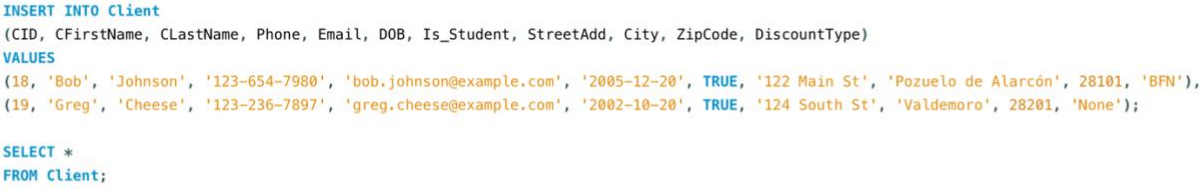
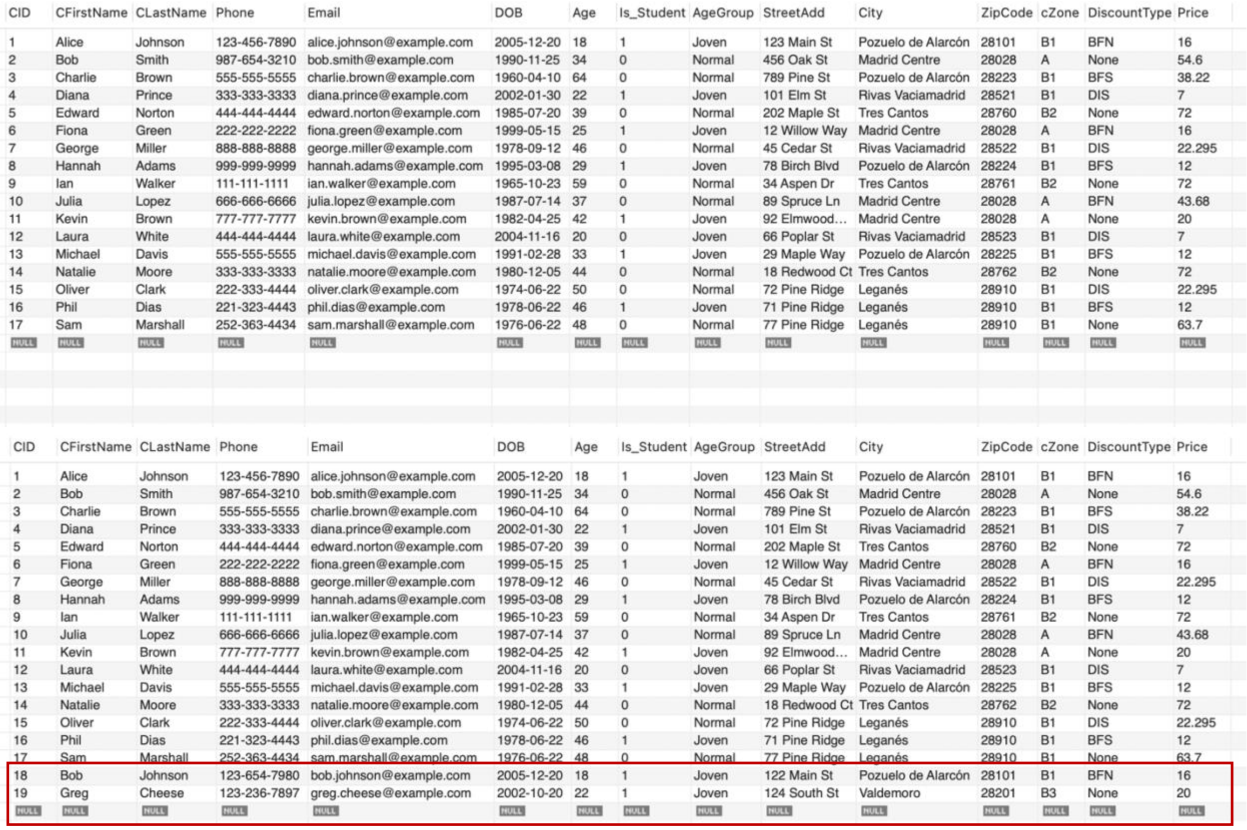
Basic Functionality Report:

Insertions, Triggers, and Queries

- Register a new client (insert data into the “Client” table).



```
INSERT INTO Client
(CID, CFirstName, CLastName, Phone, Email, DOB, Is_Student, StreetAdd, City, ZipCode, DiscountType)
VALUES
(18, 'Bob', 'Johnson', '123-654-7980', 'bob.johnson@example.com', '2005-12-20', TRUE, '122 Main St', 'Pozuelo de Alarcón', 28101, 'BFN'),
(19, 'Greg', 'Cheese', '123-236-7897', 'greg.cheese@example.com', '2002-10-20', TRUE, '124 South St', 'Valdemoro', 28201, 'None');

SELECT *
FROM Client;
```



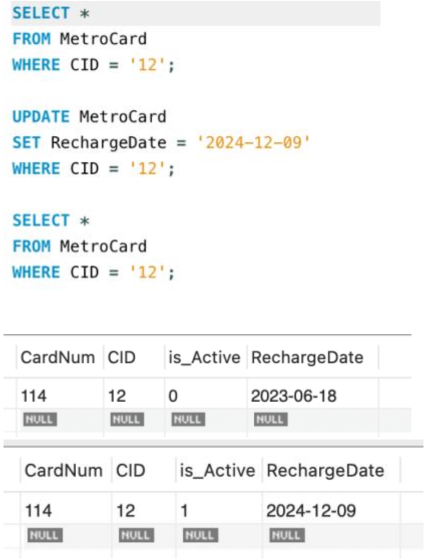
CID	CFirstName	CLastName	Phone	Email	DOB	Age	Is_Student	AgeGroup	StreetAdd	City	ZipCode	cZone	DiscountType	Price
1	Alice	Johnson	123-456-7890	alice.johnson@example.com	2005-12-20	18	1	Joven	123 Main St	Pozuelo de Alarcón	28101	B1	BFN	16
2	Bob	Smith	987-654-3210	bob.smith@example.com	1990-11-25	34	0	Normal	456 Oak St	Madrid Centre	28028	A	None	54.6
3	Charlie	Brown	555-555-5555	charlie.brown@example.com	1960-04-10	64	0	Normal	789 Pine St	Pozuelo de Alarcón	28223	B1	BFS	38.22
4	Diana	Prince	333-333-3333	diana.prince@example.com	2002-01-30	22	1	Joven	101 Elm St	Rivas Vaciamadrid	28521	B1	DIS	7
5	Edward	Norton	444-444-4444	edward.norton@example.com	1985-07-20	39	0	Normal	202 Maple St	Tres Cantos	28760	B2	None	72
6	Fiona	Green	222-222-2222	fiona.green@example.com	1999-05-15	25	1	Joven	12 Willow Way	Madrid Centre	28028	A	BFN	16
7	George	Miller	888-888-8888	george.miller@example.com	1978-09-12	46	0	Normal	45 Cedar St	Rivas Vaciamadrid	28522	B1	DIS	22.295
8	Hannah	Adams	999-999-9999	hannah.adams@example.com	1995-03-08	29	1	Joven	78 Birch Blvd	Pozuelo de Alarcón	28224	B1	BFS	12
9	Ian	Walker	111-111-1111	ian.walker@example.com	1965-10-23	59	0	Normal	34 Aspen Dr	Tres Cantos	28761	B2	None	72
10	Julia	Lopez	666-666-6666	julia.lopez@example.com	1987-07-14	37	0	Normal	89 Spruce Ln	Madrid Centre	28028	A	BFN	43.68
11	Kevin	Brown	777-777-7777	kevin.brown@example.com	1982-04-25	42	1	Joven	92 Elmwood…	Madrid Centre	28028	A	None	20
12	Laura	White	444-444-4444	laura.white@example.com	2004-11-16	20	0	Joven	66 Poplar St	Rivas Vaciamadrid	28523	B1	DIS	7
13	Michael	Davis	555-555-5555	michael.davis@example.com	1991-02-28	33	1	Joven	29 Maple Way	Pozuelo de Alarcón	28225	B1	BFS	12
14	Natalie	Moore	333-333-3333	natalie.moore@example.com	1980-12-05	44	0	Normal	18 Redwood Ct	Tres Cantos	28762	B2	None	72
15	Oliver	Clark	222-333-4444	oliver.clark@example.com	1974-06-22	50	0	Normal	72 Pine Ridge	Leganés	28910	B1	DIS	22.295
16	Phil	Dias	221-323-4443	phil.dias@example.com	1978-06-22	46	1	Joven	71 Pine Ridge	Leganés	28910	B1	BFS	12
17	Sam	Marshall	252-363-4434	sam.marshall@example.com	1976-06-22	48	0	Normal	77 Pine Ridge	Leganés	28910	B1	None	63.7
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

CID	CFirstName	CLastName	Phone	Email	DOB	Age	Is_Student	AgeGroup	StreetAdd	City	ZipCode	cZone	DiscountType	Price
1	Alice	Johnson	123-456-7890	alice.johnson@example.com	2005-12-20	18	1	Joven	123 Main St	Pozuelo de Alarcón	28101	B1	BFN	16
2	Bob	Smith	987-654-3210	bob.smith@example.com	1990-11-25	34	0	Normal	456 Oak St	Madrid Centre	28028	A	None	54.6
3	Charlie	Brown	555-555-5555	charlie.brown@example.com	1960-04-10	64	0	Normal	789 Pine St	Pozuelo de Alarcón	28223	B1	BFS	38.22
4	Diana	Prince	333-333-3333	diana.prince@example.com	2002-01-30	22	1	Joven	101 Elm St	Rivas Vaciamadrid	28521	B1	DIS	7
5	Edward	Norton	444-444-4444	edward.norton@example.com	1985-07-20	39	0	Normal	202 Maple St	Tres Cantos	28760	B2	None	72
6	Fiona	Green	222-222-2222	fiona.green@example.com	1999-05-15	25	1	Joven	12 Willow Way	Madrid Centre	28028	A	BFN	16
7	George	Miller	888-888-8888	george.miller@example.com	1978-09-12	46	0	Normal	45 Cedar St	Rivas Vaciamadrid	28522	B1	DIS	22.295
8	Hannah	Adams	999-999-9999	hannah.adams@example.com	1995-03-08	29	1	Joven	78 Birch Blvd	Pozuelo de Alarcón	28224	B1	BFS	12
9	Ian	Walker	111-111-1111	ian.walker@example.com	1965-10-23	59	0	Normal	34 Aspen Dr	Tres Cantos	28761	B2	None	72
10	Julia	Lopez	666-666-6666	julia.lopez@example.com	1987-07-14	37	0	Normal	89 Spruce Ln	Madrid Centre	28028	A	BFN	43.68
11	Kevin	Brown	777-777-7777	kevin.brown@example.com	1982-04-25	42	1	Joven	92 Elmwood…	Madrid Centre	28028	A	None	20
12	Laura	White	444-444-4444	laura.white@example.com	2004-11-16	20	0	Joven	66 Poplar St	Rivas Vaciamadrid	28523	B1	DIS	7
13	Michael	Davis	555-555-5555	michael.davis@example.com	1991-02-28	33	1	Joven	29 Maple Way	Pozuelo de Alarcón	28225	B1	BFS	12
14	Natalie	Moore	333-333-3333	natalie.moore@example.com	1980-12-05	44	0	Normal	18 Redwood Ct	Tres Cantos	28762	B2	None	72
15	Oliver	Clark	222-333-4444	oliver.clark@example.com	1974-06-22	50	0	Normal	72 Pine Ridge	Leganés	28910	B1	DIS	22.295
16	Phil	Dias	221-323-4443	phil.dias@example.com	1978-06-22	46	1	Joven	71 Pine Ridge	Leganés	28910	B1	BFS	12
17	Sam	Marshall	252-363-4434	sam.marshall@example.com	1976-06-22	48	0	Normal	77 Pine Ridge	Leganés	28910	B1	None	63.7
18	Bob	Johnson	123-654-7980	bob.johnson@example.com	2005-12-20	18	1	Joven	122 Main St	Pozuelo de Alarcón	28101	B1	BFN	16
19	Greg	Cheese	123-236-7897	greg.cheese@example.com	2002-10-20	22	1	Joven	124 South St	Valdemoro	28201	B3	None	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 3 Before and after client insertion

- Recharge a metro card (update the balance/status in the “Metro Card” table). (need to add multi entrance card before doing this), also need to change foreign keys to cascade)

For rechargeable metro card – we can “recharge the card” so the rechargedate is changed, this activating the card.



