RICKY GULATI 101146071

COMP3005B Assignment 4 – Winter 2021

Problem 1:

R1.1: // List of functional dependencies

MovieTitle, Director -> Producer

SPTitle, Author -> SPTitle, Author

MovieTitle, Director -> SPTitle, Author

ActorName -> Phone, Address, AgentName

SPTitle, Author, SceneNo -> StoryLocation, FilmLocation

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

ActorName, SceneNo, SPTitle, Author -> temp

TakeNo, SceneNo, SPTitle, Author -> SceneNo, SPTitle, Author

SceneNo, SPTitle, Author -> SPTitle, Author

R1.2: // Minimal cover

MovieTitle,Director->Producer,SPTitle,Author

ActorName->Phone,Address,AgentName

SPTitle,Author,SceneNo->StoryLocation,FilmLocation

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

ActorName,SceneNo,SPTitle,Author->temp

R1.3: // Loss-less Join, Dependency Preserving, 3NF tables:

[Primary Key Attributes | Non-Primary Key Attributes]

[MovieTitle,Director | Producer,SPTitle,Author]

[ActorName | Phone,Address,AgentName]

[SPTitle,Author,SceneNo | StoryLocation,FilmLocation]

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

[ActorName,SceneNo,SPTitle,Author | ~~temp~~]

[MovieTitle,Director,ActorName,SceneNo,TakeNo |]

Problem 2:

R2.1: // Functional dependencies

code -> BOOKCODES_title, publisher, date

songID -> SONGS_title, composer

userID -> password, name, emailAddress

code, songID -> length, page

code, userID -> pdfFileName, pageOffset

R2.2: // Minimal cover

code->BOOKCODES_title,publisher,date

songID->SONGS_title,composer

userID->password,name,emailAddress

code,songID->length,page

code,userID->pdfFileName,pageOffset

R2.3: // Loss-less Join, Dependency Preserving, 3NF tables:

[Primary Key Attributes | Non-Primary Key Attributes]

[code | BOOKCODES_title,publisher,date]

[songID | SONGS_title,composer]

[userID | password,name,emailAddress]

[code,songID | length,page]

[code,userID | pdfFileName,pageOffset]

[code,songID,userID |]

Problem 3:

R3.1: // Functional dependencies

stdnum -> name, email

course_num -> course_name, department_name

course_num, course_section -> room, period

course_section -> term

stdnum -> stationcode

city -> areacode, officecode

postcode ->city

stdnum -> strnum, street, postcode

room_num -> building

stdnum, course_num -> grade

stdnum, course_section, course_num -> temp

R3.2: // Minimal cover

stdnum->name,email,stationcode,strnum,street,postcode

course_num->course_name,department_name

course_num,course_section->room,period

course_section->term

city->areacode,officecode

postcode->city

room_num->building

stdnum,course_num->grade

stdnum,course_section,course_num->temp

R3.3: // Loss-less Join, Dependency Preserving, 3NF tables:

[Primary Key Attributes | Non-Primary Key Attributes]

[stdnum | name,email,stationcode,strnum,street,postcode] // email is a candidate key

[course_num | course_name,department_name]

[course_num,course_section | room,period]

[course_section | term]

[city | areacode,officecode]

[postcode | city]

[room_num | building]

[stdnum,course_num | grade]

[stdnum,course_section,course_num | ~~temp~~]

[stdnum,course_num,course_section,room_num |]

Problem 4:

Assumptions and constraints:

1. Users can have only one name and email.
2. Games can have only one name, release year, genre, steam price, epic price and manufacturer price.
3. A user can be uniquely identified by their user ID.
4. A game can be uniquely identified by its game ID.
5. A user can only play one game at a time which is listed as their current game ID.
6. A user cannot have the same game in their game library more than once.
7. The developer ID is taken from the user IDs.

R4.1: // Functional dependencies

U_id -> Users_Name, Email

G_id -> Games_Name, Year, Steam_price, Epic_price, Manu_price, Genre

G_id -> curgame_id

U_id -> Dev_id

U_id, Friend-of_U_id -> temp

U_id, G_id -> Game_num

R4.2: // Minimal cover

U_id->Users_Name,Email,Dev_id

G_id->Games_Name,Year,Steam_price,Epic_price,Manu_price,Genre,curgame_id

U_id,Friend-of_U_id->temp

U_id,G_id->Game_num

R4.3: // Dependency Preserving, 3NF tables:

[Primary Key Attributes | Non-Primary Key Attributes]

[U_id | Users_Name,Email,Dev_id]

[G_id | Games_Name,Year,Steam_price,Epic_price,Manu_price,Genre,curgame_id]

[U_id,Friend-of_U_id | ~~temp~~]

[U_id,G_id | Game_num]

[U_id,G_id,Friend-of_U_id |]