Introduction to Databases - Application

July 2022

UBC MFRE | Master of Food and Resource Economics

Roadmap

- Conceptual Overview of Databases
- Big picture of database design
- Learn basic SQL queries with Harry Potter Database
- How data analysts use databases: Steven Zhang

Learning Outcomes

- Identify the code required to utilize data

- Write SQL queries to
  - Select data
  - Sort and remove duplicates
  - Filter values
  - Calculate new values
  - Aggregate data
  - Combine or join tables

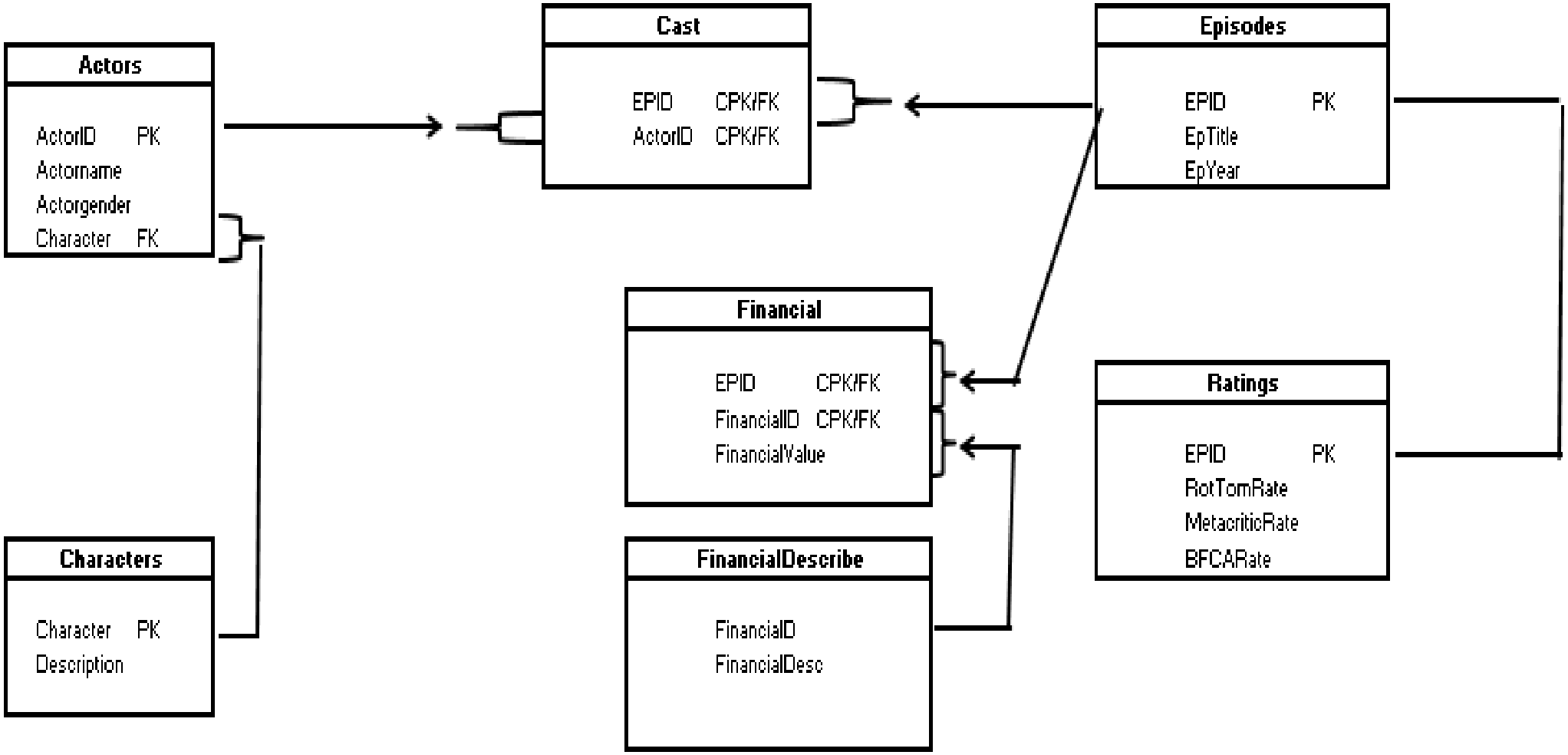
Basic Commands

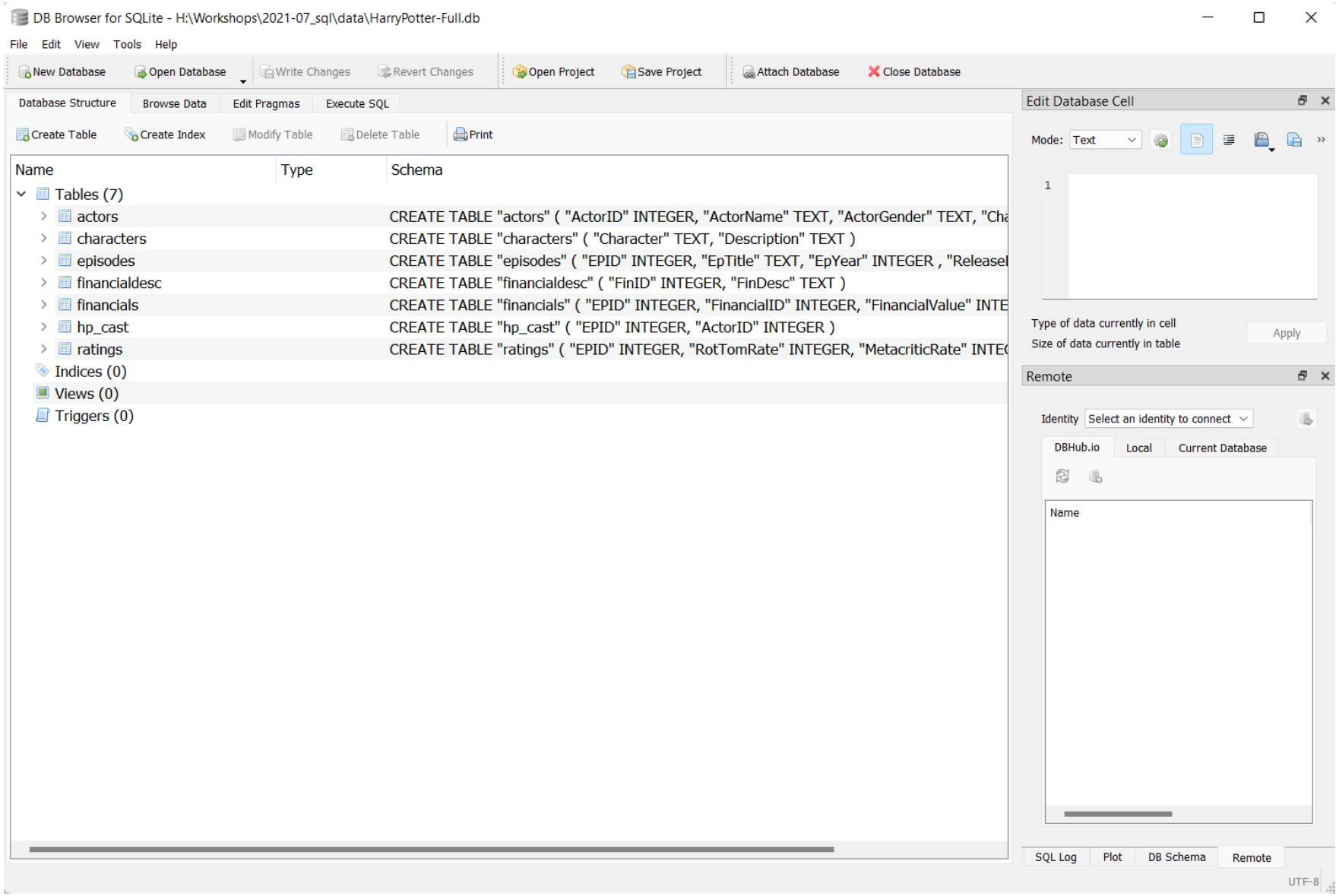
- Selecting data - SELECT, FROM, WHERE, LIKE
- Aggregation - min, max, avg, count, GROUP BY
- Combining data - JOIN, ON

Using DB Browser

- As a data analyst, you will most likely work with an existing database, so it will be to your advantage to become familiar with SQL commands.

