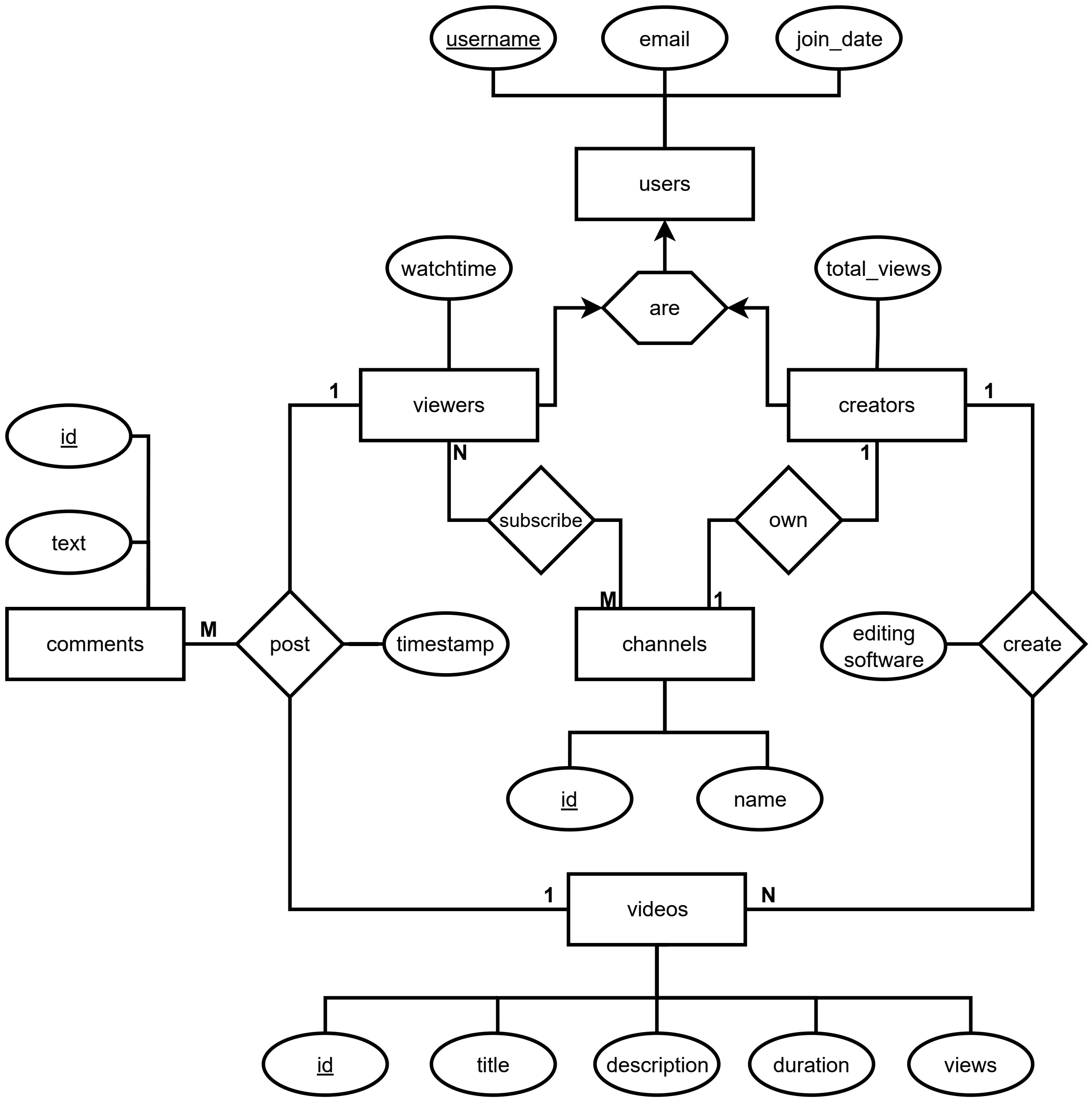
1 Translation to the Relational Model (7 points)

Translate the following entity-relationship model into the relational model, i.e. translate each entity type and relationship type into a relational schema. Simplify the relational schemas as much as possible and use meaningful data types. Use the notation from the lecture.



2 Translation into Relational Algebra (6 points)

Consider the following relational schemas.

$$
\begin{aligned}
\text{[persons]} &: \{[\underline{\text{pid: int}}, \text{ name: varchar, birth\_year: int}]\}\\
\text{[libraries]} &: \{[\underline{\text{lid: int}}, \text{ city: varchar}]\}\\
\text{[members]} &: \{[\underline{\text{mid: (persons)}}, \text{ favorite\_author: (authors), late\_fees: int}]\}\\
\text{[membership]} &: \{[\underline{\text{member: (members), library: (libraries)}}]\}\\
\text{[authors]} &: \{[\underline{\text{aid: (persons)}}, \text{ salary: int}]\}\\
\text{[books]} &: \{[\underline{\text{bid: int}}, \text{ author: (authors), title: varchar, year: int, genre: varchar}]\}\\
\text{[borrow]} &: \{[\underline{\text{member: (members), library: (libraries), book: (books)}}, \text{ borrow\_date: date,}\\
&\quad \text{due\_date: date}]\}\\
\text{[reserve]} &: \{[\underline{\text{member: (members), library: (libraries), book: (books), reservation\_date: date}}]\}
\end{aligned}
$$

