Ben Juarez

PS1

1) ⓐ $\Pi_{person\_name}(\sigma_{company\_name = \text{"First Bank Corporation"}}(works))$

ⓑ $\Pi_{person\_name,\, city}(\sigma_{company\_name = \text{"First Bank Corporation"}}(employee \bowtie works))$

ⓒ $\Pi_{person\_name,\, street,\, city}(\sigma_{salary > 10{,}000}(\sigma_{company\_name = \text{"First Bank Corporation"}}(employee \bowtie works)))$

ⓓ $e\_x\_w \leftarrow employee \bowtie works$

$\Pi_{person\_name}(\sigma_{e\_x\_w.city = company.city}(\sigma_{e\_x\_w.company\_name = company.company\_name}(e\_x\_w \times company)))$

ⓔ $\Pi_{test.city}(\sigma_{company.city = test.city}(\sigma_{company.company\_name = \text{"Small Bank Corporation"}}(company \times \rho_{test}(company))))$

2) $\Pi_{customer\_name,\, customer\_city,\, loan\_number,\, amount}(borrower \bowtie loan \bowtie customer)$

ⓐ Jackson is not included in this result b/c the natural join w/ customer will eliminate the tuple for Jackson (since Jackson is not in customer relation)

ⓑ The database could be modified such that a new tuple is added for Jackson in customer relation:

customer_name	customer_street	customer_city
Jackson	California	Pasadena

ⓒ $\Pi_{customer\_name,\, customer\_city,\, loan\_number,\, amount}((borrower \bowtie loan) ⟕ customer)$

3) ⓐ $works \leftarrow \Pi_{person\_name,\, company\_name,\, (salary * 1.1)}(\sigma_{company\_name = \text{"First Bank Corporation"}}(works))$

ⓑ $managers \leftarrow \Pi_{manager\_name \text{ as } person\_name}(works)$

$works \leftarrow \Pi_{person\_name,\, company\_name,\, (salary * 1.1)}(\sigma_{salary * 1.1\ 100{,}000 \vee salary * 1.1 = 100{,}000}(works \bowtie managers)) \cup \Pi_{person\_name,\, company\_name,\, (salary * 1.03)}(\sigma_{salary * 1.1 > 100{,}000}(works \bowtie managers)) \cup$
$(works - (works \bowtie managers))$

ⓒ $works \leftarrow works - \sigma_{company\_name = \text{"Small Bank Corporation"}}(works)$

4) ⓐ $\Pi_{account\_number}(\sigma_{n > 2}({}_{account\_number}\mathcal{G}_{count(customer\_name) \text{ as } n}(depositor)))$

ⓑ $d \leftarrow depositor$

$\Pi_{d.account\_number}(\sigma_{d.customer\_name \neq t1.customer\_name \vee d.customer\_name \neq t2.customer\_name \vee t1.customer\_name \neq t2.customer\_name}(\sigma_{d.account\_number = t1.account\_number \wedge d.account\_number = t2.account\_number}$
$(d \times \rho_{t1}(d) \times \rho_{t2}(d))))$

5) ⓐ $company\_info \leftarrow {}_{company\_name}\mathcal{G}_{count(person\_name) \text{ as } employees}(works)$

$\Pi_{company\_name}(company\_info) - \Pi_{company\_info.company\_name}(\sigma_{company\_info.employees < test.employees}(company\_info \times \rho_{test}(company\_info)))$

ⓑ $payroll\_info \leftarrow {}_{company\_name}\mathcal{G}_{sum(salary) \text{ as } payroll}(works)$

$\Pi_{company\_name}(payroll\_info) - \Pi_{payroll\_info.company\_name}(\sigma_{payroll\_info.payroll > test.payroll}(payroll\_info \times \rho_{test}(payroll\_info)))$

(c) $company\_salaries \leftarrow {}_{company\_name}G_{average(salary)\ as\ avg\_salary}(works)$

$FBC \leftarrow \sigma_{company\_name = \text{"First Bank Corporation"}}(company\_salaries)$

$other \leftarrow company\_salaries - FBC$

$\Pi_{other.company\_name}\left(\sigma_{FBC.avg\_salary < other.avg\_salary}\left(FBC \times other\right)\right)$

monkey_likes.name	monkey_food.food
Bobo	apples
Bobo	bananas
⋮	⋮

(6) $r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r)) \Rightarrow$ $monkey\_likes \div monkey\_foods = \Pi_{name}(monkey\_likes) - \Pi_{name}((\Pi_{name}(monkey.likes) \times monkey\_foods) - \Pi_{name,food}(monkey\_likes))$

name
Bobo
Jojo ...

name	food
Bobo	apples

monkey_likes

Guenter would appear in monkey_likes ÷ monkey_foods b/c Guenter satisfies the 2 conditions given by the definition of relational division:

① t is in $\Pi_{R-S}(r) \Rightarrow$ Guenter is in $\Pi_{name}(monkey\_likes)$ ✓

② For every food item in monkey_foods, there is is a tuple t_{monkey_likes} in monkey_likes satisfying:

ⓐ $t_{monkey\_likes}[food] = t_{monkey\_foods}[food] \Rightarrow$ each food item in monkey_foods appears in a tuple w/ Guenter in monkey_likes ✓

ⓑ $t_{monkey\_likes}[name] = t \Rightarrow$ Guenter = Guenter ✓

(7) $r \div_E s = (r \div s) - \Pi_{R-S}(\Pi_{R-S,S}(r) - \Pi_{R-S}(r) \times s)$

