

PostgreSQL

a) Características de la BD, ventajas e inconvenientes.

Características

- Modelo relacional

Uso del modelo relacional de bases de datos, organizando los datos en tablas relacionales.

- Amplia variedad de tipos nativos

Ofrece una amplia gama de tipos de datos nativos que permiten a los desarrolladores modelar y almacenar información de manera precisa, además de tipos de datos especializados para manipular fechas o direcciones IP entre otros.

- Flexibilidad y extensibilidad

Permite la creación de funciones personalizadas en varios lenguajes de programación, lo que brinda a los desarrolladores la posibilidad de ampliar su funcionalidad según sus necesidades específicas (Perl, Java, Python, C++).

- Soporte para JSON y XML

Incluye funciones nativas para trabajar con datos JSON y XML.

- Alta concurrencia

Capacidad para manejar alta concurrencia gracias a su método de control de concurrencia multiversión (MVCC), con la cual puede ofrecer transacciones consistentes en un rendimiento óptimo.

Ventajas

- Escalabilidad

Capacidad para manejar grandes volúmenes de datos y cargas de trabajo intensivas.

- Licencia de código abierto

Permite su uso, modificación y distribución de forma gratuita.

- Soporte de estándares

PostgreSQL se adhiere a los estándares de SQL y ofrece un amplio conjunto de características que cumplen con las especificaciones del lenguaje. Esto

garantiza la portabilidad de las aplicaciones y facilita la migración desde otros sistemas de gestión de bases de datos.

- Comunidad activa y soporte

La comunidad de PostgreSQL está llena de desarrolladores y usuarios activos que brindan soporte, comparten conocimientos y contribuyen al desarrollo continuo.

- Seguridad avanzada

Proporciona funciones de seguridad avanzadas, incluyendo autenticación, encriptación y control de acceso a nivel de columna, lo que garantiza la protección de los datos sensibles.

Inconvenientes

- Sintaxis compleja

La sintaxis de algunos comandos o sentencias no es del todo intuitiva, y si no se tienen los conocimientos necesarios puede resultar complejo.

- Lentitud en inserciones y actualizaciones en bases de datos pequeñas

Está diseñado para ambientes de alto volumen. Esto hace que la velocidad de respuesta pueda parecer lenta en comparación con bases de datos de pequeño tamaño.

b) Manual de instalación de esta.

1. Enlace de descarga: <https://www.postgresql.org/download/>
2. Seleccionar el sistema operativo.

Downloads

PostgreSQL Downloads

PostgreSQL is available for download as ready-to-use packages or installers for various platforms, as well as a source code archive if you want to build it yourself.

Packages and Installers

Select your operating system family:



3. Seleccionar "Download the installer".

Windows installers





Interactive installer by EDB

Download the installer certified by EDB for all supported PostgreSQL versions.

