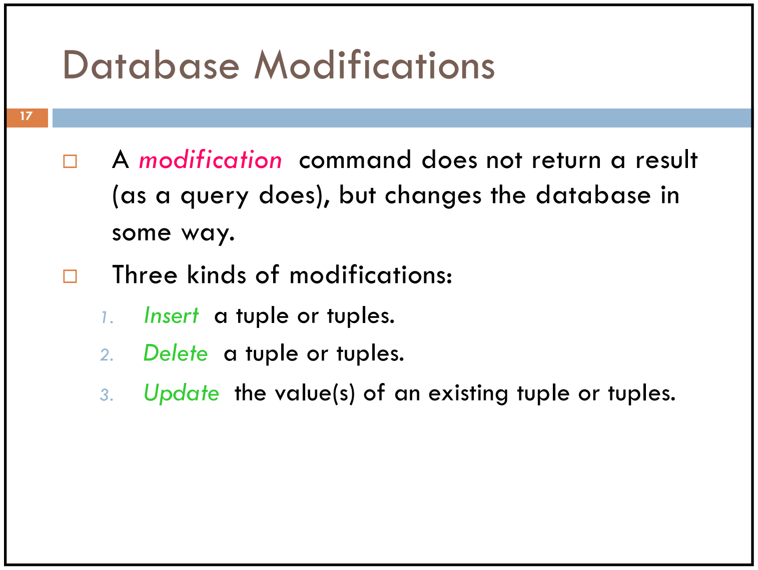
DATABASE MODIFICATIONS

16

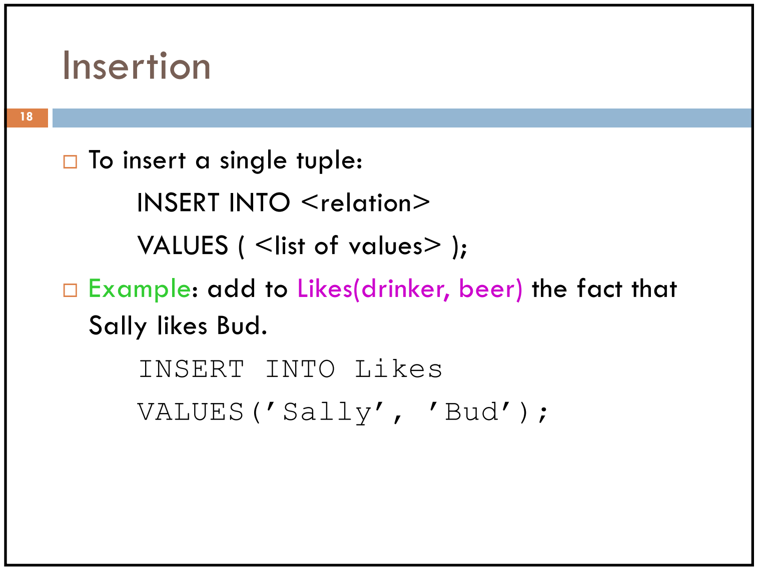


Database Modifications

17

- A *modification* command does not return a result (as a query does), but changes the database in some way.
- Three kinds of modifications:
  1. *Insert* a tuple or tuples.
  2. *Delete* a tuple or tuples.
  3. *Update* the value(s) of an existing tuple or tuples.

