

Tarea 2.1

Alumna:

Aranzza Abascal Fararoni - A01329203

Profesor:

Dr. Daniel Pérez Rojas

Materia:

Bases de datos avanzadas

Fecha de entrega:

23 de Febrero de 2018

ITESM Campus Puebla

1. Escenario 1: Sistema para gimnasio

1.1. Stored procedure:



Imagen 1: Tabla para las sucursales, la cual tiene un id como llave primaria.

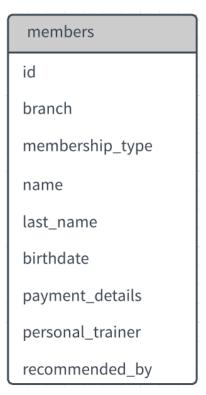


Imagen 2: Tabla para miembros del gimasio, cuenta con id como llave primaria, payment_details y personal_trainer como llaves foráneas.

payment_due_date

card_number

date_time

amount

Imagen 3: Tabla para los detalles del pago, card number es su llave primaria.

classes

id

name

spots_available

member_id

date_time

Imagen 4: Tabla para las clases, cuenta con id como llave primaria y member_id como llave foránea.

- Stored procedure que recibe como parámetro la fecha del día actual y devuelve la tabla de los id's de los miembros que su mensualidad vence en ese día.
- Stored procedure que recibe como parámetro el id de un entrenador personal y devuelve la tabla de los id's de los miembros que tienen algún entrenamiento con él/ella.
- Stored procedure que recibe como parámetro membership_type para aplicar descuentos dependiendo el tipo.
- Stored procedure para eliminar las clases que ya terminaron, dependiendo de su date time.

1.2. Triggers:

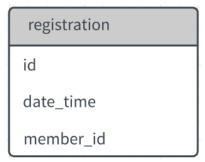


Imagen 5: Tabla para registrar la asistencia de los miembros.

- Trigger para regalarle 3 pases personales a los miembros que vayan al club, por lo menos,
 5 veces a la semana.
- Trigger para descontarle cierto porcentaje en la próxima mensualidad al id guardado en recommended_by en la tabla de members, cuando un nuevo miembro es agregado a la misma tabla.
- Trigger para aplicar descuento en mensualidad en el mes de cumpleaños del miembro.

1.3. Functions:

- Función para calcular la antigüedad de una sucursal.
- Función para calcular cuántas mensualidades se pagaron en el respectivo mes.
- Función para calcular el total de las mensualidades que se pagaron en el respectivo mes.

1.4. Transaction:

- Transacción para agregar el costo extra por entrenador personal a los miembros que cuentan con uno.
- Transacción para restarle al spots available cuando un miembro decida asistir a una clase.

1.5. Scheduled event:

- Scheduled event para hacer el cargo a card_number de los miembros que su mensualidad vence el día actual.
- Scheduled event para saber cuantas personas asistieron al club en el día.

2. Escenario 2: Servicio de reservaciones

2.1. Stored procedure:



Imagen 6: Tabla para las reservaciones, cuenta con id como llave primaria. "terrace. es un atributo de tipo boolean si el cliente desea estar en la parte de la terraza.

- Stored procedure que aplique alguna promoción si la reservación es para más de 4 personas.
- Stored procedure para desactivar las reservaciones después de 15 minutos de su date_time.

2.2. Triggers:

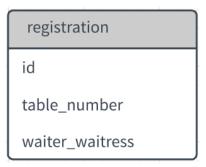


Imagen 7: Tabla para registrar las reservaciones que sí se realizaron. ID es llave foránea al id de reservation.

 Trigger para desactivar una reservación cuando su mismo id es insertado en la tabla de registration.

2.3. Functions:

- Función para calcular cuantas reservaciones atendió un mesero en especial.
- Función para calcular cuantas reservaciones se realizaron en la terraza.

2.4. Transaction:

 Transacción para hacer un insert en la tabla registration cuando se ocupe el mismo id de reservation.

2.5. Scheduled event:

• Evento para calcular cuantas personas fueron en el día.

3. Escenario 3: Administración de página de facebook

3.1. Stored procedure:

Suponiendo que existe una tabla para las publicaciones, la cual tiene un id como llave primaria, un date time, el número de personas alcanzadas, número de likes y un autor.

• Stored procedure para obtener el id de las publicaciones hechas por un autor en específico.

3.2. Triggers:

Suponiendo que existe una tabla cruce entre publicaciones y usuarios, llamada likes, la cual tiene un id como llave primaria y el id de ambas tablas como llaves foráneas.

 Trigger para aumentar el número de likes dentro de la tabla de publicaciones cuando hay una nueva inserción en la tabla de likes.

3.3. Functions:

• Función para calcular qué autor ha tenido más número de likes en sus publicaciones.

3.4. Transaction:

Suponiendo que el administrador de la página puede promocionar alguna publicación.

■ Transacción para hacer la promoción de la publicación, para poder obtener más likes.

3.5. Scheduled event:

Suponiendo que el administrador de la página puede programar cuando se publicará el contenido.

• Evento para publicar las publicaciones programadas.