

INTEGRANTES:

-Rene Antonio Vasquez Torres - 00122224.

-Kevin Daniel Rosa Pacheco - 00210024.

-Bryan Gabriel Alfaro Canizalez - 00221324.

1. Introducción

El presente informe documenta el diseño, implementación y aseguramiento de la base de datos del Sistema de Reservas de Gimnasio implementado en SQL Server 2022. El sistema está pensado para la administración integral de varias sedes de un gimnasio, permitiendo gestionar miembros, entrenadores, empleados administrativos, clases, equipamiento, inscripciones y pagos, así como los procesos de mantenimiento y auditoría de la información.

El documento se estructura de manera que acompaña el ciclo de vida del proyecto de base de datos: desde la descripción general del sistema y el modelo relacional, hasta las políticas de seguridad, las consultas optimizadas y la estrategia de respaldo y recuperación. De esta forma, el informe sirve tanto como evidencia técnica del trabajo realizado como guía de referencia para futuras extensiones del sistema, incluyendo la integración con herramientas de inteligencia de negocios.

1.1 Presentación del Sistema de Reservas de Gimnasio

El Sistema de Reservas de Gimnasio tiene como objetivo centralizar la gestión operativa de un conjunto de sedes deportivas. La solución contempla la administración de sedes (gimnasios), personas asociadas al gimnasio (miembros, entrenadores y empleados), clases o áreas de entrenamiento, equipamiento físico, inscripciones a clases e historial de pagos. Adicionalmente, incorpora el registro de actividades de mantenimiento sobre los equipos y un módulo de auditoría que almacena los cambios relevantes sobre tablas críticas.

Desde la perspectiva de negocio, el sistema permite responder preguntas como: ¿cuáles son las sedes con mayor ingreso por pagos?, ¿qué miembros mantienen membresías activas?, ¿qué entrenadores están asignados a determinadas áreas?, o ¿qué equipos requieren mantenimiento con mayor urgencia? Todo ello se apoya en un modelo relacional normalizado y en consultas optimizadas que facilitan el análisis posterior de la información.

1.2 Objetivo del informe

El objetivo principal de este informe es documentar de forma clara y estructurada el diseño lógico de la base de datos del Sistema de Reservas de Gimnasio, así como las decisiones técnicas asociadas a seguridad, auditoría, rendimiento y respaldo. En particular, se busca:

- Describir el propósito y alcance funcional del sistema desde la perspectiva de la base de datos.
- Presentar el modelo entidad–relación y las principales entidades del esquema lógico.
- Detallar el modelo de roles, permisos y mecanismos de control de acceso implementados.
- Resumir las consultas optimizadas, los índices aplicados y su impacto esperado en el

rendimiento.

- Exponer la estrategia de dimensionamiento y la propuesta de respaldo y recuperación de la información.

De esta manera, el documento constituye un entregable académico alineado a los criterios de evaluación institucional y una base técnica para la operación y evolución futura del sistema.

1.3 Tecnologías y enfoques utilizados

La implementación del proyecto se realizó empleando las siguientes tecnologías y enfoques:

- SQL Server 2022 como motor de base de datos relacional, aprovechando características de seguridad, auditoría y respaldo.
- Modelado relacional normalizado, diferenciando esquemas lógicos para separar datos maestros, datos de membresía y datos operativos.
- Mecanismos de seguridad basados en esquemas, roles y permisos granulares, complementados con procedimientos almacenados seguros.
- Auditoría mediante triggers que registran las operaciones de inserción, actualización y eliminación sobre tablas críticas en una tabla centralizada de auditoría.
- Optimización de consultas a través de índices no cluster y vistas de negocio, orientadas a análisis por sede, tipo de membresía y comportamiento de pagos.
- Integración futura con Power BI para construir tableros de análisis de reservas, ingresos y uso de instalaciones, a partir de vistas y consultas ya definidas.

