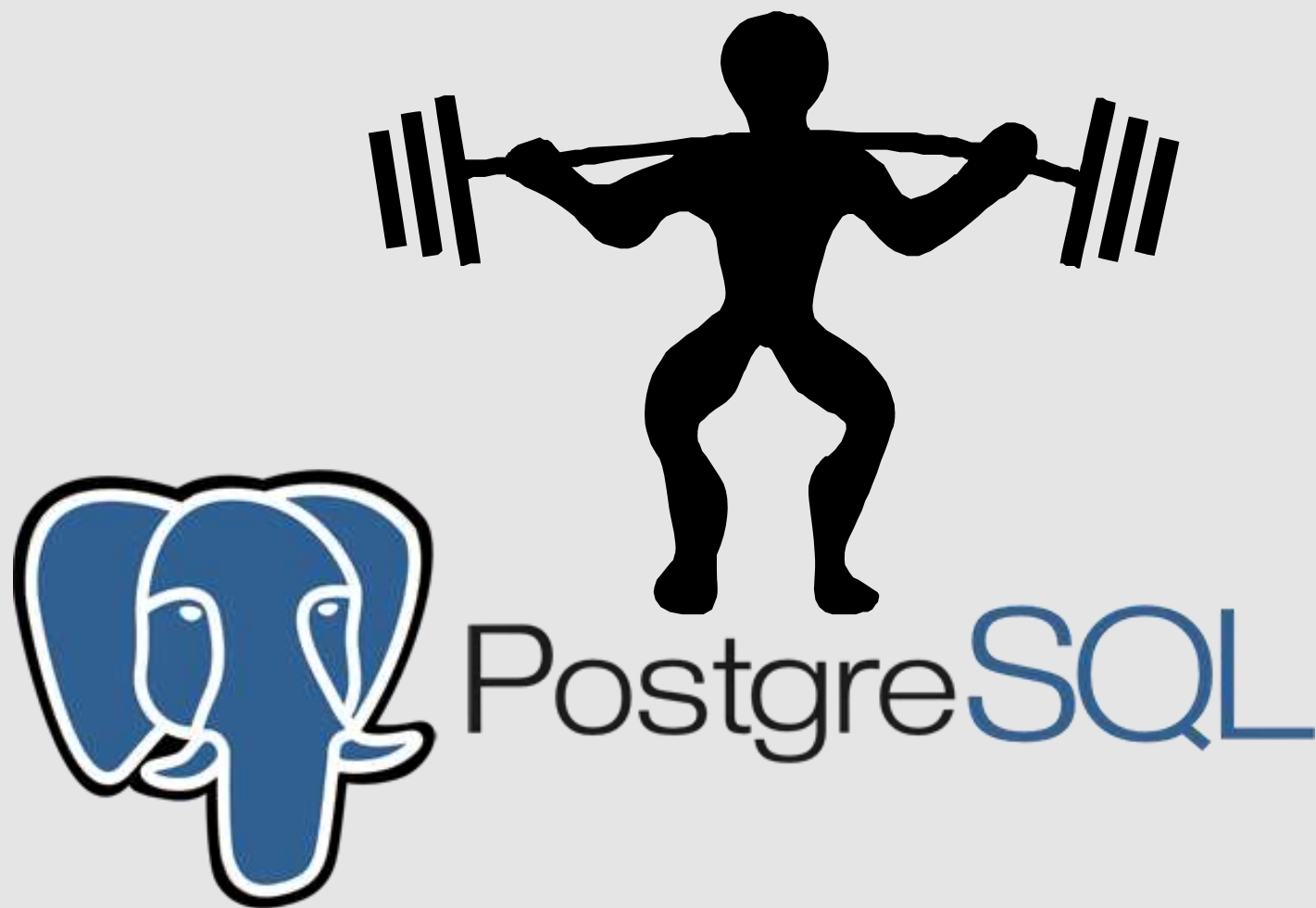


Sistema de Gestión de Gimnasio con PostgreSQL



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"



Hecho por:
OSCAR GUILLERMO SIERRA LOZANO.
YEFERSON ESMID HEREDIA PERDOMO.