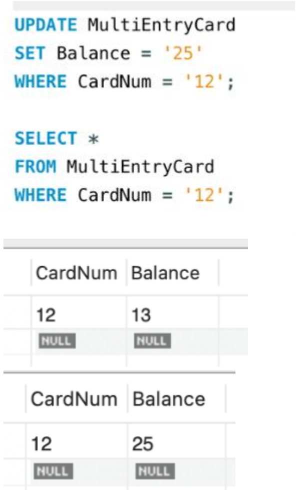
```
SELECT *
FROM MetroCard
WHERE CID = '12';

UPDATE MetroCard
SET RechargeDate = '2024-12-09'
WHERE CID = '12';

SELECT *
FROM MetroCard
WHERE CID = '12';
```

CardNum	CID	is_Active	RechargeDate
114	12	0	2023-06-18
NULL	NULL	NULL	NULL

CardNum	CID	is_Active	RechargeDate
114	12	1	2024-12-09
NULL	NULL	NULL	NULL

Figure 4 Recharging Metro Card

For multi entry card, we can increase the balance (add money), which makes the card useable



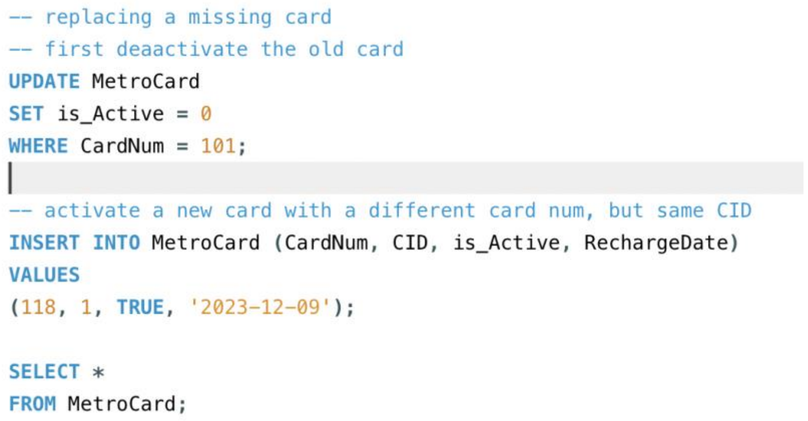
```
UPDATE MultiEntryCard
SET Balance = '25'
WHERE CardNum = '12';

SELECT *
FROM MultiEntryCard
WHERE CardNum = '12';
```

CardNum	Balance
12	13
NULL	NULL

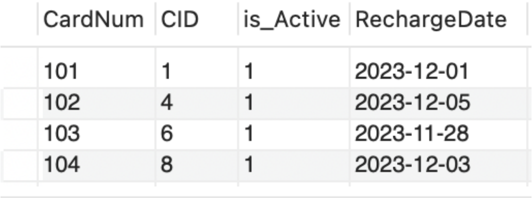
CardNum	Balance
12	25
NULL	NULL

Figure 5 Adding money to multi entry card

- Replace a missing card (update card status in the “Metro Card” table).

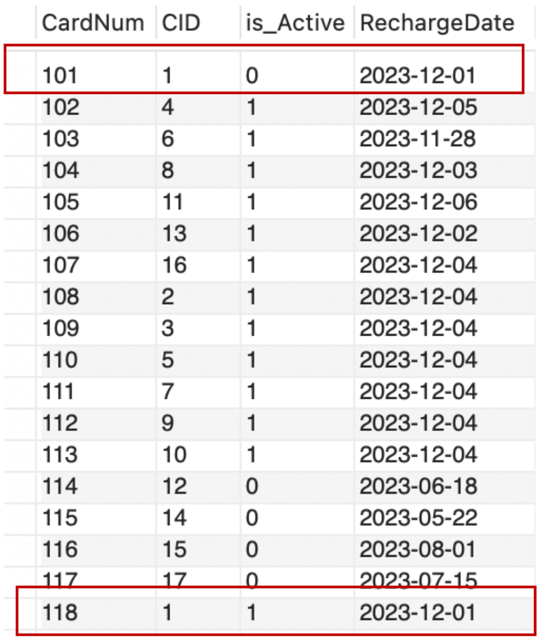


```
-- replacing a missing card
-- first deactivate the old card
UPDATE MetroCard
SET is_Active = 0
WHERE CardNum = 101;

-- activate a new card with a different card num, but same CID
INSERT INTO MetroCard (CardNum, CID, is_Active, RechargeDate)
VALUES
(118, 1, TRUE, '2023-12-09');

SELECT *
FROM MetroCard;
```



CardNum	CID	is_Active	RechargeDate
101	1	1	2023-12-01
102	4	1	2023-12-05
103	6	1	2023-11-28
104	8	1	2023-12-03

(original table)



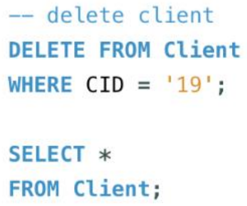
CardNum	CID	is_Active	RechargeDate
101	1	0	2023-12-01
102	4	1	2023-12-05
103	6	1	2023-11-28
104	8	1	2023-12-03
105	11	1	2023-12-06
106	13	1	2023-12-02
107	16	1	2023-12-04
108	2	1	2023-12-04
109	3	1	2023-12-04
110	5	1	2023-12-04
111	7	1	2023-12-04
112	9	1	2023-12-04
113	10	1	2023-12-04
114	12	0	2023-06-18
115	14	0	2023-05-22
116	15	0	2023-08-01
117	17	0	2023-07-15
118	1	1	2023-12-01

(after operations)

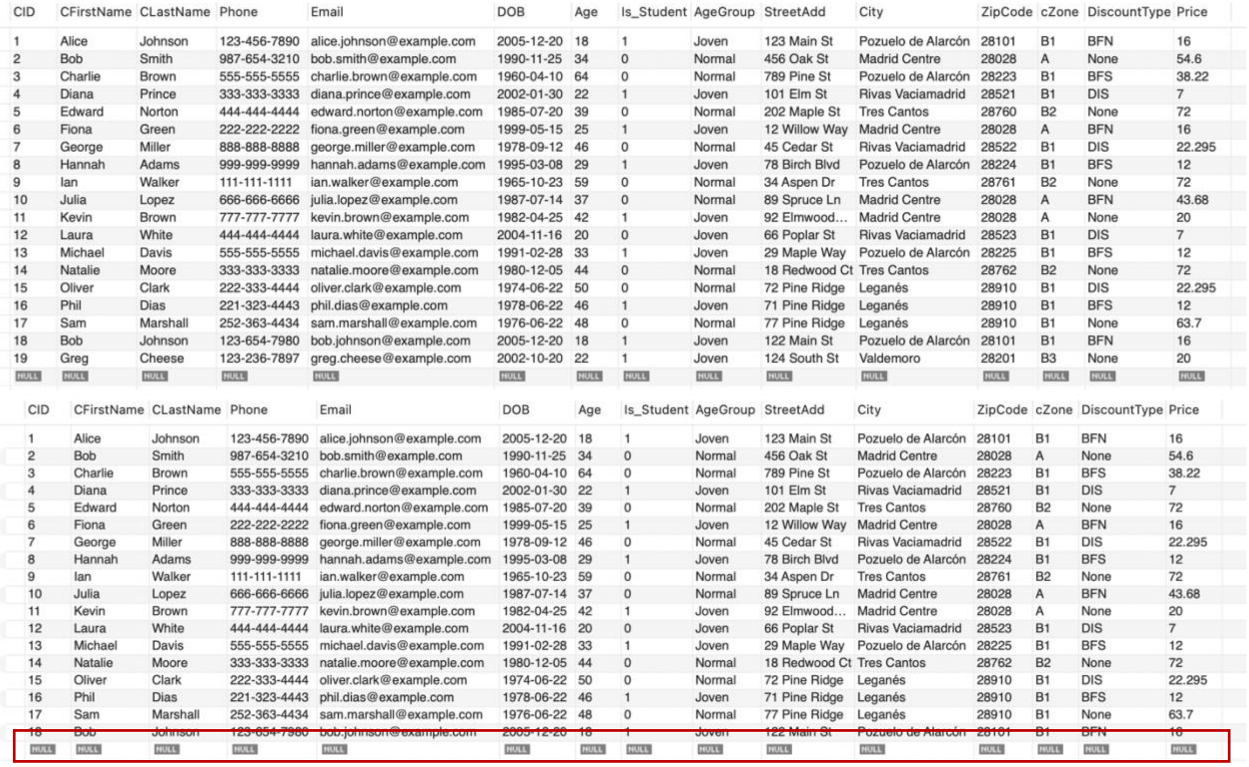
Figure 6 Replacing Missing Card

As we can see, we cancel the missing metro card (set in inactive), and when we insert a new card with the same CID, it takes the old recharge date (meaning they don’t have to reset the whole card)

- Cancel client registration (delete client data from the “Client” table).



```
-- delete client
DELETE FROM Client
WHERE CID = '19';

SELECT *
FROM Client;
```

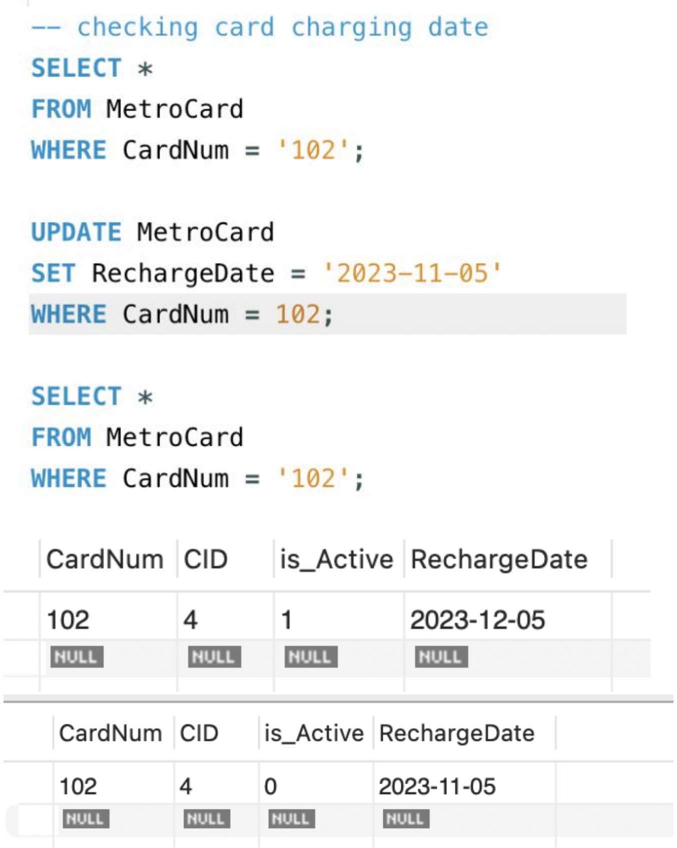


CID	CFirstName	CLastName	Phone	Email	DOB	Age	Is_Student	AgeGroup	StreetAdd	City	ZipCode	cZone	DiscountType	Price
1	Alice	Johnson	123-456-7890	alice.johnson@example.com	2005-12-20	18	1	Joven	123 Main St	Pozuelo de Alarcón	28101	B1	BFN	16
2	Bob	Smith	987-654-3210	bob.smith@example.com	1990-11-25	34	0	Normal	456 Oak St	Madrid Centre	28028	A	None	54.6
3	Charlie	Brown	555-555-5555	charlie.brown@example.com	1960-04-10	64	0	Normal	789 Pine St	Pozuelo de Alarcón	28223	B1	BFS	38.22
4	Diana	Prince	333-333-3333	diana.prince@example.com	2002-01-30	22	1	Joven	101 Elm St	Rivas Vaciamadrid	28521	B1	DIS	7
5	Edward	Norton	444-444-4444	edward.norton@example.com	1985-07-20	39	0	Normal	202 Maple St	Tres Cantos	28760	B2	None	72
6	Fiona	Green	222-222-2222	fiona.green@example.com	1999-05-15	25	1	Joven	12 Willow Way	Madrid Centre	28028	A	BFN	16
7	George	Miller	888-888-8888	george.miller@example.com	1978-09-12	46	0	Normal	45 Cedar St	Rivas Vaciamadrid	28522	B1	DIS	22.295
8	Hannah	Adams	999-999-9999	hannah.adams@example.com	1995-03-08	29	1	Joven	78 Birch Blvd	Pozuelo de Alarcón	28224	B1	BFS	12
9	Ian	Walker	111-111-1111	ian.walker@example.com	1965-10-23	59	0	Normal	34 Aspen Dr	Tres Cantos	28761	B2	None	72
10	Julia	Lopez	666-666-6666	julia.lopez@example.com	1987-07-14	37	0	Normal	89 Spruce Ln	Madrid Centre	28028	A	BFN	43.68
11	Kevin	Brown	777-777-7777	kevin.brown@example.com	1982-04-25	42	1	Joven	92 Elmwood…	Madrid Centre	28028	A	None	20
12	Laura	White	444-444-4444	laura.white@example.com	2004-11-16	20	0	Joven	66 Poplar St	Rivas Vaciamadrid	28523	B1	DIS	7
13	Michael	Davis	555-555-5555	michael.davis@example.com	1991-02-28	33	1	Joven	29 Maple Way	Pozuelo de Alarcón	28225	B1	BFS	12
14	Natalie	Moore	333-333-3333	natalie.moore@example.com	1980-12-05	44	0	Normal	18 Redwood Ct	Tres Cantos	28762	B2	None	72
15	Oliver	Clark	222-333-4444	oliver.clark@example.com	1974-06-22	50	0	Normal	72 Pine Ridge	Leganés	28910	B1	DIS	22.295
16	Phil	Dias	221-323-4443	phil.dias@example.com	1978-06-22	46	1	Joven	71 Pine Ridge	Leganés	28910	B1	BFS	12
17	Sam	Marshall	252-363-4434	sam.marshall@example.com	1976-06-22	48	0	Normal	77 Pine Ridge	Leganés	28910	B1	None	63.7
18	Bob	Johnson	123-654-7980	bob.johnson@example.com	2005-12-20	18	1	Joven	122 Main St	Pozuelo de Alarcón	28101	B1	BFN	16
19	Greg	Cheese	123-236-7897	greg.cheese@example.com	2002-10-20	22	1	Joven	124 South St	Valdemoro	28201	B3	None	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

