Universidad Distrital Francisco José de Caldas

Facultad de Ingeniería

Ingeniería de Sistemas

Fundamentos de Bases de Datos

Reporte proyecto

Yaxel Steven Morales Suarez-20212020060

Carlos Andres Sierra Virguez

Bogotá – Colombia

mayo 2024

STAKEHOLDERS:

Stakeholders:

End Users:Our key stakeholders are all those individuals who are committed to improving their health and wellness through regular exercise. These users are looking for a personalized and practical solution to help them achieve their fitness goals, providing them with the tools and support they need to stay motivated and focused on their path to a healthier lifestyle.

Developers and administrators of the "Fitia" app: This group is in charge of designing, developing, maintaining, and optimizing the "Fitia" platform. Their main objective is to ensure that the app works efficiently, securely, and attractively for users, constantly incorporating new features and improvements based on the needs and feedback from the community.

Support and customer service staff: Our support and customer service team is dedicated to providing an exceptional user experience, offering fast and effective assistance with any query, problem, or suggestion related to the app. Their mission is to ensure user satisfaction and trust, building strong and lasting relationships with our community.

Business Model: "Fitia" presents itself as a comprehensive platform designed to offer its users a unique and personalized fitness experience. The app combines exercise routines tailored to individual needs, real-time progress tracking, and valuable health and wellness tips. All of this is offered through a monthly subscription that provides unlimited access to all of the app's premium features, ensuring a complete and enriching experience for users.

ONTOLOGY:

The ontology of the gym database defines the entities and their relationships. Clients have attributes such as client_id, client_name, gender, age, height, weight, physical_condition, training_goal, and training_availability. Trainers have attributes including trainer_id, trainer_name, gender, and specialty. Activities are characterized by activity_id, activity_name, and duration. Trainings consist of training_id, client_id, trainer_id, activity_id, and date, with client_id referencing Clients, trainer_id referencing Trainers, and activity_id referencing Activities. Memberships include membership_id, membership_name, price, and duration. Payments are identified by payment_id, client_id, membership_id, payment_date, and amount, with client_id referencing Clients and membership_id referencing Memberships. Schedules comprise schedule_id, trainer_id, activity_id, day_of_week, start_time, and end_time, with trainer_id referencing Trainers and activity_id referencing Activities. Finally, Training Logs contain log_id, training_id, action, and action_date, with training_id referencing Trainings.

STEPS MODEL E-R:

1.Analyze Tables: Review each table in the SQL to understand the data it stores and the relationships between them.

2.Identify Entities: Identify the main entities in the system, such as Users (Clients), Trainers, Activities, Memberships, Payments, etc.

3.Identify Attributes: For each entity, identify the attributes that describe the characteristics of that entity, such as name, gender, age, etc.

4.Identify Relationships: Examine the foreign keys in the tables to identify the relationships between the entities.

5.Create Conceptual Diagram: Use an entity-relationship diagram to represent the entities, their attributes, and the relationships between them.

6.Add Entities: Add each entity to the diagram, representing them as rectangles.

7.Add Attributes: For each entity, add the attributes inside the rectangles representing them.

8.Add Relationships: Add the relationships between the entities to the diagram, using lines and labels to indicate the nature of the relationship.

9.Refine the Model: Review the diagram to ensure that all entities, attributes, and relationships are accurately represented.

10.Document the Model: Provide a detailed description of each entity, attribute, and relationship in the model, along with any relevant constraints or business rules.

HISTORY USERS:

1. As a client, I want to register my profile in the system so I can access gym services and schedule my workouts.

- Acceptance Criteria:

- The client should be able to enter their name, gender, age, height, weight, physical condition, training goal, and training availability.

- The system should assign a unique ID to the client.

- The client’s data should be correctly saved in the `clientes` table.

2. As a trainer, I want to register my profile in the system so I can manage my activities and workouts.

- Acceptance Criteria:

  - The trainer should be able to enter their name, gender, and specialty.

  - The system should assign a unique ID to the trainer.

  - The trainer's data should be correctly saved in the `entrenadores` table.

3. As an administrator, I want to add new activities to the system so clients can enroll and schedule workouts.

- Acceptance Criteria:

  - The administrator should be able to enter the activity name and duration.

  - The system should assign a unique ID to each activity.

  - The activity data should be correctly saved in the `actividades` table.

4. As a client, I want to schedule a workout with a specific trainer and activity to improve my physical condition.

- Acceptance Criteria:

  - The client should be able to select a date, trainer, and activity.

  - The workout should be registered in the `entrenamientos` table with the corresponding client, trainer, and activity IDs.

  - The system should validate that the date is not in the past.

5. As a client, I want to see a summary of my scheduled workouts to better plan my time and progress.

- Acceptance Criteria:

  - The client should be able to view a list of their scheduled workouts.

