

## Soluciones del Examen Ordinario de Prácticas de Bases de Datos Mayo'08

Se desea crear una base de datos para almacenar información sobre los autores de artículos científicos publicados en revistas internacionales:

**Autor** (CodAutor, Nombre, Apellido1, Filiación, Dirección, Telefono, email)

```
create table Autor (
CodAutor varchar2(10) not null,
Nombre varchar2(30) not null,
Apellido1 varchar2(50) not null,
Filiación varchar2(50) not null,
Dirección varchar2 (50) not null,
email varchar2(50) not null,
constraint AutorPK PRIMARY KEY (CodAutor)
);
```

Tabla Autor:

Nombre	¿Nulo?	Tipo
CODAUTOR	NOT NULL	VARCHAR2(10)
NOMBRE	NOT NULL	VARCHAR2(30)
APELLIDO1	NOT NULL	VARCHAR2(50)
FILIACIÓN	NOT NULL	VARCHAR2(50)
DIRECCION	NOT NULL	VARCHAR2(50)
EMAIL	NOT NULL	VARCHAR2(50)

**Revista** (CodRevista, Nombre)

```
create table Revista (
CodRevista varchar2(10) not null,
Nombre varchar2(50) not null,
constraint RevistaPK Primary Key (CodRevista)
);
```

Nombre	¿Nulo?	Tipo
CODREVISTA	NOT NULL	VARCHAR2(10)
NOMBRE	NOT NULL	VARCHAR2(50)

**Artículo** (CodArticulo, TituloArticulo, Año, Revista, Volumen, Numero, PaginaInicial, PaginaFinal)

```
create table Articulo (
CodArticulo varchar2(10) not null,
TituloArticulo varchar2(50) not null,
Anho number not null,
Revista varchar2(10) not null,
Volumen number,
Numero number,
PaginaInicial number,
PaginaFinal number,
constraint ArticuloPK Primary Key (CodArticulo),
Constraint RevistaFK foreign key (Revista) references Revista
```

```
);
```

Nombre	¿Nulo?	Tipo
CODARTICULO	NOT NULL	VARCHAR2(10)
TITULOARTICULO	NOT NULL	VARCHAR2(50)
ANHO	NOT NULL	NUMBER
REVISTA	NOT NULL	VARCHAR2(10)
VOLUMEN		NUMBER
NUMERO		NUMBER
PAGINAINICIAL		NUMBER
PAGINAFINAL		NUMBER

**1. (1 punto) Escribir el código SQL para crear la tabla AutorEscribeArticulo**

```
create table AutorEscribeArticulo (
CodAutor varchar2(10) not null,
CodArticulo varchar2(10) not null,
OrdenAutor number not null,
constraint AEA_FK_Autor Foreign Key (CodAutor) references Autor,
constraint AEA_FK_Articulo Foreign Key (CodArticulo) references Articulo
);
```

Nombre	¿Nulo?	Tipo
CODAUTOR	NOT NULL	VARCHAR2(10)
CODARTICULO	NOT NULL	VARCHAR2(10)
ORDENAUTOR	NOT NULL	NUMBER

```
select constraint_name, constraint_type from user_constraints
where table_name='AUTORESCRIBEARTICULO';
```

CONSTRAINT_NAME	C
SYS_C0045014	C
SYS_C0045015	C
SYS_C0045016	C
AEA_FK_AUTOR	R
AEA_FK_ARTICULO	R

**2. (1 punto) Añadir a la tabla Revista dos campos nuevos: uno booleano llamado EsJCR y otro double llamado FactorImpacto.**

Como en Oracle no hay tipo boolean tenemos que simularlo con '0'/'1', 'V'/'F' ...

```
alter table Revista add (
EsJCR number (1) default 0,
FactorImpacto decimal,
constraint ValoresEsJCR check (EsJCR in (0,1))
);
```

Otra opción hubiera sido haber tratado el boolean como carácter, y la sentencia hubiera sido:

```
alter table Revista add (
EsJCR varchar2 (1) default 'N',
FactorImpacto number (6,4),
constraint ValoresEsJCR check (EsJCR in ('N','S'))
);
```

Nombre	¿Nulo?	Tipo
CODREVISTA	NOT NULL	VARCHAR2(10)
NOMBRE	NOT NULL	VARCHAR2(50)
ESJCR		NUMBER(1)
FACTORIMPACTO		NUMBER(6,4)