17



Insertion

18

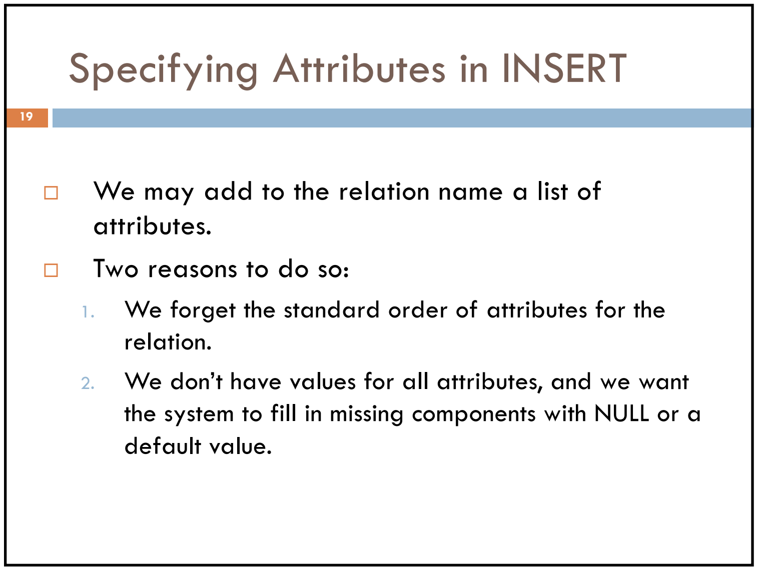
- To insert a single tuple:

  INSERT INTO <relation>

  VALUES (<list of values>);
- Example: add to Likes(drinker, beer) the fact that Sally likes Bud.

```
INSERT INTO Likes
VALUES('Sally', 'Bud');
```

18

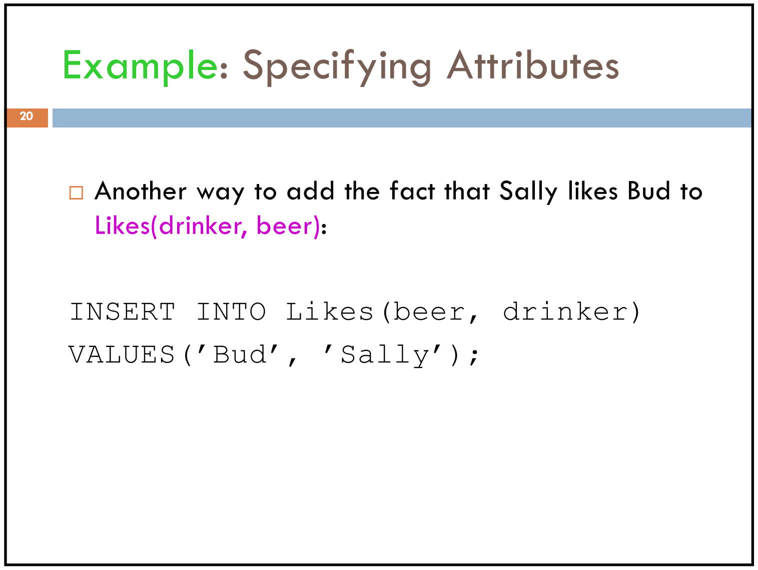


Specifying Attributes in INSERT

19

- We may add to the relation name a list of attributes.
- Two reasons to do so:
  1. We forget the standard order of attributes for the relation.
  2. We don't have values for all attributes, and we want the system to fill in missing components with NULL or a default value.

19

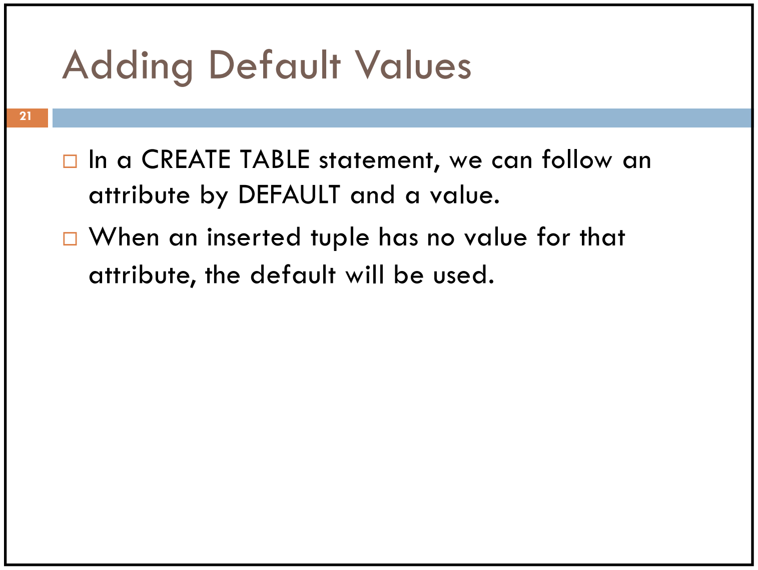


Example: Specifying Attributes

- Another way to add the fact that Sally likes Bud to Likes(drinker, beer):

```
INSERT INTO Likes(beer, drinker)
VALUES('Bud', 'Sally');
```

