

Examen Final

EJERCICIO DE REPASO

Contexto de la Empresa

TechSolutions es una empresa de tecnología que tiene varios departamentos, cada uno de los cuales trabaja en múltiples proyectos. La base de datos contiene las siguientes tablas:

- **empleados:** Información sobre los empleados.
- **departamentos:** Información sobre los departamentos.
- **proyectos:** Información sobre los proyectos.
- **empleados_proyectos:** Relación entre empleados y proyectos.

Estructura de las Tablas

empleados:

- **id:** Identificador del empleado (PRIMARY KEY)
- **nombre:** Nombre del empleado
- **apellido:** Apellido del empleado
- **departamento_id:** Identificador del departamento (FOREIGN KEY)
- **salario:** Salario del empleado

departamentos:

- **id:** Identificador del departamento (PRIMARY KEY)
- **nombre:** Nombre del departamento

proyectos:

- **id:** Identificador del proyecto (PRIMARY KEY)
- **nombre:** Nombre del proyecto
- **departamento_id:** Identificador del departamento (FOREIGN KEY)
- **presupuesto:** Presupuesto del proyecto

empleados_proyectos:

- **empleado_id:** Identificador del empleado (FOREIGN KEY)
- **proyecto_id:** Identificador del proyecto (FOREIGN KEY)

Ejercicio

1. Creación de la Base de Datos y Tablas:

a. Crea una base de datos llamada techsolutions.

```
create database techsolutions;
```

b. Crea las tablas mencionadas en la base de datos techsolutions.

```
create table empleados(
    id serial primary key,
    nombre varchar (50),
    apellido varchar (100),
    departamento_id int,
    salario int,
    constraint fk_departamento foreign key (departamento_id) references departamentos(id)
);

create table departamentos(
    id serial primary key,
    nombre varchar (100)
);

create table proyectos(
    id serial primary key,
    nombre varchar (200),
    departamento_id int,
    presupuesto int,
    constraint fk_departamento foreign key (departamento_id) references departamentos(id) --todo lo que saen columnas de una
);
```

```
create table empleados_proyectos(
    empleado_id int,
    proyecto_id int,
    primary key (empleado_id, proyecto_id), --indicar la clave primaria conjunta entre parentesis--
    constraint fk_empleado foreign key (empleado_id) references empleados(id),
    constraint fk_proyecto foreign key (proyecto_id) references proyectos(id)
);
```

2. Creación de Roles:

a. Crea un rol llamado analista.

```
create role analista
```

b. Crea un rol llamado gerente.

```
create role gerente
```

c. Crea un rol llamado admin que tenga todos los permisos para iniciar sesión, crear bases de datos, crear roles y añadir usuarios a roles. El máximo de conexiones paralelas será de 2000.

```
create role admin
    login
    createdb
    createrole
    conection limit 2000;
```

3. Asignación de Permisos:

a. Concede permisos de lectura al rol analista sobre todas las tablas.

```
grant select on empleados, departamentos, proyectos, empleados_proyectos to analista;
```

b. Concede permisos de lectura y escritura al rol gerente sobre todas las tablas.

```
grant select, insert, update, delete on empleados, departamentos, proyecto, empleados_proyectos to gerente;
```

c. Concede todos los permisos al rol admin sobre la base de datos y las tablas.

```
grant all privileges on database techsolutions to admin;

grant all privileges on table empleados, departamentos, proyectos, empleados_proyectos to admin;
```

4. Revocación de Permisos:

a. Revoca los permisos de escritura al rol gerente sobre la columna salario en la tabla empleados.

```
revoke insert, update, delete on empleados from gerente;

grant insert, update, delete (id, nombre, apellido, departamento_id) on empleadoto gerente
```