2. Descripción General del Sistema

El sistema de información se centra en la gestión de varias sedes de un gimnasio, donde cada sede posee sus propias clases o áreas de entrenamiento y su inventario de equipo. En este contexto se administran distintos tipos de personas: miembros regulares, entrenadores y empleados administrativos o de servicio. Cada miembro puede estar asociado a una membresía (plan Silver, Gold o Diamond), inscribirse a clases y generar pagos periódicos. Por su parte, los entrenadores se vinculan a sedes y clases específicas de acuerdo con su especialidad, mientras que el personal administrativo se ocupa de tareas de mantenimiento, limpieza y gestión.

El sistema resuelve la necesidad de contar con una base de datos coherente, integridad referencial controlada y trazabilidad de operaciones, de forma que se pueda dar soporte a la operación diaria del gimnasio y a futuros procesos de análisis de datos.

2.1 Propósito del sistema

El propósito de la base de datos del Sistema de Reservas de Gimnasio es proporcionar una plataforma centralizada y confiable para almacenar toda la información relevante de las sedes, sus miembros y las operaciones asociadas. Esta plataforma debe:

- Garantizar la integridad de los datos mediante claves primarias, foráneas, restricciones y reglas de negocio.
- Facilitar el registro y consulta eficiente de inscripciones, pagos y mantenimiento de equipos.
- Permitir el control de acceso diferenciado según el rol del usuario (administrador, entrenador,

recepción).

- Servir como base para la generación de reportes operativos y analíticos.

2.2 Alcance funcional

Desde la perspectiva de la base de datos, el sistema cubre las siguientes funciones principales:

- Administración de sedes (gimnasios), incluyendo datos de contacto y localización.
- Registro de personas en una tabla maestra de miembros, diferenciando tipos: Member, Trainer y Employee.
- Gestión de membresías para miembros regulares, con fechas de inicio y fin y estado de vigencia.
- Registro de perfiles de entrenadores, con sus especialidades y vigencia.
- Registro de perfiles de empleados, con el tipo de servicio que brindan (limpieza, mantenimiento, administración).
- Administración de clases o áreas por sede, así como del equipamiento asociado a cada clase.
- Inscripción de miembros a clases, controlando duplicidades mediante claves únicas.
- Gestión de pagos de membresía y su posible archivado para fines de retención de datos.
- Registro de mantenimientos realizados a equipos, incluyendo responsables y fechas programadas.
- Registro de auditoría de operaciones sobre tablas clave a través de triggers especializados.

2.3 Flujo general de operación

De forma resumida, el flujo de operación contemplado en el sistema es el siguiente:

- 1) Se registran las sedes del gimnasio en la tabla de gimnasios. A cada sede se le asocian clases o áreas de entrenamiento y el inventario de equipo correspondiente.
- 2) Se registran las personas en la tabla maestra de miembros, indicando el tipo de miembro. A partir de esta tabla se derivan los perfiles de miembro regular, entrenador y empleado mediante tablas 1:1.
- 3) Para los miembros de tipo "Member" se crean registros de membresía, indicando el tipo de plan, fechas de vigencia y estado calculado. Estos miembros pueden inscribirse a clases mediante la tabla de inscripciones.
- 4) El pago de las membresías se registra en la tabla de pagos, ya sea mediante procedimientos almacenados que calculan el monto según el tipo de plan, o mediante cargas masivas de pagos aleatorios para propósitos de prueba.
- 5) Los entrenadores se asocian a sedes y clases de acuerdo con su especialidad, permitiendo posteriormente consultas sobre disponibilidad y carga de trabajo.
- 6) El personal de mantenimiento y otros empleados registran actividades sobre equipos en la tabla de mantenimiento, dejando evidencia sobre el tipo de intervención y la fecha del próximo mantenimiento sugerido.
- 7) A lo largo de estas operaciones, triggers de auditoría insertan registros en una tabla centralizada de log cada vez que se inserta, actualiza o elimina información en las tablas críticas de miembros, perfiles, clases y equipamiento.

3. Modelo de Datos (ER)

El modelo de datos del sistema se diseñó siguiendo principios de normalización y separando la información en esquemas lógicos que representan distintos dominios: GymCore para la estructura física del gimnasio, MemberData para la información de personas y OperationsData para los procesos operativos. Adicionalmente, se dispone del esquema dbo para elementos transversales como el registro de auditoría.