4. Seleccionar la versión (16.1).

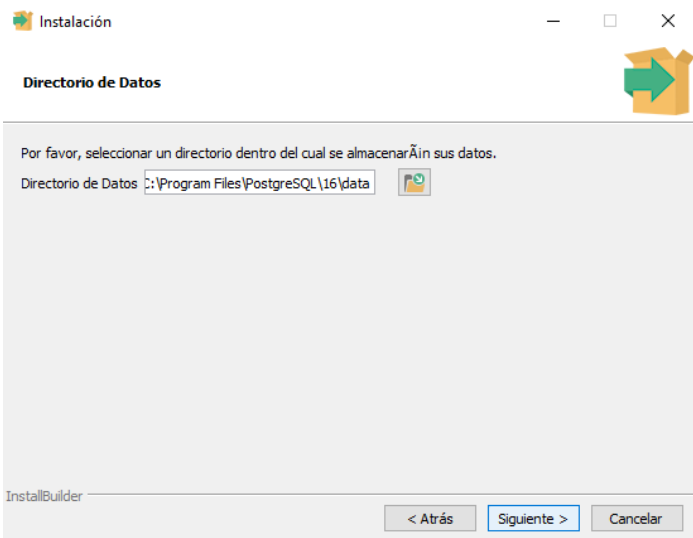
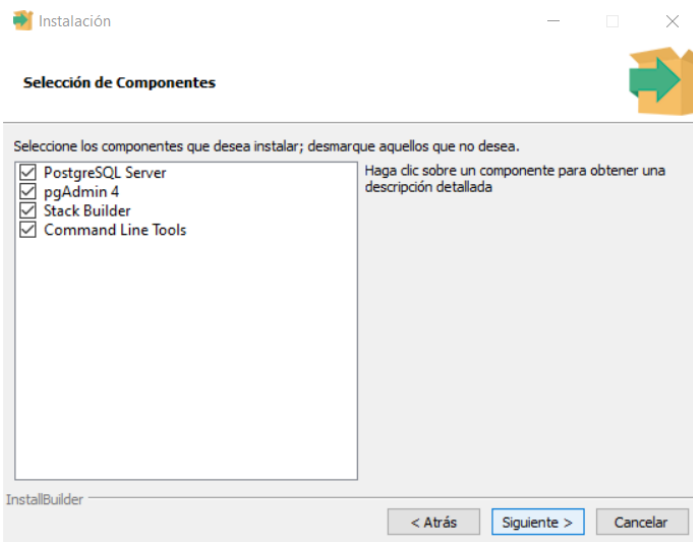
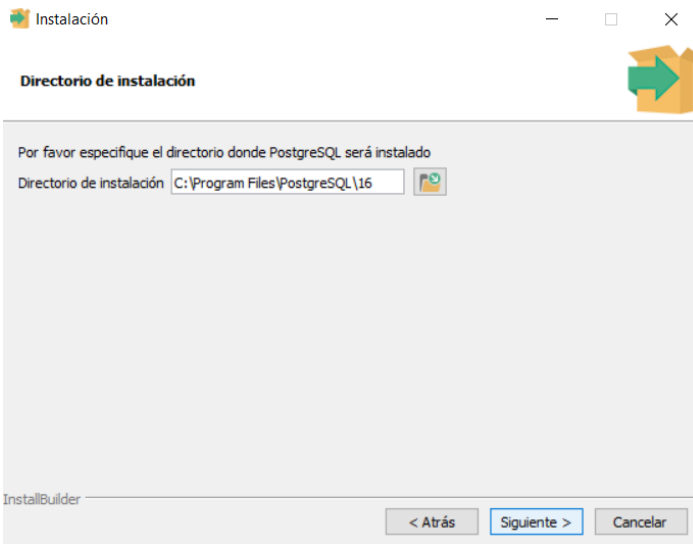
Download PostgreSQL

Open source PostgreSQL packages and installers from EDB

PostgreSQL Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
16.1	postgresql.org	postgresql.org			Not supported
15.5	postgresql.org	postgresql.org			Not supported

5. Se van siguiendo los pasos de instalación.





Instalación

Contraseña

Por favor, proporcione una contraseña para el superusuario base de datos postgres).

Contraseña

Reingresar la contraseña

InstallBuilder

< Atrás

Siguiente >

Cancelar

Instalación

Puerto

Por favor seleccione un número de puerto en el que el servidor deberá escuchar.

Puerto

InstallBuilder

< Atrás

Siguiente >

Cancelar

Instalación

Opciones Avanzadas

Seleccione la configuración regional a ser usada por el nuevo clúster de base de datos.

