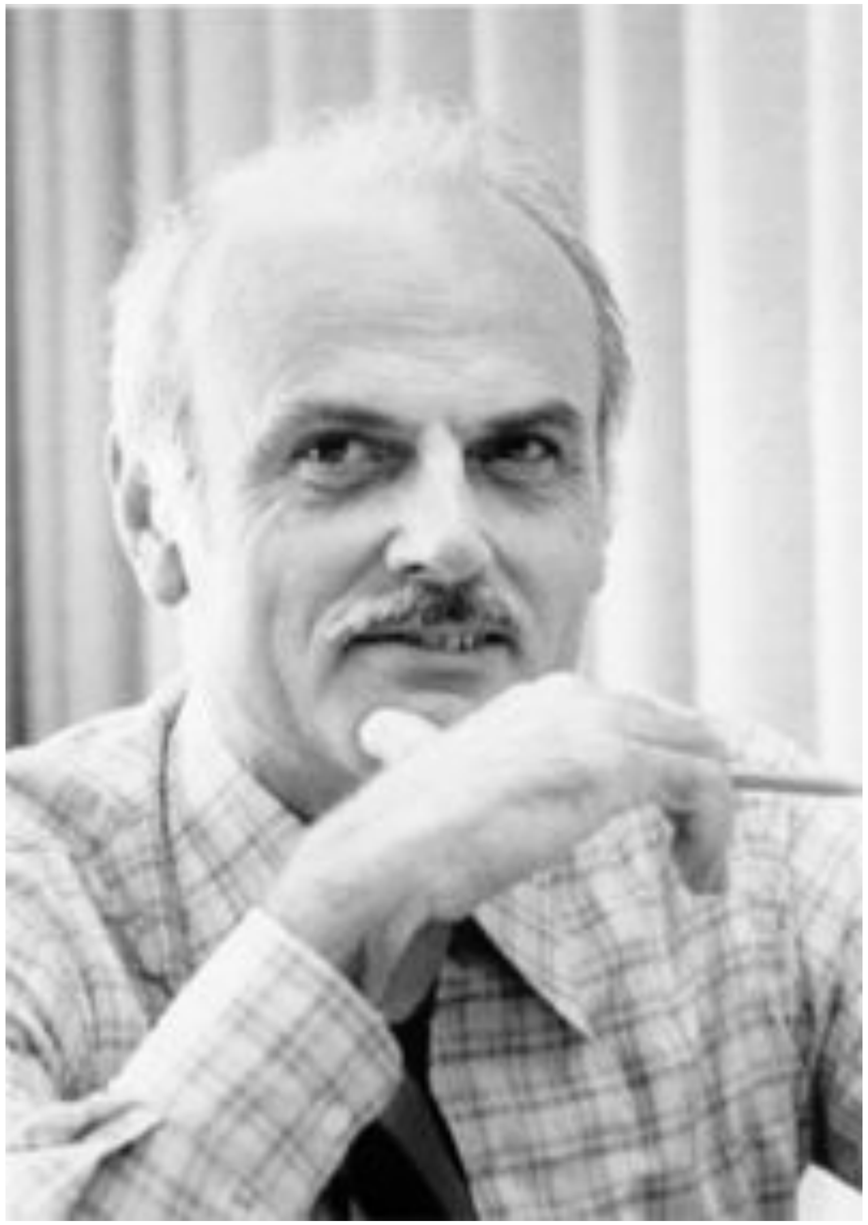
Relational Databases

Lecturer: Азат Якупов (Azat Yakupov)
https://datalaboratory.one



The main application of **relational algebra** is to provide a theoretical foundation for RDBMS, particularly query languages for such databases, chief among which is SQL.

Edgar Frank Codd proposed such an algebra as a basis for database query languages at 1970.

1st Query Language was ***ALPHA***
(developed by Dr. Codd, but never implemented)

Then ***QUEL*** for Ingres then ***POSTQUEL*** and finally ***SQL***

```
retrieve (s.all)
where s.state = "FL"
```

QUEL

~

SQL

```
SELECT *
FROM student
WHERE state = 'FL'
```

```
range of s is student
append to s
(name = "Ivan", age = 17,
 sex = "m", state = "FL")
```

QUEL

~

SQL

```
INSERT INTO student
(name, age, sex, state)
VALUES ('Ivan', 17, 'm', 'FL')
```

```
delete s where
s.name="Ivan"
```

QUEL

~

SQL

```
DELETE FROM student
WHERE name='Ivan'
```

replace s
(age=s.age+1)

QUEL

~

SQL

UPDATE student
SET age=age+1

QUEL

```
range of e is EMPLOYEE retrieve into W
 (COMP = e.Salary / (e.Age - 18))
 where e.Name = "Jones"
```

