

```
ages_sum = {}
ages_count = {}

with open_csv('data.csv', 'r') as f:
    for tuple in f:
        rank = fields[2]
        if rank not in ages_sum:
            ages_sum[rank] = 0
        if rank not in ages_count:
            ages_count[rank] = 0
        ages_sum[rank] += fields[3]
        ages_count[rank] += 1
```

For L

Student (sid, name)

Course (cid, title)

~~Signed Up For (sid, cid)~~

Taken

```
for student in 'students.csv':  
    if student.name == "James T. Kirk":
```

```
        for taken in 'taken.csv':
```

```
            if taken.sid == student.sid:
```

```
                for course in 'courses.csv':
```

```
                    if taken.cid == course.cid
```

```
                        print course.title
```

NL5

filter (

student.name =

1 cid = cid

1 sid = sid

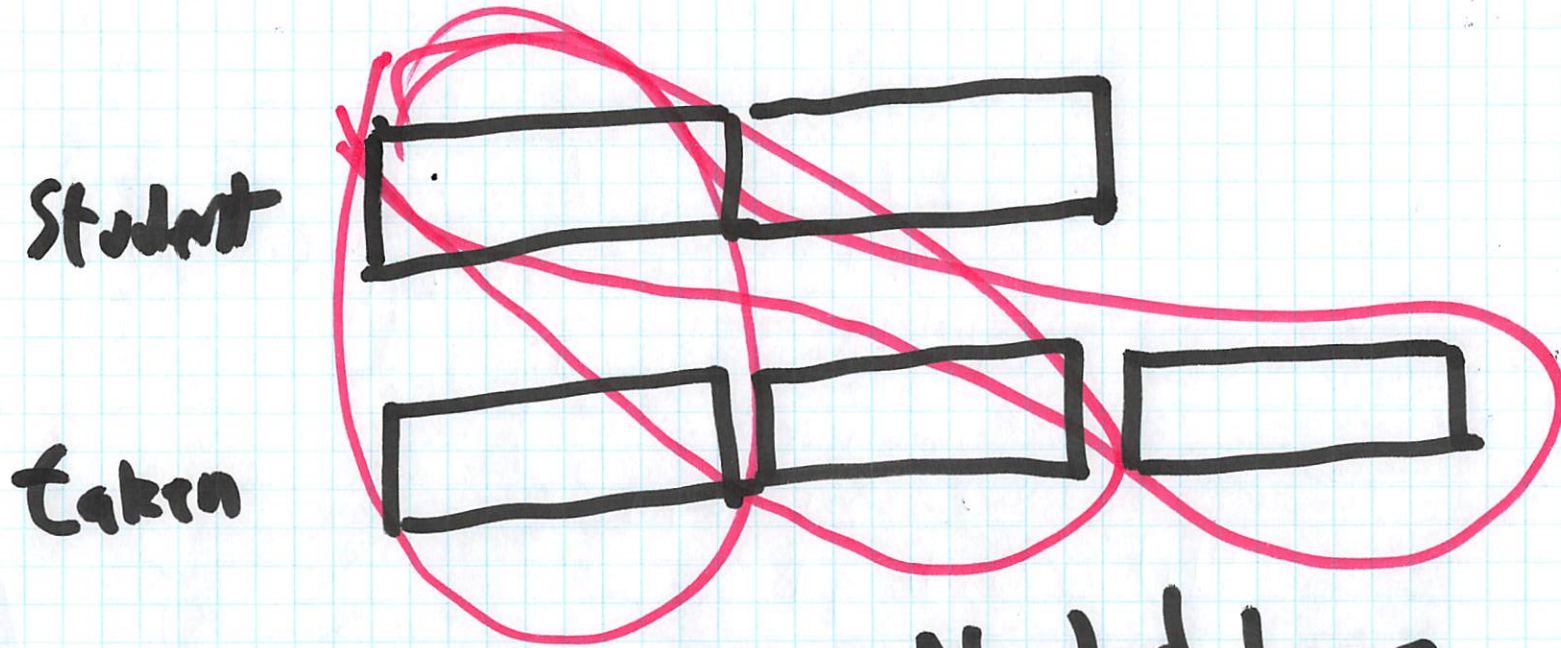
(student x taken x course)