- Open up the HarryPotter.db on DB Browser for SQLite
  - https://sqliteonline.com/


Actors
ActorID PK
Actorname
Actorgender
Character FK
Cast
EPID CPK/FK
ActorID CPK/FK
Episodes
EPID PK
EpTitle
EpYear
Financial
EPID CPK/FK
FinancialID CPK/FK
FinancialValue
Ratings
EPID PK
RotTomRate
MetacriticRate
BFCARate
Characters
Character PK
Description
FinancialDescribe
FinancialID
FinancialDesc


DB Browser for SQLite - H:\Workshops\2021-07_sql\data\HarryPotter-Full.db
File Edit View Tools Help
New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Create Table
Create Index
Modify Table
Delete Table
Print
Name
Type
Schema
Tables (7)
actors
CREATE TABLE "actors" ("ActorID" INTEGER, "ActorName" TEXT, "ActorGender" TEXT, "Cha
characters
CREATE TABLE "characters" ("Character" TEXT, "Description" TEXT)
episodes
CREATE TABLE "episodes" ("EPID" INTEGER, "EpTitle" TEXT, "EpYear" INTEGER , "Releasel
financialdesc
CREATE TABLE "financialdesc" ("FinID" INTEGER, "FinDesc" TEXT)
financials
CREATE TABLE "financials" ("EPID" INTEGER, "FinancialID" INTEGER, "FinancialValue" INTE
hp_cast
CREATE TABLE "hp_cast" ("EPID" INTEGER, "ActorID" INTEGER)
ratings
CREATE TABLE "ratings" ("EPID" INTEGER, "RotTomRate" INTEGER, "MetacriticRate" INTE
Indices (0)
Views (0)
Triggers (0)
Edit Database Cell
Mode: Text
1
Type of data currently in cell
Size of data currently in table
Apply
Remote
Identity
Select an identity to connect
DBHub.io
Local
Current Database
Name
SQL Log
Plot
DB Schema
Remote
UTF-8

Selecting data

- A typical query has the form

```
SELECT <column1>, <column2>, …
FROM <table name>
WHERE <CONDITIONS>;
```

- The result is a new table called the result set.

