Homework #2
CS486/586
Prof. Charles Winstead
CRN: 40876

Jiacheng Zhao
Jordan Witte
Resheet Schultz
Due: 01-25-2019

Part One — Relational Algebra (40 pts ; 5 pts each)

Exercise 1

Write the following queries in Relational Algebra, using ONLY the select, project and cross product operators:

a) *Find the unique agent names and cities who have Magellon clearance for whom we don't have a known address. (Using one Selection Operator)*

$$\pi_{\text{first, last, city}}\left(\sigma_{\text{clearance\_id = sc\_id AND sc\_level = 'Magellon' AND address = NULL}}\left(\text{agent} \times \text{securityclearance}\right)\right)$$

b) *Same question as (a) but using three Selection Operators*

$$\pi_{\text{first, last, city}}\left(\sigma_{\text{clearance\_id = sc\_id}}\left(\sigma_{\text{address = NULL}}(\text{agent}) \times \sigma_{\text{sc\_level = 'Magellon'}}(\text{securityclearance})\right)\right)$$

c) *Find the unique agent names and country of all agents who are Snipers*

$$\pi_{\text{first, last, country}}\left(\sigma_{\text{agent.agent\_id=skillrel.agent\_id}}\left(\text{agent} \times \sigma_{\text{skillrel.skill\_id = skill.skill\_id}}\left(\text{skillrel} \times \sigma_{\text{skill='Sniper'}}(\text{skill})\right)\right)\right)$$

Exercise 2

Write the following queries in Relation Algebra, using the join operator:

a) *Find the unique agent names and country of all agents who are Snipers (same as 1c)*

$$\pi_{\text{first, last, country}}\left(\left(\text{agent} \bowtie \left(\text{skillrel} \bowtie \sigma_{\text{skill='Sniper'}}(\text{skill})\right)\right)\right)$$

b) *What are the (unique) agent names who are on ongoing, Top Secret missions (and their team and mission names)? (Problem 7 from HW1)*

$$\begin{aligned}
&\pi_{\text{first, last, team.name, mission.name}}\Big(\\
&\quad \big(\sigma_{\text{mission_status = 'ongoing'}}(\text{mission}) \\
&\qquad \bowtie_{\text{access_id = sc_id}} \\
&\quad \sigma_{\text{sc_level = 'Top Secret'}}(\text{securityclearance}) \big) \\
&\quad \bowtie_{\text{mission.team_id = team.team_id}} \\
&\quad \left(\text{team} \bowtie \text{teamrel} \bowtie \text{agent}\right) \Big)
\end{aligned}$$

Exercise 3

State whether these expressions are equivalent and a brief explanation why. (Equivalent means true under all potential data entries that meet the schema criteria, not just the current instance of data in the db, but assume that foreign keys cannot be NULL)

a) $\pi_{mission.name}(mission \bowtie_{mission.team\_id=team.team\_id} team) \quad vs \quad \pi_{name}(mission)$

These expressions are equivalent. Proof by argument that each expression's returned set contains the other.

If mission name n is returned in $\pi_{\text{mission.name}}(\text{mission} \bowtie_{\text{mission.team\_id=team.team\_id}} \text{team})$, then n must appear as a value in the mission portion of the join expression, so n trivially appears in $\pi_{\text{name}}(\text{mission})$.

If mission name n is returned in $\pi_{\text{name}}(\text{mission})$, then the containing mission tuple has a non-null team_id which is a foreign key to team.team_id . Thus the join expression will match the mission tuple to a team tuple with equal team_id, so the tuple with name n also appears in the join expression.

b) $\pi_{team.name}(mission \bowtie_{mission.team\_id=team.team\_id} team) \quad vs \quad \pi_{name}(team)$

These expressions are not equivalent. If the mission relation is empty, then the join expression is also empty, even if the team relation is non-empty. In this case, the two expressions would return different sets of values.

c) $\pi_{team\_id}(\sigma_{meeting\_frequency='weekly'}(mission \bowtie_{mission.team\_id=team.team\_id} team)) \quad vs$
$\pi_{team\_id}(\sigma_{meeting\_frequency='weekly'}(team)) \cap \pi_{team\_id}(missions)$

These expressions are equivalent. In the first expression, the σ operation ensures that only teams that meet weekly are returned; and the join operation ensures that only teams assigned to missions are returned. These conditions are expressed equivalently by the two clauses of the second expression, and the intersect operation applies both conditions to the final returned relation.

Part Two - More SQL (60 pts)

Write a single SQL statement for each of the following queries. Show at least the first five rows of the result for each query (or fewer, if the result is smaller) and the number of rows returned.

Note: If you are asked to write a query two ways, you only need to include the results and row count once.

