**NailDB Data Dictionary**

Organized by Table, With Domain Context

* **Polish**
  + - **Polish\_ID** 
      * **Primary key;** an integer indentifier for each individual nail polish shade in the database.
      * Method of increment TBD.
    - ShadeName
      * The name of the shade, as stylized by the brand.
      * Varchar(35) to account for long polish names.
    - Description
      * A quick description of the nail polish. May mention color, texture, collection, ect.
      * Varchar(80)
    - UndertoneID
      * Identifier for the undertone of a nail polish.
      * INT
    - ToneID
      * Unique identifier for the tone of the polish. See Tone for more detail.
      * INT
    - OpacityID
      * Unique identifier for the opacity of a nail polish. See Opacity for more detail.
    - LineID
      * Unique identifier for the line that a nail polish may be on.
      * INT
    - Price
      * The price of the nail polish, in USD. Can be NULL.
      * Decimal(5, 2)
    - BrandID
      * Foreign key.
      * Identifies the brand associated with the polish.
      * INT
    - EventID
      * Foreign key
      * Identifies the event associated with the polish. Can be NULL.
      * INT
    - CollectionID
      * Foreign key
      * Identifies the collection associated with the polish. Can be NULL.
      * INT
    - IsDriesMatte
      * Identifies whether or not a nail polish is engineered to dry down matte on its own.
      * Boolean
    - IsGlowitheDark
      * Identifies whether or not a nail polish is engineered to glow in the dark.
      * Boolean.
    - IsUV
      * Identifies whether or not a nail polish is engineered to glow underneath ultraviolet lighting.
      * Boolean
    - IsTopper
      * Identifies whether or not a nail polish was intended to be used as a topper shade.
      * Boolean
    - IsThermal
      * Identifies whether or not a nail polish has thermal properties.
      * Boolean
* Undertone
  + **UndertoneID**
    - Unique identifier for the undertone of a nail polish.
    - INT
  + UndertoneName
    - The name for the overall warmth of the nail polish.
    - Varchar(12)
  + UndertoneDescription
    - The description of an undertone.
    - Varchar(80)
* Tone
  + **ToneID**
    - A unique identifier for the tone of a nail polish.
    - INT
  + ToneName
    - The name for the lightness of the nail polish.
    - Varchar(8)
  + ToneDescription
    - The description of the tone for a nail polish.
    - Varchar(80)
* Opacity
  + **OpacityID**
    - Unique identifier for the opacity for a nail polish.
    - INT
  + OpacityName
    - The name for the class of opacity.
    - Varchar(12)
  + OpacityDescription
    - The description for an opacity.
    - Varchar(80)
* PolishColorRole
  + **PolishColorID** 
    - An ID that uniquely identifies the colors a nail polish may have. Accounts for the idea that a single polish can have multiple colors with different purposes
    - INT
  + PolishID
    - Foreign key
    - Identifies the polish that may have that color and role combination. Accounts for the idea that a single polish can have multiple colors with different purposes.
    - INT
  + ColorID
    - Foreign Key
    - The color or one of the colors, of a nail polish. Color data is stored in the Color table.
    - INT
  + RoleID
    - Foreign Key
    - The identifying role that a color in a nail polish may hold. See the Role table for more information.
    - INT
* Role:
  + **RoleID** 
    - An integer key that identifies the specific role and description for a nail polish.
    - **INT**
  + RoleName
    - The role that a nail polish color may be part of. Examples include:
      * A nail polish with a red base and green shimmer particles.
      * A nail polish with a black base and red shimmer particles.
      * A nail polish with a pink base, blue shimmer particles, purple flakes, and silver magnetic particles.
      * A nail polish with a transparent base, and blue AND green flakes.
    - Varchar(25)
  + RoleDescription
    - The description for a specific role that a color may have in a nail polish.
    - Varchar(80)
* Color
  + **ColorID**
    - An individual identifier for each color in the database.
    - ColorName
      * The name of the color.
      * Varchar(25)
    - ColorDescription
      * The description of the color, including the forms that it can take.
      * Varchar(80)
    - ColorFamily
      * The family that a color may be in (warm, cool, neutral).
      * Varchar(12)
* Brand
  + - **BrandID**
      * The unique identifier for a nail polish brand.
      * INT
    - BrandName
      * The name of the nail polish brand.
      * Varchar(25)
    - Country
      * The country that a nail polish brand is based.
      * Varchar(20)
    - BrandCategoryID
      * Unique identifier for the type of brand a brand is. See BrandCategory table for more details.
      * INT
    - Parent Company
      * The parent company of a nail polish brand. Can be null.
      * Varchar(25)
    - FoundingYear
      * The year that a nail polish brand was founded (according to its most recent alias, if applicable)
      * INT
    - OperatingCity
      * The city in which a nail polish brand may operate, if known.
      * Varchar(25)
    - OperatingRegion
      * The region where a nail polish brand may operate. Can be a state, providence, or otherwise.