~

SQL

```
CREATE TABLE W AS
SELECT (e.salary / (e.age - 18)) AS comp
   FROM employee AS e
WHERE e.name = 'Jones'
```

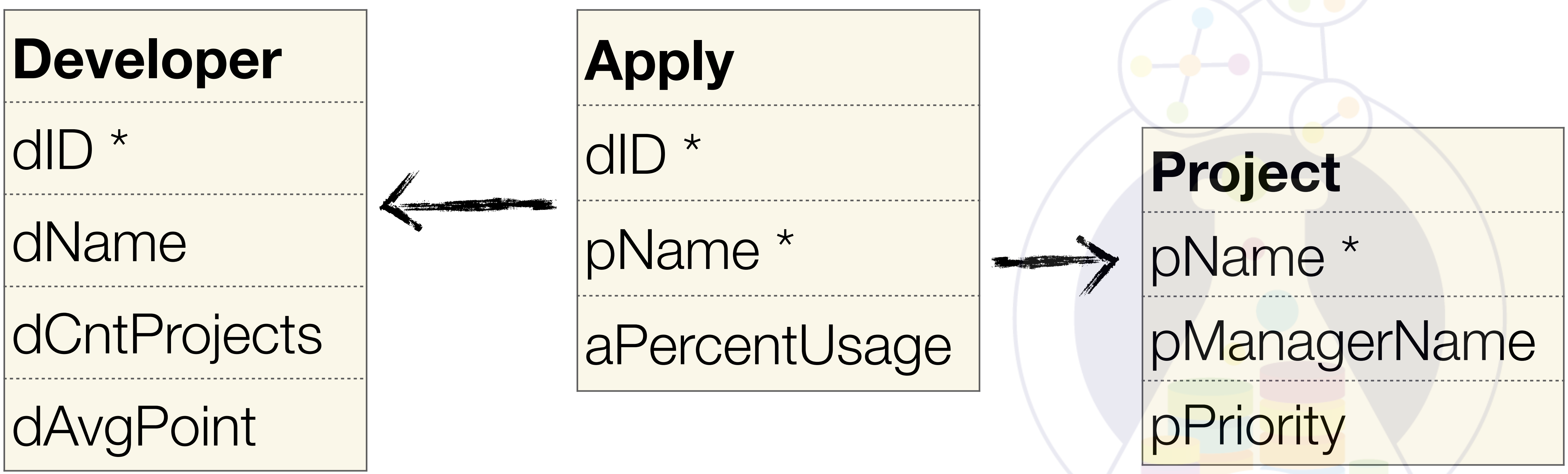
Primitives Operators

- Selection (Restriction)
- Projection
- Cartesian Product ~ cross join
- Union
- Difference
- Rename

Developer (**dID**, dName, dCntProjects, dAvgPoint)

Project (**pName**, pManagerName, pPriority)

Apply (**dID**, **pName**, pPercentUsage)



Selection (restriction)

$$\sigma_{A\theta B}(R)$$

A, B are attribute names

θ is a binary operation in the set

$<, \leq, =, \neq, >, \geqslant$

R is a relation

$$\sigma_{dAvgPoint \geqslant 4.0}(Developer)$$

attributes

binary operation

relations

$$\sigma_{pManagerName = Ivanov}(Project)$$

Generalized Selection

$$\sigma_{\varphi}(R)$$

AND $$\sigma_{\varphi \wedge \psi}\left(R\right) = \sigma_{\varphi}\left(R\right) \cap \sigma_{\psi}\left(R\right)$$

OR $$\sigma_{\varphi \vee \psi}\left(R\right) = \sigma_{\varphi}\left(R\right) \cup \sigma_{\psi}\left(R\right)$$

NOT $$\sigma_{\neg \varphi}\left(R\right) = R - \sigma_{\varphi}\left(R\right)$$

$$\sigma_{dAvgPoint \geqslant 4.0 \wedge dCntProjects=3}\ (Developer)$$

$$\sim$$

$$\sigma_{dAvgPoint \geqslant 4.0\ AND\ dCntProjects=3}\ (Developer)$$

Developer

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5
3	Ivan	5	5

$$\sigma_{dAvgPoint \geqslant 4.0\ AND\ dCntProjects=3}(Developer)$$

dID	dName	dCntProjec	dAvgPoint
1	Ivan	3	5

Project

pName	pManagerNam	pPriority
Project #1	Ivan Ivanov	high
Project #2	Petr Petrov	middle
Project #3	Sergey Ivanov	low

$$\sigma_{pManagerName="Ivan\ Ivanov"\ OR\ pManagerName="Petr\ Petrov"}(Project)$$

