT)
Child with columns	SIGI_MESURA (MES_FK_PNT)
Update rule	None
Delete rule	Cascade

Relationship: SIGI_PUNT_SIGI_VELOC	
Name	SIGI_PUNT_SIGI_VELOC
Comment	
Relationship type	Non Identifying
Cardinality	Zero Or More
Parent with columns	SIGI_PUNT (PNT_ID_PUNT)
Child with columns	SIGI_VELOC (VEL_FK_PNT)
Update rule	None
Delete rule	Cascade

Relationship: SIGI_SATELLIT_SIGI_IMATGE	
Name	SIGI_SATELLIT_SIGI_IMATGE
Comment	
Relationship type	Non Identifying
Cardinality	Zero Or More
Parent with columns	SIGI_SATELLIT (SAT_ID_SAT)
Child with columns	SIGI_IMATGE (IMA_ID_SAT)
Update rule	None
Delete rule	None

Relationship: SIGI_ZONA_SIGI_PUNT	
Name	SIGI_ZONA_SIGI_PUNT
Comment	
Relationship type	Non Identifying
Cardinality	Zero Or More
Parent with columns	SIGI_ZONA (ZON_ID_ZONA)
Child with columns	SIGI_PUNT (PNT_FK_ZON)
Update rule	None
Delete rule	None

Relationship: SIGI_ZONA_SIGI_ZON_IMA	
Name	SIGI_ZONA_SIGI_ZON_IMA
Comment	
Relationship type	Identifying
Cardinality	Zero Or More
Parent with columns	SIGI_ZONA (ZON_ID_ZONA)
Child with columns	SIGI_ZON_IMA (ZIM_FK_ZON)
Update rule	None
Delete rule	None

List of sequences

Name	Comment
SIGI_HME_SEQ	Suma un valor cada vegada que s'insereix un registre
SIGI_HP_N_SEQ	Suma un valor cada vegada que s'insereix un registre
SIGI_HVE_SEQ	Suma un valor cada vegada que s'insereix un registre
SIGI_IMA_SEQ	Suma un valor cada vegada que s'insereix un registre
SIGI_MES_SEQ	Suma un valor cada vegada que s'insereix un registre
SIGI_PNT_SEQ	Suma un valor cada vegada que s'insereix un registre
SIGI_SAT_SEQ	Suma un valor cada vegada que s'insereix un registre
SIGI_VEL_SEQ	Suma un valor cada vegada que s'insereix un registre
SIGI_ZON_SEQ	Suma un valor cada vegada que s'insereix un registre

Sequence details

Sequence: SIGI_HME_SEQ	
Name	SIGI_HME_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	
Min	0

Sequence: SIGI_HP_N_SEQ	
Name	SIGI_HP_N_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	
Min	0

Sequence: SIGI_HVE_SEQ	
Name	SIGI_HVE_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	
Min	0

Sequence: SIGI_IMA_SEQ	
Name	SIGI_IMA_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	
Min	0

Sequence: SIGI_MES_SEQ	
Name	SIGI_MES_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	

Min	0
-----	---

Sequence: SIGI_PNT_SEQ	
Name	SIGI_PNT_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	
Min	0

Sequence: SIGI_SAT_SEQ	
Name	SIGI_SAT_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	
Min	0

Sequence: SIGI_VEL_SEQ	
Name	SIGI_VEL_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	
Min	0

Sequence: SIGI_ZON_SEQ	
Name	SIGI_ZON_SEQ
Comment	Suma un valor cada vegada que s'insereix un registre
Increment	1
Seed	0
Max	
Min	0

