COMS W4111: Introduction to Databases Spring 2023, Sections 002, V02

</span>

Non-Programming Track, HW2, Part 2

Introduction

Environment

- Test environment.
- Set your MySQL user and password below.

In [1]:

```
mysql_user = "root"
mysql_pw = "dbuserbdbuser"
```

In [2]:

```
%load_ext sql
```

In [3]:

```
full_url = f"mysql+pymysql://{mysql_user}:{mysql_pw}@localhost"
full_url
```

Out[3]:

```
'mysql+pymysql://root:dbuserbdbuser@localhost'
```

In [4]:

```
%sql $full_url
```

In [5]:

```
%sql select * from db_book.student;
```

```
 * mysql+pymysql://root:***@localhost
13 rows affected.
```

Out[5]:

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60

98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

Submission Instructions

- See Ed for instructions.

Data and Scheme Cleanup

characters and name_basics_all

- The task is to "clean up" `characters` and produce a table `charactersFixed`.

- The task will require adding missing rows to `name_basics_all`. There are two row's in `characters` that have an `actorLink` and `actorName` got which there is no matching row in `name_basics_all`.

- `characters` has two actors with actorNames `Barry John O'Connor and Barry O'Connor` who are the same actor.

- My `charactersFixed` has the following columns:
  - `characterId` is a generated primary key. See below for an explanation.
  - `characterName` : The value from `characters`.
  - `characterImdbID` : The `characterLink` from `characters` with `/character/` removed.
  - `characterLink` : The `characterLink` from `characters`.
  - `actorNConst` : `actorLink` from `characters`.
  - `actorLink` : A value of the form `/names/` followed by the `actorNConst`.
  - `characterImageFull` : The value from `characters`.
  - `characterImageThumb` : The value from `characters`.
  - `kingsguard` : The value from `characters`.
  - `royal` : The value from `characters`.

- The algorithm for generating the `characterID` on insert is the following:
  - The prefix for the `character` is either:
    - The substring of `characterName` preceeding the first ' '.
    - The `characterName` is there is no ' '.
  - If there are `N` rows in the table, the number after the prefix is `N+1`.
  - Implementing this is tricky. Your first attempt might rely on `auto-increment`, but this does not work. You may also be tempted to count rows, but that does not work. A hint is that you will need to use a trigger and some other table/data that you create.

- The directory with this notebook containers data from my version of `charactersFixed`.

- The cells below load the data to allow you to examine. In your SQL table, `NaN` will be `NULL`.

In [6]:

```
import pandas as pd
```

In [7]:

```
characters_df = pd.read_csv('./charactersFixed.csv')
```

In [45]:

```
characters_df[['characterId','characterName']]
```

Out[45]:

	characterId	characterName
0	Addam1	Addam Marbrand
1	Aegon2	Aegon Targaryen
2	Aeron3	Aeron Greyjoy
3	Aerys4	Aerys II Targaryen
4	Akho5	Akho
...	...	...
384	Young385	Young Nan
385	Young386	Young Ned
386	Young387	Young Ned Stark
387	Young388	Young Rodrik Cassel
388	Zanrush389	Zanrush

389 rows × 2 columns

- Your answer below should show all of your SQL statements, including DDL, for creating and loading `charactersFixed` as well as changes to `name_basics_all` .
- You can use the data in the CSV file to test your work. Show at least one test.

In [9]:

```
%%sql

use s23_w4111_hw2_yz4366

/* SQL statements in this cell. You may use multiple cells. */
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
```

Out[9]:

```
[]
```

In [76]:

```
%%sql
select *
from characters
limit 20
```

```
 * mysql+pymysql://root:***@localhost
20 rows affected.
```

Out[76]:

characterLink	characterName	actorLink	actorName	
/character/ch0305333/	Addam Marbrand	nm0389698	B.J. Hogg	
None	Aegon Targaryen	None	None	
/character/ch0540081/	Aeron Greyjoy	nm0269923	Michael Feast	amazon.com/images/M/MV5
/character/ch0541362/	Aerys II Targaryen	nm0727778	David Rintoul	amazon.com/images/M/MV5BMWQzOWViN2ItNDZh

