Arquitectura de Base de Dades

Pràctica 1: Extensió *Object-Relational*

Pregunta 1 (2.5 punts)

Enunciat

Donades les afirmacions següents indica, per a cadascuna d'elles, si són certes o falses. Justifica breument la teva resposta.

a) Degut a les limitacions de l'extensió *object-relational*, les herències entre objectes només es poden realitzar mitjançant referències del tipus REF.

b) Sota l'extensió *object-relational*, tant les claus foranes com les referències a objectes poden quedar orfes o incertes (*dangling*) si desapareix la dada a la qual apunten.

c) Des del punt de vista de gestió de reserves, els VARRAY permeten un major nivell de concurrència que les taules aniuades (*nested tables*) donat que només requereixen una operació d'entrada/sortida per recuperar tota la col·lecció d'elements.

d) En el cas de crear un tipus (*type*) a Oracle amb el paràmetre NOT INSTANTIABLE, sota cap condició en podrem crear instàncies.

Criteris d'avaluació

Els criteris d'avaluació que s'aplicaran en la correcció d'aquesta pregunta són els següents:

- Totes les preguntes tenen el mateix pes.
- Les preguntes no contestades no penalitzen.
- Les preguntes sense argumentació no seran avaluades.
- Es valorarà la qualitat de la resposta i el fet de no entrar en contradiccions amb la resposta donada (cert o fals).

Solució

a) **FALS**. Sota el model relacional, les herències es realitzen o simulen mitjançant claus foranes. En el model *object-relational*, és cert que tenim l'opció d'utilitzar les referències REF com a apuntadors i així simular l'herència, no obstant això, disposem d'un mecanisme d'herència (UNDER) que ens permet heretar tant els atributs com els mètodes.

b) **FALS**. Les referències a objectes són un mecanisme essencial per tal de fer complir restriccions d'integritat, igual que les claus foranes al model relacional. Cal tenir en compte que sota l'extensió *object-relational* conviuen ambdós mecanismes, essent responsabilitat del desenvolupador l'elecció d'un o altre.

Les claus foranes apunten a les claus candidates d'altres taules, mentre que les referències ho fan cap a objectes. Una de les principals diferències entre aquest mecanisme és precisament el fet de que les referències a objectes poden quedar orfes (*dangling*), és a dir, que pot donar-se el cas que l'objecte al què apuntaven deixi d'existir.

Aquest fet no pot succeir en les claus foranes, aquestes sempre garantiran que la clau candidata a la qual fan referència existeix, per tant, l'afirmació és falsa. Val a dir però que es poden implementar regles d'integritat referencial en l'ús de referències per evitar que aquestes ens quedin orfes en un moment donat.

c) **FALS**. VARRAY és una solució orientada a columnes, capaç de llegir, de forma general, tota la col·lecció d'elements d'una sola operació d'entrada/sortida. Únicament si el VARRAY s'emmagatzema fora de la taula consultada o és massa gran, aquest serà emmagatzemat com un BLOB (*binary large object*) i en aquest cas sí que requerirà més d'una operació d'entrada/sortida.

Per altra banda, les taules aniuades (*nested tables*) treballen a nivell de fila i per tant, ens poden caldre moltes operacions d'entrada/sortida per a recuperar totes les files d'una consulta. És precisament aquesta orientació a fila la que permet a les taules aniuades obtenir un major nivell de concurrència, ja que podem treballar amb reserves a nivell de fila i per tant, no bloquejar la resta d'objectes que componen l'estructura.

Per contra, els VARRAY, al treballar a nivell de columna, requereixen d'un bloqueig de la columna i per tant, de tots els elements que composen el VARRAY sobre el qual es vol treballar, per tant, l'afirmació és falsa.

d) **FALS**. Si bé a priori el paràmetre NOT INSTANTIABLE sobre un tipus, fa que aquest no pugui ser instanciat, mitjançant la sentència ALTER podem canviar-ne les propietats i modificar la condició de no instanciable a instanciable, aplicant la sentència:

ALTER TYPE nom_tipus INSTANTIABLE;

En el cas d'Oracle, no es pot modificar un tipus definit com a instanciable, a no instanciable, si aquest té alguna dependència sobre ell.

