# 1. Unary Relational Operations: SELECT

**SELECT statement**

```
select distinct *
from SECTION
where Year > 2016;
```

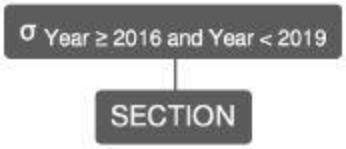
σ Year > 2016
SECTION

$\sigma_{\text{Year} > 2016}$ SECTION

| SECTION.Section_id | SECTION.Course_number | SECTION.Semester | SECTION.Year | SECTION.Instructor |
|---|---|---|---|---|
| KJ2 | CCPS305 | Fall | 2018 | Harry |
| YJ2 | CIND110 | Winter | 2019 | Larry |
| YJ3 | CIND110 | Fall | 2018 | Sally |
| KJ3 | CIND110 | Winter | 2019 | Garry |

**SELECT statement**

```
select  distinct *
from  SECTION
where  Year  >= 2016  and  Year < 2019;
```



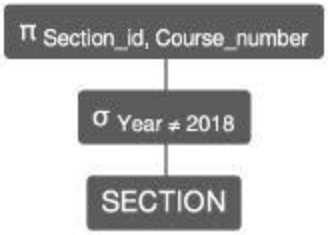
$\sigma_{\text{Year} \geq 2016 \text{ and Year} < 2019}$ SECTION

| SECTION.Section_id | SECTION.Course_number | SECTION.Semester | SECTION.Year | SECTION.Instructor |
|---|---|---|---|---|
| YJ5 | CIND123 | Fall | 2016 | Sally |
| KJ2 | CCPS305 | Fall | 2018 | Harry |
| YJ3 | CIND110 | Fall | 2018 | Sally |

## 2. Unary Relational Operations: PROJECT

**PROJECT statement**

```
select distinct Section_id, Course_number
from SECTION
where Year <> 2018;
```

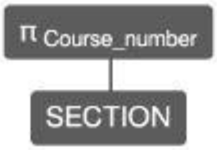


$\pi_{\text{Section\_id, Course\_number}}\ \sigma_{\text{Year} \neq 2018}$ SECTION

| SECTION.Section_id | SECTION.Course_number |
|---|---|
| YJ5 | CIND123 |
| YJ2 | CIND110 |
| KJ3 | CIND110 |

**PROJECT statement**

```
select distinct Course_number
from SECTION;
```



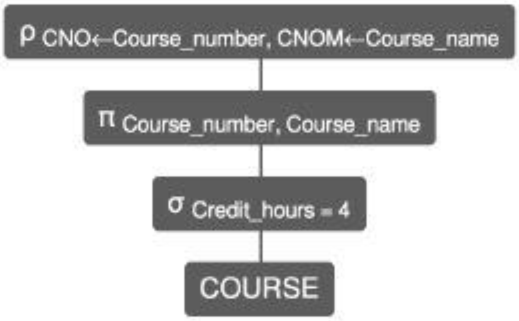
$\pi_{\text{Course\_number}}$ SECTION

| SECTION.Course_number |
|---|
| CIND123 |
| CCPS305 |
| CIND110 |

# 3. Unary Relational Operations: RENAME

**As statement**

```
select distinct Course_number as CNO, Course_name
   as CNOM
from COURSE
Where Credit_hours = 4;
```



ρ CNO←Course_number, CNOM←Course_name
π Course_number, Course_name
σ Credit_hours = 4
COURSE

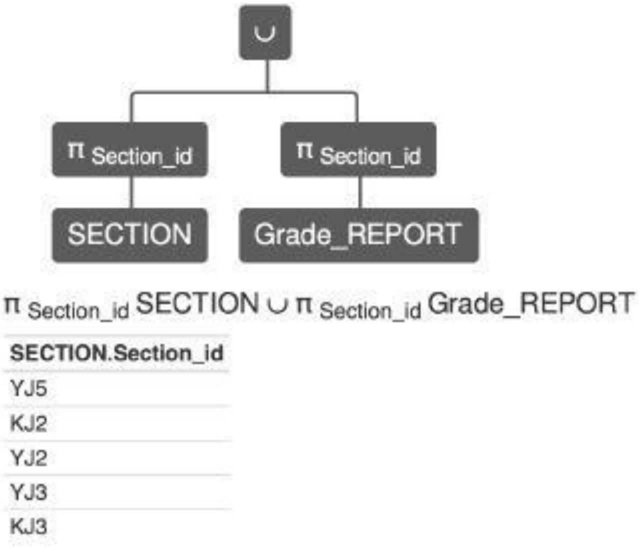
$\rho_{\text{CNO}\leftarrow\text{Course\_number, CNOM}\leftarrow\text{Course\_name}}\ \pi_{\text{Course\_number, Course\_name}}\ \sigma_{\text{Credit\_hours} = 4}\ \text{COURSE}$

| COURSE.CNO | COURSE.CNOM |
|---|---|
| CPS305 | Data Structures |
| CIND110 | Data Organization |
| CIND123 | Data Analytics |