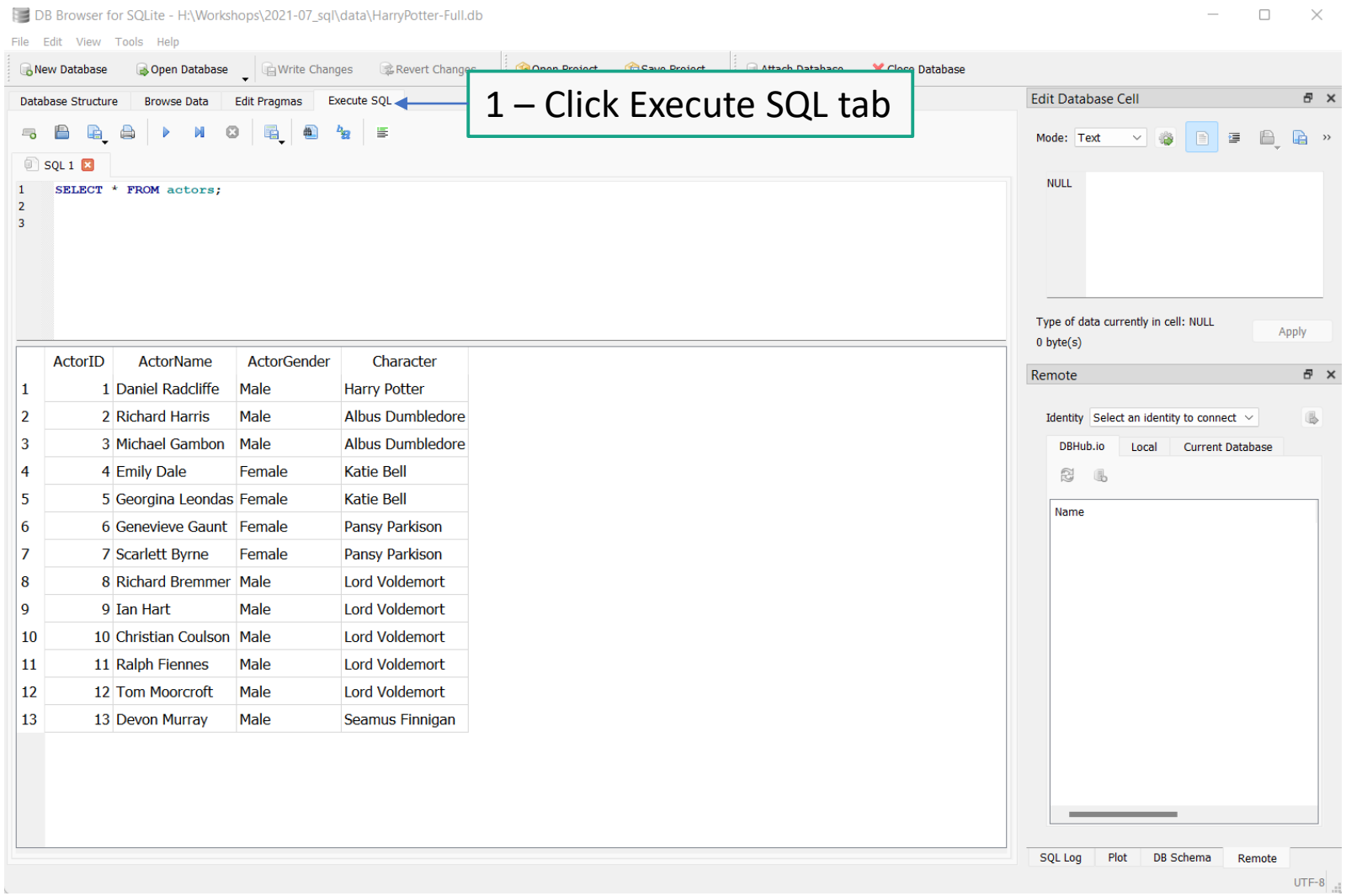
Selecting Data

- How do we know all the columns and data of a particular table?
  - Use an asterisk (*) to select all the columns of the queried table(s)
    - SELECT * FROM actors;
  - What if we only want to see 5 data points?
    - SELECT * FROM actors LIMIT 5;

Working with DB for SQLite

2 – Type your queries then ctrl + enter or cmd + enter

3 – see output here



	ActorID	ActorName	ActorGender	Character
1	1	Daniel Radcliffe	Male	Harry Potter
2	2	Richard Harris	Male	Albus Dumbledore
3	3	Michael Gambon	Male	Albus Dumbledore
4	4	Emily Dale	Female	Katie Bell
5	5	Georgina Leondas	Female	Katie Bell
6	6	Genevieve Gaunt	Female	Pansy Parkison
7	7	Scarlett Byrne	Female	Pansy Parkison
8	8	Richard Bremmer	Male	Lord Voldemort
9	9	Ian Hart	Male	Lord Voldemort
10	10	Christian Coulson	Male	Lord Voldemort
11	11	Ralph Fiennes	Male	Lord Voldemort
12	12	Tom Moorcroft	Male	Lord Voldemort
13	13	Devon Murray	Male	Seamus Finnigan

Selecting data

- What is the gender distribution of our actors database?

Selecting data

- What is the gender distribution of our actors database?

```
SELECT ActorName, ActorGender FROM actors;
```

Selecting Data

- Use AS to name columns of the result set

```
SELECT ActorName AS Name,
       ActorGender AS Gender
       FROM actors;
```

Activity – Selecting Data

- What are the names of the characters that we have actor data of?

Sorting and Removing Duplicates

- Database records are not stored in a particular order.
  - If we want the query results to be ordered in a particular way, we can use the ORDER BY keyword.

```
SELECT Character, ActorName FROM actors
ORDER BY ActorName (ASC/DESC);
```

```
SELECT Character, ActorName FROM actors
ORDER BY Character ASC, ActorName DESC;
```

Filtering rows with WHERE

- What if we want to see the different actors who played Lord Voldemort?

```
SELECT * FROM actors
WHERE Character = 'Lord Voldemort';
```

Filtering with wildcards

- To filter by partial matches, we use the LIKE keyword. The % symbol acts as a wildcard, matching any characters in that place.

