

Query: register a new client

Command for inserting a new client

```
INSERT INTO MetroClient (id_num, id_type, first_name, middle_initial, last_name, email, phone, street, city, zipcode, birthday, student_state, discount)
VALUES
('10000000A', 'Passport', 'Juan', 'M', 'Gomez', 'juan.gomez@example.com', '123456789', 'Calle Mayor, 5', 'Madrid', '28001', '1985-06-15', FALSE, NULL);
```

After inserting a new client from the insert command above. The client type is determined by age and age is determined from birthday.

id_num	id_type	client_type	first_name	middle_initial	last_name	email	phone	street	city	zipcode	birthday	age	discount	student_sta...
10000000A	Passport	Normal	Juan	M	Gomez	juan.gomez@example.com	123456789	Calle Mayor, 5	Madrid	28001	1985-06-15	38	NULL	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query: recharge prepaid

Before the card is inactive becuase the expiration is past

card_num	card_state	card_fee	last_charge	expiration	zone_category
1000000000	inactive	54.60	2024-04-01	2024-05-01	A
NULL	NULL	NULL	NULL	NULL	NULL

This command updates the prepaid card to set the last change date to today (03/05)

```
UPDATE PrepaidCard
SET last_charge = CURRENT_DATE
WHERE card_num = '1000000000';
```

The card is now active because the last charge was changed to 03/05/24 so the expiration day has not passed

card_num	card_state	card_fee	last_charge	expiration	zone_category
1000000000	active	54.60	2024-05-03	2024-06-02	A
NULL	NULL	NULL	NULL	NULL	NULL

These are the triggers that update the expiration date and card state

```
DELIMITER //
CREATE TRIGGER update_expiration
BEFORE UPDATE ON PrepaidCard
FOR EACH ROW
BEGIN
    SET NEW.expiration = DATE_ADD(NEW.last_charge, INTERVAL 30 DAY);
END; //
DELIMITER ;
```

```

DELIMITER //
CREATE TRIGGER update_card_state
BEFORE UPDATE ON PrepaidCard
FOR EACH ROW
BEGIN
    IF NEW.expiration < CURDATE() THEN
        SET NEW.card_state = 'inactive';
    ELSE
        SET NEW.card_state = 'active';
    END IF;
END; //
DELIMITER ;

```

## Query: recharge multi-entrance

This card has a balance of 15.00

card_num	initial_amount	balance	cost_per_ri...	add_money_amount	ride_metro
2100000000	15.00	15.00	1.70	NULL	0
NULL	NULL	NULL	NULL	NULL	NULL

This command should update the card balance by adding 3 to the balance

```

UPDATE MultiEntranceCard
SET add_money_amount = 3.00
WHERE card_num = '2100000000';

```

The result has the updated balance of 18.00

card_num	initial_amount	balance	cost_per_ri...	add_money_amount	ride_metro
2100000000	15.00	18.00	1.70	0.00	0
NULL	NULL	NULL	NULL	NULL	NULL

This is the trigger that adds money to the balance through the add\_money\_amount attribute. It verifies the amount is at least €1.70 and then updates the balance.

```

drop trigger add_money_multicard;
DELIMITER //
CREATE TRIGGER add_money_multicard
BEFORE UPDATE ON MultiEntranceCard
FOR EACH ROW
) BEGIN
) IF NEW.add_money_amount < 1.70 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid amount, must be at least €1.70';
    ELSE
        SET NEW.balance = OLD.balance + NEW.add_money_amount;
        SET NEW.add_money_amount = 0;
    END IF;
END; //
DELIMITER ;

```

## Query: replace a missing card

The first card is the original card that will be canceled and the second card is the replacement card

card_num	card_state	card_fee	last_charge	expiration	zone_category
1000000000	inactive	54.60	2024-04-01	2024-05-01	A
2000000000	inactive	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL

Here is the original relationship between client and card

owner_id_num	card_num
10000000A	1000000000
NULL	NULL

This is the command that creates the relationship between the owner of the first card and the replacement card with a new number

```
INSERT INTO OwnsPrepaidCard(owner_id_num, card_num)
VALUES
('10000000A', '2000000000');
```

The client now owns the replacement card, and the replacement card has all the same information as the original card.