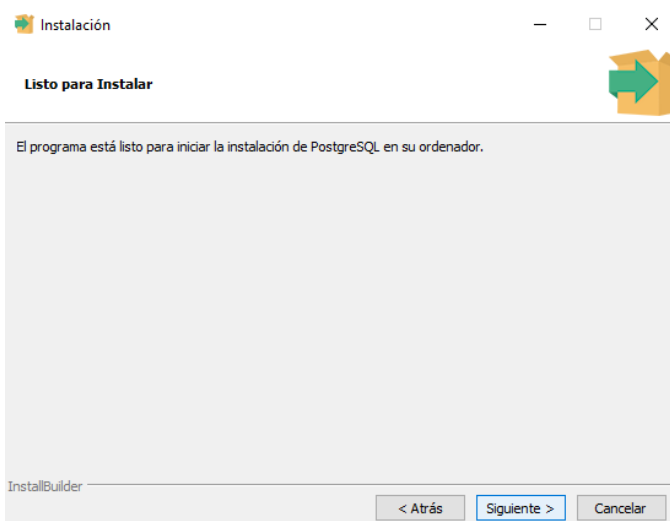
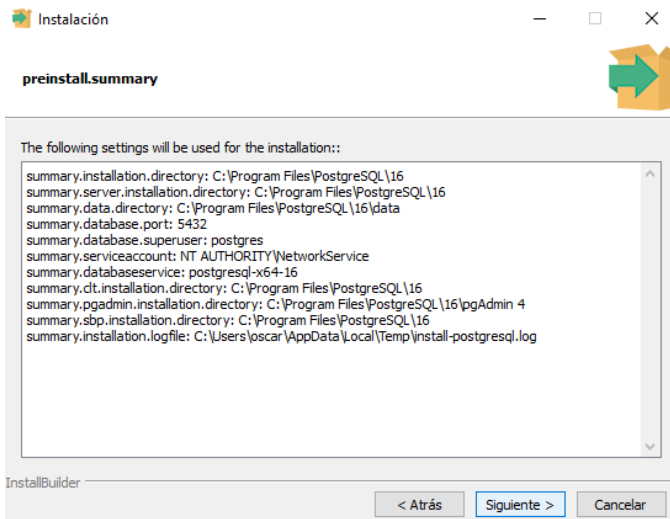
Configuración Regional

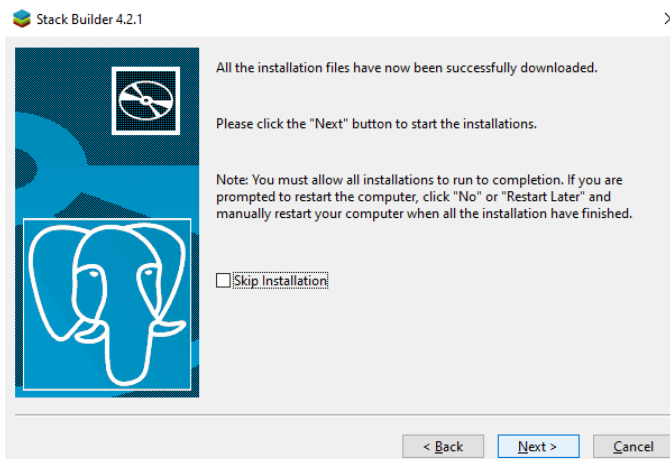
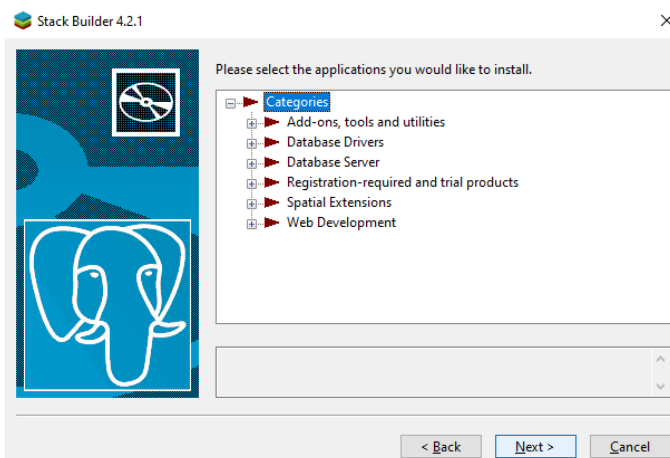
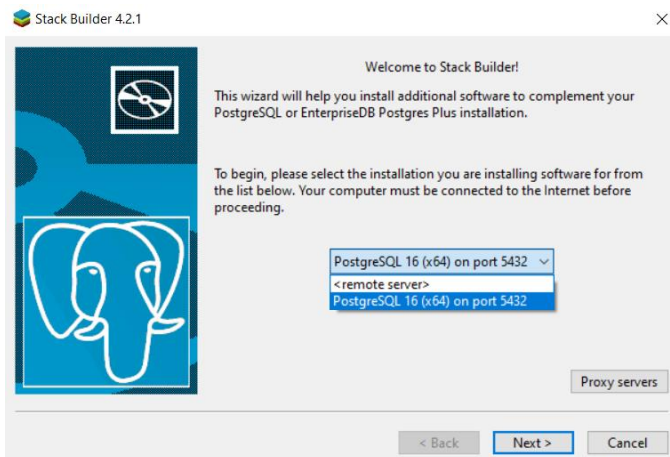
InstallBuilder

< Atrás

Siguiente >

Cancelar





c) Entornos de desarrollo existentes para la BBDD

- pgAdmin

Herramienta de gestión de bases de datos con interfaz gráfica y de código abierto que da soporte a PostgreSQL y otros sistemas de gestión de bases de datos.