SELECT title

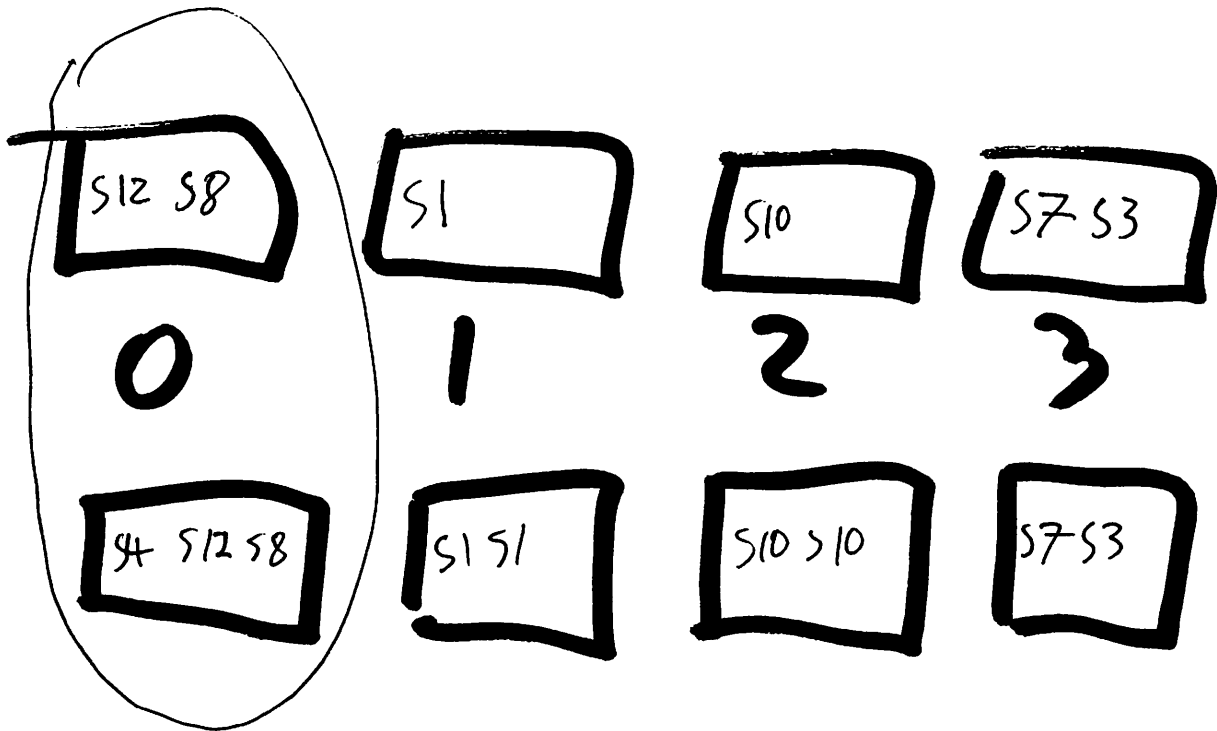
FROM student, taken,
course

= WHERE student.name =
AND cid = cid
AND sid = sid

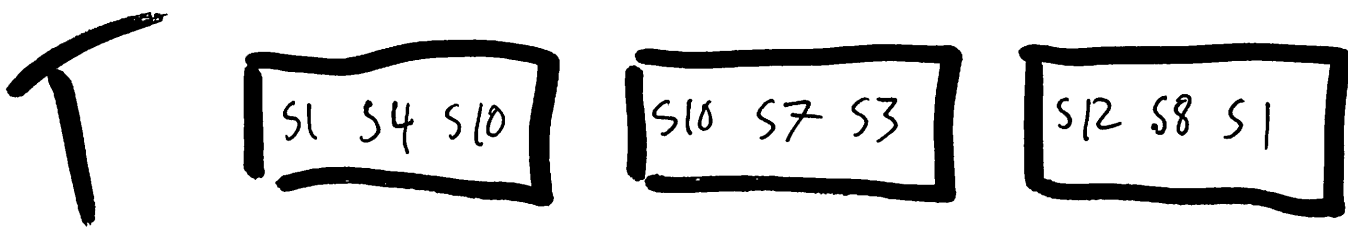
```
SELECT name, cid
FROM student, taken
WHERE student.sid = conn. taken.sid
```