CID	CFirstName	CLastName	Phone	Email	DOB	Age	Is_Student	AgeGroup	StreetAdd	City	ZipCode	cZone	DiscountType	Price
1	Alice	Johnson	123-456-7890	alice.johnson@example.com	2005-12-20	18	1	Joven	123 Main St	Pozuelo de Alarcón	28101	B1	BFN	16
2	Bob	Smith	987-654-3210	bob.smith@example.com	1990-11-25	34	0	Normal	456 Oak St	Madrid Centre	28028	A	None	54.6
3	Charlie	Brown	555-555-5555	charlie.brown@example.com	1960-04-10	64	0	Normal	789 Pine St	Pozuelo de Alarcón	28223	B1	BFS	38.22
4	Diana	Prince	333-333-3333	diana.prince@example.com	2002-01-30	22	1	Joven	101 Elm St	Rivas Vaciamadrid	28521	B1	DIS	7
5	Edward	Norton	444-444-4444	edward.norton@example.com	1985-07-20	39	0	Normal	202 Maple St	Tres Cantos	28760	B2	None	72
6	Fiona	Green	222-222-2222	fiona.green@example.com	1999-05-15	25	1	Joven	12 Willow Way	Madrid Centre	28028	A	BFN	16
7	George	Miller	888-888-8888	george.miller@example.com	1978-09-12	46	0	Normal	45 Cedar St	Rivas Vaciamadrid	28522	B1	DIS	22.295
8	Hannah	Adams	999-999-9999	hannah.adams@example.com	1995-03-08	29	1	Joven	78 Birch Blvd	Pozuelo de Alarcón	28224	B1	BFS	12
9	Ian	Walker	111-111-1111	ian.walker@example.com	1965-10-23	59	0	Normal	34 Aspen Dr	Tres Cantos	28761	B2	None	72
10	Julia	Lopez	666-666-6666	julia.lopez@example.com	1987-07-14	37	0	Normal	89 Spruce Ln	Madrid Centre	28028	A	BFN	43.68
11	Kevin	Brown	777-777-7777	kevin.brown@example.com	1982-04-25	42	1	Joven	92 Elmwood…	Madrid Centre	28028	A	None	20
12	Laura	White	444-444-4444	laura.white@example.com	2004-11-16	20	0	Joven	66 Poplar St	Rivas Vaciamadrid	28523	B1	DIS	7
13	Michael	Davis	555-555-5555	michael.davis@example.com	1991-02-28	33	1	Joven	29 Maple Way	Pozuelo de Alarcón	28225	B1	BFS	12
14	Natalie	Moore	333-333-3333	natalie.moore@example.com	1980-12-05	44	0	Normal	18 Redwood Ct	Tres Cantos	28762	B2	None	72
15	Oliver	Clark	222-333-4444	oliver.clark@example.com	1974-06-22	50	0	Normal	72 Pine Ridge	Leganés	28910	B1	DIS	22.295
16	Phil	Dias	221-323-4443	phil.dias@example.com	1978-06-22	46	1	Joven	71 Pine Ridge	Leganés	28910	B1	BFS	12
17	Sam	Marshall	252-363-4434	sam.marshall@example.com	1976-06-22	48	0	Normal	77 Pine Ridge	Leganés	28910	B1	None	63.7
~~18~~	~~Bob~~	~~Johnson~~	~~123-654-7980~~	~~bob.johnson@example.com~~	~~2005-12-20~~	~~18~~	~~1~~	~~Joven~~	~~122 Main St~~	~~Pozuelo de Alarcón~~	~~28101~~	~~B1~~	~~BFN~~	~~16~~
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7 Deleting a Client

- Trigger: Check card charging date and change card state to “0” if it has been 30 days since the last charge.



```
-- checking card charging date
SELECT *
FROM MetroCard
WHERE CardNum = '102';

UPDATE MetroCard
SET RechargeDate = '2023-11-05'
WHERE CardNum = 102;

SELECT *
FROM MetroCard
WHERE CardNum = '102';
```

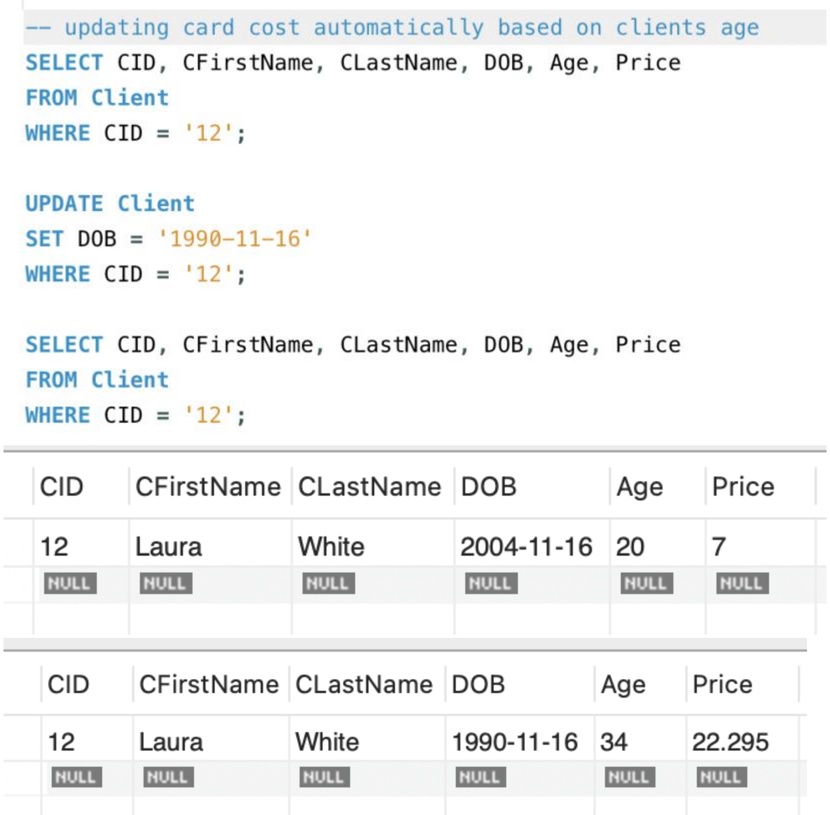
CardNum	CID	is_Active	RechargeDate
102	4	1	2023-12-05
NULL	NULL	NULL	NULL

CardNum	CID	is_Active	RechargeDate
102	4	0	2023-11-05
NULL	NULL	NULL	NULL

Figure 8 Trigger Checking Charging Date

As we can see, when the RechargeDate becomes past 30 days ago, the card becomes inactive.

- Trigger: Update the card cost automatically based on the client's age.



```
-- updating card cost automatically based on clients age
SELECT CID, CFirstName, CLastName, DOB, Age, Price
FROM Client
WHERE CID = '12';

UPDATE Client
SET DOB = '1990-11-16'
WHERE CID = '12';

SELECT CID, CFirstName, CLastName, DOB, Age, Price
FROM Client
WHERE CID = '12';
```

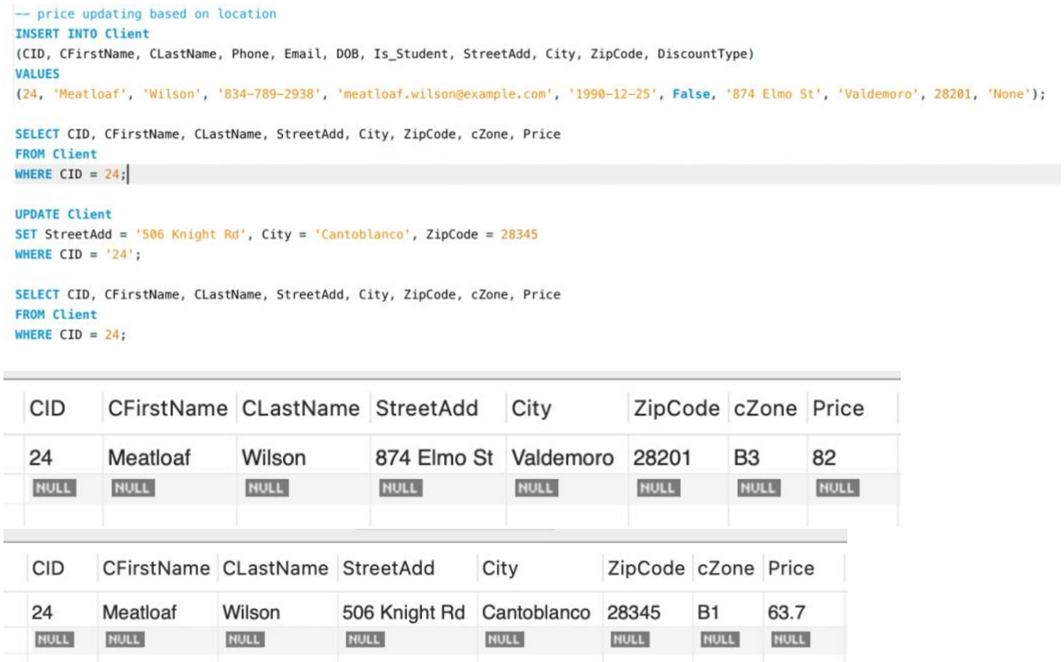
CID	CFirstName	CLastName	DOB	Age	Price
12	Laura	White	2004-11-16	20	7
NULL	NULL	NULL	NULL	NULL	NULL

CID	CFirstName	CLastName	DOB	Age	Price
12	Laura	White	1990-11-16	34	22.295
NULL	NULL	NULL	NULL	NULL	NULL

Figure 9 Trigger Price Change When Age Changes

*NOTE that this client has a discount, but we can still see the clear functionality as the client goes from joven to normal, and their price increases automatically

- Trigger: Determine the card fee according to the city (zone) when the client registers or updates their address.



```sql
-- price updating based on location
INSERT INTO Client
(CID, CFirstName, CLastName, Phone, Email, DOB, Is_Student, StreetAdd, City, ZipCode, DiscountType)
VALUES
(24, 'Meatloaf', 'Wilson', '834-789-2938', 'meatloaf.wilson@example.com', '1990-12-25', False, '874 Elmo St', 'Valdemoro', 28201, 'None');

SELECT CID, CFirstName, CLastName, StreetAdd, City, ZipCode, cZone, Price
FROM Client
WHERE CID = 24;

UPDATE Client
SET StreetAdd = '506 Knight Rd', City = 'Cantoblanco', ZipCode = 28345
WHERE CID = '24';

SELECT CID, CFirstName, CLastName, StreetAdd, City, ZipCode, cZone, Price
FROM Client
WHERE CID = 24;
```

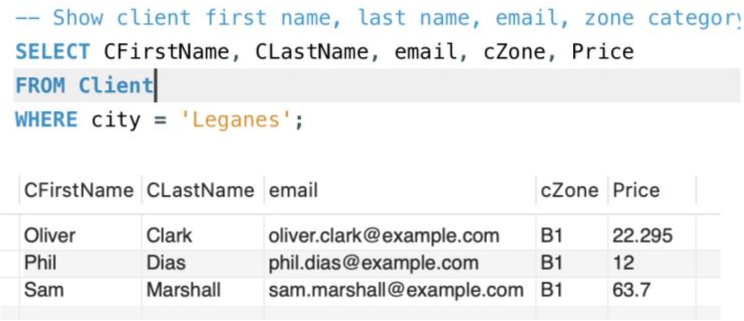
CID	CFirstName	CLastName	StreetAdd	City	ZipCode	cZone	Price
24	Meatloaf	Wilson	874 Elmo St	Valdemoro	28201	B3	82
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

CID	CFirstName	CLastName	StreetAdd	City	ZipCode	cZone	Price
24	Meatloaf	Wilson	506 Knight Rd	Cantoblanco	28345	B1	63.7
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 10 Price Changing when Address is Inserted and Updated

We can see how the price is automatically calculated based on the client’s inserted address, and how it changes as we go and update their address.

- Query: Show client first name, last name, email, zone category, and card monthly price for all clients living in ‘Leganes’.