pName	pManagerNam	pPriority
Project #1	Ivan Ivanov	high
Project #2	Petr Petrov	middle

$$\sigma_A(R) = \sigma_A \sigma_A(R)$$

```
SELECT *
  FROM R
WHERE A = 1
```

```
SELECT *
  FROM R
WHERE A = 1
      AND
      A = 1
```

$$\sigma_A \sigma_B(R) = \sigma_B \sigma_A(R)$$

```
SELECT *
  FROM R
WHERE A = 1
        AND
        B = 1
```

```
SELECT *
  FROM R
WHERE B = 1
        AND
        A = 1
```

$$\sigma_{A \wedge B}(R) = \sigma_A(\sigma_B(R))$$

```
SELECT *
  FROM R
WHERE A = 1
         AND
         B = 1
```

```
SELECT *
FROM (
   SELECT *
     FROM R
   WHERE A = 1
)
WHERE B = 1
```

$$\sigma_{A \wedge B}(R) = \sigma_B(\sigma_A(R))$$

```sql
SELECT *
  FROM R
WHERE A = 1
          AND
          B = 1
```

```sql
SELECT *
FROM (
    SELECT *
      FROM R
    WHERE B = 1
)
WHERE A = 1
```

Azat Yakupov

Projection

$$\pi_{A_1,\ldots,A_n}(R)$$

$A_1, \ldots, A_n$ are attribute names

R is a relation

$$\pi_{dName}\,(Developer)$$

attributes

relations

$$\pi_{dName,\,2*dCntProjects}\,(Developer)$$

Developer

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5
3	Ivan	5	5

$$\pi_{dName}(Developer)$$

dName
Ivan
Peter

Developer

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5
3	Ivan	5	5

$$\pi_{dName,\ 2*dCntProjects}(Developer)$$

dName	2*dCntProjects
Ivan	6
Peter	4
Ivan	10

SELECT **A, B, C** $\longrightarrow \pi$

FROM R

WHERE **A = 1**

AND $\longrightarrow \sigma$

B = 1

$$\pi_{a_1,\ldots,a_n}(\sigma_A(R)) = \sigma_A(\pi_{a_1,\ldots,a_n}(R))$$

$$A \subseteq \{a_1, \ldots, a_n\}$$

```
SELECT A, B, C
  FROM R
WHERE A = 1
        AND
        B = 1
```

```
SELECT *
FROM (
SELECT A, B, C
  FROM R )
WHERE A = 1
      AND
      B = 1
```

$$\pi_{a_1,\ldots,a_n}(\pi_{b_1,\ldots,b_m}(R)) = \pi_{a_1,\ldots,a_n}(R)$$

$$\{a_1, \ldots, a_n\} \subseteq \{b_1, \ldots, b_n\}$$

```
SELECT A, B, C
  FROM R
```

```
SELECT A, B
FROM (
SELECT A, B, C
   FROM R )
```

A	B
1	1
1	2
3	2

R

$\times$

W	Y	Z
1	1	1
3	2	1

S

$=$

A	B	W	Y	Z
1	1	1	1	1
1	1	3	2	1
1	2	1	1	1
1	2	3	2	1
3	2	1	1	1
3	2	3	2	1

Z

$$\pi_{*}\,(Developer \times Project \times Apply)$$

naming anomaly

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

pName	pManagerName	pPriority
Project #1	Ivan Ivanov	high

dID	pName	aPercentUsage
2	Project #1	100 %

$$\pi_* \, (Developer \times Project \times Apply)$$

naming anomaly

```
SELECT dID, pName
  FROM Developer
         CROSS JOIN Project
         CROSS JOIN Apply
```

Azat Yakupov

Rename

$$\rho_{A|B}(R)$$

$$\rho_{A \to B}(R)$$

A, B are attribute names

R is a relation

$A \to B$ or $(A \mid B)$ is renaming

Developer

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5
3	Ivan	5	5

$$\rho_{dID,\ dCntProjects\ |\ dCnt,\ dAvgPoint\ |dAvg}(Developer)$$

dID	dCnt	dAvg
1	3	5
2	2	3,5
3	5	5

$$\pi_{*}\ (Developer \times Project \times \rho_{dID|aID,\ pName|aName}(Apply))$$

no naming anomaly

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

pName	pManagerName	pPriority
Project #1	Ivan Ivanov	high

aID	aName	aPercentUsage
2	Project #1	100 %

```sql
SELECT A AS A_1,
       B AS B_2,
       C AS C_NEW
FROM R
```

$\longrightarrow \rho$

```sql
SELECT A AS A_1,
       B AS B_2,
       C AS C_NEW
  FROM R AS R_NEW
```