DIAGRAMA DE ENTIDAD-RELACIÓN

El modelo relacional representa un sistema completo de gestión para un gimnasio. Incluye módulos clave como:

- Socios: Información personal y membresías activas.
- Clases grupales: Programación, instructores asignados y horarios disponibles.
- Reservas de clases: Registro de asistencia de socios a clases.
- Pagos de cuota: Historial de pagos con estados definidos.
- Control de acceso: Registro de entradas y salidas al gimnasio.

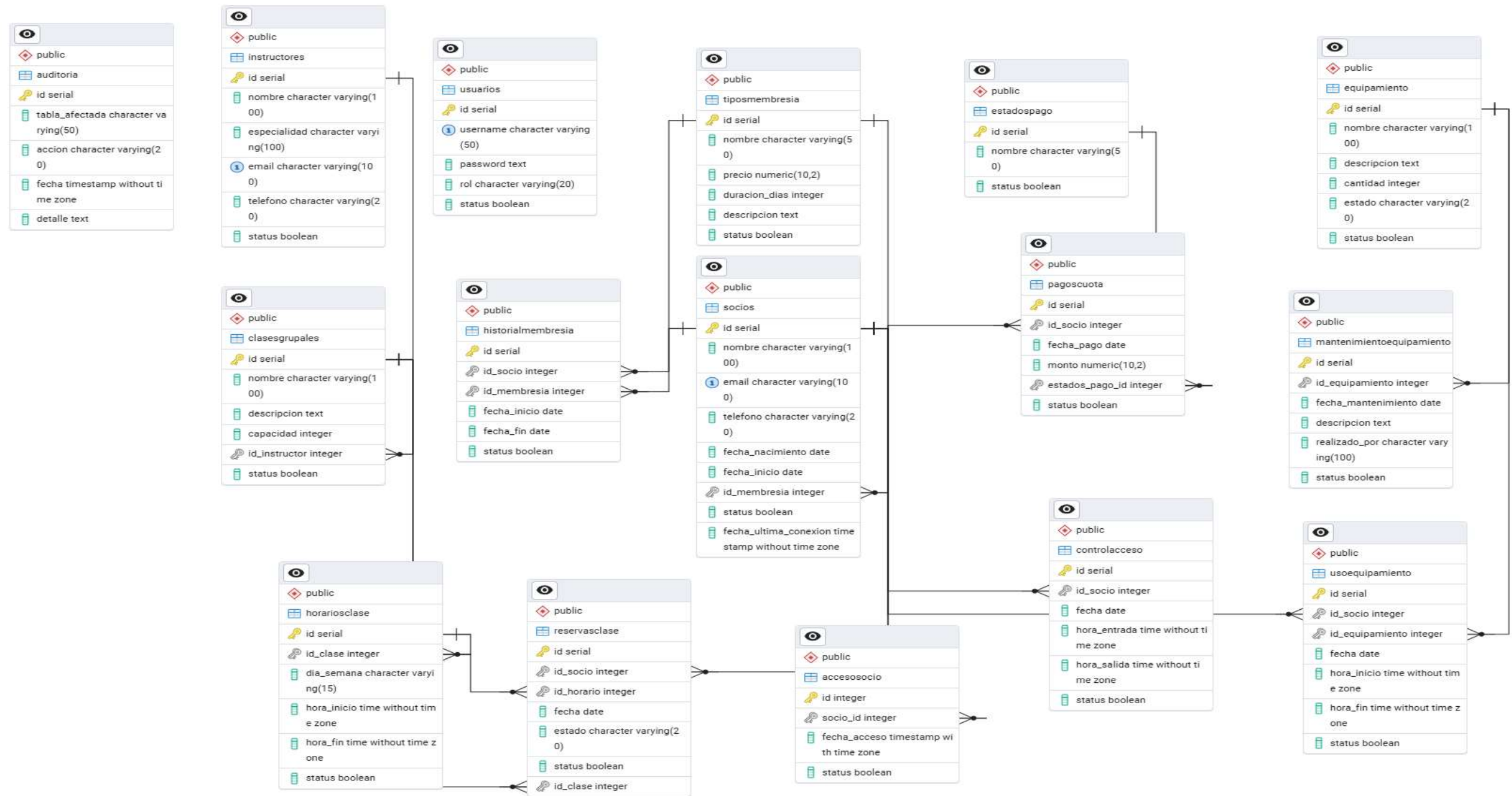
Equipamiento: Uso, mantenimiento y asignación a socios.