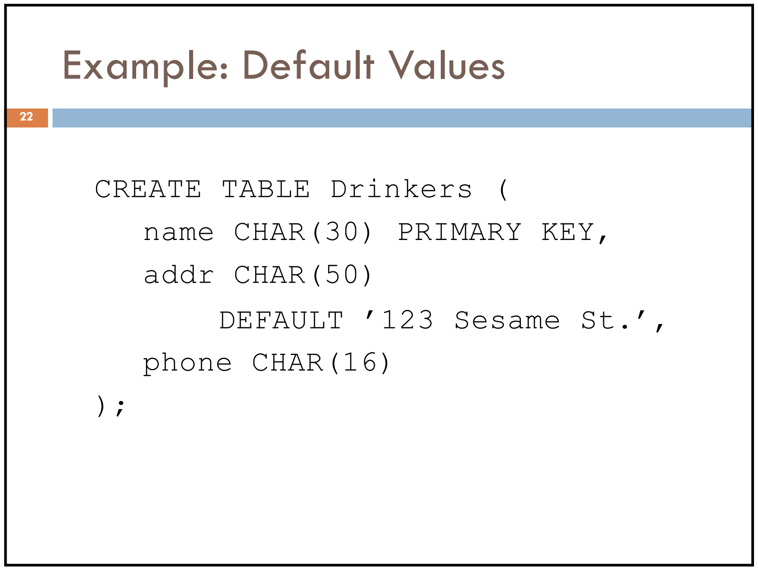
20



Adding Default Values

- In a CREATE TABLE statement, we can follow an attribute by DEFAULT and a value.
- When an inserted tuple has no value for that attribute, the default will be used.

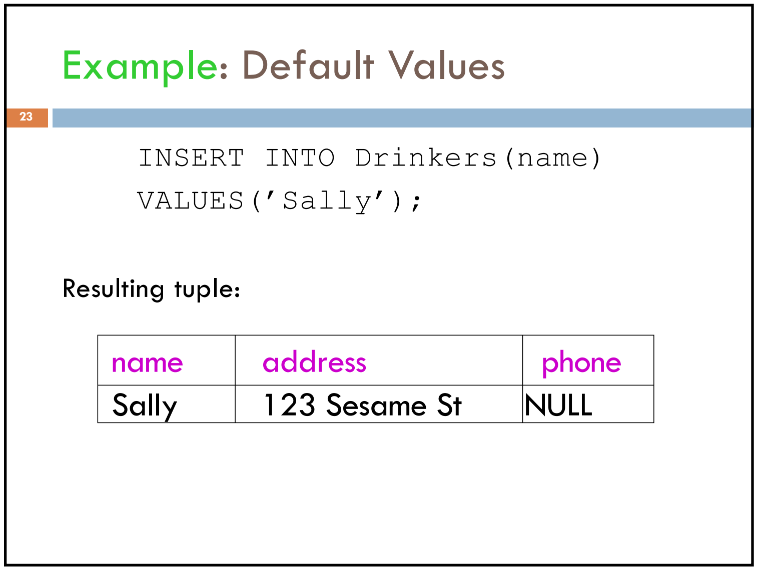
21



Example: Default Values

```
CREATE TABLE Drinkers (
   name CHAR(30) PRIMARY KEY,
   addr CHAR(50)
        DEFAULT '123 Sesame St.',
   phone CHAR(16)
);
```