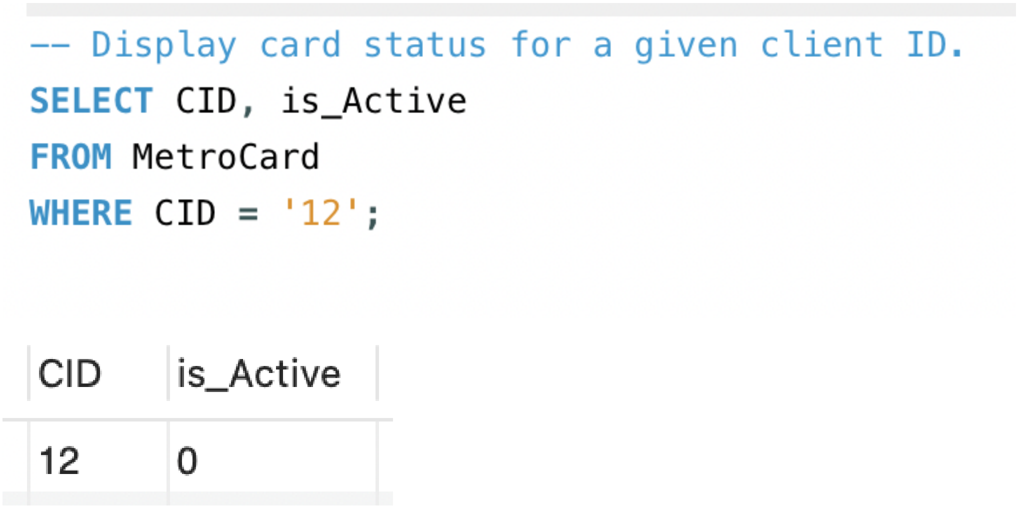
```
-- Show client first name, last name, email, zone categor
SELECT CFirstName, CLastName, email, cZone, Price
FROM Client
WHERE city = 'Leganes';
```

CFirstName	CLastName	email	cZone	Price
Oliver	Clark	oliver.clark@example.com	B1	22.295
Phil	Dias	phil.dias@example.com	B1	12
Sam	Marshall	sam.marshall@example.com	B1	63.7

Figure 11 Query for all clients from Leganes

- Query: Display card status for a given client ID.

Ex. Using CID = 12



```
-- Display card status for a given client ID.
SELECT CID, is_Active
FROM MetroCard
WHERE CID = '12';
```

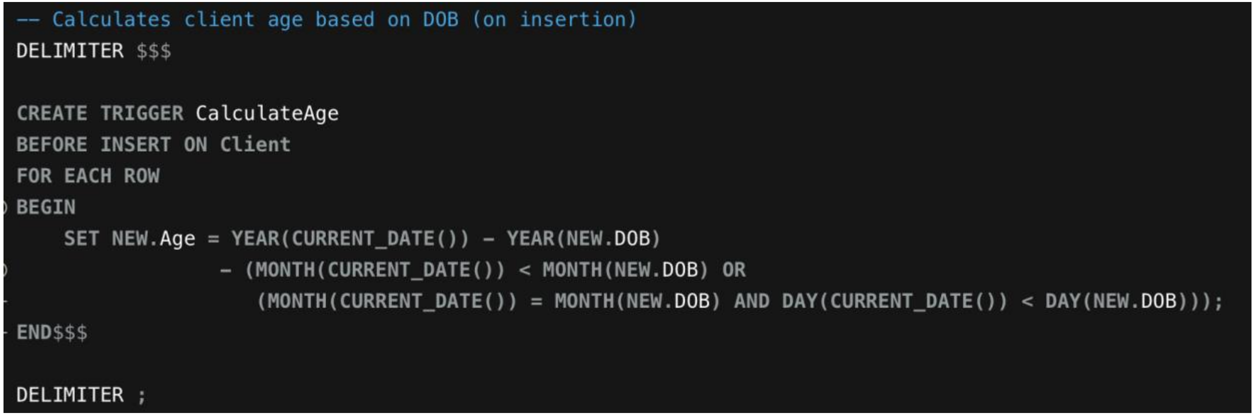
CID	is_Active
12	0

Figure 12 Finding card status for given client ID

All Triggers and Their Uses:

For Client:

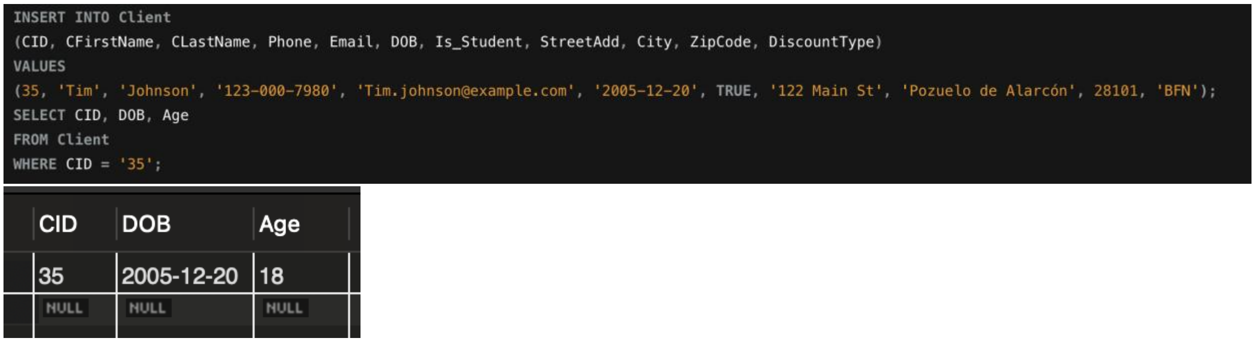
1) **CalculateAge**



```
-- Calculates client age based on DOB (on insertion)
DELIMITER $$$

CREATE TRIGGER CalculateAge
BEFORE INSERT ON Client
FOR EACH ROW
BEGIN
    SET NEW.Age = YEAR(CURRENT_DATE()) - YEAR(NEW.DOB)
                 - (MONTH(CURRENT_DATE()) < MONTH(NEW.DOB) OR
                    (MONTH(CURRENT_DATE()) = MONTH(NEW.DOB) AND DAY(CURRENT_DATE()) < DAY(NEW.DOB)));
END$$$

DELIMITER ;
```

Figure 13 CalculateAge Trigger Code

CalculateAge simply fills the attribute age based on the client's inserted Date of Birth.

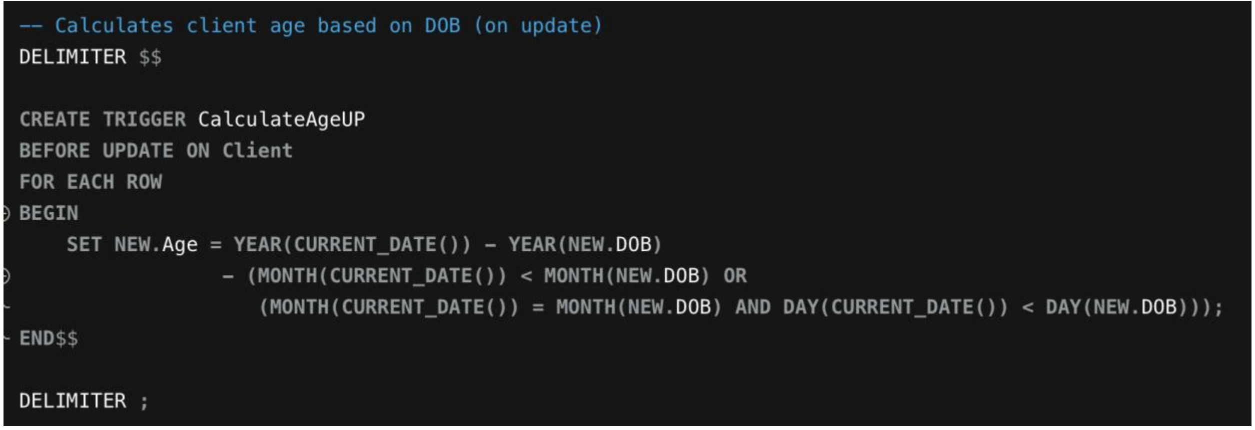


```
INSERT INTO Client
(CID, CFirstName, CLastName, Phone, Email, DOB, Is_Student, StreetAdd, City, ZipCode, DiscountType)
VALUES
(35, 'Tim', 'Johnson', '123-000-7980', 'Tim.johnson@example.com', '2005-12-20', TRUE, '122 Main St', 'Pozuelo de Alarcón', 28101, 'BFN');
SELECT CID, DOB, Age
FROM Client
WHERE CID = '35';
```

CID	DOB	Age
35	2005-12-20	18
NULL	NULL	NULL

Figure 14 Example of CalculateAge Trigger Functioning

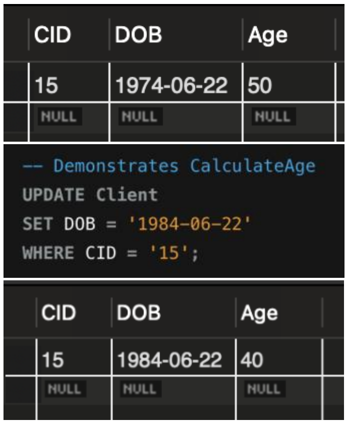
2) CalculateAgeUP



```
-- Calculates client age based on DOB (on update)
DELIMITER $$

CREATE TRIGGER CalculateAgeUP
BEFORE UPDATE ON Client
FOR EACH ROW
BEGIN
    SET NEW.Age = YEAR(CURRENT_DATE()) - YEAR(NEW.DOB)
                 - (MONTH(CURRENT_DATE()) < MONTH(NEW.DOB) OR
                    (MONTH(CURRENT_DATE()) = MONTH(NEW.DOB) AND DAY(CURRENT_DATE()) < DAY(NEW.DOB)));
END$$

DELIMITER ;
```

Figure 15 CalculateAgeUP Trigger Code

CalculateAgeUP simply changes the client's age if we change their Date of Birth.



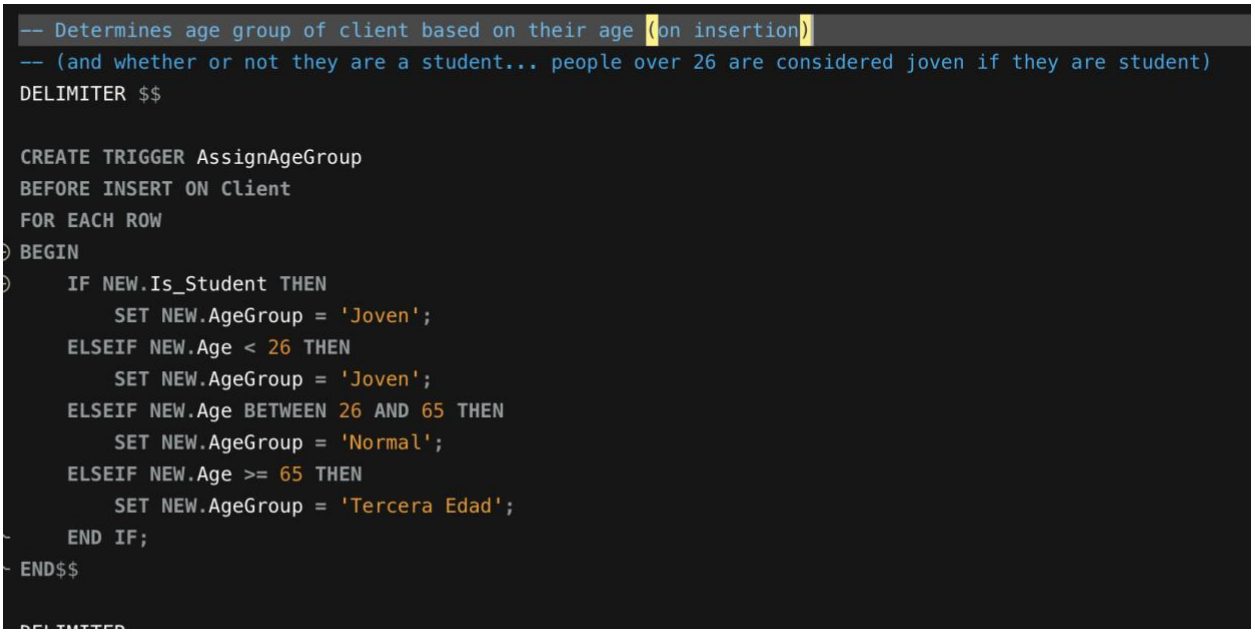
CID	DOB	Age
15	1974-06-22	50
NULL	NULL	NULL

```
-- Demonstrates CalculateAge
UPDATE Client
SET DOB = '1984-06-22'
WHERE CID = '15';
```

CID	DOB	Age
15	1984-06-22	40
NULL	NULL	NULL

Figure 16 Example of CalculateAgeUP Trigger Functioning

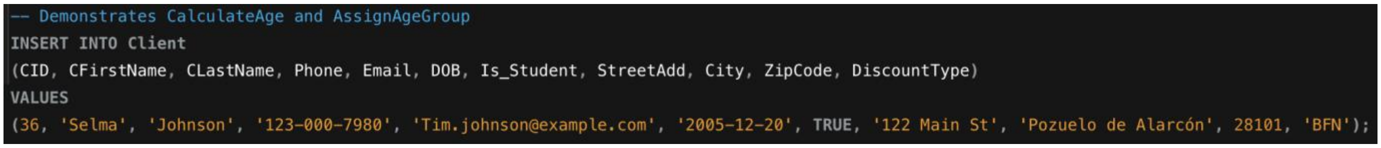
3) AssignAgeGroup



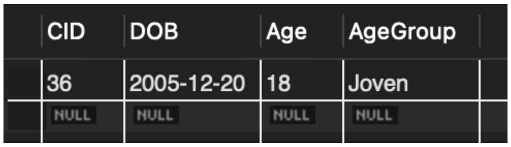
```
-- Determines age group of client based on their age (on insertion)
-- (and whether or not they are a student... people over 26 are considered joven if they are student)
DELIMITER $$

CREATE TRIGGER AssignAgeGroup
BEFORE INSERT ON Client
FOR EACH ROW
BEGIN
    IF NEW.Is_Student THEN
        SET NEW.AgeGroup = 'Joven';
    ELSEIF NEW.Age < 26 THEN
        SET NEW.AgeGroup = 'Joven';
    ELSEIF NEW.Age BETWEEN 26 AND 65 THEN
        SET NEW.AgeGroup = 'Normal';
    ELSEIF NEW.Age >= 65 THEN
        SET NEW.AgeGroup = 'Tercera Edad';
    END IF;
END$$

DELIMITER
```

Figure 17 AssignAgeGroup Trigger Code

Fills a client's AgeGroup attribute based on their age at insertion.



