Group 3

Berke Udunman, Bülent Arda Yılmaz, Nil Su Demir, Sezin Çağla Serficeli, Talha Arık

In this project, we are aiming to create a database for a Music Shop. This music shop is in the 90s and we sell products such as albums, and singles in various physical forms such as CDs, cassettes, and records. Thus, in this music store, there will not be any digital records and sales. It will have some staff who will manage the purchases by the customers and will take care of the products in general.

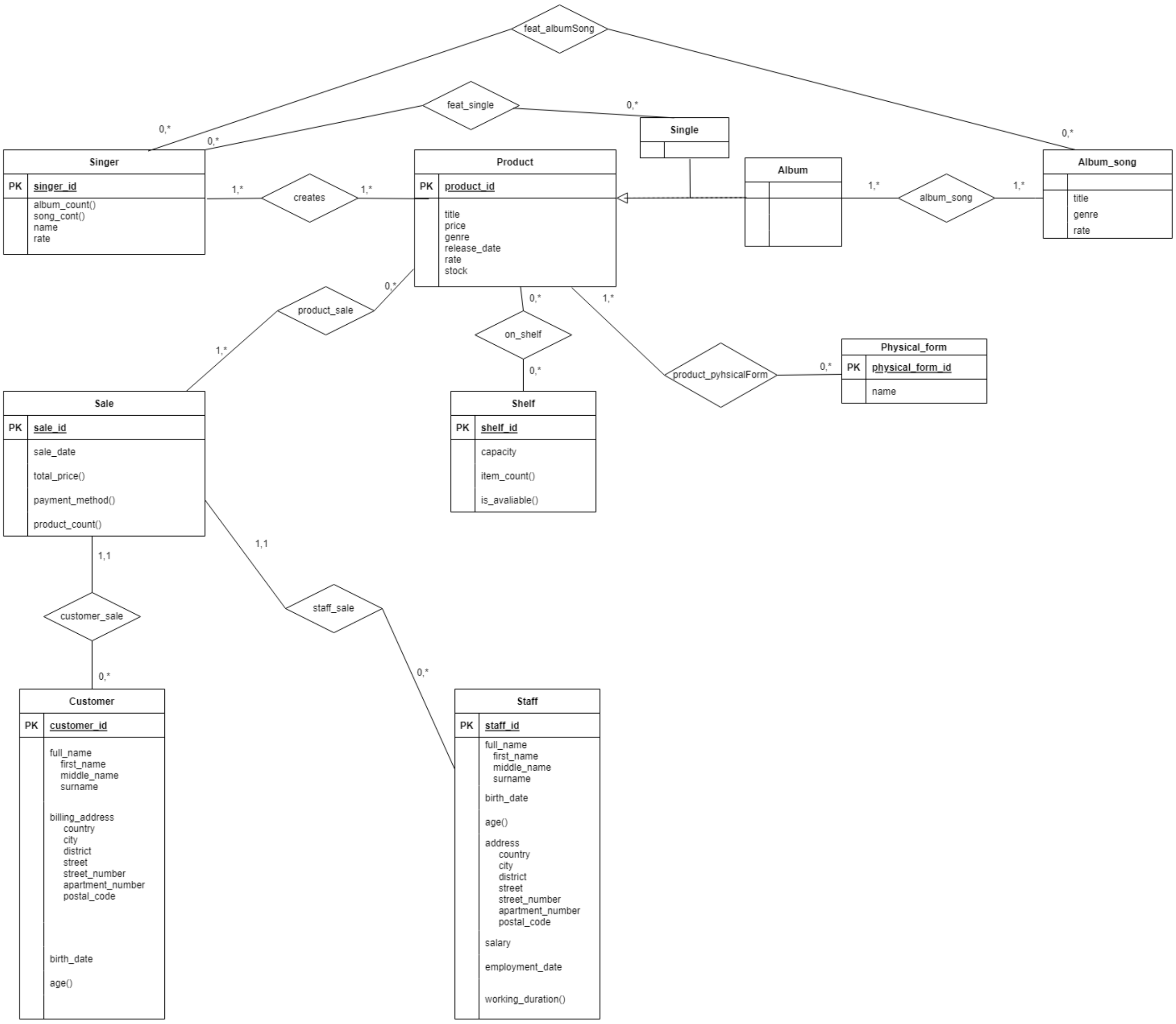
Project Step 2

- Describe your project in detail:

    Entity Sets & Attributes:

    - Singer: singer_id, name, song_count(), album_count(), rate
    - Product: product_id, title, price, genre, release_date, rate
    - Album: product_id, title, price, genre, release_date, rate
    - Single: product_id, title, price, genre, release_date, rate
    - Album_song: title, genre, rate
    - Physical_form: physical_form_id, name
    - Shelf: shelf_id, capacity, item_count(), is_available()
    - Customer: customer_id, first_name, middle_name, surname, country, city, district, street, street_number, apartment_number,postal_code, birth_date, age()
    - Staff: staff_id, first_name, middle_name, surname, birth_date, age(), salary, country, city, district, street, street_number, apartment_number,postal_code, employment_date, working_time()
    - Sale: sale_id, product_count(), sale_date, payment_method, total_price()