8.2 Codi d'implementació de la Base de dades

```
/* ----- */
/* Script generated with: DeZign for Databases v6.2.1 */
/* Target DBMS: Oracle 10g */
/* Project file: disseny_REVISAT_2.dez */
/* Project name: */
/* Author: */
/* Script type: Database creation script */
/* Created on: 2010-10-28 09:10 */
/* ----- */
```

```
/* ----- */
/* Sequences */
/* ----- */
```

```
CREATE SEQUENCE SIGI_PNT_SEQ
START WITH 1
INCREMENT BY 1
MINVALUE 1
NOMAXVALUE
nocycle
noorder;
```

```
CREATE SEQUENCE SIGI_ZON_SEQ
START WITH 1
INCREMENT BY 1
MINVALUE 1
NOMAXVALUE
nocycle
noorder;
```

```
CREATE SEQUENCE SIGI_MES_SEQ
START WITH 1
INCREMENT BY 1
MINVALUE 1
NOMAXVALUE
nocycle
noorder;
```

```
CREATE SEQUENCE SIGI_VEL_SEQ
```

```
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
NOMAXVALUE  
nocycle  
noorder;
```

```
CREATE SEQUENCE SIGI_IMA_SEQ  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
NOMAXVALUE  
nocycle  
noorder;
```

```
CREATE SEQUENCE SIGI_SAT_SEQ  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
NOMAXVALUE  
nocycle  
noorder;
```

```
CREATE SEQUENCE SIGI_HVE_SEQ  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
NOMAXVALUE  
nocycle  
noorder;
```

```
CREATE SEQUENCE SIGI_HME_SEQ  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
NOMAXVALUE  
nocycle  
noorder;
```

```
CREATE SEQUENCE SIGI_HP_N_SEQ  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1
```

```

NOMAXVALUE
nocycle
noorder;

/* ----- */
/* Tables */
/* ----- */

/* ----- */
/* Add table "SIGI_PUNT" */
/* ----- */
/*
CREATE TABLE SIGI_PUNT (
  PNT_ID_PUNT NUMBER(10) CONSTRAINT NN_SIGI_PUNT_PNT_ID_PUNT
  NOT NULL,
  PNT_GEOMPUNT          SDO_GEOMETRY          CONSTRAINT
  NN_SIGI_PUNT_PNT_GEOMPUNT NOT NULL,
  PNT_FK_ZON NUMBER(10) CONSTRAINT NN_SIGI_PUNT_PNT_FK_ZON
  NOT NULL,
  CONSTRAINT PK_SIGI_PUNT PRIMARY KEY (PNT_ID_PUNT)
);
*/

CREATE INDEX IDX_SIGI_PUNT_1_FK ON SIGI_PUNT (PNT_FK_ZON);

COMMENT ON TABLE SIGI_PUNT IS 'Aquesta taula tindrà tots els punts que
s'estudien a Catalunya.';

COMMENT ON COLUMN SIGI_PUNT.PNT_ID_PUNT IS 'Identificador únic de
cada punt';

COMMENT ON COLUMN SIGI_PUNT.PNT_GEOMPUNT IS 'Permet
l'emagatzematge de la geometria de cada punt';

COMMENT ON COLUMN SIGI_PUNT.PNT_FK_ZON IS 'FK de la zona, és a dir
que cada punt mostra l'identificador de la zona a la qual pertany';

/* ----- */
/* Add table "SIGI_MESURA" */
/* ----- */

CREATE TABLE SIGI_MESURA (

```

```

    MES_ID_MESU          NUMBER(10)          CONSTRAINT
NN_SIGI_MESURA_MES_ID_MESU NOT NULL,
    MES_DIST FLOAT(5)   CONSTRAINT NN_SIGI_MESURA_MES_DIST NOT
NULL,
    MES_FK_PNT          NUMBER(10)          CONSTRAINT
NN_SIGI_MESURA_MES_FK_PNT NOT NULL,
    MES_FK_IMAT NUMBER(10),
    CONSTRAINT PK_SIGI_MESURA PRIMARY KEY (MES_ID_MESU)
);

```

```

CREATE INDEX IDX_SIGI_MESURA_1_FK ON SIGI_MESURA (MES_FK_PNT);

```

```

CREATE INDEX IDX_SIGI_MESURA_2_FK ON SIGI_MESURA (MES_FK_IMAT);