- For example, we want to know the title and release year of the episode that contains “Deathly”

```
SELECT * FROM episodes WHERE EpTitle LIKE '%Deathly%';
```

Filtering

- We can use Boolean operators such as AND, OR, IN

- We can query which episodes were released from 2005 to 2007.

```
SELECT EpTitle FROM episodes WHERE EpYear >= 2005 AND EpYear <= 2007;
```

Activity

- Write a query that will display the episode ID and financial value of worldwide box office sales ordered in descending order
  - Hint financial id = 1 for worldwide box office sales

Basic Math & Stats in SQL

- We use the SELECT keyword (similar to di command in Stata)
  - SELECT 2 + 2;
  - SELECT 9 – 1;
  - SELECT 3 * 4;

Adding columns

- How do we create a new index called “TotalRating” that adds the Rotten Tomato rating and BFCA Rating?

```
SELECT RotTomRate, BFCARate,
RotTomRate + BFCARate AS "TotalRating"
FROM ratings;
```

Calculating new values

- How do we convert the box office sales from USD to CDN?

Calculating new values

- How do we convert the box office sales from USD to CDN?

```
SELECT *,
1.23 * FinancialValue AS "in CDN"
FROM financials;
```

Aggregation

- Common aggregate functions:
  - min(), max(), avg(), sum()

- Calculate average worldwide box office sales of all 8 episodes

```
SELECT round(avg(FinancialValue),0) FROM financials
WHERE FinancialID = 1;
```

Activity - Aggregation

- Which episode had the lowest worldwide box office sales?
- What is the average worldwide box office sales of all 8 episodes?
- How many different actors played Lord Voldemort?

Activity

- So far we have talked about
  - Select data
  - Sort and remove duplicates
  - Filter values
  - Calculate new values

- Using these statements, what queries can you write in a sales database?

Joining tables

- IN SQL, joins are used to link the tables within a relational database

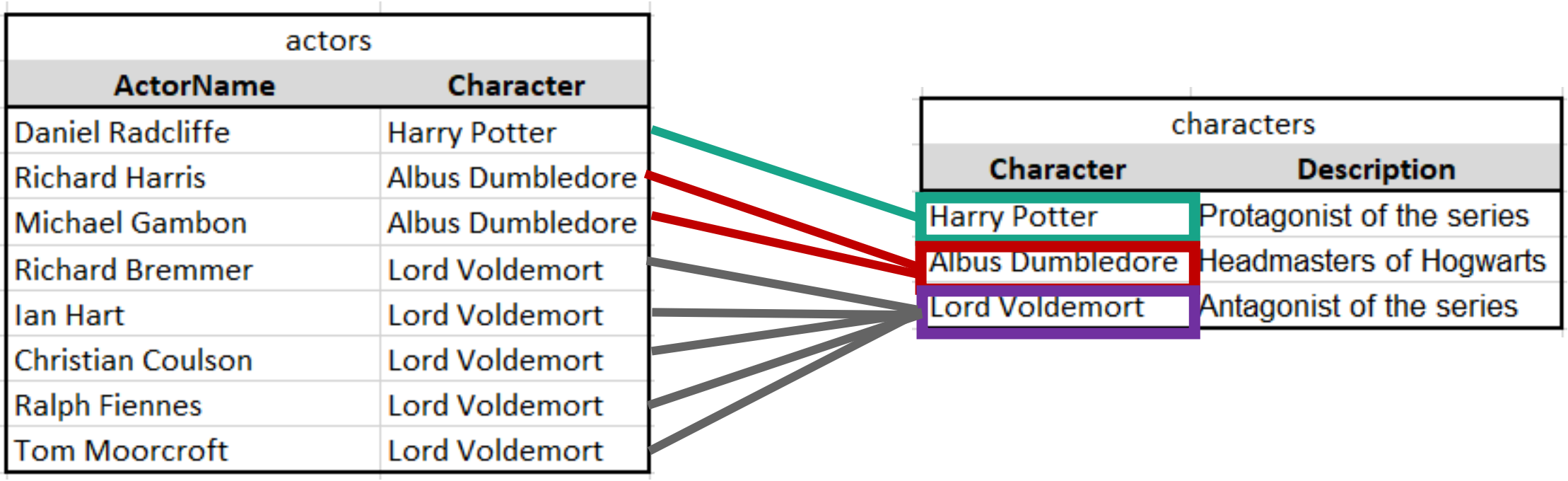
- Types of joins
  - JOIN or INNER JOIN
  - OUTER JOINs – LEFT*, RIGHT, FULL
  - CROSS JOIN

Review on keys

- What is a primary key?

- What is a foreign key?

actors - characters



actors

ActorName	Character
Daniel Radcliffe	Harry Potter
Richard Harris	Albus Dumbledore
Michael Gambon	Albus Dumbledore
Richard Bremmer	Lord Voldemort
Ian Hart	Lord Voldemort
Christian Coulson	Lord Voldemort
Ralph Fiennes	Lord Voldemort
Tom Moorcroft	Lord Voldemort

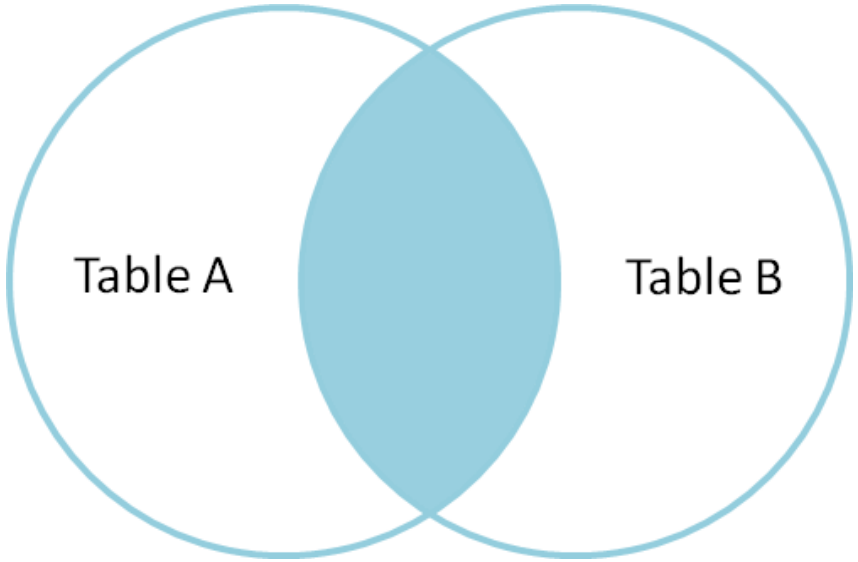
characters

Character	Description
Harry Potter	Protagonist of the series
Albus Dumbledore	Headmasters of Hogwarts
Lord Voldemort	Antagonist of the series

Inner join

- An inner join produces only the set of records that match in both Table A and Table B

```
SELECT * FROM tableA
INNER JOIN tableB
ON tableA.PK = tableB.FK;
```



Activity – Inner Join

- Write a query that returns the episode title and worldwide box office sales

- Write a query that gives us the name of the actors, the character they played, and a description of their characters

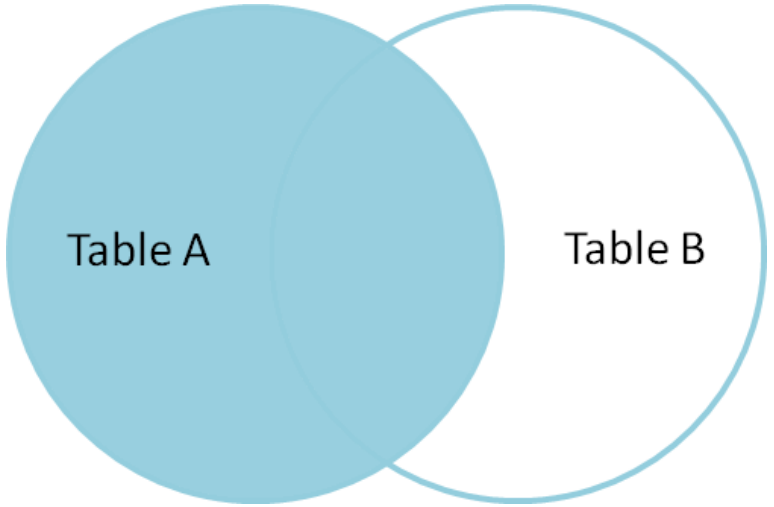
characters	
Character	Description
Harry Potter	Protagonist of the series
Albus Dumbledore	Headmasters of Hogwarts
Katie Bell	A Gryffindor student at Hogwarts
Pansy Parkison	A Slytherin student at Hogwarts
Lord Voldemort	Antagonist of the series
Hermione Granger	Best friend of Harry Potter
Ron Weasley	Best friend of Harry Potter
Severus Snape	Potions professor

actors	
ActorName	Character
Daniel Radcliffe	Harry Potter
Richard Harris	Albus Dumbledore
Michael Gambon	Albus Dumbledore
Emily Dale	Katie Bell
Georgina Leondas	Katie Bell
Genevieve Gaunt	Pansy Parkison
Scarlett Byrne	Pansy Parkison
Richard Bremmer	Lord Voldemort
Ian Hart	Lord Voldemort
Christian Coulson	Lord Voldemort
Ralph Fiennes	Lord Voldemort
Tom Moorcroft	Lord Voldemort
Devon Murray	Seamus Finnigan

Left join

- A left join produces a complete set of records from Table A, with matching records when available in Table B. If there is no match, the right side will contain null.

```
SELECT * FROM tableA
LEFT JOIN tableB
ON tableA.PK = tableB.FK;
```



Activity – Left join

- Write a query that will give us the result table on the right

characters	
Character	Description
Harry Potter	Protagonist of the series
Albus Dumbledore	Headmasters of Hogwarts
Katie Bell	A Gryffindor student at Hogwarts
Pansy Parkison	A Slytherin student at Hogwarts
Lord Voldemort	Antagonist of the series
Hermione Granger	Best friend of Harry Potter
Ron Weasley	Best friend of Harry Potter
Severus Snape	Potions professor

actors	
ActorName	Character
Daniel Radcliffe	Harry Potter
Richard Harris	Albus Dumbledore
Michael Gambon	Albus Dumbledore
Emily Dale	Katie Bell
Georgina Leondas	Katie Bell
Genevieve Gaunt	Pansy Parkison
Scarlett Byrne	Pansy Parkison
Richard Bremmer	Lord Voldemort
Ian Hart	Lord Voldemort
Christian Coulson	Lord Voldemort
Ralph Fiennes	Lord Voldemort
Tom Moorcroft	Lord Voldemort
Devon Murray	Seamus Finnigan

	Character	ActorName
1	Harry Potter	Daniel Radcliffe
2	Albus Dumbledore	Michael Gambon
3	Albus Dumbledore	Richard Harris
4	Katie Bell	Emily Dale
5	Katie Bell	Georgina Leondas
6	Pansy Parkison	Genevieve Gaunt
7	Pansy Parkison	Scarlett Byrne
8	Lord Voldemort	Christian Coulson
9	Lord Voldemort	Ian Hart
10	Lord Voldemort	Ralph Fiennes
11	Lord Voldemort	Richard Bremmer
12	Lord Voldemort	Tom Moorcroft
13	Hermione Granger	*NULL*
14	Ron Weasley	*NULL*
15	Severus Snape	*NULL*

Joining (more) data

- What if we want to show the episode title, financial description, box office sales?

```
SELECT episodes.EpTitle, financialdesc.FinDesc, financials.FinancialValue
FROM financials JOIN episodes JOIN financialdesc
ON financials.EPID = episodes.EPID
AND financials.FinancialID = financialdesc.FinID;
```

Activity

1. Create a query that shows Top 3 highest ranked episodes according to the Rotten Tomato rating
2. Write a SQL statement that displays the name of the actors, the episodes, and the rotten tomato rating for ratings with greater than 90
3. What were average Worldwide and North American box office sales of all 8 movie episodes?

.

Recap

- Write SQL queries to
  - Get data from database using SELECT
  - Sort and filter values using ORDER BY and DISTINCT
  - Subset data using WHERE and LIKE
  - Perform basic math and data aggregation using min(), avg(), count()
  - Join tables using INNER JOIN and LEFT JOIN to generate user-defined outputs