***Note1:*** *All labs are mandatory, and you will be marked on 4, randomly chosen out of 11 labs. These will account for 20% of your overall mark (5% each).*

***Note2::*** *If a marker requests for a specific lab (e.g. Lab 3) and you do not have Lab 3 available, you can ask for another lab to be marked (e.g. Lab 2) but the total marks for Lab 2 will be reduced by 50%.*

***Note3:*** *All the labs should be created on one single document following the* [*provided template*](https://docs.google.com/document/d/1zqwnw-F49eG5GMX6uPiiZ62f3icV8jDdgUu4_XYISu8/edit?usp=sharing)*, saved in your drive (or computer) and accessible anytime. Labs are individual works, and they will be submitted into Moodle dropbox - see deadline.*

***Note4:*** *If a LAB is not finished in the scheduled session, you should complete it at your most convenient time by the beginning of the next session.*

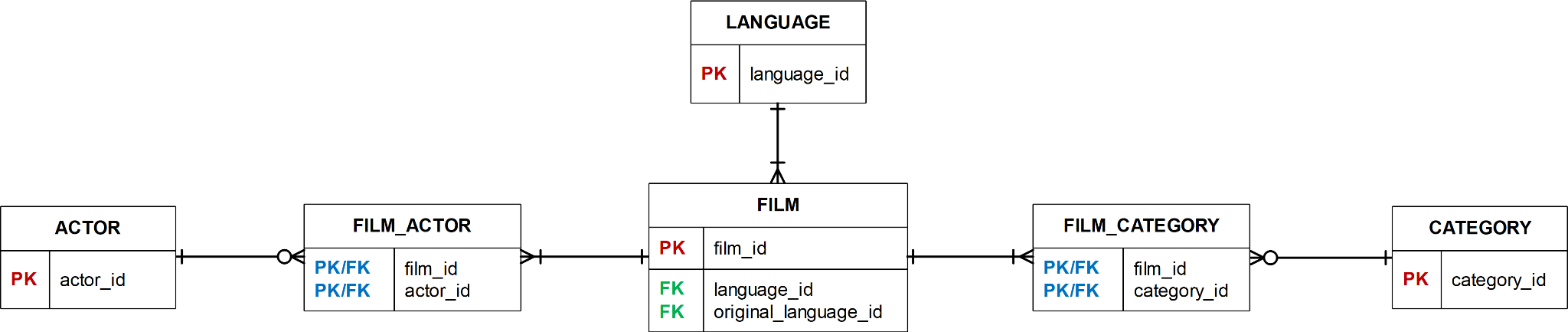
***Note5:*** *Use a code editor and do not type into VM directly*

*Online documentation:* [*https://devdocs.io/*](https://devdocs.io/)

LAB 3 - Transactions

1. You are given the following ERD without any additional attributes. Build up your database using your VM and **write 4 separate queries** that will contain a **SELECT**, **INSERT,** **UPDATE** and **DELETE** that will query the database. Create the ***Transaction Analysis Matrix*** for those transactions.

Do not spend time making the matrix “pretty” but accurate. You can use a spreadsheet or a table inserted in a document. Examples of the matrix are in the lecture presentation. Don't forget to add your work to the logbook.

(Query example - You should add some attributes to make the database functional (e.g. actor\_name; actor\_lname etc) 

***List the name of all movies where the actor's id (or surname) is X***.

| SELECT  actor\_name, film\_name  FROM  table\_name  JOIN table\_name ON table1.attr = table2.attr  JOIN table\_name ON table3.attr = table4.attr  WHERE  table.attr = 'X'; |
| --- |

Same for a new INSERT and for an UPDATE

1. Define in your own understanding (do not copy/paste definitions) of these terms: atomicity, consistency, isolation, durability, schedule, blind write, dirty read, unrepeatable read, serializable schedule, recoverable schedule, avoids-cascading-aborts schedule.

Note: *Some of the answers were in lecture, some in the book.*

1. Consider the situation below, in which a number of account records have the following values:

***A1 = £40; A2 = £50; A3 = £30***

To transfer £10 from A3 to A1 while concurrently calculating the total funds in the three accounts, the following sequence of events may occur. Show the value of each data item in the last column, and discuss the reason for an incorrect summary value.

Note: Look at the lecture example and/or chapter 22 in Connolly and Begg.

| **Time** | **Transaction1** | **Transaction2** | **SUM** |
| --- | --- | --- | --- |
| t1 | sum = 0 |  |  |
| t2 | read(**A1**) |  |  |
| t3 | sum = sum + A1 |  |  |
| t4 |  | read(**A3**) |  |
| t5 |  | A3 = A3 - 10 |  |
| t6 |  | write(A3) |  |
| t7 |  | read(A1) |  |
| t8 |  | A1 = A1 + 10 |  |
| t9 |  | write(A1) |  |
| t10 |  | commit; |  |
| t11 | read(**A3**) |  |  |
| t12 | sum = sum + A3 |  |  |
| t13 | read(**A2**) |  |  |
| t14 | sum = sum + A2 |  |  |

If you have finished and you are up for a challenge, see [SQL Challenge 1](https://docs.google.com/document/d/17LvFRF0u8VitrMggwJ6RWoYArghrtrLm0jN11cdqM90/edit). The Challenge itself will not be marked as part of the Labs but it might bring additional discretionary marks for attempting and solving. [You should submit](https://moodle.port.ac.uk/mod/assign/view.php?id=2497325) only 1 SQL file.

.