En esta sección se describen las entidades principales y las relaciones estructurales que conforman el diagrama entidad-relación general del sistema. En el documento físico se reserva un espacio para la inclusión del diagrama ER completo generado a partir del modelo lógico.

3.1 Entidades principales

Las entidades principales del modelo se agrupan de acuerdo con el esquema al que pertenecen:

- Esquema GymCore:
 - TB_Gym: representa cada sede del gimnasio, con su nombre, dirección, teléfono y correo electrónico.
 - TB_Class: modela las clases o áreas de entrenamiento disponibles en cada sede (sala de pesas, área cardiovascular, piscina, etc.).
 - TB_Equipment: describe el equipamiento disponible en cada clase, registrando tipo, descripción y fecha del último mantenimiento.
- Esquema MemberData:
 - TB_Member: tabla maestra de personas, que almacena todos los individuos asociados a la organización. Incluye atributos de identificación, contacto, tipo de miembro y referencia a la sede a la que pertenecen.
 - TB_Membership: almacena la información de la membresía de los miembros regulares, con tipo de plan, fechas de vigencia y un campo calculado de estado (activo/expirado).
 - TB_TrainerProfile: representa el perfil de los entrenadores, indicando la especialidad y el periodo en que ejercen dicha función.
 - TB_EmployeeProfile: registra el perfil de los empleados de servicio o administrativos, incluyendo el tipo de servicio y fechas de vigencia.
- Esquema OperationsData:
 - TB_Enrollment: registra la inscripción de miembros a clases específicas.
 - TB_Payment: almacena el historial de pagos efectuados por los miembros.
 - TB_PaymentArchive: conserva los pagos archivados como parte de la política de retención de datos cuando un miembro es eliminado.
 - TB_Maintenance: almacena los reportes de mantenimiento realizados sobre equipos, asociados a un miembro responsable.
- Esquema dbo:
 - TB_AuditLog: tabla general de auditoría donde se registran la tabla afectada, el tipo de acción, el usuario y la fecha del evento.

3.2 Relaciones estructurales

El diagrama entidad-relación general se caracteriza por las siguientes relaciones clave:

- Una sede (TB_Gym) se relaciona con múltiples clases (TB_Class) en una relación 1:N.
- Cada clase se relaciona con múltiples equipos (TB_Equipment), también en una relación 1:N.
- Cada sede tiene múltiples miembros (TB_Member), establecidos mediante una clave foránea desde MemberData.TB_Member hacia GymCore.TB_Gym.
- La tabla TB_Member actúa como entidad maestra para los distintos tipos de perfil:
 - Relación 1:1 entre TB_Member y TB_Membership para miembros de tipo "Member".
 - Relación 1:1 entre TB_Member y TB_TrainerProfile para miembros de tipo "Trainer".
 - Relación 1:1 entre TB_Member y TB_EmployeeProfile para miembros de tipo "Employee".
- Los miembros de tipo "Member" pueden inscribirse a múltiples clases mediante TB_Enrollment, que implementa una relación N:M entre miembros y clases, restringida con una clave única para evitar inscripciones duplicadas.
- Cada miembro puede generar múltiples pagos en TB_Payment, y estos, a su vez, pueden ser movidos a TB_PaymentArchive cuando se realiza un proceso de archivado previo a la eliminación del miembro.
- Las actividades de mantenimiento (TB_Maintenance) se relacionan tanto con el equipo intervenido (TB_Equipment) como con el miembro responsable, a través de claves foráneas.
- Finalmente, las operaciones de inserción, actualización y eliminación en tablas críticas generan registros en TB_AuditLog mediante triggers, relacionando lógicamente la bitácora con el resto del modelo.

En el documento se reserva un espacio para incorporar el diagrama ER completo, donde estas relaciones se observan de forma gráfica.

