- User

  {
  UserID: String, required, unique, primary key, can be generated by database

  Nickname: String, required
  Email: String, required, unique
  Password: String, required, >= 8 characters
  Realname: String
  Avatar: URL String to the path in file system
  }

- Event

  {
  EventID: String, required, unique, primary key
  CreatorID: String, required, foreign key references User(UserID)
  CreatedTIme: TIme, required
  Title: String, required
  Location: String, required
  EventDate: Date, required
  Start time: Time, required
  End time: TIme, required
  Link to event page: String
  Food type: String
  Admittance (who can come): String
  OtherInfo: String
  }

- Event Comment

  {
  CommentID: String, required, unique, primary key
  EventID: String, required, foreign key references Event(EventID)
  PosterID: String, required, foreign key references User(UserID)
  PostTime: TIme, required
  Content: String, required
  }

- Event Watches

  {
  watchID: String, required, unique, primary key

EventID: String, required, foreign key references Event(EventID)
WatcherID: String, required, foreign key references User(UserID)
}

- Offer

{
OfferID: String, required, unique, primary key
CreatorID: String, required, foreign key references User(UserID)
isAnonymous: bool, required. if true, set “CreatorID” to a dummy ID associated with UserName “anonymous” and the default avatar
CreatedTIme: TIme, required
Description: String, required
Location: String, required
Date: Date, required
Start time: Time, required
End time: TIme, required
OtherInfo: String
}

- Offer Comment

{
CommentID: String, required, unique, primary key
OfferID: String, required, foreign key references Offer(OfferID)
PosterID: String, required, foreign key references User(UserID)
PostTime: TIme, required
Content: String, required
}

- Offer Watches

{
watchID: String, required, unique, primary key
offerID: String, required, foreign key references Event(EventID)
WatcherID: String, required, foreign key references User(UserID)
}

------- April 4 New Schema -------

- Post (for networking)

{
PostID: String, required, unique, primary key

CreatorID: String, required, foreign key references User(UserID)
isAnonymous: bool, required. if true, set “CreatorID” to a dummy ID associated with
CreatedTIme: TIme, required
Content: String
}

- Post Comment

{
CommentID: String, required, unique, primary key
PostID: String, required, foreign key references Post(PostID)
PosterID: String, required, foreign key references User(UserID)
PostTime: TIme, required
Content: String, required
}

- Feedback

{
FeedbackID: String, required, unique, primary key,
Poster Name: String,
Email: String,
Feedback: String
}