5. Crea los siguientes usuarios y añádelos al rol correspondiente.

Oficina	Usuario	Contraseña	Validez Hasta	Permisos extra
analista	pmartinez	unicornio	2025	Ninguno
	jsuarez	123pescado	ilimitada	Ninguno
admin	acasanova	patito	Julio 2026	Ninguno
	rsegura	mazapan	ilimitada	Puede cambiar la estructura de las tablas
	smolina	cucudrilo	ilimitada	Ninguno
gerente	msanchez	#67sacarino	ilimitada	Puede ejecutar procedimientos
	bgonzalez	west33	ilimitada	Puede crear tablas y vistas

```
create user pmartinez
    password 'unicornio'
    valid until '31-12-2024'
    in role analista;

create user jsuarez
    password '123pescado'
    valid until 'infinty'
    in role analista;
```

```

create user acasanova
  password 'papito'
  valid until '31-06-2024'
  in role admin;

create user rsegura
  password 'mazapan'
  valid until 'infinity'
  in role admin;

  grant alter on techsolution to rsegura;

create user smolina
  password 'cucudrilo'
  valid until 'infinity'
  in role gerente;

create user msanchez
  password '#67sacarino'
  valid until 'infinity'
  in role gerente;

  grant execute on all functions in schema public to msanchez

create user bgonzalez
  password 'west33'
  valid until 'infinity'
  in role gerente;

  grant create on techsolution to bgonzalez

```

Comandos SQL del Documento TCL - DCL

Control de las Transacciones (TCL)

1. BEGIN

- Inicia una transacción.

- `BEGIN;`

2. SAVEPOINT

- Establece un punto de guardado dentro de la transacción.

- `SAVEPOINT nombre_punto;`

3. COMMIT

- Guarda todos los cambios realizados en la transacción.

- `COMMIT;`

4. ROLLBACK

- Revierte todos los cambios realizados desde el inicio de la transacción o desde un punto de guardado.

- `ROLLBACK;`

- `ROLLBACK TO SAVEPOINT nombre_punto;`

Lenguaje de Control de Datos (DCL)

1. CREATE USER

- Crea un nuevo usuario con una contraseña.

- `CREATE USER nombre_usuario WITH PASSWORD 'contraseña';`

2. CREATE ROLE

- Crea un nuevo rol.

- `CREATE ROLE nombre_rol;`

- Con atributos:

- `CREATE ROLE nombre_rol WITH LOGIN PASSWORD 'contraseña';`

- `CREATE ROLE nombre_rol WITH SUPERUSER;`

- `CREATE ROLE nombre_rol WITH CREATEDB;`

- `CREATE ROLE nombre_rol WITH CREATEROLE;`

- `CREATE ROLE nombre_rol WITH CONNECTION LIMIT 100;`

- `CREATE ROLE nombre_rol WITH VALID UNTIL 'fecha';`

3. GRANT

- Otorga permisos específicos a un usuario o rol.

- `GRANT permisos ON objeto TO usuario_o_rol;`
- Ejemplos de permisos:
 - `SELECT` : Permite ver datos en una tabla o vista.
 - `INSERT` : Permite agregar nuevos registros a una tabla.
 - `UPDATE` : Permite modificar los datos existentes en una tabla.
 - `DELETE` : Permite eliminar registros de una tabla.
 - `ALTER` : Permite modificar la estructura de una tabla.
 - `CREATE` : Permite crear nuevos objetos en la base de datos.
 - `DROP` : Permite eliminar objetos de la base de datos.
 - `EXECUTE` : Permite ejecutar procedimientos almacenados.

4. REVOKE

- Revoca permisos previamente otorgados a un usuario o rol.
- `REVOKE permisos ON objeto FROM usuario_o_rol;`
- Ejemplos de revocación de permisos:
 - `REVOKE SELECT, INSERT ON empleados FROM user1;`

5. ALTER DEFAULT PRIVILEGES

- Otorga permisos automáticamente sobre nuevas tablas y vistas en un esquema.
- `ALTER DEFAULT PRIVILEGES IN SCHEMA esquema GRANT permisos ON TABLES TO usuario_o_rol;`

Ejemplos de Uso

1. Conceder permisos de lectura e inserción a un usuario sobre una tabla

- `GRANT SELECT, INSERT ON empleados TO user1;`

2. Revocar permisos de lectura e inserción a un usuario sobre una tabla

- `REVOKE SELECT, INSERT ON empleados FROM user1;`

3. Crear un rol con permisos de creación de base de datos y conexión

- `CREATE ROLE dba CREATEDB LOGIN PASSWORD 'securePass1';`

4. Crear un rol con un límite de conexión

- `CREATE ROLE api LOGIN PASSWORD 'securePass1' CONNECTION LIMIT 1000;`

5. Conceder permisos de lectura en todas las tablas de un esquema a un rol

- `GRANT SELECT ON ALL TABLES IN SCHEMA public TO readonly;`

6. Revocar permisos de creación en el esquema `public` del rol `public`

- `REVOKE CREATE ON SCHEMA public FROM PUBLIC;`

7. Conceder permisos de conexión a una base de datos a un rol

- `GRANT CONNECT ON DATABASE mydatabase TO readonly;`

8. Conceder permisos de uso en un esquema a un rol

- `GRANT USAGE ON SCHEMA myschema TO readonly;`

9. Conceder permisos de lectura y escritura en todas las tablas de un esquema a un rol

- `GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA myschema TO readwrite;`

