**CS425 -- HW3 Part 2**

**Group Name: TCBT (The Country’s Best Theaters)**

**Feature Ownership:**

**Andrew Caron:** specified functional dependencies, checked the reliability of decompositions and restructuring of tables and rearranged where needed.

**Ayesha Ahmed: (EDITOR)** specified functional dependencies, decompositions and restructured tables.

**Emily Warman:** specified functional dependencies, checked the need for decompositions.

**\*\*//\*\***

Since Dhruv left our group here are the previous HW3 part 1 updated feature ownerships:

Emily:  Membership, staffing, movies, Recommendations

Andrew: Web Development link btwn java and sql, Discussion Board

Ayesha: Reward points, Location management, Tickets, Web Dev. link btwn java and sql,

~~Dhruv: Web Development link btwn java and sql, Discussion Boards~~

**\*\*//\*\***

**Schedule**(showing\_id, room\_id, time, movie, ticket\_price, tickets\_sold)

*FK: movie references title in Movies*

*FK: room\_id references room\_id in TheaterInfo*

*PK: showing\_id*

*tickets\_sold cannot exceed capacity of room\_id*

**FD: showing\_id -> room\_id, time, movie, ticket\_price, tickets\_sold (BCNF)**

**Decomposition: removed theater\_id to make BCNF since Theater info already specifies theater\_id for room\_id**

**TheaterInfo**(room\_id, theater\_id, room\_num, capacity, room\_type)

*FK: theater\_id references theater\_id in Locations*

*PK: room\_id*

**FD: room\_id -> room\_num, theater\_id, capacity, room\_type (BCNF)** (each room

**Decomposition: NONE already BCNF** (added room\_type which affects pricing if it has IMAX 4D)

**Locations**(theater\_id,name, street\_1, street\_2, city, state, zip) ← *to search theaters by zip code*

*street\_2 can be null*

*PK: theater\_id*

**FD: theater\_id -> name and all address fields (BCNF)** (all theaters have different addresses)

**Decomposition: NONE already BCNF**

**Movies**(title, star1, star2, star3, director, movie\_type1, movie\_type2, movie\_type3, description)

*movie\_type(n) can only be certain genres*

*star2, star3, movie\_type2, movie\_type3 can be NULL*

*PK: title*

**FD: title ->all other movie info fields (BCNF)** (Because the title of a movie is unique and contains unique stars/type/description. Whereas stars/type/director have many movies. [sequels/ect.])

**Decomposition: NONE already BCNF**

**RegisteredUsers**(user\_id, username, password, name, ccn, street1, street2, city, state, zip, phone, email)

*PK: user\_id*

*FK: ccn references ccn in CC table*

*username can be NULL (guest purchase)*

*password can be NULL (guest purchase)*

**FD: user\_id -> all other user info (BCNF) (each user has unique name/ ccn/ addr/ phone/ email/ password)**

**CC**(ccn,cc\_name, card\_type, ccn\_code, exp\_date)

*PK: ccn*

**FD: ccn -> name, card\_type, ccn\_code, exp\_date (BCNF)** (each ccn has unique set of name on card, card type, security code, and expiration)

**Decomposition: originally: RegisteredUsers**(user\_id, username, password, name, ccn, cc\_name, cc\_type, cc\_code, cc\_exp, street1, street2, city, state, zip, phone, email)

**with PK: user\_id**

**so we got error: NOT BCNF OR 3NF because:**

**user\_id ->** username, password, name, ccn, street1, street2, city, state, zip, phone, email

**ccn ->** cc\_name, cc\_type, cc\_code, cc\_exp [where ccn is not PK it causes error]

**So, we decomposed the 2 dependencies**

**Tickets**(user\_id, showing\_id, date\_purchased, ticket\_no)

*PK: (user\_id,showing\_id,date\_purchased)*

*FK: user\_id references user\_id in RegisteredUsers*

*FK: showing\_id references showing\_id in Schedule*

**FD: user\_id, showing\_id, date\_purchased -> ticket\_no (BCNF)**(for showing\_id user has unique **amount of tickets)**