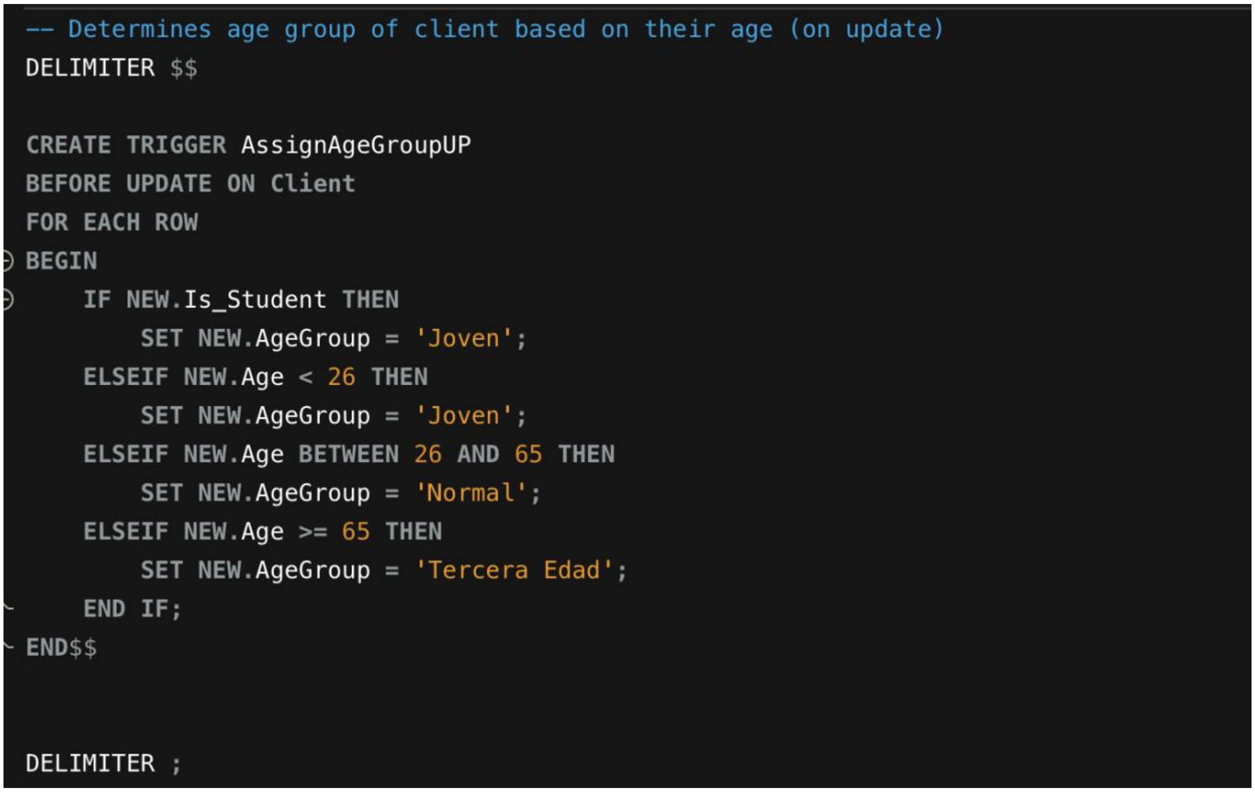
```
-- Demonstrates CalculateAge and AssignAgeGroup
INSERT INTO Client
(CID, CFirstName, CLastName, Phone, Email, DOB, Is_Student, StreetAdd, City, ZipCode, DiscountType)
VALUES
(36, 'Selma', 'Johnson', '123-000-7980', 'Tim.johnson@example.com', '2005-12-20', TRUE, '122 Main St', 'Pozuelo de Alarcón', 28101, 'BFN');
```



CID	DOB	Age	AgeGroup
36	2005-12-20	18	Joven
NULL	NULL	NULL	NULL

Figure 18 Example of AssignAgeGroup Trigger Functioning

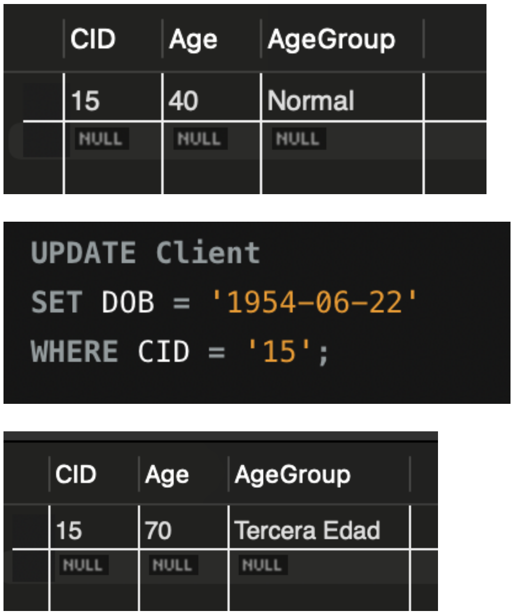
4) AssignAgeGroupUP



```sql
-- Determines age group of client based on their age (on update)
DELIMITER $$

CREATE TRIGGER AssignAgeGroupUP
BEFORE UPDATE ON Client
FOR EACH ROW
BEGIN
    IF NEW.Is_Student THEN
        SET NEW.AgeGroup = 'Joven';
    ELSEIF NEW.Age < 26 THEN
        SET NEW.AgeGroup = 'Joven';
    ELSEIF NEW.Age BETWEEN 26 AND 65 THEN
        SET NEW.AgeGroup = 'Normal';
    ELSEIF NEW.Age >= 65 THEN
        SET NEW.AgeGroup = 'Tercera Edad';
    END IF;
END$$


DELIMITER ;
```

Figure 19 AssignAgeGroupUP Trigger Code

Updates a client’s AgeGroup attribute based on their new age.



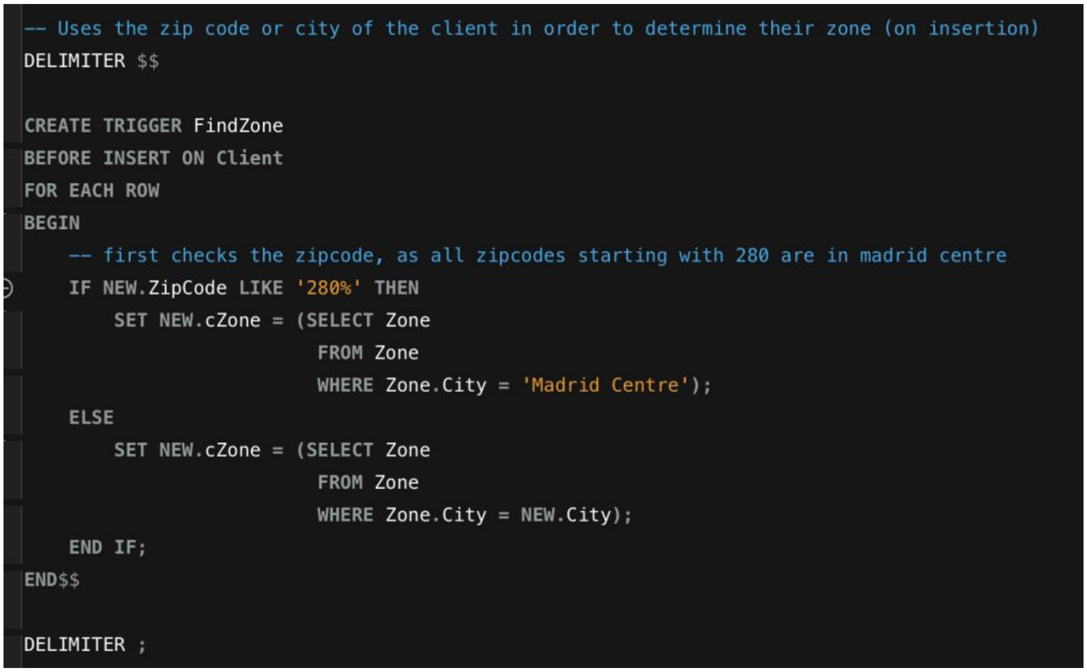
CID	Age	AgeGroup
15	40	Normal
NULL	NULL	NULL

```sql
UPDATE Client
SET DOB = '1954-06-22'
WHERE CID = '15';
```

CID	Age	AgeGroup
15	70	Tercera Edad
NULL	NULL	NULL

Figure 20 Example of AssignAgeGroup Trigger Functioning

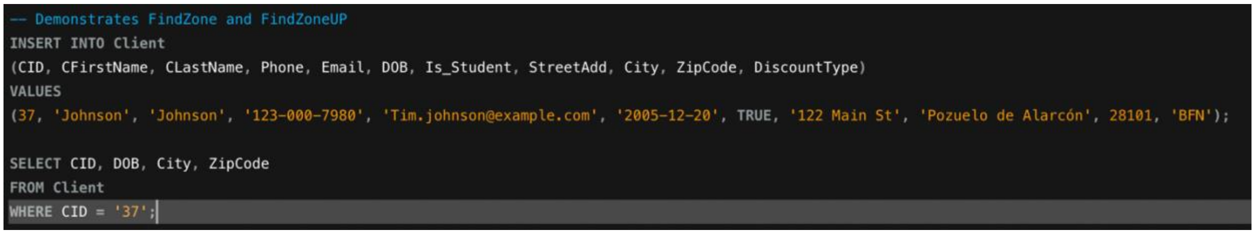
5) FindZone



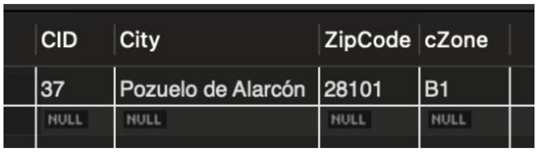
```sql
-- Uses the zip code or city of the client in order to determine their zone (on insertion)
DELIMITER $$

CREATE TRIGGER FindZone
BEFORE INSERT ON Client
FOR EACH ROW
BEGIN
    -- first checks the zipcode, as all zipcodes starting with 280 are in madrid centre
    IF NEW.ZipCode LIKE '280%' THEN
        SET NEW.cZone = (SELECT Zone
                         FROM Zone
                         WHERE Zone.City = 'Madrid Centre');
    ELSE
        SET NEW.cZone = (SELECT Zone
                         FROM Zone
                         WHERE Zone.City = NEW.City);
    END IF;
END$$

DELIMITER ;
```

Figure 21 FindZone Trigger Code

Fills a client's cZone attribute on insertion according to their city and zip code from the Zone table (if their zip code begins with 280, they are from Madrid Centre). The zone later determines the price of their card!



