Query + Subquery Solution

- Find the price Joe charges for Bud.
- Find the bars that serve Miller at that price.

Sells(bar, beer, price)

```
SELECT bar
FROM Sells
WHERE beer = 'Miller' AND price
        = (SELECT price
           FROM Sells
           WHERE bar = 'Joe''s Bar'
               AND beer = 'Bud');
```

The price at which Joe sells Bud

What if price of Bud is NULL?

41

Query + Subquery Solution

```
SELECT bar
FROM Sells
WHERE beer = 'Miller' AND
      price = (SELECT price
                  FROM Sells
                  WHERE beer = 'Bud');
```

What if subquery returns multiple values?

42

Recap: Conditions in WHERE Clause

- Boolean operators AND, OR, NOT.
- Comparisons =, <>, <, >, <=, >=.
- LIKE operator

- SQL includes a **between** comparison operator
- Example: Find the names of all instructors with salary between $90,000 and $100,000 (that is, ≥ $90,000 and ≤ $100,000)
  - **select** *name*
    from *instructor*
    where *salary* **between** 90000 **and** 100000

43

The Operator ANY

- *x* = ANY(<subquery>) is a boolean condition that is true iff *x* equals at least one tuple in the subquery result.
  - = could be any comparison operator.
- Example: *x* >= ANY(<subquery>) means *x* is not the uniquely smallest tuple produced by the subquery.
  - Note tuples must have one component only.

44

The Operator ALL

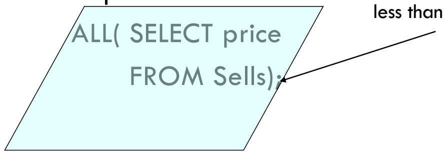
- *x* <> ALL(<subquery>) is true iff for every tuple *t* in the relation, *x* is not equal to *t*.
  - That is, *x* is not in the subquery result.
- <> can be any comparison operator.
- Example: *x* >= ALL(<subquery>) means there is no tuple larger than *x* in the subquery result.

45

Example: ALL

- From Sells(bar, beer, price), find the beer(s) sold for the highest price.

```
SELECT beer
FROM Sells
WHERE price >=
    ALL( SELECT price
         FROM Sells);
```



price from the outer Sells must not be less than any price.

46

The IN Operator

- <value> IN (<subquery>) is true if and only if the <value> is a member of the relation produced by the subquery.
  - Opposite: <value> NOT IN (<subquery>).
- IN-expressions can appear in WHERE clauses.
- WHERE col IN (value1, value2, …)

47

IN is Concise

- SELECT * FROM Cartoons
  WHERE LastName IN (‘Jetsons', 'Smurfs’, ’Flintstones’)

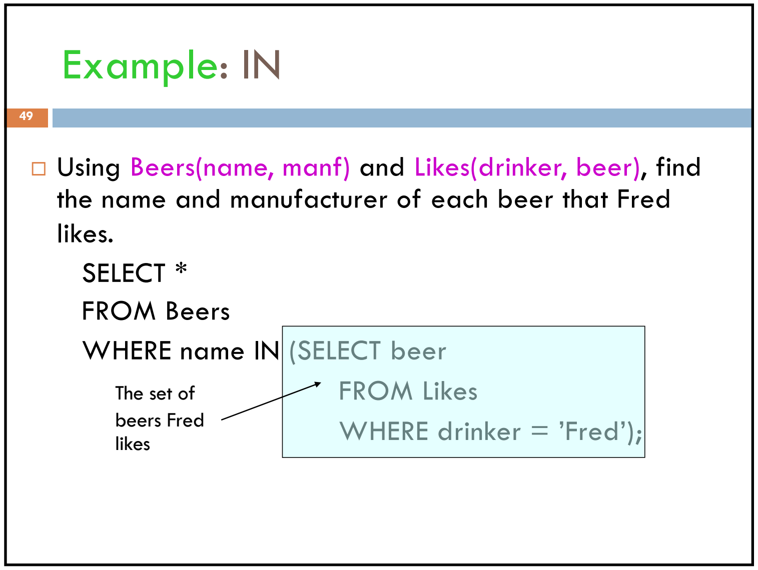
- SELECT * FROM Cartoons

WHERE LastName = ‘Jetsons’

OR LastName = ‘Smurfs’

OR LastName = ‘Flintstones’