      * Varchar(25)
    - WebsiteURL
      * An absolute link to the website for a brand.
      * Varchar(2083)
* Line
  + **LineID**
    - A unique identifier for the line that a polish is part of. Most polishes are not part of a line. However, all polishes will be required to identify a line such that bottle size may be obtained.
      * Examples include:
        + Sally Hansen – Color Therapy, Xtreme Wear, Gel Finish
        + Essie – Expressie, Gel Couture
        + ILNP – Studio Color
    - INT
  + LineName
    - The name of the specific polish line.
    - Varchar(25)
  + LineDescription
    - The description of the line, including properties, notes, and other interesting facts.
    - Varchar(80)
  + BottleSize
    - The reported amount of liquid polish (in ml) that a bottle holds. May be inconsistent with actual average fill quantities.
    - Decimal(4, 1)
* BrandCategory
  + **BrandCategoryID**
    - A unique identifier for the type of brand.
    - INT
  + BrandCategoryName
    - Holds the name of the type of category a brand may be. Usually limited to: mainstream, boutique, luxury, indie.
    - Varchar(12)
  + BrandCategoryDescription
    - A description of what makes a brand ascribe to said category.
    - Varchar(80).
* Alias
  + **AliasID**
    - An integer that stores the unique identifier for a specific brand alias.
    - INT
  + BrandID
    - Foreign key
    - The brand that has had the alias.
    - INT
  + AliasName
    - The name of the alias.
    - Varchar(25)
  + StartDate
    - The date in which a brand began using their alias. May be NULL.
    - date
  + EndDate
    - The last day that a brand began using their alias. May be NULL.
    - date
  + AliasDescription
    - A description of the alias, including motivations, context, or otherwise.
    - Varchar(80)
* Collection
  + **CollectionID**
    - An integer identifier for the collection that a brand enacts.
    - INT
  + BrandID
    - Foreign Key
    - Identifies the brand that is associated with the collection.
  + CollectionName
    - The name of the collection.
    - Varchar(35)
  + Season
    - The season in which a collection may be released. Can take values ‘spring’ ‘summer’ ‘fall’ ‘winter’
    - Varchar(12)
  + CollectionYear
    - The year in which a collection is released.
    - INT
    - Varchar(25)
  + IsInfluencerCollaboration
    - Identifies whether or not a collection may be in collaboration with an online personality or influencer. Examples include:
      * Holo Taco x Safiya Nyagaard
      * Starrily x Kelli Marissa
    - Boolean
  + InfluencerName
    - Identifies the name of the influencer. May be NULL.
    - Varchar(25)
  + IsIPCollaboration
    - Identifies whether or not a collection may be part of an IP collab, or with another brand. Examples include:
      * OPI x Hello Kitty
      * Mooncat x Star Wars
      * Sally Hansen x Jelly Belly
      * Holo Taco x David’s Tea
  + IPName
    - Identifies the name of the collaboration. Can be NULL.
    - Varchar(25)
  + IsCharity
    - Identifies whether or not a nail polish is released in collaboration with a charity.
    - Boolean
  + CharityName
    - The name of the charity that the nail polish is raising funds for. May be NULL.
    - Varchar(25)
  + ThemeID
    - Foreign Key
    - Identifier for the theme of the collection. See Theme table for more details.
    - INT
* Finish
  + **FinishID**
    - A unique identifier for the primary finish. V
    - INT
  + FinishName
    - The name of the finish. Can take values ‘crème’ ‘shimmer’ ‘jelly’ ‘flake’ ‘crelly’ ‘glitter’
    - Varchar(12)
  + FinishDescription
    - A description of said finish.
    - Varchar(80)
* FinishDetail:
  + **FinishDetailID**
    - A unique identifier for each additional subcategory of primary finishes.
    - INT
  + FinishID
    - Foreign Key
    - Holds the corresponding finish that the subcategory is part of.
    - INT
  + DetailName
    - The name of the subcategory of finish. Examples include, but are not limited to:
      * Shimmer polishes can be:
        + Ultrafine.
        + Fine.
        + Medium
        + Linear Holographic
        + Scattered Holographic
      * Flake polishes can be:
        + Small.
        + Medium
        + Large
        + Metallic
        + Holographic
    - Varchar(25)
  + DetailDescription
    - A further description of the specific subcategory of finish.
    - Varchar(80)
* PolishFinish
  + **PolishFinishID**
    - A unique identifier that identifies each unique PolishFinish for each individual nail polish.
    - INT
  + PolishID
    - Foreign key
    - Identifies the corresponding nail polish that such a PolishFinish may belong to, accounting for possibly duplicate PolishFinish combinations. Attempts to account for the fact that one polish may have several finishes.
      * Example: a nail polish with blue fine shimmer particles, suspended in a purple jelly base, with green magnetic particles, that glows in the dark.
