Leo Forster
Maximilian Prinz
Bastian Simon
Marlon Paci

Big Data Engineering
Assignment 2

1. Translation to the Relational Model

[users] : {[username: varchar, email: varchar, join_date: date]}

[viewers] : {[vid: (users), watchtime: time]}

[creators] : {[cid: (users), total_views: int]}

[channels] : {[id: int, name: varchar, owner: (creators)]}

[comments] : {[id: int, text: varchar, poster: (viewers), video: (videos), timestamp: datetime]}

[videos] : {[id: int, title: varchar, description: varchar, duration: time, views: int, creator: (creators), editing_software: varchar]}

[subscribe] : {[viewer: (viewers), channel: (channels)]}

Big Data Engineering: Assignment 2

Leo Forster, Maximilian Prinz, Bastian Simon, Marlon Paci

1

Translation into Relational Algebra

Subproblem 1

Translate the following colloquial queries into relational algebra expressions

1.
$\pi_{name,genre}((books) \bowtie_{author=aid} ((persons) \bowtie_{pid=aid} (\sigma_{salary>2000}(authors))))$

2.
$\pi_{name}(\sigma_{birth_year<2000}((persons) \bowtie_{pid=mid} (\pi_{mid}(members) - \pi_{member}(borrow))))$

3.
$R_1 := \pi_{rb,rl}((borrow) \bowtie_{member=rm,library=rl,book=rb} (\rho_{rm \leftarrow member,rl \leftarrow library,rb \leftarrow book}(reserve)))$

$R_2 := \sigma_{genre='fantasy'}((books) \bowtie_{bid=rb} (\sigma_{borrow\_date=reservation\_date}(R)))$

$\pi_{city}((libraries) \bowtie_{lid=rl} (R_2))$

4.
$M := ((member) \bowtie_{favorite\_author=name} (\sigma_{name='StephenKing'}(persons) \bowtie_{pid=aid} (authors)))$

$L := \sigma_{city,pid}((libraries) \bowtie_{lid=library} ((membership) \bowtie_{member=mid} (M)))$

$\gamma_{city,avg(*)}(L)$

Subproblem 2

Translate the following relational algebra queries into natural language

1.
This query gives back all birth years where someone has have/has borrowed at least 6 books in a library in Saarbrücken.

2.
It gives back the birth year of the youngest person that have/has borrowed or reserved a book by an author whose salary is above 2500.

Aufgabe 3:

2. $R_1 := \sigma_{role = \text{'James Bond'}}(roles)$

$\gamma_{count(*)}(movies) - \gamma_{count(*)}(\sigma_{count(*)>1}(\gamma_{actor\_id, count(*)}(R_1)))$

1. $R_2 :=$ Filme mit Denzel Washington die nach 1999 rausgekommen sind.

$\Pi_{name}(\sigma_{rank = min(rank)}(R_2))$