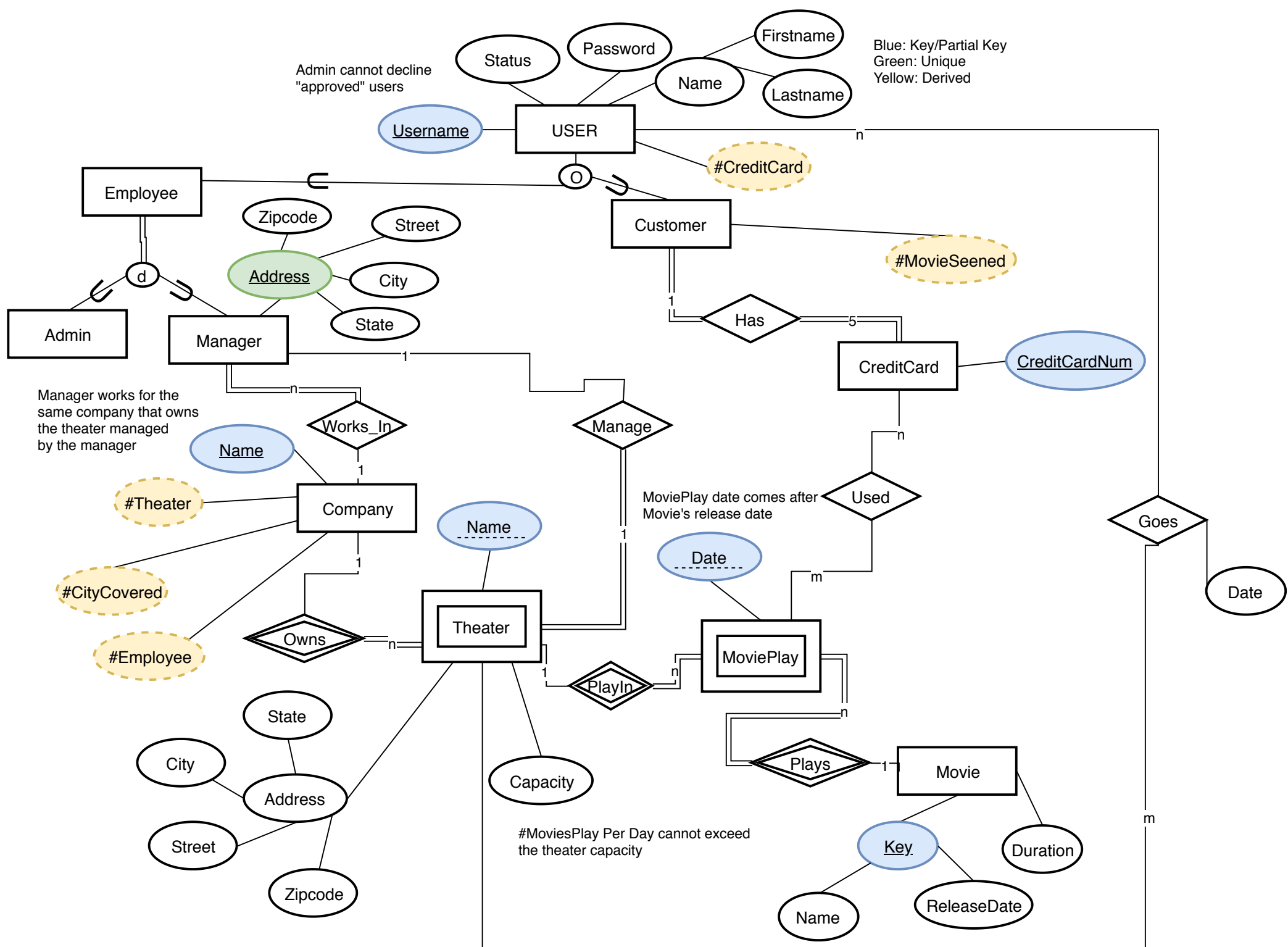# TEAM 44
Phase 2

Ibrahim Alzkeri (ialzkeri3)
Vishal Hansalia (vhansalia7)
Ryan Konopka (rkonopka3)
Hussain Raza (hraza8)
Amadia Tillman (atillman36)

This is an Entity-Relationship (ER) diagram.

