QUERIES WITH QUANTIFIERS (PART 2)

Venn Diagram and SQL Templates

EXAMPLE DATABASE SCHEMA

Student

sid	sname	major	byear

Course

cno	cname	dept

Enroll

sid	cno	grade

Department

dept	location

QUERIES WITH QUANTIFIERS

- Find the sid of each student who takes some CS courses
- Find the sid of each student who takes no CS courses
- Find the sid of each student who takes not only CS courses
- Find the sid of each student who takes only CS courses
- Find the sid of each student who takes not all CS courses
- Find the sid of each student who takes all CS courses

QUERIES WITH QUANTIFIERS

- Find each (s,d) pair such that student s takes some courses offered by department d
- Find each (s,d) pair such that student s takes all courses offered by department d
- Find each (s,d) pair such that student s takes fewer than 5 courses offered by department d
- …

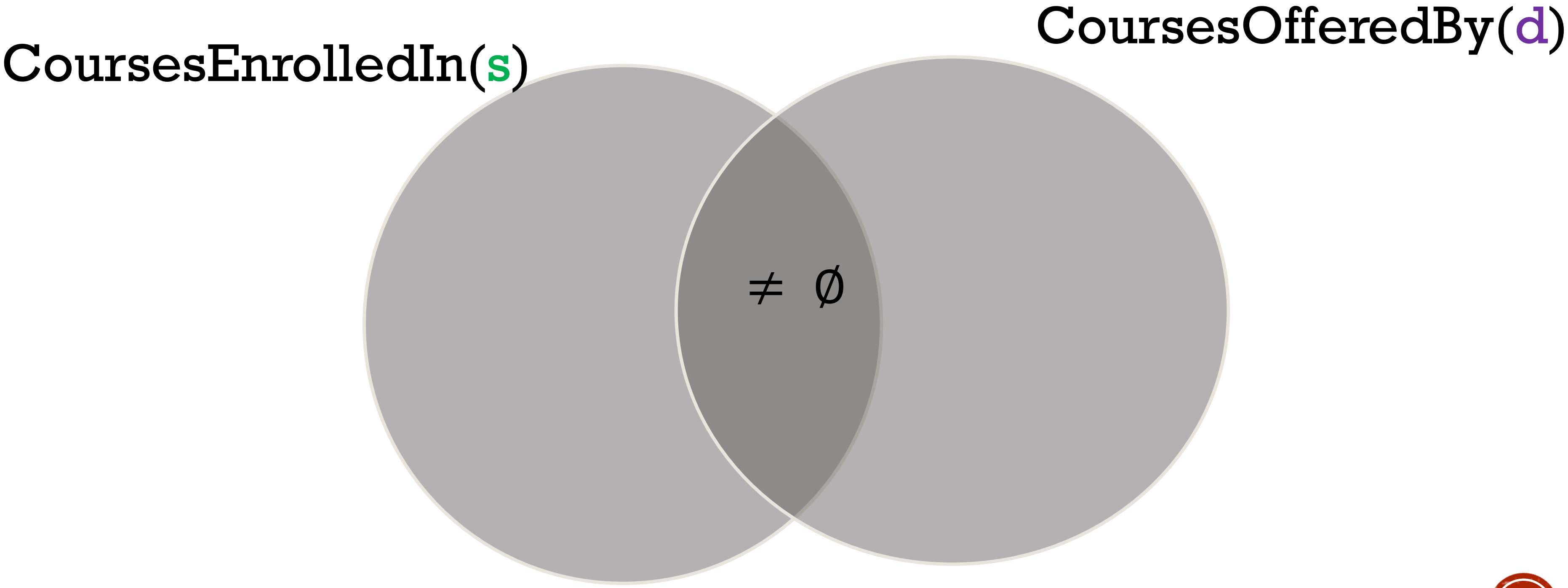
QUERIES WITH QUANTIFIERS RETURNING PAIRS

Find each (s,d) pair such that student s takes ‘quantifier’ courses offered by department d.

