COMS 4111 Intro to DB HW2

Zachary Huang (zh2408)

March 5, 2022

1 Relational Algebra

1.1 Q1.1

$\pi_{Person.ssn}((\sigma_{companyid=601}(Person)) \bowtie_{Person.ssn=Holdings.ssn} (\sigma_{companyid=700 \wedge sharenum>500}(Holding)))$
Note: Natural join between Person and Holdings is wrong, as you are joining on both ssn and companyid.
Rubric:
+0.5 Select google employees
+0.25 Select facebook shareholders
+0.25 Select the shareholds that has sharenum > 500
+0.5 Correctly joined both relations
+0.5 Correctly project the ssn

1.2 Q1.2

// PersonStock are the stocks each person owns
$\rho(PersonStock, \pi_{Person.ssn, Holding.companyid}(Person \bowtie_{Person.ssn=Holdings.ssn} Holding))$
// ManagerStock are the stocks each person's manager owns
$\rho(ManagerStock, \pi_{Person.ssn, Holding.companyid}(Person \bowtie_{Person.managerid=Holdings.ssn} Holding))$
// PersonIntersection are persons whose stocks intersect with manager's stocks
$\rho(PersonIntersection, \pi_{ssn}(PersonStock \cap ManagerStock))$
$\pi_{ssn}(Person) - PersonIntersection$
Note: The problem doesn't specify corner cases when a person or manager doesn't own any stock. For those without stocks, both excluding and including them are fine.
Rubric:
+0.25 compute person stocks
+0.5 compute manager stocks
+0.25 attribute references and projections are unambiguous
+0.75 compute persons that have no stocks in common with their manager (may be solved in multiple ways)
+0.25 correct final projection

1.3 Q1.3

$\rho(h1, Holding))$
$\rho(h2, Holding))$
$\pi_{ssn}(\sigma_{Holding.companyid \neq h1.companyid \wedge Holding.companyid \neq h2.companyid \wedge h1.companyid \neq h2.companyid}(Holding \bowtie_{ssn} h1 \bowtie_{ssn} h2))$
Note: We do not have aggregation in relational algebra.
Rubric:
+1.0 for using 3 self joins
+0.5 for using the right selection
+0.5 For correctly projecting the final result

2 More Relational Algebra

2.1 Q2.1

A	B
1	x
2	y
2	z

Rubric:
+1.0 Correct Column Names
+1.0 Correct Rows (Duplicates removed)

2.2 Q2.2

D
c
a

Rubric:
+1.0 Correct Column Names
+1.0 Correct Rows (Duplicates removed)

2.3 Q2.3

(A)	B	C	(A)
1	x	a	1
2	y	c	1
2	y	b	1
2	z	c	1
1	x	a	2
2	y	c	2
2	y	b	2
2	z	c	2

Rubric:
+0.25 First Column A represented correctly parenthesis or TableName.ColumnName
+0.25 Last Column A represented correctly parenthesis or TableName.ColumnName
+0.5 column order is correct
+1.0 all rows are correct (as per the column order)

2.4 Q2.4

A	T1.B	T1.C	T2.B	T2.C	D
1	x	a	1	x	c
1	x	a	3	x	a
2	y	c	2	y	c
2	y	b	2	y	c

Note: The schema keeps all 6 attributes because it's a theta join.
Rubric:
+0.75 if all column names are correct and has proper order of all attributes of first relation with all attributes of second relation
+1.0 all rows are correct according to the column order
+0.25 parenthesis or TableName.ColumnName mentioned correctly

2.5 Q2.5

A	B	C
1	x	a
2	y	b
2	z	c

Note: For set operation, we use the left argument's schema. Schema BCD is wrong.
Rubric:
+1.0 schema is correct
+0.25 no row contains intersection (2,y,c)
+0.75 rows are all correct

2.6 Q2.6

A	T1.B	T1.C	T2.B	T2.C	T2.D
2	y	c	1	x	c
2	y	b	1	x	c
2	z	c	1	x	c

Rubric:
+0.75 correct Schema Order
+0.25 appropriate notation for same column names
+0.25 no row contains T2.B = 2 (so selection is right)
+0.25 no row contains T1.A>T2.B (so join condition is satisfied)
+0.5 rows are all correct

3 SQL

3.1 Q3.1

Find the ids and names of stores which is in NY or has at most 100 employees.

```
SELECT storeid, s_name
FROM Store
WHERE employee_number <= 100 OR city = "New York"
```

Note: Distinct is not necessary because $(storeid, g\_id)$ is primary key.
Rubric:
+0.25 description is correct
+0.5 employee_number <= 100 is correct
+0.5 city = "New York" is correct
+0.5 "OR" is correct
+0.25 "storeid, s_name" is correctly selected

3.2 Q3.2

Find the names of stores which are supplied with pencils.

```
SELECT DISTINCT s_name
FROM Store, Goods, Supply
WHERE Store.storeid = Supply.storeid AND Supply.g_id = Goods.g_id AND g_name = "pencil"
```

Note: Distinct is necessary here.
Rubric:
+0.25 description is correct
+0.25 all three relations "Store, Goods, Supply" are in "From" clause
+0.5 "DISTINCT" is correct
+0.5 join conditions are correct
+0.5 "g_name = "pencil"" is correct

3.3 Q3.3

Find the goods each store doesn't have.

```
(SELECT storeid, g_id FROM Store, Goods)
EXCEPT
(SELECT * FROM Supply );
```

Rubric:
+0.25 description is correct
+0.5 cross product is computed
+0.5 the schema of the cross product "storeid, g_id" is correct
+0.25 supply relation is correctly selected
+0.5 "EXCEPT" is correct (can also use nested query)