Legend:
Blue: Key/Partial Key
Green: Unique
Yellow: Derived

USER entity attributes: Status, Password, Firstname, Name, Lastname, Username (key), #CreditCard (derived)

Note: Admin cannot decline "approved" users

Employee entity with specialization (disjoint, d): Admin, Manager

Manager attributes: Address (unique) with Zipcode, Street, City, State

Note: Manager works for the same company that owns the theater managed by the manager

USER relates to **Customer** via "O" (overlapping/optional) with cardinality n.
Customer attributes: #MovieSeened (derived)

Customer — **Has** (1 ... 5) — **CreditCard** (CreditCardNum key)

Manager — (1) — **Works_In** (n) — **Company** (Name key)

Manager — (1) — **Manage** — **Theater**

Company attributes: #Theater (derived), #CityCovered (derived), #Employee (derived)

Company — (1) — **Owns** (n) — **Theater**

CreditCard — (n) — **Used** (m) — **MoviePlay**

Note: MoviePlay date comes after Movie's release date

Theater attributes: Name (partial key), Capacity, Address with State, City, Street, Zipcode

Theater — (1) — **PlayIn** (n) — **MoviePlay**

MoviePlay attributes: Date (partial key)

MoviePlay — (n) — **Plays** (1) — **Movie**

Movie attributes: Key (key), Name, ReleaseDate, Duration

USER — (n) — **Goes** (m) — ... with Date attribute

Note: #MoviesPlay Per Day cannot exceed the theater capacity

User = (Username, Status, Password, Firstname, Lastname, UserType)

CreditCard = (CreditCardNum, Customer [fk1])
Fk1: Customer -> User.Username, Customer is not null

Movie = (Moviename, ReleaseDate, Duration)

Company = (Compname)

Manager = (Username, Compname [fk2], Street, City, State, Zipcode)
Fk2: Compname -> Company.Compname, Compname is not null

Theater = (Theatername, Compname [fk3], Manager [fk4], Street, City, State, Zipcode, Capacity)
Fk3: Compname -> Company.Compname, Compname is not null
Fk4: Manager -> Manager.Username, Manager is not null

MoviePlay = (Moviename, ReleaseDate [fk5], Theatername, Compname [fk6], Playdate)
Fk5: Moviename, ReleaseDate -> Movie.Moviename, Movie.ReleaseDate, Moviename is not null, ReleaseDate is not null
Fk6: Theatername, Compname -> Theater.Theatername, Theater.Compname, Theatername is not null, Compname is not null

MovieView = (Moviename, ReleaseDate, Theatername, Compname, Playdate [fk7], CreditCardNum [fk8])
Fk7: Moviename, ReleaseDate, Theatername, Compname, Playdate -> MoviePlay.Moviename, MoviePlay.ReleaseDate, MoviePlay.Theatername, MoviePlay.Compname, MoviePlay.Playdate, Moviename is not null, ReleaseDate is not null, Theatername is not null, Compname is not null, Playdate is not null
Fk8: CreditCardNum -> CreditCard.CreditCardNum, CreditCardNum is not null