owner_id_num	card_num
10000000A	2000000000
NULL	NULL

card_num	card_state	card_fee	last_charge	expiration	zone_category
2000000000	inactive	54.60	2024-04-01	2024-05-01	A
NULL	NULL	NULL	NULL	NULL	NULL

This trigger looks for the old card and copies the information to the new card and deletes the old card

```
DELIMITER //
CREATE TRIGGER replacePrepaid
BEFORE INSERT ON OwnsPrepaidCard
FOR EACH ROW
BEGIN
    DECLARE old_card_no VARCHAR(10);
    DECLARE old_card_status ENUM('active', 'inactive');
    DECLARE old_card_date DATE;
    DECLARE new_card_num VARCHAR(10);

    SELECT card_num INTO old_card_no
    FROM OwnsPrepaidCard WHERE owner_id_num = NEW.owner_id_num AND card_num != NEW.card_num;

    IF old_card_no IS NOT NULL THEN
        SELECT card_state, last_charge INTO old_card_status, old_card_date
        FROM PrepaidCard WHERE old_card_no = card_num;

        UPDATE PrepaidCard
        SET card_state = old_card_status,
            last_charge = old_card_date
        WHERE NEW.card_num = card_num;

        DELETE FROM PrepaidCard WHERE old_card_no = card_num;
    END IF;
END; //
DELIMITER ;
```

Query: cancel client registration

These are the tables showing the client, card, and relationship between the two

id_num	id_type	client_type	first_name	middle_initial	last_name	email	phone	street	city	zipcode	birthday	age	discount	student_sta...
10000000A	Passport	Normal	Juan	M	Gomez	juan.gomez@example.com	123456789	Calle Mayor, 5	Madrid	28001	1985-06-15	38	NULL	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

card_num	card_state	card_fee	last_charge	expiration	zone_category
1000000000	inactive	54.60	2024-04-01	2024-05-01	A
NULL	NULL	NULL	NULL	NULL	NULL

owner_id_num	card_num
10000000A	1000000000

This command cancels the client registration by deleting the client identified by the id

```
DELETE FROM MetroClient
WHERE id_num = '10000000A';
```

The after shows the empty client table, card, and relationship because the client was deleted

id_num	id_type	client_type	first_name	middle_initial	last_name	email	phone	street	city	zipcode	birthday	age	discount	student_sta...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

  

card_num	card_state	card_fee	last_charge	expiration	zone_category
NULL	NULL	NULL	NULL	NULL	NULL

  

owner_id_num	card_num

This is the trigger that deletes the client and the card. The relationship gets deleted in the foreign key specifications (on delete cascade)

```

DELIMITER //
CREATE TRIGGER cancel_client
BEFORE DELETE ON MetroClient
FOR EACH ROW
BEGIN
    DECLARE card_no VARCHAR(10);

    SELECT card_num INTO card_no
    FROM OwnsPrepaidCard WHERE OLD.id_num = owner_id_num;

    IF card_no IS NOT NULL THEN
        DELETE FROM PrepaidCard
        WHERE card_no = card_num;
    END IF;

    SELECT card_num INTO card_no
    FROM OwnsMultiEntranceCard WHERE OLD.id_num = owner_id_num;

    IF card_no IS NOT NULL THEN
        DELETE FROM MultiEntranceCard
        WHERE card_no = card_num;
    END IF;
END; //
DELIMITER ;

```

**Trigger: check card charging date and change card state to “0” if it is passed 30 days from the last charge**

This card is active because it is within 30 days of the expiration date

card_num	card_state	card_fee	last_charge	expiration	zone_category
1000000000	active	54.60	2024-04-12	2024-05-12	A
NULL	NULL	NULL	NULL	NULL	NULL

I manually changed the last charge date of the client to over 30 days past the current date. This updates the card state to inactive

card_num	card_state	card_fee	last_charge	expiration	zone_category
1000000000	inactive	54.60	2024-03-01	2024-03-31	A
NULL	NULL	NULL	NULL	NULL	NULL

This is the trigger that updates the card state

```

DELIMITER //
CREATE TRIGGER update_card_state
BEFORE UPDATE ON PrepaidCard
FOR EACH ROW
BEGIN
    IF NEW.expiration < CURDATE() THEN
        SET NEW.card_state = 'inactive';
    ELSE
        SET NEW.card_state = 'active';
    END IF;
END; //
DELIMITER ;