Usuarios: Control de autenticación y roles administrativos.

- Auditoría: Registro de acciones realizadas en el sistema.

Cada módulo está relacionado con claves foráneas y permite una gestión integral de operaciones, membresías, pagos, clases y recursos del gimnasio.





INDEXACIÓN

Consulta Historial de Consultas

```
1 CREATE INDEX idx_reservas_socio ON ReservasClase(id_socio);
```

Data Output Mensajes Notificaciones

CREATE INDEX

Consulta retornó exitosamente en 225 msec.

CREATE INDEX idx_reservas_socio ON
ReservasClase(id_socio);

CREATE INDEX idx_clase_instructor ON ClasesGrupales(id_instructor);

CREATE INDEX idx_socio_email ON Socios(email);

Consulta Historial de Consultas

```
1 CREATE INDEX idx_socio_email ON Socios(email);
```

```
2
```

Data Output Mensajes Notificaciones

CREATE INDEX

Consulta retornó exitosamente en 887 msec.

Consulta Historial de Consultas

```
1 CREATE INDEX idx_clase_instructor ON ClasesGrupales(id_instructor);
```

```
2
```

Data Output Mensajes Notificaciones

CREATE INDEX

Consulta retornó exitosamente en 783 msec.



Consulta Historial de Consultas

```
1 CREATE VIEW ClasesConCupo AS
2 SELECT c.id, c.nombre, c.capacidad, COUNT(r.id) AS reservados
3 FROM ClasesGrupales c
4 JOIN HorariosClase h ON c.id = h.id_clase
5 LEFT JOIN ReservasClase r ON r.id_horario = h.id
6 GROUP BY c.id
7 HAVING COUNT(r.id) < c.capacidad;
```

Data Output Mensajes Notificaciones

CREATE VIEW

Consulta retornó exitosamente en 175 msec.

Consulta Historial de Consultas

```
1 CREATE VIEW HistorialAcceso AS
2 SELECT s.id AS id_socio, s.nombre, c.fecha, c.hora_entrada, c.hora_salida
3 FROM Socios s
4 JOIN ControlAcceso c ON s.id = c.id_socio;
```

Data Output Mensajes Notificaciones

CREATE VIEW

Consulta retornó exitosamente en 179 msec.

Consulta Historial de Consultas

```
1 CREATE VIEW EquiposPendientesMantenimiento AS
2 SELECT e.id, e.nombre, MAX(m.fecha_mantenimiento) AS ultimo_mantenimiento
3 FROM Equipamiento e
4 LEFT JOIN MantenimientoEquipamiento m ON e.id = m.id_equipamiento
5 GROUP BY e.id
6 HAVING CURRENT_DATE - MAX(m.fecha_mantenimiento) > 30;
```

Data Output Mensajes Notificaciones

CREATE VIEW

Consulta retornó exitosamente en 190 msec.

Consulta Historial de Consultas

```
1 CREATE VIEW SociosPendientes AS
2 SELECT s.id, s.nombre, p.fecha_pago, e.nombre AS estado
3 FROM Socios s
4 JOIN PagosCuota p ON s.id = p.id_socio
5 JOIN EstadosPago e ON p.estados_pago_id = e.id
6 WHERE e.nombre = 'Pendiente';
7
```

Data Output Mensajes Notificaciones

CREATE VIEW

Consulta retornó exitosamente en 360 msec.

Consulta Historial de Consultas

```
1 CREATE VIEW UsoRecienteEquipamiento AS
2 SELECT u.id_equipamiento, e.nombre, u.fecha, u.hora_inicio, u.hora_fin
3 FROM UsoEquipamiento u
4 JOIN Equipamiento e ON u.id_equipamiento = e.id
5 WHERE u.fecha >= CURRENT_DATE - INTERVAL '7 days';
```

Data Output Mensajes Notificaciones

CREATE VIEW

Consulta retornó exitosamente en 214 msec.

