

## 1. Natural Join

**?** Notation:  $r \bowtie s$

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

$$S = (E, B, D)$$

**?** Result schema =  $(A, B, C, D, E)$

**?**  $r \bowtie s$  is defined as:

$$\Pi_{r.A, r.B, r.C, r.D, s.E} (\sigma_{r.B = s.B \wedge r.D = s.D} (r \times s))$$

**?** Relations  $r, s$ :

A	B	C	D
$\alpha$	1	$\alpha$	a
$\beta$	2	$\gamma$	a
$\gamma$	4	$\beta$	b
$\alpha$	1	$\gamma$	a
$\delta$	2	$\beta$	b

$r$

B	D	E
1	a	$\alpha$
3	a	$\beta$
1	a	$\gamma$
2	b	$\delta$
3	b	$\epsilon$

$s$

**?**  $r \bowtie s$

A	B	C	D	E
$\alpha$	1	$\alpha$	a	$\alpha$
$\alpha$	1	$\alpha$	a	$\gamma$
$\alpha$	1	$\gamma$	a	$\alpha$
$\alpha$	1	$\gamma$	a	$\gamma$
$\delta$	2	$\beta$	b	$\delta$

Columns of the same name will be appear only once.

```
SELECT *
FROM Table1
NATURAL JOIN Table2;      (When working for Table 1 and Table 2)
```

```
,A,B,C,D,E
0,Alpha,1,Alpha,a,Alpha
1,Alpha,1,Alpha,a,Gamma
2,Alpha,1,Gamma,a,Alpha
3,Alpha,1,Gamma,a,Gamma
4,Rho,2,Beta,b,rho
```

```
SELECT *
FROM ratings
NATURAL JOIN tags;      (When working for ratings.csv and tags.csv)
```

## 2. Inner Join

This join selects records that have matching values in both table.

```
SELECT *
FROM Table1
INNER JOIN Table2 ON Table1.A>Table2.E
(When working for Table 1 and Table 2)
```

```
SELECT *
FROM ratings
INNER JOIN tags ON Table1.movieID>Table2.userID
(When working for ratings.csv and tags.csv)
```

**2.a Left Inner Join :** This join works similar to the one above, it just selects all records from the left table even if there are no matches in the right table. The result is Null on the right side.

```
SELECT *
FROM Table1
LEFT JOIN Table2 ON Table1.A>Table2.E
(When working for Table 1 and Table 2)
```

```
SELECT *
FROM ratings
LEFT JOIN tags ON Table1.movieID>Table2.userID
```

To calculate the left outer join, I am using `numpy.setdiff1d()` to find all the rows of `df_t1` that not in the inner join

Difference between natural and Inner join?

**Inner join, join** two table where column name is same. **Natural join, join** two table where column name and data types are same

**3. Aggregate function-** Takes a collection of values and returns a single value.

There are 5 types-

average, min, max , sum, count.

Implemented count on either .csv file depending on what the user wants. The output shows the counts of each value in each column of the table.

`G(count(all))^ratings/tags`