quantifier
some
no
not only
only
not all
all
all but only
at least 2
…

SOME

Find each (s,d) pair such that student s takes some courses offered by department d



CoursesEnrolledIn(s)

CoursesOfferedBy(d)

$\neq \emptyset$

DEFINING RELEVANT SETS WITH VIEWS

- Definition of CoursesEnrolledIn(sid)

```
CREATE FUNCTION CoursesEnrolledIn(sid TEXT)
    RETURNS TABLE (cno TEXT) AS
    $$
      SELECT  E.cno
      FROM     Enroll E
      WHERE  E.sid = CoursesEnrolledIn.sid;
    $$  LANGUAGE SQL
```

DEFINING RELEVANT SETS WITH VIEWS

- Definition of CoursesOfferedBy(dept)

```
CREATE FUNCTION Courses OfferedBy(dept TEXT)
    RETURNS TABLE (cno TEXT) AS
    $$
      SELECT   C.cno
      FROM     Course C
      WHERE   C.dept = CoursesOfferedBy.dept;
    $$  LANGUAGE SQL
```

SOME FOR PAIRS

$$\boldsymbol{A} \cap \boldsymbol{B} \neq \emptyset \qquad\qquad \exists\, \boldsymbol{x}\, (\boldsymbol{x} \in \boldsymbol{A} \cap \boldsymbol{B})$$

```
SELECT S.sid, D.dept
FROM   Student S, Department D
WHERE EXISTS (SELECT cno
              FROM   CoursesEnrolledIn(S.sid)
                INTERSECT
              SELECT cno
              FROM   CoursesOfferedBy(D.dept))
```

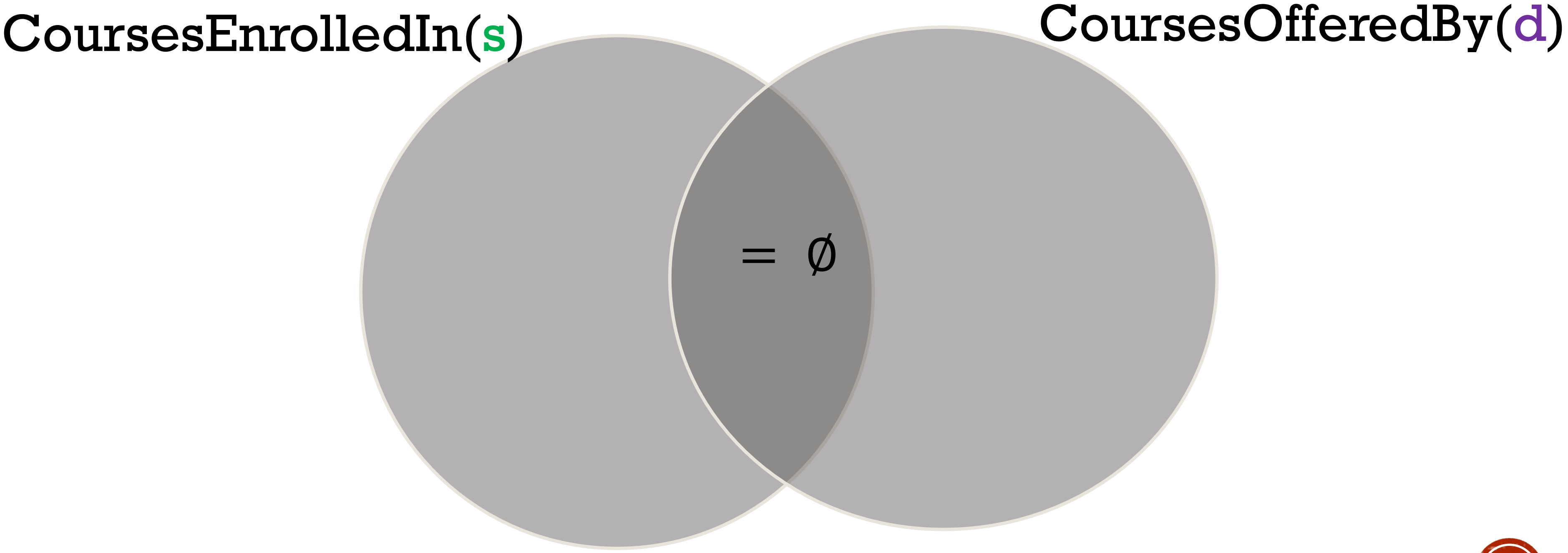
SOME FOR PAIRS

```
SELECT  S.sid, D.dept
FROM    Student S, Department D
WHERE EXISTS (SELECT  cno
              FROM    CoursesEnrolledIn(S.sid)
              WHERE
                   cno IN (SELECT cno
                           FROM    CoursesOfferedBy(D.dept)
```