$\longrightarrow \rho$

Union

$$\pi_{r_1,\ldots,r_n}(R) \cup \pi_{s_1,\ldots,s_n}(S)$$

r_i, s_i are attribute names

R, S are relations

$$\pi_{dName}(Developer) \cup \pi_{pName}(Project)$$

naming anomaly

dName
Ivan
Peter

$\cup$

pName
Project #1

$=$

???
Ivan
Peter
Project #1

$$\pi_{dName}(Developer) \cup \rho_{pName|dName}(Project)$$

no naming anomaly

dName
Ivan
Peter

$\cup$

dName
Project #1

$=$

dName
Ivan
Peter
Project #1

```
SELECT A,
       B
  FROM R
UNION
SELECT A,
       C AS B
  FROM S
```

$\longrightarrow \cup$

Difference

$$\pi_{r_1,\ldots,r_n}(R) \setminus \pi_{s_1,\ldots,s_n}(S)$$

r_i, s_i are attribute names

R, S are relations

$$\pi_{dID}\,(Developer) \setminus \pi_{dID}\,(Project)$$

dID
1
2
3
4

$\setminus$

dID
2
3
4

$=$

dID
1

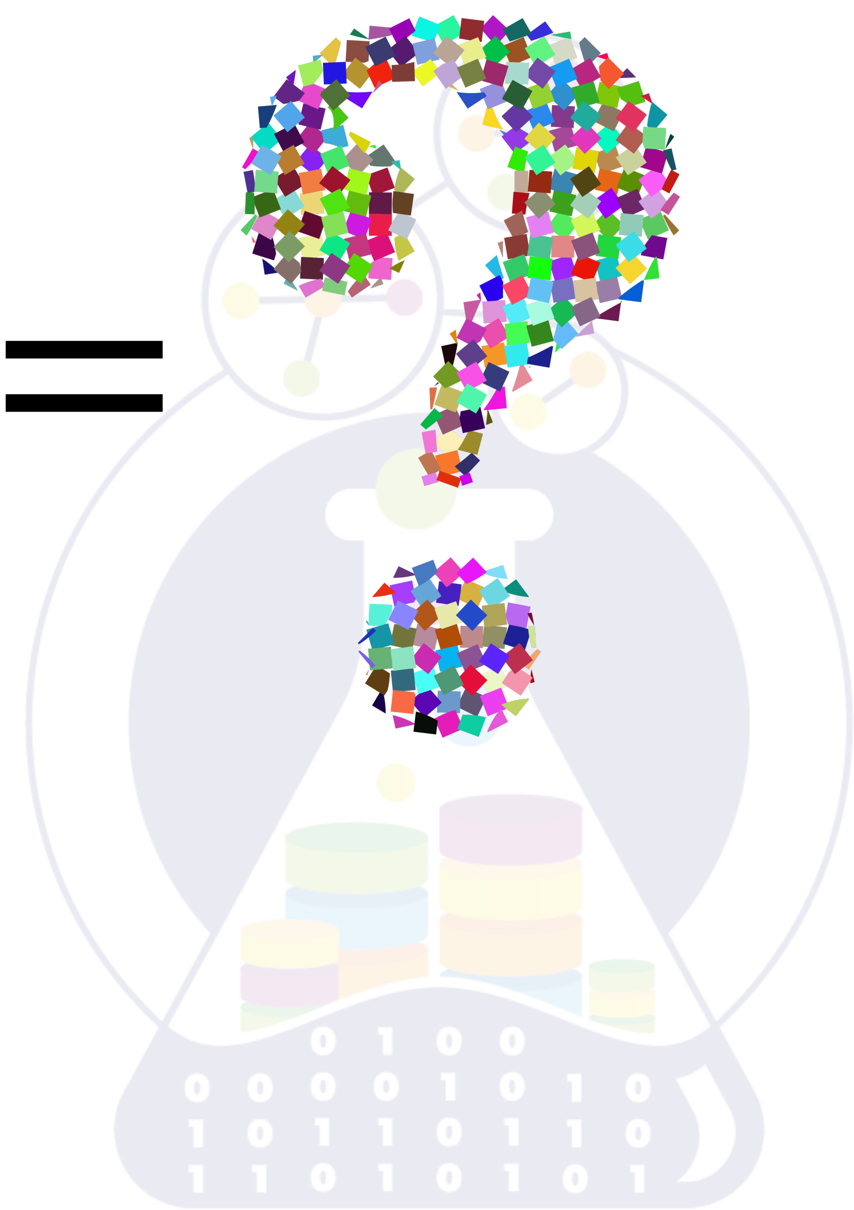
$$\pi_{dID}\,(Developer) \setminus \pi_{dID}\,(Project)$$