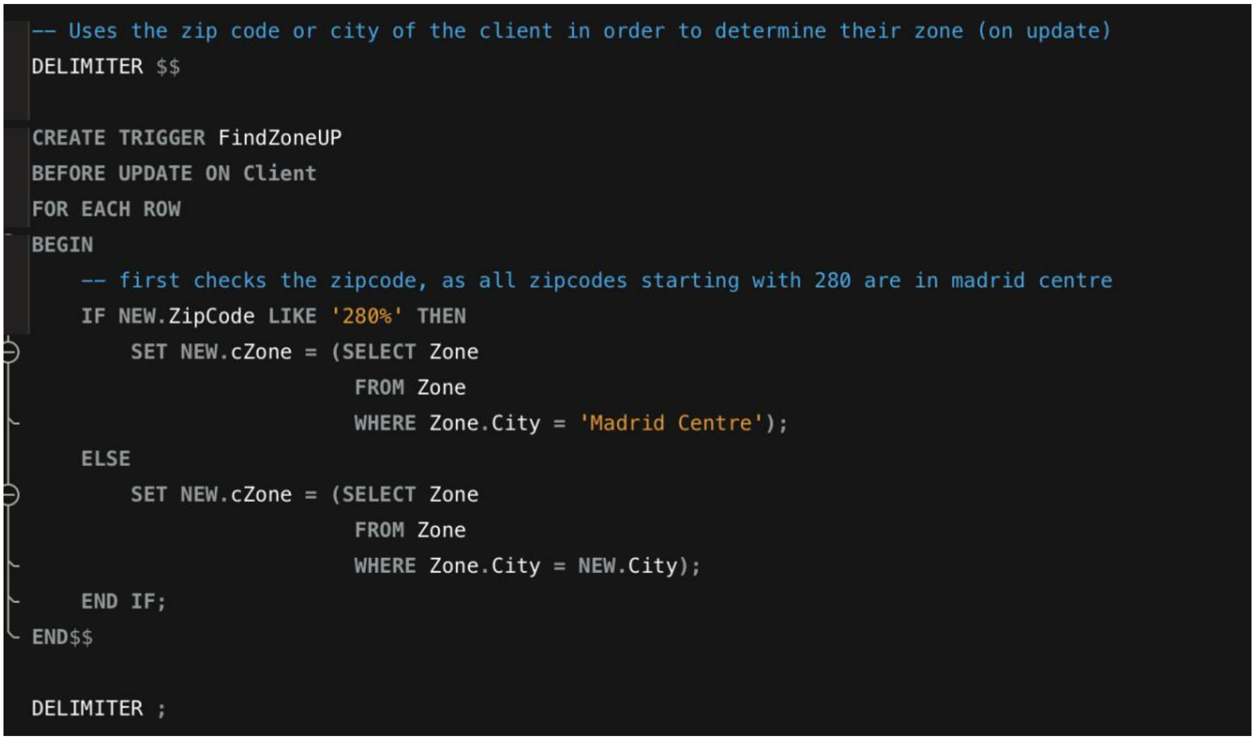
```sql
-- Demonstrates FindZone and FindZoneUP
INSERT INTO Client
(CID, CFirstName, CLastName, Phone, Email, DOB, Is_Student, StreetAdd, City, ZipCode, DiscountType)
VALUES
(37, 'Johnson', 'Johnson', '123-000-7980', 'Tim.johnson@example.com', '2005-12-20', TRUE, '122 Main St', 'Pozuelo de Alarcón', 28101, 'BFN');

SELECT CID, DOB, City, ZipCode
FROM Client
WHERE CID = '37';
```



CID	City	ZipCode	cZone
37	Pozuelo de Alarcón	28101	B1
NULL	NULL	NULL	NULL

Figure 22 Example of FindZone Trigger Functioning

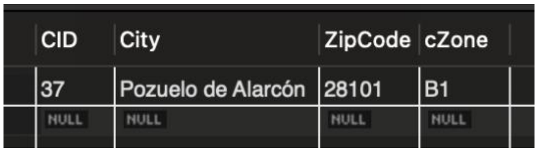
6) FindZoneUP



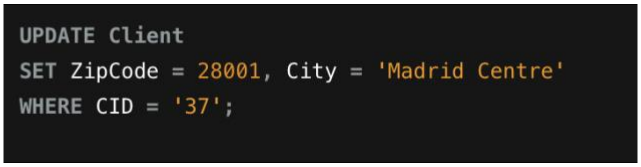
```
-- Uses the zip code or city of the client in order to determine their zone (on update)
DELIMITER $$

CREATE TRIGGER FindZoneUP
BEFORE UPDATE ON Client
FOR EACH ROW
BEGIN
    -- first checks the zipcode, as all zipcodes starting with 280 are in madrid centre
    IF NEW.ZipCode LIKE '280%' THEN
        SET NEW.cZone = (SELECT Zone
                          FROM Zone
                          WHERE Zone.City = 'Madrid Centre');
    ELSE
        SET NEW.cZone = (SELECT Zone
                          FROM Zone
                          WHERE Zone.City = NEW.City);
    END IF;
END$$

DELIMITER ;
```

Figure 23 FindZoneUP Trigger Code

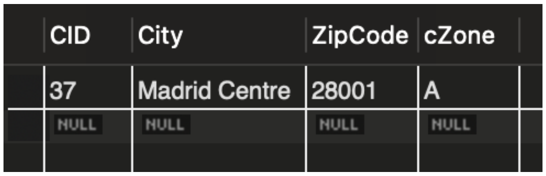
Updates a client's cZone attribute on insertion according to changes made to their city and zip code from the Zone table (if their zip code begins with 280, they are from Madrid Centre). The zone helps determine the price of their card!



CID	City	ZipCode	cZone
37	Pozuelo de Alarcón	28101	B1
NULL	NULL	NULL	NULL



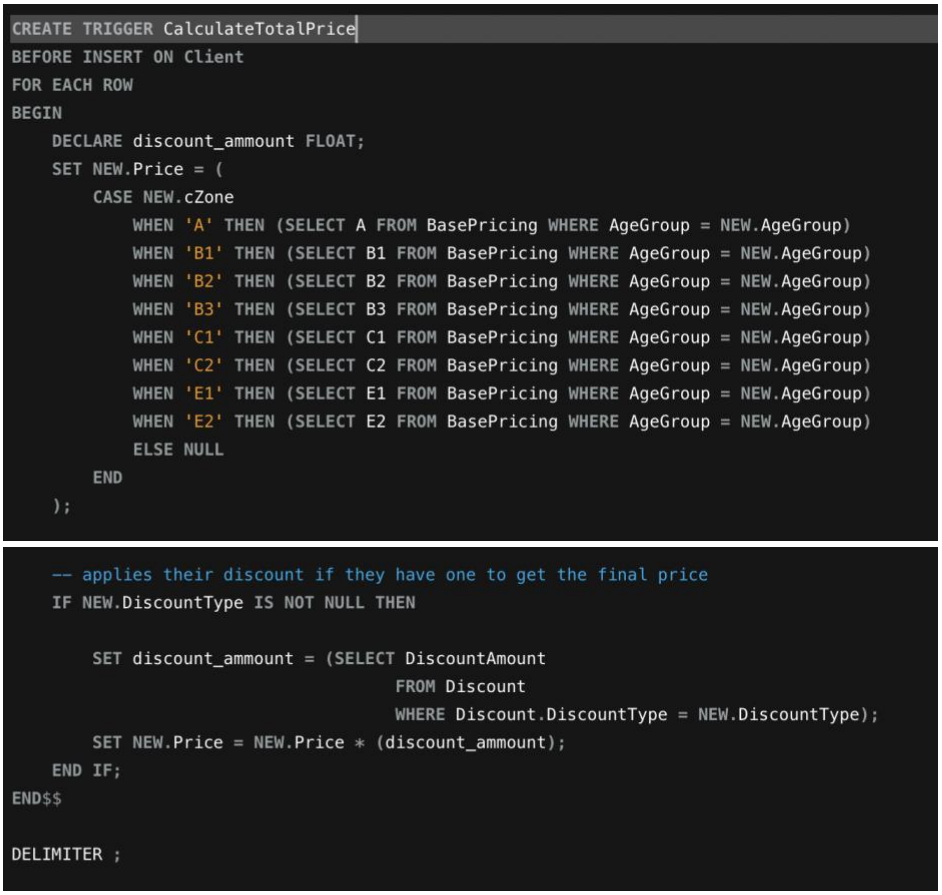
```
UPDATE Client
SET ZipCode = 28001, City = 'Madrid Centre'
WHERE CID = '37';
```



CID	City	ZipCode	cZone
37	Madrid Centre	28001	A
NULL	NULL	NULL	NULL

Figure 24 Example of FindZoneUP Trigger Functioning

7) CalculateTotalPrice

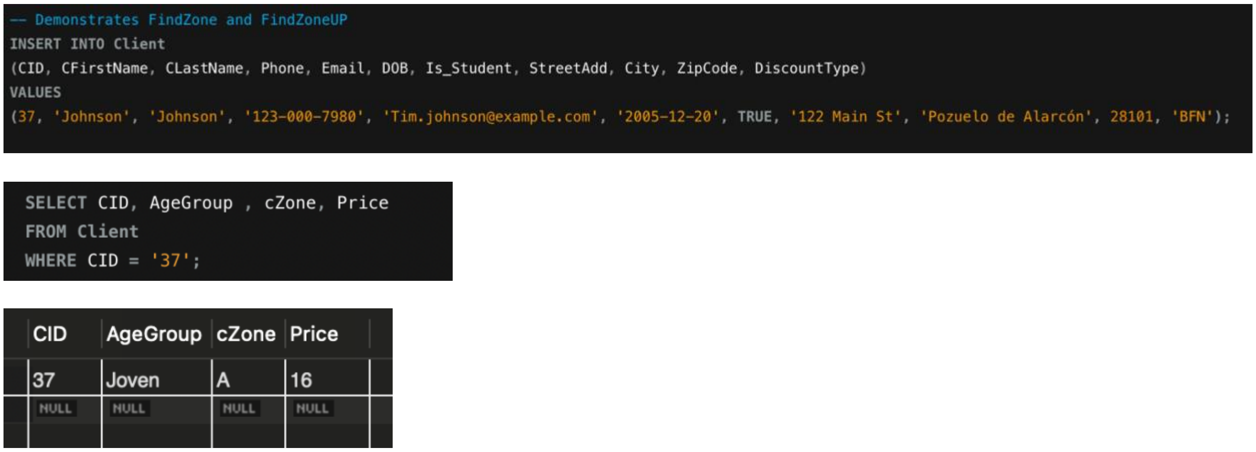


```
CREATE TRIGGER CalculateTotalPrice
BEFORE INSERT ON Client
FOR EACH ROW
BEGIN
    DECLARE discount_ammount FLOAT;
    SET NEW.Price = (
        CASE NEW.cZone
            WHEN 'A' THEN (SELECT A FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'B1' THEN (SELECT B1 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'B2' THEN (SELECT B2 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'B3' THEN (SELECT B3 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'C1' THEN (SELECT C1 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'C2' THEN (SELECT C2 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'E1' THEN (SELECT E1 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'E2' THEN (SELECT E2 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            ELSE NULL
        END
    );
```

```
    -- applies their discount if they have one to get the final price
    IF NEW.DiscountType IS NOT NULL THEN

        SET discount_ammount = (SELECT DiscountAmount
                              FROM Discount
                              WHERE Discount.DiscountType = NEW.DiscountType);
        SET NEW.Price = NEW.Price * (discount_ammount);
    END IF;
END$$

DELIMITER ;
```

Figure 25 CalculateTotalPrice Trigger Code

First, this trigger uses the client's zone and age group to determine their base price of a card from the base pricing table. It then checks if the client uses a discount, and if so, it applies the discount to the base price. Then, the client's card price is filled.



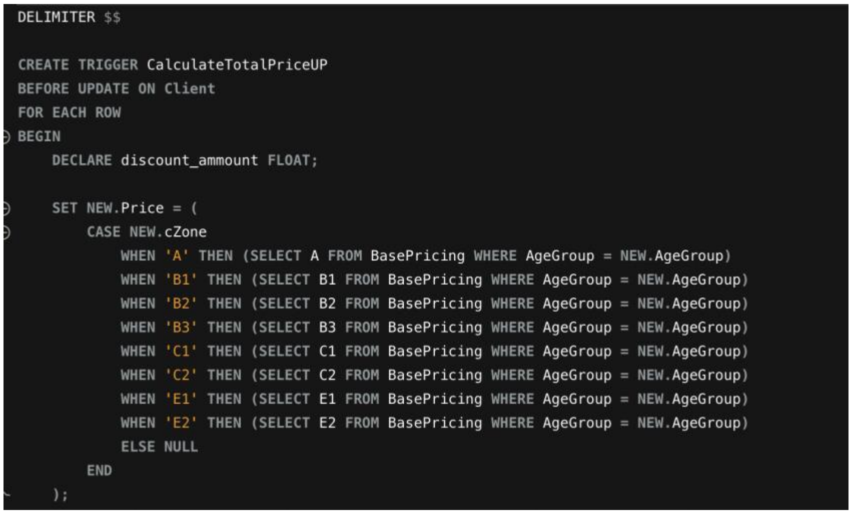
```
-- Demonstrates FindZone and FindZoneUP
INSERT INTO Client
(CID, CFirstName, CLastName, Phone, Email, DOB, Is_Student, StreetAdd, City, ZipCode, DiscountType)
VALUES
(37, 'Johnson', 'Johnson', '123-000-7980', 'Tim.johnson@example.com', '2005-12-20', TRUE, '122 Main St', 'Pozuelo de Alarcón', 28101, 'BFN');
```

```
SELECT CID, AgeGroup , cZone, Price
FROM Client
WHERE CID = '37';
```

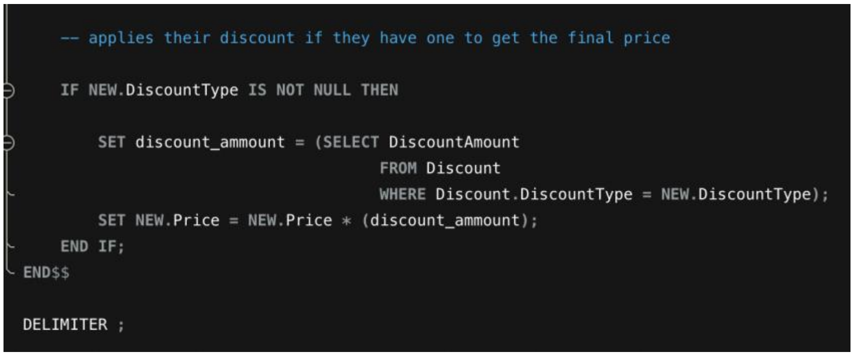
CID	AgeGroup	cZone	Price
37	Joven	A	16
NULL	NULL	NULL	NULL