4. Políticas de Seguridad Implementadas

La seguridad de la base de datos se implementó combinando la organización por esquemas, la definición de roles específicos y la asignación controlada de permisos sobre objetos y procedimientos almacenados. Adicionalmente, se incorporó un mecanismo de auditoría que permite rastrear operaciones sobre tablas sensibles. Esta sección describe el modelo de seguridad adoptado y su relación con las funciones de negocio.

4.1 Modelo de roles

Se definieron roles de base de datos alineados a los perfiles funcionales del gimnasio:

- Role_Admin: rol de administración completa de la base de datos a nivel lógico del proyecto. Este rol tiene control sobre los esquemas de negocio y puede ejecutar todos los procedimientos almacenados.
- Role_Trainer: rol destinado a usuarios que representan entrenadores. Tiene permisos de lectura sobre la información de miembros y clases, y puede registrar actividades de mantenimiento mediante la tabla correspondiente y el procedimiento almacenado asociado.
- Role_Reception: rol orientado al personal de recepción. Puede administrar la información de miembros (altas, modificaciones y consultas), gestionar pagos e inscripciones a clases, y ejecutar los procedimientos almacenados para registrar pagos, con restricciones explícitas para evitar

eliminaciones directas de miembros.

Cada rol se vincula a usuarios de base de datos, los cuales a su vez se asocian a logins definidos a nivel de servidor, asegurando una separación clara entre autenticación (login) y autorización (rol dentro de la base de datos).

4.2 Arquitectura de esquemas

La base de datos se organiza en cuatro esquemas principales: GymCore, MemberData, OperationsData y dbo. Esta arquitectura permite agrupar objetos con características funcionales similares y aplicar políticas de permisos a nivel de esquema. De forma general:

- GymCore agrupa las tablas centrales de estructura del gimnasio (sedes, clases y equipos).
- MemberData concentra la información de personas y perfiles asociados.
- OperationsData incluye las tablas de procesos transaccionales (inscripciones, pagos y mantenimiento).
- dbo se reserva para componentes transversales, como la tabla de auditoría y los procedimientos almacenados.

Sobre esta base, el rol de administrador tiene control completo sobre los esquemas de negocio, mientras que los roles de entrenador y recepción reciben permisos más restringidos, según sus necesidades.

4.3 Políticas de permisos por esquema

Las políticas de permisos se definen principalmente a nivel de esquema, complementadas con permisos específicos sobre tablas y procedimientos:

- Role_Admin recibe permisos de CONTROL sobre los esquemas GymCore, MemberData y OperationsData, además de permiso EXECUTE sobre todos los procedimientos almacenados. Esto le permite administrar el modelo y ejecutar la lógica de negocio.
- Role_Trainer tiene permisos de SELECT sobre los esquemas MemberData y GymCore, así como permisos de INSERT y UPDATE sobre la tabla de mantenimiento, y EXECUTE sobre el procedimiento de registro de mantenimiento.
- Role_Reception cuenta con permisos de SELECT, INSERT, UPDATE y DELETE sobre las tablas de MemberData, excepto la eliminación directa de miembros, que se deniega explícitamente para obligar el uso de procedimientos seguros. Además, puede gestionar pagos e inscripciones y ejecutar el procedimiento encargado de generar pagos según la membresía.

Estas políticas aseguran el principio de mínimo privilegio, otorgando a cada rol únicamente los permisos necesarios para desempeñar sus funciones.

4.4 Mecanismos de control de acceso

El control de acceso se implementa utilizando la combinación de logins de servidor, usuarios de base de datos, roles y restricciones mediante DENY en operaciones sensibles. La eliminación de miembros, por ejemplo, se restringe mediante un DENY DELETE sobre la tabla de miembros para el rol de recepción; en su lugar, se proporciona un procedimiento almacenado que primero archiva los pagos asociados y luego elimina de forma ordenada los registros relacionados. Esto permite preservar la integridad referencial y mantener un historial de pagos en tablas de archivo.

Adicionalmente, el uso de procedimientos almacenados centraliza la lógica de negocio crítica (registro de pagos, mantenimiento y eliminación de miembros y sedes), reduciendo la exposición de las tablas a operaciones directas por parte de los usuarios finales.