VISTAS



Procedimiento	Descripción	Tabla Afectada	Operación
insertar_socio	Inserta un nuevo socio en el sistema.	Socios	INSERT
crear_clase_grupal	Crea una clase grupal y su horario asociado.	ClasesGrupales, HorariosClase	INSERT
actualizar_clase_grupal	Modifica los datos de una clase grupal existente.	ClasesGrupales	UPDATE
eliminar_clase_grupal	Elimina una clase grupal del sistema.	ClasesGrupales	DELETE
crear_reserva	Crea una nueva reserva de clase para un socio.	ReservasClase	INSERT
actualizar_reserva	Modifica los datos de una reserva existente.	ReservasClase	UPDATE
eliminar_reserva	Elimina una reserva existente.	ReservasClase	DELETE
crear_pago_cuota	Registra un nuevo pago de cuota para un socio.	PagosCuota	INSERT
actualizar_pago_cuota	Actualiza la información de un pago de cuota.	PagosCuota	UPDATE
eliminar_pago_cuota	Elimina un pago de cuota registrado.	PagosCuota	DELETE

STORED PROCEDURES



TRIGGERS

Consulta Historial de Consultas

```
1 CREATE TRIGGER auditoria_clase
2 AFTER INSERT OR UPDATE OR DELETE ON ClasesGrupales
3 FOR EACH ROW EXECUTE FUNCTION auditoria_funcion();
```

Data Output Mensajes Notificaciones

CREATE TRIGGER

Consulta retornó exitosamente en 174 msec.

Consulta Historial de Consultas

```
1 CREATE TRIGGER auditoria_pago
2 AFTER INSERT OR UPDATE OR DELETE ON PagosCuota
3 FOR EACH ROW EXECUTE FUNCTION auditoria_funcion();
```

Data Output Mensajes Notificaciones

CREATE TRIGGER

Consulta retornó exitosamente en 483 msec.

Consulta Historial de Consultas

```
1 CREATE TRIGGER auditoria_reserva
2 AFTER INSERT OR UPDATE OR DELETE ON ReservasClase
3 FOR EACH ROW EXECUTE FUNCTION auditoria_funcion();
```

Data Output Mensajes Notificaciones

CREATE TRIGGER

Consulta retornó exitosamente en 149 msec.

Consulta Historial de Consultas

```
1 CREATE TRIGGER auditoria_socio
2 AFTER INSERT OR UPDATE OR DELETE ON Socios
3 FOR EACH ROW EXECUTE FUNCTION auditoria_funcion();
```

Data Output Mensajes Notificaciones

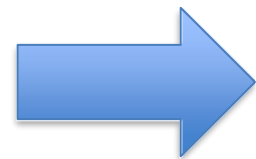
CREATE TRIGGER

Consulta retornó exitosamente en 165 msec.



CONSULTAS

CONSULTA TIPOS MENBRESIA -1



SUB CONSULTA SOCIO-1

Query

Query History

```

1 SELECT *
2 FROM Socios
3 WHERE id_membresia IN (
4     SELECT id FROM TiposMembresia WHERE precio > 100
5 );

```

Data Output

Messages

Notifications

SQL

Showing rows: 1 to 6

Page No

	id [PK] integer	nombre character varying (100)	email character varying (100)	telefono character varying (20)	fecha_nacimiento date	fecha_inicio date	id_membresia integer	status boolean	fecha_ultima_conexion timestamp without time zone
1	1	Ana Martínez	ana@example.com	3001230001	1992-03-10	2025-05-14	1	true	[null]
2	2	Luis Pérez	luis@example.com	3001230002	1988-07-22	2025-05-14	2	true	[null]
3	3	María López	maria@example.com	3001230003	1995-11-30	2025-05-14	3	true	[null]
4	4	Carlos García	carlos@example.com	3001230004	1990-01-15	2025-05-14	4	true	[null]
5	5	Elena Torres	elena@example.com	3001230005	1985-09-18	2025-05-14	5	true	[null]
6	8	Juan Pérez	juan@example.com	3001234567	1990-05-01	2025-05-16	1	true	[null]