- ER Diagram & Its Explanation:



- The singer creates the product, and the product (there is an inheritance relation here) can be a single or an album. In addition, the album can have multiple album songs.
- The product needs to have a singer because the singer must produce an album or single for sale.
- The singer needs at least one product to be a singer.
- The shelf contains products. It should contain zero or more products (albums or singles), and customers can look for the shelf for buying music products.
- Every product can come in different physical forms. However, a product does not need to have every single different physical form in our Music Store (Ex: An album can be a CD or Record, but not a cassette or a different kind of physical form in our inventory).
- Sometimes a singer may contribute to another singer’s album song or single, and this relationship is called a “feat”.
- A sale can not exist without the single, the album, the staff, or the customer, so its participation is total in its relationships because the staff sells the music product(it can be a single or an album) to the customer. Customer buys at

least one product. Staff sells at least zero products because staff can fail the sale. In the sale, at least one single or one album should be purchased.

- Assumptions:
  - In this music store, there is no storage. We put the products on the shelves so that every product needs to be on a shelf.
  - Our music store only contains physical products, such as CDs, cassettes, and records.
  - Staff members are only salesmen.

- Business Rules:
  - G3’s Music Store is a single branch store and sells only physical copies of the music products.
  - On a sale, the staff member should sell the product to the customer face-to-face.
  - A single or an album can only be held on a single shelf.
  - In a sale, at least one single or one album should be purchased. An album consists of album songs, so a sale of a single album also means a sale of various songs.

- Reduction to Relational Schemas:

singer(singer_id, name, rate)
album(album_id, title, price, genre, release_date, rate, stock)
singles(single_id, title, price, genre, release_date, rate, stock)

singer_creates_album(singer_id, album_id)
singer_creates_single(singer_id, single_id)

feat_album_song(singer_id, album_song_id)
feat_single(singer_id, single_id)

album_song(album_song_id, title, genre, rate)
album_album_song(album_id, album_song_id)

sale(sale_id, sale_date, payment_method, customer_id, staff_id)

shelf(shelf_id, capacity)

customer(customer_id, first_name, middle_name, surname, country, city, district, street, street_number, apartment_number, postal_code, birth_date)

staff(staff_id, first_name, middle_name, surname, birth_date, country, city, district, street, street_number, apartment_number, postal_code, employment_date)

album_sale(album_id, sale_id)
single_sale(single_id, sale_id)

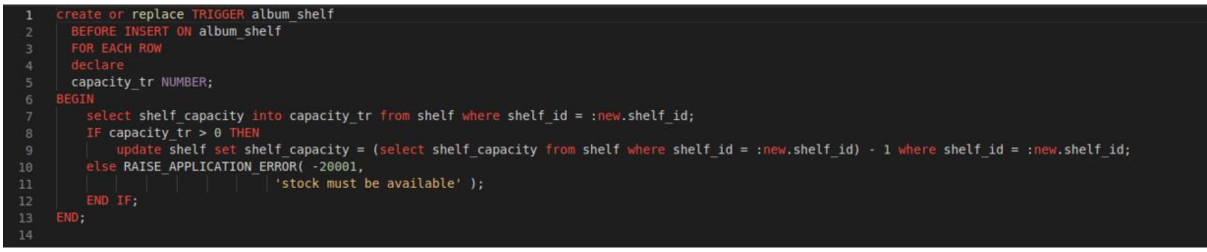
album_shelf(album_id, shelf_id)
single_shelf(single_id, shelf_id)

physical_form(physical_form_id, name)

album_physical_form(album_id, physical_form_id)
single_physical_form(single_id, physical_form_id)

- All Users of Our System:
  - The customer is the person who buys the albums or the songs.
  - The staff member is an employee of our Music Store who sells the products.

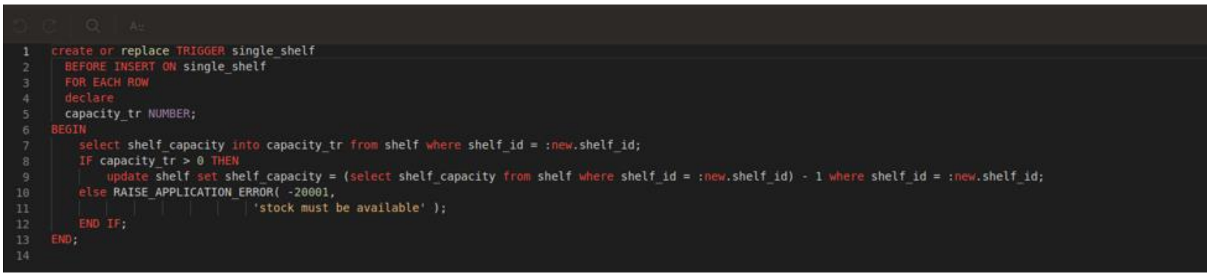
- Extra Features:
  - In addition to our tables, we have created various triggers to ensure our system works as planned.
  - Trigger 1: album_shelf



```
create or replace TRIGGER album_shelf
  BEFORE INSERT ON album_shelf
  FOR EACH ROW
  declare
  capacity_tr NUMBER;
BEGIN
    select shelf_capacity into capacity_tr from shelf where shelf_id = :new.shelf_id;
    IF capacity_tr > 0 THEN
        update shelf set shelf_capacity = (select shelf_capacity from shelf where shelf_id = :new.shelf_id) - 1 where shelf_id = :new.shelf_id;
    else RAISE_APPLICATION_ERROR( -20001,
                        'stock must be available' );
    END IF;
END;
```

Explanation of album_shelf trigger: If Album Shelf's capacity is reduced to zero, raise an error. In addition, if capacity > 0, capacity can be reduced by one.

