

Cal que proposeu un esquema de base de dades relacional (disseny lògic amb les taules amb atributs, claus, restriccions, etc) per a enmagatzemar la informació del sistema especificat i determineu quins mecanismes faríeu servir per tal d'assegurar la consistència de la base de dades (per comprovar les RI).

Realitzaré les taules segons la jerarquia d'especialització *Concrete Table Inheritance*, ja que és un nivell intermig entre la *Class Table Inheritance* i la *Single Table Inheritance*. He decidit fer-ho així perquè la *Class Table Inheritance* em sembla massa explícita, inclús redundant, d'aquí que sigui menys eficient. Per altra banda, la *Single Table Inheritance* em sembla massa enrebuscada, d'aquí que hagi decidit fer-ho amb el nivell intermig d'especificació.

Administratiu(nomLloc, tipusLlocAdministratiu, detallLloc, niellLloc)

Religiós(nomLloc, tipusLlocReligios, detallLloc, niellLloc)

Baptisme(num-partida, pagina, cara, localització)

Participant(nom, cognom1, cognom2, ofici_carrec)

Batejat_da(nom, cognom1, cognom2, nom)

Anomenat(nom, cognom1, cognom2, ofici-carrec)

Presona(nom, cognom1, cognom2, sexe, estat_vital, ofici_carrec, alies, estat_civil, data_naixement, hora_naixement, tipus_fill, natural, residència)

Les restriccions d'integritat que es mostren a continuació són d'obligat compliment per la consistència del esquema, i per tal de que es compleixin, ho farem per mitjà de disparadors. D'aquesta forma, serà la pròpia Base de Dades la que se'n encarregarà de que només es puguin fer correctament.

Nom: unicBatejat

Descripció: Només pot haver-hi un sol batejat per cada acta de baptisme. OCL:
context Baptisme inv unicBatejat : self.rol.oclAsType(Batejat_da)->one(not isUndefined())

Nom: minimOficiant

Descripció: Hi ha com a mínim un Oficiant per cada acte de baptisme. OCL:
context Baptisme inv minimOficiant :
self.rol.oclAsType(Oficiant)->select(m | m.isUndefined()=false)->notEmpty()

Nom: dataNaixInfBaptisme

Descripció: La data de naixement, si hi consta, ha de ser inferior o igual a la data del baptisme. OCL:
context Baptisme inv dataNaixInfBaptisme :
let x : Integer = self.rol.oclAsType(Batejat_da) ->select(not isUndefined()))->asSequence()->first().data_naixement
in
if not x.isUndefined() then x <= self.data_baptisme
else true
endif

Nom: altreRelacioSiMateix

Descripció: La classe associativa entre Participant i Rol no permet que existeixi una AltraRelacio amb una mateixa instància. OCL:
context Participant inv altreRelacioSiMateix : AltreRelacio.allInstances()->select(participant=rol)->isEmpty()

Nom: unaPersonaXPartida

Descripció: Una instància de persona no pot participar en la mateixa partida en dos rols diferents. OCL:
context Rol inv unaPersonaXPartida :
Rol.allInstances()->forAll(v1,v2 |
(v1.persona = v2.persona implies (v1.partida <> v2.partida)) and (v1.partida = v2.partida implies (v1.persona <> v2.persona) or (v1=v2))

Nom: clausPrimaries

Descripció: Una partida s'identifica perquè aquestes claus primàries són vàlides i úniques. OCL:
context Baptisme inv clausPrimaries : Baptisme.allInstances()->
isUnique(Set{self.pagina,self.cara,self.localitzacio,self.num_partida})