2	Básica	Ana Martínez
3	Premium	Luis Pérez
4	Trimestral	María López
5	Anual	Carlos García
6	Clases Solo	Elena Torres

CONSULTAS INSTRUCTORES -1



Query Query History

```
1 SELECT i.nombre AS instructor, c.nombre AS clase
2 FROM Instructores i
3 INNER JOIN ClasesGrupales c ON i.id = c.id_instructor;
```

Data Output Messages Notifications

	instructor character varying (100)	clase character varying (100)
1	Ana Gómez	Yoga Relax
2	Luis Mora	Crossfit Intenso
3	Marta León	Pilates Básico
4	David Pardo	Entrenamiento Funcional
5	Ana Gómez	Yoga

SUB CONSULTA CLASES GRUPALES -2



Query Query History

```
1 SELECT *
2 FROM ClasesGrupales
3 WHERE id_instructor IN (
4     SELECT id FROM Instructores WHERE especialidad = 'Yoga'
5 );
```

Data Output Messages Notifications

	id [PK] integer	nombre character varying (100)	descripcion text	capacidad integer	id_instructor integer	status boolean
1	2	Yoga Relax	Relajación y respiración	12	2	true
2	6	Yoga	Clase de relajación y estiramien...	15	2	true

Query Query History

```
1 SELECT e.nombre AS equipo, m.fecha_mantenimiento
2 FROM Equipamiento e
3 LEFT JOIN MantenimientoEquipamiento m ON e.id = m.id_equipamiento;
```

Data Output Messages Notifications

	equipo character varying (100)	fecha_mantenimiento date
1	Caminadoras	2024-05-01
2	Bicicletas	2024-04-15
3	Pesas	2024-03-22
4	Máquinas de pecho	2024-03-10
5	Elípticas	2024-02-28

CONSULTAS DE EQUIPAMIENTO -2



OPTIMIZACIÓN

Query Query History

```
1 WITH CuposDisponibles AS (  
2     SELECT c.id AS id_clase, c.capacidad, COUNT(r.id) AS reservados  
3     FROM ClasesGrupales c  
4     JOIN HorariosClase h ON c.id = h.id_clase  
5     LEFT JOIN ReservasClase r ON h.id = r.id_horario  
6     GROUP BY c.id  
7 )  
8 SELECT id_clase, capacidad - reservados AS cupos_disponibles  
9 FROM CuposDisponibles;
```

Data Output Messages Notifications

	id_clase integer	cupos_disponibles bigint
1	3	19
2	5	17
3	4	9
4	2	11

Query Query History

```
1 WITH SociosActivos AS (  
2     SELECT id FROM Socios WHERE status = TRUE  
3 )  
4 SELECT COUNT(*) FROM SociosActivos;
```

Data Output Messages Notifications

	count bigint
1	6

Rol	Usuario	Descripción	Permisos principales
admin_gimnasio	admin	Administrador general con control total sobre la base de datos.	- Conexión a la BD- Todos los privilegios sobre todas las tablas y secuencias del esquema public
repcionista_role	repcionista	Personal de recepción que gestiona socios, pagos y reservas.	- Conexión a la BD- SELECT, INSERT, UPDATE en: Socios, PagosCuota, ControlAcceso, ReservasClase, HistorialMembresia- Uso limitado de secuencias
socio_role	socio	Usuario del gimnasio que gestiona sus reservas e información personal.	- Conexión a la BD- SELECT, INSERT en: ReservasClase, ControlAcceso- SELECT en: TiposMembresia, PagosCuota- Acceso limitado a secuencias
instructor_role	instructor	Instructor que consulta clases, horarios y control de asistencia.	- Conexión a la BD- SELECT en: ClasesGrupales, HorariosClase, Socios, ReservasClase, UsoEquipamiento- Lectura en secuencias relacionadas
auditor_role	auditor	Auditor que revisa la base de datos sin hacer modificaciones.	- Conexión a la BD - Todos los privilegios sobre todas las tablas y secuencias del esquema public

GRACIAS POR SU ATENCIÓN

