



PROCESO SELECCIÓN TÉCNICO/A INFORMÁTICO/A 2019.

EJERCICIO PRÁCTICO: BASES DE DATOS

NORMAS PARA LA CORRECTA REALIZACIÓN DEL EJERCICIO:

Lea detenidamente el contenido del supuesto y conteste convenientemente a cada una de las preguntas planteadas.

Para las preguntas de desarrollo por favor cññase al espacio propuesto para la respuesta, siendo lo más claro y preciso posible, evitando en la medida de lo posible tachones y borrones.

Para las preguntas que ofrecen ya las posibles soluciones, marce con una X la que considere correcta.

La valoración total de este ejercicio es de 20 puntos, a razón de los puntos indicados en el enunciado de cada pregunta contestada correctamente, siendo necesario alcanzar un mínimo de 10 puntos para superar el ejercicio.



PROCESO SELECCIÓN TÉCNICO/A INFORMÁTICO/A 2019.

EJERCICIO PRÁCTICO: BASES DE DATOS

DESCRIPCIÓN DEL ESCENARIO:

Se nos plantea la necesidad de llevar el control de un pequeño inventario del material informático (no fungible) del Ayuntamiento, permitiéndonos con él almacenar cada artículo con información de su nombre, tipo, descripción, fecha de compra, fecha finalización de garantía y ubicación.

Este control de existencias de material también nos deberá permitir la realización de préstamos del que no esté asignado a ninguna ubicación/departamento, debiendo dejar reflejado en cada préstamo la fecha de inicio, la de fin y el código del empleado que realice la solicitud, así como una descripción del motivo de esta.

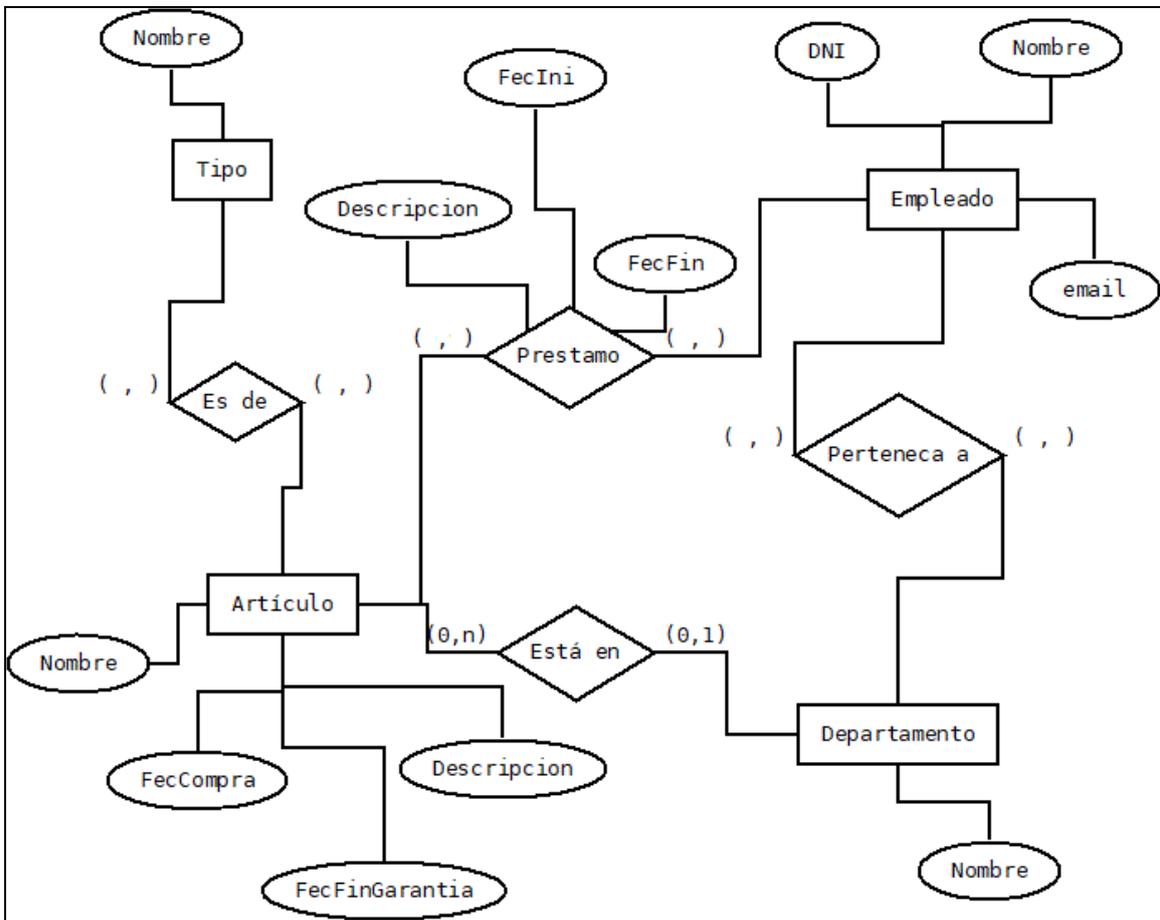
Del Empleado, además de estar relacionado con un departamento, deberemos almacenar exclusivamente el DNI, el nombre y el email.