**3. (2 puntos) Suponiendo las tablas sin datos añadir el artículo:**

- a. **Título del Artículo: Estudios Empíricos de Bases de Datos**
- b. **Autores: Manuel Serrano y Coral Calero**
- c. **Revista: Quality Journal, Volumen 16, Número 1, Páginas: 79-106, Año 2008**
- d. **Filiación de los autores: UCLM**
- e. **Dirección: E.S. Informatica, 13071, España**
- f. **Correos: [Manuel.Serrano@uclm.es](mailto:Manuel.Serrano@uclm.es), [Coral.Calero@uclm.es](mailto:Coral.Calero@uclm.es)**
- g. **Otros datos: la Revista es JCR y tiene un factor de impacto de 0,4.**

Dada las integridades referenciales, tenemos que introducir los datos primero en aquellas tablas que no reverencien a ninguna otra, teniendo en cuenta lo que nos dice el enunciado (que no hay datos). En nuestro caso o introducimos primero los autores, o introducimos primero la revista

```
insert into Autor values('mserrano', 'Manuel', 'Serrano', 'UCLM', 'E.S.
Informatica, 13071, España', 'Manuel.Serrano@uclm.es');

insert into Autor values('ccalero', 'Coral', 'Calero', 'UCLM', 'E.S.
Informatica, 13071, España', 'Coral.Calero@uclm.es');
```

Que deja la tabla autor como sigue:

CODAUTOR	NOMBRE	APELLIDO1	FILIACION	DIRECCION	EMAIL
mserrano	Manuel	Serrano	UCLM	E.S. Informatica, 13071, España	Manuel.Serrano@uclm.es
ccalero	Coral	Calero	UCLM	E.S. Informatica, 13071, España	Coral.Calero@uclm.es

Insertamos la revista:

```
insert into Revista values ('QJ', 'Quality Journal', 1, 0.4);
```

CODREVISTA	NOMBRE	ESJCR	FACTORIMPACTO
QJ	Quality Journal	1	,4

Ahora insertamos el artículo, teniendo en cuenta que en el campo revista tenemos que insertar el código revista, que en nuestro caso es 'QJ'. Insertamos los valores para el artículo antes que el hecho de qué autor ha escrito qué artículo, porque esta última tabla apunta a Artículo y se podría violar la integridad referencial si no existe antes en Artículo los datos del artículo que pretendemos guardar.

```
insert into Articulo values ('EEBD','Estudios Empíricos en Bases de Datos',
2008, 'QJ', 16, 1, 79, 106)
```

CODARTICULO	TITULOARTICULO	ANHO	REVISTA	VOLUMEN	NUMERO	PAGINAINICIAL	PAGINAFINAL
EEBD	Estudios Empíricos en Bases de Datos	2008	QJ	16	1	79	106

Ahora hacemos las inserciones correspondientes al hecho de que los dos autores han escrito ese artículo. Suponemos que el orden en el que nos dan los autores es el mismo en el que aparecen en el artículo.

```
insert into AutorEscribeArticulo values ('mserrano', 'EEBD', 1);
insert into AutorEscribeArticulo values ('ccalero', 'EEBD', 2);
```

CODAUTOR	CODARTICULO	ORDENAUTOR
mserrano	EEBD	1
ccalero	EEBD	2

4. (1 punto) Seleccionar el nombre de las revistas que siendo JCR tengan un factor de impacto entre 1 y 2.

Consideremos los siguientes datos en la tabla Revistas:

CODREVISTA	NOMBRE	ESJCR	FACTORIMPACTO
IJIQ	International Journal on Information Quality	0	0
QJ	Quality Journal	1	,4
IMandP	Information Management and Processing	1	1,573

La consulta que nos piden

```
select Nombre from Revista where EsJCR = 1 and FactorImpacto >= 1 and
FactorImpacto <= 2
```

devolverá el resultado

NOMBRE
Information Management and Processing

5. (1,5 puntos) Seleccionar todos los artículos (Título de artículo, Nombre de Revista y Año) en los que Mario Piattini no aparezca como primer autor.

De los datos contenidos en artículos:

CODARTICULO	TITULOARTICULO	ANHO	REVISTA	VOLUMEN	NUMERO	PAGINAINICIAL	PAGINAFINAL
EEBD	Estudios Empíricos en Bases de Datos	2008	QJ	16	1	79	106
DQenBD	Calidad en Bases de Datos	2009	IJIQ	5	1	15	35
DQenWeb	Calidad en Aplicaciones Web	2009	IMandP	45	8	120	135
EEWEB	Estudios Empíricos en Aplicaciones Web	2009	QJ	25	2	80	96

Y de los contenidos en AutorEscribeArticulo

CODAUTOR	CODARTICULO	ORDENAUTOR
mserrano	EEBD	1
ccalero	EEBD	2
<b>mpiattini</b>	<b>DQenWeb</b>	<b>1</b>
icaballero	DQenWeb	2
ccalero	DQenWeb	3
ccalero	DQenBD	1
icaballero	DQenBD	2
<b>mpiattini</b>	<b>DQenBD</b>	<b>3</b>
mserrano	DQenBD	4

