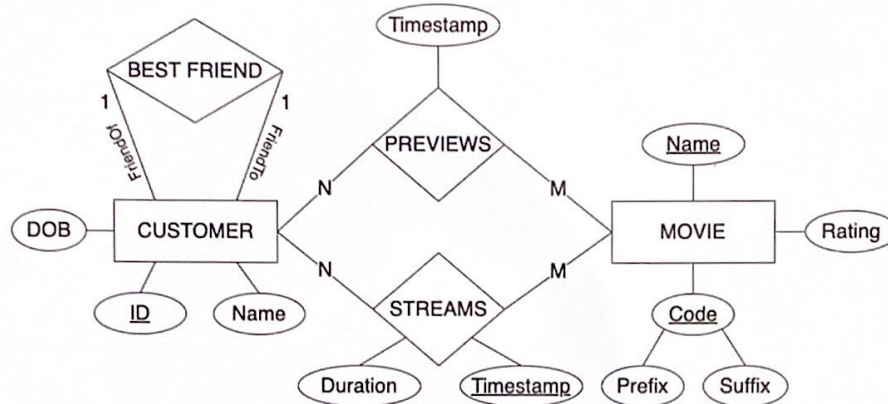


## ER Diagram:



## Contextual Information:

A relational database has been setup to track customer browsing activity for an online movie streaming service. Movies may be identified by a unique code which consists of a four-character prefix and four-digit suffix. Additionally, each movie is assigned a content rating which must be one of the following options: "G", "PG", "M", "MA15+" or "R18+". A user may preview a movie before they stream it however, they cannot preview a movie after they have started to stream it. Finally, the database should also not allow for the same customer to stream two movies at the same time. (This ensures no two streaming events overlap with each other). You may assume "Duration" refers to the time in seconds a customer spent streaming a particular movie after the "Timestamp".

## Relational Schema:

```
Customer [id, name, bob, bestFriend]
Customer.bestFriend references Customer.id

Movie [prefix, suffix, name, rating]

Previews [customer, moviePrefix, movieSuffix, timestamp]
Previews.customer references Customer.id
Previews.{moviePrefix, movieSuffix} reference Movie.{prefix, suffix}

Streams [customer, moviePrefix, movieSuffix, timestamp, duration]
Streams.customer reference Customer.id
Streams.{moviePrefix, movieSuffix} reference Movie.{prefix, suffix}
```

**INSTANCE DATA ON NEXT PAGE**

### Instance Data:

Customer			
<i>id</i>	<i>name</i>	<i>dob</i>	<i>bestFriend</i>
10234	Grace Smith	1998-09-12	NULL
23444	Jack Smith	2000-01-01	25321
24444	Levi Ramos	2001-08-23	26789
10235	Zhihui Wang	2003-04-26	NULL
25321	Yeseul Seo	1999-09-23	23444
26789	Samantha King	2002-12-06	24444
96721	Yumi Kim	1976-03-19	NULL

Movie			
<i>prefix</i>	<i>suffix</i>	<i>name</i>	<i>rating</i>
ABCD	1234	UP	PG
CDEF	1234	WALL-E	G
CDEF	2345	Titanic	M
ABCD	2345	Iron Man	M
EEDD	2345	Speed Racer	PG
MILK	4895	Inception	M
RTEH	4895	The Notebook	M

Previews			
<i>customer</i>	<i>moviePrefix</i>	<i>movieSuffix</i>	<i>timestamp</i>
10234	ABCD	1234	2022-02-20 19:12:56
23444	CDEF	2345	2022-02-20 19:42:08
24444	ABCD	1234	2022-02-20 19:45:31
10235	MILK	4895	2022-02-20 19:12:56
10235	ABCD	1234	2022-02-21 00:32:16
10234	CDEF	2345	2022-02-21 01:47:12
25321	CDEF	2345	2022-02-21 09:29:23
96721	RTEH	4895	2022-02-21 10:49:30

Streams				
<i>customer</i>	<i>moviePrefix</i>	<i>movieSuffix</i>	<i>timestamp</i>	<i>duration</i>
10234	ABCD	1234	2022-02-20 20:03:42	1623
25321	CDEF	1234	2022-02-20 20:04:11	4593
96721	CDEF	1234	2022-02-20 20:04:11	4593
24444	ABCD	1234	2022-02-20 20:05:31	967
23444	MILK	4895	2022-02-20 21:57:59	1243
24444	ABCD	2345	2022-02-20 22:42:01	2190
10235	MILK	4895	2022-02-21 03:45:02	1523
10234	ABCD	1234	2022-02-21 03:45:02	567