- Navicat for PostgreSQL

Herramienta de administración de base de datos de pago, y por tanto, cuenta con muchas más funciones. Da soporte a PostgreSQL y otros sistemas de gestión de bases de datos. Además, es compatible con MongoDB y MySQL.

- DBeaver

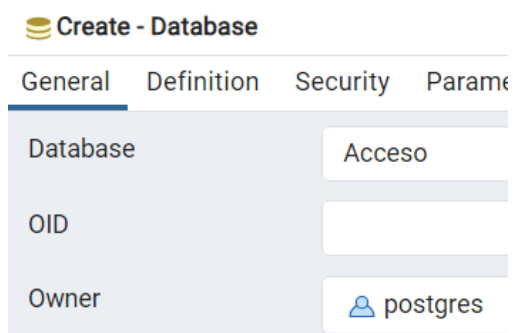
Herramienta de gestión de bases de datos de código abierto que da soporte a PostgreSQL y otros sistemas de gestión de bases de datos.

- DataGrip

Entorno de desarrollo integrado (IDE) multiplataforma de JetBrains que da soporte a PostgreSQL y otros sistemas de gestión de bases de datos.

d) Creación de la base de datos. Tipo de datos, altas y restricciones.

Creación Base de Datos



```
CREATE DATABASE "Acceso"
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LOCALE_PROVIDER = 'libc'
  CONNECTION LIMIT = -1
  IS_TEMPLATE = False;
```

Archivo para hacer Querys



Creación tabla TEMPLE


```
CREATE TABLE TEMPLE (  
  NUMEM INTEGER PRIMARY KEY NOT NULL,  
  NUMDE INTEGER NOT NULL,  
  EXTEL INTEGER,  
  FECNA DATE,  
  FECIN DATE,  
  SALAR NUMERIC(7,2),  
  COMIS NUMERIC(7,2),  
  NUMHI INTEGER,  
  NOMEM VARCHAR(90)  
);
```

Creación tabla TDEPTO

```
CREATE TABLE TDEPTO (  
  NUMDE INTEGER PRIMARY KEY NOT NULL,  
  NUMCE INTEGER NOT NULL,  
  DIREC INTEGER,  
  TIDIR CHAR CHECK (TIDIR IN ('P', 'F')),  
  PRESU NUMERIC(7,2),  
  DEPDE INTEGER,  
  NOMDE VARCHAR(90)  
);
```

Creación tabla TCENTR

```
CREATE TABLE TCENTR (  
  NUMCE INTEGER PRIMARY KEY NOT NULL,  
  NOMCE VARCHAR(90),  
  SEÑCE VARCHAR(90)  
);
```