```

```

COMMENT ON TABLE SIGI_MESURA IS 'Conté les mesures referents a cada
punt';

```

```

COMMENT ON COLUMN SIGI_MESURA.MES_ID_MESU IS 'Identificador
mesures';

```

```

COMMENT ON COLUMN SIGI_MESURA.MES_DIST IS 'Valor d"interferometria,
mesura ';

```

```

COMMENT ON COLUMN SIGI_MESURA.MES_FK_PNT IS 'És la FK del punt,
que ens diu de quin punt és cada mesura';

```

```

COMMENT ON COLUMN SIGI_MESURA.MES_FK_IMAT IS 'És la FK de imatge,
que ens diu quina imatge té cada mesura';

```

```

/* ----- */
/* Add table "SIGI_ZONA" */
/* ----- */
/*

```

```

CREATE TABLE SIGI_ZONA (
    ZON_ID_ZONA NUMBER(10) CONSTRAINT NN_SIGI_ZONA_ZON_ID_ZONA
NOT NULL,
    ZON_NOM VARCHAR2(20) CONSTRAINT NN_SIGI_ZONA_ZON_NOM NOT
NULL,
    ZON_ORBITA VARCHAR2(2) CONSTRAINT NN_SIGI_ZONA_ZON_ORBITA
NOT NULL,
    ZON_DATAIN DATE CONSTRAINT NN_SIGI_ZONA_ZON_DATAIN NOT
NULL,
    ZON_DATAFI DATE CONSTRAINT NN_SIGI_ZONA_ZON_DATAFI NOT NULL,

```

```

        ZON_GEOMPO          SDO_GEOMETRY          CONSTRAINT
NN_SIGI_ZONA_ZON_GEOMPO NOT NULL,
        ZON_OBSERV VARCHAR2(2000),
        PRIMARY KEY (ZON_ID_ZONA)
);
*/

```

COMMENT ON TABLE SIGI_ZONA IS 'Proporciona informació sobre les zones d'estudi';

COMMENT ON COLUMN SIGI_ZONA.ZON_ID_ZONA IS 'Identificador únic de cada zona';

COMMENT ON COLUMN SIGI_ZONA.ZON_NOM IS 'Nom de cada zona';

COMMENT ON COLUMN SIGI_ZONA.ZON_ORBITA IS 'Indica si la òrbita és ascendent (Sí, No)';

COMMENT ON COLUMN SIGI_ZONA.ZON_DATAIN IS 'Data d'inici de les dades d'aquesta zona';

COMMENT ON COLUMN SIGI_ZONA.ZON_DATAFI IS 'Data de l'última imatge de la zona';

COMMENT ON COLUMN SIGI_ZONA.ZON_GEOMPO IS 'Permet l'emmagatzematge de la geometria de cada zona';

COMMENT ON COLUMN SIGI_ZONA.ZON_OBSERV IS 'Proporciona informació que no és recollida en cap altra atribut de la taula';

```

/* ----- */
/* Add table "SIGI_IMATGE" */
/* ----- */

```

```

CREATE TABLE SIGI_IMATGE (
        IMA_ID_IMAT NUMBER(10) CONSTRAINT NN_SIGI_IMATGE_IMA_ID_IMAT
NOT NULL,
        IMA_DATA DATE CONSTRAINT NN_SIGI_IMATGE_IMA_DATA NOT NULL,
        IMA_TRACK VARCHAR2(10) CONSTRAINT NN_SIGI_IMATGE_IMA_TRACK
NOT NULL,
        IMA_FRAME VARCHAR2(10) CONSTRAINT NN_SIGI_IMATGE_IMA_FRAME
NOT NULL,

```



```

    IMA_ID_SAT NUMBER(10) CONSTRAINT NN_SIGI_IMATGE_IMA_ID_SAT
NOT NULL,
    CONSTRAINT PK_SIGI_IMATGE PRIMARY KEY (IMA_ID_IMAT)
);