/character/ch0544520/	Akho	nm6729880	Chuku Modu	amazon.com/images/M/MV5BO(
/character/ch0246938/	Alliser Thorne	nm0853583	Owen Teale	https://images-na.ssl-images-amazon.com/im
/character/ch0305012/	Alton Lannister	nm0203801	Karl Davies	https://images-na.ss
/character/ch0576836/	Alys Karstark	nm8257864	Megan Parkinson	
/character/ch0305002/	Amory Lorch	nm0571654	Fintan McKeown	amazon.com/images/M/MV5BC
/character/ch0316930/	Anguy	nm1528121	Philip McGinley	amazon.com/images/M/MV5BN
/character/ch0578265/	Archmaester Marwyn	nm0000980	Jim Broadbent	
/character/ch0507107/	Areo Hotah	nm0649046	Deobia Oparei	amazon.com/images/M/MV5BN\
/character/ch0305014/	Armeca	nm1783582	Sahara Knite	amazon.com/images/M/MV5BZr
/character/ch0305326/	Arthur	nm8127149	Nathanael Saleh	
/character/ch0540097/	Arthur Dayne	nm1074361	Luke Roberts	amazon.com/images/M/MV5BOI
/character/ch0158604/	Arya Stark	nm3586035	Maisie Williams	https://images-na.ssl-images-amazon.com/image
/character/ch0547881/	Baby Sam	None	None	
/character/ch0292152/	Balon Greyjoy	nm0538869	Patrick Malahide	https://images-na.ssl-images-amazon.com/image
/character/ch0350989/	Baratheon Guard	nm4207240	Phil Barnhill	
/character/ch0241346/	Barristan Selmy	nm0568400	Ian McElhinney	https://images-na.ssl-images-amazon.com/image

In [72]:

```
%%sql
drop table if exists charactersFixed;
create table charactersFixed (
  characterId VARCHAR(255) PRIMARY KEY,
  characterName VARCHAR(255),
  characterImdbID VARCHAR(255),
  characterLink VARCHAR(255),
  actorNConst VARCHAR(255),
  actorLink VARCHAR(255),
  characterImageFull VARCHAR(255),
  characterImageThumb VARCHAR(255),
  kingsguard VARCHAR(255),
  royal VARCHAR(255)
);
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
```

Out[72]:

```
[]
```

In [73]:

```
%%sql
```

```
insert into charactersFixed (
  characterID, characterName, characterImdbID, characterLink,
  actorNConst, actorLink, characterImageFull,
  characterImageThumb, kingsguard, royal
)
select
  CONCAT(substring_index(characterName,' ',1), (ROW_NUMBER() over (order by characterName)
  characterName, TRIM(TRAILING '/' FROM REPLACE(characterLink, '/character/', '')), charac
  actorLink, CONCAT('/names/', actorLink), characterImageFull,
  characterImageThumb, kingsguard, royal
from characters;
```

```
 * mysql+pymysql://root:***@localhost
389 rows affected.
```

Out[73]:

```
[]
```

In [75]:

```
%%sql

select *
from charactersFixed
limit 20
/* SQL test to show result. */
```

```
 * mysql+pymysql://root:***@localhost
20 rows affected.
```

Out[75]:

characterId	characterName	characterImdbID	characterLink	actorNConst	actorLink	
Addam1	Addam Marbrand	ch0305333	/character/ch0305333/	nm0389698	/names/nm0389698	
Aegon2	Aegon Targaryen	None	None	None	None	
Aeron3	Aeron Greyjoy	ch0540081	/character/ch0540081/	nm0269923	/names/nm0269923	
Aerys4	Aerys II Targaryen	ch0541362	/character/ch0541362/	nm0727778	/names/nm0727778	ama
Akho5	Akho	ch0544520	/character/ch0544520/	nm6729880	/names/nm6729880	
Alliser6	Alliser Thorne	ch0246938	/character/ch0246938/	nm0853583	/names/nm0853583	
Alton7	Alton Lannister	ch0305012	/character/ch0305012/	nm0203801	/names/nm0203801	
Alys8	Alys Karstark	ch0576836	/character/ch0576836/	nm8257864	/names/nm8257864	
Amory9	Amory Lorch	ch0305002	/character/ch0305002/	nm0571654	/names/nm0571654	
Anguy10	Anguy	ch0316930	/character/ch0316930/	nm1528121	/names/nm1528121	
Archmaester11	Archmaester Marwyn	ch0578265	/character/ch0578265/	nm0000980	/names/nm0000980	
Areo12	Areo Hotah	ch0507107	/character/ch0507107/	nm0649046	/names/nm0649046	
Armeca13	Armeca	ch0305014	/character/ch0305014/	nm1783582	/names/nm1783582	
Arthur14	Arthur	ch0305326	/character/ch0305326/	nm8127149	/names/nm8127149	
Arthur15	Arthur Dayne	ch0540097	/character/ch0540097/	nm1074361	/names/nm1074361	
Arya16	Arya Stark	ch0158604	/character/ch0158604/	nm3586035	/names/nm3586035	htt

Baby17	Baby Sam	ch0547881	/character/ch0547881/	None	None	
Balon18	Balon Greyjoy	ch0292152	/character/ch0292152/	nm0538869	/names/nm0538869	ht
Baratheon19	Baratheon Guard	ch0350989	/character/ch0350989/	nm4207240	/names/nm4207240	
Barristan20	Barristan Selmy	ch0241346	/character/ch0241346/	nm0568400	/names/nm0568400	ht

In [37]:

```
characters_df
```

Out[37]:

	characterId	characterName	characterImdbID	characterLink	actorNconst	actorLink	
0	Addam1	Addam Marbrand	ch0305333	/character/ch0305333	nm0389698	/names/nm0389698	
1	Aegon2	Aegon Targaryen	NaN	NaN	NaN	NaN	
2	Aeron3	Aeron Greyjoy	ch0540081	/character/ch0540081	nm0269923	/names/nm0269923	h a
3	Aerys4	Aerys II Targaryen	ch0541362	/character/ch0541362	nm0727778	/names/nm0727778	h a
4	Akho5	Akho	ch0544520	/character/ch0544520	nm6729880	/names/nm6729880	h a
...	...	...	...	...	...	...	
384	Young385	Young Nan	ch0305018	/character/ch0305018	nm1519719	/names/nm1519719	
385	Young386	Young Ned	ch0154681	/character/ch0154681	nm7075019	/names/nm7075019	
386	Young387	Young Ned Stark	ch0154681	/character/ch0154681	nm7509185	/names/nm7509185	
387	Young388	Young Rodrik Cassel	ch0171391	/character/ch0171391	nm7509186	/names/nm7509186	
388	Zanrush389	Zanrush	ch0540870	/character/ch0540870	nm0503319	/names/nm0503319	h a

389 rows × 10 columns

name_basics_all

- The column `primaryProfessions` is multi-valued and non-atomic. This violates good relational design principle.
- Create a new table `name_basics_all_fixed` which does not have the column `primaryProfessions`.
- You will need to use SQL to create and load other tables with information from `name_basics_all` to enable you to create a view `name_basics_all_fixed_view` that recreates the data in `name_basics_all`. The tables you create should have atomic columns, primary keys and foreign keys, etc.

In [77]:

```
%%sql
```

```
select *
from name_basics_all
limit 20
/* Use this cell and others to create tables, load data, etc. */
```

```
 * mysql+pymysql://root:***@localhost
20 rows affected.
```