```

**Trigger: check the date of the client birthday and current date, once the client age status changed the card cost should be update automatically to the new price that suit that age**

The client starts as a Abono Joven because their age is 24. This means their card fee is 20€

id_num	id_type	client_type	first_name	middle_initial	last_name	email	phone	street	city	zipcode	birthday	age	discount	student_sta...
15000000F	NIE	Abono Joven	Javier	R	Garcia	javier.garcia@example.com	432109876	Calle Princesa, 15	Ribatejada	28300	2000-04-30	24	NULL	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

card_num	card_state	card_fee	last_charge	expiration	zone_category
1500000000	active	20.00	2024-04-06	2024-05-06	C1
NULL	NULL	NULL	NULL	NULL	NULL

I manually changed the client birthday to pretend they “aged” past 26. After their prepaid card last change is updated, meaning they recharged their prepaid card, the birthday is checked to see if their client type has changed. If it has, then the client type and the card fee gets updated. The client isn’t an Abono Joven anymore, so their card fee is 89.50€.

id_num	id_type	client_type	first_name	middle_initial	last_name	email	phone	street	city	zipcode	birthday	age	discount	student_sta...
15000000F	NIE	Normal	Javier	R	Garcia	javier.garcia@example.com	432109876	Calle Princesa, 15	Ribatejada	28300	1980-04-30	44	NULL	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

card_num	card_state	card_fee	last_charge	expiration	zone_category
1500000000	active	89.50	2024-05-02	2024-06-01	C1
NULL	NULL	NULL	NULL	NULL	NULL

This is the trigger that checks the age and updates the client type and card fee if the age is different.

```

DELIMITER //
CREATE TRIGGER update_age
BEFORE UPDATE ON PrepaidCard
FOR EACH ROW
BEGIN
    DECLARE current_year INT;
    DECLARE birth_year INT;
    DECLARE curr_date DATE;
    DECLARE bday DATE;
    DECLARE new_age INT;
    DECLARE owner_id VARCHAR(15);
    DECLARE curr_age INT;
    DECLARE stu_state BOOL;
    DECLARE base_price DECIMAL(5,2);
    DECLARE owner_type VARCHAR(10);
    DECLARE owner_city VARCHAR(30);
    DECLARE new_client_type ENUM('Normal', 'Senior', 'Abono Joven');

    SELECT owner_id_num INTO owner_id
    FROM OwnsprepaidCard WHERE NEW.card_num = card_num;

    SELECT age INTO curr_age
    FROM MetroClient WHERE owner_id = id_num;

    SELECT student_state INTO stu_state
    FROM MetroClient WHERE owner_id = id_num;

    SET curr_date = CURRENT_DATE();
    SET current_year = YEAR(curr_date);

    SELECT birthday INTO bday
    FROM MetroClient WHERE owner_id = id_num;

    SET birth_year = YEAR(bday);
    SET new_age = current_year - birth_year;

    IF MONTH(bday) > MONTH(curr_date) OR (MONTH(bday) = MONTH(curr_date) AND DAY(bday) > DAY(curr_date)) THEN
        SET new_age = new_age - 1;
    END IF;

    IF curr_age != new_age THEN
        UPDATE MetroClient
        SET age = new_age
        WHERE owner_id = id_num;

        IF new_age < 26 OR stu_state = TRUE THEN
            UPDATE MetroClient
            SET client_type = 'Abono Joven'
            WHERE owner_id = id_num;
            SET new_client_type = 'Abono Joven';
        ELSEIF new_age > 65 THEN
            UPDATE MetroClient
            SET client_type = 'Senior'
            WHERE owner_id = id_num;
            SET new_client_type = 'Senior';
        ELSE
            UPDATE MetroClient
            SET client_type = 'Normal'
            WHERE owner_id = id_num;
            SET new_client_type = 'Normal';
        END IF;

        IF new_client_type = 'Abono Joven' OR new_client_type = 'Senior' THEN
            SELECT price INTO base_price
            FROM ClientType_Price WHERE new_client_type = type;
        ELSE
            SELECT city INTO owner_city
            FROM MetroClient WHERE owner_id = id_num;

            SELECT fee INTO base_price
            FROM ZoneFee WHERE owner_city = zone_name;
        END IF;

        SET NEW.card_fee = base_price;
    END IF;
END //
DELIMITER ;