Inserción de registros TEMPLE

```

INSERT INTO TEMPLE (NUMEM, NUMDE, EXTEL, FECNA, FECIN, SALAR, COMIS, NUMHI, NOMEM) VALUES
(110, 121, 350, '1929-11-10', '1950-02-15', 3100.00, NULL, 3, 'PONS, CESAR'),
(120, 112, 840, '1935-06-09', '1968-10-01', 3500.00, 1100.00, 1, 'LASA, MARIO'),
(130, 112, 810, '1945-09-09', '1969-02-01', 2900.00, 1100.00, 2, 'TEROL, LUCIANO'),
(150, 121, 340, '1930-08-10', '1948-01-15', 4400.00, NULL, 0, 'PEREZ, JULIO'),
(160, 111, 740, '1939-07-09', '1968-11-11', 3100.00, 1100.00, 2, 'AGUIRRE, AUREO'),
(180, 110, 508, '1934-10-18', '1956-03-18', 4800.00, 500.00, 2, 'PEREZ, MARCOS'),
(190, 121, 350, '1932-05-12', '1962-02-11', 3000.00, NULL, 4, 'VEIGA, JULIANA'),
(210, 100, 200, '1940-09-28', '1959-01-22', 3800.00, NULL, 2, 'GALVEZ, PILAR'),
(240, 111, 760, '1942-02-26', '1966-02-24', 2800.00, 1000.00, 3, 'SANZ, LAVINIA'),
(250, 100, 250, '1946-10-27', '1967-03-01', 4500.00, NULL, 0, 'ALBA, ADRIANA'),
(260, 100, 220, '1943-12-03', '1968-07-12', 7200.00, NULL, 6, 'LOPEZ, ANTONIO'),
(270, 112, 800, '1945-05-21', '1966-09-10', 3800.00, 800.00, 3, 'GARCIA, OCTAVIO'),
(280, 130, 410, '1948-01-11', '1971-10-08', 2900.00, NULL, 5, 'FLOR, DOROTEA'),
(285, 122, 620, '1949-10-25', '1968-02-15', 3800.00, NULL, 0, 'POLO, OTILIA'),
(290, 120, 910, '1947-11-30', '1968-02-14', 2700.00, NULL, 3, 'GIL, GLORIA'),
(310, 130, 480, '1946-11-21', '1971-01-15', 4200.00, NULL, 0, 'GARCIA, AUGUSTO'),
(320, 122, 620, '1957-12-25', '1978-02-05', 4050.00, NULL, 2, 'SANZ, CORNELIO'),
(330, 112, 850, '1948-08-19', '1972-03-01', 2800.00, 900.00, 0, 'DIEZ, AMELIA'),
(350, 122, 610, '1949-04-13', '1984-09-10', 4500.00, NULL, 1, 'CAMPS, AURELIO'),
(360, 111, 750, '1958-10-28', '1968-10-10', 2500.00, 1000.00, 2, 'LARA, DORINDA'),
(370, 121, 360, '1967-06-22', '1987-01-20', 1900.00, NULL, 1, 'RUIZ, FABIOLA'),
(380, 112, 880, '1968-03-30', '1988-01-01', 1800.00, NULL, 0, 'MARTIN, MICAELA'),
(390, 110, 500, '1966-02-19', '1986-10-08', 2150.00, NULL, 1, 'MORAN, CARMEN'),
(400, 111, 780, '1969-08-18', '1987-11-01', 1850.00, NULL, 0, 'LARA, LUCRECIA'),
(410, 122, 660, '1968-07-14', '1988-10-13', 1750.00, NULL, 0, 'MUÑOZ, AZUCENA'),
(420, 130, 450, '1966-10-22', '1988-11-19', 4000.00, NULL, 0, 'FIERRO, CLAUDIA'),
(430, 122, 650, '1967-10-26', '1988-11-19', 2100.00, NULL, 1, 'MORA, VALERIANA'),
(440, 111, 760, '1966-09-26', '1986-02-28', 2100.00, 1000.00, 0, 'DURAN, LIVIA'),
(450, 112, 880, '1966-10-21', '1986-02-28', 2100.00, 1000.00, 0, 'PEREZ, SABINA'),
(480, 111, 760, '1965-04-04', '1986-02-28', 2100.00, 1000.00, 1, 'PINO, DIANA'),
(490, 112, 880, '1964-06-06', '1988-01-01', 1800.00, 1000.00, 0, 'TORRES, HORACIO'),
(500, 111, 750, '1965-10-08', '1987-01-01', 2000.00, 1000.00, 0, 'VAZQUEZ, HONORIA'),
(510, 110, 550, '1966-05-04', '1986-11-01', 2000.00, NULL, 1, 'CAMPOS, ROMULO'),
(550, 111, 780, '1970-01-10', '1988-01-21', 1000.00, 1200.00, 0, 'SANTOS, SANCHO');

```

Inserción de registros TDEPTO

```

INSERT INTO TDEPTO (NUMDE, NUMCE, DIREC, TIDIR, PRESU, DEPDE, NOMDE) VALUES
(100, 10, 260, 'P', 120.00, NULL, 'DIRECCIÓN GENERAL'),
(110, 20, 180, 'P', 150.00, 100, 'DIRECCIÓN COMERCIAL'),
(111, 20, 180, 'F', 110.00, 110, 'SECTOR INDUSTRIAL'),
(112, 20, 270, 'P', 90.00, 110, 'SECTOR SERVICIOS'),
(120, 10, 150, 'F', 30.00, 100, 'ORGANIZACIÓN'),
(121, 10, 150, 'P', 20.00, 120, 'PERSONAL'),
(122, 10, 350, 'P', 60.00, 120, 'PROCESO DE DATOS'),
(130, 10, 310, 'P', 20.00, 100, 'FINANZAS');