Comandos DDL (Data Definition Language)

1. CREATE TABLE

- Crea una nueva tabla en la base de datos.
- `CREATE TABLE nombre_tabla (columna1 tipo_datos, columna2 tipo_datos, ...);`

2. ALTER TABLE

- Modifica la estructura de una tabla existente.
- Agregar columna:
 - `ALTER TABLE nombre_tabla ADD columna tipo_datos;`
- Eliminar columna:
 - `ALTER TABLE nombre_tabla DROP COLUMN columna;`
- Modificar tipo de dato de una columna:
 - `ALTER TABLE nombre_tabla ALTER COLUMN columna TYPE nuevo_tipo_datos;`

3. DROP TABLE

- Elimina una tabla de la base de datos.
- `DROP TABLE nombre_tabla;`

4. CREATE INDEX

- Crea un índice en una tabla para mejorar el rendimiento de las consultas.

- `CREATE INDEX nombre_indice ON nombre_tabla (columna1, columna2, ...);`
5. **DROP INDEX**
- Elimina un índice de la base de datos.
 - `DROP INDEX nombre_indice;`
6. **CREATE VIEW**
- Crea una vista basada en una consulta SQL.
 - `CREATE VIEW nombre_vista AS SELECT columna1, columna2 FROM nombre_tabla WHERE condición;`
7. **DROP VIEW**
- Elimina una vista de la base de datos.
 - `DROP VIEW nombre_vista;`
8. **CREATE SCHEMA**
- Crea un nuevo esquema en la base de datos.
 - `CREATE SCHEMA nombre_esquema;`
9. **DROP SCHEMA**
- Elimina un esquema de la base de datos.
 - `DROP SCHEMA nombre_esquema;`
10. **CREATE SEQUENCE**
- Crea una secuencia generadora de números.
 - `CREATE SEQUENCE nombre_secuencia START WITH valor_inicio INCREMENT BY valor_incremento;`
11. **ALTER SEQUENCE**
- Modifica las propiedades de una secuencia.
 - `ALTER SEQUENCE nombre_secuencia RESTART WITH nuevo_valor;`
12. **DROP SEQUENCE**
- Elimina una secuencia de la base de datos.
 - `DROP SEQUENCE nombre_secuencia;`
13. **TRUNCATE TABLE**
- Elimina todos los registros de una tabla de manera rápida y eficiente.
 - `TRUNCATE TABLE nombre_tabla;`

Ejemplos de Uso en Transacciones

1. Crear una tabla dentro de una transacción

```
BEGIN;
CREATE TABLE empleados ( id SERIAL PRIMARY KEY, nombre VARCHAR(50), apellido VARCHAR(50), departamento_id INT, salario DECIMAL
);
COMMIT;
```

2. Modificar una tabla dentro de una transacción

```
BEGIN;
ALTER TABLE empleados ADD fecha_contratacion DATE;
COMMIT;
```

3. Eliminar una tabla dentro de una transacción

```
BEGIN;
DROP TABLE empleados;
COMMIT;
```

4. Crear una vista dentro de una transacción

```
BEGIN;
CREATE VIEW vista_empleados AS
SELECT nombre, apellido, salario FROM empleados WHERE salario > 50000;
COMMIT;
```

5. Eliminar una vista dentro de una transacción

```
BEGIN;
DROP VIEW vista_empleados;
COMMIT;
```

6. Crear un índice dentro de una transacción

```
BEGIN;
CREATE INDEX idx_empleados_salario ON empleados (salario);
COMMIT;
```

7. Eliminar un índice dentro de una transacción

```
BEGIN;
DROP INDEX idx_empleados_salario;
COMMIT;
```

Estos comandos DDL son fundamentales y se utilizan frecuentemente en la gestión de la estructura de la base de datos. Todos ellos pueden ser utilizados dentro de transacciones para asegurar que los cambios se realicen de manera atómica y coherente.