Pregunta 2 (5 punts)

Enunciat

Com a arquitecte de les bases de dades de l'empresa Enforma, SL, encarregada de la gestió de gimnasos, se t'ha encarregat la tasca de dissenyar i desenvolupar el model de dades pel nou sistema gestor de clients i cursos dels gimnasos.

El gimnàs té tres tipus de clients diferents, els que només van a la piscina (*Swiming Customer*), els clients que només poden assistir al gimnàs al matí (*Morning Customer*) i els usuaris que disposen d'accés total al centre (*Global Customer*).

L'empresa està interessada en registrar per cada client, les seves dades personals: DNI, nom, cognoms, data de naixement, telèfon, correu electrònic, tipus de pagament de quota (anual, mensual), número de compte corrent, en el cas dels clients de piscina també registrarem el nivell dels clients en format numèric, si no sap nedar aquest és zero.

El gimnàs disposa d'un conjunt de monitors (*Monitor Staff*), amb diferents especialitzacions, que s'encarreguen d'impartir classes dirigides i donar suport als usuaris del gimnàs tant en les sales de màquines com en la piscina.

Dels monitors també volem conèixer les dades personals (DNI, nom, cognoms, data de naixement, telèfon), també registrarem el compte bancari on es realitza el pagament de la nòmina, la data d'incorporació a l'empresa, i el codi numèric que l'identifica dins de la base de dades de personal.

L'oferta de cursos d'entrenament (*Training Courses*) ve marcada per les especialitzacions (*Specialization*), on un curs pot donar resposta fins a tres d'aquestes. Les especialitzacions defineixen el funcionament del curs i les activitats que s'hi realitzen. De cada curs volem registrar-ne el codi únic alfanumèric que l'identifica, el nom de curs, la durada en hores, les especialitzacions a que dona resposta, dies de la setmana i hores que es realitza (aquesta informació es registrarà en format textual). Cada curs té capacitat per a 50 clients.

De les especialitzacions voldrem conèixer el codi identificador, el nom, i la descripció d'aquestes.

Cada vegada que un client accedeix al gimnàs, introduint el seu codi personal en el sistema de validació situat a l'entrada, es realitza un registre de l'hora, persona, i la sala per on s'està accedint. Aquesta informació d'accés (*Access Information*) permetrà controlar els accessos dels usuaris i mantenir un històric.

El mateix sistema de control es troba en les portes d'accés a les diferents sales del gimnàs (*Rooms*), i així es controla que els usuaris que estan accedint a cada una de les activitats o sales disposa d'una quota que li permet fer-ho. De les sales del gimnàs, on incloem l'entrada del gimnàs, en registrarem el codi intern, i el nom.

El gimnàs permet que els clients creïn grups d'entrenament (*Training Group*), aquests poden tenir fins a 10 usuaris i se li assigna un monitor, segons l'objectiu del curs. El preu d'aquest servei varia segons el nombre de clients que s'hi apunten.

Dels grups en volem saber el codi intern, nom, monitor assignat, preu i llista de clients.

Es demana realitzar:

a) L'esquema UML que il·lustri les associacions entre les classes identificades en el vostre disseny.

b) La implementació de l'esquema UML anterior en forma *d'script* SQL sobre Oracle DB XE 11g2, utilitzant la orientació a objectes (extensió *object-relational*). També cal lliurar les sentències d'inserció d'informació que permetin validar el correcte funcionament de la implementació.

c) Un informe detallat explicant totes i cadascuna de les decisions d'implementació que s'hagin pres durant la implementació.