```

Inserción de registros TCENTR

```

INSERT INTO TCENTR (NUMCE, NOMCE, SEÑCE) VALUES
(10, 'SEDE CENTRAL', 'C. ALCALA, 820, MADRID'),
(20, 'RELACIÓN CON CLIENTES', 'C. ATOCHA. 405, MADRID');

```

Restricciones tabla TEMPLE

```
ALTER TABLE TEMPLA
ADD CONSTRAINT fk_temple_numde FOREIGN KEY (NUMDE) REFERENCES TDEPTO(NUMDE);
```

Restricciones tabla TDEPTO

```
ALTER TABLE TDEPTO
ADD CONSTRAINT fk_tdepto_numce FOREIGN KEY (NUMCE) REFERENCES TCENTR(NUMCE);
```

```
ALTER TABLE TDEPTO
ADD CONSTRAINT fk_tdepto_direc FOREIGN KEY (DIREC) REFERENCES TEMPLA(NUMEM);
```

e) Operaciones directas a la base de datos.

CONSULTAS

1. Obtener los nombres y sueldos de los empleados con más de tres hijos por orden alfabético.

```
SELECT NOMEM, SALAR FROM TEMPLA WHERE NUMHI > 3 ORDER BY NOMEM;
```

2. Obtener los números de departamento y para los nombres de éstos con más de doce caracteres extraer los cinco siguientes, mostrándolos por orden de número de departamento.

```
SELECT NUMDE, SUBSTRING(NOMDE, 12, 5) as "NOMBRE" FROM TDEPTO
WHERE LENGTH(NOMDE) > 12 ORDER BY NUMDE;
```

3. Para los departamentos 111 y 112 hallar la media aritmética de los años de servicio de sus empleados en el día de hoy.

```
SELECT NUMDE, TRUNC(AVG(EXTRACT(YEAR FROM AGE(NOW(), FECIN))), 0)
"MEDIA DE AÑOS DE SERVICIO" FROM TEMPLA WHERE NUMDE IN (111, 112)
GROUP BY NUMDE;
```

SUBCONSULTAS

1. Obtener por orden alfabético los nombres de los empleados cuyos sueldos igualan o superan al de Claudia Fierro en más del 50%

```
SELECT NOMEM FROM TEMPLA WHERE SALAR >= (SELECT SALAR * 1.5 FROM
TEMPLA WHERE NOMEM = 'FIERRO, CLAUDIA') ORDER BY NOMEM;
```

2. Obtener los nombres y salarios de los empleados cuyo salario coincide con la comisión de algún otro o la suya propia. Ordenarlos alfabéticamente.

```
SELECT NOMEM, SALAR FROM TEMPLA WHERE SALAR = ANY (SELECT SALAR
FROM TEMPLA WHERE COMIS IS NOT NULL) ORDER BY NOMEM;
```

JOINS

1. Para los departamentos cuyo director lo sea en funciones, hallar el número de empleados y la suma de sus salarios, comisiones y número de hijos.

```
SELECT TDEPTO.NUMDE, COUNT(TEMPLE.NUMEM) AS "TOTAL EMPLEADOS",  
SUM(TEMPLE.SALAR) AS "TOTAL SALARIOS", SUM(TEMPLE.COMIS) AS "TOTAL  
COMISIONES", SUM(TEMPLE.NUMHI) AS "TOTAL HIJOS" FROM TDEPTO, TEMPLE  
WHERE TDEPTO.NUMDE = TEMPLE.NUMDE and TDEPTO.TIDIR = 'F' GROUP BY  
TDEPTO.NUMDE;
```