  - The view should display the client name, trainer, activity, and workout date.

  - The data should be retrieved from the `vista_resumen_entrenamientos` view.

6. As a client, I want to make a payment for my membership to maintain my access to the gym and its services.

- Acceptance Criteria:

- The client should be able to select a membership and make the corresponding payment.

- The payment should be registered in the `pagos` table with the client ID, membership ID, payment date, and amount.

- The system should validate that the amount matches the price of the selected membership.

7. As an administrator, I want to manage trainers' schedules to optimize activity availability and avoid conflicts.

- Acceptance Criteria:

- The administrator should be able to assign schedules to trainers for specific activities.

- The schedules should be registered in the `horarios` table with the trainer ID, activity ID, day of the week, start time, and end time.

- The system should validate that schedules do not overlap for the same trainer.

8. As a client, I want to view my payment history to keep track of my expenses and memberships.

- Acceptance Criteria:

- The client should be able to view a list of their payments.

- The view should display the payment ID, client name, membership name, payment date, and amount.

- The data should be retrieved from the `vista_historial_pagos` view.

9. As an administrator, I want to keep a log of all actions performed on workouts for auditing and service improvement.

- Acceptance Criteria:

- Whenever a new workout is inserted, a log entry should be recorded in the `log_entrenamientos` table with the workout ID, action performed, and action date.

- The `after_insert_entrenamiento` trigger should be set up to automatically record this information.

10. As an administrator, I want to generate reports of a client's payment history for financial analysis and queries.

- Acceptance Criteria:

- The administrator should be able to select a client and generate a report of their payment history.

- The report should include the payment ID, client name, membership name, payment date, and amount.

- The data should be retrieved using the `reporte_historial_pagos` stored procedure.

BUSINESS MODEL:

"Fitia Gym" presents itself as a comprehensive platform designed to offer its users a unique and personalized fitness experience. The app provides a variety of exercise routines tailored to individual needs, real-time progress tracking, and valuable health and wellness tips. All of this is offered through a monthly subscription that provides unlimited access to all of the app's premium features, ensuring a complete and enriching experience for users who want to stay fit and healthy.

CONCEPTUAL MODEL:

The conceptual model of the gym database comprises several interrelated entities. "Clients" represent registered users, with attributes such as name, gender, age, and training goals. "Trainers" have their own specialties and are associated with specific activities. "Activities" include details such as name and duration. "Trainings" record scheduled sessions between clients and trainers. "Memberships" describe available plans, while "Payments" reflect membership transactions made by clients. "Schedules" indicate trainers' availability for different activities on specific days and times of the day. Finally, the "Training Log" records actions taken on trainings, such as insertions or updates. This model provides a comprehensive structure of the gym database and the relationships between its elements.

DATA SOURCE:

A data source for the gym database could be various types of data repositories where the information is stored. Here's a simplified representation of how the data might be stored:

1. Clients Table:

- Columns: `con_id` (Client ID), `con_nombre` (Client Name), `con_genero` (Client Gender), `con_edad` (Client Age), `con_altura` (Client Height), `con_peso` (Client

Weight), `con_estadoFisico` (Client Physical Condition), `con_objetivoEntrenamiento` (Client Training Goal), `con_disponibilidadEntrenar` (Client Availability to Train).

2. Trainers Table:

- Columns: `ent_id` (Trainer ID), `ent_nombre` (Trainer Name), `ent_genero` (Trainer Gender), `ent_especialidad` (Trainer Specialization).

3. Activities Table:

- Columns: `act_id` (Activity ID), `act_nombre` (Activity Name), `act_duracion` (Activity Duration).

4. Trainings Table:

- Columns: `entrenamiento_id` (Training ID), `cliente_id` (Client ID), `entrenador_id` (Trainer ID), `actividad_id` (Activity ID), `fecha` (Date).

5. Memberships Table:

- Columns: `mem_id` (Membership ID), `mem_nombre` (Membership Name), `mem_precio` (Membership Price), `mem_duracion` (Membership Duration).

6. Payments Table:

- Columns: `pago_id` (Payment ID), `cliente_id` (Client ID), `mem_id` (Membership ID), `fecha_pago` (Payment Date), `monto` (Amount).

7. Schedules Table:

- Columns: `horario_id` (Schedule ID), `entrenador_id` (Trainer ID), `actividad_id` (Activity ID), `dia_semana` (Day of the Week), `hora_inicio` (Start Time), `hora_fin` (End Time).

8. Training Log Table:

- Columns: `log_id` (Log ID), `entrenamiento_id` (Training ID), `accion` (Action), `fecha_accion` (Action Date).

Each of these tables represents a different aspect of the gym's operations and is linked through foreign key relationships as defined in the schema. This data source structure allows for efficient storage and retrieval of information related to clients,

trainers, activities, memberships, payments, schedules, and training logs within the gym system.