1. Preguntas normales   
   Obtener el número total de noticias publicadas en la sección de Política:

SELECT COUNT(\*) AS TotalNoticias

FROM Noticias n

JOIN ClasificacionesNoticias cn ON n.idNoticia = cn.idNoticia

JOIN Secciones s ON cn.idSección = s.idSección

WHERE s.Nombre = 'Política';

1. Obtener el nombre de la sección y la cantidad de noticias publicadas en cada sección:

SELECT s.Nombre AS Sección, COUNT(\*) AS CantidadNoticias

FROM Secciones s

JOIN ClasificacionesNoticias cn ON s.idSección = cn.idSección

GROUP BY s.Nombre;

1. Obtener el título, la fecha de publicación y el número de visualizaciones de las 3 noticias más populares:

SELECT n.Título, n.FechaPublicación, v.NumeroVisualizaciones

FROM Noticias n

JOIN Visualizaciones v ON n.idNoticia = v.idNoticia

ORDER BY v.NumeroVisualizaciones DESC

LIMIT 3;

1. Obtener el título y la fecha de publicación de las noticias que contienen la palabra "crisis" en el título:

SELECT Título, FechaPublicación

FROM Noticias

WHERE Título LIKE '%crisis%';

Vistas:  
  
1. Esta vista muestra el cuerpo de la noticia y su respectiva categoría.

CREATE VIEW VistaNoticiasCategorias AS

SELECT N.Cuerpo AS Noticia, C.Nombre AS Categoría

FROM Noticias AS N

JOIN ClasificacionesNoticias AS CN ON N.idNoticia = CN.idNoticia

JOIN Categorías AS C ON CN.idSección = C.idCategoría;

1. Esta vista muestra el nombre del usuario y los detalles de su suscripción.

CREATE VIEW VistaSuscripcionesUsuario AS

SELECT U.Nombre, TS.Nombre AS TipoSuscripción, S.PrecioPagado, S.FechaInicio, S.FechaFin

FROM Usuarios AS U

JOIN Suscripciones AS S ON U.idUsuario = S.idUsuario

JOIN TiposSuscripciones AS TS ON S.idTipoSuscripción = TS.idTipoSuscripción;

1. Esta vista muestra el cuerpo de la noticia y los comentarios asociados a ella.

CREATE VIEW VistaComentariosNoticias AS

SELECT N.cuerpo, C.Comentario

FROM Noticias AS N

JOIN Comentarios AS C ON N.idNoticia = C.idNoticia;

1. Procedimientos almacenados   
     
   ¿Cómo podemos insertar un nuevo usuario y su suscripción?

CREATE OR REPLACE PROCEDURE InsertarUsuarioYSuscripcion(

idUsuario INT,

nombreUsuario VARCHAR(200),

correoElectronico VARCHAR(200),

idSuscripcion INT,

nombreSuscripcion VARCHAR(200),

fechaInicio DATE,

fechaFin DATE

)

AS

$BODY$

DECLARE

usuario Usuarios%rowtype;

BEGIN

RAISE NOTICE 'Insertando suscripción: %', nombreSuscripcion;

INSERT INTO Suscripciones (idSuscripcion, nombreSuscripcion, fechaInicio, fechaFin)

VALUES (idSuscripcion, nombreSuscripcion, fechaInicio, fechaFin);

RAISE NOTICE 'Suscripción insertada: %', nombreSuscripcion;

SELECT \* INTO usuario

FROM Usuarios

WHERE Usuarios.idUsuario = idUsuario;

IF usuario.idUsuario IS NULL THEN

RAISE NOTICE 'COMMIT';

COMMIT;

ELSE

RAISE NOTICE 'ROLLBACK';

ROLLBACK;

END IF;

RAISE NOTICE 'Insertando usuario: %', nombreUsuario;

INSERT INTO Usuarios (idUsuario, nombre, correoElectronico, idSuscripcion)

VALUES (idUsuario, nombreUsuario, correoElectronico, idSuscripcion);

RAISE NOTICE 'Usuario insertado: %', nombreUsuario;

-- Mostrar los datos insertados

SELECT \*

FROM Usuarios

WHERE Usuarios.idUsuario = idUsuario;

END;

$BODY$

LANGUAGE plpgsql;

1. ¿Cómo podemos insertar una nueva publicación y su autor?

CREATE OR REPLACE PROCEDURE InsertarPublicacionYAutor(

idPublicacion INT,

titulo VARCHAR(200),

contenido TEXT,

idAutor INT,

nombreAutor VARCHAR(200),

apellido VARCHAR(200)

)

AS

$BODY$

DECLARE

autor Autores%rowtype;

BEGIN

RAISE NOTICE 'Insertando autor: %', nombreAutor;

INSERT INTO Autores (idAutor, nombre, apellido)

VALUES (idAutor, nombreAutor, apellido);

RAISE NOTICE 'Autor insertado: %', nombreAutor;

SELECT \* INTO autor

FROM Autores

WHERE Autores.idAutor = idAutor;

IF autor.idAutor IS NULL THEN

RAISE NOTICE 'COMMIT';

COMMIT;

ELSE

RAISE NOTICE 'ROLLBACK';

ROLLBACK;

END IF;

RAISE NOTICE 'Insertando publicación: %', titulo;

INSERT INTO Publicaciones (idPublicacion, titulo, contenido, idAutor)

VALUES (idPublicacion, titulo, contenido, idAutor);

RAISE NOTICE 'Publicación insertada: %', titulo;

-- Mostrar los datos insertados

SELECT \*

FROM Publicaciones

WHERE Publicaciones.idPublicacion = idPublicacion;

END;

$BODY$

LANGUAGE plpgsql;

3) "¿Cómo podemos insertar una nueva noticia junto con su categoría correspondiente?"

CREATE OR REPLACE PROCEDURE InsertarNoticiaYCategoría(

idNoticia INT,

título VARCHAR(200),

contenido TEXT,

fechaPublicación DATE,

idCategoría INT,

nombreCategoría VARCHAR(200)

)

AS

$BODY$

DECLARE

noticia Noticias%rowtype;

BEGIN

RAISE NOTICE 'Insertando categoría: %', nombreCategoría;

INSERT INTO Categorias (idCategoría, nombre)

VALUES (idCategoría, nombreCategoría);

RAISE NOTICE 'Categoría insertada: %', nombreCategoría;

SELECT \* INTO noticia

FROM Noticias

WHERE Noticias.idNoticia = idNoticia;

IF noticia.idNoticia IS NULL THEN

RAISE NOTICE 'COMMIT';

COMMIT;

ELSE

RAISE NOTICE 'ROLLBACK';

ROLLBACK;

END IF;

RAISE NOTICE 'Insertando noticia: %', título;

INSERT INTO Noticias (idNoticia, título, contenido, fechaPublicación, idCategoría)

VALUES (idNoticia, título, contenido, fechaPublicación, idCategoría);

RAISE NOTICE 'Noticia insertada: %', título;

-- Mostrar los datos insertados

SELECT \*

FROM Noticias

WHERE Noticias.idNoticia = idNoticia;

END;

$BODY$

LANGUAGE plpgsql;