```
USE  cursos;

/*1.Crear un procedimiento almacenado llamado proc1 que  muestre los
apellidos, nombres y salarios de los trabajadores  nacidos antes del año
1970.*/

--Inserción de datos necesarios para probar el procedimiento almacenado
INSERT INTO trabajador (CodTrab, NomTrab, APeTrab, FechNac, Salario)
VALUES
(1, 'Carlos', 'Fernández', '1970-03-15', 3000.00), -- 55 años
(2, 'María', 'Gómez', '1969-07-22', 3200.00), -- 55 años
(3, 'José', 'López', '1969-11-05', 3100.00), -- 55 años
(4, 'Ana', 'Martínez', '1985-06-12', 2800.00),
(5, 'Luis', 'Sánchez', '1990-09-25', 2900.00),
(6, 'Elena', 'Díaz', '1982-01-18', 2700.00),
(7, 'Pedro', 'Rodríguez', '1995-04-30', 2600.00),
(8, 'Lucía', 'Hernández', '1998-12-09', 2500.00),
(9, 'Javier', 'Pérez', '1987-08-14', 2750.00),
(10, 'Sara', 'Torres', '1993-05-07', 2650.00);

CREATE PROCEDURE proc1 AS
BEGIN
     SELECT ApeTrab, NomTrab, Salario
     FROM trabajador
     WHERE YEAR(FechNac)<1970
END;

--EJECUCIÓN
EXEC proc1;

/*2. Crear un procedimiento almacenado llamado  proc2, este debe llamar a
PROC1  y dar un mensaje de finalización.*/
CREATE PROCEDURE proc2
AS
BEGIN
     EXEC proc1
     PRINT 'PROCESO FINALIZADO'
END;

--EJECUCIÓN
EXEC proc2;
```

```
/*3. Crear un procedimiento almacenado llamado proc3 que muestre para cada
curso en el que se haya inscrito algún trabajador, el nombre del curso,
nombre y apellidos del trabajador y si es apto o no. Si aún no tiene valor
en el campo apto, bien porque no se le haya evaluado o porque el curso
aún no se ha realizado, debe salir "-" (guión medio)*/

--Inserción de datos necesarios para probar el procedimiento almacenado
INSERT INTO curso (CodCurso, NomCurso, Horas, Fecha)
VALUES
(1, 'Seguridad en Redes', 40, '2024-03-10'),
(2, 'Administración de Bases de Datos', 60, '2024-05-15'),
(3, 'Desarrollo Web con JavaScript', 50, '2025-08-20'),
(4, 'Machine Learning Básico', 45, '2026-01-10');

INSERT INTO cursado (CodCursado, CodCurso, CodTrab, Apto)
VALUES
-- Curso 1 (Seguridad en Redes) - Algunos aptos, otros no
(1, 1, 1, 'S'),
(2, 1, 2, 'N'),
(3, 1, 3, 'S'),
(4, 1, 4, 'N'),

-- Curso 2 (Administración de Bases de Datos) - Algunos aptos, otros no
(5, 2, 5, 'S'),
(6, 2, 6, 'N'),
(7, 2, 7, 'S'),

-- Curso 3 (Desarrollo Web con JavaScript) - Sin valor en Apto
(8, 3, 8, NULL),
(9, 3, 9, NULL);

CREATE OR ALTER PROCEDURE proc3
AS
BEGIN
     SELECT NomTrab, ApeTrab, NomCurso , ISNULL(Apto,'-')
     FROM (trabajador T JOIN cursado CU ON (T.CodTrab=CU.CodTrab) )
          JOIN  CURSO C ON (CU.CodCurso=C.CodCurso)
END;

--EJECUCIÓN
EXEC proc3;
```

```
/*4. Crear un procedimiento almacenado llamado proc4 que reciba el nombre
de un curso como parámetro e indique en un mensaje de texto cuantas
personas lo han realizado o han comenzado a hacerlo, por lo
tanto la fecha de comienzo debe ser anterior o igual al día de hoy. (El
mensaje tendrá la siguiente forma: "El curso NOMBRE_CURSO ha sido
realizado por X trabajadores").
*/
CREATE PROCEDURE proc4 @NOMBRECURSO VARCHAR(50)
AS
BEGIN

      DECLARE @CUANTOS INT

      SELECT @CUANTOS=COUNT(*)
      FROM curso C JOIN cursado CU ON (C.CodCurso=CU.CodCurso)
      WHERE  C.NomCurso=@NOMBRECURSO AND Fecha <= CONVERT(DATE, GETDATE());

      PRINT 'EL CURSO ' + @NOMBRECURSO + ' ha sido realizado por '
       + CONVERT(VARCHAR(10),@CUANTOS)+ ' trabajadores'
END;

--EJECUCIÓN
EXEC proc4 'Administración de Bases de Datos';
EXEC proc4 'Desarrollo Web con JavaScript';
EXEC proc4'Machine Learning Básico';

/*5. Crear un procedimiento almacenado llamado proc5 que reciba como
parámetro de entrada el código de un trabajador y deposite en un parámetro
de salida cuántos cursos ha realizado que hayan sido aptos.*/
CREATE PROCEDURE proc5 @CODIGO INT, @CUANTOS INT OUTPUT
AS
BEGIN

      SELECT @CUANTOS=COUNT(*)
      FROM  cursado
      WHERE CodTrab=@CODIGO AND Apto='S'

END;

--EJECUCIÓN
DECLARE @NUM INT;
EXEC proc5 3,@NUM OUTPUT;
PRINT @NUM;
```

```
/*6. Crear un procedimiento almacenado llamado proc6 que devuelva,
utilizando la instrucción RETURN, el  valor 1 si existe algún curso que no
tenga fecha, en caso contrario devolver 2.