NO FOR PAIRS

$\boldsymbol{A} \cap \boldsymbol{B} = \emptyset \qquad \neg \exists \boldsymbol{x} (\boldsymbol{x} \in \boldsymbol{A} \cap \boldsymbol{B})$

Find each (s,d) pair such that student s takes no courses offered by department d



CoursesEnrolledIn(s)

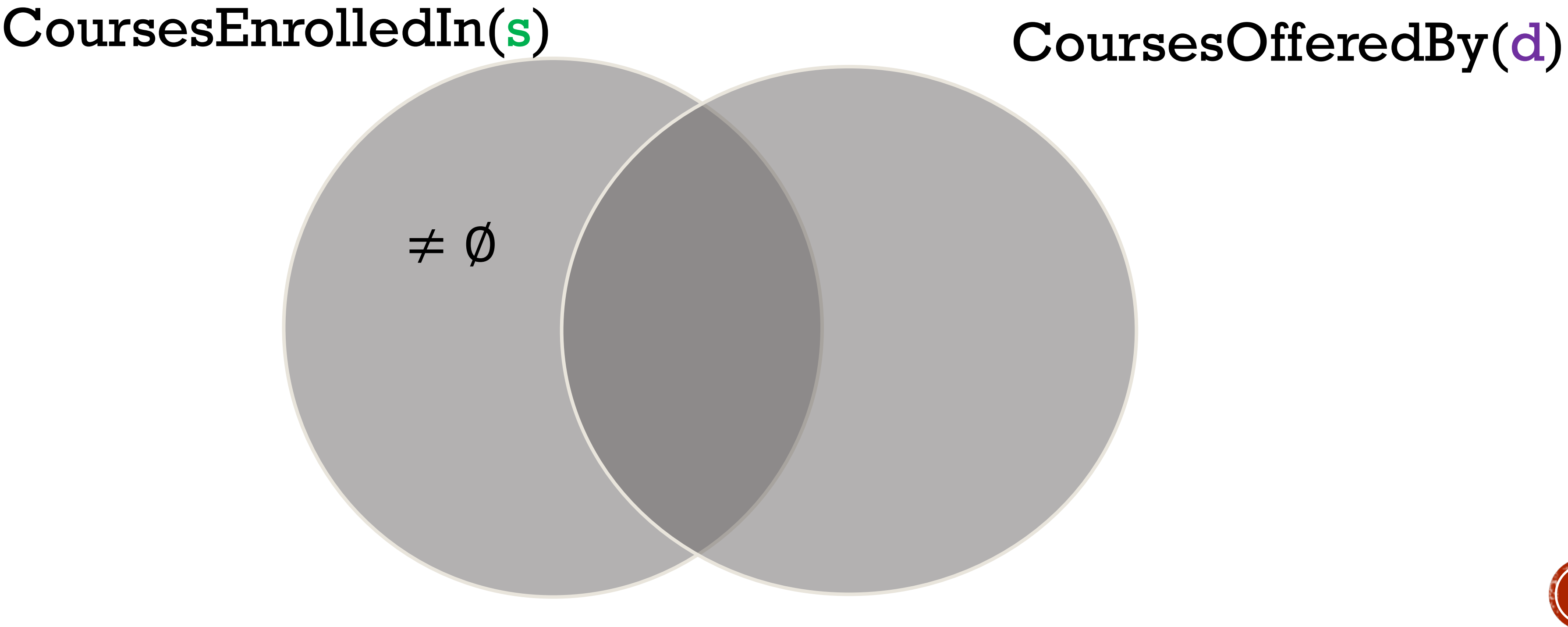
CoursesOfferedBy(d)