Figure 26 Example of CalculateTotalPrice Trigger Functioning

8) CalculateTotalPriceUP



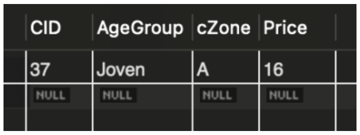
```
DELIMITER $$

CREATE TRIGGER CalculateTotalPriceUP
BEFORE UPDATE ON Client
FOR EACH ROW
BEGIN
    DECLARE discount_ammount FLOAT;

    SET NEW.Price = (
        CASE NEW.cZone
            WHEN 'A' THEN (SELECT A FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'B1' THEN (SELECT B1 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'B2' THEN (SELECT B2 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'B3' THEN (SELECT B3 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'C1' THEN (SELECT C1 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'C2' THEN (SELECT C2 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'E1' THEN (SELECT E1 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            WHEN 'E2' THEN (SELECT E2 FROM BasePricing WHERE AgeGroup = NEW.AgeGroup)
            ELSE NULL
        END
    );
```



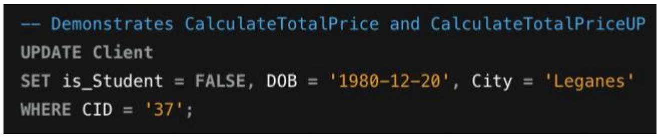
```
    -- applies their discount if they have one to get the final price

    IF NEW.DiscountType IS NOT NULL THEN

        SET discount_ammount = (SELECT DiscountAmount
                                FROM Discount
                                WHERE Discount.DiscountType = NEW.DiscountType);
        SET NEW.Price = NEW.Price * (discount_ammount);
    END IF;
END$$

DELIMITER ;
```

Figure 27 CalculateTotalPriceUP Trigger Code

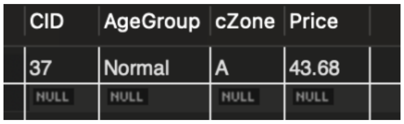
This trigger uses the client's zone and age group to determine their base price of a card from the base pricing table. It does this whenever the client's address or age may change. It then checks if the client uses a discount, and if so, it applies the discount to the base price. Then, the client's card price is updated.



CID	AgeGroup	cZone	Price
37	Joven	A	16
NULL	NULL	NULL	NULL



```
-- Demonstrates CalculateTotalPrice and CalculateTotalPriceUP
UPDATE Client
SET is_Student = FALSE, DOB = '1980-12-20', City = 'Leganes'
WHERE CID = '37';
```

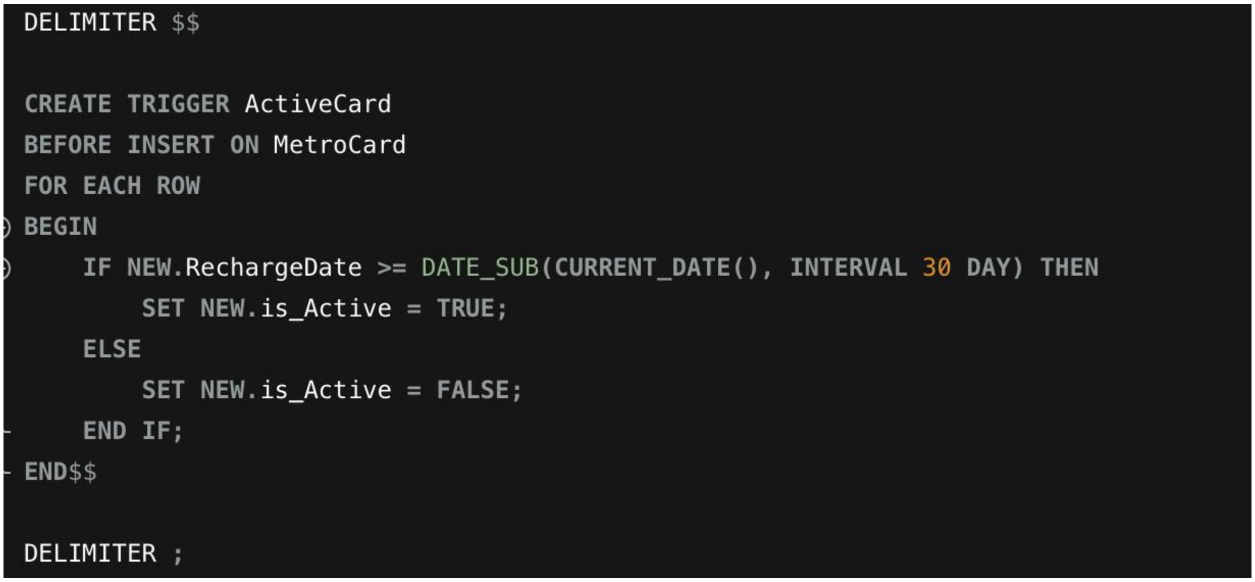


CID	AgeGroup	cZone	Price
37	Normal	A	43.68
NULL	NULL	NULL	NULL

Figure 28 Example of CalculateTotalPriceUP Trigger Functioning

For MetroCard:

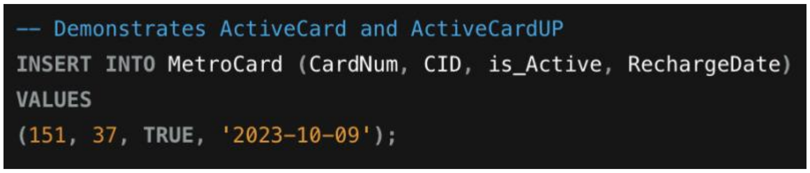
9) ActiveCard



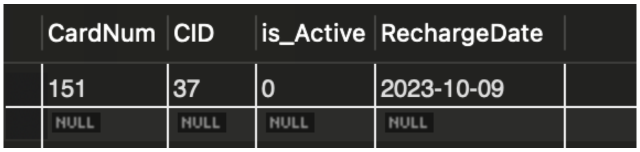
```
DELIMITER $$

CREATE TRIGGER ActiveCard
BEFORE INSERT ON MetroCard
FOR EACH ROW
BEGIN
    IF NEW.RechargeDate >= DATE_SUB(CURRENT_DATE(), INTERVAL 30 DAY) THEN
        SET NEW.is_Active = TRUE;
    ELSE
        SET NEW.is_Active = FALSE;
    END IF;
END$$

DELIMITER ;
```

Figure 29 ActiveCard Trigger Code

This trigger serves to determine if a MetroCard is active or not on insertion. While we assume that metro cards will only be added with a recharge date of the current day (thus active) I still included this trigger in case of extenuating circumstances.



```
-- Demonstrates ActiveCard and ActiveCardUP
INSERT INTO MetroCard (CardNum, CID, is_Active, RechargeDate)
VALUES
(151, 37, TRUE, '2023-10-09');
```

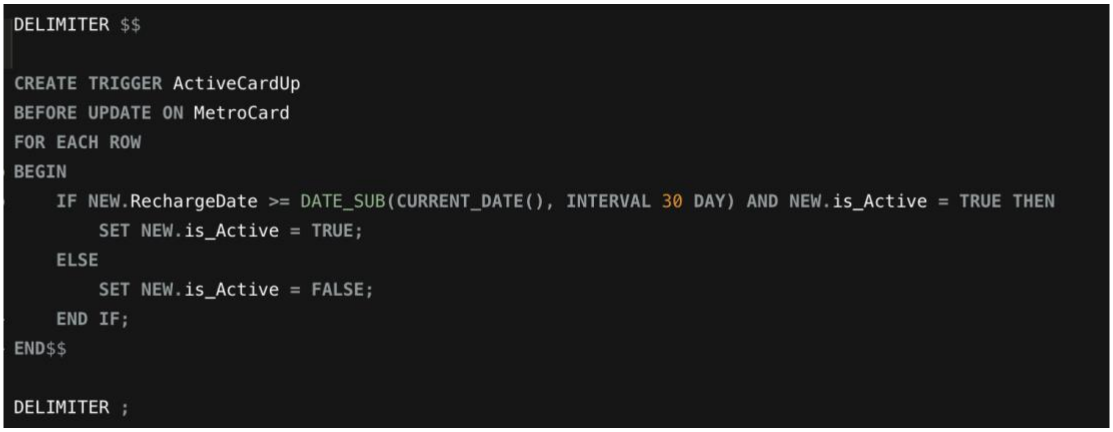


CardNum	CID	is_Active	RechargeDate
151	37	0	2023-10-09
NULL	NULL	NULL	NULL

Figure 30 Example of ActiveCard Trigger Functioning

As we can see, despite the card being inserted as active, since the recharge date is over 30 days before the current date, it is set to inactive.

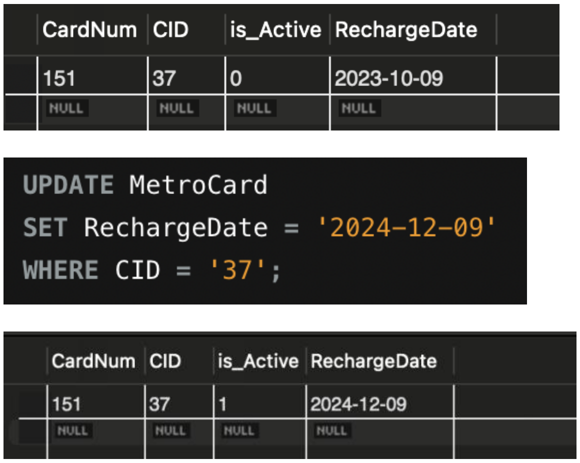
10) ActiveCardUP



```
DELIMITER $$

CREATE TRIGGER ActiveCardUp
BEFORE UPDATE ON MetroCard
FOR EACH ROW
BEGIN
    IF NEW.RechargeDate >= DATE_SUB(CURRENT_DATE(), INTERVAL 30 DAY) AND NEW.is_Active = TRUE THEN
        SET NEW.is_Active = TRUE;
    ELSE
        SET NEW.is_Active = FALSE;
    END IF;
END$$

DELIMITER ;
```

Figure 31 ActiveCardUP Trigger Code

This trigger serves to determine if a MetroCard is active or not according to the current date. If the date is 30 days past the last recharge of the card, the card changes to inactive. This is checked each day with an event that has been created within the code, and whenever the card becomes expired, it is set to inactive and must be "recharged". This also allows for functionality of the system being able to deactivate a card.



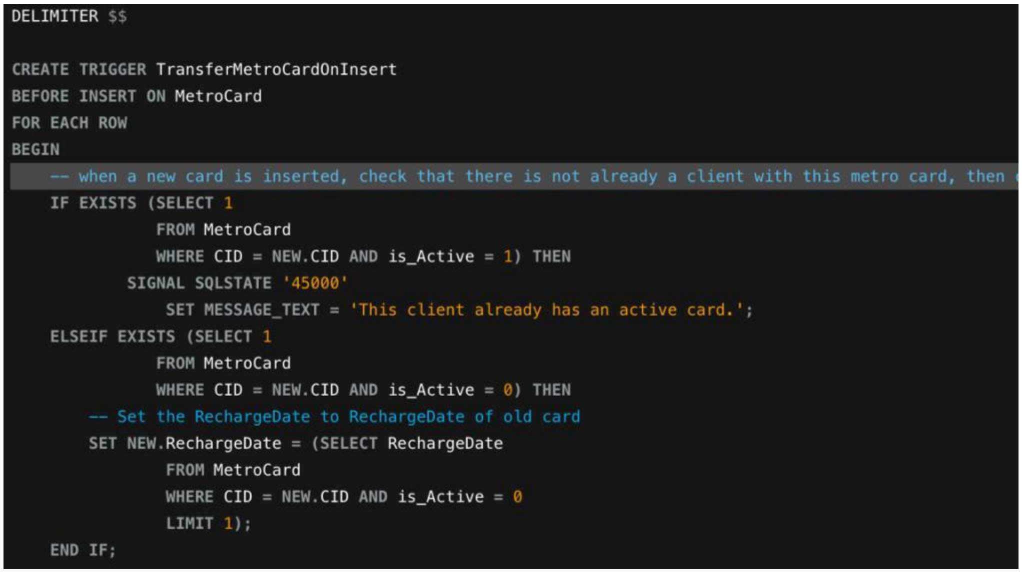
CardNum	CID	is_Active	RechargeDate
151	37	0	2023-10-09
NULL	NULL	NULL	NULL

```
UPDATE MetroCard
SET RechargeDate = '2024-12-09'
WHERE CID = '37';
```

CardNum	CID	is_Active	RechargeDate
151	37	1	2024-12-09
NULL	NULL	NULL	NULL

Figure 32 Example of ActiveCardUP Trigger Functioning

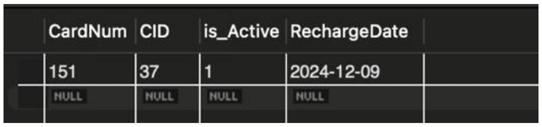
11) TransferMetroCardOnInsert



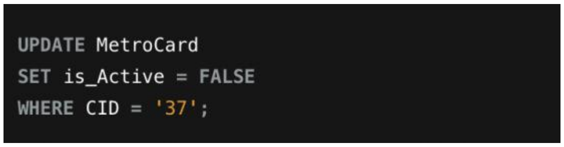
```
DELIMITER $$

CREATE TRIGGER TransferMetroCardOnInsert
BEFORE INSERT ON MetroCard
FOR EACH ROW
BEGIN
    -- when a new card is inserted, check that there is not already a client with this metro card, then
    IF EXISTS (SELECT 1
               FROM MetroCard
               WHERE CID = NEW.CID AND is_Active = 1) THEN
            SIGNAL SQLSTATE '45000'
                SET MESSAGE_TEXT = 'This client already has an active card.';
    ELSEIF EXISTS (SELECT 1
               FROM MetroCard
               WHERE CID = NEW.CID AND is_Active = 0) THEN
        -- Set the RechargeDate to RechargeDate of old card
        SET NEW.RechargeDate = (SELECT RechargeDate
                FROM MetroCard
                WHERE CID = NEW.CID AND is_Active = 0
                LIMIT 1);
    END IF;
```

Figure 33 TransferMetroCardOnInsert Trigger Code

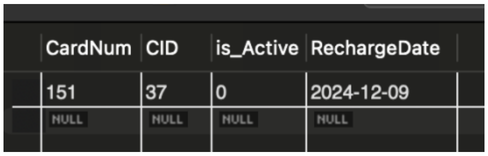
This triggers helps in replacing a metro card. When we set a metro card to inactive, and then we create a new one with the same CID (client creates a new one), it transfers the old recharge date to the new card, essentially making it so they don't lost out on the days they paid for. It also ensures that a client can only have 1 active metro card.



