Data-Intensive Systems

Spring 2025

Lab 1 Description

SQL Developer – SQL IDE

- If not already set up in previous lab session

- Java client for the Oracle RDBMS
  - Available for download, ideally via the download link: https://www.oracle.com/tools/downloads/sqldev-downloads.html

- Most SQL IDEs work (that has JDBC connection –this should be standard):
  - DBeaver (open source): https://dbeaver.io/
  - JetBrains DataGrip (free for students): https://www.jetbrains.com/datagrip/
  - You may use other software if you prefer so

SQL Developer + Oracle DBMS

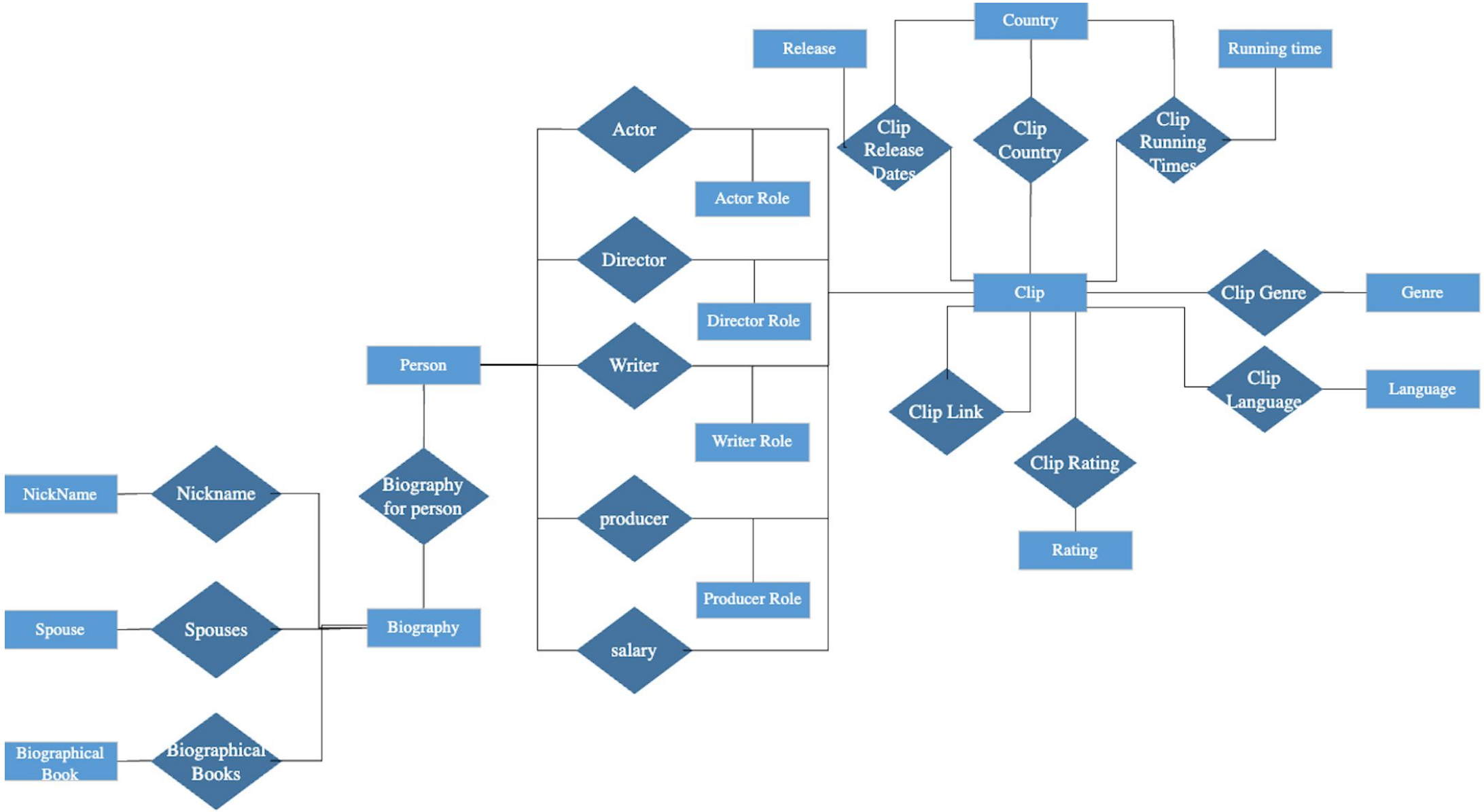
- Create a new connection with parameters:
  - username: C##DB2025_STUDENT
  - password: DB2025Lab1
  - hostname: cs322-db.epfl.ch
  - port: 1521
  - SID: ORCLCDB
  - Make sure you are connected to the EPFL network, or that you are using a VPN!
- All table names should be prefixed with C##DB2025
- The query file must contain ONLY the SQL query (1 statement)

The IMBD Dataset

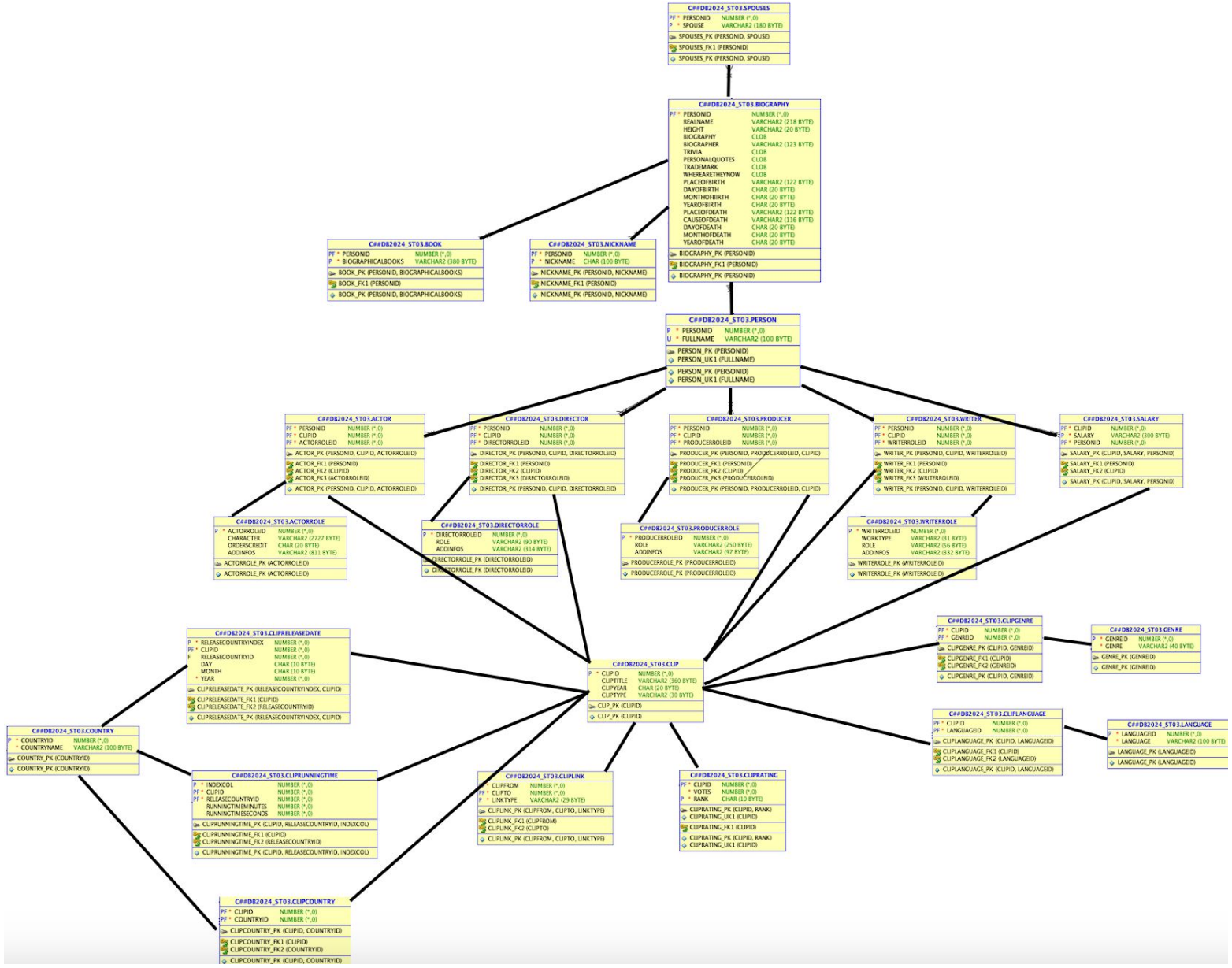
- Actors, Directors, Writers, Producers
- Clips:
  - rating, link, genre, language, country, running times, release dates

- Check lab 1 description for more details on the ER, attributes and values

The ER


Release
Country
Running time
Actor
Clip Release Dates
Clip Country
Clip Running Times
Actor Role
Director
Director Role
Clip
Clip Genre
Genre
Person
Writer
Clip Link
Clip Language
Language
Writer Role
Clip Rating
NickName
Nickname
Biography for person
producer
Rating
Producer Role
Spouse
Spouses
Biography
salary
Biographical Book
Biographical Books

The Database tables



Lab 1 Parts and Methodology

- Analyze the tables
- Understand the entities and the ER model
- Get familiar with the database

- Write a series of queries

The queries

1. **For each FULLNAME, compute the total number of distinct WRITERROLEID values across all clips. Print FULLNAME and the total count of unique WRITERROLEIDs. Order first by the total number of distinct WRITERROLEID descending, then by FULLNAME ascending, and print only the first 10 rows. Output should be in the format FULLNAME, TOTAL_WRITER_ROLES.**

2. **Print the top 10 languages that appear in the greatest number of CLIPs based on CLIPLANGUAGE. Print LANGUAGE and the count of CLIPs as CLIP_COUNT. Order by CLIP_COUNT descending. Output should be in the format LANGUAGE, CLIP_COUNT.**

3. **For each CLIPID, compute the total number of directors who co-directed the clip (i.e., had the 'co-director' role). Print CLIPID and the total count of co-directors. Order first by the total number of co-directors descending, then by CLIPID ascending, and print only the first 10 rows. Output should be in the format CLIPID, NUM_CO_DIRECTORS.**

The queries

4. **Find the 10 clips with the maximum total running time. Compute total running time in seconds using RUNNINGTIMEMINUTES and RUNNINGTIMESECONDS. Print CLIPID, RELEASECOUNTRYID, and total running time in seconds. Order by TOTAL_RUNNINGTIME descending. Output should be in the format CLIPID, RELEASECOUNTRYID, TOTAL_RUNNINGTIME.**

5. **Compute the average running time (in seconds) of clips in each country. Convert minutes to seconds and add RUNNINGTIMESECONDS. Print COUNTRYNAME and the computed average. Round the average to seconds using the ROUND function. Order by COUNTRYNAME ascending and print only the first 10 rows. Output should be in the format COUNTRYNAME, AVERAGE_RUNNINGTIME_SECONDS.**

The queries

6. **For each PERSONID, compute the total number of distinct ACTORROLEID values across all clips. Print PERSONID and the total count of unique ACTORROLEIDs. Order by the total descending and print only the first 10 rows. Output should be in the format PERSONID, DISTINCT_ACTOR_ROLES.**

7. **For each CLIPID, compute the total number of languages associated with the clip. Print CLIPID and the total count of languages. Order first by the total number of languages descending, then by CLIPID ascending, and print only the first 10 rows. Output should be in the format CLIPID, NUM_LANGUAGES.**

8. **Identify the languages with the largest number of distinct clips featuring at least one actor in a leading role (ACTORROLEID = 1). Print LANGUAGE and the total count of such clips. Order by the total descending and print only the first 10 rows. Output should be in the format LANGUAGE, CLIP_COUNT.**

The queries

9. **Find all PERSONIDs that appear as both a director and a producer for a clip for at least 2 distinct clips. Print PERSONID only. Order by PERSONID ascending and print only the first 10 rows. Output should be in the format PERSONID.**

10. **For each PERSONID, compute the total number of distinct clips in which they appear in any of the following roles: ACTOR, WRITER, or DIRECTOR. Print PERSONID and the total count of unique clips. Order by the total descending and print only the first 10 rows.Output should be in the format PERSONID, TOTAL_CLIPS.**

Individual Assignment

- The lab assignment is individual
- The SQL code you write should be your own and should not be copied from someone else, or the internet
- We will run plagiarism checks

Lab 1 Deadline

Lab 1 Submission 25/03

	Lecture (Monday, 11am, CE14)		Books chapters	Lab session (Wednesday, 4pm BCH2201)	Lab release	Lab deadline
	Date	**Topic**		**Date**		
Week 1	Feb 17	Intro, overview: ER	Chapter 1.1-1.5 6.1-6.9	Feb 19		
Week 2	Feb 24	Relational Model & Relational Algebra & SQL	Chapter 2,3.1-3.7	Feb 26	Lab 1: 1am Feb 24	
Week 3	Mar 3	Storage, Files, and Indexing	Chapter 13.1-13.4, 14.1, 14.2	Mar 5		
Week 4	Mar 10	Storage and Buffer Management	Chapter 12.1-12.5 13.5	Mar 12		
Week 5	Mar 17	Indexes: B-Tree	Chapter 14.1-14.4	Mar 19		
Week 6	Mar 24	Hashing / Sorting	Chapter 14.5, 24.5, 15.4	Mar 26	Lab 2: 1am Mar 24	Lab 1: 11.59pm Mar 25
Week 7	Mar 31	Query Operators I (not included in midterm)	Chapter 15.3, 15.5.1-15.5.2, 15.6.1-15.6.2, 15.7	Apr 2		
Week 8	Apr 7	**Midterm**		Apr 9		
Week 9	Apr 14	Query Operators II	Chapter 15.5.3 - 15.5.6 & 15.6.3 & 15.6.5	Apr 16		
Week 10	Apr 21	**Spring break**		Apr 23		
Week 11	Apr 28	Query Optimization	Chapter 16	Apr 30		
Week 12	May 5	Transactions and Concurrency Control & Concurrency I	Chapter 17, 18	May 7	Lab 3: 9 AM May 5	Lab 2: 11.59pm May 5
Week 13	May 12	Concurrency Control and Eventual Consistency & Concurrency II	Chapter 17, 18	May 14		
Week 14	May 19	Parallel and Distributed data systems	Chapter 20.4, 20.5, 21.1, 21.2, 22.1-22.7, 23.1-23.4	May 21		
Week 15	May 26	Extra		May 28		
	Jun 2			Jun 4		Lab 3: 11.59pm Jun 2

Lab 1 submission

1. Write each query in a separate .sql file, for example, 1.sql is the correct file name for the 1st query
2. Then, push these files to the main branch of your GitHub Classroom repository for the Lab 1 assignment
3. Check the feedback pull request in your repository for the autograder output in the case that your queries are not accepted

Lab 1 Grading Scheme

- The autograder run on the latest commit at the time of the deadline will be your final grade for the assignment

- Full points are awarded for a correct response in the queries:

Task	Number of points / 100
Points per query (x 10)	10
Total points for all queries	100

Questions

- Lab1 published – start from understanding the ER model
- Your frequent questions will be added to Ed
- Lab session on Wednesdays 17:15-19:00