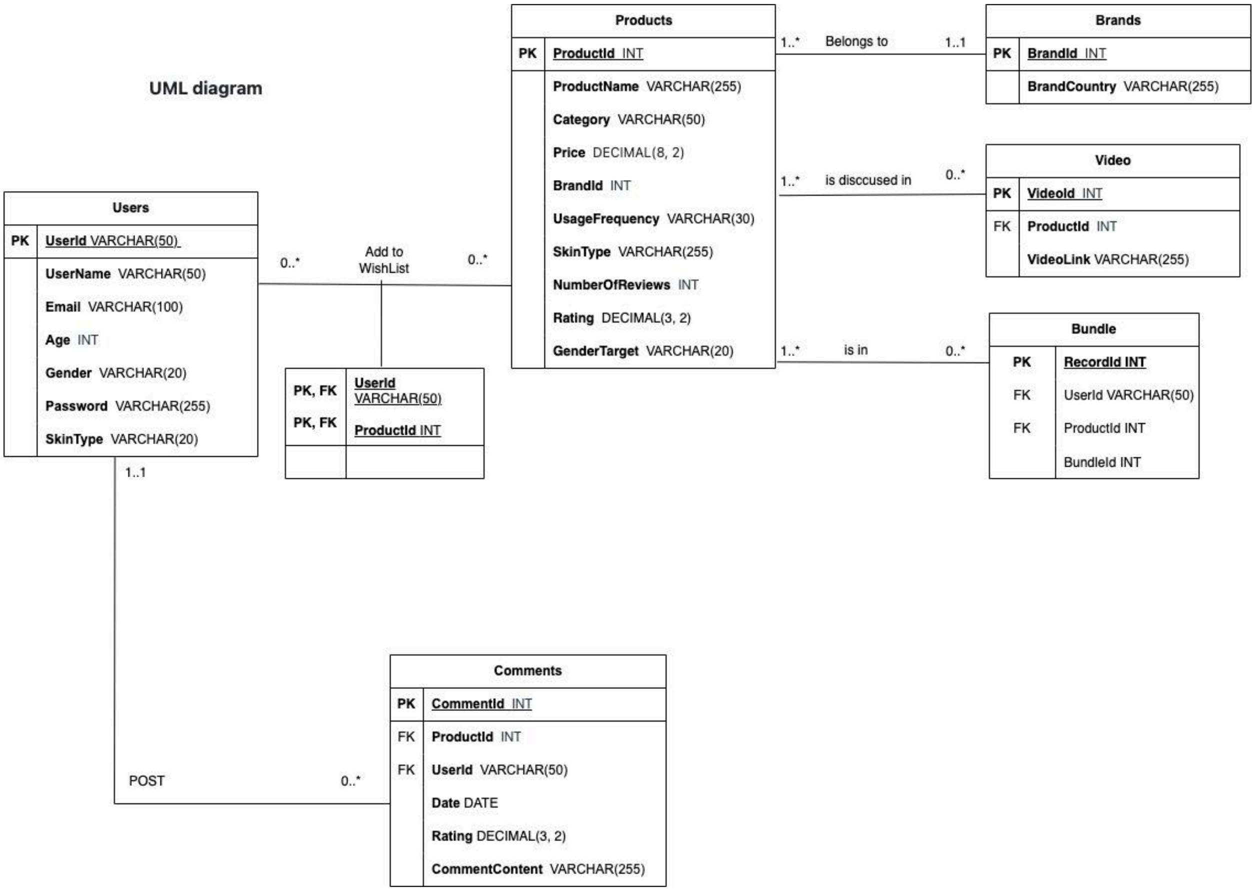
UML diagram

Revised the graph to uml graph, and changed the wishlist item to a relation between users and products.

BCNF Normalization

Our schema already appears to be in BCNF because all attributes depend only on the primary key, and no partial or transitive dependencies exist.

Users Table

- Primary Key: UserId
- Attributes: UserName, Email, Age, Gender, Password, SkinType

There is a clear primary key (`UserId`), and all other attributes are functionally dependent on this key. No transitive dependencies exist.

Products Table

- Primary Key: ProductId

- Attributes: ProductName, Category, Price, BrandId, UsageFrequency, SkinType, NumberOfReviews, Rating, GenderTarget

All attributes are functionally dependent on `ProductId`. No attribute is dependent on any non-key attribute.

Brands Table

- Primary Key: BrandId
- Attributes: BrandCountry

The attribute `BrandCountry` depends solely on the `BrandId`, which is the primary key.