dID
2
3
4

$\setminus$

dID
1
2
3
4

$=$



```
SELECT A,
       B
  FROM R
MINUS
SELECT A,
       C AS B
  FROM S
```

$\setminus$

```
SELECT A,
       B
  FROM R
EXCEPT
SELECT A,
       C AS B
  FROM S
```

$\setminus$

Intersection

$$\pi_{r_1,\ldots,r_n}(R) \;\cap\; \pi_{s_1,\ldots,s_n}(S)$$

r_i, s_i are attribute names

R, S are relations

$$\pi_{dID}\,(Developer)\ \cap\ \pi_{dID}\,(Project)$$

dID
1
2
3
4

$\cap$

dID
2
3
4

$=$

dID
2
3
4

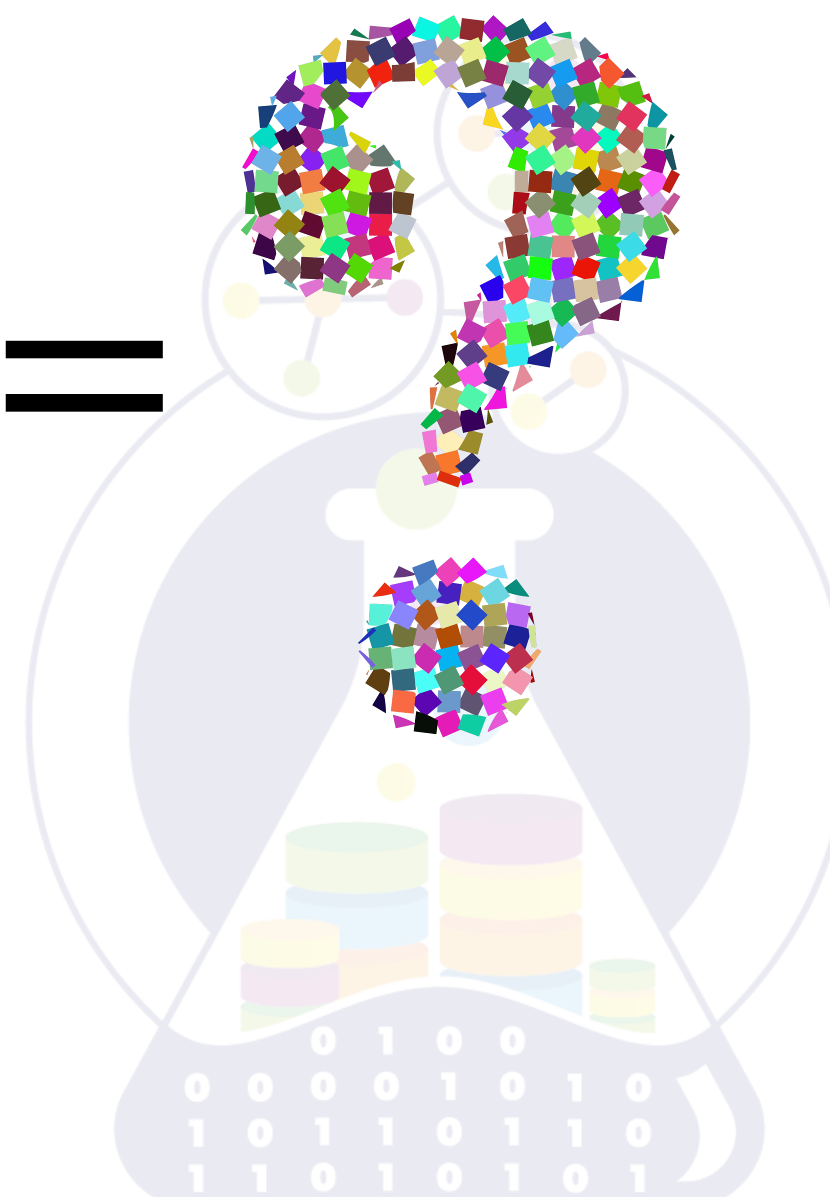
$$\pi_{dID}\,(Developer) \;\cap\; \pi_{dID}\,(Project)$$

dID
2
3
4

$\cap$

dID
1
2
3
4

$=$



```sql
SELECT A,
       B
  FROM R
INTERSECT
SELECT A,
       C AS B
  FROM S
```

$\cap$

- Natural Join
- θ - Join
- Semijoin
- Antijoin (~ semidifference)
- Division
- Left/Right/Full Outer Joins