Se considerará que un artículo está disponible (stock) si no tiene asignada ninguna ubicación y a su vez no está relacionado con un préstamo vigente.

En base a la descripción de requisitos del enunciado se nos proporciona el modelo conceptual que se muestra a continuación en la figura 1:



Figura 1.- Modelo conceptual



J.1.- (0,7 puntos) ¿Cuál sería la cardinalidad más apropiada para la relación “Es de” del modelo?

- a) (1,1) ◇ (0,1)
- b) (1,1) ◇ (0,n)
- c) (0,1) ◇ (1,n)

J.2.- (0,7 puntos) ¿Cuál sería la cardinalidad más apropiada para la relación “Prestamo” del modelo?

- a) (1,n) ◇ (1,1)
- b) (1,1) ◇ (0,n)
- c) (0,n) ◇ (1,1)



J.3.- (0,7 puntos) ¿Cuál sería la cardinalidad más apropiada para la relación “Pertenece a” del modelo?

- a) (1,1) ◇ (1,n)
- b) (1,n) ◇ (1,1)
- c) (0,1) ◇ (1,1)

J.4.- (0,7 puntos) Una vez realizado el modelo Entidad-Relación, podemos decir que:

- a) Préstamo es una relación con atributos que se tendrán que reflejar en las entidades Artículo y Empleado, cuando realicemos el modelo lógico.
- b) **Préstamo es una relación con atributos que se transformará en una tabla a partir del modelo lógico.**
- c) Préstamo es una relación sin atributos que se convertirá en una tabla en el modelo físico.

J.5.- (0,7 puntos) ¿Con cuál de las siguientes herramientas podríamos realizar el modelo lógico de nuestra base de datos?

- a) **Oracle Data Modeler**
- b) Oracle Developer
- c) Oracle DBModeler

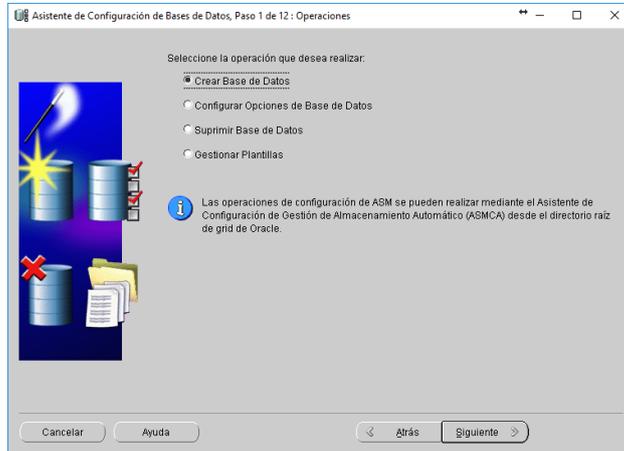
DISEÑO FISICO (Todos los pasos correspondientes a este apartado se entenderán realizados en el SGBD Oracle 11g).

J.6.- (0,7 puntos) Tras configurar el Listener, este escuchará por defecto en el puerto

- a) 1158
- b) **1521**
- c) 1156



J.7.- (0,7 puntos) ¿Cuál es el comando que podríamos utilizar desde el símbolo de sistema de Windows para abrir el “Asistente de Configuración de Bases de Datos” desde el que podremos crear la BBDD?



