26/40

① a) If $X \rightarrow Y$ and $YW \rightarrow Z$, then $XW \rightarrow Z$
This expression can be broken down more into, if $X \rightarrow Y$ and $Y \rightarrow Z$ and $W \rightarrow Z$. Since $Y \subseteq X$ and $Z \subseteq Y$, we know $X \subseteq Z$ by transitivity.
Since $Z \subseteq W$ and $X \rightarrow Z$, we know $Z \subseteq W$ and $Z \subseteq X$ by augmentation.
This resulting expression can be rewritten $XW \rightarrow Z$ by reflexivity.

b) If $X \rightarrow YZ$, then $X \rightarrow Y$ & $X \rightarrow Z$
Since $YZ \subseteq X$, this can be expressed $Y \subseteq X$ and $Z \subseteq X$ by reversing augmentation. The resulting expression can be rewritten $X \rightarrow Y$ and $X \rightarrow Z$

c) If $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$
Since $X \rightarrow Y$ can be expressed $Y \subseteq X$ through transitivity, $Y \subseteq X$ and $X \rightarrow Z$ is equal to $X \rightarrow YZ$ according to the definition of augmentation

② a) minimal −0

b) $\{ab \rightarrow cd, c \rightarrow d, d \rightarrow a, db \rightarrow ac\} = \{ab \rightarrow c, b \rightarrow d\}$

$a \rightarrow c, b \rightarrow d, c \rightarrow d, d \rightarrow a, d \rightarrow a, b \rightarrow c$

$ab \rightarrow c, b \rightarrow d$ too far −1

c) $\{ab \rightarrow c, c \rightarrow de, c \rightarrow a\} = \{b \rightarrow acde\}$

$a \rightarrow c, b \rightarrow c, c \rightarrow d, c \rightarrow e, c \rightarrow a$

$b \rightarrow acde$ same −0 (already minimal)

③ a) $\{$~~a~~$\}$ −1

b) $\{b, d\}$

c) $\{$~~a~~$\}$ −1

d) $\{$~~c~~$\}$ −1

$c \rightarrow b, bd \rightarrow e, a \rightarrow d, e \rightarrow a$

$c \rightarrow b, b \rightarrow e, d \rightarrow e, a \rightarrow d$

$c \rightarrow b, b \rightarrow e, a \rightarrow e$

$c \rightarrow e$

④ Stock (broker, office, investor, stock, quantity, dividend)

FDs : { broker → office, investor ; investor → stock, quantity, dividend

Candidate key : { broker }

5. a) CK: { director } 1NF

b) ~~CK: { title } 1NF~~ -1

c) CK: { award_type, award_year } 3~~NF~~ BCNF -1

d) CK: { award_type, award_year } 2~~NF~~ " -1

e) CK: { director }. 3~~NF~~ -1

6. Movie_info(title, director, studio, studio_loc, award_type, award_year)

Movie(title, director, studio, award_type, award_year) Studio(studio, studio_loc)

Movie(title, director, studio, award_year) studio(studio, studio_loc) award(award_year, type)

7. a) CK: { Student, ~~course }~~ ~~2NF~~

b) CK: { Student, ~~course } 3NF~~

c) CK : { student } 1~~NF~~ -1

Both (a) & (b) have 2 keys & are BCNF -2

8. Based on the current functional dependancies, the highest normal form the relation can be in is ~~2NF~~ because the prime grade, key in student, course → grade, instructor, time violates the expectation that there be no prime FDs. Course → grade creates a prime FD

3NF -1

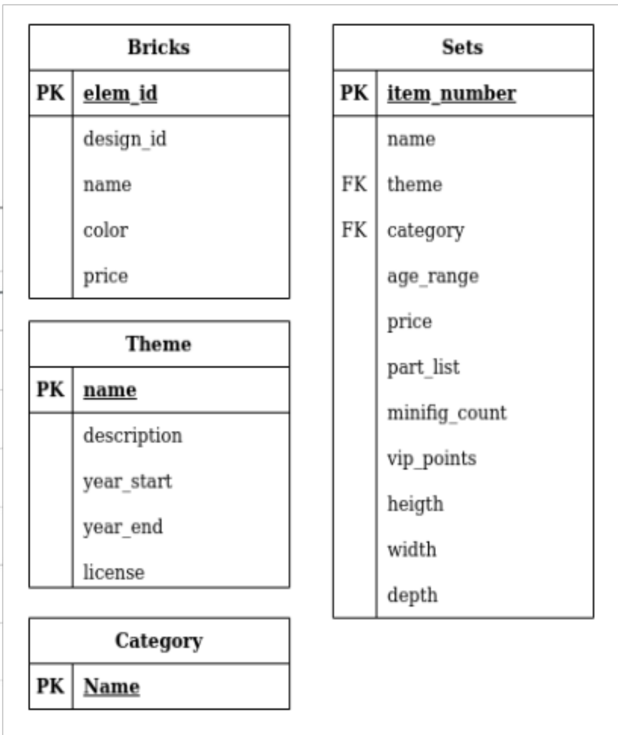
9. Steam_games (title, game_file, year, publisher, game_assets)

title, year, publisher → game_file, game_assets

game_file → game_assets

{game_file, game_assets} ∩ {game_assets} = {game_assets}

Game asset is not a key in {title, year, publisher} or {game_file}, so it is not lossless, & therefore not in BCNF



Bricks	
PK	elem_id
	design_id
	name
	color
	price

Theme	
PK	name
	description
	year_start
	year_end
	license

Category	
PK	Name

Sets	
PK	item_number
	name
FK	theme
FK	category
	age_range
	price
	part_list
	minifig_count
	vip_points
	heigth
	width
	depth

See included schema At end

(10) Bricks

elem-id, design-id → name, color, price

Theme

name → description, lisense

license → description, year_start, year-end

Sets

item-number → name, theme, category, age_range, price, part_list, minifig-count, vip-points

theme → category, age-range

part_list → minifig-count

(11) (a) Bricks(elem_id, design-id, name, color, price)

brickID(elem_id, design_id, name) brickDesc(name, color, price)

Theme(name, description, year_start, year-end, license)

ThemeDesc(name, description, license) LicenseInfo(license, description, year_start, year-end)

Sets(item_number, name, theme, category, age-range, price, part_list, minifig-count, vip-points, hight, width, depth)

setID(item-number, name, theme, part_list, vip-pts, height, width, depth)

SetTheme(theme, category, age-range)

Parts(part_list, minifig-count)