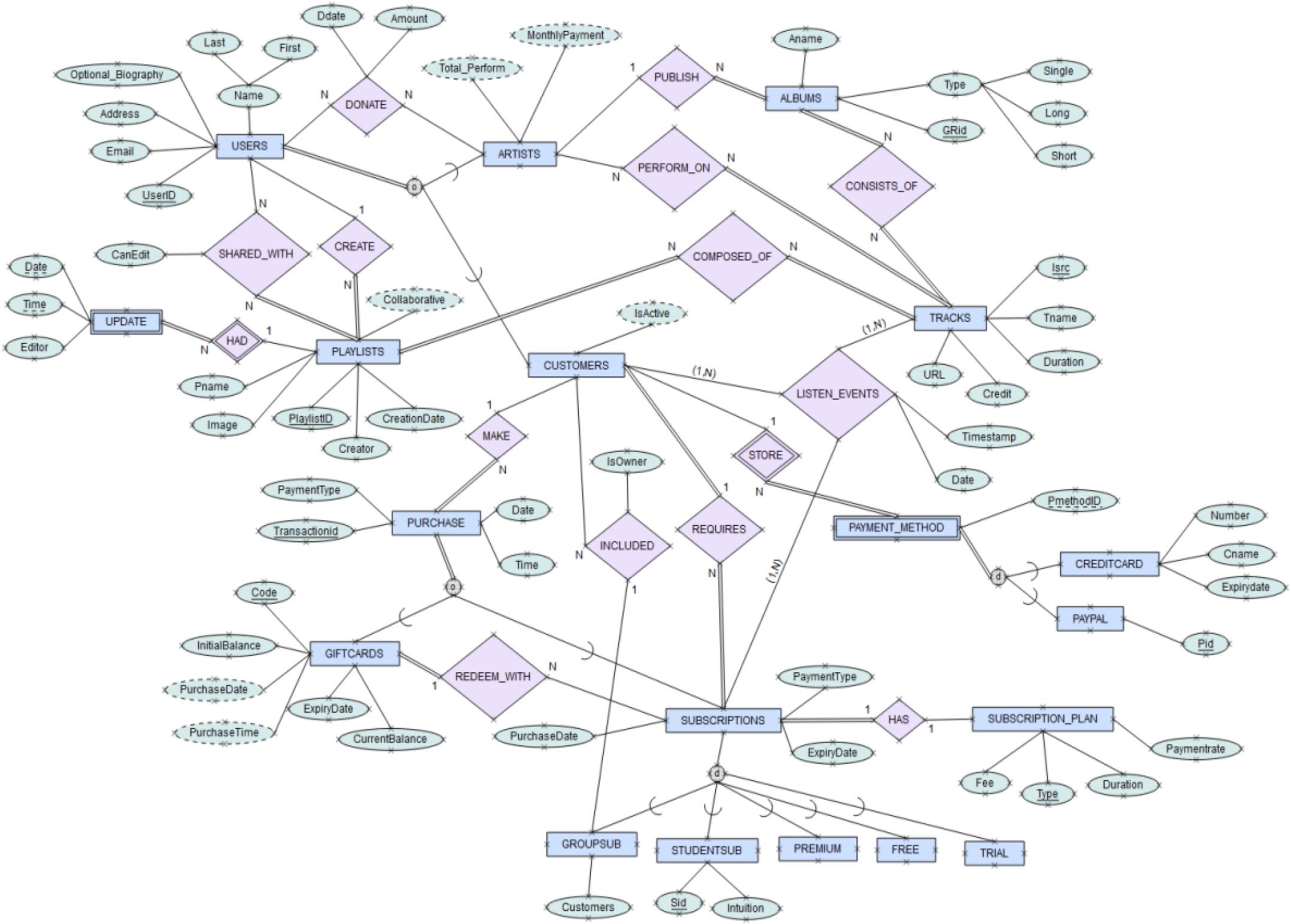
Ddate
Amount
MonthlyPayment
Aname
Last
First
Total_Perform
PUBLISH
Optional_Biography
Name
DONATE
ALBUMS
Type
Single
Long
Short
GRid
Address
Email
USERS
ARTISTS
PERFORM_ON
CONSISTS_OF
UserID
SHARED_WITH
CREATE
CanEdit
COMPOSED_OF
Isrc
Tname
Duration
Date
Time
Editor
UPDATE
HAD
Collaborative
PLAYLISTS
IsActive
TRACKS
URL
Credit
Pname
Image
PlaylistID
CreationDate
Creator
CUSTOMERS
LISTEN_EVENTS
Timestamp
Date
MAKE
STORE
IsOwner
PaymentType
Transactionid
PURCHASE
Date
Time
INCLUDED
REQUIRES
PmethodID
PAYMENT_METHOD
Number
Cname
Expirydate
CREDITCARD
PAYPAL
Pid
Code
InitialBalance
PurchaseDate
PurchaseTime
GIFTCARDS
ExpiryDate
CurrentBalance
REDEEM_WITH
PurchaseDate
SUBSCRIPTIONS
PaymentType
ExpiryDate
HAS
SUBSCRIPTION_PLAN
Paymentrate
Fee
Type
Duration
GROUPSUB
STUDENTSUB
PREMIUM
FREE
TRIAL
Customers
Sid
Intuition

Users

- Each user has a name (first and last), address, email and an optional biography.
- A user must be an artist and/or a customer.
- Each customer requires a single active subscription to stream music.
- Users can create playlists.
- Users may donate money to artists and the amount donated and date of donation are recorded.

