Cody Kesselring

DBMS 2024

Madrid Metro Database

1. ER Diagram

A diagram of a company

Description automatically generated

2. Relational Schema

A screenshot of a computer screen

Description automatically generated

The Clients table contains all the information of the client. Importantly, it uses triggers to determine the age based on the DOB, it stores the zoneCode for pricing, and the family type in order to use discounts.

The ZonePrices table contains all the zones and their price; it also includes a ‘J’ for kids, ‘M’ for seniors’, ‘S’ for special, and ‘NA’ for families with no discounts.

The discounts table contains all of the possible discounts as well as the percent that the discount takes off of the price.

The monthly rechargeable card is split into 2 groups, one of them stores its data and activation status. The others use that data to track the amount that is due, and the other.

The multi entry metro card is split into 3 groups, one contains the initialization data, the other tracks the usage and deducts from the balance, and the last one is made for adding funds the card.

3. SQL

Register new client

A screenshot of a computer

Description automatically generated

Figure 2 insert new client automatically calculates age, zone, and discount. Triggers: CalculateAge, AssignZoneCode, AssignDiscountIDBasedOnFamilyType

Recharge a metro card

A screenshot of a computer

Description automatically generated

Figure 1 create a multi entry metro card with an initial balane of 12, then create an instance of recharging the card with 5. The figure updates the metro card table to 12+5, giving a balance of 17. Trigger: UpdateBalanceOnRecharge

Replace a missing card

A screenshot of a computer

Description automatically generated

Figure 2 create a usable monthly metro card that is active and paid for, then update to inactive when marked as lost. This sets off a trigger that creates a new card that is active and paid for under the same client, this also updates the expiration date using a trigger. Trigger: trg\_check\_lost\_card, SetExpirationDateOnCardCreation

Delete client

A screenshot of a computer

Description automatically generated

Figure 3 deleting the client from fig. 3, this removes their data from every table.

TRIGGERS

Before and after paying for an expired card

before:

A screenshot of a computer

Description automatically generated

Figure 4 insert an inactive and unpaid card for a client

after:

A screenshot of a computer

Description automatically generated

Figure 5 create instance of MonthlyRecharge using the same card number as fig.5, this calculates the amount that is owed and a trigger turns on the isActive attribute and paymentStatus is changed to ‘Paid’

Change price based off age

A screenshot of a computer

Description automatically generated

Figure 6 a trigger takes the date of birth for a client and calculates their age which assigns it correctly in the client table. Another trigger then changes their ZoneCode to Joven, which changes the price to 20 when a metro card recharge instance occurs.

Changing address updates zone code:

A screenshot of a computer

Description automatically generated

Figure 7 initialize a client within Madrid so their ZoneCode will be A

A screenshot of a computer

Description automatically generated

Figure 8 update their address to Alcobendas, which moves their ZoneCode to B1

Determine ZoneCode based off age and location.

A screenshot of a computer

Description automatically generated

Figure 9 this shows 3 clients, a 34, 18, and 104 year old who all live in Legumes. The ZoneCode is assigned based on their address if they are between 26 and 65 years old, otherwise, the ZoneCode is assigned based on their age. The 18 year old has ZoneCode J and pays 20 euros to recharge as seen on the bottom table. The 34 year old recharges his card 3 times for 63.7 because he has no discount and that is the base price of zone B1. Lastly we see the 104 year old with zone code ‘M’ pay 6.3 as he is in the senior category.

Display card status for a given ClientID

A screenshot of a computer

Description automatically generated

Figure 10 we have 2 clients, in the bottom table we can see that client C001 has an unpaid card which is not active and C002 has an active and paid for card.

MORE TRIGGERS

Assign price based off discount, age, and location

A screenshot of a computer

Description automatically generated

Figure 11 this table has 5 clients, 3 of them are adults that live in different areas and have differing family types, one of them is an adolescent and another is a senior. Using the family type and zone code, you can see in bottom table that each of them paid a different amount based on these factors.

Using discounts

A screenshot of a computer

Description automatically generated

Figure 12 here we have 5 clients all in the same zone but with different family types, there is a trigger that assigns their discount id based on this attribute. Each of them owns a monthly card, and as you can see at the bottom table, they all pay different prices due to their discounts.

Ensure Family Type is correctly input for discount

A screenshot of a computer

Description automatically generated

Figure 13 familyType 'Normal' was misspelled which flagged a trigger which tells the user how to correct it

Enforce Initial Multi Use Balance > 12A screen shot of a computer

Description automatically generated

Figure 14 the user inputs 11 euros as their initial balance and a trigger notifies them that 12 is the minimum. Trigger used: EnforceInitialMultiEntryCardBalance

Deduct Balance on multi use card usage

A screenshot of a computer

Description automatically generated

Figure 15 the figure shows a multi use metro card with an initial balance of 12 euros, we then insert a card use which deducts 1.7 euros from the balance. Trigger: DeductBalanceOnCardUsage

Update balance on recharge

A screenshot of a computer

Description automatically generated

Figure 16 starting in the ending state of the previous figure, we add a recharge instance with 5 euros, which increases the balance to 15.3 euros. Trigger used: UpdateBalanceOnRecharge

Card Inactive if Unpaid

A screenshot of a computer

Description automatically generated

Figure 17 this figure shows a monthly metro card being created as active but it is unpaid, so a trigger automatically sets the isActive status to 0

RELATIONAL ALGEBRA

A screenshot of a computer

Description automatically generated

Figure 18 this figure displays the name, email, zonecode, and price for all people living in laganes

A screen shot of a computer

Description automatically generated

Figure 19 this figure selects the isActive Boolean flag and paymentStatus of the card for clientId=C001