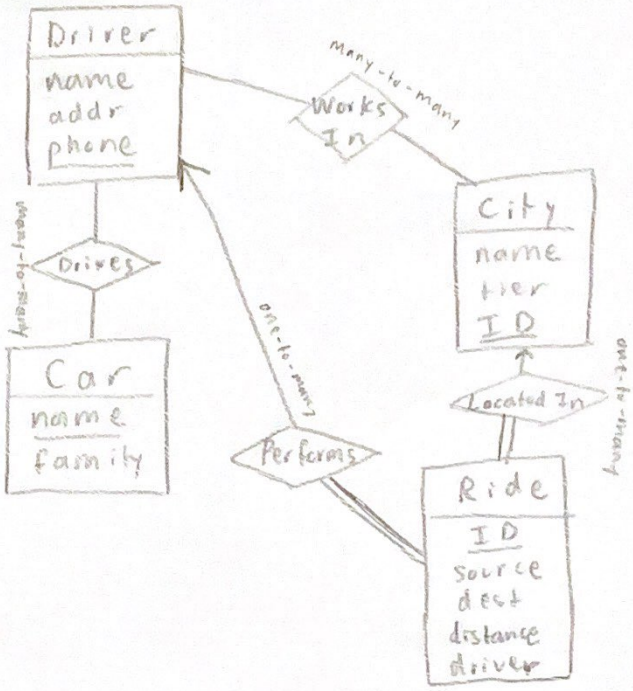
CS 143 HW 4

Ethan Wang
305319061
10/24/21

1. a) b)



Driver(name, addr, phone)
City(name, tier, ID)
Car(name, family)
Ride(ID, source, destination, distance, driver)

Drives(Driver.phone, Car.name)
Performs(Driver.phone, Ride.ID)
LocatedIn(Ride.ID, City.ID)
WorksIn(Driver.phone, City.ID)

Note: Driver.phone is considered a "key" since there can be drivers with the same name. The phone number acts as a sort of ID since phone numbers are unique.

```
CREATE TABLE Driver (name varchar(255), addr varchar(255), phone int,
                     PRIMARY KEY(phone));
CREATE TABLE City (name varchar(255), tier varchar(255), ID int,
                   PRIMARY KEY(ID));
CREATE TABLE Car (name varchar(255), family varchar(255), PRIMARY KEY(name));
CREATE TABLE Ride (ID int, source varchar(255), destination varchar(255)
                   distance int, driver varchar(255), PRIMARY KEY(ID));
```