Artists

- An artist can publish albums.
- An artist can perform on many tracks including those not published by them.
- Artists receive monthly payments. The amount earned each month is based on the total listen events for all tracks the artist performs on. This amount is calculated from the number of listen events and the payment rate of the customer's subscription associated with each listen event.

Albums

- An album has a name, GRid (Global Release Identifier) and a type.
- Album types include single, long play and short play.
- An album consists of one or more tracks and a track may appear on more than one album.

Tracks

- Each track has a unique ISRC (International Standard Recording Code), name, duration, credit(s) (writer credit, composer credit, producer credit, etc.), file URL and listen events (each with a date and timestamp).
- A listen event is created each time a customer listens to a track.

Playlists

- A playlist has a name, image, creator and creation date.
- A playlist is composed of one or more tracks.

- Playlists can be shared with other users. Users that have access to a shared playlist may be given permission to edit the playlist. A playlist that has more than one editor is known as a collaborative playlist.
- For each update made to a playlist, the date, time and editor are known.

Purchases

- Customers can purchase subscriptions and/or gift cards.
- For each purchase, the transaction id, the date and time of the purchase and the payment type are known.
- When making a payment, customers may choose to save their credit card or PayPal details for quicker purchases next time. A customer may save multiple credit cards.
- For credit cards, the card number, name and expiry date are saved.
- For PayPal, a PayPal ID is stored.

Subscriptions

- A subscription plan has a type (free, premium, group, student and trial), fee, duration and a payment rate for listen events.
- For each subscription, the purchase date, expiry, owner (the purchaser) and details of the subscription plan are known.
- Student subscriptions require a student ID and an institution.
- A group subscription can include up to 5 customers and is purchased by a single customer. Customers can be added to a group plan by the owner of the subscription. A subscription can be paid for via credit card, PayPal or a gift card.
- Customer subscription history is kept.

Gift cards

- A gift card has a unique code, initial balance and date and time of purchase. The expiry and current balance are also known.
- A valid (non-expired) gift card may be redeemed to fully purchase one or more subscriptions.

Final EER Diagram Report

This EER diagram models the core entities and relationships of a music streaming service, focusing on how users interact with content and subscription plans. It represents various actors such as users, artists, and customers, as well as important activities including listening to music, creating playlists, making purchases, and redeeming gift cards.

The CUSTOMERS entity includes a derived attribute IsActive, which is calculated based on the associated subscription's PurchaseDate and ExpiryDate. This attribute indicates whether a customer currently holds an active subscription, helping to avoid storing redundant data. Similarly, for ARTISTS, a derived attribute Total_Perform is calculated based on the many-to-many PERFORM_ON relationship with tracks, giving insight into each artist's activity on the platform.

The LISTEN_EVENTS relationship captures listening behavior and connects customers, tracks, and subscriptions. By analyzing this data, it is possible to calculate user-level metrics such as monthly listening time or engagement, which could further support insights into content recommendation and billing models.

Subscription payments are handled carefully. While a general PAYMENT_METHOD entity was created to represent common methods like credit cards and PayPal, gift cards are modeled separately. This decision was made to reflect their unique role: gift cards can be redeemed for one or more subscriptions using the REDEEM_WITH relationship. The GIFTCARDS entity tracks important attributes like Code, CurrentBalance, and ExpiryDate, ensuring that only valid cards can be used for purchases. Notably, gift cards are not included in the PAYMENT_METHOD generalization to avoid confusion, as they do not follow the same payment flow.

The playlist-sharing feature is implemented through the SHARED_WITH relationship. Each shared instance includes a CanEdit attribute, allowing for permission control. Rather than storing an explicit “collaborative” flag within the playlist entity, this is treated as a derived attribute, determined by whether multiple users have editing rights. This design simplifies the schema while maintaining flexibility.

Group subscriptions are managed using a many-to-one (N:1) relationship between CUSTOMERS and GROUPSUB, with a cardinality constraint of a maximum of five customers per group. Additionally, student subscriptions are captured through a STUDENTSUB specialization, which stores institution and student ID details.

Finally, the SUBSCRIPTIONS entity omits an explicit owner attribute, since ownership can be inferred from the existing 1:N REQUIRES relationship with CUSTOMERS. This design choice helps reduce redundancy and ensures consistency in identifying subscription owners.

Overall, the model prioritizes clarity, normalization, and accurate representation of complex business rules while minimizing redundancy through well-placed derived attributes and structured relationships.