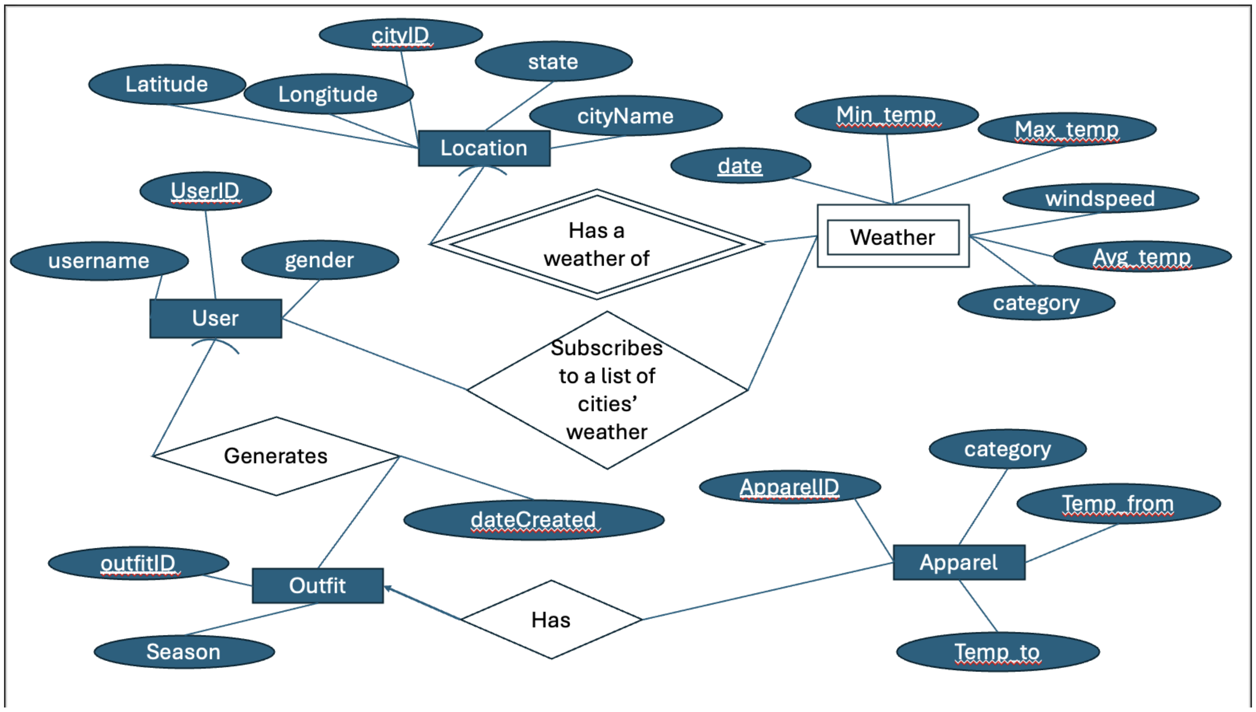

Normalization:

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->Season*
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 now we are using a m to m relationship to represent the user's list.
So weather data is in a separate table, and the user information mapping to the weather is in the user_subscribe_list relationship.

Weather:*city_id,date->windspeed,avg_temp,max_temp,min_temp,category*
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.
Location:City_Id->State,cityName,latitude,longitude

Relational schema

Entities:

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

Weather
(date:DATETIME[PK],
city_id:INT[FK to Citystate.city_id] [PK],
Category :Varchar(20),
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_id[FK to Outfit.outfit_id]:INT
)

Outfit
(outfit_id:INT[PK],
Season : Varchar(10),

Datecreated :DATETIME,
user_id:INT[FK to User.user_id])

Location
(City_ID: INT [PK],
State:Varchar(20),
City_name:Varchar(20),
Latitude:REAL,
Longitude:REAL)

N to M relationship:
User_subscribes_list:
(user_id:INT[FK to User.user_id][PK],
date:DATETIME[FK to Weather.date][PK],
city_id:INT[FK to Weather.city_id] [PK])

Entity description

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

Weather
Represents the weather information for a specific city on a particular date. Allows access to the weather conditions and related statistics. Weather is uniquely identified by the date and the CityID of the Location for that weather.

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.

Location

Is the list of City and the corresponding state and GPS coordinates. Uniquely identified by cityID. It is the supporting relationship for weather, since it's primary key is required to uniquely identify a row in the Weather table.

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 location can have one weather for 1 date
- Each outfit can have 0 to * apparel
- Each apparel is in at most 1 Outfit.
- Multiple users can subscribe to a list of weathers of various cityIDs.(many to many)

Changes for stage 2:
ER diagram remade completely:

Weather is a weak entity with a supporting relationship with Location since each weather is uniquely identified by the location of the weather. For eg. "Snowy" isn't unique since multiple locations can have the same snowy weather. So "Snowy","Urbana" is a unique combination.

Made user-list into a many to many relationship between user and weather, thus the schema remains the same but the relationship is more clearly defined.

Added more information to Location entity set.Now State is uniquely defined by City_Id and weather is uniquely defined by City_id and date.

Fixed inconsistencies between schema and ER diagram.