

R1.1

MovieTitle, Director -> Producer

SPTitle, Author -> temp

SPTitle, Author, SceneNo -> Storylocation, Filmlocation

SPTitle, Author, SceneNo, TakeNo -> Date, Time

ActorName -> Phone, Address, AgentName

MovieTitle, Director -> Movieof_ SPTitle, Movieof_ Author //Movie of

ActorName, SPTitle, Author, SceneNo -> AppearsIn _temp //Appears In

R1.2

MovieTitle, Director -> Producer

MovieTitle, Director -> Movieof_ SPTitle

MovieTitle, Director -> Movieof_ Author

SPTitle, Author -> temp

SPTitle, Author, SceneNo -> Storylocation

SPTitle, Author, SceneNo -> Filmlocation

SPTitle, Author, SceneNo, TakeNo -> Date

SPTitle, Author, SceneNo, TakeNo -> Time

ActorName -> Phone

ActorName -> Address

ActorName -> AgentName

ActorName, SPTitle, Author, SceneNo -> AppearsIn _temp

Minimal Cover:

MovieTitle, Director -> Producer, Movieof_ SPTitle, Movieof_ Author

SPTitle, Author -> temp

SPTitle, Author, SceneNo -> Storylocation, Filmlocation

SPTitle, Author, SceneNo, TakeNo -> Date, Time

ActorName -> Phone, Address, AgentName

MovieTitle, Director -> Movieof_ SPTitle, Movieof_ Author

ActorName, SPTitle, Author, SceneNo -> AppearsIn _temp

R1.3

[MovieTitle, Director | Producer, Movieof_ SPTitle, Movieof_ Author]

[SPTitle, Author | ~~temp~~]

[SPTitle, Author, SceneNo | Storylocation, Filmlocation]

[SPTitle, Author, SceneNo, TakeNo | Date, Time]

[ActorName | Phone, Address, AgentName]

[ActorName, SPTitle, Author, SceneNo | ~~AppearsIn _temp~~]

R2.1

code -> booktitle, publisher, date

songID -> songtitle, composer

userID -> password, name, emailAddress

userID, code -> ownsbook_pdfFilename, ownsbook_pageOffset

songID, code -> appearsin_length, appearsin_page

R2.2

code -> booktitle

code -> publisher

code -> date

songID -> songtitle

songID -> composer

userID -> password

userID -> name

userID -> emailAddress

userID, code -> ownsbook_pdfFilename

userID, code -> ownsbook_pageOffset

songID, code -> appearsin_length

songID, code -> appearsin_page

Minimal Cover:

code -> booktitle, publisher, date

songID -> songtitle, composer

userID -> password, name, emailAddress

userID, code -> ownsbook_pdfFilename, ownsbook_pageOffset

songID, code -> appearsin_length, appearsin_page

R2.3

[code | booktitle, publisher, date]

[songID | songtitle, composer]

[userID | password, name, emailAddress]

[userID, code | ownsbook_pdfFilename, ownsbook_pageOffset]

[songID, code | appearsin_length, appearsin_page]

R3.1

stdnum -> email, name, strnum, street, city, area_code, office_code, station_code
email -> stdnum, name, strnum, street, city, area_code, office_code, station_code
area_code, office_code -> city
postcode -> city
course_num -> department_name
period, term, room_num -> course_section
room_num -> building
stdnum, course_section -> enrolledin_temp //N:N
stdnum, course_num -> grade

R3.2

stdnum -> email	
stdnum -> name	stdnum -> email and email -> name //transitive rule
stdnum -> strnum	stdnum -> email and email -> strnum //transitive rule
stdnum -> street	stdnum -> email and email -> street //transitive rule
stdnum -> city	stdnum -> email and email -> city //transitive rule
stdnum -> area_code	stdnum -> email and email -> area_code //transitive rule
stdnum -> office_code	stdnum -> email and email -> office_code //transitive rule
stdnum -> station_code	stdnum -> email and email -> station_code //transitive rule
email -> stdnum	
email -> name	
email -> strnum	
email -> street	
email -> city	
email -> area_code	
email -> office_code	
email -> station_code	
area_code, office_code -> city	
postcode -> city	
course_num -> department_name	
period, term, room_num -> course_section	
room_num -> building	
stdnum, course_section -> enrolledin_temp	
stdnum, course_num -> grade	

R3.3

[stdnum | email]
[email | stdnum, name, strnum, street, city, area_code, office_code, station_code]
[area_code, office_code | city]
[postcode | city]
[course_num | department_name]
[period, term, room_num | course_section]
[room_num | building]
[stdnum, course_section | enrolledin_temp]
[stdnum, course_num | grade]

Minimal Cover:

stdnum -> email
email -> stdnum
email -> name
email -> strnum
email -> street
email -> city
email -> area_code
email -> office_code
email -> station_code
area_code, office_code -> city
postcode -> city
course_num -> department_name
period, term, room_num -> course_section
room_num -> building
stdnum, course_section -> enrolledin_temp
stdnum, course_num -> grade

R4.1

Attribute	Comment
partsid	car part number
name	car part name
brand	car part brand
carid	car number
make	car make e.g. subaru
model	car model e.g. wrx
year	car year
categoryname	category name for car parts e.g. suspension
videoid	installation video number
link	installation video web link
userid	user name that can contain letters or/and numbers e.g. 05subie
password	user password
carlistid	list of cars number owned by a user

R4.2

- 1) Car parts can have multiple brands, and each brand have many part names in the database.
- 2) You can use the partsid to uniquely identify a car part brand and name combination
- 3) Cars can have different makes and models, there would be more years per model and there would be more models per make. However, a carid can be used to uniquely identify a make, model, and year combination.
- 4) Since cars and car parts have a N:N relationship because many cars can have many car parts and many car parts can belong to many cars, car parts can be separated into different categories which would have their own names for each section e.g. suspension.
- 5) Each car part would have an installation video. Each installation video would have only one link attached to it. A video number can uniquely identify a video link.
- 6) Each user must have a unique username to store in the data base different from everyone else, however passwords can be coincidentally the same. This means the userid identifies their user password.
- 7) A user can also contain a list of cars (which is kept using carlistid to separate from the other users) they've previously inputted into the system that they have the option to store and save onto their profile. That is you would need the combination of the userid and carslistid to retrieve information.

R4.3

partsid -> name, brand

carid -> make, model, year

videoid -> link

userid -> password

userid, carlistid -> temp

carlistid, carid -> carlist_has_cars_temp

carid, partsid -> categories_categoryname

partsid -> has_videoid

R4.4

partsid -> name

partsid -> brand

partsid -> has_videoid

carid -> make

carid -> model

carid -> year

videoid -> link

userid -> password

userid, carlistid -> temp

carlistid, carid -> carlist_has_cars_temp

carid, partsid -> categories_categoryname

Minimal Cover:

partsid -> name, brand, has_videoid

carid -> make, model, year

videoid -> link

userid -> password

userid, carlistid -> temp

carlistid, carid -> carlist_has_cars_temp

carid, partsid -> categories_categoryname

R4.5

[partsid | name, brand, has_videoid]

[carid | make, model, year]

[videoid | link]

[userid | password]

[userid, carlistid | temp]

[carlistid, carid | carlist_has_cars_temp]

[carid, partsid | categories_categoryname]