    - INT
  + FinishID
    - Foreign key
    - Identifies the finish, or one of the finishes, a nail polish may have.
    - INT
  + FinishDetailID
    - Foreign key
    - Identifies the subcategory of finish a nail polish may have, if applicable (can be NULL).
      * For example: a crème finish is a crème finish and cannot be categorized further. However, a shimmer finish can have many unique shimmer subtypes.
    - INT
* Event
  + **EventID**
    - Identifies the unique event that a nail polish may be part of.
    - INT
  + EventName
    - The name of the event that a nail polish belongs to. *Event* is a very specific category:
      * An *event* is defined as a period of release such several unrelated polish makers may release an exclusive, themed polish all at the same time. Events:
        + May only have one or two polish shade released per brand.
        + May have several brands that participate in the event.
        + Are usually organized by a group of individuals that do not make nail polish themselves (a Facebook group, a Discord server, ect).
        + Polishes released during said event have very limited availability.
      * Examples of events include Polish Pick Up and Hella Handmade
    - Varchar(25)
  + EventDescription
    - A description of the event, including themes, context, or otherwise important information.
    - Varchar(80)
  + StartDate
    - The start of an event. Can be NULL if unknown.
    - Date
  + EndDate
    - The end of an event. Can be NULL if unknown.
    - Date
  + OrganizerID
    - Foreign key
    - The organizing group that may coordinate such an event.
    - INT
  + ThemeID
    - Foreign key
    - Identifier for the specific theme of the event. See Theme table.
    - INT
* Theme
  + **ThemeID**
    - Unique identifier for the theme that an event or a collection may have.
    - INT
  + ThemeName
    - The name of theme of an event or collection. Not all events or collections may be themed (although this is rare).
    - Examples include:
      * OPI Downtown LA Collection – Themed around the colors in Los Angeles, CA.
      * Cirque Colors Surfer’s Crush – Themed around a day at the beach.
      * Bee’s Knees Lacquer Elden Ring – Themed around the role-playing action game, Elden Ring.
      * Polish Pick Up Tropical Forest May 2025 – Several nail polish brands released a single themed polish on the Polish Pick Up website, themed around things found in a tropical forest.
    - Varchar(35)
  + ThemeDescription
    - The description of a theme, including what may count as part of a theme.
    - Varchar(80)
* Organizer
  + OrganizerID
    - An identifying integer that corresponds to an organizer for a nail polish event, or set of events.
    - INT
  + OrganizerName
    - The name for a nail polish organizer that is responsible for conducting events.
    - Varchar(25)
  + FoundingYear
    - The year in which an organizer first was established.
    - INT
  + OrganizerDescription
    - A description of the organizers behind an event. Key details, names, locations, ect.
    - Varchar(80)
* Image
  + **ImageID**
    - A unique identifier that identifies an image that goes with a nail polish.
    - INT
  + PolishID
    - Foreign key
    - The ID of the polish that an uploaded image corresponds to.Can be null.
    - INT
  + BrandID
    - Foreign key
    - The ID of the brand that an uploaded image corresponds to. Can be null.
    - INT
  + FilePath
    - Contains the absolute path to the appropriate image for the nail polish.
    - Varchar(255)
  + UploadDate
    - The date in which an image is uploaded to the database.
    - Date
  + ShotDate
    - The date in which an image is shot using a camera, if applicable. Can be NULL (e.g. a random user takes an image and does not have shot data).
    - Date
  + UserID
    - (Eventual) Foreign key
    - Unique identifier for the user that uploads the image.
    - INT
  + ApprovalStatus
    - States the approval status for an image (whether or not it has been successfully uploaded to the database).
    - Varchar(12)
* User
  + **UserID**
    - Unique identifier for a registered user for the database.
    - INT
  + UserName
    - The login name for a user—will currently correspond with display name as well.
    - Varchar(60)
  + UserPass
    - HASHED
    - The password for a User, such that they can log in.
    - Varchar(255)
  + UserEmail
    - The registered email for a user.
    - Varchar(40)
  + UserRegistrationDate
    - A datetime that logs the moment of registration for a user.
    - Datetime
  + UserRoleID
    - Defines the level of role that a user may have in the database. (I.e. normal, moderator, administrator).
    - INT
* Group
  + **GroupID**
    - Unique identifier for the group that a user may be part of.
    - INT
  + GroupName
    - The name for the group a user may be part of.
    - Varchar(12)
  + GroupDescription
    - The description for the group, including superficial information about permissions, qualifications, ect.
    - Varchar(80)
* UserGroup
  + **UserGroupID**
    - Unique identifier for the specific joint combination of a user, and their group identification. Accounts for the idea that one user can have multiple group memberships (regular, moderator/editor, admin, ect).
    - INT
  + UserID
    - Foreign key
    - Identifier for a user.
    - INT
  + GroupID
    - Foreign key
    - Identifier for the group.