$= \emptyset$

NOT ONLY FOR PAIRS

$A - B \neq \emptyset \qquad \exists x \, (x \in A - B)$

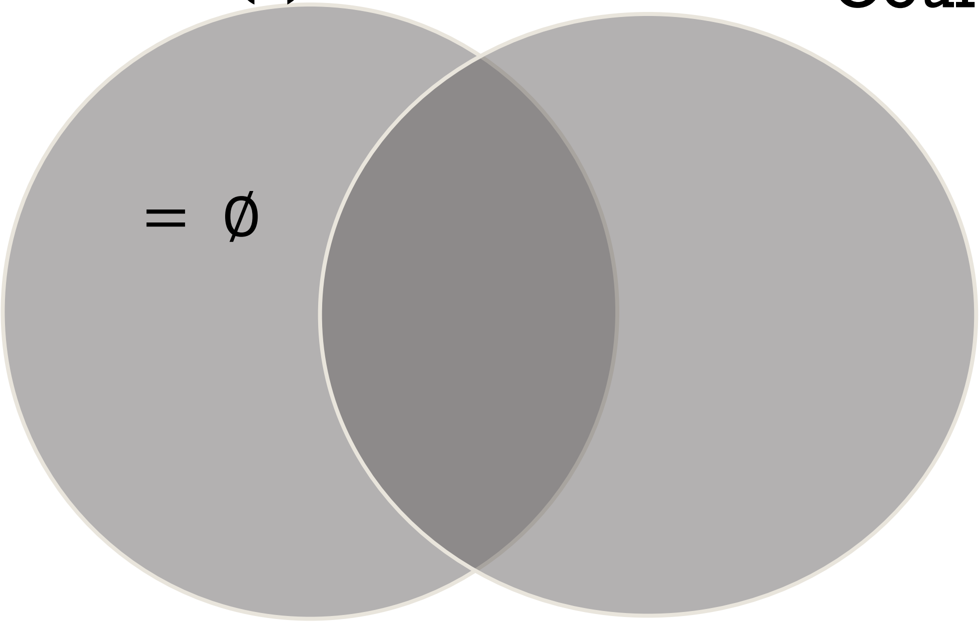
Find each (s,d) pair such that student s takes not only courses offered by department d



ONLY FOR PAIRS (SUBSET JOIN)

$A - B = \emptyset \qquad \neg \exists x \, (x \in A - B)$

Find each (s,d) pair such that student s takes only courses offered by department d



CoursesEnrolledIn(s)