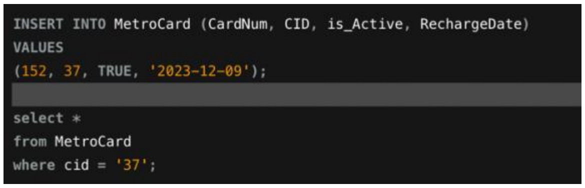
CardNum	CID	is_Active	RechargeDate
151	37	1	2024-12-09
NULL	NULL	NULL	NULL



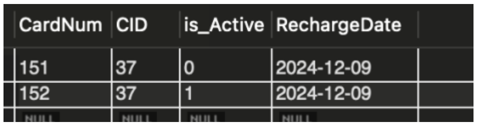
```
UPDATE MetroCard
SET is_Active = FALSE
WHERE CID = '37';
```



CardNum	CID	is_Active	RechargeDate
151	37	0	2024-12-09
NULL	NULL	NULL	NULL



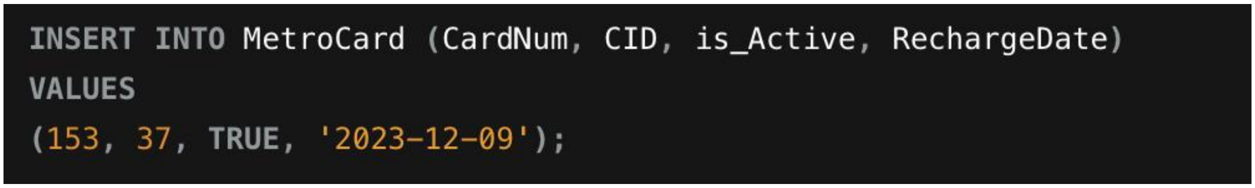
```
INSERT INTO MetroCard (CardNum, CID, is_Active, RechargeDate)
VALUES
(152, 37, TRUE, '2023-12-09');

select *
from MetroCard
where cid = '37';
```



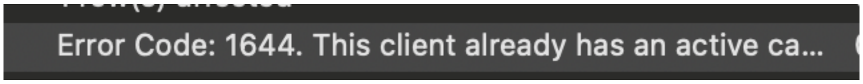
CardNum	CID	is_Active	RechargeDate
151	37	0	2024-12-09
152	37	1	2024-12-09
NULL	NULL	NULL	NULL

Figure 34 Example of TransferMetroCardOnInsert Trigger Functioning Part 1

As we can see, the new card inserted takes the recharge date of the inactivated card!



```
INSERT INTO MetroCard (CardNum, CID, is_Active, RechargeDate)
VALUES
(153, 37, TRUE, '2023-12-09');
```



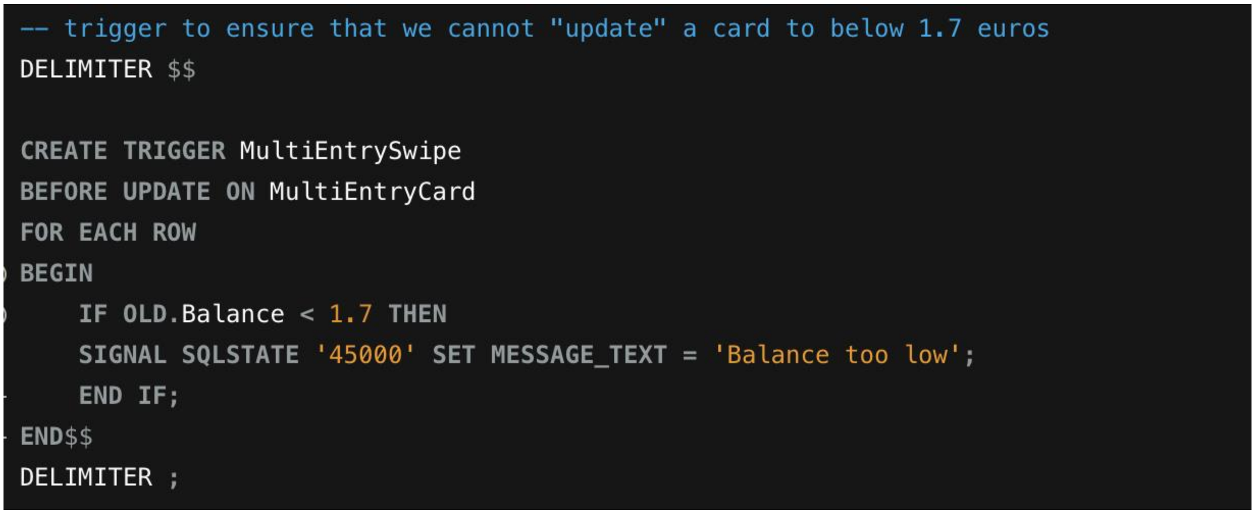
```
Error Code: 1644. This client already has an active ca...
```

Figure 35 Example of TransferMetroCardOnInsert Trigger Functioning Part 2

Additionally, we see that we also can't add another active metro card for this client since they already have one!

For MultiEntry:

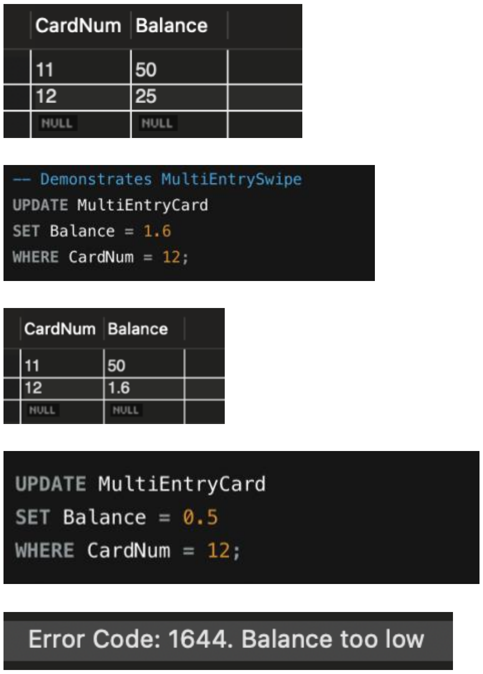
12) MultiEntrySwipe



```
-- trigger to ensure that we cannot "update" a card to below 1.7 euros
DELIMITER $$

CREATE TRIGGER MultiEntrySwipe
BEFORE UPDATE ON MultiEntryCard
FOR EACH ROW
BEGIN
    IF OLD.Balance < 1.7 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Balance too low';
    END IF;
END$$
DELIMITER ;
```

Figure 36 MultiEntrySwipe Trigger Code

This keeps track of updates to the balance of a multi entry card. If the balance of an existing card becomes below 1.7, and then is attempted to be used, then the balance is too low (since minimum swipe is 1.7 euros), and it sends an error, causing the user to have to put more money on. (there must be more than 1.7 euros to use card)



CardNum	Balance
11	50
12	25
NULL	NULL

```
-- Demonstrates MultiEntrySwipe
UPDATE MultiEntryCard
SET Balance = 1.6
WHERE CardNum = 12;
```

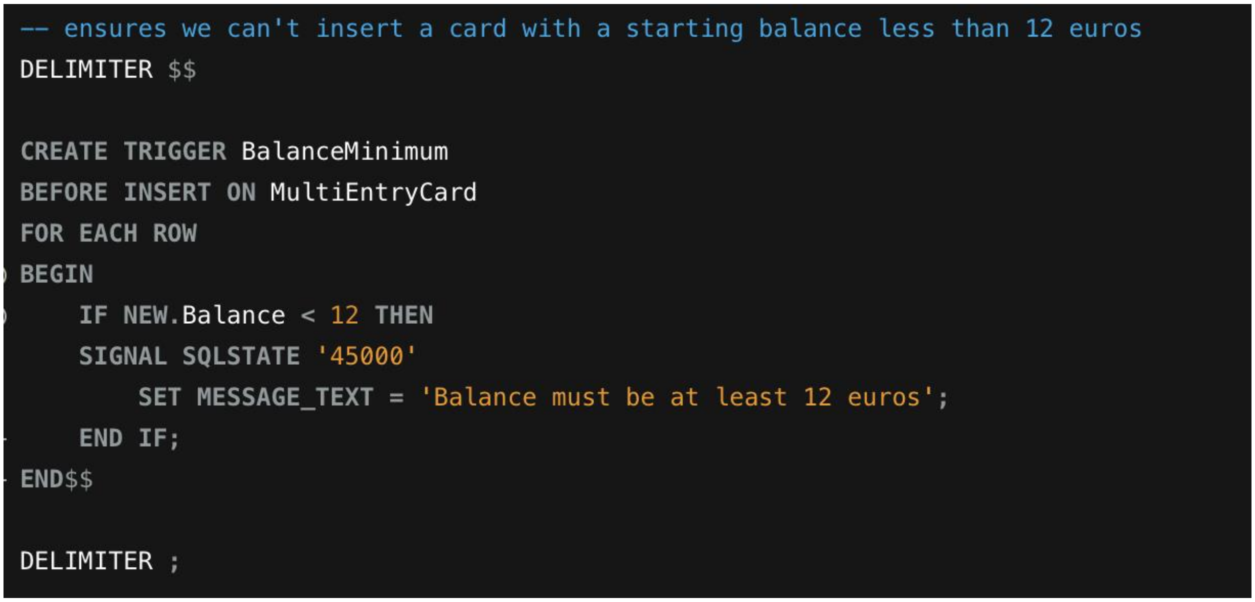
CardNum	Balance
11	50
12	1.6
NULL	NULL

```
UPDATE MultiEntryCard
SET Balance = 0.5
WHERE CardNum = 12;
```

```
Error Code: 1644. Balance too low
```

Figure 37 Example of MultiEntrySwipe Trigger Functioning

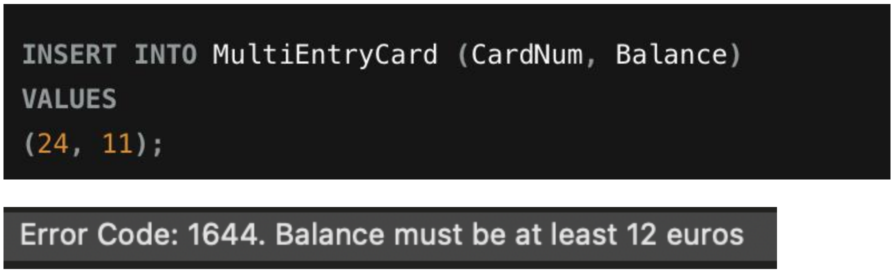
13) BalanceMinimum



```
-- ensures we can't insert a card with a starting balance less than 12 euros
DELIMITER $$

CREATE TRIGGER BalanceMinimum
BEFORE INSERT ON MultiEntryCard
FOR EACH ROW
BEGIN
    IF NEW.Balance < 12 THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Balance must be at least 12 euros';
    END IF;
END$$

DELIMITER ;
```

Figure 38 BalanceMinimum Trigger Code

This keeps track of insertions of multi entrance cards, and ensures that they are created with a balance of at least 12 euros.



```
INSERT INTO MultiEntryCard (CardNum, Balance)
VALUES
(24, 11);
```

```
Error Code: 1644. Balance must be at least 12 euros
```

Figure 39 Example of BalanceMinimum Trigger Functioning