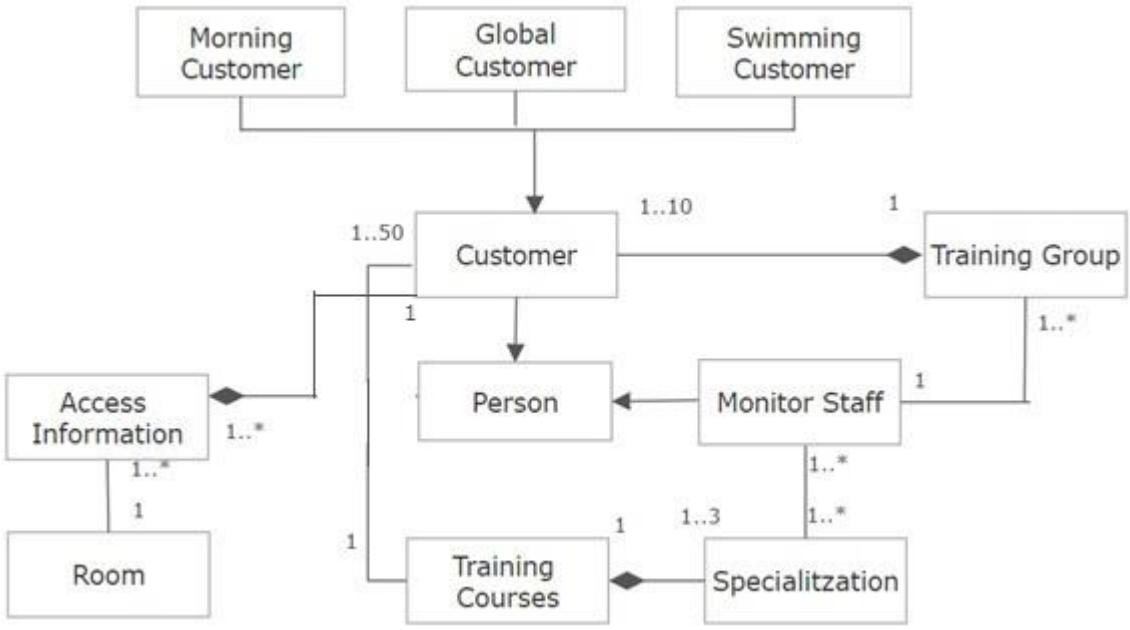
Criteris d'avaluació

Els criteris d'avaluació que s'aplicaran en la correcció d'aquesta pregunta són els següents:

- Cal lliurar tots els apartats per tal que l'exercici sigui avaluat.
- Es valorarà la qualitat de la resposta, així com l'ús dels conceptes estudiats en el mòdul 2 corresponents a l'extensió *Object-Relational*.
- Es valorarà la coherència entre el disseny proposat i la seva argumentació.
- Es valorarà que el codi SQL lliurat funcioni i pugui ser executat.
- Les respostes incorrectes no descompten.

Solució

a) Es presenta a continuació un possible esquema UML que dona resposta al plantejament de l'enunciat:



b) A continuació es presenta una proposta *d'script* SQL en base al model UML presentat anteriorment:

```
CREATE OR REPLACE TYPE Specialization AS OBJECT(
code VARCHAR(15),
name VARCHAR(50),
description VARCHAR(300)
);

CREATE TABLE Specializations OF Specialization (code PRIMARY KEY);

CREATE TYPE tSpecialization AS TABLE OF REF Specialization;

CREATE OR REPLACE TYPE specializationArray AS VARRAY(3) OF REF
Specialization;

CREATE OR REPLACE TYPE Person AS OBJECT(
NIF VARCHAR(15),
name VARCHAR(60),
surname VARCHAR(150),
birthDate               DATE,
phone                   VARCHAR(15),
```

```
bankAccount VARCHAR(50)
)NOT FINAL;

CREATE OR REPLACE TYPE Customer UNDER Person(
tPayment                CHAR(1)
)NOT FINAL;

CREATE TABLE Customers OF Customer (NIF PRIMARY KEY);

CREATE OR REPLACE TYPE MorningCustomer UNDER Customer(
)FINAL;

CREATE OR REPLACE TYPE GlobalCustomer UNDER Customer(
)FINAL;

CREATE OR REPLACE TYPE SwimmingCustomer UNDER Customer(
levelSwiming NUMBER
)FINAL;

CREATE OR REPLACE TYPE courseCustArray AS VARRAY(50) OF REF Customer;
CREATE OR REPLACE TYPE groupCustArray AS VARRAY(10) OF REF Customer;

CREATE OR REPLACE TYPE MonitorStaff UNDER Person(
cEmployee VARCHAR(10),
contractDate            DATE,
MonitorSpecialization tSpecialization
)FINAL;

CREATE TABLE tMonitorStaff OF MonitorStaff(cEmployee PRIMARY KEY)
OBJECT IDENTIFIER PRIMARY KEY
NESTED TABLE MonitorSpecialization STORE AS specializationList;

CREATE OR REPLACE TYPE TrainingCourse AS OBJECT(
codeTC                  VARCHAR(8),
name VARCHAR(60),
totalHours              NUMBER,
weekDays                VARCHAR(200),
tCourseSpec             specializationArray,
tCourseCust             courseCustArray
);

CREATE TABLE TrainingCourses OF TrainingCourse (codeTC PRIMARY KEY);

CREATE OR REPLACE TYPE TrainingGroup AS OBJECT(
codeTG                  VARCHAR(8),
name VARCHAR(60),
price                   VARCHAR(4),
custList                groupCustArray,
monitor                 REF MonitorStaff
);
```

```
CREATE TABLE TrainingGroups OF TrainingGroup (codeTG PRIMARY KEY);

CREATE OR REPLACE TYPE Rooms AS OBJECT(
idRoom                  NUMBER,
name                    VARCHAR(60)
);

CREATE TABLE tRooms OF Rooms (idRoom PRIMARY KEY);

CREATE OR REPLACE TYPE AccessInformation AS OBJECT(
idAccessInfo            VARCHAR(5),
timeAccess              VARCHAR(12),
roomAcc                 REF Rooms,
personAcc               REF Customer
);

CREATE TABLE AccessInformations OF AccessInformation (idAccessInfo
PRIMARY KEY);
```

