

Project Description

Project goals:

The goal of the project is to create a database for a fitness club that will enable effective management of clients, memberships, classes, trainers, equipment, and feedback. The database is designed to support the daily operations of the club, such as client registration, membership management, class reservations, attendance tracking, and equipment maintenance. The database should also enable data analysis and report generation to support business decisions.

Main assumptions:

- **Database Structure** - The database consists of 14 tables that are interconnected through foreign key relationships.
- **Inheritance** - Inheritance is utilised through the person, client, trainer, and employee tables, where person is the parent table.
- **Dynamic data** - The database stores data that changes over time, such as membership status, class attendance, or equipment condition.
- **Automation** - Stored procedures and triggers are used to automate processes such as class reservation, reservation cancellation, or membership status updates.
- **Data security** - Additional data integrity constraints have been implemented, such as CHECK, UNIQUE, and NOT NULL constraints, to ensure data consistency.

Capabilities:

- **Client Management** - Registration of new clients, assigning memberships, tracking membership history.
- **Class Reservation** - Clients can reserve spots in classes, and the system automatically checks for availability.
- **Attendance Tracking** - The system records client attendance at classes, enabling attendance analysis.
- **Trainer Management** - Assigning trainers to classes, tracking their specialisations and availability.

- **Equipment Maintenance** - The system stores information about the technical condition of equipment and repair history.

- **Client Feedback** - Clients can provide feedback about trainers and classes, allowing for service quality assessment.

- **Financial Analysis** - Generating revenue reports from memberships and classes, which facilitates club financial management.

Limitations:

- **System Complexity** - The large number of interconnected tables requires careful query design and performance optimization.

- **Maintenance** - Regular monitoring of data integrity, index optimization, and backup creation are necessary.

- **Data Security** - Due to the storage of sensitive information (client data, payments), appropriate protection mechanisms are required, such as encryption and access control.

- **Scalability** - With a large number of clients, optimization of the database structure or migration to a more efficient system may be necessary.

Database Maintenance Strategy:

Backups:

- Full backup: daily at 2:00 AM.
- Differential backup: every 6 hours (8:00 AM, 2:00 PM, 8:00 PM).
- Transaction log backup: every 30 minutes.

Retention Policy:

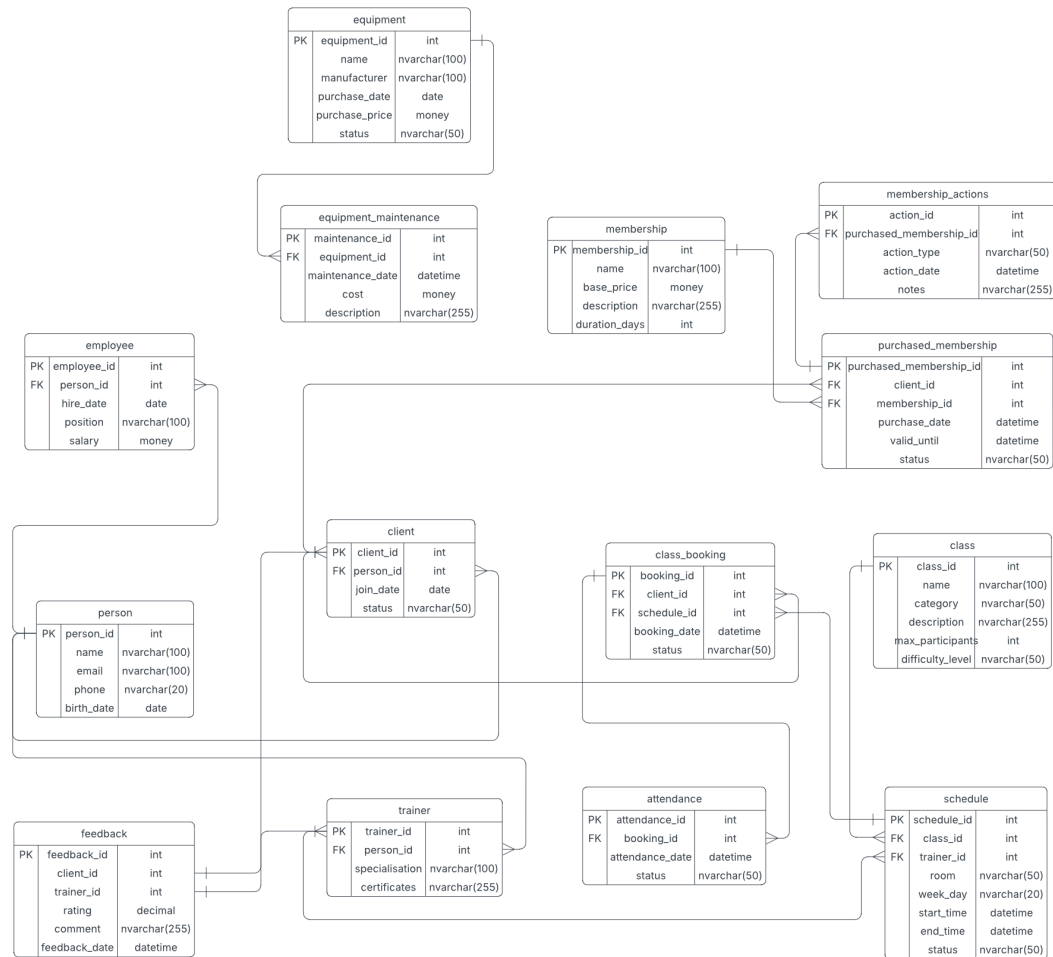
- Full backups: stored for 30 days.
- Differential backups: stored for 7 days.
- Transaction logs: stored for 3 days.

