

Stage 2: Database Design

Team number: 33

Team name: TT

Team member:

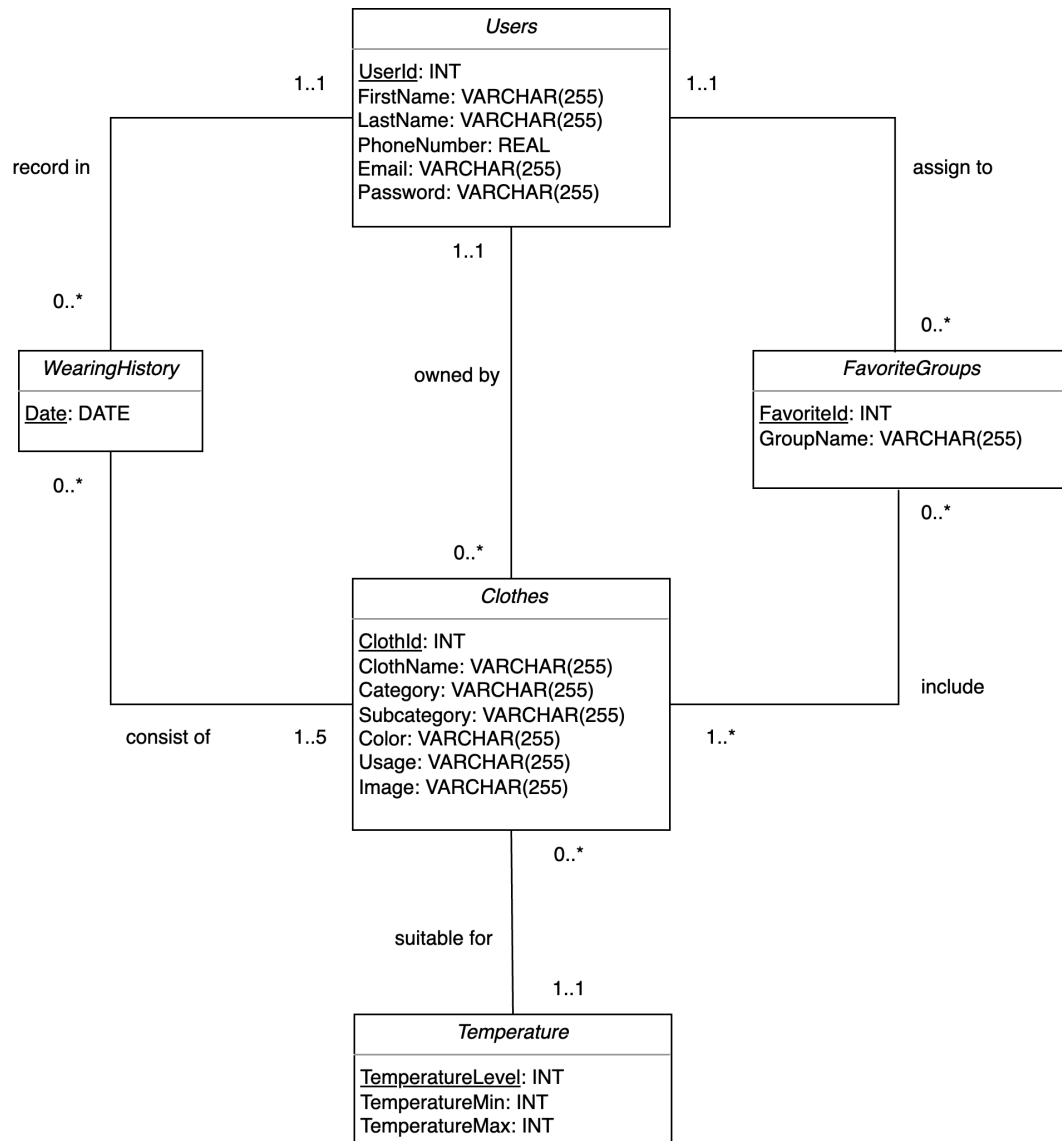
Supawit Sutthiboriban (supawit3)

Yu-Ting Cheng (ytcheng4)

Yu-Chien Lin (yuchien4)

Han-Chih Chang (hanchih2)

1. UML Diagram



2. Relational Schema

Users(

UserId: INT [PK],
FirstName: VARCHAR(255),
LastName: VARCHAR(255),
PhoneNumber: REAL,
Email: VARCHAR(255),
Password: VARCHAR(255)

)

