Example: =ANY

1

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

1

The Exists Operator

2

- EXISTS(<subquery>) is true if and only if the subquery result is not empty.
- Example: From Beers(name, manf) , find those beers that are the unique (only) beer made by their manufacturer.

Credit: Renee J. Miller

2

Example: EXISTS

3

```
SELECT name
  FROM Beers b1
  WHERE NOT EXISTS (
      SELECT *
      FROM Beers
      WHERE manf = b1.manf AND
            name <> b1.name);
```

Notice scope rule: manf refers to closest nested FROM with a relation having that attribute. (Some DBMS consider this ambiguous.)

Set of beers with the same manf as b1, but not the same beer

Notice the SQL "not equals" operator

3

Union, Intersection, and Difference

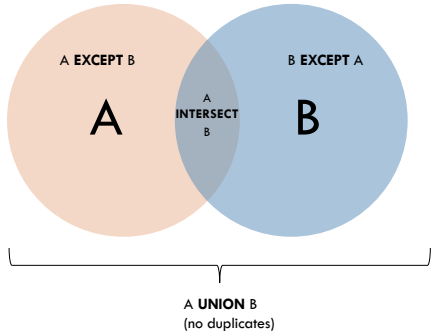
4

- Union, intersection, and difference of relations are expressed by the following forms, each involving subqueries:
  - (<subquery>) UNION (<subquery>)
  - (<subquery>) INTERSECT (<subquery>)
  - (<subquery>) EXCEPT (<subquery>)

4

Visually

5



5

Example: Intersection

6

- Using Likes(drinker, beer), Sells(bar, beer, price), and Frequents(drinker, bar), find the drinkers and beers such that:
  - The drinker likes the beer, and
  - The drinker frequents at least one bar that sells the beer.

6

Solution

7

subquery is really a stored table.

The drinker frequents a bar that sells the beer.

```
(SELECT * FROM Likes)
   INTERSECT
(SELECT drinker, beer
 FROM Sells, Frequents
 WHERE Frequents.bar = Sells.bar
);
```

7

Bag Semantics

8

- A *bag* (or *multiset*) is like a set, but an element may appear more than once.
- Example: {1,2,1,3} is a bag.
- Example: {1,2,3} is also a bag that happens to be a set.

8

Bag (Multiset) Semantics

9

- SQL primarily uses bag semantics
- The SELECT-FROM-WHERE statement uses bag semantics
  - originally for efficiency reasons
- The default for union, intersection, and difference is set semantics.
  - That is, duplicates are eliminated as the operation is applied.

9

Motivation: Efficiency

10

- When doing projection, it is easier to avoid eliminating duplicates.
  - Just work tuple-at-a-time.
- For intersection or difference, it is most efficient to sort the relations first.
  - At that point you may as well eliminate the duplicates anyway.

10

Controlling Duplicate Elimination

11

- Force the result to be a set by SELECT DISTINCT . . .
- Force the result to be a bag (i.e., don't eliminate duplicates) by ALL, as in
  . . . UNION ALL . . .

11

Example: DISTINCT

12

- From Sells(bar, beer, price), find all the different prices charged for beers:

```
SELECT DISTINCT price
FROM Sells;
```

Notice that without DISTINCT, each price would be listed as many times as there were bar/beer pairs at that price.

12

Example: ALL

13

- Using relations Frequents(drinker, bar) and Likes(drinker, beer):
- Lists drinkers who frequent more bars than they like beers, and do so as many times as the difference of those counts.

```
(SELECT drinker FROM Frequents)
   EXCEPT ALL
(SELECT drinker FROM Likes);
```

13

Ordering the Display of Tuples

14

- List in alphabetic order the names of all instructors

  select *name*
  from *instructor*
  order by *name*

- We may specify **desc** for descending order or **asc** for ascending order, for each attribute; ascending order is the default.
  - Example: **order by** *name* **desc**

Credit: Silberchatz, Korth & Sudarshan

14

Humour

15

SQL query walks into a bar, and approaches two tables and asks, can I join you?

15