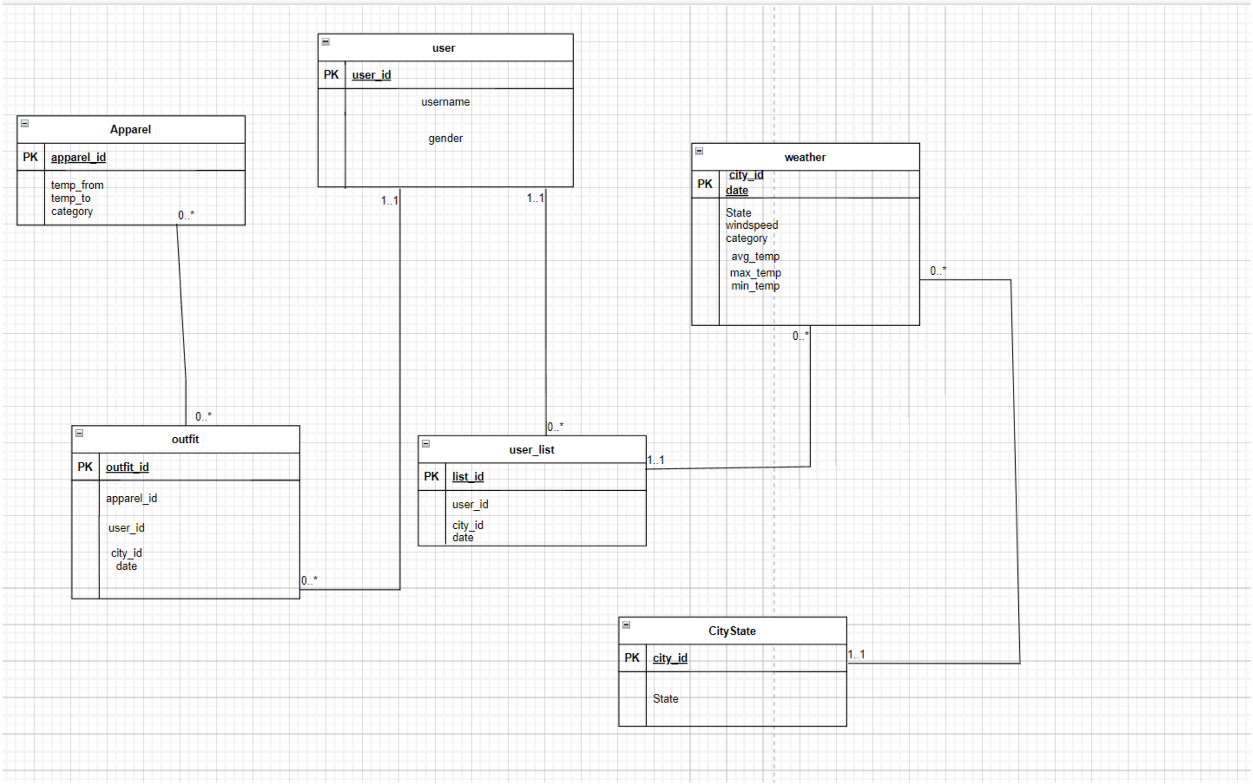

Normalization

Our schema is in 3NF. There are no transitive dependencies in our schema and all the attributes are dependent on the primary keys.
The functional dependencies of each relation are:
User: *userid->username,gender*
Apparel: *apparel_id->temp_from,temp_to,category*
Outfit: *outfit_id->apparel_id*
Originally we had a combined denormalized entity containing the user_id and the weather information Userlist(user_id,city_id,date,windspeed,category,state)
But there were transitive dependencies that didn’t depend on the primary key(user_id) like
city_id->state
city_id,date->windspeed,avg temp, max temp, min temp
To normalize this, we removed user data separately and added a user_list entity.
So weather data is in a separate table, and the user information mapping to the weather is in the user_list relation.
user_list:*list_id->user_id,city_id,date*
Weather:*city_id,date->windspeed,avg_temp,max_temp,min_temp*

city_id->state
Originally we had state information in the weather table.
Since this FD is preventing our relation from being normalized,
We create a new table with CityState information and remove state from the weather relation.
CityState:*city_id->state*

Relational schema

Entities:
User_list(list_id:INT [PK],
user_id: INT [FK to user.user_id],
city_id:INT[FK to weather.city_id])

User
(user_id:INT [PK],
username:Varchar(20),
gender:varchar(10))

Weather
(date:DATETIME[PK],
city_id:INT[FK to Citystate.city_id],
Windspeed: Varchar(20),
avg_temp:INT,
max_temp:INT,
min_temp:INT)

Apparel
(apparel_id:INT[PK] ,
temp_from:INT,
temp_to:INT,
category:Varchar(20）)

Outfit
(outfit_if:INT[PK],
apparel_id:INT [FK to Apparel.apparel_id],
user_id:INT[FK to user.user_id],
city_id:INT[FK to weather.city_id],
date:INT[FK to weather.date])

CityState
(city_id:INT[PK],
State: VARCHAR(50)
)

Many to many relationship:
Outfit_consists_of: (outfit_id:INT [PK][FK to outfit.outtfit_id], apparel_id:INT[PF] [FK to Apparel.apparel_id])

Entity description

User_list
Represents a list associated with each user that contains weather information for specific cities. Can be used to store cities for each user to bookmark different locations. These locations would be front and center in the user’s dashboard.

User
Represents a user in the system with a unique identifier, username, and gender information. Allows each user to have stored data that lets them cater their experiences for themselves.

Weather
Represents the weather information for a specific city on a particular date. Allows access to the weather conditions and related statistics.

Apparel
Represents various apparel items with temperature range suitability and category information. Allows users to create outfits based on the weather conditions. The various categories would be headwear, outerwear, footwear etc.

Outfit
Represents an outfit chosen by a user for a specific date and city. Incorporates apparel items and weather information. Users view different outfits that are available for the respective weather conditions for different locations. The records of outfit are generated using a stored procedure that determines apparel that is appropriate based on the weather conditions and a single piece of apparel from each category.

Relationship description (* = any number of)

- Each user can have 0 to * cities
- Each user can have 0 to * apparel
- Each user can have 0 to * outfits
- Each apparel can be in 0 to * outfits
- Each location can have one weather for 1 date
- Each outfit can have 0 to * apparel
- Each user can have 1 user_list
- Each state has 0 to * cities