22



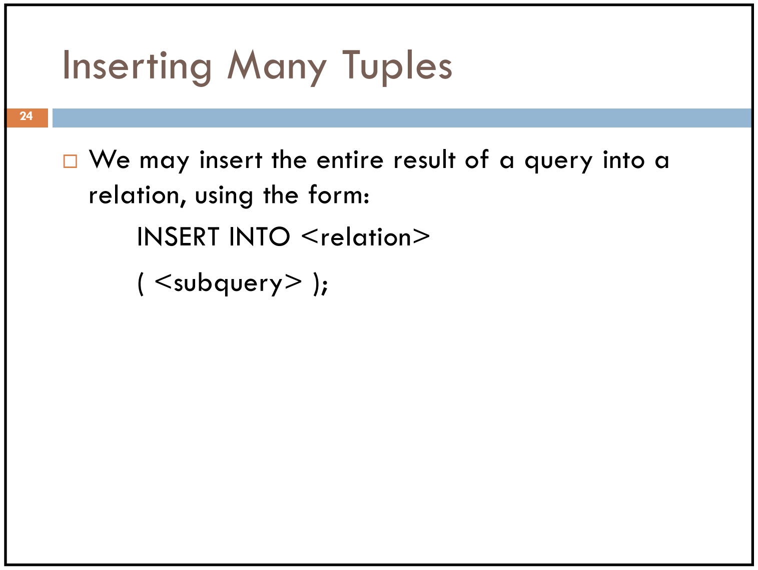
Example: Default Values

```
INSERT INTO Drinkers(name)
VALUES('Sally');
```

Resulting tuple:

name	address	phone
Sally	123 Sesame St	NULL

23



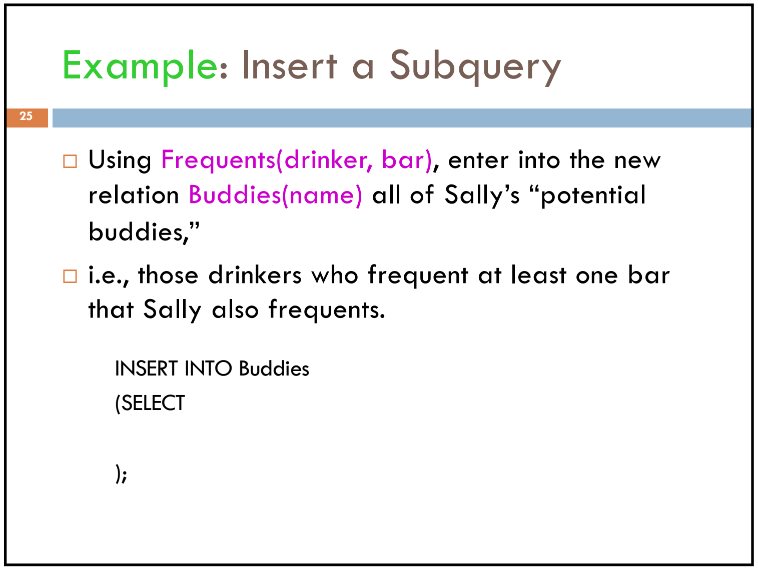
Inserting Many Tuples

24

- □ We may insert the entire result of a query into a relation, using the form:

```
INSERT INTO <relation>
( <subquery> );
```