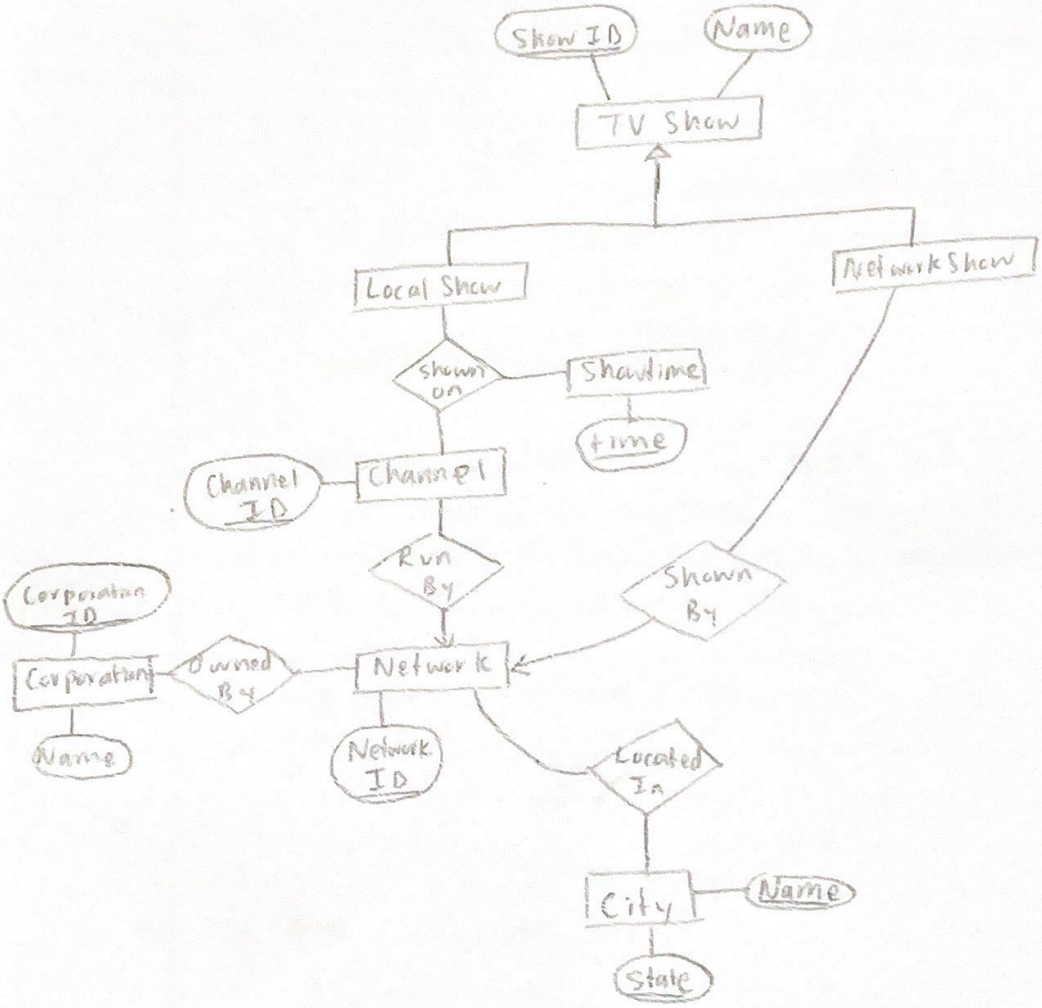
varchar(255) int

CREATE TABLE WorksIn (driver_phone, city_id, primary key(driver_phone, city_id))

CREATE TABLE LocatedIn (Ride_ID int , City_id int primary key (id))

etc

- TV Show
  TV Network
  Cities
  Channels
  Showtimes



3. Programmer (name)
   TeamLeader (name, team)
   Leads (TeamLeader.name, Programmer.name(s), project)

Ethan
Wong
10124

HW 4 - Relational Design Theory

1. R(A, B, C, D, E, F)
   (A, B, C, F) (A, D, E)
   A → BC CD → E B → D E → A

   • (A, B, C, F) ∩ (A, D, E) = A
   ↳ A is a candidate key for (A, D, E) so the decomposition is lossless

2. A → B
   (a_1 implies b_1, a_2 implies b_1) ✓
   C → B
   (c_1 implies b_1, c_2 implies b_1, c_3 implies b_1) ✓
   • A does not imply C as $a_2 \rightarrow c_1, c_2$ (inconsistent)
   • B does not imply A, C as $b_1 \rightarrow a_1, a_2$ (inconsistent)
   and $b_1 \rightarrow c_1, c_2, c_3$ (inconsistent)

3. a) sid → dept, cnum dept, cnum → sid
   b) sid → dept, cnum

4. R(A, B, C, D, E)
   A → BC CD → E B → D E → A
   a) Yes, 'A' is a candidate key
   A → BC
   ↳ D → E → A
   b) Yes, 'BC' is a candidate key
   BC
   ↳ D → E → A

REA

CREATE TABLE Ride (ID int, source varchar(255), destination varchar(255)
distance int, driver varchar(255), PRIMARY KEY(ID));

5. R(A, B, C, D, E, F)

A → BC, C → E, B → D

Key is AF - A reaches everything besides F, so just "AF"

NOT BCNF

R(A, B, C, D, E, F)

B → D: R_1(A, B, C, E, F) R_2(B, D) ✓

C → E: R_3(A, B, C, F) R_4(C, E) ✓

A → BC: R_5(A, F) R_6(A, B, C)

R_2 (B, D)

R_4 (C, E)

R_5 (A, F)

R_6 (A, B, C)