- a) dbca
- b) netca
- c) dbnet

J.8.- (0,7 puntos) ¿Dónde guardará, el “Asistente de Configuración de Bases de Datos” de la pregunta anterior, en caso de seleccionar la opción correspondiente, los scripts de creación de la BBDD? (considerando que DB_NAME se corresponde con el nombre de la instancia)

- a) En %ORACLE_BASE%\oradata\DB_NAME\scripts
- b) En %ORACLE_BASE%\oradata\admin\DB_NAME\scripts
- c) En %ORACLE_BASE%\admin\DB_NAME\scripts

J.9.- (0,7 puntos) Si creáramos la base de datos con el comando CREATE DATABASE, ¿en qué modo tendría que estar la instancia?

- a) Arrancada en modo “NOMOUNT”.
- b) Arrancada en modo “MOUNT”.
- c) Arrancada en modo “OPEN”.

J.10.- (0,7 puntos) ¿Qué parámetro de instancia no podremos cambiar en la Base de Datos después de haberla creado?

- a) Si la instancia está en “NOMOUNT” todos pueden cambiarse.
- b) Control Files.
- c) DB_BLOCK_SIZE.



J.11.- (0,7 puntos) ¿Qué ficheros genera el script de comandos SQL dentro del proceso de creación de base de datos?

- a) Ficheros de control y ficheros redo log.
- b) Datafiles de los espacios de tablas de SYSAUX y de SYSTEM.
- c) **Los indicados en las opciones a) y b) + otros.**

J.12.- (0,7 puntos) Una vez creada la BBDD, si deseamos acceder a ella desde Enterprise Manager, ¿cuál sería la ruta que podríamos utilizar?

- a) <https://localhost:1521/em>
- b) **https://localhost:1158/em**
- c) http://localhost:1158/em

J.13.- (0,7 puntos) Para crear un tablespace denominado INVENTARIO que sea permanente, que tenga un tamaño inicial de 20MB e incrementos automáticos de 5MB hasta un máximo de 100MB, necesitamos ejecutar:

- a) CREATE TABLESPACE "INVENTARIO" DATAFILE
"DATOS_INVENTARIO.DBF" SIZE 20M NEXT 5M MAXSIZE 100M;
- b) CREATE TABLESPACE "INVENTARIO" IN
'DATOS_INVENTARIO.DBF' SIZE (20M, 5M,100M);
- c) **CREATE TABLESPACE "INVENTARIO" DATAFILE
'DATOS_INVENTARIO.DBF' SIZE 20M AUTOEXTEND ON NEXT 5M
MAXSIZE 100M;**

J.14.- (0,7 puntos) ¿Qué necesitamos ejecutar para crear un usuario llamado US_INVENTARIO que tenga como clave inicial "123456" y que su tablespace por defecto sea el que hemos creado en el paso anterior?

- a) CREATE USER US_INVENTARIO IDENTIFIED BY "123456"
TABLESPACE "INVENTARIO" TEMPORARY TABLESPACE "TEMP";
- b) **CREATE USER US_INVENTARIO IDENTIFIED BY "123456"
DEFAULT TABLESPACE "INVENTARIO" TEMPORARY
TABLESPACE "TEMP";**
- c) CREATE USER US_INVENTARIO PASSWORD BY "123456"
TABLESPACE "INVENTARIO" TEMPORARY TABLESPACE "TEMP";



J.15.- (0,7 puntos) ¿Qué sentencia/sentencias tendríamos que ejecutar para dar permisos a nuestro usuario para crear (Triggers, Tablas, Vistas, Secuencias) y permitirle el inicio de sesión?

- a) **GRANT CREATE TRIGGER TO "US_INVENTARIO" ;
GRANT CREATE VIEW TO "US_INVENTARIO" ;
GRANT CREATE SESSION TO "US_INVENTARIO" ;
GRANT CREATE TABLE TO "US_INVENTARIO" ;
GRANT CREATE SEQUENCE TO "US_INVENTARIO" ;**
- b) **GRANT CREATE (TRIGGER, VIEW, SESSION, TABLE, SEQUENCE)
TO "US_INVENTARIO";**
- c) Las opciones a) y b) son correctas.

J.16.- (0,7 puntos) Ya hemos creado las tablas y queremos establecer la clave primaria de la tabla ARTICULO, ¿Cuál sería la sentencia correcta?