24



Example: Insert a Subquery

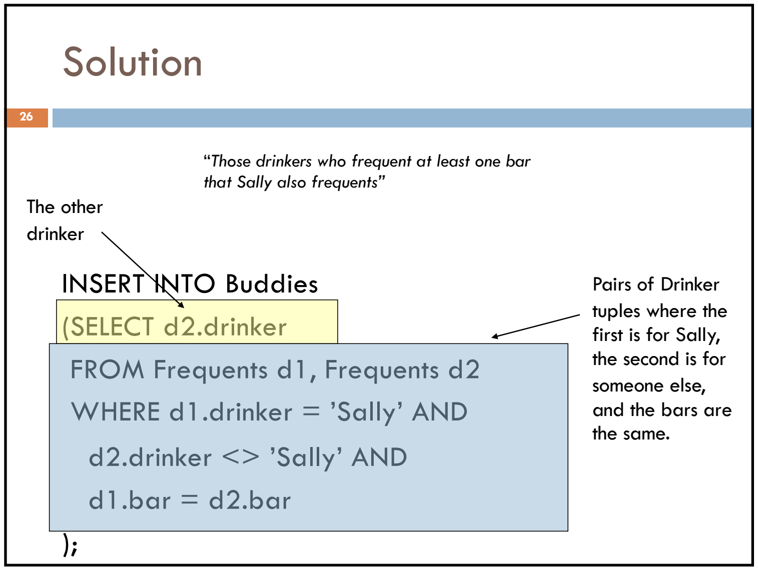
25

- □ Using Frequents(drinker, bar), enter into the new relation Buddies(name) all of Sally's "potential buddies,"
- □ i.e., those drinkers who frequent at least one bar that Sally also frequents.

```
INSERT INTO Buddies
(SELECT


);
```

25



Solution

26

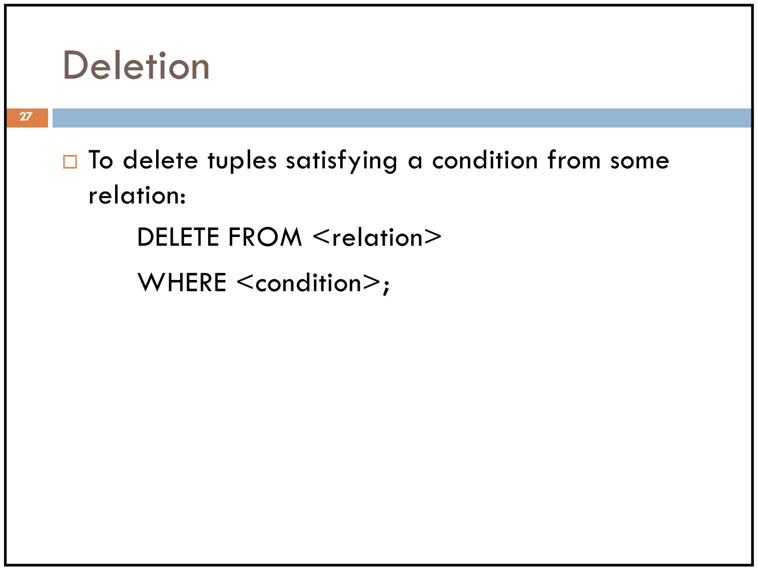
"Those drinkers who frequent at least one bar that Sally also frequents"

The other drinker

```
INSERT INTO Buddies
(SELECT d2.drinker
 FROM Frequents d1, Frequents d2
 WHERE d1.drinker = 'Sally' AND
   d2.drinker <> 'Sally' AND
   d1.bar = d2.bar
);
```

Pairs of Drinker tuples where the first is for Sally, the second is for someone else, and the bars are the same.

26



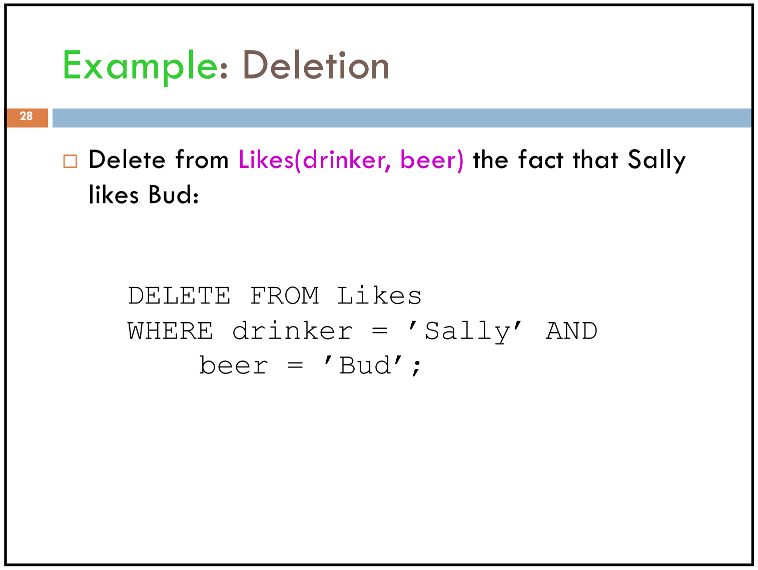
Deletion

27

- □ To delete tuples satisfying a condition from some relation:

```
DELETE FROM <relation>
WHERE <condition>;
```