```

```

CREATE INDEX IDX_SIGI_IMATGE_1_FK ON SIGI_IMATGE (IMA_ID_SAT);

```

```

COMMENT ON TABLE SIGI_IMATGE IS 'És un recull de totes les imatges de
satèl·lit que hi ha';

```

```

COMMENT ON COLUMN SIGI_IMATGE.IMA_ID_IMAT IS 'Codi identificador de la
imatge';

```

```

COMMENT ON COLUMN SIGI_IMATGE.IMA_DATA IS 'Data de cada imatge';

```

```

COMMENT ON COLUMN SIGI_IMATGE.IMA_TRACK IS 'Track de cada orbita';

```

```

COMMENT ON COLUMN SIGI_IMATGE.IMA_FRAME IS 'Frame de cada òrbita';

```

```

COMMENT ON COLUMN SIGI_IMATGE.IMA_ID_SAT IS 'Identificador de satèl·lit';

```

```

/* ----- */
/* Add table "SIGI_SATELLIT" */
/* ----- */

```

```

CREATE TABLE SIGI_SATELLIT (
    SAT_ID_SAT NUMBER(10) CONSTRAINT NN_SIGI_SATELLIT_SAT_ID_SAT
NOT NULL,
    SAT_NOM VARCHAR2(100) CONSTRAINT NN_SIGI_SATELLIT_SAT_NOM
NOT NULL,
    SAT_SENS VARCHAR2(10) CONSTRAINT NN_SIGI_SATELLIT_SAT_SENS
NOT NULL,
    SAT_RESOL          VARCHAR2(10)          CONSTRAINT
NN_SIGI_SATELLIT_SAT_RESOL NOT NULL,
    SAT_ANGLE          VARCHAR2(10)          CONSTRAINT
NN_SIGI_SATELLIT_SAT_ABGLE NOT NULL,
    CONSTRAINT PK_SIGI_SATELLIT PRIMARY KEY (SAT_ID_SAT)
);

```

```

COMMENT ON TABLE SIGI_SATELLIT IS 'Taula de les propietats del satèl·lit';

```

```

COMMENT ON COLUMN SIGI_SATELLIT.SAT_ID_SAT IS 'Identificador de
satèl·lit';