48



Example: IN

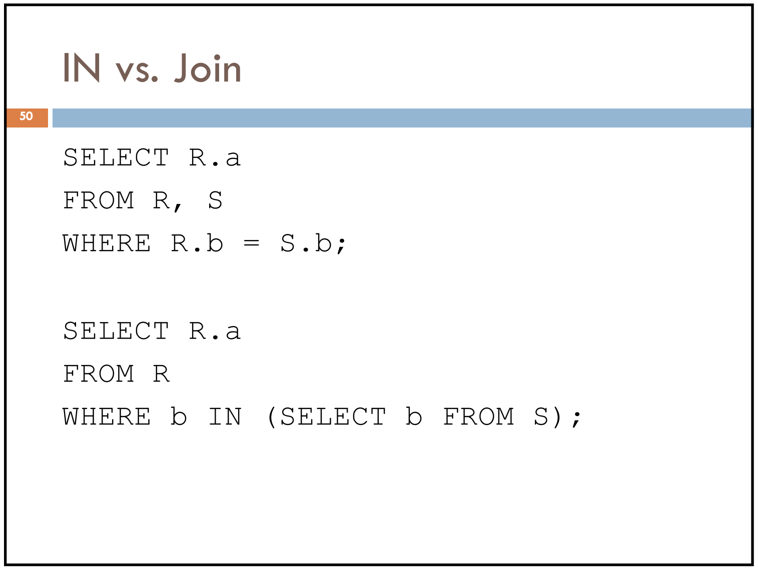
49

- Using Beers(name, manf) and Likes(drinker, beer), find the name and manufacturer of each beer that Fred likes.

```
SELECT *
FROM Beers
WHERE name IN (SELECT beer
                 FROM Likes
                 WHERE drinker = 'Fred');
```

The set of beers Fred likes

49



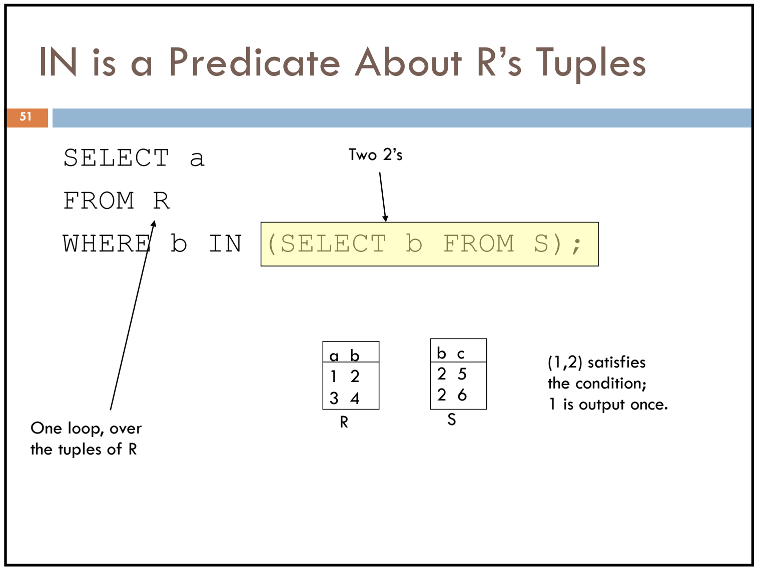
IN vs. Join

50

```
SELECT R.a
FROM R, S
WHERE R.b = S.b;

SELECT R.a
FROM R
WHERE b IN (SELECT b FROM S);
```

50



IN is a Predicate About R's Tuples

51

```
SELECT a
FROM R
WHERE b IN (SELECT b FROM S);
```

Two 2's

One loop, over the tuples of R

a	b
1	2
3	4

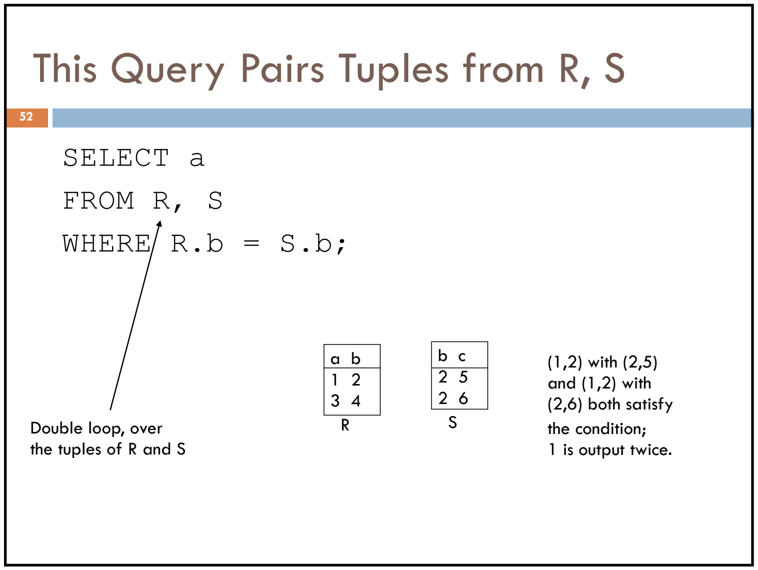
R

b	c
2	5
2	6

S

(1,2) satisfies the condition; 1 is output once.

