

# COMP-421 Database Systems, Winter 2018

## Written Assignment 1: Data Modelling / Relational Algebra

Due Date Feb 1, 11:59pm

**This is an individual assignment. You need to solve these questions on your own. Use the discussion forum on mycourses for assignment 1 if you have any questions. TA office hours will be setup on the days before the due date.**

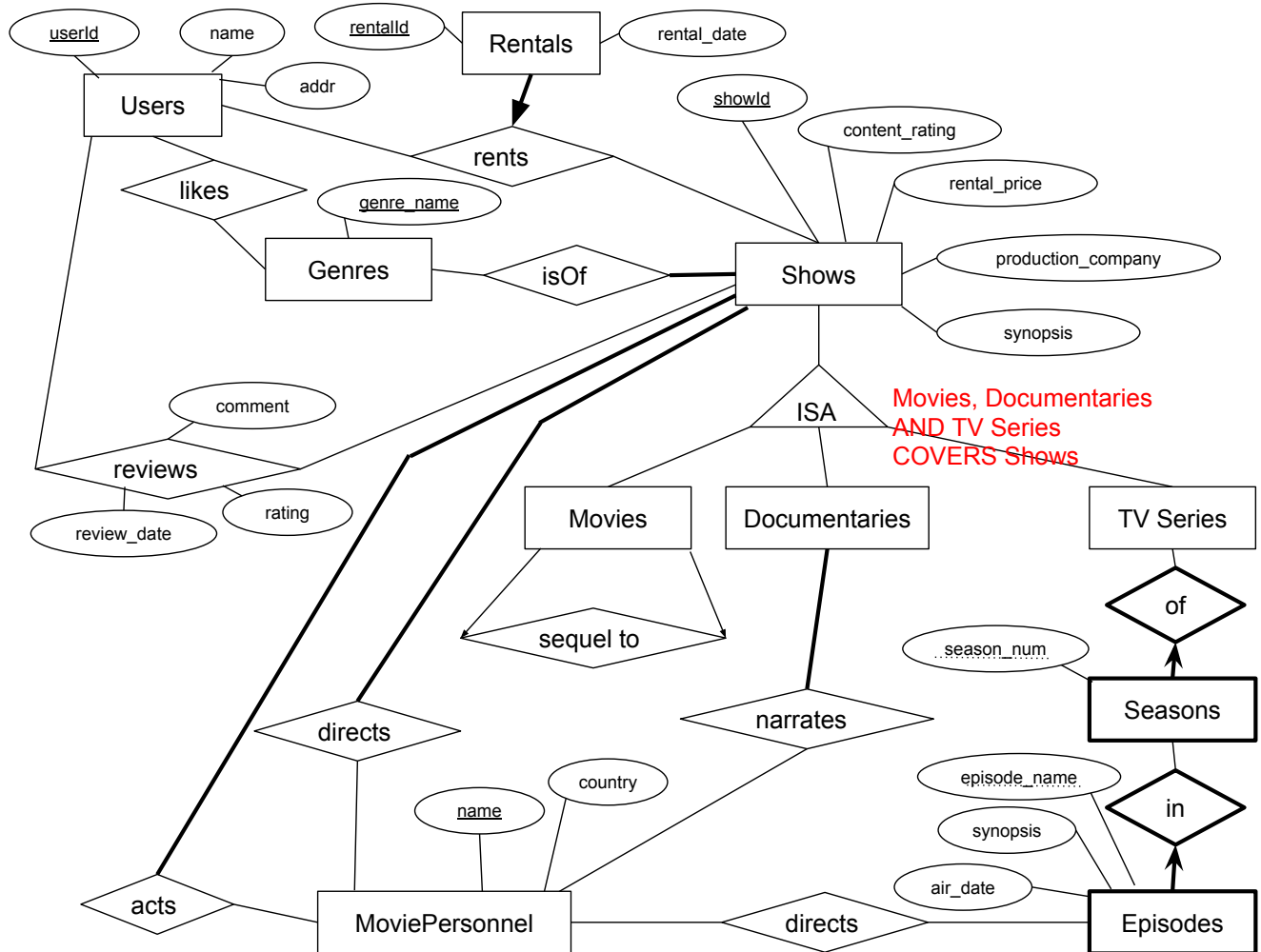
### Ex. 1 — ER Modelling (45 Points)

OnStream, an online streaming company is planning to use a database to manage its shows that users can rent and watch online.