Video Table

- Primary Key: VideoId
- Foreign Key: ProductId
- Attributes: VideoLink

The `VideoLink` depends directly on `VideoId`, and `ProductId` is included for referencing products.

WishListItem Table

- Primary Key: RecordId
- Foreign Keys: UserId, ProductId

This table represents a many-to-many relationship between users and products. There are no non-key attributes, and `RecordId` is the only candidate key.

Comments Table

- Primary Key: CommentId
- Foreign Keys: ProductId, UserId
- Attributes: Date, Rating, CommentContent

The `CommentContent`, `Date`, and `Rating` are functionally dependent on `CommentId`. No other dependencies are observed.

Bundle Table

- Primary Key: RecordId
- Foreign Keys: UserId, ProductId
- Attributes: BundleId

The `BundleId`, `UserId`, and `ProductId` are functionally dependent on `RecordId`. No other dependencies observed.

Entity Descriptions

User: Contains user login information and skin preferences.
- User ID: Uniquely identifies the user.
- userName: The username decided by the user.
- email: The user's email address.
- Age: The age of the user.
- Gender: The gender of the user.
- password: The password for the user's account.
- skin_type: The user's skin type (e.g., oily, dry, combination).

Products: Contains detailed information about each product.
- Pro ID: Uniquely identifies the product.
- Name: The name of the product.
- Category: The category to which the product belongs (e.g., skincare, makeup).
- Price: The price of the product.
- brand id: The ID of the brand producing the product.
- usage_frequency: Recommended usage frequency (e.g., daily, weekly).
- skin_type: The suitable skin type for the product.
- number of reviews: Number of user reviews for the product.
- rating: Average user rating for the product.
- gender_target: The gender the product is aimed at.

Brand: Specifies information about the brand.
- Brand ID: Uniquely identifies the brand.
- brand country: The country where the brand is based.

Video: Contains information about product-related videos.
- video id: Uniquely identifies the video.
- pro id: The product ID featured in the video.
- video link: The URL to the video.

WishList: A list containing products that the user has saved. It is a weak entity because it is uniquely identified by the combination of User ID and Product ID.
- User ID: Identifies the user who created the wishlist.
- product id: Identifies the product added to the wishlist.

Comment: Contains user feedback on products.
- comment ID: Uniquely identifies the comment.
- user ID: The ID of the user who posted the comment.
- date: The date the comment was posted.
- rating: The rating assigned to the product by the user.
- comment content: The textual content of the comment.
- product ID: The ID of the product the comment relates to.

Bundle: Represents a collection or grouping of products that a user creates.

- RecordId: This is the primary key that uniquely identifies each entry in the Bundle table.
-UserId: A foreign key referring to the user who created or interacted with the bundle. This links the bundle to a specific user in the Users table.
-ProductId: A foreign key linking to the products included in the bundle. Each product in a bundle is referenced from the Products table. -BundleId: This groups multiple products together under a unique bundle.

Relational Schema Translation

- Users(UserId: INT [PK], UserName: VARCHAR(50), Email: VARCHAR(100), Age: INT, Gender: VARCHAR(20), Password: VARCHAR(255), SkinType: VARCHAR(20))
- Products(ProductId: INT [PK], ProductName: VARCHAR(255), Category: VARCHAR(50), Price: DECIMAL(8,2), BrandId: INT [FK to Brands.BrandId], UsageFrequency: VARCHAR(30), SkinType: VARCHAR(255), NumberOfReviews: INT, Rating: DECIMAL(3,2), GenderTarget: VARCHAR(20))
- Brands(BrandId: INT [PK], BrandCountry: VARCHAR(255))
- Video(VideoId: INT [PK], ProductId: INT [FK to Products.ProductId], VideoLink: VARCHAR(255))
- WishListItem(RecordId: INT [PK], UserId: INT [FK to Users.UserId], ProductId: INT [FK to Products.ProductId])
- Comments(CommentId: INT [PK], ProductId: INT [FK to Products.ProductId], UserId: INT [FK to Users.UserId], Date: DATE, Rating: DECIMAL(3,2), CommentContent: VARCHAR(255))
- Bundle(RecordId: INT [PK], UserId: INT [FK to Users.UserId], ProductId: INT [FK to Products.ProductId], BundleId: INT)

Relationship Assumption
(* = any number of)
Each user can have 0 to * comments. Each comment must be owned by one user.
Each user adds 1 to * wish list item into wishlist. Each wish list item belongs to exactly one user.
Each wish list item is a product.
Each product belongs to one brand. Each brand can have 1 to * products.
Each product is discussed in 0 to * videos. Each video has 1 to * products.
Each product is in 0 to * bundles. Each bundle has 1 to * products.