c) S'han creat els objectes, i en cas necessari, les seves taules d'objectes, dels Clients (*Customers*), dels monitors (*MonitorStaff*), de les especialitzacions (*Specializations*), els grups d'entrenament (*TrainingGroups*), els cursos (*TrainingCourses*) i la informació dels accessos (*AccessInformation*) i les sales a on es poden accedir (*Rooms*).

A partir del tipus Person (*Person*), s'ha fet una especialització mitjançant herències, per una banda els clients (*Customers)* i per altra banda els monitors (*TrainingMonitor*). A partir de l'objecte *Customer* s'ha definit els tres possibles clients utilitzant també herència, on s'hi ha inclòs els atributs específics de cada un (*MorningCustomer*, *SwimingCustomer*, *GlobalCustomer*).

S'han creat una estructura de VARRAY per tal de donar resposta a les limitacions de clients que poden participar en un curs, en un grup d'entrenament i també per les especialitzacions associades a un curs (*courseCustArray* , *groupCustArray* ,*specializationArray*) .

Per altra banda, s'ha utilitzat una estructura de NESTED TABLE per tal de donar resposta a les múltiples especialitzacions associades a un monitor (amb nombre no acotat).

Per tal d'enllaçar els diferents objectes, com per exemple el monitor assignat a un grup (*MonitorStaff*), s'han utilitzat les referències a objectes REF, sempre garantint la navegabilitat entre objectes.

Pregunta 3 (2.5 punts)

Enunciat

La pàgina web del gimnàs disposa d'un espai de notícies on els monitors publiquen notícies i els clients poden fer-hi comentaris. El sistema disposa de les taules següents:

```
CREATE TABLE newsPost (
postID INTEGER NOT NULL,
titlePost VARCHAR(100) NOT NULL,
contentPost VARCHAR(400),
authorPostID INTEGER NOT NULL,
CONSTRAINT postPK PRIMARY KEY (postID),
CONSTRAINT authorFK FOREIGN KEY(authorPostID) REFERENCES
authorPost(authorID)
);

CREATE TABLE authorPost (
authorID INTEGER NOT NULL,
name VARCHAR(100),
surname VARCHAR(200),
CONSTRAINT authorPK PRIMARY KEY(authorID)
);

CREATE TABLE commentsPost(
postID INTEGER NOT NULL,
commentPost VARCHAR(150) NOT NULL,
CONSTRAINT comentPK PRIMARY KEY (postID, commentPost),
CONSTRAINT commentPostFK FOREIGN KEY (postID) REFERENCES newsPost
(postID)
);
```

Es demana, sobre Oracle DB XE 11g2:

a) Crear els objectes necessaris per tal de transformar les taules anteriors al model *object relational*. Incloure sentències INSERT que permetin validar el correcte funcionament del model.

b) Crear una vista que proporcioni per cada una de les notícies el títol d'aquesta, el nom i cognoms de l'autor i el nombre de comentaris. La vista retornarà els resultats ordenats pel nombre de comentaris de major a menor, amb el format següent:

```
SELECT * FROM postWithComm;
```

PostTitle	NameAuthor	NumComments
Title	Surname, Name	10

Criteris d'avaluació

Els criteris d'avaluació que s'aplicaran en la correcció d'aquesta pregunta són els següents:

- Totes les preguntes tenen el mateix pes.
- Les preguntes no contestades no penalitzen.
- Les preguntes sense argumentació no seran avaluades.
- Es valorarà la qualitat de la resposta.
- Es valorarà que el codi SQL lliurat funcioni i pugui ser executar.