```

```
COMMENT ON COLUMN SIGI_SATELLIT.SAT_NOM IS 'Nom del satèl·lit';

COMMENT ON COLUMN SIGI_SATELLIT.SAT_SENS IS 'Sensor del satèl·lit';

COMMENT ON COLUMN SIGI_SATELLIT.SAT_RESOL IS 'Resolució de cada
sensor';

COMMENT ON COLUMN SIGI_SATELLIT.SAT_ANGLE IS 'Angle incidència';

/* ----- */
/* Add table "SIGI_VELOC" */
/* ----- */

CREATE TABLE SIGI_VELOC (
  VEL_ID_VELO NUMBER(10) CONSTRAINT NN_SIGI_VELOC_VEL_ID_VELO
NOT NULL,
  VEL_VELO FLOAT(5) CONSTRAINT NN_SIGI_VELOC_VEL_VELO NOT
NULL,
  VEL_COHE FLOAT(5) CONSTRAINT NN_SIGI_VELOC_VEL_COHE NOT
NULL,
  VEL_FK_PNT NUMBER(10) CONSTRAINT NN_SIGI_VELOC_VEL_FK_PNT
NOT NULL,
  CONSTRAINT PK_SIGI_VELOC PRIMARY KEY (VEL_ID_VELO)
);

CREATE INDEX IDX_SIGI_VELOC_1_FK ON SIGI_VELOC (VEL_FK_PNT);

COMMENT ON TABLE SIGI_VELOC IS 'Ens permet veure la velocitat a la que es
mou cada punt i la coherència d'aquesta velocitat, la informació d'aquesta taula
s'obté a partir de les dades de la taula de mesures associades a cada punt';

COMMENT ON COLUMN SIGI_VELOC.VEL_ID_VELO IS 'Identificador de
velocitat';

COMMENT ON COLUMN SIGI_VELOC.VEL_VELO IS 'Velocitat del moviment per
cada punt';

COMMENT ON COLUMN SIGI_VELOC.VEL_COHE IS 'Coherència de les dades
de velocitat';

COMMENT ON COLUMN SIGI_VELOC.VEL_FK_PNT IS 'Punt al que es
refereixen les dades de velocitat';
```

```

/* ----- */
/* Add table "SIGI_ZON_IMA" */
/* ----- */

CREATE TABLE SIGI_ZON_IMA (
  ZIM_FK_ZON NUMBER(10) CONSTRAINT NN_SIGI_ZON_IMA_ZIM_FK_ZON
  NOT NULL,
  ZIM_FK_IMA NUMBER(10) CONSTRAINT NN_SIGI_ZON_IMA_ZIM_FK_IMA
  NOT NULL,
  CONSTRAINT PK_SIGI_ZON_IMA PRIMARY KEY (ZIM_FK_ZON,
  ZIM_FK_IMA)
);

CREATE INDEX IDX_SIGI_ZON_IMA_1_FK ON SIGI_ZON_IMA (ZIM_FK_ZON);

CREATE INDEX IDX_SIGI_ZON_IMA_2_FK ON SIGI_ZON_IMA (ZIM_FK_IMA);

COMMENT ON TABLE SIGI_ZON_IMA IS 'Permet la implementació de la relació
N a N entre imatges i zones. Una imatge ens pot donar informació de moltes zones
i una zona pot estar informada per més d"una imatge';

COMMENT ON COLUMN SIGI_ZON_IMA.ZIM_FK_ZON IS 'FK de la zona';

COMMENT ON COLUMN SIGI_ZON_IMA.ZIM_FK_IMA IS 'Fk de la imatge';

/* ----- */
/* Add table "SIGI_HISTMESURA" */
/* ----- */

CREATE TABLE SIGI_HISTMESURA (
  HME_ID_HMES NUMBER(10) CONSTRAINT
  NN_SIGI_HISTMESURA_HME_ID_HMES NOT NULL,
  HME_DIST FLOAT(5) CONSTRAINT NN_SIGI_HISTMESURA_HME_DIST
  NOT NULL,
  HME_FK_PNT NUMBER(10) CONSTRAINT
  NN_SIGI_HISTMESURA_HME_FK_PNT NOT NULL,
  HME_FK_IMAT NUMBER(10) CONSTRAINT
  NN_SIGI_HISTMESURA_HME_FK_IMAT NOT NULL,
  HME_DATAHIST DATE,
  CONSTRAINT PK_SIGI_HISTMESURA PRIMARY KEY (HME_ID_HMES)
);

```

COMMENT ON TABLE SIGI_HISTMESURA IS 'Conté els registres històrics de les mesures referents a cada punt';

COMMENT ON COLUMN SIGI_HISTMESURA.HME_ID_HMES IS 'Identificador mesures';

COMMENT ON COLUMN SIGI_HISTMESURA.HME_DIST IS 'Valor d"interferometria, mesura ';

COMMENT ON COLUMN SIGI_HISTMESURA.HME_FK_PNT IS 'És la FK del punt, que ens diu de quin punt és cada mesura';

COMMENT ON COLUMN SIGI_HISTMESURA.HME_FK_IMAT IS 'És la FK de imatge, que ens diu quina imatge té cada mesura';