Natural Join

$$R \bowtie S$$

R, S are relations

The result is a set of combinations of tuples between R and S using equal **common attribute names**

$$R \bowtie S = \sigma_{\theta}(R \times S)$$

$$\pi_{*}\,(Developer \bowtie Project \bowtie Apply)$$

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

$\bowtie$

pName	pManagerName	pPriority
Project #1	Ivan Ivanov	high

$\bowtie$

dID	pName	aPercentUsage
2	Project #1	100 %

=

dID	dName	dCntProjects	dAvgPoint	pName	pManager Name	pPriority	aPercent Usage
2	Peter	2	3,5	Project #1	Ivan Ivanov	high	100 %

$$R \bowtie S = \sigma_{\theta}(R \times S)$$

$$\pi_{*}\ (Developer \bowtie Project \bowtie Apply)$$

$$\sim$$

$$\pi_{*}\ (\sigma_{dID=aID \wedge pName=aName}(Developer \times Project \times \rho_{dID|aID,\ pName|aName}(Apply)))$$

A	B
1	1
1	2
3	2

R

$\bowtie$

W	Y	Z
1	1	1
3	2	1

S

$=$

no attributes with similar names

A	B
1	1
1	2
3	2

R

$\bowtie$

W	Y	Z
1	1	1
3	2	1

S

$=$

A	B	W	Y	Z
1	1	1	1	1
1	1	3	2	1
1	2	1	1	1
1	2	3	2	1
3	2	1	1	1
3	2	3	2	1

Z

```
SELECT R.A,
       S.B
FROM R NATURAL JOIN S
```

$$\bowtie$$

$$R \bowtie S = \sigma_\theta(R \times S)$$

SELECT R.A,
S.B
FROM R **CROSS JOIN** S $\times$
WHERE R.A = S.A AND
R.B = S.B σ_θ

θ-join

$$R \bowtie_{A\theta B} S$$

R, S are relations

θ is a binary operation in the set

$<, \ \leq, \ =, \ \neq, \ >, \ \geqslant$

A, B - attribute names

$$R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$$

$$\pi_{*}\ (Project \bowtie_{pPriority='high'} (Apply))$$

pName	pManagerName	pPriority
Project #1	Ivan Ivanov	high

$$\bowtie_{pPriority='high'}$$

dID	pName	aPercentUsage
2	Project #1	100 %

=

dID	pName	pManagerName	pPriority	aPercentUsage
2	Project #1	Ivan Ivanov	high	100 %

```sql
SELECT R.A,
       S.B
  FROM R NATURAL JOIN S
WHERE R.C = 123
```

$\bowtie_{\theta}$

Semijoin

$$R \ltimes S$$

$$R \rtimes S$$

R, S are relations

$$R \ltimes S = \pi_{a_1,\ldots,a_n}(R \bowtie S), \{a_1, \ldots, a_n\} \in R$$

$$R \rtimes S = \pi_{a_1,\ldots,a_n}(R \bowtie S), \{a_1, \ldots, a_n\} \in S$$

$$\pi_* \, (Developer \ltimes Apply)$$

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

$\ltimes$

dID	pName	aPercentUsage
2	Project #1	100 %

=

dID	dName	dCntProjects	dAvgPoint
2	Peter	2	3,5

$$\pi_* \; (Developer \rtimes Apply)$$

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

$\rtimes$

dID	pName	aPercentUsage
2	Project #1	100 %

$=$

dID	pName	aPercentUsage
2	Project #1	100 %

```
SELECT *
  FROM R
WHERE A IN (SELECT A
              FROM S)
```

$\longrightarrow \ltimes$

```
SELECT *
  FROM R
WHERE EXISTS (SELECT A
                  FROM S
                WHERE  S.A = R.A)
```

$\longrightarrow \ltimes$

```
SELECT *
  FROM S
WHERE A IN (SELECT A
              FROM R)
```

$\ltimes$

```sql
SELECT *
  FROM S
WHERE EXISTS (SELECT A
                FROM R
              WHERE  S.A = R.A)
```

$\longrightarrow \rtimes$

Antijoin (~ semidifference)

$$R \triangleright S$$

R, S are relations

$$R \triangleright S = R \setminus (R \ltimes S)$$

$$\pi_{*}\left(Developer \triangleright Apply\right)$$

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

$$\triangleright$$

dID	pName	aPercentUsage
2	Project #1	100 %

$$=$$

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5

```
SELECT *
  FROM S
WHERE A NOT IN (SELECT A
                  FROM R)
```

$\triangleright$

```
SELECT *
  FROM R
WHERE NOT EXISTS (SELECT A
                           FROM S
                    WHERE  S.A = R.A)
```

$\triangleright$

Left / Right joins

$$R ⟕ S$$

$$R ⟖ S$$

R, S are relations

$$\pi_{*}\left(Developer ⟕ Apply\right)$$

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

$⟕$

dID	pName	aPercentUsage
2	Project #1	100 %

$=$

dID	dName	dCntProjects	dAvgPoint	pName	aPercentUsage
1	Ivan	3	5	ω	ω
2	Peter	2	3,5	Project #1	100 %

$$\pi_{*}\left(Developer \rtimes Apply\right)$$

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

$\rtimes$

dID	pName	aPercentUsage
2	Project #1	100 %
3	Project #3	75 %

=

dID	dName	dCntProjects	dAvgPoint	pName	aPercentUsage
3	ω	ω	ω	Project #3	75 %
2	Peter	2	3,5	Project #1	100 %