Solució

a) Per tal de resoldre l'enunciat cal crear tres tipus d'objectes, un per cada taula, especificant les referències necessàries entre aquests. També creem les taules necessàries, una per autors(*tAuthorPost*) i una per publicació (*tNewPost*).

```
CREATE OR REPLACE TYPE AuthorPost AS OBJECT(
authorID INTEGER,
name VARCHAR(100),
surname VARCHAR(200)
);
CREATE TABLE tAuthorPost OF AuthorPost (authorID PRIMARY KEY);

CREATE TYPE commentsList AS TABLE OF VARCHAR(150);

CREATE OR REPLACE TYPE NewPost AS OBJECT(
idPost INTEGER,
titlePost VARCHAR(100),
contentPost VARCHAR(400),
authorPostID REF AuthorPost,
comments commentsList
);

CREATE TABLE tNewPost OF NewPost (idPost PRIMARY KEY)
OBJECT IDENTIFIER PRIMARY KEY
NESTED TABLE comments STORE AS CommentListStore;
```

b) Caldrà definir primer el tipus d'objecte que ens permetran crear el format de sortida de la vista.

```
CREATE OR REPLACE TYPE moreComments AS OBJECT(
titlePost VARCHAR(100),
authorName VARCHAR(300),
numComents INTEGER
);
```

El següent pas serà crear la vista, per calcular el nombre de comentaris d'una notícia utilitzarem la funció CARDINALITY que ens retornarà el nombre d'elements de la *nested table* que conté els comentaris per cada notícia.

```
CREATE OR REPLACE VIEW postWithComm OF moreComments WITH OBJECT
IDENTIFIER(titlePost) AS
SELECT np.titlePost, np.AUTHORPOSTID.surname ||',
'||np.AUTHORPOSTID.name , CARDINALITY(np.COMMENTS) cardinal  FROM
    tNewPost np
    order by cardinal DESC;
```

A continuació es presenta la sentència per tal d'invocar la vista:

```
SELECT *
FROM postWithComm;
```

Recursos

Per tal de resoldre aquesta pràctica caldrà utilitzar els recursos que s'enumeren a continuació:

- Mòdul 2 de l'assignatura Arquitectura de Bases de Dades (únicament la secció 1 que fa referència l'extensió *object-relational*)
- Guia "*Database Object-Relational Developer's Guide*" publicada per Oracle: https://docs.oracle.com/database/121/ADOBJ/toc.htm
- Software: "*Oracle DB Express Edition 11g Release 2*"

Criteris de valoració

A l'enunciat de cada exercici s'indica el valor del mateix sobre la puntuació total de la pràctica. Aquesta activitat representa el 50% de la nota de pràctiques de l'assignatura i la seva realització és **obligatòria** per tal de superar l'assignatura.

No s'acceptaran ni es tindran en compte els lliuraments realitzats fora dels terminis indicats al calendari de l'aula.

Format i data de lliurament

Heu d'enviar la PR1 a la bústia de lliuraments com un arxiu .zip el nom del qual contingui el codi de l'assignatura, el vostre cognom i el vostre nom, així com la indicació de que es tracta de la PR1.

- Lliurament individual: Cal incloure el nom i cognoms a cadascun dels fitxers que lliureu.
- Lliurament en parella: Cal incloure el nom i cognom dels dos estudiants a cadascun dels fitxers que lliureu. Únicament un dels components ha de fer el lliurament al Campus.

A l'arxiu .zip s'ha d'incloure:

- Un document PDF que contingui les respostes dels exercicis 1, 2 i 3.
- Un fitxer SQL que contingui la resposta a l'exercici 2.b) i un altre fitxer SQL que contingui la resposta a l'exercici 3.
- Un fitxer SQL que contingui les sentències DROP de l'exercici 2.b) i un altre per a l'exercici 3.
- Un fitxer SQL que contingui les sentències INSERT necessàries per a demostrar la correcta creació de les estructures de l'exercici 2.b) i un altre per l'exercici 3. No més de 2 files per taula.

La data límit per lliurar la PR1 és el **20/10/2019**.