*/

--Inserción de datos necesarios para probar el procedimiento almacenado
INSERT INTO curso (CodCurso, NomCurso, Horas, Fecha)
VALUES (5, 'Inteligencia Artificial', 50, NULL);

CREATE PROCEDURE proc6
AS
BEGIN
      IF (SELECT COUNT(*) FROM curso WHERE Fecha IS NULL) > 0
      BEGIN
            RETURN 1
      END
      ELSE
      BEGIN
            RETURN 2
      END
END;

--EJECUCIÓN
DECLARE @VALOR INT;
EXEC @VALOR=proc6;
PRINT @VALOR;
```

```
/*7.Hacer un procedimiento almacenado que indique  mediante un texto, si
existe un determinado curso cuyo nombre se le pasa como parámetro. Hacer
el ejercicio de tres maneras:*/

 /*A. Con la instrucción PRINT dentro del procedimiento
  almacenado.(proc7A)
*/
CREATE PROCEDURE proc7A @NOMCURSO VARCHAR(50)
AS
BEGIN
IF(SELECT  COUNT(*) FROM curso WHERE NomCurso=@NOMCURSO)>0
      BEGIN
            PRINT 'EL CURSO EXISTE'
      END
ELSE
      BEGIN
            PRINT 'EL CURSO NO EXISTE'
      END
END;

--EJECUCIÓN
proc7A 'Linux avanzado'

/*B.Con la instrucción RETURN dentro del procedimiento almacenado, de modo
que si se devuelve 1 indicará que el curso existe y si devuelve 2 indicará
que no. Cuando ejecutes el procedimiento almacenado, utiliza la nstrucción
PRINT para mostrar el mensaje correspondiente dependiendo del valor
devuelto. (proc7B)
 */
CREATE PROCEDURE proc7B @NOMCURSO VARCHAR(50)
AS
BEGIN
      IF (SELECT  COUNT(*) FROM curso WHERE NomCurso=@NOMCURSO)>0
      BEGIN
            RETURN 1
      END
      ELSE
      BEGIN
            RETURN 2
      END
END;
```

```
--EJECUCIÓN
DECLARE @VALOR INT;
EXEC @VALOR=proc7B 'Linux avanzado';
IF (@VALOR=1) PRINT 'EL CURSO EXISTE';
ELSE PRINT 'EL CURSO NO EXISTE';

DECLARE @VALOR INT;
EXEC @VALOR=proc7B 'Inteligencia Artificial';
IF (@VALOR=1) PRINT 'EL CURSO EXISTE';
ELSE PRINT 'EL CURSO NO EXISTE';

/*C. Utilizando un parámetro de salida donde depositamos  el mensaje y en
la ejecución mostramos el parámetro de  salida. (proc7C)
*/
CREATE PROCEDURE proc7C @NOMCURSO VARCHAR(50), @EXISTE VARCHAR(30) OUTPUT
AS
BEGIN

      IF(SELECT  COUNT(*) FROM curso WHERE NomCurso=@NOMCURSO)>0
      BEGIN
            SET @EXISTE='EL CURSO EXISTE'
      END
      ELSE
      BEGIN
            SET @EXISTE='EL CURSO NO EXISTE'
      END
END;

--EJECUCIÓN
DECLARE @VALOR VARCHAR(30);
EXEC proc7C 'LINUX AVANZADO',@VALOR OUTPUT;
PRINT @VALOR;

DECLARE @VALOR VARCHAR(30);
EXEC proc7C 'Inteligencia Artificial',@VALOR OUTPUT;
PRINT @VALOR;
```

```
/*D.Ejecuta de nuevo el procedimiento almacenado  realizado según la
primera forma y muestra el valor de estado.*/
DECLARE @ESTADO INT;
EXEC @ESTADO=proc7A 'LINUX AVANZADO';
PRINT @ESTADO;

/*11.A Utilizando el LDD, crea una tabla con los campos: DNI, Nombre y
Apellidos. No le declares primary key. El DNI no se debe poder repetir y
debe admitir 8 dígitos seguidos de una letra. Utiliza para ello la función
REPLICATE.Introduce algunas filas, por último, añade un campo de tipo
entero.
*/

CREATE TABLE nuevo
(DNINuevo CHAR(9) NOT NULL UNIQUE CHECK (DNINuevo LIKE
REPLICATE('[0-9]',8) + '[A-Z]'),
 NomNuevo VARCHAR(50),
 ApeNuevo VARCHAR(90)
);

INSERT INTO nuevo
VALUES('12345678A','Nuevo1','Apellidos Nuevo1'),
      ('12345678B','Nuevo2','Apellidos Nuevo2'),
      ('12345678C','Nuevo3','Apellidos Nuevo3');

ALTER TABLE nuevo ADD CodNuevo INT NULL;
```

```
/*11.B Hacer un procedimiento almacenado en el que introduzcas valores no
repetidos en la columna añadida (no conoces cuántas filas hay
almacenadas). Consulta la tabla antes y después de ejecutar el
procedimiento almacenado.
*/
CREATE PROCEDURE proc11
AS
BEGIN

      DECLARE @NUM INT = 0

      UPDATE nuevo
      SET @NUM = CodNuevo = @NUM  + 1

END;

SELECT * FROM nuevo;

EXEC proc11;

/*11.C Modificar el campo anterior que se convierta en clave primaria.*/

/*Modificamos la columna para que no admita nulos, sino no nos
permite ejecutar la siguiente instrucción*/
ALTER TABLE nuevo ALTER COLUMN CodNuevo INT NOT NULL;

ALTER TABLE nuevo ADD CONSTRAINT res_primary PRIMARY KEY (CodNuevo);

SELECT * FROM nuevo;
```