- This definition would work in this case (and all other cases) because we know that $r \div s$ gives us $\{(Jojo), (Guenter)\}$, but we want to get rid of Guenter since there is a tuple with Guenter in monkey_likes that does not match the contents of monkey_foods (tofu not in monkey_foods). So, the right side of this definition ($\Pi_{R-S}(\Pi_{R-S,S}(r) - \Pi_{R-S}(r) \times s)$) will end up projecting the name Guenter because it basically gives us the tuples in monkey_likes that are leftover after taking the set difference with all of the possibilities of name, food combinations from monkey_foods. Specifically, $\Pi_{R-S,S}(r) - \Pi_{R-S}(r) \times s$, in regards to Guenter, can be imagined as so:

Guenter	apples
Guenter	oranges
Guenter	bananas
Guenter	tofu

—

Guenter	apples
Guenter	oranges
Guenter	bananas

which will include Guenter in the result since (Guenter, tofu) will not be eliminated so $\Pi_{R-S}()$ will project, in this case, names. Therefore, Guenter will be included on the right side of the set difference in the definition of $r \div_E s$, meaning that the final, accurate result of monkey_likes $\div_E$ monkey_foods will be $\{(Jojo)\}$, as desired. Note that this logic applies in all cases.

(8) $\pi_{R-S}(\sigma_{counts=c}({}_{R-S}G_{count(S)\text{ as }counts}(r \bowtie s)) \times G_{count(S)\text{ as }c}(s)))$

(9) $\sigma_{\theta}({}_{A}G_{F}(r))$ vs. ${}_{A}G_{F}(\sigma_{\theta}(r))$

These expressions would be equivalent since $\sigma_\theta()$ only uses predicates from A, so it will always select the same tuples whether or not ${}_{A}G_{F}()$ is applied before or after $\sigma_\theta()$ (because it has to be grouped by A). If θ could use predicates not from A, then these expressions would not be equivalent. Again, the same tuples will always be selected in this case due to given definitions in this case. This goes both ways such that the aggregation is not affected by the selection.

$\Longrightarrow$ $\boxed{\text{equivalent}}$

(10) $\Pi_A(r-s)$ vs. $\Pi_A(r) - \Pi_A(s)$ *let Π_A be Π_a

r

a	b
dog	red
dog	white
fish	blue

s

a	b
dog	red
cat	white
turtle	blue

$\Longrightarrow$

$r-s \Rightarrow$

a	b
dog	white
fish	blue

$\Rightarrow \Pi_a\left(\right.$

a	b
dog	white
fish	blue

$\left.\right) \Rightarrow$

a
dog
fish

$\Pi_a(r) - \Pi_a(s) \Rightarrow$

a
dog
fish

$-$

b
dog
cat
turtle

$\Rightarrow$

a
fish

We have clearly provided a counterexample where $\Pi_A(r-s) \neq \Pi_A(r) - \Pi_A(s)$ $\boxed{\Longrightarrow \text{NOT equivalent}}$

(11) $(r ⟕ s) ⟕ t$ $\quad$ $r ⟕ (s ⟕ t)$ $\quad$ $r(a, b1)$ $\quad$ $s(a, b2)$ $\quad$ $t(a, b3)$

r

a	b1
0	v
1	w

s

a	b2
2	x
3	y

t

a	b3
0	v
1	w

$r ⟕ s$:

a	b1	b2
0	v	null
1	w	null

$\Downarrow$

$(r ⟕ s) ⟕ t$:

a	b1	b2	b3
0	v	null	v
1	w	null	w

$s ⟕ t$:

a	b2	b3
2	x	null
3	y	null

$\Downarrow$

$r ⟕ (s ⟕ t)$:

a	b1	b2	b3
0	v	null	null
1	w	null	null

$\Rightarrow$ Thus, we have shown that $(r ⟕ s) ⟕ t \neq r ⟕ (s ⟕ t)$ when r and t have at least 1 equivalent value under attribute a (while s has zero common values in attribute a) $\Rightarrow$ $\boxed{\text{NOT equivalent}}$

(12) $\sigma_\theta (r ⟕ s)$ vs. $\sigma_\theta(r) ⟕ s$

We can determine that these expressions are equivalent since we are only able to select using predicates using attributes from r. With this, the same tuples will be selected appropriately whether or not the left outer join was applied first. If θ could be a predicate using attributes from s, then these expressions would not be equivalent. Again, since it is a left outer join and we can only select from attributes from r (the "left" side), we can see how this holds true.

$\Longrightarrow$ $\boxed{\text{equivalent}}$

(13) $\sigma_\theta(r ⟗ s)$ vs. $r ⟗ \sigma_\theta(s)$ $\theta \to b2 = x$

r:

a	b1
0	w
1	x

s:

a	b2
1	x
2	y

$r ⟗ s$:

a	b1	b2
0	w	null
1	x	x

$\sigma_{b2=x}(r ⟗ s)$:

a	b1	b2
1	x	x

$\sigma_{b2=x}(s)$:

a	b2
1	x

$r ⟗ \sigma_{b2=x}(s)$

a	b1	b2
0	w	null
1	x	x

By counter-example, we can clearly see that $\sigma_\theta(r ⟗ s) \neq r ⟗ \sigma_\theta(s)$ $\Rightarrow$ $\boxed{\text{NOT equivalence}}$