COMMENT ON COLUMN SIGI_HISTMESURA.HME_DATAHIST IS 'Data d"inserció del registre a la taula d"històrics. Permetrà obtenir la vigència temporal de la mesura, que serà la diferència entre HME_DATAHIST I ZON_DATAIN';

```
/* ----- */
/* Add table "SIGI_HISTVELOC" */
/* ----- */
```

```
CREATE TABLE SIGI_HISTVELOC (
  HVE_ID_HVELO          NUMBER(10)          CONSTRAINT
  NN_SIGI_HISTVELOC_HVE_ID_HVELO NOT NULL,
  HVE_VELO FLOAT(5) CONSTRAINT NN_SIGI_HISTVELOC_HVE_VELO NOT
  NULL,
  HVE_COHE FLOAT(5) CONSTRAINT NN_SIGI_HISTVELOC_HVE_COHE
  NOT NULL,
  HVE_FK_PNT           NUMBER(10)          CONSTRAINT
  NN_SIGI_HISTVELOC_HVE_FK_PNT NOT NULL,
  HVE_DATAHIST DATE,
  CONSTRAINT PK_SIGI_HISTVELOC PRIMARY KEY (HVE_ID_HVELO)
);
```

COMMENT ON TABLE SIGI_HISTVELOC IS 'Conté els registres històrics de la velocitat a la que es mou cada punt i la coherència d"aquesta velocitat, la informació d"aquesta taula s"obté a partir de les dades de la taula de mesures associades a cada punt';

COMMENT ON COLUMN SIGI_HISTVELOC.HVE_ID_HVELO IS 'Identificador de velocitat';

COMMENT ON COLUMN SIGI_HISTVELOC.HVE_VELO IS 'Velocitat del moviment per cada punt';

COMMENT ON COLUMN SIGI_HISTVELOC.HVE_COHE IS 'Coherència de les dades de velocitat';

COMMENT ON COLUMN SIGI_HISTVELOC.HVE_FK_PNT IS 'Punt al que es refereixen les dades de velocitat';

COMMENT ON COLUMN SIGI_HISTVELOC.HVE_DATAHIST IS 'Data d'inserció del registre a la taula d'històrics. Serà la data en la què s'elimina el registre de la taula veloc. Ens permetrà conèixer la vigència temporal de la informació de velocitat, que serà la diferència entre HVE_DATAHIST i ZON_DATAIN';

```
/* ----- */
/* Add table "SIGI_HISTPUNT" */
/* ----- */
```

```
CREATE TABLE SIGI_HISTPUNT (
  HPN_ID_HPUNT          NUMBER(10)          CONSTRAINT
NN_SIGI_HISTPUNT_HP_N_ID_HPUNT NOT NULL,
  HPN_GEOMPUNT          SDO_GEOMETRY        CONSTRAINT
NN_SIGI_HISTPUNT_HP_N_GEOMPUNT NOT NULL,
  HPN_FK_ZON            NUMBER(10)          CONSTRAINT
NN_SIGI_HISTPUNT_HP_N_FK_ZON NOT NULL,
  HPN_DATAHIST DATE,
  CONSTRAINT PK_SIGI_HISTPUNT PRIMARY KEY (HPN_ID_HPUNT)
);
```

COMMENT ON TABLE SIGI_HISTPUNT IS 'Conté els registres històrics dels punts referents a cada zona';

COMMENT ON COLUMN SIGI_HISTPUNT.HPN_ID_HPUNT IS 'Identificador únic de cada punt';

COMMENT ON COLUMN SIGI_HISTPUNT.HPN_GEOMPUNT IS 'Permet l'emagatzematge de la geometria de cada punt';

COMMENT ON COLUMN SIGI_HISTPUNT.HPN_FK_ZON IS 'FK de la zona, és a dir que cada punt mostra l'identificador de la zona a la qual pertany';