**Decomposition: NONE already BCNF**

**Points**(user\_id, current\_points, total\_points)

*PK: user\_id*

*FK: user\_id references user\_id in RegisteredUsers*

**FD: user\_id-> current\_points, total\_points (BCNF)** (because each unique user has set of points which belongs to them)

**Decomposition: NONE already BCNF**

**Rewards**(theater\_id, level, level\_boundary, offers, deals, movie\_points, review\_points)

FK: theater\_id

PK: (theater\_id, level)

**FD: theater\_id, level -> level\_boundary, offers, deals, move\_points, review (BCNF)** (each theater has different level requirements if needed)

**Decomposition: We had the below tables which we realized could be merged:**

**RewardLevels**(name, level\_points)

**Offers**(level, offer, deal)

**RewardPoints**(theater\_id, level, movie\_points, review\_points)

**-- So, we merged them into one Rewards table which we made BCNF while constructing**

**TheaterThreads**(id, theater, user\_id, text)

*PK: id*

*FK: user\_id references user\_id in RegisteredUsers*

*FK: theater references theater\_id in Locations*  
When a registered user comments on a thread, give them rewards points

**FD: id, theater -> user\_id, text (BCNF)** (unique user can start a unique thread, therefore thread = user)

**Decomposition: NONE already BCNF**

**TheaterComments**(thread\_id, comment\_number, user\_id, text)

*PK: (thread\_id, comment\_number)*

*FK: thread\_id references id in TheaterThreads*

*FK: user\_id references user\_id in RegisteredUsers*

**FD: thread\_id, comment\_number, user\_id -> text (BCNF)** (users can comment multiple times)

**Decomposition: NONE already BCNF**

**MovieThreads**(id, title, star1, star2, star3, director, user\_id, text)

*PK: id*

*FK: user\_id references user\_id in RegisteredUsers*

**FD: id, title, star1, star2, star3, director -> user\_id, text (BCNF)**

**Decomposition: NONE already BCNF**

**MovieComments**(thread\_id, comment\_number, user\_id, text)

*PK: (thread\_id,comment\_number)*

*FK: thread\_id references id in MovieThreads*

*FK: user\_id references user\_id in RegisteredUsers*

**FD: thread\_id, comment\_number, user\_id -> text (BCNF)** (users can comment multiple times)

**Decomposition: NONE already BCNF**

**Employees**(emp\_id, f\_name, l\_name, address, phone, ssn, hiring\_emp\_id, theater\_id)

*PK: emp\_id*

*FK: theater\_id references theater\_id in TheaterInfo*

*FK: hiring\_emp\_id references emp\_id in Management*

**FD: emp\_id, theater\_id -> f\_name, l\_name, address, phone, ssn, hiring\_emp\_id (BCNF)**

**Decomposition: NONE already BCNF**

**JobTypes**(job\_type,description)

*PK: job\_type*

**FD: job\_type -> description (BCNF)** (each job has a unique description)

**Decomposition: NONE already BCNF**

**EmpSchedule**(date\_time,emp\_id,job\_type,theater\_id)

*PK: (date\_time, emp\_id)*

*FK: theater\_id references theater\_id in TheaterInfo*

*FK: job\_type references job\_type in JobTypes*

**FD: date\_time, emp\_id -> job\_type, theatre\_id (BCNF)** (a given employee can only be working 1 job at 1 theatre at any given time)

**Decomposition: NONE already BCNF**

**Management**(emp\_id,job\_type,theater\_id, username, sched\_password)

*PK: emp\_id*

*FK: emp\_id references emp\_id in Employees*

*FK: job\_type references job\_type in JobTypes*

*FK: theater\_id references theater\_id in TheaterInfo*

*theater\_id can be NULL (as in the case of web admin and owner)*

*sched\_password can be NULL (as in case of web admin)*

**FD: emp\_id, job\_id, username -> theater\_id, sched\_password**

**Decomposition: NONE already BCNF**