- a) **ALTER TABLE articulo ADD PRIMARY KEY (idarticulo);**
- b) **ALTER TABLE articulo ADD CONSTRAINT articulo_pk PRIMARY
KEY (idarticulo);**
- c) **ALTER COLUMN OF articulo ADD CONSTRAINT articulo_pk PRIMARY
KEY (idarticulo,tipo_idtipo);**

J.17.- (0,7 puntos) En caso de que después de haber creado todas las tablas, con sus claves foráneas y primarias, quisiésemos borrar la tabla TIPO ¿Cuál sería la sentencia que nos lo permitiría?

- a) **DROP TABLE TIPO CASCADE CONSTRAINTS;**
- b) **DROP TABLE TIPO CASCADE;**
- c) **DROP TABLE TIPO ON CASCADE;**



J.18.- (1,2 puntos) Tras el borrado de la tabla anterior ejecutamos un script con las siguientes sentencias:

```
CREATE TABLE tipo (  
  idtipo INTEGER NOT NULL,  
  nombre VARCHAR2(25) NOT NULL  
);  
ALTER TABLE articulo ADD CONSTRAINT articulo_tipo_fk FOREIGN KEY (  
  tipo_idtipo )  
  REFERENCES tipo ( idtipo )  
  NOT DEFERRABLE;
```

Describe brevemente lo que ocurre tras la ejecución del script:

Oracle nos devuelve un error indicándonos que no hay una clave única o primaria en la tabla referenciada con la cláusula REFERENCES, ya que no se ha creado la PRIMARY KEY de la tabla TIPO.

J.19.- (1,2 puntos) Dada la creación del siguiente Trigger:

```
create or replace trigger DIS_TIPO  
  before insert on "US_INVENTARIO"."TIPO"  
  for each row  
begin  
  if inserting then  
    if :NEW."IDTIPO" is null then  
      select SEC_TIPO.nextval into :NEW."IDTIPO" from dual;  
    end if;  
  end if;  
end;
```

¿Qué nos aporta y qué necesitamos crear previamente para que funcione de la manera esperada?

Sería necesario crear una secuencia, denominada SEC_TIPO, que nos proporcione un valor autonumérico. Tras eso, este trigger introducirá el valor generado por la secuencia en el campo IDTIPO de la tabla TIPO, cada vez que creemos un nuevo registro.



J.20.- (0,7 puntos) Tras intentar introducir datos en una de las tablas, con el usuario US_INVENTARIO, Oracle nos devuelve el siguiente error:

ORA-01950: no existen privilegios en tablespace 'INVENTARIO'

¿Qué sería necesario ejecutar, con el usuario system, para obtener los privilegios necesarios?

- a) GRANT SYSDBA US_INVENTARIO quota unlimited on INVENTARIO;
- b) GRANT US_INVENTARIO quota unlimited on INVENTARIO;
- c) **ALTER USER US_INVENTARIO quota unlimited on INVENTARIO;**

J.21.- (0,7 puntos) Tras haber creado las tablas nos damos cuenta de que al campo idDepartamento, que sería la clave foránea de los departamentos en la Tabla ARTICULO, le hemos dado un nombre demasiado largo y queremos cambiarlo. ¿Cuál sería la sentencia correcta para hacerlo?

- a) ALTER TABLE ARTICULO FROM COLUMN DEPARTAMENTO_IDDEPARTAMENTO TO IDDEPARTAMENTO;
- b) **ALTER TABLE ARTICULO RENAME COLUMN DEPARTAMENTO_IDDEPARTAMENTO TO IDDEPARTAMENTO;**
- c) ALTER COLUMN INTO ARTICULO FROM DEPARTAMENTO_IDDEPARTAMENTO TO IDDEPARTAMENTO;

J.22.- (1,2 puntos) Queremos crear una vista que nos muestre la fecha de garantía de cada artículo. Dada la siguiente descripción de la tabla ARTICULO:

Nombre	¿Nulo?	Tipo
-----	-----	-----
IDARTICULO	NOT NULL	NUMBER(38)
NOMBRE	NOT NULL	VARCHAR2(15 CHAR)
FECCOMPRA		DATE
FECFINGARANTIA		DATE
DESCRIPCION		VARCHAR2(100 CHAR)
TIPO_IDTIPO	NOT NULL	NUMBER(38)
IDDEPARTAMENTO		NUMBER(38)

¿Cuál sería la sentencia correcta?