27



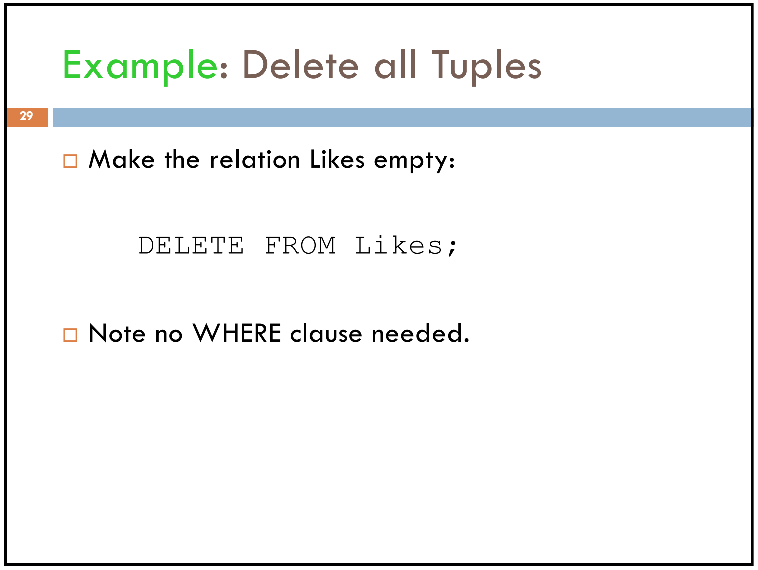
Example: Deletion

28

- □ Delete from Likes(drinker, beer) the fact that Sally likes Bud:

```
DELETE FROM Likes
WHERE drinker = 'Sally' AND
    beer = 'Bud';
```

28



Example: Delete all Tuples

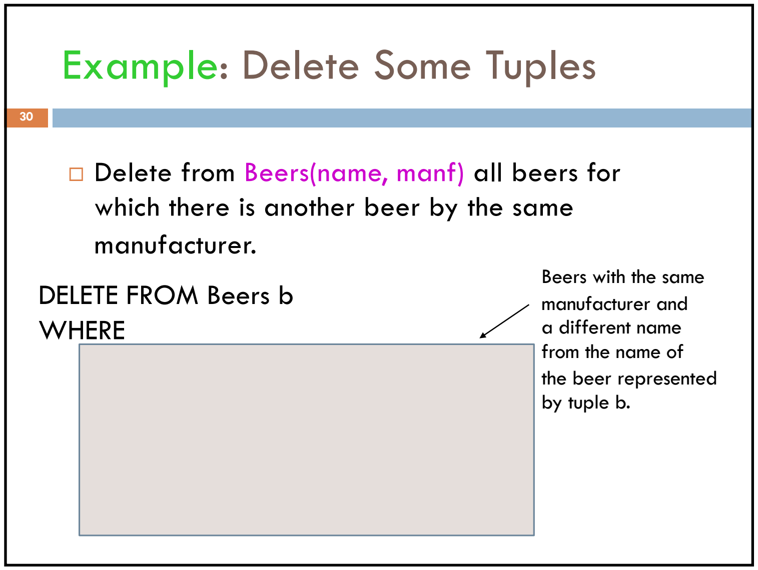
29

- □ Make the relation Likes empty:

```
DELETE FROM Likes;
```

- □ Note no WHERE clause needed.

29



Example: Delete Some Tuples

30

- □ Delete from Beers(name, manf) all beers for which there is another beer by the same manufacturer.

DELETE FROM Beers b
WHERE

Beers with the same manufacturer and a different name from the name of the beer represented by tuple b.

30