COMMENT ON COLUMN SIGI_HISTPUNT.HPN_DATAHIST IS 'Data d'inserció del registre a la taula d'històrics. Permetrà obtenir la vigència temporal dels punts, que serà la diferència entre HPN_DATAHIST i ZON_DATAIN';

```
/* ----- */
/* Foreign key constraints */
/* ----- */
```

```
ALTER TABLE SIGI_PUNT ADD CONSTRAINT SIGI_ZONA_SIGI_PUNT
  FOREIGN KEY (PNT_FK_ZON) REFERENCES SIGI_ZONA (ZON_ID_ZON);
```

```
ALTER TABLE SIGI_MESURA ADD CONSTRAINT SIGI_PUNT_SIGI_MESURA
  FOREIGN KEY (MES_FK_PNT) REFERENCES SIGI_PUNT (PNT_ID_PUN)
ON DELETE CASCADE;
```

```
ALTER TABLE SIGI_MESURA ADD CONSTRAINT
SIGI_IMATGE_SIGI_MESURA
  FOREIGN KEY (MES_FK_IMAT) REFERENCES SIGI_IMATGE
(IMA_ID_IMAT);
```

```
ALTER TABLE SIGI_IMATGE ADD CONSTRAINT SIGI_SATELLIT_SIGI_IMATGE
  FOREIGN KEY (IMA_ID_SAT) REFERENCES SIGI_SATELLIT (SAT_ID_SAT);
```

```
ALTER TABLE SIGI_VELOC ADD CONSTRAINT SIGI_PUNT_SIGI_VELOC
  FOREIGN KEY (VEL_FK_PNT) REFERENCES SIGI_PUNT (PNT_ID_PUN) ON
DELETE CASCADE;
```

```
ALTER TABLE SIGI_ZON_IMA ADD CONSTRAINT SIGI_ZONA_SIGI_ZON_IMA
  FOREIGN KEY (ZIM_FK_ZON) REFERENCES SIGI_ZONA (ZON_ID_ZON);
```

```
ALTER TABLE SIGI_ZON_IMA ADD CONSTRAINT
SIGI_IMATGE_SIGI_ZON_IMA
  FOREIGN KEY (ZIM_FK_IMA) REFERENCES SIGI_IMATGE (IMA_ID_IMAT);
```

```
/* ----- */
/* Triggers */
/* ----- */
```

```
CREATE OR REPLACE TRIGGER SIGI_PNT_TR_SEQ
BEFORE INSERT ON SIGI_PUNT
FOR EACH ROW
BEGIN
  IF :NEW.PNT_ID_PUNT IS NULL THEN
```

```

                SELECT SIGI_PNT_SEQ.NEXTVAL
                INTO :NEW.PNT_ID_PUNT
                FROM DUAL;
                END IF;
END;

/

create or replace TRIGGER SIGI_HPNT_TR_DELETE
BEFORE DELETE ON SIGI_PUNT
FOR EACH ROW
BEGIN
    INSERT INTO SIGI_HISTPUNT(HPN_ID_HPUNT, HPN_GEOMPUNT,
HPN_FK_ZON, HPN_DATAHIST)
VALUES ( :OLD.PNT_ID_PUNT, :OLD.PNT_GEOMPUNT, :OLD.PNT_FK_ZON,
SYSDATE);
END;

/

CREATE OR REPLACE TRIGGER SIGI_MES_TR_SEQ
BEFORE INSERT ON SIGI_MESURA
FOR EACH ROW
BEGIN
    IF :NEW.MES_ID_MESU IS NULL THEN
        SELECT SIGI_MES_SEQ.NEXTVAL
        INTO :NEW.MES_ID_MESU
        FROM DUAL;
        END IF;
END;

/

create or replace TRIGGER SIGI_MES_TR_DELETE
BEFORE DELETE ON SIGI_MESURA
FOR EACH ROW
BEGIN
INSERT INTO SIGI_HISTMESURA(HME_ID_HMES, HME_DIST, HME_FK_PNT,
HME_FK_IMAT, HME_DATAHIST)
VALUES ( :OLD.MES_ID_MESU, :OLD.MES_DIST, :OLD.MES_FK_PNT,
:OLD.MES_FK_IMAT, SYSDATE);
END;

```

/

```
CREATE OR REPLACE TRIGGER SIGI_ZON_TR_SEQ
BEFORE INSERT ON SIGI_ZONA
FOR EACH ROW
BEGIN
```

```
    IF :NEW.ZON_ID_ZONA IS NULL THEN
    SELECT SIGI_ZON_SEQ.NEXTVAL
    INTO :NEW.ZON_ID_ZONA
    FROM DUAL;
    END IF;
```

```
END;
```