⟕

SELECT *

FROM **S LEFT JOIN R ON S.A = R. B**

⟖

```
SELECT *
  FROM S RIGHT JOIN R ON S.A = R. B
```

Full join

$$R ⟗ S$$

R, S are relations

$$R ⟗ S = (R ⟖ S) \cup (R ⟕ S)$$

$$\pi_{*}\left(Developer ⟗ Apply\right)$$

dID	dName	dCntProjects	dAvgPoint
1	Ivan	3	5
2	Peter	2	3,5

⟗

dID	pName	aPercentUsage
2	Project #1	100 %
3	Project #3	75 %

=

dID	dName	dCntProjects	dAvgPoint	pName	aPercentUsage
1	Ivan	3	5	ω	ω
2	Peter	2	3,5	Project #1	100 %
3	ω	ω	ω	Project #3	75 %

SELECT *

FROM S FULL JOIN R ON S.A = R. B

⟗

Division

$$R \div S$$

R, S are relations

$$\pi_{*}\,(R \div S)$$

A	B	C
1	1	2
2	3	4
3	1	2
4	5	6
1	5	6
4	1	2
1	3	4

$\div$

B	C
1	2
5	6

$=$

A
1
4

SELECT *

FROM **S** ***“DIVISION”*** **R**

Extended Operators

- duplicate-elimination
- aggregation operators
- grouping operator
- extended projection
- sorting operator

Duplicate Elimination

$$\delta(R)^{*}$$

$R =$

A	B
1	1
1	2
3	2
1	1
1	1
3	2

$\delta(R) =$

A	B
1	1
1	2
3	2

* for real physical database systems

Aggregation Operators

$R =$

A	B
1	1
1	2
3	2

SUM(B) = 1+2+2 = 5

AVG(A) = (1+1+3)/3

MIN(A) = 1

MAX(B) = 2

COUNT(A) = 3

Grouping Operator

$$\gamma_L(R)$$

$R =$

A	B
1	1
1	2
3	2
3	4
2	1

$\gamma_A(R) =$

A
1
3
2

$$\rho_{A,\ count(B)\ |cnt,\ min(B)\ |min}(\gamma_{A,\ count(B),\ min(B)}(R))$$

$R =$

A	B
1	1
1	2
3	2
3	4
2	1

A	cnt	min
1	2	1
3	2	2
2	1	1

Extended Projection

$x \rightarrow y$ $\quad$ x, y are attributes

$R =$

A	B
1	1
1	2
3	2
1	1
1	1
3	2

$\pi_{A,A+B \rightarrow C}(R) =$

A	C
1	2
1	3
3	5
1	2
1	2
3	5

Sorting Operator

$$\tau_L(R)^*$$

$R =$

A	B
1	1
1	2
3	2
1	1
1	1
3	2

$\tau_{A,B}(R) =$

A ↓	B ↓
1	1
1	1
1	1
1	2
3	2
3	2

* for real physical database systems

Azat Yakupov

Datalog (database logic)

Relational Algebra

Relation R

Relation with attributes

$R(A, B, C)$

Datalog

Predicate R

Atom with attributes

$R(A, B, C)$

Azat Yakupov

Atom $R(A, B, C, D) = TRUE$

if $(A, B, C, D) \in R$

$R =$

A	B
1	2
3	4

$R(1,2) = TRUE$

$R(3,4) = TRUE$

$R(3,5) = FALSE$

$$R = \pi_{dID,dName}(\sigma_{dAvgPoint \geqslant 4.0}\ (Developer))$$

$$\underbrace{\underbrace{R(di, dn)}_{\text{head}} \leftarrow \underbrace{Developer(di, dn, dc, da)\ AND\ da \geqslant 4}_{\text{body}}}_{\text{datalog rule}}$$

Datalog query

$$\pi_{dID,dName}(\sigma_{dAvgPoint \geqslant 4.0}(Developer) \cap \sigma_{dCntProjects=3}(Developer))$$

1. $W(di, dn, dc, da) \leftarrow Developer(di, dn, dc, da)\ AND\ da \geq 4.0$
2. $X(di, dn, dc, da) \leftarrow Developer(di, dn, dc, da)\ AND\ dc = 3$
3. $Y(di, dn, dc, da) \leftarrow W(di, dn, dc, da)\ AND\ X(di, dn, dc, da)$
4. $Answer(di, dn) \leftarrow Y(di, dn, dc, da)$