4.5 Arquitectura de auditoría

La auditoría se implementa mediante una arquitectura basada en triggers y una tabla central de log. Cada vez que se inserta, actualiza o elimina información en tablas seleccionadas, un trigger registra la tabla afectada, el tipo de acción realizada, el usuario que ejecutó la operación y la fecha y hora del evento. Esto proporciona trazabilidad y soporte para revisiones posteriores.

Si bien la auditoría se concentra principalmente a nivel de base de datos, se apoya en la gestión de logins a nivel de servidor para identificar correctamente al usuario responsable de cada transacción.

4.5.1 Auditoría a nivel de servidor

A nivel de servidor, la seguridad se establece mediante la creación de logins específicos para cada perfil (por ejemplo, GymAdminLogin, TrainerLogin y ReceptionLogin). Estos logins se utilizan para la autenticación de los usuarios y se asocian posteriormente a usuarios de base de datos. De esta forma, las entradas en la tabla de auditoría pueden registrar el nombre del usuario que originó cada cambio.

3.5.2 Auditoría a nivel de base de datos

Dentro de la base de datos, la auditoría se materializa en la tabla TB_AuditLog, complementada por triggers definidos sobre tablas clave como TB_Member, TB_Membership, TB_TrainerProfile, TB_EmployeeProfile, TB_Class y TB_Equipment. Cada trigger se dispara después de operaciones de INSERT, UPDATE o DELETE e inserta en TB_AuditLog un registro con el nombre de la tabla, el tipo de acción (INSERT, UPDATE o DELETE), el usuario y la fecha y hora del evento.

Este enfoque permite monitorear modificaciones relevantes sobre la estructura de miembros, perfiles y recursos físicos del gimnasio.

4.5.3 Beneficios del sistema de auditoría

El sistema de auditoría aporta varios beneficios concretos:

- Trazabilidad de cambios sobre datos críticos, útil para revisiones internas y resolución de incidencias.
- Apoyo al control interno, al permitir identificar qué usuario realizó determinadas operaciones.
- Base para futuros reportes de auditoría o indicadores de calidad de datos.
- Complemento a los mecanismos de seguridad, al disuadir modificaciones indebidas y facilitar su detección temprana.

4.5.4 Acceso y consulta de logs

El acceso a la tabla TB_AuditLog se restringe a perfiles administrativos, de forma que su contenido no sea visible para usuarios operativos. Las consultas sobre esta tabla pueden filtrar por

tabla, tipo de operación, usuario o rango de fechas, permitiendo reconstruir la secuencia de cambios sobre una entidad específica o detectar patrones inusuales de modificación.

5. Consultas Optimizadas e Índices Aplicados

El rendimiento de la base de datos se abordó mediante la creación de índices no cluster sobre columnas frecuentemente utilizadas en filtros y agrupamientos, así como mediante consultas diseñadas para aprovechar estas estructuras. Además, se definieron vistas de negocio para facilitar el análisis de información por parte de herramientas externas.

Entre los índices implementados destacan:

- IDX_GymId_BirthDate sobre MemberData.TB_Member, que indexa conjuntamente el identificador de la sede y la fecha de nacimiento, incluyendo nombre, apellido y tipo de miembro. Este índice mejora consultas que segmentan a los miembros por sede y rangos de edad.
- IDX_MemberType_MemberId sobre MemberData.TB_Member, que permite filtrar de manera eficiente por tipo de miembro (Member, Trainer, Employee) y recorrer rápidamente los identificadores asociados.

A nivel de consultas, se desarrollaron sentencias que responden preguntas de negocio como:

- Sedes con mayor ingreso total por pagos.
- Distribución de miembros por tipo de membresía y estado de vigencia.
- Relación entre entrenadores, sus especialidades y las clases a las que están vinculados.
- Equipos que requieren mantenimiento por haber superado cierto número de días desde su último servicio.
- Antigüedad de los empleados administrativos en cada sede.
- Ingreso acumulado por sede a lo largo del tiempo, utilizando funciones de ventana.

