**PRUEBAS IRON GYM DB**

Gimnasio que cuenta con 3 sedes y varias activiades por sede.

Hay posibilidad de ampliar tanto las sedes como las actividades.

El gimnasio permite la reserva de actividades tanto a los clientes que

estan registrados en su base de datos, como a no clientes (que tambien

se registran), pero a estos ultimos no se le ofrece una membresia.

Hoy hay 3 tipos de membresias (se puede ampliar), que permiten una

serie o cantidad de actividades a reservar y un descuento por tenerla.

El pago de la membresia es mensual.

Si un cliente con membresia no esta al dia con sus pagos, al momento de

hacer una reserva no se le da el descuento y debe abonar la actividad al 100% como un "no" cliente.

El acceso para hacerse cliente es a traves de la pagina principal, con un correo electronico (o un numero de telefono) + una password.

El cliente (una vez que accede al portal) puede ver sus reservas de acti-

vidades (las que hizo y las pendientes) y puede cacelar alguna (siempre

y cuando no haya pasado la fecha).

No hay proceso para ver como se le devuelve lo que abono si cancela una reserva.

Hay actividades con cupo para reservar (y otras que no). En las primeras

si se alcanzó el cupo ya no es posible hacer una reserva (salvo que se

cancele alguna por un cliente o por la propia administracion del gimnasio).

los datos solicitados a los clientes o a los no clientes no son los mismos.

El sistema permite además interactuar con la administracion del gimnasio a traves de un ws que está en un link de la pagina principal.

No se puede hacer una reserva de actividad (con o sin cupo) para el dia actual.

* **Duplicación de Información**:

La tabla *actividad\_sede* no tiene un campo que verifique si una actividad ya está asignada a una sede específica. Esto puede provocar duplicados no intencionados. Deberías implementar una restricción de unicidad en (id\_actividad, id\_sede).

ALTER TABLE actividad\_sede  
ADD CONSTRAINT unique\_actividad\_sede UNIQUE (id\_actividad, id\_sede);

* **Clarificar Relaciones en reservas:**

En reservas, se permite id\_no\_cliente y id\_cliente al mismo tiempo, lo cual puede ser confuso. Tal vez deberías hacer que uno de estos campos sea obligatorio y el otro opcional.  
  
ALTER TABLE reservas  
ADD CHECK (  
 (id\_cliente IS NOT NULL AND id\_no\_cliente IS NULL) OR   
 (id\_cliente IS NULL AND id\_no\_cliente IS NOT NULL)  
);

* **Cupos Negativos o Cero:**

En la tabla dia\_horario, el campo cupos no debe permitir valores negativos o cero. Esto debe ser validado al momento de insertar o actualizar datos.

* **Verificar Tipos de Documento en cliente:**

Agrega una clave foránea entre cliente.id\_tipo\_documento y tipo\_documento.id\_tipo\_documento

ALTER TABLE cliente  
ADD CONSTRAINT fk\_cliente\_tipo\_documento FOREIGN KEY (id\_tipo\_documento) REFERENCES tipo\_documento(id\_tipo\_documento);

* **Evitar Cupos Negativos en dia\_horario:**

ALTER TABLE dia\_horario  
MODIFY cupos INT UNSIGNED;