Katie Stevens

CPSC 321 01

HW3

10/5/21

Part 1

1.

a) X -> Y and YW -> Z

By transitivity X can sub for Y due to X -> Y therefore XW -> Z

b) X -> YZ

By reflexivity YZ -> Y.

By Transitivity since X -> YZ and YZ -> Y therefore X -> Y.

By reflexivity YZ -> Z.

By Transitivity X -> YZ and YZ -> Z therefore X -> Z

c) X -> Y and X -> Z

By augmentation X -> XZ.

By augmentation XZ -> YZ.

By transitivity X can sub for XZ therefore X -> YZ.

2. Minimal Cover

{a -> c, c -> b}

{a -> cd, c -> d, d -> a}

{ab -> c, c -> ade}

3. Candidate keys

a) a, b

b) b, d

c) a, b

d) c

4. FDs

Investor -> broker

Broker -> office

Investor, stock -> quantity

Stock -> dividend

Candidate keys: investor, stock

5.

dsssa) candidate keys: director

form: 2NF

b) candidate keys: title, director

form: BCNF

c) candidate keys: award_type, award_year

form: BCNF

d) candidate keys: studio_loc

form: BCNF

e) candidate keys: director

form: BCNF

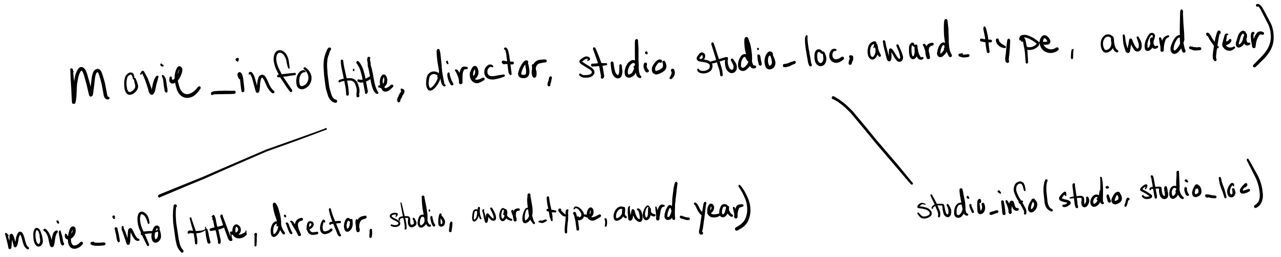
6. Decomposition

“bad” key – studio -> studio_loc

movie_info(title, director, studio, award_type, award_year)

studio_info(studio, studio_loc)

candidate keys: award_type, award_year, studio


movie_info(title, director, studio, studio_loc, award_type, award_year)
movie_info(title, director, studio, award_type, award_year)
studio_info(studio, studio_loc)

7.

a) candidate keys: student, course & student, time

form: 3NF

b) candidate keys: student, course

form: 3NF

c) candidate keys: student, instructor, grade

form: BCNF

8. enrolled_in relation is in 3NF. It is in 3NF because for every functional dependency X -> Y, Y is a part of some candidate key.

9. restaurants(name, address, city, state, zip, rating)

Name, address, city, state -> zip

Zip -> state

10. FDs from HW-1

LegoBrick(brick id, design id, name, color, price)

LegoSet(set item number, year range, category, name, age range, price, minifigure count, vip points, height, width, depth, theme, total pieces)

Foreign keys: theme, category

Theme(name, year range, description, license)

PartList(set item number, brick id, num_pieces)

Brick_id -> brick_name, color, price

Set_item_number, year_range, category -> set_name, age_range, price, vip points, total pieces

Theme -> theme_name, year_range

theme_name, year_range -> description, license

Part_list -> set_item_number, brick_id, num_pieces

11. step (a)

LegoBrick(brick_id, design_id, name, color, price)

LegoSet(set_item_num, year_range, category, name, price, age_range, minifigure_count, vip_points, total_pieces)

LegoSetAttributes(set_item_num, theme, category, height, width, depth)

Theme(theme_name, year_range, description, license)

PartList(set_item_number, brick_id, num_pieces)