```

**Trigger: determine the card fee according to the city (zone) when the client register or update address**

The card fee is 54.60€ because the client lives in Madrid

card_num	card_state	card_fee	last_charge	expiration	zone_category
1000000000	inactive	54.60	2024-04-01	2024-05-01	A
NULL	NULL	NULL	NULL	NULL	NULL

id_num	id_type	client_type	first_name	middle_initial	last_name	email	phone	street	city	zipcode	birthday	age	discount	student_sta...
10000000A	Passport	Normal	Juan	M	Gomez	juan.gomez@example.com	123456789	Calle Mayor, 5	Madrid	28001	1985-06-15	38	NULL	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

This command changes client city from Madrid to Pinto which changes their card fee

```
UPDATE MetroClient
SET city = 'Pinto'
WHERE id_num = '10000000A';
```

The city is changed to Pinto so their card fee is now 72€ because it is in B1

card_num	card_state	card_fee	last_charge	expiration	zone_category
1000000000	inactive	72.00	2024-04-01	2024-05-01	B2
NULL	NULL	NULL	NULL	NULL	NULL

id_num	id_type	client_type	first_name	middle_initial	last_name	email	phone	street	city	zipcode	birthday	age	discount	student_sta...
10000000A	Passport	Normal	Juan	M	Gomez	juan.gomez@example.com	123456789	Calle Mayor, 5	Pinto	28001	1985-06-15	38	NULL	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

This is the trigger that changes the card fee on a prepaid card if the city changes

```
DELIMITER //
CREATE TRIGGER change_address
AFTER UPDATE ON MetroClient
FOR EACH ROW
BEGIN
    DECLARE base_price DECIMAL(5,2);
    DECLARE card_no VARCHAR(10);
    DECLARE owner_zone ENUM('A', 'B1', 'B2', 'B3', 'C1', 'C2', 'E1', 'E2');
    IF NEW.city != OLD.city THEN
        IF NEW.client_type = 'Abono Joven' OR NEW.client_type = 'Senior' THEN
            SELECT price INTO base_price
            FROM ClientType_Price WHERE NEW.client_type = type;
        ELSE
            SELECT zone INTO owner_zone
            FROM CityKey WHERE NEW.city = zone_name;

            SELECT zone_fee INTO base_price
            FROM Zonekey WHERE owner_zone = zone_fee;
        END IF;

        SELECT card_num INTO card_no
        FROM OwnsPrepaidCard WHERE NEW.id_num = owner_id_num;

        IF card_no IS NOT NULL THEN
            UPDATE PrepaidCard
            SET card_fee = base_price
            WHERE card_no = card_num;
        END IF;
    END IF;
END; //
DELIMITER ;
```

**Show the client First Name, Last Name, email, zone category, and the card price for all clients living in ‘Leganes’**

Total MetroClient table:



## The prepaid cards:

### Command:

**Result:**

## Display card status for a given client ID no.

Total MetroClient table, PrepaidCard, and OwnsPrepaidCard table:

[illegible]

card_num	card_state	card_fee	last_charge	expiration	owner_id_num	card_num
1000000000	active	54.60	2024-05-01	2024-05-31	10000000A	1000000000
1100000000	active	63.70	2024-05-02	2024-06-01	11000000B	1100000000
1200000000	active	57.60	2024-05-03	2024-06-02	12000000C	1200000000
1300000000	active	18.80	2024-05-04	2024-06-03	13000000D	1300000000
1400000000	active	20.00	2024-05-05	2024-06-04	14000000E	1400000000
1500000000	active	20.00	2024-04-06	2024-05-06	15000000F	1500000000
1600000000	active	12.00	2024-04-07	2024-05-07	16000000G	1600000000
1700000000	active	7.00	2024-04-08	2024-05-08	17000000H	1700000000
1800000000	active	6.30	2024-04-09	2024-05-09	18000000I	1800000000
1900000000	active	2.21	2024-04-10	2024-05-10	19000000J	1900000000
2000000000	active	5.04	2024-04-11	2024-05-11	20000000K	2000000000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Command:

```

SELECT
    ownscard.card_num AS "Card Number",
    card.card_state AS "Card Status"
FROM
    OwnsPrepaidCard ownscard
JOIN PrepaidCard card ON ownscard.card_num = card.card_num
WHERE
    ownscard.owner_id_num = '10000000A';

```

Result:

Card Number	Card Status
1000000000	active