La consulta que nos piden:

```
select Articulo.TituloArticulo, Revista.Nombre, Articulo.Anho
from Articulo, AutorEscribeArticulo, Autor, Revista
where Articulo.CodArticulo = AutorEscribeArticulo.CodArticulo
and Autor.CodAutor = AutorEscribeArticulo.CodAutor
and Articulo.Revista = Revista.CodRevista
and Autor.Nombre = 'Mario'
and Autor.Apellido1='Piattini'
and AutorEscribeArticulo.OrdenAutor <> 1
```

Produce los siguientes resultados:

TITULOARTICULO	NOMBRE	ANHO
Calidad en Bases de Datos	International Journal on Information Quality	2009

6. (1 punto) Crear una vista que para cada autor y año nos diga el número de artículos publicados en revistas JCR.

Como se recomendó en la última sesión de de prácticas, se recomienda ir haciendo aproximaciones sucesivas a la solución.

En este caso la primera aproximación podría ser una consulta que nos indicara el número de artículos publicados por autor y por año.

Así se tendría que la consulta sobre la que montaremos la vista

```
select codautor, anho, count(AutorEscribeArticulo.codarticulo)
from AutorEscribeArticulo, Articulo, Revista
where AutorEscribeArticulo.CodArticulo = Articulo.CodArticulo
```

```
and Articulo.Revista = Revista.CodRevista
and Revista.EsJCR = 1
group by (codautor,anho);
```

generaría los siguientes resultados:

CODAUTOR	ANHO	COUNT(AUTORESCRIBEARTICULO.CODARTICULO)
ccalero	2008	1
ccalero	2009	1
mserrano	2008	1
mpiattini	2009	1
icaballero	2009	1

Con esta consulta tenemos es más sencillo montar la vista que se nos pide:

```
create or replace view JCRporAutorYAnho as
select AutorEscribeArticulo.codautor, Articulo.anho,
count(AutorEscribeArticulo.codarticulo) as NumeroArticulos
from AutorEscribeArticulo, Articulo, Revista
where AutorEscribeArticulo.CodArticulo = Articulo.CodArticulo
and Articulo.Revista = Revista.CodRevista
and Revista.EsJCR = 1
group by (codautor,anho);
```

Obsérvese que es importante ponerle una nombre a la expresión encargada de contar el número de artículos.

Si quisiéramos utilizarla, ser haría de la forma

```
select * from JCRporAutorYAnho
where CodAutor = 'ccalero' and anho = 2008
```

y obtendríamos los resultados:

CODAUTOR	ANHO	NUMEROARTICULOS
ccalero	2008	1

O incluso:

```
select * from JCRporAutorYAnho
where CodAutor = 'ccalero'
order by anho
```

CODAUTOR	ANHO	NUMEROARTICULOS
ccalero	2008	1
ccalero	2009	1

7. (2,5 puntos) Desarrollar un procedimiento en PL/SQL que permita mostrar los datos de los artículos publicados en el formato siguiente: Apellido1, N., Apellido1, N., ... (Año). Título, Publicación, Volumen (Número), PáginaInicial-PáginaFinal.

**Ejemplo: Serrano, M., Calero, C., (2008). Estudios Empíricos en Bases de Datos. Quality Journal. 16(1). 79-106.**

Para hacer este ejercicio es preciso recurrir a la creación de funciones en PL/SQL.

Nuevamente se propone seguir la estrategia recomendada en la última sesión de clases de teoría de ir haciendo aproximaciones a la solución.

Lo primero es identificar dónde están los datos que nos piden. Como puede observarse hay que combinar todas las tablas del problema.

En concreto se piden listar todos los autores por orden de aparición en el artículo. Para eso se puede elaborar la consulta

```
select Nombre, Apellido1, CodArticulo, OrdenAutor from
Autor, AutorEscribeArticulo
where Autor.CodAutor= AutorEscribeArticulo.CodAutor
order by CodArticulo, OrdenAutor
```

que permitiría mostrar resultados como sigue:

NOMBRE	APELLIDO1	CODARTICUL	ORDENAUTOR
Coral	Calero	DQenBD	1
Ismael	Caballero	DQenBD	2
Mario	Piattini	DQenBD	3
Manuel	Serrano	DQenBD	4
Mario	Piattini	DQenWeb	1
Ismael	Caballero	DQenWeb	2
Coral	Calero	DQenWeb	3
Manuel	Serrano	EEBD	1
Coral	Calero	EEBD	2