- A show can be either one of a Movie (Life is Beautiful) / Documentary (Frozen Planet) / TV Series (Murdoch Mysteries).
  - Shows have a name (not necessarily unique), content rating, the production company associated with it, and a rental price. They are also associated with one or more genre.
  - TV series can have many seasons. Each season has a season number (incremental within the same TV Series)
  - A particular season can have many episodes. However, all of them are related with the same genre as the TV series. Each episode has an airing date and its own episode name.
  - Each show has a synopsis. Additionally, each episode of a TV series has its own synopsis.
  - Episode names are guaranteed to be unique only within a particular TV series.
  - Each show has associated with it one or more directors and actors.
  - Optionally, each episode of a TV series may have one or more episode directors.
  - Each documentary has one or more narrator(s) associated with it.
  - Same person can perform multiple roles (actor/director/narrator ) within the same show or across different shows. We need to keep track of their names and country of citizenship (no dual citizens). We will assume that their name is unique.
  - A movie can have a sequel that needs to be kept track of (for simplicity, we will consider only the immediately following movie as the sequel). For example Lost World is a sequel to Jurassic Park and Jurassic Park III is a sequel to Lost World.
  - Users can rent a show. Their name, address and user id are stored in the database. User id is unique to each user.
  - Users can also optionally indicate their interest in certain genres.
  - Once a user rents a show, they have 7 days to watch it (any number of times). So the date of rental needs to be kept track. They are free to rent (and watch) the same show multiple times. We do not have to keep track of how many times a user watches a particular show, but we need to keep track of all the rentals a user makes for a particular show.
  - Users can also optionally review (date, 1-5 rating and comment) a show (for TV series also reviews are done for the entire series not for a specific season or episode). Only one review is allowed per show for a user which they can edit and update later. We only keep track of the latest version of this review.
1. (30 Points) Design the ER Model for the following specification. Indicate any constraints that you might not express in the ER model. For each entity, decide on the primary keys. If you think there is no appropriate primary key, you can introduce an artificial key attribute. Also include any reasonable assumptions you have made for your model (that does not violate common sense or explicit requirements).
  2. (15 Points) Transform your diagram into the relational model (in the form  $R(\underline{a}, b, c)$ . Indicate primary keys by underlining them. Indicate foreign keys by letting them point to the relation to which they relate, eg.  $Q(\underline{d}, e, a)$  (a ref R) . DO NOT write SQL statements. Indicate if there are other constraints depicted in the ER model that you cannot (yet) describe in the relational model.

## Answer (Ex. 1) —

- Below is the ER diagram for the given requirements.



**Notes:-** rentalid and showid are artificial keys to ensure uniqueness.

In this ER diagram, we cannot capture the requirement that the season numbers should be incremental within a TV Series.

We also do not capture that review rating should be 1-5.

We have also made the assumption that a TV Series can exist without seasons or Seasons without any episodes yet. (As in when a new show/season is announced but is not available yet).

- Following are the relations obtained using the above ER model.

**Shows**(showid, content\_rating, rental\_price, production\_company, synopsis)

**Movies**(showid) showid is foreign key to Shows

**Documentaries**(showid) showid is foreign key to Shows

**TVSeries**(showid) showid is foreign key to Shows

**Seasons**(season\_num, showid) showid is foreign key to Shows

**Episodes**(episode\_name, season\_num, showid, air\_date, synopsis) (season\_num, showid) is foreign key to Seasons

**MoviePersonnel**(name, country)

**Users**(userid, name, addr)

**Rentals**(rentalid, userid, showid, rental\_date) userid foreign key to Users, showid foreign key to Shows

**Reviews**(userid, showid, comment, review\_date, rating) userid foreign key to Users, showid foreign key to Shows

Genres(genre\_name)  
 GenresOfInterest(userid, genre\_name) userid foreign key to Users, genre\_name foreign key to Genres  
 MovieGenre(showid, genre\_name) showid foreign key to Shows, genre\_name foreign key to Genres  
 Sequels(showid, sequelshowid) showid foreign key to Movies, sequelshowid foreign key to Movies(showid)  
 Acts(name, showid) name foreign key to MoviePersonnel, showid foreign key to Shows  
 DirectsShows(name, showid) name foreign key to MoviePersonnel, showid foreign key to Shows  
 DirectsEpisodes(name, episode\_name, season\_num, showid) name foreign key to MoviePersonnel, (episode\_name, season\_num, showid) foreign key to Episodes  
 Narrates(name, showid) name foreign key to MoviePersonnel, showid foreign key to Documentaries