## 4. Operations from Set Theory

### Union

```
SELECT distinct Section_id FROM SECTION
UNION
SELECT distinct Section_id FROM Grade_REPORT;
```

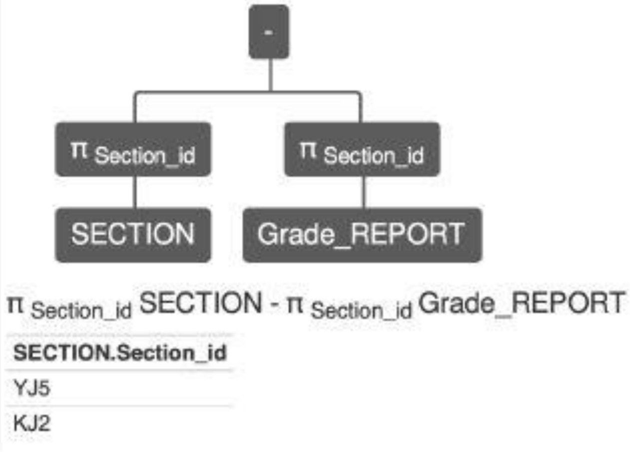


$\pi_{\text{Section\_id}}$ SECTION $\cup$ $\pi_{\text{Section\_id}}$ Grade_REPORT

| SECTION.Section_id |
|---|
| YJ5 |
| KJ2 |
| YJ2 |
| YJ3 |
| KJ3 |

### Except

```
SELECT distinct Section_id FROM SECTION
EXCEPT
SELECT distinct Section_id FROM Grade_REPORT;
```

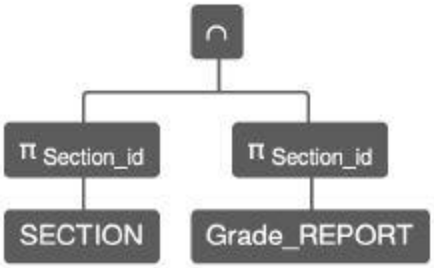


$\pi_{\text{Section\_id}}$ SECTION - $\pi_{\text{Section\_id}}$ Grade_REPORT

| SECTION.Section_id |
|---|
| YJ5 |
| KJ2 |

### INTERSECT

```
SELECT distinct Section_id FROM SECTION
INTERSECT
SELECT distinct Section_id FROM Grade_REPORT;
```



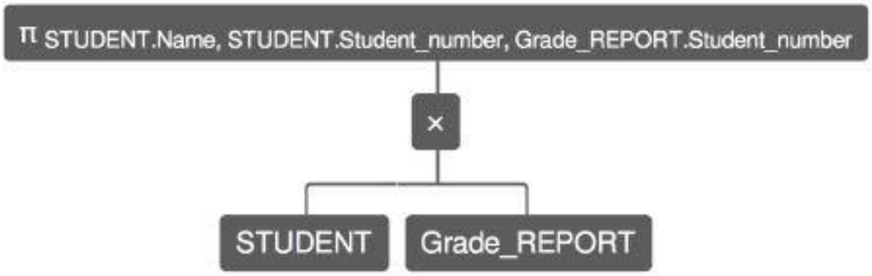
$\pi_{\text{Section\_id}}\ \text{SECTION} \cap \pi_{\text{Section\_id}}\ \text{Grade\_REPORT}$

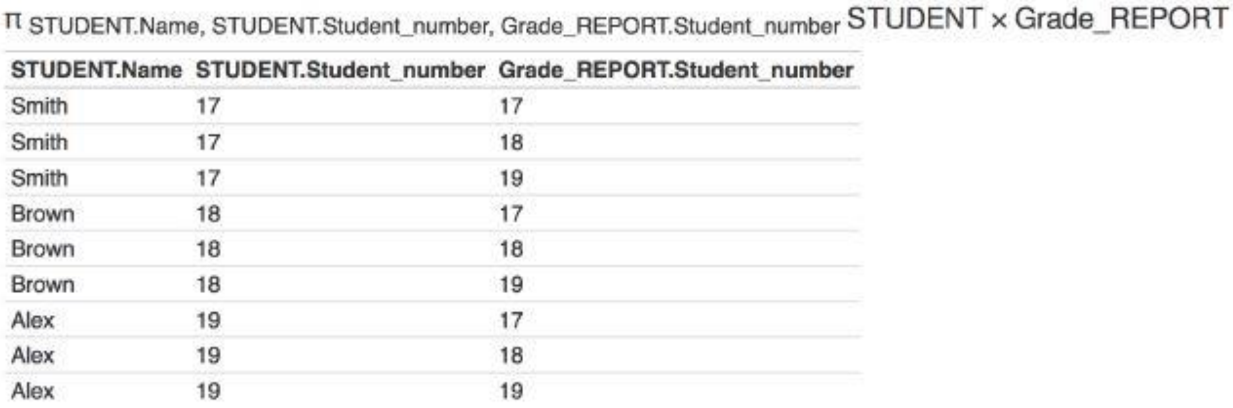
| SECTION.Section_id |
|---|
| YJ2 |
| YJ3 |
| KJ3 |