Exercise 4

Different types of JOINs and SET operators (10 pts each)

a) *List agents first and last names who are not affiliated with any known organizations and who live in Seattle.*

```
SELECT DISTINCT FIRST, LAST FROM AGENT WHERE CITY = 'Seattle' EXCEPT
SELECT DISTINCT FIRST, LAST from AGENT JOIN AFFILIATIONREL ON AFFILIATIONREL.AGENT_ID
    =AGENT.AGENT_ID;
```

Result: 16 rows

first_name	last_name
Bill	Herrera
Lois	Oviat
Chris	McGee
Tim	Galantine
Dan	Virata

b) *Find the team name for all teams with at least one agent who can speak Pashtu. Do this query two ways: Once using NATURAL JOIN and once without any JOIN operator in the FROM clause.*

Method 1:

```
SELECT DISTINCT TEAM.NAME FROM TEAM
NATURAL JOIN TEAMREL
NATURAL JOIN AGENT
NATURAL JOIN LANGUAGE
NATURAL JOIN LANGUAGEREL
where LANGUAGE.LANGUAGE = 'Pashtu';
```

Method 2:

```
SELECT DISTINCT T.NAME FROM TEAM T, TEAMREL TR, AGENT A, LANGUAGE L, LANGUAGEREL LR
WHERE TR.TEAM_ID = T.TEAM_ID
AND TR.AGENT_ID = A.AGENT_ID
AND L.LANG_ID = LR.LANG_ID
AND LR.AGENT_ID = A.AGENT_ID
AND L.LANGUAGE = 'Pashtu';
```

Result: (29 rows)

name
F Sharp
Camaro
Ghost Hunters
Vikings
ShowBiz

c) *List the team name for each team that has at least one agent with the 'Nuclear Intelligence' skill, at least one agent who is a 'Computer Hacker' and no agents with only 'Classified' or 'Unclassified' security clearance.*

```
(
SELECT DISTINCT T.NAME FROM TEAM T
NATURAL JOIN TEAMREL TR
NATURAL JOIN AGENT A
NATURAL JOIN SKILL S
NATURAL JOIN SKILLREL SR
where S.SKILL = 'Nuclear Intelligence'

INTERSECT

SELECT DISTINCT T.NAME
FROM TEAM T
NATURAL JOIN TEAMREL TR
NATURAL JOIN AGENT A
NATURAL JOIN SKILL S
NATURAL JOIN SKILLREL SR
where S.SKILL = 'Computer Hacker'
)
```

```
EXCEPT

SELECT DISTINCT T.NAME
FROM TEAM T
NATURAL JOIN TEAMREL TR
NATURAL JOIN AGENT A
JOIN SECURITYCLEARANCE C ON C.SC_ID = A.CLEARANCE_ID

WHERE (C.SC_LEVEL = 'Classified' OR
      C.SC_LEVEL = 'Unclassified')
;
```

Result: (2 rows)

name
Boat Team 6
Giraffe

Exercise 5

Aggregation, Group by, Having (10 pts each)

a) *Find the total number of agents that are affiliated to terrorist organizations.*

```
SELECT COUNT( DISTINCT(A.AGENT_ID)) AGENT_COUNT
FROM AGENT A
NATURAL JOIN AFFILIATIONREL AFR
NATURAL JOIN AFFILIATION AF
where (AF.description like '\%terrorist\%' ) OR (AF.description like '\%Terrorist\%');
```

Result: (1 row)

agent_count
158

b) *Produce a table of the number of agents that are affiliated with terrorist organizations, broken down by the strength of their affiliation*

```
SELECT AFR.AFFILIATION_STRENGTH, COUNT( DISTINCT(A.AGENT_ID)) AGENT_COUNT
FROM AGENT A
NATURAL JOIN AFFILIATIONREL AFR
NATURAL JOIN AFFILIATION AF
WHERE (AF.description like '\%terrorist\%' ) OR (AF.description like '\%Terrorist\%')
GROUP BY afr.affiliation_strength;
```

Result: (3 rows)

affiliation_strength	agent_count
medium	54
strong	35
weak	82

c) *List the Locations (Address and City) and the number of agents who live there for locations where more than one agent lives. (Filtering out agents with missing locations).*

```
SELECT A.ADDRESS, A.CITY, COUNT(*) AS AGENT_COUNT
FROM AGENT A
WHERE ADDRESS IS NOT NULL
AND CITY IS NOT NULL

GROUP BY (A.ADDRESS, A.CITY)
HAVING COUNT(*) > 1
;
```

Result: (10 rows)

address	city	agent_count
56 98th Avenue	San Francisco	2
34 7th Avenue	Cairo	2
29 20th Avenue	Warsaw	2
17 65th Avenue	Seattle	2
17 56th Avenue	San Francisco	2