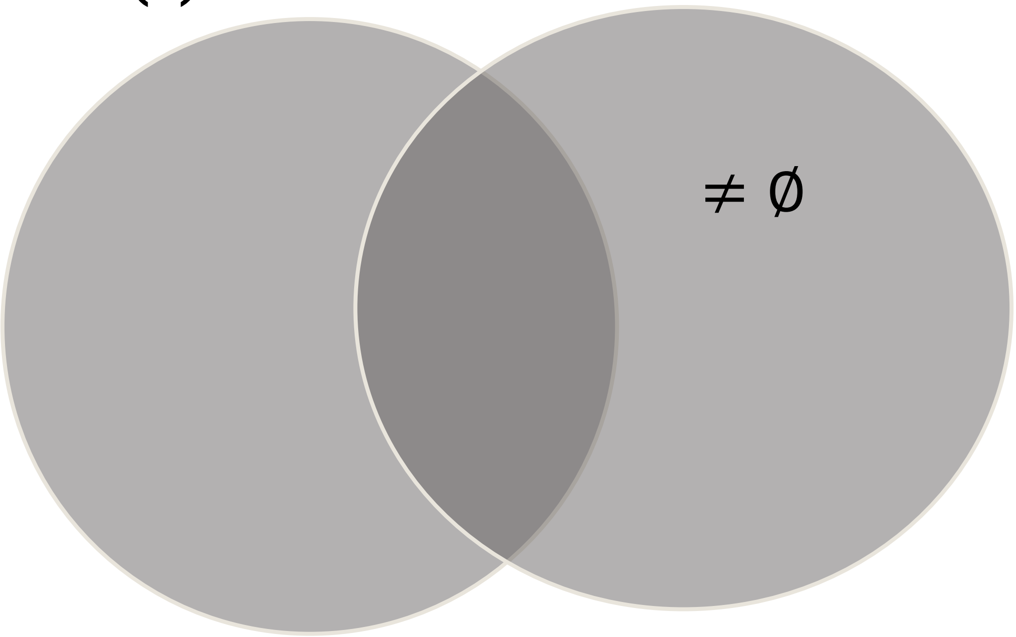
CoursesOfferedBy(d)

$= \emptyset$

NOT ALL FOR PAIRS

$B - A \neq \emptyset \qquad \exists\, x\, (x \in B - A)$

Find each (s,d) pair such that student s takes not all courses offered by department d



CoursesEnrolledIn(s)

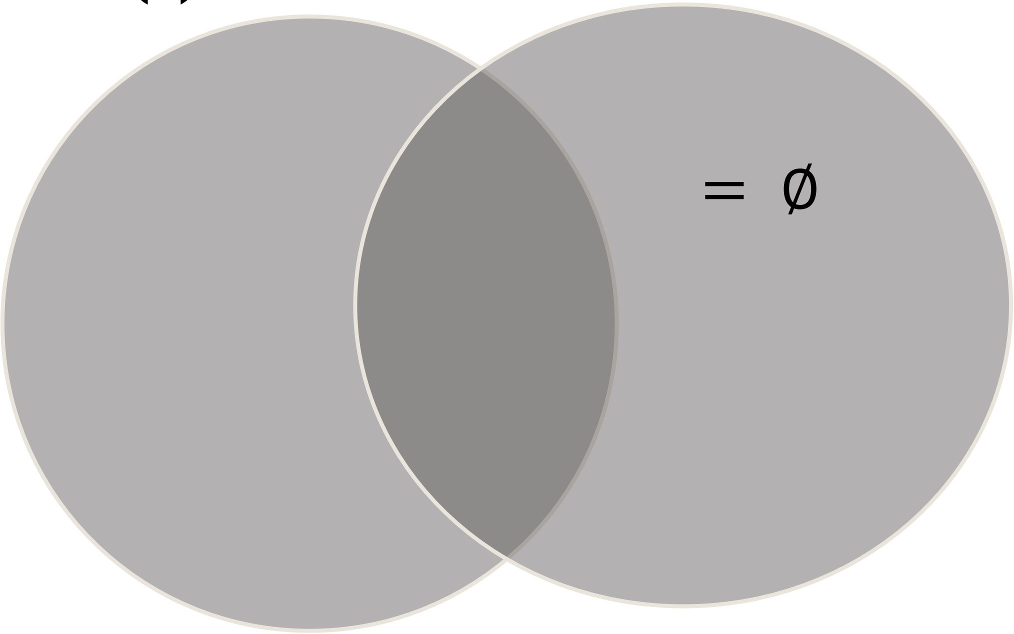
CoursesOfferedBy(d)

$\neq \emptyset$

ALL FOR PAIRS

$B - A = \emptyset$ $\neg \exists\, x\, (x \in B - A)$

Find each (s,d) pair such that student s takes all courses offered by department d



CoursesEnrolledIn(s)

CoursesOfferedBy(d)

$= \emptyset$