2. Para los departamentos cuyo presupuesto anual supera los 100.000 €, hallar cuántos empleados hay en promedio por cada extensión telefónica.

```
SELECT TEMPLE.EXTEL, COUNT(TEMPLE.NUMEM) AS "EMPLEADOS POR LÍNEA  
TELFÓNICA" FROM TDEPTO, TEMPLE WHERE TDEPTO.NUMDE =  
TEMPLE.NUMDE AND TDEPTO.PRESU > 100.00 GROUP BY TDEPTO.NUMDE,  
TEMPLE.EXTEL;
```

SENTENCIAS DML

1. Dar de alta en la tabla de TEMPLE a Constantino Garcés que no tenga comisión y con fecha de hoy.

```
INSERT INTO TEMPLE VALUES ((SELECT MAX(NUMEM)+1 FROM TEMPLE),  
(SELECT NUMDE FROM TDEPTO WHERE NOMDE = 'PERSONAL'), 123, '1990-01-  
01', CURRENT_DATE, 2500.00, NULL, 2, 'GARCÉS, CONSTANTINO');
```

2. Como consecuencia del convenio, se aumenta el sueldo a todos los empleados en un 5.33% y la comisión en un 6.19% a todos los vendedores.

```
UPDATE TEMPLE SET SALAR = SALAR * 1.0533, COMIS = COMIS * 1.0619 WHERE  
COMIS IS NOT NULL;
```

PROCEDIMIENTOS Y FUNCIONES

1. Crear un programa que muestre los empleados de la tabla TEMPLE que trabajan un departamento introducido por teclado.

```
CREATE OR REPLACE FUNCTION mostrarEmpleadosDepartamento(P_NUMDE  
INTEGER) RETURNS TABLE (NUMEM INTEGER, NOMEM VARCHAR(90))  
LANGUAGE plpgsql AS
```

```
$$
```

```
BEGIN
```

```
    RAISE NOTICE 'Empleados del departamento %', P_NUMDE;
```

```
    RETURN QUERY SELECT t.NUMEM, t.NOMEM FROM TEMPLE t WHERE  
    t.NUMDE = P_NUMDE;
```

```
END;
```

```
$$;
```

```
/*****
```

```
SELECT * FROM mostrarEmpleadosDepartamento (121);
```

2. Realizar un programa que muestre los códigos de los empleados y cuente aquellos cuya comisión es nula.

```
CREATE OR REPLACE FUNCTION codigoEmpleado()
```

```
RETURNS INT LANGUAGE plpgsql AS
```

```
$$
```

```
DECLARE
```

```
    contador INT := 0;
```

```
    R RECORD;
```

```
BEGIN
```

```
    FOR R IN SELECT NUMEM FROM TEMPLE WHERE COMIS IS NULL
```

```
    LOOP
```

```
        RAISE NOTICE 'Empleado con comisión nula: %', R.NUMEM;
```

```
        contador := contador + 1;
```

```
    END LOOP;
```

```
    RETURN contador;
```

```
END;
```

```
$$
```

```
/*****
```

```
DO
```

```
$$
```

```
DECLARE
```

```
    result INT;
```

```
BEGIN
```

```
    result := codigoEmpleado ();
```

```
    RAISE NOTICE 'Número de empleados con comisión nula: %', result;
```

END

\$\$;

3. Escribir un bloque PL que reciba una cadena y visualice el apellido y el número de empleado de todos los empleados cuyo apellido contenga la cadena especificada. Al finalizar se visualizará el número de empleados mostrados.

```
CREATE OR REPLACE FUNCTION cadenaEmpleado(CADENA VARCHAR) RETURNS  
INT LANGUAGE plpgsql AS
```

\$\$

```
DECLARE
```

```
    contador INT := 0;
```

```
    R RECORD;
```

```
BEGIN
```

```
    FOR R IN SELECT NUMEM, NOMEM FROM TEMPLE WHERE NOMEM ILIKE  
    '%' || CADENA || '%'
```

```
    LOOP
```

```
        RAISE NOTICE 'NUMEM: %, Apellido: %', R.NUMEM, R.NOMEM;  
        contador := contador + 1;
```

```
    END LOOP;
```

```
    RETURN contador;
```

```
END;
```

\$\$

```
/*****
```

```
DO
```

\$\$

```
DECLARE
```

```
    result INT;
```

```
BEGIN
```

```
    result := cadenaEmpleado ('IA');
```

```
    RAISE NOTICE 'Total de empleados mostrados: %', result;
```

END

\$\$