Clothes(

ClothId: INT [PK],
UserId: INT [FK to Users.UserId],
ClothName: VARCHAR(255),
Category: VARCHAR(255),
Subcategory: VARCHAR(255),
Color: VARCHAR(255),
Usage: VARCHAR(255),
Image: VARCHAR(255),
TemperatureLevel: INT [FK to Temperature.TemperatureLevel],
LatestWearTwoWeeks: DATE [FK to WearingHistory.Date]

)

FavoriteGroups(

Favoriteld: INT [PK],
UserId: INT [FK to Users.UserId],
GroupName: VARCHAR(255)

)

Include(

Favoriteld: INT [FK to FavoriteGroups.Favoriteld],
ClothId: INT [FK to Clothes.ClothId]

)

Note: this table comes from the relationship between FavoriteGroups entity and Clothes entity.

WearingHistory(

Date: DATE,
UserId: INT [FK to Users.UserId],
Cloth1: INT [FK to Clothes.ClothId],
Cloth2: INT [FK to Clothes.ClothId],
Cloth3: INT [FK to Clothes.ClothId],
Cloth4: INT [FK to Clothes.ClothId],
Cloth5: INT [FK to Clothes.ClothId]

)

```
Temperature(  
    TemperatureLevel: INT [PK],  
    TemperatureMin: INT,  
    TemperatureMax: INT  
)
```

3. Normalization (BCNF or 3NF)

This database schema is normalized in 3NF. The following shows how every table in our schema is 3NF:

1. Users:

UserId -> FirstName, LastName, PhoneNumber, Email, Password

Explanation:

This functional dependency indicates that the UserId can uniquely determine the values of the rest of the table.

Email -> UserId, FirstName, LastName, PhoneNumber, Password

Explanation:

This functional dependency indicates that the Email can uniquely determine the values of the rest of the table.

This means both UserId and Email are candidate keys

Since every functional dependency in the table has its left-hand side as a superkey, this means the table is already in 3NF.

2. Clothes:

ClothId -> UserId, ClothName, Category, Subcategory, Color, Usage, Image, TemperatureLevel, LatestWearTwoWeeks

Explanation: This functional dependency indicates that the ClothId uniquely determines the values of the rest of the table. We are making the assumption that only ClothId can be the unique identifier of the table and other attributes cannot (e.g. ClothName might not be able to identify a unique row of the table, there's a probability that it could identify multiple rows).

Since every functional dependency in the table has its left-hand side as a superkey, this means the table is already in 3NF.

3. FavoriteGroups:

FavoriteId -> UserId, GroupName

Explanation:

This functional dependency specifies that each FavoriteId uniquely determines a specific UserId and GroupName. We are making the assumption that only FavoriteId can be the unique identifier of the table and other attributes cannot (e.g. The GroupName and UserId might not be able to identify a unique row of the table, since different users use the same GroupName for their favorite groups, and one user can have multiple favorite groups).

Since every functional dependency in the table has its left-hand side as a superkey, this means the table is already in 3NF.

4. WearingHistory:

Date, UserId -> Cloth1, Cloth2, Cloth3, Cloth4, Cloth5

Explanation: This functional dependency specifies that for each unique combination of Date and UserId, there is a single corresponding set of Cloth1, Cloth2, Cloth3, Cloth4, Cloth5. In other words, the combination of Date and UserId uniquely determines the clothing items for that day.

Since every functional dependency in the table has its left-hand side as a superkey, this means the table is already in 3NF.

5. Temperature

TemperatureLevel -> TemperatureMax, TemperatureMin

Explanation: This functional dependency specifies that each TemperatureLevel uniquely identifies a maximum and minimum temperature pair.

Thus, all of our tables are normalized in 3NF.

4. Description and Assumption of Relations

We have tables: Users, Clothes, FavoriteGroups, WearingHistory, Temperature.

- **Assumption:**

Users:

1. We assume that one user has zero to many clothing items.
2. We assume that one user has zero to many favorite groups.
3. We assume that one user has zero to many wearing history.

Clothes:

1. We assume that one clothing item belongs to one user. Even if two users have exactly the same clothes, as long as they are different users, the clothId of the clothing item will be different.
2. We assume that one clothing item can be included in zero to many favorite groups. A clothing item can be categorized into zero to many favorite groups. For example, a clothing item with clothId=1 can be classified into favorite groups like "cute" and "bright", while a clothing item with clothId=2 might not be classified into any favorite group at all.
3. We assume that one clothing item has zero to many WearingHistory. A clothing item can be worn on zero to many dates in WearingHistory.
4. We assume that one clothing item is suitable at only one temperature level.