51



This Query Pairs Tuples from R, S

52

```
SELECT a
FROM R, S
WHERE R.b = S.b;
```

Double loop, over the tuples of R and S

a	b
1	2
3	4

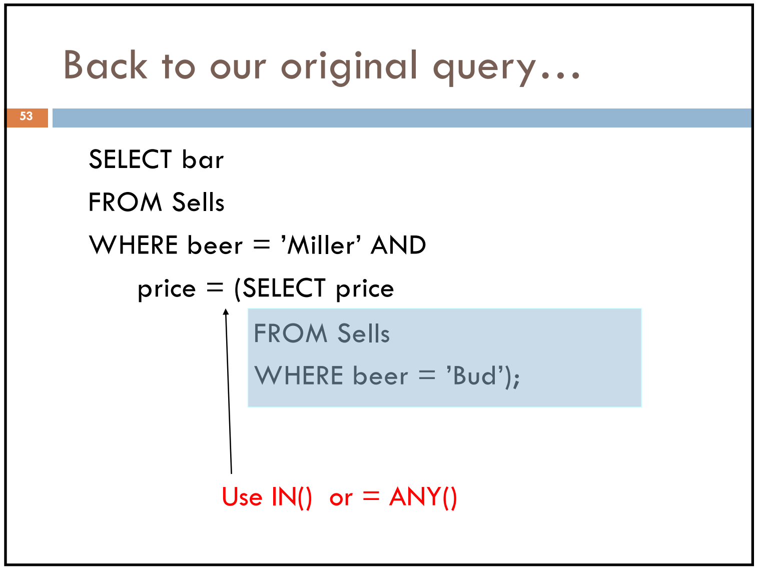
R

b	c
2	5
2	6

S

(1,2) with (2,5) and (1,2) with (2,6) both satisfy the condition; 1 is output twice.

52

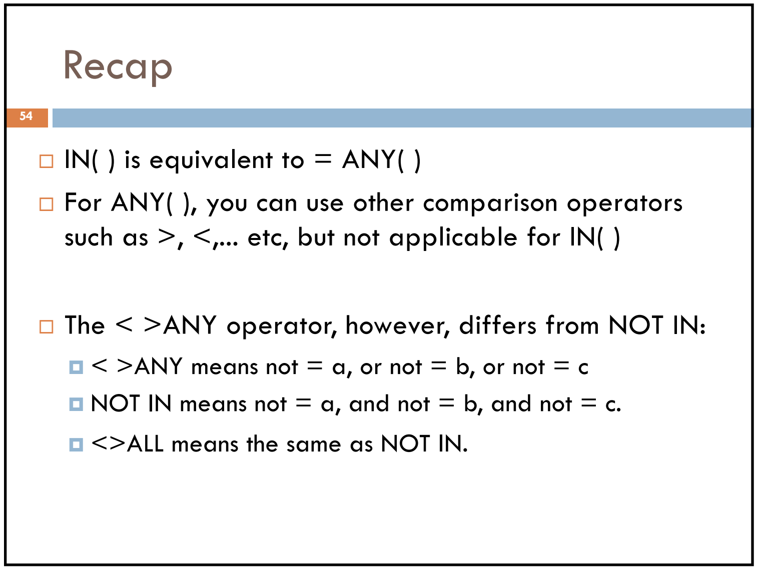


Back to our original query…

```
SELECT bar
FROM Sells
WHERE beer = ’Miller’ AND
      price = (SELECT price
               FROM Sells
               WHERE beer = ’Bud’);
```

Use IN() or = ANY()

53

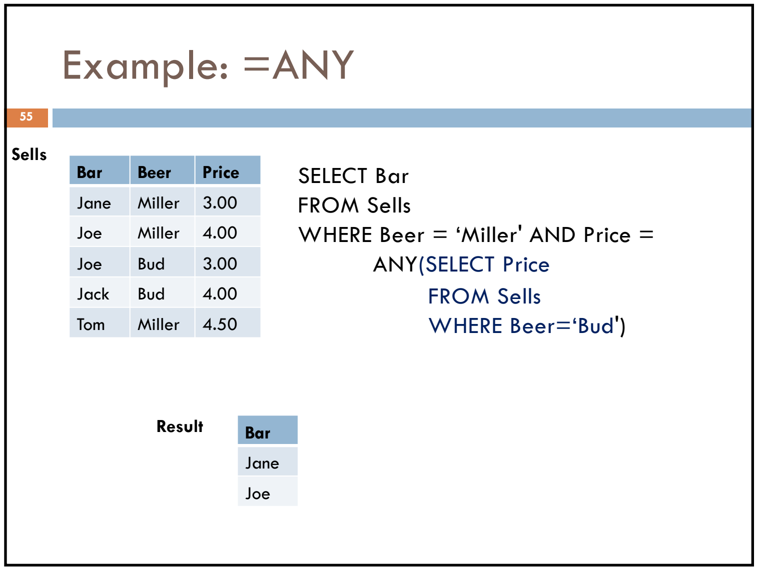


Recap

- IN( ) is equivalent to = ANY( )
- For ANY( ), you can use other comparison operators such as >, <,… etc, but not applicable for IN( )

- The < >ANY operator, however, differs from NOT IN:
  - < >ANY means not = a, or not = b, or not = c
  - NOT IN means not = a, and not = b, and not = c.
  - <>ALL means the same as NOT IN.

54



Example: =ANY

Sells

Bar	Beer	Price
Jane	Miller	3.00
Joe	Miller	4.00
Joe	Bud	3.00
Jack	Bud	4.00
Tom	Miller	4.50

```
SELECT Bar
FROM Sells
WHERE Beer = ‘Miller' AND Price =
        ANY(SELECT Price
             FROM Sells
             WHERE Beer=‘Bud')
```

Result

Bar
Jane
Joe

55