Asimismo, se definieron vistas como VW_MiembrosActivosDetalle y VW_HistoricoPagosResumen, que consolidan información de varias tablas y sirven como fuente directa para reportes o herramientas de análisis.

6. Estrategia de Dimensionamiento, Respaldo y Recuperación

La base de datos se creó considerando un dimensionamiento inicial suficiente para soportar cargas de datos masivas y un esquema de respaldo que permita recuperar la información ante fallos. Aunque algunos elementos de la estrategia de respaldo aún se encuentran en fase de propuesta, ya se estableció el modelo de recuperación y se realizaron respaldos iniciales completos, diferenciales y de log.

En esta sección se detalla el enfoque general adoptado y los escenarios de recuperación previstos, así como las implicaciones para la operación del gimnasio.

6.1 Enfoque general de respaldo y recuperación

El enfoque general parte de configurar la base de datos en modelo de recuperación FULL, lo que permite realizar respaldos del log de transacciones y posibilita la recuperación a un punto en el tiempo. Sobre esta base, se plantea una estrategia de tres niveles:

- Respaldo completo periódico de la base de datos, que captura el estado completo de los datos.
- Respaldo diferencial que almacene únicamente los cambios desde el último backup completo.
- Respaldo de log de transacciones en intervalos más cortos, para minimizar la pérdida potencial de datos.

En el contexto del proyecto, se generaron respaldos completos, diferenciales y de log de manera manual para validar el funcionamiento del esquema. La automatización mediante trabajos de SQL Server Agent se deja como línea de trabajo futura.

6.2 Procesos de recuperación

A partir de los respaldos disponibles, se contemplan dos escenarios principales de recuperación:

- Escenario de recuperación completa: ante una pérdida total de la base de datos, se restauraría primero el respaldo completo más reciente y, a continuación, el último respaldo diferencial disponible. Finalmente, se aplicarían los respaldos de log en orden cronológico para llevar la base de datos al punto deseado. Este procedimiento permite una recuperación casi completa de la información, con pérdida mínima dependiendo de la frecuencia de los respaldos de log.
- Recuperación selectiva: en situaciones donde el problema afecta únicamente a un subconjunto de datos (por ejemplo, eliminación accidental de registros), es posible restaurar los respaldos en una base de datos temporal y extraer desde allí los registros necesarios para reinsertarlos en la base de datos principal. Este enfoque reduce el impacto sobre la operación diaria, ya que no requiere sustituir completamente la base de datos en producción.

6.3 Monitoreo y verificación de respaldos

Aunque en el alcance del proyecto los respaldos se ejecutan de forma manual, se enfatiza la importancia de:

- Verificar que los archivos de respaldo se generan correctamente y se almacenan en ubicaciones seguras.
- Realizar pruebas periódicas de restauración en entornos controlados para validar la consistencia de los respaldos y estimar los tiempos de recuperación.
- Documentar el calendario de respaldos y los responsables de su ejecución y supervisión.

En un entorno productivo, estas tareas se complementarían con monitoreo automatizado, alertas ante fallos de backup y políticas formales de retención de respaldos.

6.4 Beneficios para la operación del gimnasio

Contar con una estrategia de dimensionamiento y respaldo bien definida aporta beneficios directos a la operación del gimnasio:

- Reduce el riesgo de pérdida de información de miembros, pagos y mantenimiento, que son

críticos para la continuidad del negocio.

- Permite recuperar rápidamente la base de datos ante fallos, minimizando tiempos de inactividad.
- Proporciona una base sólida para llevar la solución a producción o escalarla a nuevos escenarios con mayor volumen de datos.
- Aporta confianza a los usuarios y a la administración, al demostrar que se han considerado aspectos de seguridad, auditoría y continuidad operativa desde el diseño del sistema.

En conjunto, el modelo relacional, las políticas de seguridad, las consultas optimizadas y la estrategia de respaldo y recuperación convierten a la base de datos del Sistema de Reservas de Gimnasio en un componente robusto y preparado para soportar tanto la operación actual como futuras integraciones con herramientas de análisis y visualización de datos.