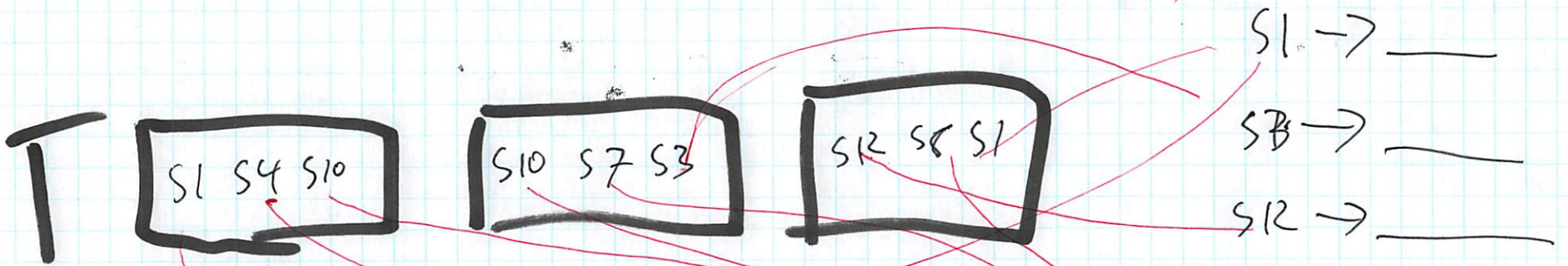
Block Nested Loop



2-pass
Hash Index
Join



Step 1 build dictionary on S



Step 2
Scan through T, find pairings

1-pass Hash Index Join

Optional

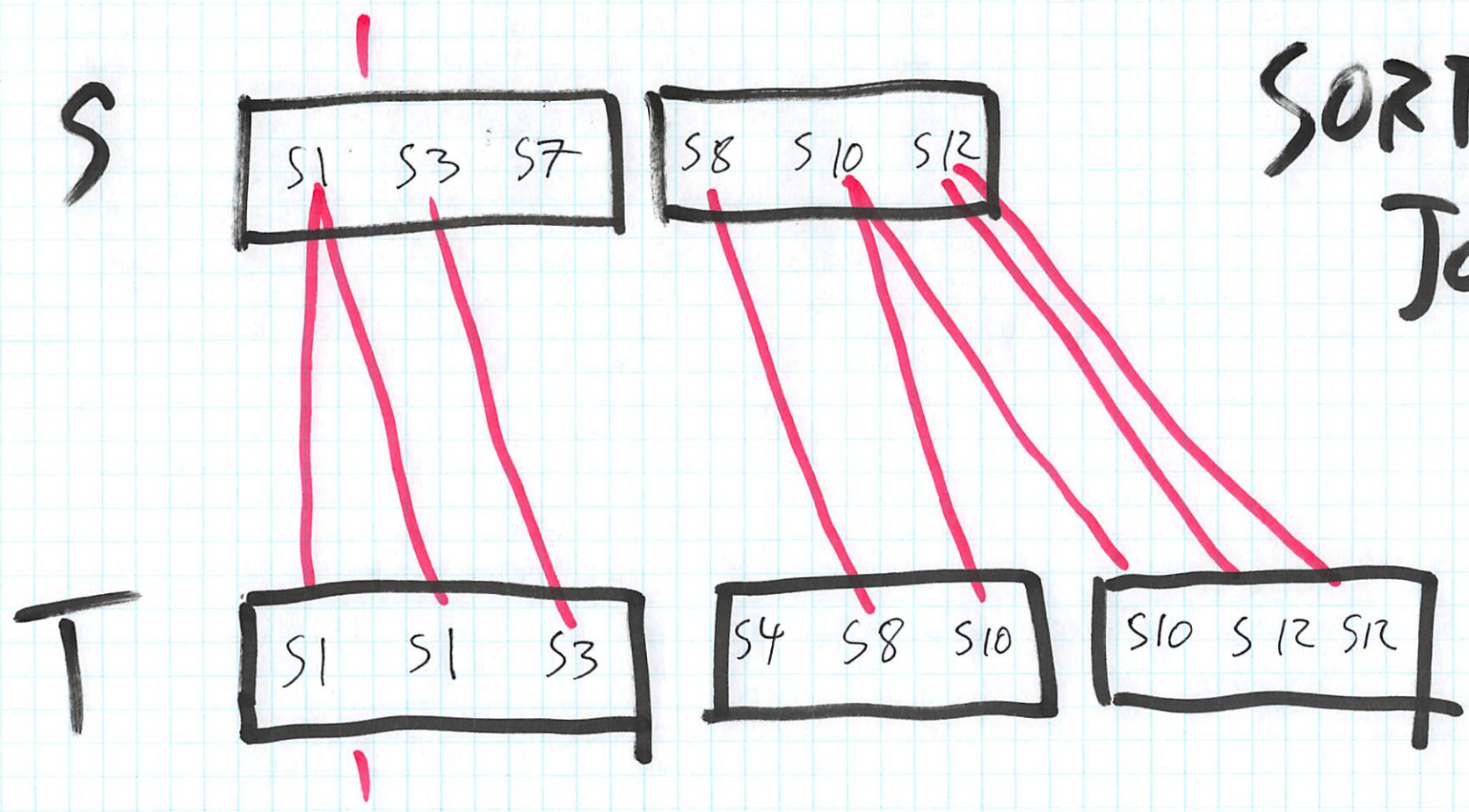
S1, S3, S8, S7, S10, S12

1-pass Tree Index Join

```
for block1 in students.csv  
for block2 in taken.csv  
for student in block1  
for taken in block2
```

~~Block~~
~~Index~~ Nested
Loop Join

Sort-Merge Join



Have index on T

```
for student in 'student.csv'  
  for T.index lookup(student)  
    ^  
    taken in  
      print student.name, taken.cid
```

Index Nested Loop Join