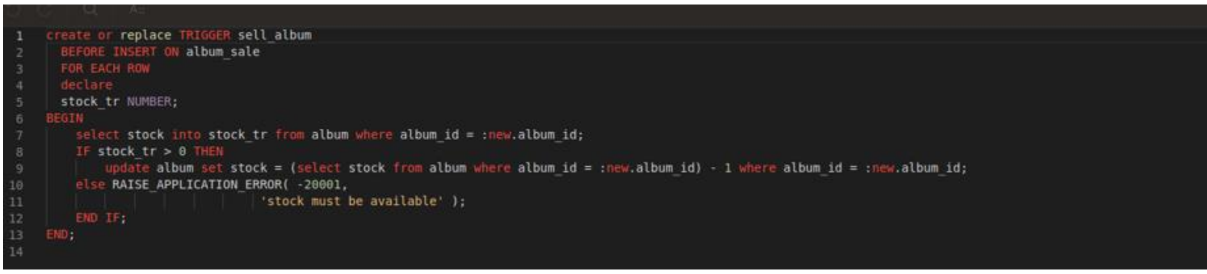
  - Trigger 2: single_shelf



```
create or replace TRIGGER single_shelf
  BEFORE INSERT ON single_shelf
  FOR EACH ROW
  declare
  capacity_tr NUMBER;
BEGIN
    select shelf_capacity into capacity_tr from shelf where shelf_id = :new.shelf_id;
    IF capacity_tr > 0 THEN
        update shelf set shelf_capacity = (select shelf_capacity from shelf where shelf_id = :new.shelf_id) - 1 where shelf_id = :new.shelf_id;
    else RAISE_APPLICATION_ERROR( -20001,
                        'stock must be available' );
    END IF;
END;
```

Explanation of album_shelf trigger: If Single Shelf's capacity is reduced to zero, raise an error. In addition, if capacity > 0, capacity can be reduced by one.

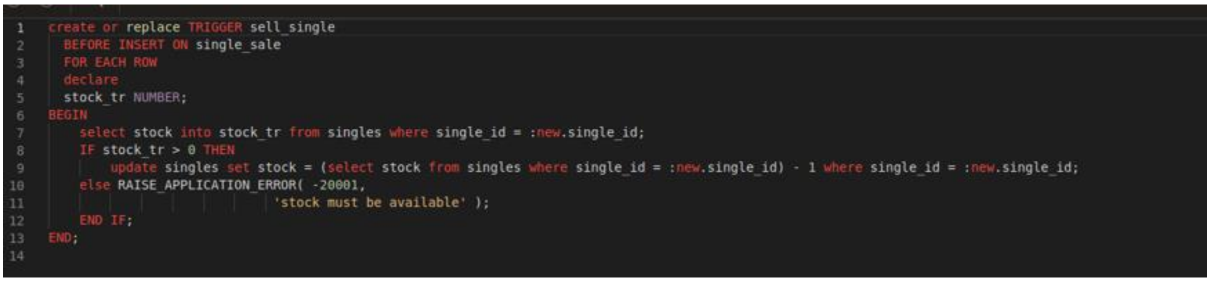
- Trigger 3: sell_album_trigger



```
create or replace TRIGGER sell_album
  BEFORE INSERT ON album_sale
  FOR EACH ROW
  declare
  stock_tr NUMBER;
BEGIN
    select stock into stock_tr from album where album_id = :new.album_id;
    IF stock_tr > 0 THEN
        update album set stock = (select stock from album where album_id = :new.album_id) - 1 where album_id = :new.album_id;
    else RAISE_APPLICATION_ERROR( -20001,
                                'stock must be available' );
    END IF;
END;
```

Explanation of sell_album_trigger: If album stock is bigger than 0, the album is available and it can be sold, otherwise raise an error.

- Trigger 4: sell_single_trigger



```
create or replace TRIGGER sell_single
  BEFORE INSERT ON single_sale
  FOR EACH ROW
  declare
  stock_tr NUMBER;
BEGIN
    select stock into stock_tr from singles where single_id = :new.single_id;
    IF stock_tr > 0 THEN
        update singles set stock = (select stock from singles where single_id = :new.single_id) - 1 where single_id = :new.single_id;
    else RAISE_APPLICATION_ERROR( -20001,
                                'stock must be available' );
    END IF;
END;
```

Explanation of sell_single_trigger: If single stock is bigger than 0, the single is available and it can be sold, otherwise raise an error.