FavoriteGroups:

1. We assume that one favorite group includes one to many clothing items.
2. We assume that one favorite group belongs to one user.

WearingHistory:

1. We assume that one wearing history belongs to one user.
2. We assume that one wearing history includes one to five clothing items. It is because one outfit of the day can be created only with at least one and up to five clothing items.

Temperature:

1. We assume that one temperature level is suitable for zero to many clothing items.

● **Description of relations:**

1. Users & Clothes: One user can own **zero to many** clothing items and one clothing item belongs **to only one** user.
2. Users & FavoriteGroups: One user can have **zero to many** favorite groups and one favorite group is assigned **to one** user.
3. Users & WearingHistory: One user can record **zero to many** WearingHistory. Each WearingHistory can only be **recorded by one** user.
4. Clothes & FavoriteGroups: One clothing item can be included in **zero to many** favorite groups. And one favorite group can include **one to many** clothing items.
5. Clothes & WearingHistory: One WearingHistory consists of **one to five** clothing items, and one clothing item can be recorded in **zero to many** WearingHistory.
6. Clothes & Temperature: One clothing item can be suitable to wear at **one** temperature level, and one temperature level is suitable for **zero to many** clothing items.

● **Description of each entity:**

1. **Users:**

This is an entity for general user information as well as user authentication information.

1. UserId: a unique identifier to distinguish between users. This should be an integer and the primary key for this table.
2. FirstName: the first name of a user. This should be a varchar attribute.
3. LastName: the last name of a user. This should be a varchar attribute.
4. PhoneNumber: the phone number of a user. This should be a real attribute.
5. Email: the email address of a user for login to the platform. This should be a varchar attribute.
6. Password: the password of a user for login to the platform. This should be a varchar attribute.

2. Clothes:

This is an entity for the clothing information of each user.

1. ClothId: a unique identifier. This should be an integer and the primary key for this table.
2. UserId: UserId of the user who owns this clothing item. This should be a foreign key referencing Users.
3. ClothName: the name of the clothing item. This should be a varchar attribute.
4. Category: the main category of the clothing item including apparel, footwear, and accessories. this should be a varchar attribute.
5. Subcategory: the subcategory of the clothing item, the value depends on the main category, for example, subcategories under the main category Apparel include Topwear and BottomWear. this should be a varchar attribute.
6. Color: the main color of clothing item. This should be a varchar attribute.
7. Usage: the suitable occasions for the clothing item. This should be a varchar attribute.
8. Image: the url pointing to the image of the clothing item. This should be a varchar attribute.
9. TemperatureLevel: the suitable temperature level for the clothing item. This should be a foreign key referencing Temperature.
10. LatestWearTwoWeeks: This records the date this clothing item was worn in the past two weeks. This should be a foreign key referencing WearingHistory.

3. WearingHistory

This is an entity for the everyday outfit information of each user. It can store from one to five clothing items for each day.

1. Date: the date of the wearing history. This should be a date type and a part of the primary key for this table.

2. UserId: UserId of the user to whom this WearingHistory belongs. This should be a foreign key referencing Users.UserId and a part of the primary key for this table.
3. Cloth1: ClothId for the first clothing item in this WearingHistory. This should be a foreign key referencing Clothes.ClothId.
4. Cloth2: ClothId for the second clothing item in this WearingHistory. This should be a foreign key referencing Clothes.ClothId.
5. Cloth3: ClothId for the third clothing item in this WearingHistory. This should be a foreign key referencing Clothes.ClothId.
6. Cloth4: ClothId for the fourth clothing item in this WearingHistory. This should be a foreign key referencing Clothes.ClothId.
7. Cloth5: ClothId for the fifth clothing item in this WearingHistory. This should be a foreign key referencing Clothes.ClothId.

The tuple of (Date, UserId) is the primary key of this entity.

4. FavoriteGroups

This is an entity for the information of favorite groups for each user.

1. FavoriteId: a unique identifier. This should be an integer and the primary key for this table.
2. UserId: UserId of the user to whom this FavoriteGroup belongs. This should be a foreign key referencing Users.UserId.
3. GroupName: the name of a favorite group. This should be a varchar attribute.

5. Temperature

This entity contains the information for each temperature level.

1. TemperatureLevel: a unique identifier. This should be an integer and the primary key for this table.
2. TemperatureMin: The minimum temperature value of the level. This should be an integer.
3. TemperatureMax: The maximum temperature value of the level. This should be an integer.