Sobre esta consulta se podría montar, para simplificar el trabajo una vista llamada NombreAutoresDeUnArticulo

```
create or replace view NombreAutoresDeUnArticulo
as
select Nombre, Apellido1, CodArticulo, OrdenAutor from
Autor, AutorEscribeArticulo
where Autor.CodAutor= AutorEscribeArticulo.CodAutor
order by CodArticulo, OrdenAutor
```

Sobre esta vista, podríamos manejar cursores parametrizados que sirvieran de base para crear posteriormente una función que devolviera una cadena de texto con los autores para un determinado artículo en el mismo orden en el que aparecen.

```
SET SERVEROUTPUT ON;
DECLARE
v_Nombre NombreAutoresDeUnArticulo.Nombre%TYPE;
v_Apellido1 NombreAutoresDeUnArticulo.Apellido1%TYPE;
ListadoAutores Varchar2(255);
```

```

CURSOR c_Autores(v_COD NombreAutoresDeUnArticulo.CodArticulo%TYPE) IS
    SELECT Nombre, Apellido1
    FROM NombreAutoresDeUnArticulo
    WHERE CodArticulo= v_COD
    ORDER BY OrdenAutor;

BEGIN
OPEN c_Autores ('DQenBD');
LOOP
FETCH c_Autores INTO v_Nombre, v_Apellido1;
EXIT WHEN c_Autores%NOTFOUND;
ListadoAutores:= ListadoAutores || v_Apellido1 || ', ' ||
substr(v_Nombre,1,1) || ', ';
END LOOP;
CLOSE c_Autores;
DBMS_OUTPUT.PUT_LINE(ListadoAutores);

END;
/

```

La función ListarAutoresDeUnArtículo correspondiente podría crearse así:

```

SET SERVEROUTPUT ON;

create or replace Function ListarAutoresDeUnArticulo (El_CodArticulo IN
NombreAutoresDeUnArticulo.CodArticulo%TYPE) RETURN varchar2

AS

v_Nombre NombreAutoresDeUnArticulo.Nombre%TYPE;
v_Apellido1 NombreAutoresDeUnArticulo.Apellido1%TYPE;
ListadoAutores varchar2(255):='';

CURSOR c_Autores(v_COD NombreAutoresDeUnArticulo.CodArticulo%TYPE) IS
    SELECT Nombre, Apellido1
    FROM NombreAutoresDeUnArticulo
    WHERE CodArticulo= v_COD
    ORDER BY OrdenAutor;

BEGIN
OPEN c_Autores (El_CodArticulo);
LOOP
FETCH c_Autores INTO v_Nombre, v_Apellido1;
EXIT WHEN c_Autores%NOTFOUND;
ListadoAutores:= ListadoAutores || v_Apellido1 || ', ' ||
substr(v_Nombre,1,1) || ', ';
END LOOP;
CLOSE c_Autores;
DBMS_OUTPUT.PUT_LINE(ListadoAutores);

Return ListadoAutores;
END ListarAutoresDeUnArticulo;

```

Para terminar podríamos crear una última función que es la que resuelve el ejercicio, en el que utilizando la función anterior cree

```

SET SERVEROUTPUT ON;

```

```
CREATE or REPLACE Function ListarDatosDeUnArticulo (El_CodArticulo IN
NombreAutoresDeUnArticulo.CodArticulo%TYPE) RETURN Varchar2

AS

v_TituloArticulo Articulo.TituloArticulo%TYPE;
v_Anho Articulo.Anho%TYPE;
v_Revista Revista.Nombre%TYPE;
v_Volumen Articulo.Volumen%TYPE;
v_Numero Articulo.Numero%TYPE;
v_PaginaInicial Articulo.PaginaInicial%TYPE;
v_PaginaFinal Articulo.PaginaFinal%TYPE;
DatosTotalesDeUnArticulo varchar2(255):='';

BEGIN

SELECT
    TituloArticulo,
    Anho,
    Revista,
    Volumen,
    Numero,
    PaginaInicial,
    PaginaFinal
INTO
    v_TituloArticulo,
    v_Anho,
    v_Revista,
    v_Volumen,
    v_Numero,
    v_Paginainicial,
    v_Paginafinal
FROM Articulo, Revista
WHERE Articulo.Revista = Revista.CodRevista
    and Articulo.CodArticulo = El_CodArticulo ;

DatosTotalesDeUnArticulo := ListarAutoresDeUnArticulo (El_CodArticulo) || '('
|| v_Anho || '). ' || v_TituloArticulo || '. ' || v_Revista || '. (' ||
v_Numero || '.) ' || v_PaginaInicial || '-' || v_PaginaFinal || '.';

RETURN DatosTotalesDeUnArticulo;
END ListarDatosDeUnArticulo;
```