- a) CREATE OR REPLACE VIEW GARANTIA_ARTICULOS AS SELECT NOMBRE, DESCRIPCION, FECFINGARANTIA FROM ARTICULO;
- b) CREATE VIEW GARANTIA_ARTICULOS LIKE (SELECT NOMBRE, DESCRIPCION, FECFINGARANTIA FROM ARTICULO);
- c) **CREATE OR REPLACE VIEW GARANTIA_ARTICULOS AS SELECT NOMBRE, DESCRIPCION, FECFINGARANTIA FROM ARTICULO;**



J.23.- (0,7 puntos) Ahora nos interesa crear una función y un procedimiento almacenado en la BBDD desde el usuario activo (US_INVENTARIO). ¿Qué sentencia o sentencias tendríamos que ejecutar desde un usuario con privilegios para disponer de los permisos mínimos necesarios para crear estos objetos en nuestro esquema?

- a) **GRANT CREATE PROCEDURE TO USER "US_INVENTARIO";**
- b) **GRANT CREATE ANY PROCEDURE TO "US_INVENTARIO";**
- c) **GRANT CREATE ANY PROCEDURE TO USER "US_INVENTARIO";**

J.24.- (1,2 puntos) Dado el código de la siguiente función y entendiendo que se corresponde con un objeto almacenado en nuestra base de datos:

```
create or replace FUNCTION disponible (  
    art NUMBER  
    ) RETURN BOOLEAN  
  
IS  
    id_art prestamo.idarticulo%type;  
BEGIN  
    SELECT UNIQUE idarticulo into id_art FROM prestamo  
    where idarticulo = art  
    and idarticulo in (select idarticulo from articulo  
        where iddepartamento is null)  
    and idarticulo not in (SELECT IDARTICULO FROM PRESTAMO  
        WHERE FECFIN >= SYSDATE  
        AND IDARTICULO = art);  
    RETURN true;  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        return false;  
    WHEN OTHERS THEN  
        RAISE_APPLICATION_ERROR(-20000,'ERROR'|| sqlcode);  
END;
```

¿Qué devolvería el siguiente código PL/SQL?

```
BEGIN  
    IF disponible(3) THEN  
        DBMS_OUTPUT.PUT_LINE('Disponible');  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('No Disponible');  
    END IF;  
END;
```

- a) **“Disponible”, en caso de que el artículo con idarticulo=3 no esté asignado a ningún departamento y no tengamos ningún préstamo vigente.**
- b) **“Disponible”, en caso de que el artículo con idarticulo=3 esté asignado a un departamento y no tengamos ningún préstamo vigente.**
- c) **Un error con código 20000, ya que no se ha definido una variable en la función para devolver el valor de retorno.**



J.25.- (1,2 puntos) Suponiendo que la función anterior está bien escrita, seguidamente se nos plantea la necesidad de crear un procedimiento para realizar préstamos de los artículos disponibles.

Procedimiento Alta_Prestamo:

```
create or replace PROCEDURE Alta_Prestamo (descri VARCHAR2,ffin DATE, idemp
NUMBER, idart NUMBER)
as
  disp BOOLEAN;
  errorNoDisponible EXCEPTION;
  fechalni prestamo.fecini%type;
  idPrest NUMBER;
BEGIN
  disp := disponible(idart);
  IF disp THEN
    SELECT MAX(IDPRESTAMO) into idPrest FROM PRESTAMO;
    idPrest:=idPrest+1;

    SELECT sysdate into fechalni FROM DUAL;
    INSERT INTO PRESTAMO VALUES (idPrest,descri,fechalni,ffin,idemp,idart);
    commit;
  ELSE
    RAISE errorNoDisponible;
  END IF;

EXCEPTION
  WHEN errorNoDisponible THEN
    RAISE_APPLICATION_ERROR(-20001,'ERROR, El artículo introducido no está
disponible');
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20002,'ERROR, EL PRODUCTO NO EXISTE');
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20000,'ERROR'|| sqlcode);
END;
```

Escribe un código PL/SQL sencillo (máximo 3 líneas) que permita la inserción de un nuevo PRESTAMO haciendo uso del procedimiento “Alta_Prestamo” y que cree un préstamo con los siguientes datos:

Descripción: ‘Motivo de ejemplo’

Empleado: 3

Fecha Inicio: la fecha actual

Artículo: 4

Fecha Fin: 20/06/2019

```
BEGIN
ALTA_PRESTAMO('Motivo ejemplo','20/06/2019',3,4);
END;
```