7. Diccionario de datos

Tabla 1: dbo.TB_AuditLog

Campo	Tipo	Nulo	Clave	Descripción
TableName	VARCHAR(50)	NO	–	Nombre de la tabla afectada.
ActionType	VARCHAR(50)	NO	–	Tipo de acción (INSERT, UPDATE, DELETE).
UserName	VARCHAR(50)	NO	–	Usuario de SQL que ejecutó la acción.
AuditDate	DATETIME	NO	–	Fecha y hora del evento de auditoría.

Tabla 2: GymCore.TB_Gym

Campo	Tipo	Nulo	Clave	Descripción
GymId	INT IDENTITY(1,1)	NO	PK	Identificador de la sede/gimnasio.
GymName	VARCHAR(100)	NO	UNIQUE	Nombre de la sede (único).
Address	VARCHAR(255)	Sí	–	Dirección física del gimnasio.
Phone	VARCHAR(20)	Sí	–	Teléfono de contacto.
Email	VARCHAR(80)	Sí	–	Correo electrónico del gimnasio.

Tabla 3: MemberData.TB_Member

Campo	Tipo	Nulo	Clave	Descripción
MemberId	INT IDENTITY(1000,1)	NO	PK	Identificador único de la persona.
GymId	INT	NO	FK → GymCore.TB_Gym(GymId)	Sede a la que pertenece el miembro.
MemberType	VARCHAR(15)	NO	–	Tipo de persona: Member, Trainer, Employee. (CHECK).
FirstName	VARCHAR(50)	NO	–	Primer nombre del miembro.
LastName	VARCHAR(50)	NO	–	Apellido del miembro.
BirthDate	DATE	Sí	–	Fecha de nacimiento.
Phone	VARCHAR(20)	Sí	–	Teléfono de contacto.
Email	VARCHAR(80)	Sí	UNIQUE	Correo del miembro (único).

Tabla 4: MemberData.TB_Membership

Campo	Tipo	Nulo	Clave	Descripción
MembershipId	INT IDENTITY(1, 1)	NO	PK	Identificador del perfil de membresía.
MemberId	INT	NO	FK → MemberData.TB_Member(MemberId), UNIQUE	Miembro asociado (relación 1:1).
MembershipType	VARCHAR(20)	NO	–	Tipo de membresía: Silver, Gold, Diamond. (CHECK).
StartDate	DATE	NO	–	Fecha de inicio de la membresía (DEFAULT GETDATE()).
EndDate	DATE	Sí	–	Fecha de fin de la membresía.
IsActive	AS (calc.) PERSISTED	NO	–	Indicador calculado (1 = activa si EndDate ≥ GETDATE()).

Tabla 5: MemberData.TB_TrainerProfile

Campo	Tipo	Nulo	Clave	Descripción
TrainerProfileId	INT IDENTITY(1,1)	NO	PK	Identificador del perfil de entrenador.
MemberId	INT	NO	FK → MemberData.TB_Member(MemberId), UNIQUE	Miembro asociado (1:1, entrenador).
Specialty	VARCHAR(50)	NO	–	Especialidad del entrenador: Zumba & Yoga, General Fitness, Athletic. (CHECK).
StartDate	DATE	NO	–	Fecha de inicio como entrenador (DEFAULT GETDATE()).
EndDate	DATE	Sí	–	Fecha de finalización del perfil de entrenador.

Tabla 6: MemberData.TB_EmployeeProfile

Campo	Tipo	Nulo	Clave	Descripción
TrainerProfileId	INT IDENTITY(1,1)	NO	PK	Identificador del perfil de entrenador.
MemberId	INT	NO	FK → MemberData.TB_Member(MemberId), UNIQUE	Miembro asociado (1:1, entrenador).
Specialty	VARCHAR(50)	NO	–	Especialidad del entrenador: Zumba & Yoga, General Fitness, Athletic. (CHECK).
StartDate	DATE	NO	–	Fecha de inicio como entrenador (DEFAULT GETDATE()).
EndDate	DATE	Sí	–	Fecha de finalización del perfil de entrenador.