Out[77]:

nconst	primaryName	birthYear	deathYear	primaryProfession	knownF
nm0389698	B.J. Hogg	1955	2020	actor,music_department	tt0970411,tt0944947,tt0986233,tt
nm0269923	Michael Feast	1946	None	actor,composer	tt0472160,tt0162661,tt0120879,tt
nm0727778	David Rintoul	1948	None	actor	tt1655420,tt1139328,tt4786824,tt
nm6729880	Chuku Modu	1990	None	actor,writer,producer	tt4154664,tt2674426,tt6470478,tt
nm0853583	Owen Teale	1961	None	actor	tt0102797,tt0485301,tt0462396,tt
nm0203801	Karl Davies	1982	None	actor,producer	tt12879632,tt7366338,tt3428912,tt
nm8257864	Megan Parkinson	None	None	actress	tt0944947,tt6636246,tt5761478,tt
nm0571654	Fintan McKeown	None	None	actor	tt0944947,tt0111904,tt0166396,t
nm1528121	Philip McGinley	1981	None	actor	tt3922704,tt0053494,tt0944947,tt
nm0000980	Jim Broadbent	1949	None	actor,writer,soundtrack	tt0203009,tt1431181,tt1007029,tt
nm0649046	Deobia Oparei	1971	None	actor	tt0419706,tt0118929,tt0203009,tt
nm1783582	Sahara Knite	1975	None	actress	tt15288590,tt15249564,tt0944947,tt
nm8127149	Nathanael Saleh	None	None	actor,soundtrack	tt0944947,tt1635327,tt3498954,tt
nm1074361	Luke Roberts	1977	None	actor,producer,writer	tt2375692,tt5809150,tt0944947,tt
nm3586035	Maisie Williams	1997	None	actress,producer,soundtrack	tt0944947,tt1330018,tt3294200,tt
nm0538869	Patrick Malahide	1945	None	actor,writer,soundtrack	tt0090521,tt0120873,tt0116908,tt
nm4207240	Phil Barnhill	None	None	actor	tt
nm0568400	Ian McElhinney	1948	None	actor,director,soundtrack	tt0944947,tt3748528,tt0117039,tt
nm1152798	Joseph Mawle	1974	None	actor	tt0944947,tt1390411,tt1127876,tt
nm4730958	Eline Powell	1990	None	actress	tt2884308,tt1972591,tt2091298,tt

In [82]:

```
%sql describe name_basics_all
```

```
 * mysql+pymysql://root:***@localhost
6 rows affected.
```

Out[82]:

Field	Type	Null	Key	Default	Extra
nconst	varchar(16)	NO		None	
primaryName	text	YES		None	
birthYear	text	YES		None	
deathYear	text	YES		None	
primaryProfession	text	YES		None	

knownForTitles	text	YES	None

In [87]:

```
%%sql
with one as (
    select
        primaryProfession,
        replace(primaryProfession, ',', '') as no_comma
    from
        name_basics_all
),
    two as (select primaryProfession, length(primaryProfession) - length(no_comma) as comr
            from one)
select
    comma_count, count(*) as profession_comma_space_count
from
    two
group by comma_count order by profession_comma_space_count asc;
```

```
 * mysql+pymysql://root:***@localhost
4 rows affected.
```

Out[87]:

comma_count	profession_comma_space_count
None	2
1	68
2	97
0	183

In [103…

```
%%sql
drop table if exists profession;
create table profession (
  nconst varchar(255) not null,
  primaryProfession varchar(255),
  profession_1 varchar(255),
  profession_2 varchar(255),
  profession_3 varchar(255),
  primary key (nconst)

);
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
```

Out[103…

```
[]
```

In [104…

```
%%sql

insert into profession(
nconst, primaryProfession, profession_1, profession_2, profession_3)
select
    nconst, primaryProfession,
    SUBSTRING_INDEX(SUBSTRING_INDEX(CONCAT(primaryProfession, ','), ',', 1), ',', -1),
    SUBSTRING_INDEX(SUBSTRING_INDEX(CONCAT(primaryProfession, ','), ',', 2), ',', -1),
    SUBSTRING_INDEX(SUBSTRING_INDEX(CONCAT(primaryProfession, ','), ',', 3), ',', -1)
from name_basics_all
```

```
 * mysql+pymysql://root:***@localhost
350 rows affected.
```

Out[104…

```
[]
```

In [97]:

```
%%sql
drop table if exists name_basics_all_fixed;
create table name_basics_all_fixed as
select nconst, primaryName, birthYear, deathYear, knownForTitles
from name_basics_all
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
350 rows affected.
```