In order to improve the readability of your expressions, you may name partial results. For example, if you name an expression as follows

$$R_3 := R_1 \bowtie_{\text{A=B}} R_2$$

R_3 can now be used in subsequent expressions. Also, you may use the following syntactic sugar to rename multiple attributes at once

$$\rho_{\text{A'}\leftarrow\text{A},\ldots,\text{Z'}\leftarrow\text{Z}}\ R \;=\; \rho_{\text{A'}\leftarrow\text{A}}\ (\ldots\ (\rho_{\text{Z'}\leftarrow\text{Z}}\ R))$$

(a) Translate the following colloquial queries into relational algebra expressions:

1. The genres of books that were written by an author that has a salary of more than 2000€.
2. The names of members that were born before 2000 and have not borrowed a book yet.
3. The cities of the libraries where a person has reserved a fantasy book and borrowed it on the same day.
4. For each city, the average late fees of people that are members at a library in that city and whose favorite author is Stephen King.

(b) Translate the following relational algebra queries into natural language:

1. $R := \text{borrow} \bowtie_{\text{library=lid}} (\sigma_{\text{city='Saarbrücken'}}\ \text{libraries})$
   $\pi_{\text{birth\_year}}(\sigma_{\text{count(*)}>5}(\gamma_{\text{pid,count(*)}}\ (R \bowtie_{\text{member=pid}}\ \text{persons})))$
2. $R_1 := (\text{reserve} \bowtie_{\text{member = member'}} (\rho_{\text{member'}\leftarrow\text{member},\text{library'}\leftarrow\text{library},\text{book'}\leftarrow\text{book}}\ \text{borrow}))$
   $R_2 := (R_1 \bowtie_{\text{book = bid}}\ \text{books}) \bowtie_{\text{author = aid}}\ (\sigma_{\text{salary}>2500}\ \text{authors})$
   $\gamma_{\text{min(birth\_year)}}(R_2 \bowtie_{\text{member = pid}}\ \text{persons})$

3 Relational Queries (4 points)

Consider the simplified IMDb schema from the lecture and the notebooks.

$$
\begin{aligned}
\text{[actors]} &: \{[\underline{\text{id: int}}, \text{first\_name: varchar}, \text{last\_name: varchar}, \text{gender: varchar}]\}\\
\text{[movies]} &: \{[\underline{\text{id: int}}, \text{name: varchar}, \text{year: int}, \text{rank: float}]\}\\
\text{[directors]} &: \{[\underline{\text{id: int}}, \text{first\_name: varchar}, \text{last\_name: varchar}]\}\\
\text{[movies\_directors]} &: \{[\underline{\text{director\_id: (directors), movie\_id: (movies)}}]\}\\
\text{[movies\_genres]} &: \{[\underline{\text{movie\_id: (movies)}}, \text{genre: varchar}]\}\\
\text{[roles]} &: \{[\underline{\text{actor\_id: (actors), movie\_id: (movies)}}, \text{role: varchar}]\}\\
\text{[directors\_genres]} &: \{[\underline{\text{director\_id: (directors)}}, \text{genre: varchar}, \text{prob: float}]\}
\end{aligned}
$$

To improve the readability of your expressions, you may again use the partial result notation and the syntactic sugar for renaming multiple attributes at once introduced in the previous exercises.

Translate the following colloquial queries into relational algebra expressions:

1. The names of the worst movies Denzel Washington has acted in since 2000 (inclusive).
2. The number of movies that did not have an actor that played the role of James Bond at least two times.

4 Implementation KeyRelation (3 points)

Implement the missing parts in the cell marked *Exercise* in the attached Jupyter Notebook assignment02.ipynb

The `KeyRelation` class should extend the `Relation` class in such a way that key attributes can be specified and taken into account when inserting tuples. To do this, use suitable data structures that enable you to check duplicates in *constant runtime* $\mathcal{O}(1)$.
For an easy start, we have already provided you with a unit test with several test cases in the notebook. Your final submission must calculate the correct result on all valid inputs. The quality of your implementation is taken into account when awarding points.

Submission

Solutions must be submitted in teams of 3 to 4 students by May, 9 2024, 10:00 a.m. via your personal status page in CMS using the Team Groupings functionality. Late submissions will not be graded!

Please note that there are two submissions, under **Theoratical** you submit your solutions to exercises 1, 2 and 3 as a PDF file.

Your solution to exercise 4 must be handed in under **Practical** as txt file.

Make sure that you only copy the complete Jupyter cells you want to add and that indentation and formatting are correct.