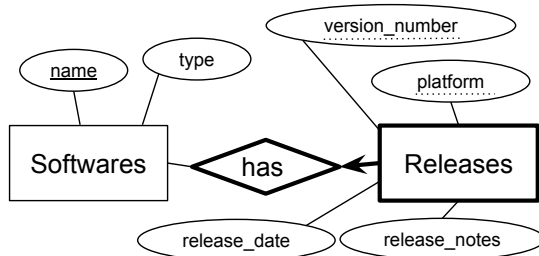
A show should be part of at the least one Genre and should have at the least one of actors and directors (as per our ER) is not captured in the relational model.

The relational model also do not enforce that there must be at least one narrator for a documentary.

For Sequels, since there is no uniqueness constraint on sequelshowid, a particular movie can end up being the sequel to two different movies.

## Ex. 2 — Testing your ER decoding skills. (16 points)

1. Consider the following ER diagram ( 8 points )

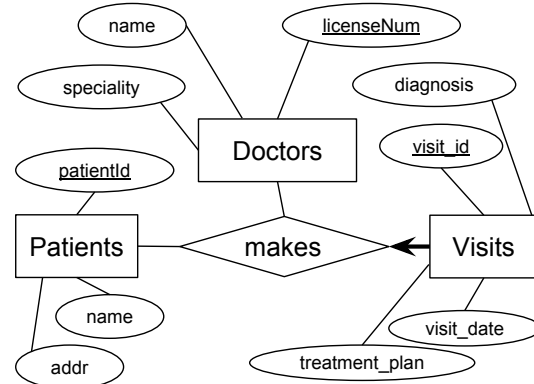


For each of the below statement, indicate whether you agree or disagree and briefly state your reasons (one liners are sufficient as long as they are to the point).

**You will get only half points if you do not state the reasons and just answer Agree/Disagree.**

- (A) A release can be not associated with any software.
- (B) A software can be not associated with any release.
- (C) A particular version of a given software may have more than one release date.
- (D) The number of entities in the **Releases** entity set will always be equal to or greater than the number of entities in the **Softwares** entity set.

2. Consider the following ER diagram ( 8 points )



For each of the below statement, indicate whether you agree or disagree and briefly state your reasons (one liners are sufficient as long as they are to the point).

**You will get only half points if you do not state the reasons and just answer Agree/Disagree.**

- (A) This model can be used to record a patient visiting different doctors on the same day for different diagnosis/treatments.
- (B) This model can be used to record a patient visiting the same doctor multiple times on the same day for different diagnosis/treatments.
- (C) For a given patient visit (recorded with a `visit_id` in the `Visits` entity set), multiple doctors can be involved in the diagnosis.
- (D) If we want to keep track of the `cost` associated with each patient visit, it can be made an attribute of either the `Visits` entity set or the `makes` relationship.

**Answer (Ex. 2) —**

1. (A) Disagree. This is not possible as **Releases** is a weak entity set and therefore each release need to be in a relationship with a software.  
(B) Agree. It does not have a participation constraint in the relationship.  
(C) Agree. The same version for a different platform could be released on a different date.  
(D) Disagree. Consider the case where the **Softwares** entity set has only one entry, but no releases has been made yet (i.e., no entities in the **Releases** entity set).
2. (A) Agree, these are stored with different visit ids for the same date.  
(B) Agree, these are also stored with different visit ids for the same date.  
(C) Disagree. A given visit id is associated with one doctor/patient interaction (key constraint).  
(D) Agree. Either approach is fine as there is a key constraint on **Visits** for the relationship.

**Ex. 3 — Relational Algebra (39 Points)**

Look at the following relational schema of an insurance company database (with some example tuples for each relation )

Drivers (dlnum, name, city, addr, expdate)  
          'L5231-623577-75', 'Jane Doe', 'Montreal', '3484 University Ave', '2020-09-30'

Automobiles (license, type, make, year)  
          'AN467TY', 'car', 'Subaru Forester', 2012

Accidents (reportid, city, addr, date)  
          612624, 'Montreal', 'Peel', '2018-01-10'

Regdrivers (dlnum, license)  
          'L5234-845666-90', 'MJ4KRT3'

Accidentinfo (dlnum, license, reportid, claim)  
          'L6452-825222-44', 'OP56K42', 533322, 2300

Some useful information.