/

```
CREATE OR REPLACE TRIGGER SIGI_ZON_TR_BUPDATE
BEFORE UPDATE ON SIGI_ZONA
FOR EACH ROW
BEGIN
```

```
    IF :NEW.ZON_DATAIN <> :OLD.ZON_DATAIN OR :NEW.ZON_DATAFI <>
:OLD.ZON_DATAFI THEN
        DELETE FROM SIGI_PUNT WHERE PNT_FK_ZON =
:NEW.ZON_ID_ZONA;
    END IF;
```

```
END;
```

/

```
CREATE OR REPLACE TRIGGER SIGI_IMA_TR_SEQ
BEFORE INSERT ON SIGI_IMATGE
FOR EACH ROW
BEGIN
```

```
    IF :NEW.IMA_ID_IMAT IS NULL THEN
    SELECT SIGI_IMA_SEQ.NEXTVAL
    INTO :NEW.IMA_ID_IMAT
    FROM DUAL;
    END IF;
```

```
END;
```

/

```
CREATE OR REPLACE TRIGGER SIGI_SAT_TR_SEQ
BEFORE INSERT ON SIGI_SATELLIT
```



```
FOR EACH ROW
BEGIN
```

```
    IF :NEW.SAT_ID_SAT IS NULL THEN
        SELECT SIGI_SAT_SEQ.NEXTVAL
        INTO :NEW.SAT_ID_SAT
        FROM DUAL;
    END IF;
```

```
END;
```

```
/
```

```
CREATE OR REPLACE TRIGGER SIGI_VEL_TR_SEQ
BEFORE INSERT ON SIGI_VELOC
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF :NEW.VEL_ID_VELO IS NULL THEN
        SELECT SIGI_VEL_SEQ.NEXTVAL
        INTO :NEW.VEL_ID_VELO
        FROM DUAL;
    END IF;
```

```
END;
```

```
/
```

```
create or replace TRIGGER SIGI_VEL_TR_DELETE
```

```
BEFORE DELETE ON SIGI_VELOC
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO SIGI_HISTVELOC(HVE_ID_HVELO, HVE_VELO, HVE_COHE,
HVE_FK_PNT, HVE_DATAHIST)
```

```
VALUES ( :OLD.VEL_ID_VELO, :OLD.VEL_VELO, :OLD.VEL_COHE,
:OLD.VEL_FK_PNT, SYSDATE);
```

```
END;
```

```
/
```

```
CREATE OR REPLACE TRIGGER SIGI_HME_TR_SEQ
```

```
BEFORE INSERT ON SIGI_HISTMESURA
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF :NEW.HME_ID_HMES IS NULL THEN
        SELECT SIGI_HME_SEQ.NEXTVAL
        INTO :NEW.HME_ID_HMES
```

```

                FROM DUAL;
                END IF;
END;

/

CREATE OR REPLACE TRIGGER SIGI_HVE_TR_SEQ
BEFORE INSERT ON SIGI_HISTVELOC
FOR EACH ROW
BEGIN
    IF :NEW.HVE_ID_HVELO IS NULL THEN
        SELECT SIGI_HVE_SEQ.NEXTVAL
        INTO :NEW.HVE_ID_HVELO
        FROM DUAL;
    END IF;
END;

/

CREATE OR REPLACE TRIGGER SIGI_HPN_TR_SEQ
BEFORE INSERT ON SIGI_HISTPUNT
FOR EACH ROW
BEGIN
    IF :NEW.HPN_ID_HPUNT IS NULL THEN
        SELECT SIGI_HPN_SEQ.NEXTVAL
        INTO :NEW.HPN_ID_HPUNT
        FROM DUAL;
    END IF;
END;

/

```

8.3 Esquema de l'aplicació