Tabla 7: GymCore.TB_Class

Campo	Tipo	Nulo	Clave	Descripción
ClassId	INT IDENTITY(1,1)	NO	PK	Identificador de la clase/área.
GymId	INT	NO	FK → GymCore.TB_Gym(GymId)	Gimnasio al que pertenece la clase.
ClassName	VARCHAR(100)	NO	–	Nombre de la clase o área de entrenamiento.

Tabla 8: GymCore.TB_Equipment

Campo	Tipo	Nulo	Clave	Descripción
EquipmentId	INT IDENTITY(1,1)	NO	PK	Identificador del equipo.
ClassId	INT	NO	FK → GymCore.TB_Class(ClassId)	Clase/área donde se encuentra el equipo.
EquipmentType	VARCHAR(100)	NO	–	Tipo de equipo (máquina, accesorio, etc.).
Description	VARCHAR(300)	Sí	–	Descripción del equipo.
LastMaintenance	DATETIME	Sí	–	Fecha de último mantenimiento realizado.

Tabla 9: OperationsData.TB_Enrollment

Campo	Tipo	Nulo	Clave	Descripción
EnrollmentId	INT IDENTITY(1,1)	NO	PK	Identificador de la inscripción.
MemberId	INT	NO	FK → MemberData.TB_Member(MemberId)	Miembro inscrito en la clase.
ClassId	INT	NO	FK → GymCore.TB_Class(ClassId)	Clase a la que se inscribe.
EnrollmentDate	DATE	NO	–	Fecha de inscripción (DEFAULT GETDATE()).
(UK_Enrollment)	(MemberId, ClassId)	–	UNIQUE	Un miembro no puede estar repetido en la misma clase.

Tabla 10: OperationsData.TB_Payment

Campo	Tipo	Nulo	Clave	Descripción
PaymentId	INT IDENTITY(1,1)	NO	PK	Identificador del pago.
MemberId	INT	NO	FK → MemberData.TB_Member(MemberId)	Miembro que realiza el pago.
Amount	DECIMAL(10,2)	NO	–	Monto del pago.
PaymentDescription	VARCHAR(150)	Sí	–	Descripción del pago (ej. Membership payment).
PaymentMethod	VARCHAR(20)	Sí	–	Método de pago (efectivo, tarjeta, etc.).
PaymentDate	DATETIME	NO	–	Fecha y hora del pago (DEFAULT GETDATE()).

Tabla 11: OperationsData.TB_PaymentArchive

Campo	Tipo	Nulo	Clave	Descripción
OriginalPaymentId	INT	Sí	–	Id del pago original archivado.
MemberId	INT	Sí	–	Miembro al que pertenecía el pago.
PaymentDate	DATETIME	Sí	–	Fecha del pago original.
Amount	DECIMAL(10,2)	Sí	–	Monto del pago archivado.
PaymentMethod	VARCHAR(20)	Sí	–	Método de pago original.
ArchiveReason	VARCHAR(200)	Sí	–	Motivo por el cual se archivó el pago.

Tabla 12: OperationsData.TB_Maintenance

Campo	Tipo	Nulo	Clave	Descripción
Maintenanceld	INT IDENTITY(1,1)	NO	PK	Identificador del registro de mantenimiento.
EquipmentId	INT	NO	FK → GymCore.TB_Equipment(EquipmentId)	Equipo que recibe mantenimiento.
MemberId	INT	NO	FK → MemberData.TB_Member(Memberld)	Miembro que reporta/realiza el mantenimiento.
MaintenanceType	VARCHAR(15)	NO	-	Tipo de mantenimiento (ej. correctivo, preventivo, etc., libre en el modelo).
MaintenanceSummary	VARCHAR(1000)	NO	-	Descripción detallada del mantenimiento.
NextMaintenanceDate	DATE	Sí	-	Fecha sugerida para el próximo mantenimiento.