**Example question:** Insert (12345, 'Elaine Feng', 'I am not a date.', NULL) into *Customer*.

Will this operation cause an integrity constraint violation? **Yes** / No

If yes, name the integrity constraint(s) which will be violated:

Domain Constraint Violation

If yes, explain how the integrity constraint(s) would be violated:

*Domain Constraint Violation:* This tuple contains a non-date value for the *dob* attribute, specifically 'I am not a date.' Based on the instance data provided, it is clear the domain for the attribute is date values and hence this operation would violate domain constraint.

A.1 Insert ('EEDD', 2345, 'Frozen', NULL) into *Movie*.

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

Key Constraint Violation

If yes, explain how the integrity constraint(s) would be violated:

Key Constraint Violation: This tuple contains the same key value 'EEDD' and 2345 that already exists in the *Movie* table. Hence this operation would violate key constraint.

A.2 Insert (NULL, 'ABCD', 4895, '2022-02-22 06:14:56') into *Previews*.

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

Entity Integrity Constraint Violation

If yes, explain how the integrity constraint(s) would be violated:

*Entity Integrity Constraint Violation:* This tuple contains primary key 'NULL' which clearly violates the entity integrity constraint.

A.3 Insert ('RTEH', 2345, 'The Notebook', 'MA15+') into *Movie*.

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

?

If yes, explain how the integrity constraint(s) would be violated:

The movie already exists?

- A.4** Modify the tuple (25321, 'CDEF', 1234, '2022-02-20 20:04:11', 4593) in the *Streams* table and change it to (25321, 'CDEF', 2345, '2022-02-20 20:04:11', 4593).

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

If yes, explain how the integrity constraint(s) would be violated:

- A.5** Modify the tuple (10234, 'Grace Smith', '1998-09-12', NULL) in the *Customer* table and change it to (10235, 'Grace Jeon', '1998-09-12', 24444).

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

Key Constraint Violation

If yes, explain how the integrity constraint(s) would be violated:

*Key Constraint Violation:* This tuple contains the same key '10235' that is the same key value as another tuple in the Customer table. Hence this operation would violate key constraint.

- A.6** Modify the *prefix* attribute and set it to NULL for all the tuples in *Movie* relation with *name* = "Notebook".

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

Entity Integrity Constraint Violation

If yes, explain how the integrity constraint(s) would be violated:

*Entity Integrity Constraint Violation:* The operation modifying such that any part of primary key contains the value 'NULL' will clearly violate entity integrity constraint. Hence, in this operation changing prefix attribute to 'NULL' is violating the entity integrity constraint.

**A.7** Delete any tuple in *Streams* where the *duration* is greater than or equal to 1700.

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

If yes, explain how the integrity constraint(s) would be violated:

**A.8** Delete any tuple in the *Customer* table with a *dob* after '2001-08-17'.

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

If yes, explain how the integrity constraint(s) would be violated:

**A.9** Modify the tuple (10234, 'ABCD', 1234, '2022-02-21 03:45:02', 567) in the *Streams* table and change it to (10234, 'MILK', 4895, '2022-02-20 20:29:42', 567.1).

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

If yes, explain how the integrity constraint(s) would be violated:

**A.10** Insert the tuple () into *Movie*.

Will this operation cause an integrity constraint violation? Yes / No

If yes, name the integrity constraint(s) which will be violated:

If yes, explain how the integrity constraint(s) would be violated:

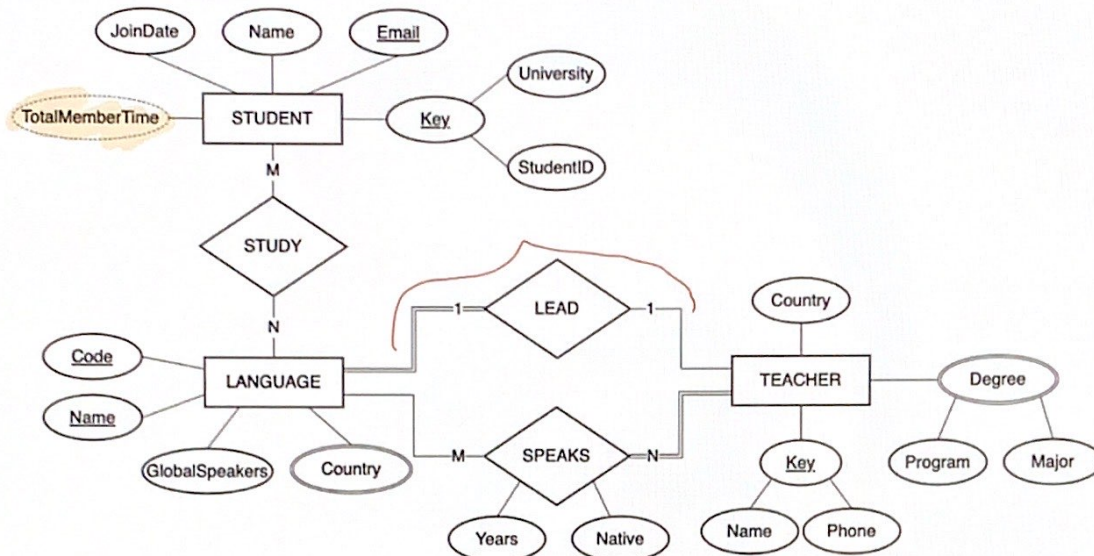
**SECTION B QUESTIONS START ON NEXT PAGE**

## Section B – Relational Mapping

The following three questions all contain ER or EER diagrams from the Module 1 Assignment solutions. For each diagram, complete ER to relation mapping and state your **final** relational schema, including any foreign keys which were created in the process.

- Your answers **must** follow the format guide posted on blackboard.
- All foreign key definitions **must** directly follow the table for which they are defined.

### B.1



Student[email, key, name, JoinDate]

Language[code, name, GlobalSpeaker, key]  
 Language.key references Teacher.key

LanguageCoun[code, name, country]  
 LanguageCoun.code references Language.code  
 LanguageCoun.name references Language.name

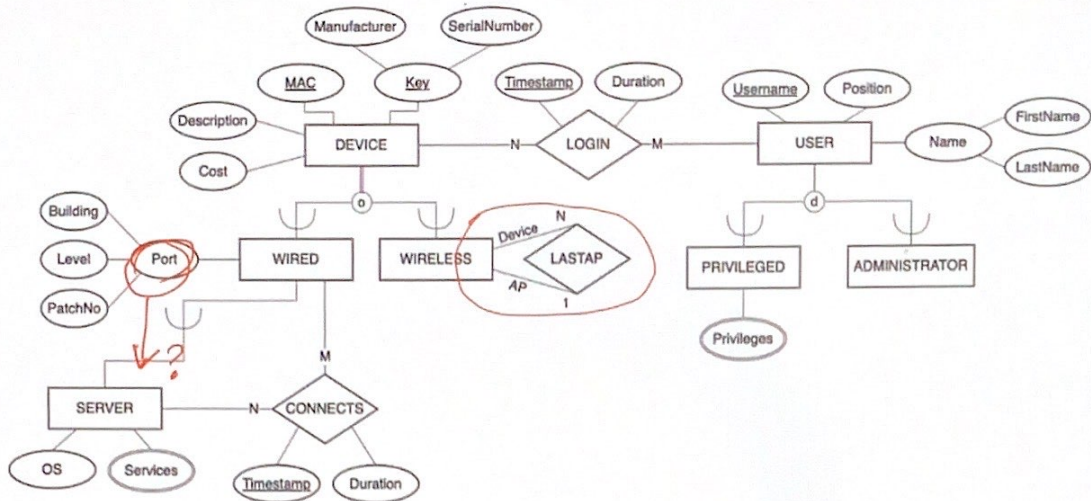
Study[email, key, name, code]  
 Study.email references Student.email  
 Study.key references Student.key  
 Study.name references Student.name  
 Study.code references Language.code

Teacher[key, country]

TeacherDeg[key, degree]  
 TeacherDeg.key references Teacher.key

Speaks[code, name, key, years, native]  
 Speaks.code references Language.code  
 Speaks.name references Language.name  
 Speaks.key references Teacher.key

## B.2



Device[MAC, key, description, cost]

Wired[MAC, key, port]