TheaterVisit = (VisitID, Theatername, Compname [fk9], Username [fk10], Visitdate)
Fk9: Theatername, Compname -> Theater.Theatername, Theater.Compname, Theatername is not null, Compname is not null
Fk10: Username -> User.Username, Username is not null

```sql
DROP DATABASE IF EXISTS atlmovie;
CREATE DATABASE atlmovie;
USE atlmovie;

-- All users, including employees (admins and managers), customers,
and regular users
CREATE TABLE user (
  username VARCHAR(20) PRIMARY KEY NOT NULL,
  status ENUM ('Pending','Approved','Declined') NOT NULL DEFAULT
'Pending',
  password VARCHAR(20) NOT NULL,
  firstname VARCHAR(16) NOT NULL,
  lastname VARCHAR(16) NOT NULL,
  usertype ENUM ('User_Only', 'Customer_Only', 'Manager_Only',
'Manager_Customer', 'Admin_Only', 'Admin_Customer') NOT NULL
);

-- Customers have one to five credit cards
CREATE TABLE creditcard (
  creditcardnum CHAR(16) PRIMARY KEY NOT NULL,
  customer VARCHAR(20) NOT NULL,

  FOREIGN KEY (customer) REFERENCES user(username)
    ON UPDATE CASCADE ON DELETE CASCADE
);

-- Movie: name/releasedate combination is unique/primary key
CREATE TABLE movie (
  moviename VARCHAR(40) NOT NULL,
  releasedate DATE NOT NULL,
  duration INT NOT NULL,

  PRIMARY KEY (moviename,releasedate)
);

-- Company: just a list of unique company names, pre-populated
CREATE TABLE company(
  compname VARCHAR(20) PRIMARY KEY NOT NULL
);

-- Managers are a type of user with a unique address, must work for
one company, permitted to manage one theater
CREATE TABLE manager(
  username VARCHAR(20) PRIMARY KEY NOT NULL,
  compname VARCHAR(20) NOT NULL,
  street VARCHAR(30) NOT NULL,
  city VARCHAR(20) NOT NULL,
  state enum ('AL','AK','AZ','AR','CA','CO','CT','DE','FL','GA',
    'HI','ID','IL','IN','IA','KS','KY','LA','ME','MD',
    'MA','MI','MN','MS','MO','MT','NE','NV','NH','NJ',
    'NM','NY','NC','ND','OH','OK','OR','PA','RI','SC',
    'SD','TN','TX','UT','VT','VA','WA','WV','WI','WY') NOT NULL,
  zipcode CHAR(5) NOT NULL,

  UNIQUE (street, city, state, zipcode),
  FOREIGN KEY (compname) REFERENCES company(compname)
```

```sql
    ON UPDATE CASCADE ON DELETE RESTRICT
);

-- Theaters are run by one manager and belong to that manager's
company; have address, capacity
CREATE TABLE theater(
  theatername VARCHAR(20) NOT NULL,
  compname VARCHAR(20) NOT NULL,
  manager VARCHAR(20) NOT NULL,
  street VARCHAR(30) NOT NULL,
  city VARCHAR(20) NOT NULL,
  state ENUM ('AL','AK','AZ','AR','CA','CO','CT','DE','FL','GA',
    'HI','ID','IL','IN','IA','KS','KY','LA','ME','MD',
    'MA','MI','MN','MS','MO','MT','NE','NV','NH','NJ',
    'NM','NY','NC','ND','OH','OK','OR','PA','RI','SC',
    'SD','TN','TX','UT','VT','VA','WA','WV','WI','WY') NOT NULL,
  zipcode CHAR(5) NOT NULL,
  capacity INT NOT NULL,

  PRIMARY KEY (theatername,compname),
  UNIQUE (manager),
  FOREIGN KEY (compname) REFERENCES manager(compname)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  FOREIGN KEY (manager) REFERENCES manager(username)
    ON UPDATE CASCADE ON DELETE RESTRICT
);

-- Movie play is one showing of one movie at one theater on one date
CREATE TABLE movieplay(
  moviename VARCHAR(40) NOT NULL,
  releasedate DATE NOT NULL,
  theatername VARCHAR(20) NOT NULL,
  compname VARCHAR(20) NOT NULL,
  playdate DATE NOT NULL,

  PRIMARY KEY (moviename, releasedate, theatername, compname,
playdate),
  FOREIGN KEY (moviename,releasedate) REFERENCES
movie(moviename,releasedate)
    ON UPDATE CASCADE ON DELETE CASCADE,
  FOREIGN KEY (theatername,compname) REFERENCES
theater(theatername,compname)
    ON UPDATE CASCADE ON DELETE CASCADE
);

-- Movie view is one movie play paid with one card
CREATE TABLE movieview(
  moviename VARCHAR(40) NOT NULL,
  releasedate DATE NOT NULL,
  theatername VARCHAR(20) NOT NULL,
  compname VARCHAR(20) NOT NULL,
  playdate DATE NOT NULL,
  creditcardnum CHAR(16) NOT NULL,

  PRIMARY KEY (moviename, releasedate, theatername, compname,
playdate, creditcardnum),
  FOREIGN KEY (moviename, releasedate, theatername, compname,
```

```
playdate) REFERENCES movieplay(moviename, releasedate, theatername,
compname, playdate)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  FOREIGN KEY (creditcardnum) REFERENCES creditcard(creditcardnum)
    ON UPDATE RESTRICT ON DELETE RESTRICT
);

-- Theater visit is one user's visit to one theater on one date (can
have duplicates so visitID is needed)
CREATE TABLE theatervisit(
  visitid INT PRIMARY KEY auto_increment,
  theatername VARCHAR(20) NOT NULL,
  compname VARCHAR(20) NOT NULL,
  username VARCHAR(20) NOT NULL,
  visitdate DATE NOT NULL,

  FOREIGN KEY (theatername,compname) REFERENCES
theater(theatername,compname)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  FOREIGN KEY (username) REFERENCES user(username)
    ON UPDATE CASCADE ON DELETE CASCADE
);
```