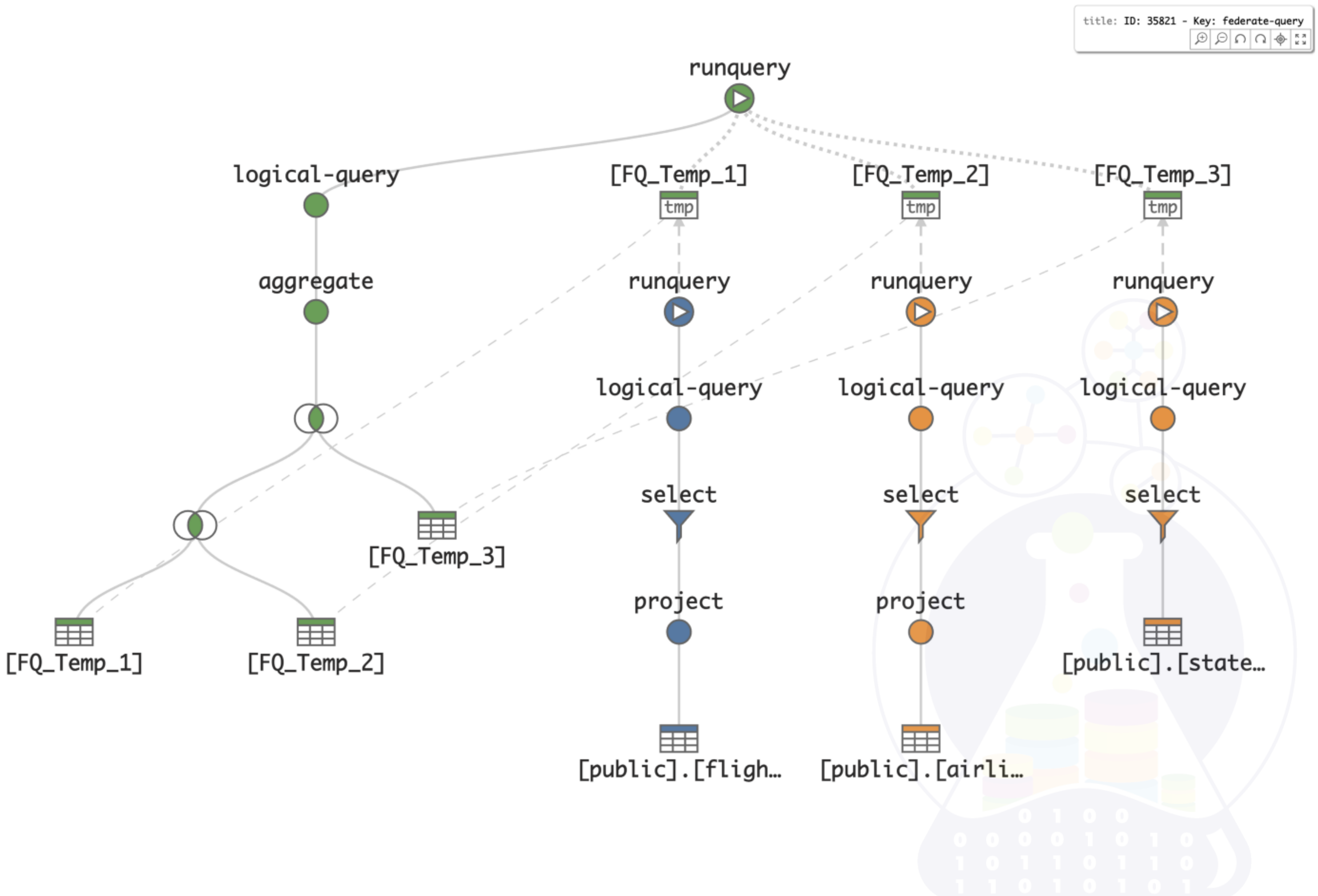
Relational Algebra	Datalog
grouping operator	ω
aggregation operators	ω
duplicate elimination	ω
ω	recursion rule

$$Path(X, Y) \leftarrow Edge(X, Y)$$

$$Path(X, Y) \leftarrow Edge(X, Z)\ AND\ Path(Z, Y)$$

$$\begin{tikzcd}
 & \pi_{pID,\ pName} & \\
 & \cap \arrow[u] & \\
\sigma_{dAvgPoint \geq 4.0} \arrow[ur] & & \sigma_{dCntProjects = 3} \arrow[ul] \\
Developer \arrow[u] & & Developer \arrow[u]
\end{tikzcd}$$

Why do we need “Relational Algebra”



Azat Yakupov

COMMIT;