Wired.MAC references Device.MAC

Wired.key references Device.key

Server[MAC, key, OS]

Server.MAC references Device.MAC

Server.key references Device.key

ServerServices[MAC, key, services]

ServerServices.MAC references Device.MAC

ServerServices.key references Device.key

Connects[MAC, key, timestamp, duration]

Connects.MAC references Device.MAC

Connects.key references Device.key

Wireless[MAC, key, ??? ]

Wireless.MAC references Device.MAC

Wireless.key references Device.key

User[username, position, name]

Login[MAC, key, username, timestamp, duration]

Login.MAC references Device.MAC

Login.key references Device.key

Login.username references User.username

Privileged[username]

Privileged.username references User.username

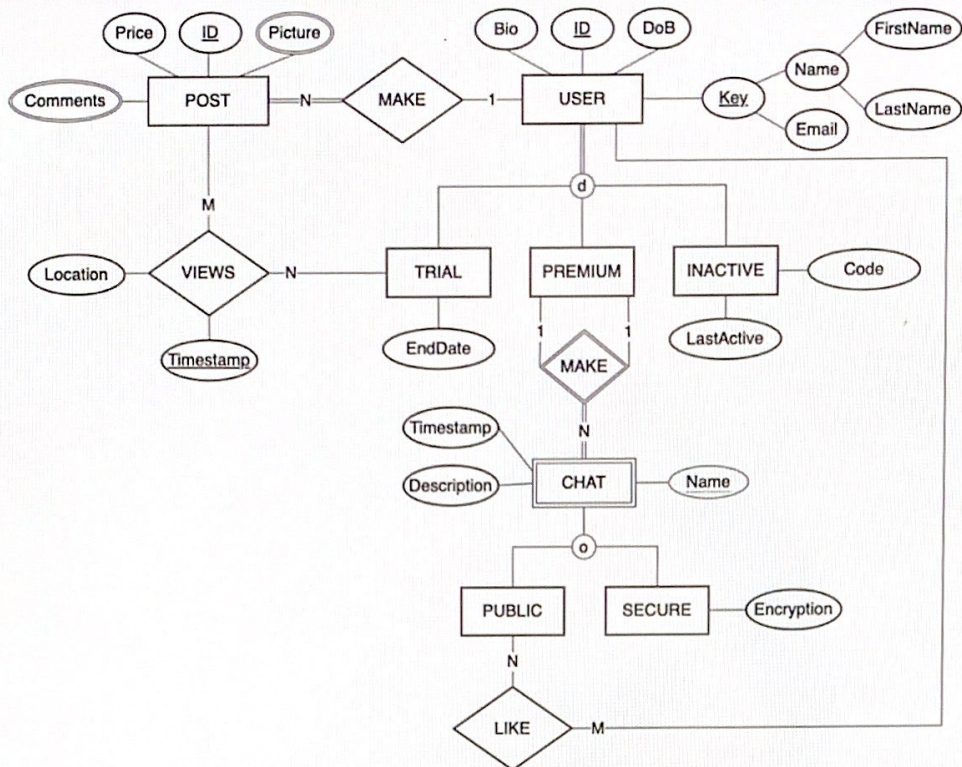
PrivilegedPrivilege[username, privileges]

PrivilegedPrivilege.username references User.username

Administrator[username]

Administration.username references User.username

### B.3



User[ID, key, bio, dob]

Post[ID, key, price, user]  
 Post.key references User.key  
 PostUser references User.ID

PostPic[ID, picture]  
 PostPic.ID references Post.ID

PostCom[ID, comments]  
 PostCom.ID references Post.ID

Views[ID, timestamp, location] key?  
 Views.ID references Post.ID

Trial[ID, key, EndDate]  
 Trial.ID references User.ID  
 Trial.key references User.key

Inactive[ID, key, code, LastActive]  
 Inactive.ID references User.ID  
 Inactive.key references User.key

Premium[ID, key]  
 Premium.ID references User.ID  
 Premium.key references User.key

Chat[Name, timestamp, description, premium]  
 Chat.premium references Premium.ID

Public[Name]  
 Public.Name references Chat.Name

Secure[Name, encryption]  
 Secure.Name references Chat Name

Like[Name, ID, key]  
 Like Name references Chat name  
 Like ID references User.ID  
 Like key references User.key