- A driver can be registered to drive multiple automobiles of any type.
- An individual has only one driver's license number, no matter how many different types of automobiles ('car', 'truck', 'motorcycle', etc.) that person is allowed to drive.
- Multiple drivers can be registered to drive the same automobile.
- We can assume that when an accident occurs, the automobile involved was driven by one of its registered drivers.

Express the following queries in relational algebra using the notations and operators taught in class:

1. (2 Pts) List the name, city and address of all drivers in the city of 'Montreal'.
2. (3 Pts) List the reportids of all accidents involving the driver with the driving license number 'L1234-567890-11' and where the claim amount is above 200.
3. (3 Pts) Find the license plates of all automobiles with 'Billy Joe' as one of the registered drivers.
4. (4 Pts) Find the driving license number of all those drivers from 'Montreal' whose driving license expired as of '2018-01-01' but are still registered drivers for an automobile.
5. (3 Pts) Find the driving license number of all the drivers, such that a driver is a registered driver for at the least one of 'truck' and 'car' type automobiles.
6. (5 Pts) Find the driving license number of all drivers, such that a driver is a registered driver for a 'truck' but not for a 'car'.
7. (6 Pts) Find the distinct driving license number of all the drivers involved in more than one accident.
8. (6 Pts) Find the highest claim amount across all accidents involving the driver with the driver's license number 'L9876-543210-99'.

9. (7 Pts) Find the driving license of drivers such that the car(s) for which they are registered as drivers were involved in an accident, but they themselves were never involved in any accidents involving any cars that they are registered for.

**Answer (Ex. 3) —**

1.  $\Pi_{name,city,addr}(\sigma_{city='Montreal'}(Drivers))$
2.  $\Pi_{reportid}(\sigma_{dlnum='L1234-567890-11' \wedge claim > 200}(Accidentinfo))$
3.  $\Pi_{license}(\sigma_{name='BillyJoe'}(Drivers) \bowtie Regdrivers)$
4.  $\Pi_{dlnum}(\sigma_{expdate < '2018-01-01' \wedge city='Montreal'}(Drivers)) \cap \Pi_{dlnum}(Regdrivers)$
5.  $\Pi_{dlnum}(\sigma_{type='truck' \vee type='car'}(Automobiles) \bowtie Regdrivers)$
6.  $\Pi_{dlnum}(\sigma_{type='truck'}(Automobiles) \bowtie Regdrivers) - \Pi_{dlnum}(\sigma_{type='car'}(Automobiles) \bowtie Regdrivers)$
7.  $\rho(A1, Accidentinfo)$   
 $\rho(A2, Accidentinfo)$   
 $\Pi_{dlnum}(A1 \bowtie_{A1.dlnum=A2.dlnum \wedge A1.reportid \neq A2.reportid} A2)$
8. Either of the two solutions:  
 $\rho(A1, \sigma_{dlnum='L9876-543210-99'}(Accidentinfo))$   
 $\rho(A2, A1)$   
 $\Pi_{claim}(A1) - \Pi_{A1.claim}(A1 \bowtie_{A1.claim < A2.claim} A2)$   
 Or this:  
 $\rho(R, \Pi_{reportid}(\sigma_{dlnum='L9876-543210-99'}(AccidentInfo)))$   
 $\rho(C1, \Pi_{claim}(R \bowtie AccidentInfo))$   
 $\rho(C2, C1)$   
 $C1 - \Pi_{C2.claim}(C2 \bowtie_{C2.claim < C1.claim} C1)$
9.  $\rho(CD, \Pi_{dlnum, Automobiles.license}(\sigma_{type='car'}(Automobiles) \bowtie Regdrivers))$   
 $\rho(CDA, \Pi_{CD.dlnum, CD.license}(CD \bowtie Accidentinfo))$   
 $\rho(AD, \Pi_{dlnum}(CDA))$   
 $\rho(AC, \Pi_{license}(CDA))$   
 $\Pi_{CD.dlnum}(AC \bowtie CD) - AD$