Recovery Tests: Regular tests of data recovery from backups.

Common Queries:

- Query for the average rating of each trainer along with the number of reviews received
- Query for the number of members by membership type
- Query for the most popular classes
- Query for the class schedule on a given day
- Query for class attendance

Relationship Diagram:



Additional Data Integrity Constraints:

CHECK:

- *purchased_membership.valid_until* must be later than *purchase_date*
- *feedback.rating* must be in the range from 1 to 5
- *equipment.status* must be one of: *active*, *requires_maintenance*, *inactive*
- *class.difficulty_level* must be one of: *Beginner*, *Intermediate*, *Advanced*
- *schedule.start_time* must be earlier than *end_time*
- *feedback.feedback_date* cannot be in the future
- *equipment.purchase_date* cannot be in the future
- *purchased_membership.status* must be one of: *active*, *inactive*, *expired*
- *class_booking.status* must be one of: *confirmed*, *cancelled*, *pending*
- *membership_actions.action_date* cannot be in the future
- *equipment_maintenance.maintenance_date* cannot be in the future

UNIQUE:

- *person.mail, person.phone must be unique*
- *The combination of schedule.room, schedule.week_day, schedule.start_time must be unique*
- *class.name must be unique*

ON DELETE CASCADE:

- *client(person_id) - deleting a person causes deletion of the client.*
- *trainer(person_id) - deleting a person causes deletion of the trainer.*
- *employee(person_id) - deleting a person causes deletion of the employee.*
- *purchased_membership(client_id) - deleting a client causes deletion of their membership.*
- *class_booking(client_id) - deleting a client causes deletion of their reservations.*
- *attendance(booking_id) - deleting a reservation causes deletion of attendance.*
- *equipment_maintenance(equipment_id) - deleting equipment causes deletion of its maintenance history.*
- *membership(purchased_membership_id) - deleting a membership causes deletion of its action history.*

NOT NULL:

- *class(name, category, max_participants, difficulty_level)*
- *membership(name, base_price, duration_days)*
- *person(name, email, phone, birth_date)*
- *client(person_id, join_date, status)*
- *trainer(person_id, specialisation)*
- *employee(person_id, hire_date, position, salary)*
- *purchased_membership(client_id, membership_id, purchase_date, valid_until, status)*
- *schedule(class_id, trainer_id, room, week_day, start_time, end_time, status)*
- *class_booking(client_id, schedule_id, booking_date, status)*
- *equipment(name, manufacturer, purchase_date, purchase_price, status)*
- *discount(name, start_date, end_date, discount)*
- *membership_actions(purchased_membership_id, action_type, action_date)*
- *attendance(booking_id, attendance_date, status)*
- *equipment_maintenance(equipment_id, maintenance_date, cost)*
- *feedback(client_id, trainer_id, rating, feedback_date)*

Indexes:

1. *IX_Trainers_Specialization: Accelerates the search for trainers by specialization.*
2. *IX_Schedule_ClubDayTime: Optimizes retrieval of class schedules by room, day of the week, and start time.*
3. *IX_Clients_JoinDate: Accelerates the search and sorting of clients by join date.*
4. *IX_Equipment_Status: Accelerates the search for equipment by status.*

Views:

1. *vw_active_clients*: Shows clients with active memberships.
2. *fun_get_schedule_by_day*: Returns the class schedule for a selected day of the week.
3. *vw_average_attendance_by_category*: Calculates the average attendance at classes broken down by category.
4. *vw_top_trainers_by_rating*: Shows trainers with the highest average ratings.
5. *vw_clients_with_expiring_memberships*: Displays clients whose memberships will expire within 7 days.
6. *vw_membership_revenue_by_month*: Shows membership revenue broken down by month.
7. *vw_equipment_needing_maintenance*: Displays equipment requiring repairs.
8. *vw_TrainerPerformance*: Shows trainer statistics, including the number of classes taught and average attendance.
9. *vw_top_attending_clients*: Displays clients with the highest number of visits.
10. *vw_longest_membership_clients*: Shows clients with the longest membership history.

Stored Procedures:

1. *sp_add_client*: Adds a new client and assigns a membership to them.
2. *sp_BookClass*: Reserves a spot in a class for a client.
3. *sp_CancelBooking*: Cancels a class reservation.
4. *sp_MarkAttendance*: Marks a client's attendance at a class.
5. *sp_LeaveFeedback*: Adds a client's feedback about a trainer.

Triggers:

1. *trg_check_class_capacity*: Checks if the number of reservations does not exceed the maximum number of participants.
2. *trg_cancel_bookings_after_membership_expiry*: Cancels reservations when a client's membership expires.

3. *trg_auto_set_membership_validity*: Automatically sets the membership validity date after purchase.

4. *trg_prevent_class_overlap*: Checks if new classes do not overlap with existing ones.

5. *trg_update_class_status_after_booking*: Updates the class status to "full" when the number of reservations reaches the maximum.