## 5. The Cartesian Product (CROSS PRODUCT)

**Union**

```
SELECT distinct STUDENT.Name, STUDENT.
   Student_number, Grade_REPORT.Student_number
FROM STUDENT
CROSS JOIN
Grade_REPORT;
```


π STUDENT.Name, STUDENT.Student_number, Grade_REPORT.Student_number
×
STUDENT
Grade_REPORT

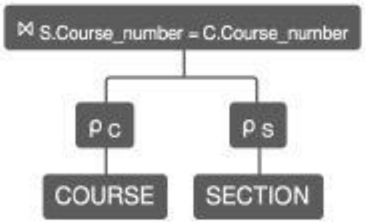

π STUDENT.Name, STUDENT.Student_number, Grade_REPORT.Student_number STUDENT × Grade_REPORT

| STUDENT.Name | STUDENT.Student_number | Grade_REPORT.Student_number |
|---|---|---|
| Smith | 17 | 17 |
| Smith | 17 | 18 |
| Smith | 17 | 19 |
| Brown | 18 | 17 |
| Brown | 18 | 18 |
| Brown | 18 | 19 |
| Alex | 19 | 17 |
| Alex | 19 | 18 |
| Alex | 19 | 19 |

## 6. Binary Relational Operations

### JOIN

```
SELECT distinct *
FROM COURSE as C
INNER JOIN SECTION as S
ON S.Course_number = C.Course_number;
```

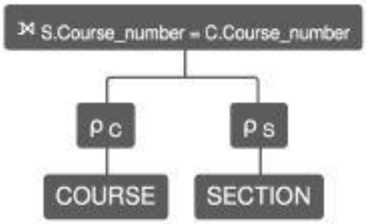


$\rho_{C}\,\text{COURSE} \bowtie_{\text{S.Course\_number} = \text{C.Course\_number}} \rho_{S}\,\text{SECTION}$

| C.Course_name | C.Course_number | C.Credit_hours | C.Department | S.Section_id | S.Course_number | S.Semester | S.Year | S.Instructor |
|---|---|---|---|---|---|---|---|---|
| Data Organization | CIND110 | 4 | MIE | YJ2 | CIND110 | Winter | 2019 | Larry |
| Data Organization | CIND110 | 4 | MIE | YJ3 | CIND110 | Fall | 2018 | Sally |
| Data Organization | CIND110 | 4 | MIE | KJ3 | CIND110 | Winter | 2019 | Garry |
| Data Analytics | CIND123 | 4 | MIE | YJ5 | CIND123 | Fall | 2016 | Sally |

### LEFT JOIN

```
SELECT distinct *
FROM COURSE as C
LEFT JOIN SECTION as S
ON S.Course_number = C.Course_number;
```



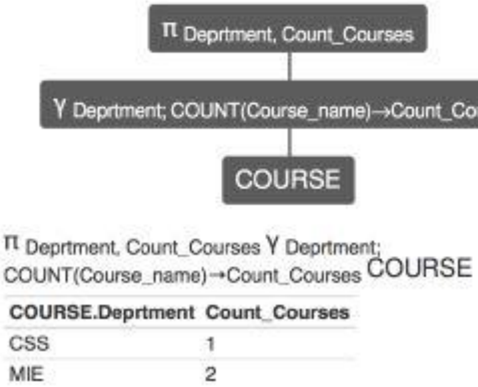
$\rho_{C}\,\text{COURSE} ⟕_{\text{S.Course\_number} = \text{C.Course\_number}} \rho_{S}\,\text{SECTION}$

| C.Course_name | C.Course_number | C.Credit_hours | C.Department | S.Section_id | S.Course_number | S.Semester | S.Year | S.Instructor |
|---|---|---|---|---|---|---|---|---|
| Data Structures | CPS305 | 4 | CSS | *null* | *null* | *null* | *null* | *null* |
| Data Organization | CIND110 | 4 | MIE | YJ2 | CIND110 | Winter | 2019 | Larry |
| Data Organization | CIND110 | 4 | MIE | YJ3 | CIND110 | Fall | 2018 | Sally |
| Data Organization | CIND110 | 4 | MIE | KJ3 | CIND110 | Winter | 2019 | Garry |
| Data Analytics | CIND123 | 4 | MIE | YJ5 | CIND123 | Fall | 2016 | Sally |

# 7. Aggregate Functions and Grouping

**COUNT**

```
SELECT Department, COUNT(Course_name) As
   Count_Courses
FROM COURSE
Group by Department;
```


π Deprtment, Count_Courses
γ Deprtment; COUNT(Course_name)→Count_Courses
COURSE

$\pi_{\text{Deprtment, Count\_Courses}}\ \gamma_{\text{Deprtment; COUNT(Course\_name)}\rightarrow\text{Count\_Courses}}$ COURSE

| COURSE.Deprtment | Count_Courses |
|---|---|
| CSS | 1 |
| MIE | 2 |