

Examen-Junio-2018-2019.pdf



Juandf03



Administración de Bases de Datos



3º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**



Estamos de
Aniversario

De la universidad al
mercado laboral:
especialízate con los posgrados
de EOI y marca la diferencia.



EOI Escuela de
organización
industrial



saber más

ESTUDIA MEJOR con las Flashcards 2.0

Recuadro de color + poderes digitales

Fichas de estudio que te permitirán mejorar tu productividad



Organiza por color



Escanea tus Flashcards con



scribbee

y Repasa desde tu móvil

Sesiones de estudio eficientes con Scribbee.



Ideales para:

Pregunta/
Respuesta
Conceptos clave
Vocabulario
Resúmenes
Esquemas
Mapas mentales



Mira cómo funciona

--1

--Creacion de tablas y trigger

```
CREATE TABLE CONTROL_NOMINA (  
    Fecha DATE NOT NULL,  
    Usuario VARCHAR2(35),  
    Empleado Number NOT NULL,  
    Antes Number,  
    Despues Number);
```

```
CREATE TABLE NOMINA (  
    Empleado NUMBER PRIMARY KEY,  
    importe_bruto NUMBER,  
    otros_datos VARCHAR2(100) -- Agrega más columnas según sea necesario  
);
```

```
CREATE TABLE EMPLEADO (  
    ID NUMBER PRIMARY KEY,  
    USUARIO VARCHAR2(35),  
    DNI VARCHAR2(20),  
    NOMBRE VARCHAR2(50),  
    APELLIDO1 VARCHAR2(50),  
    APELLIDO2 VARCHAR2(50),  
    CATEGORIA NUMBER  
);
```

```
CREATE OR REPLACE TRIGGER TR_NOMINA  
AFTER INSERT OR UPDATE ON NOMINA  
FOR EACH ROW  
BEGIN  
    IF INSERTING THEN  
        INSERT INTO CONTROL_NOMINA (  
            Fecha, Usuario, Empleado, Antes, Despues  
        ) VALUES (  
            SYSDATE, USER, :NEW.Empleado, NULL, :NEW.importe_bruto  
        );  
    ELSIF UPDATING THEN  
        INSERT INTO CONTROL_NOMINA (  
            Fecha, Usuario, Empleado, Antes, Despues  
        ) VALUES (  
            SYSDATE, USER, :NEW.Empleado, :OLD.importe_bruto, :NEW.importe_bruto  
        );  
    END IF;  
  
    COMMIT;  
END;
```

WUOLAH

```

--2
--Crear paquete con procedimientos
CREATE OR REPLACE
PACKAGE PK_NOMINAS AS

    PROCEDURE P_CREA_NOMINA (fecha_actual DATE, mes VARCHAR2);
    PROCEDURE P_BORRA_NOMINA (mes VARCHAR2);
    PROCEDURE P_VISTAS_EMPLEADOS (categoria NUMBER);

END PK_NOMINAS;

CREATE OR REPLACE
PACKAGE BODY PK_NOMINAS AS

    PROCEDURE P_CREA_NOMINA (fecha_actual DATE, mes VARCHAR2) AS
    BEGIN
        INSERT INTO NOMINA(FECHA_EMISION, IMPORTE_NETO, EMPLEADO,
        IMPORTE_BRUTO)
        (SELECT fecha_actual, IMPORTE_NETO, EMPLEADO, IMPORTE_BRUTO FROM
        NOMINA
        WHERE TO_CHAR(FECHA_EMISION, 'mm/yyyy') = mes);

        COMMIT;
    END;

    PROCEDURE P_BORRA_NOMINA (mes VARCHAR2) AS
    BEGIN
        DELETE FROM NOMINA WHERE TO_CHAR(FECHA_EMISION, 'mm/yyyy') LIKE
        mes;

        COMMIT;
    END;

    PROCEDURE P_VISTAS_EMPLEADOS (categoria NUMBER) AS
        sentencia VARCHAR2(300);
        CURSOR c_empleados IS SELECT * FROM EMPLEADO WHERE CATEGORIA =
        categoria;

    BEGIN
        FOR emp IN c_empleados LOOP

            sentencia := 'CREATE OR REPLACE VIEW ' || emp.usuario || '_NOMINA AS
            SELECT DNI, NOMBRE, APELLIDO1,
            APELLIDO2, FECHA_EMISION, IMPORTE_NETO, IMPORTE_BRUTO FROM
            EMPLEADO E JOIN

```

```

NOMINA N ON E.ID = N.EMPLEADO WHERE E.ID = ' || emp.id;

BEGIN
    EXECUTE IMMEDIATE sentencia;
    COMMIT;

    EXCEPTION
        WHEN others THEN
            DBMS_OUTPUT.put_line('ERROR: Excepcion en la creacion de la vista');
END;

END LOOP;
END;

END PK_NOMINAS;

BEGIN
    PK_NOMINAS.P_CREA_NOMINA('16/06/2025','06/2024');
END;

BEGIN
    PK_NOMINAS.P_BORRA_NOMINA('05/2024');
END;

BEGIN
    PK_NOMINAS.P_VISTAS_EMPLEADOS(4);
END;

```