Out[97]:

```
[]
```

In [119…

```
%%sql
alter table name_basics_all
add primary key(nconst)
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
```

Out[119…

```
[]
```

In [106…

```
%%sql
alter table name_basics_all_fixed
add primary key (nconst)
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
```

Out[106…

```
[]
```

In [120…

```
%%sql
alter table name_basics_all_fixed
add constraint fk
foreign key (nconst)
references name_basics_all (nconst)
```

```
 * mysql+pymysql://root:***@localhost
350 rows affected.
```

Out[120…

```
[]
```

In [121…

```
%%sql
alter table profession
add constraint fk_profession
foreign key (nconst)
references name_basics_all (nconst)
```

```
 * mysql+pymysql://root:***@localhost
350 rows affected.
```

Out[121…

```
[]
```

In [109…

```
%%sql
drop view if exists name_basics_all_fixed_view;
create view name_basics_all_fixed_view as
select name_basics_all_fixed.nconst, primaryName, birthYear, deathYear, knownForTitles, pr
from name_basics_all_fixed
join profession
on name_basics_all_fixed.nconst = profession.nconst
```

```
 * mysql+pymysql://root:***@localhost
```

```
0 rows affected.
```

Out[109…

```
[]
```

In [110…

```
%%sql
select *
from name_basics_all_fixed_view
limit 20
```

```
 * mysql+pymysql://root:***@localhost
20 rows affected.
```

Out[110…

nconst	primaryName	birthYear	deathYear	knownForTitles	profession_1	profe
nm0000293	Sean Bean	1959	None	tt0120737,tt0167261,tt0944947,tt1181791	actor	p
nm0000596	Jonathan Pryce	1947	None	tt0104348,tt8404614,tt0120347,tt3750872	actor	sou
nm0000980	Jim Broadbent	1949	None	tt0203009,tt1431181,tt1007029,tt0217505	actor	
nm0001097	Charles Dance	1946	None	tt0944947,tt0107362,tt2084970,tt0280707	actor	
nm0001290	Richard E. Grant	1957	None	tt4595882,tt0280707,tt0102070,tt0094336	actor	sou
nm0001354	Ciarán Hinds	1953	None	tt1340800,tt1596365,tt1201607,tt12789558	actor	sou
nm0001671	Diana Rigg	1938	2020	tt0054518,tt9639470,tt0064757,tt0944947	actress	sou
nm0002103	Julian Glover	1935	None	tt0082398,tt0332452,tt0080684,tt0097576	actor	sou
nm0004355	Roger Ashton-Griffiths	1957	None	tt0088846,tt0944947,tt4575576,tt0217505	actor	
nm0004692	Mark Addy	1964	None	tt0944947,tt0955308,tt0119164,tt0183790	actor	sou
nm0015382	Adewale Akinnuoye-Agbaje	1967	None	tt1127881	actor	
nm0019885	Roger Allam	1953	None	tt1486190,tt0811080,tt0434409,tt1298650	actor	sou
nm0050520	Peter Ballance	None	None	tt0408056,tt0944947,tt0323033,tt1985443	actor	misce
nm0050959	Pedro Pascal	1975	None	tt4649466,tt7126948,tt0944947,tt8111088	actor	sou
nm0057965	Clifford Barry	None	None	tt0118363,tt0944947,tt0094535,tt0189192	actor	
nm0064155	Ian Beattie	1965	None	tt2303687,tt0944947,tt0346491,tt2567712	actor	
nm0065874	Andy Beckwith	None	None	tt5691024,tt0208092,tt1355631,tt0944947	actor	
nm0072855	Paul Bentley	None	None	tt8697870,tt0944947,tt0185906,tt1007029	actor	
nm0087432	Nicholas Blane	None	None	tt2343137,tt0443543,tt0373889,tt0944947	actor	
nm0103195	David Bradley	1942	None	tt1213663,tt0241527,tt1201607,tt0425112	actor	sou

In [ ]: