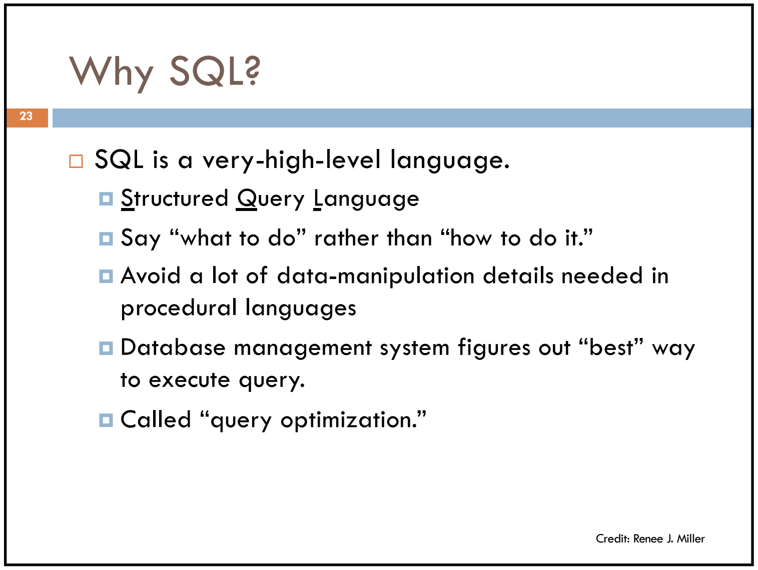
INTRODUCTION TO SQL

22

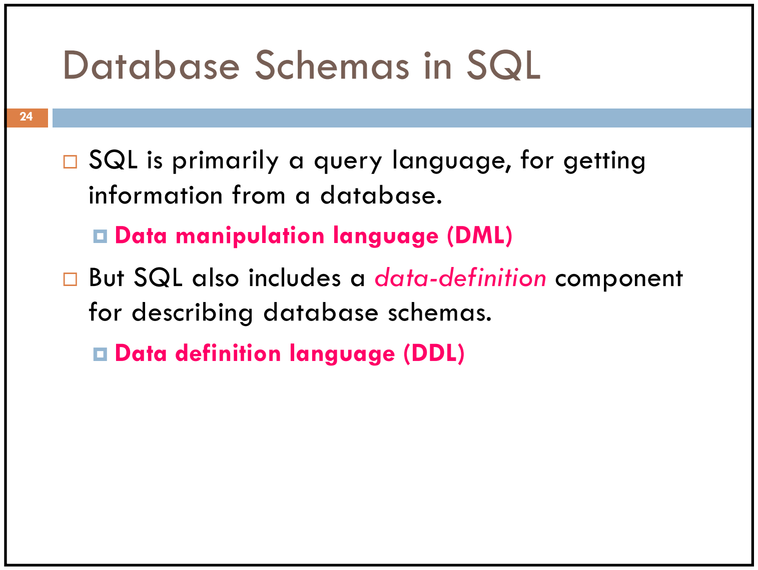


Why SQL?

- SQL is a very-high-level language.
  - Structured Query Language
  - Say "what to do" rather than "how to do it."
  - Avoid a lot of data-manipulation details needed in procedural languages
  - Database management system figures out "best" way to execute query.
  - Called "query optimization."

Credit: Renee J. Miller

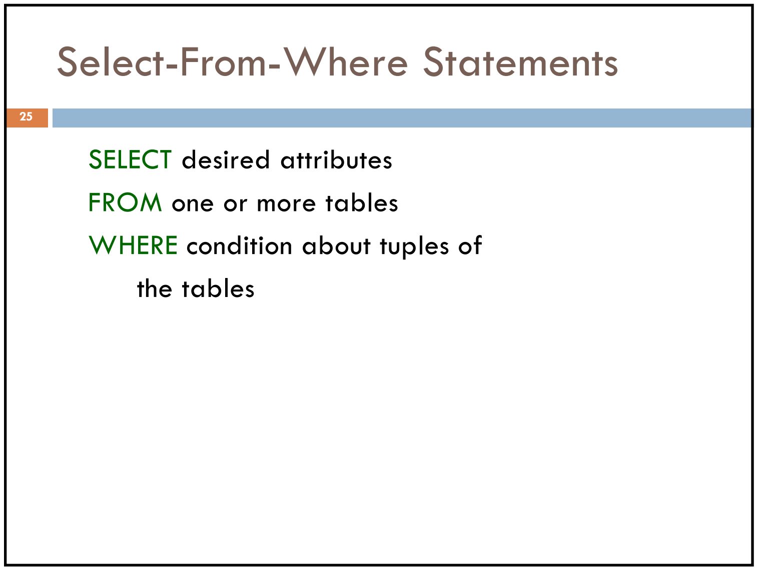
23



Database Schemas in SQL

- SQL is primarily a query language, for getting information from a database.
  - **Data manipulation language (DML)**
- But SQL also includes a *data-definition* component for describing database schemas.
  - **Data definition language (DDL)**

24



Select-From-Where Statements

SELECT desired attributes
FROM one or more tables
WHERE condition about tuples of
    the tables

25

Our Running Example

- Our SQL queries will be based on the following database schema.
  - Underline indicates key attributes.

Beers(name, manf)
Bars(name, addr, license)
Drinkers(name, addr, phone)
Likes(drinker, beer)
Sells(bar, beer, price)
Frequents(drinker, bar)

26

Example

- Using Beers(name, manf), what beers are made by Anheuser-Busch?

```
SELECT name
FROM Beers
WHERE manf = 'Anheuser-Busch';
```

27

Result of Query

name
Bud
Bud Lite
Michelob
. . .

The answer is a relation with a single attribute, name, and tuples with the name of each beer by Anheuser-Busch, such as Bud.

28

Meaning of Single-Relation Query

- Begin with the relation in the FROM clause.
- Apply the selection indicated by the WHERE clause.
- Apply the extended projection indicated by the SELECT clause.

29

Operational Semantics - General

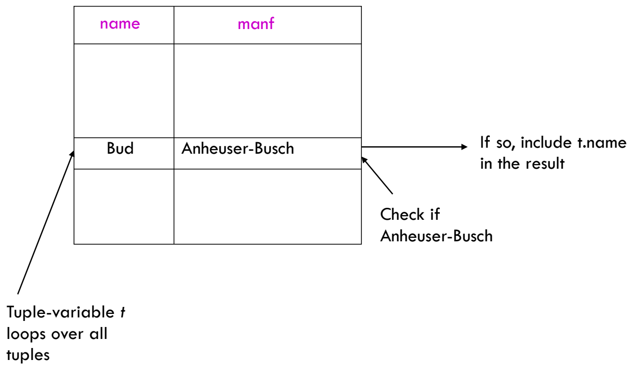
30

- □ Think of a *tuple variable* visiting each tuple of the relation mentioned in FROM.
- □ Check if the tuple assigned to the tuple variable satisfies the WHERE clause.
- □ If so, compute the attributes or expressions of the SELECT clause using the components of this tuple.

30

Operational Semantics

31



name	manf
Bud	Anheuser-Busch

31

Example

32

- □ What beers are made by Anheuser-Busch?

```
    SELECT name
    FROM Beers
    WHERE manf = ’Anheuser-Busch’;
OR:
    SELECT t.name
    FROM Beers t
    WHERE t.manf =’Anheuser-Busch’;
```

Note: these are identical queries.

32

* In SELECT clauses

33

- □ When there is one relation in the FROM clause, * in the SELECT clause stands for “all attributes of this relation.”
- □ Example: Using Beers(name, manf):

```
    SELECT *
    FROM Beers
    WHERE manf = ’Anheuser-Busch’;
```

33

Result of Query:

34

name	manf
Bud	Anheuser-Busch
Bud Lite	Anheuser-Busch
Michelob	Anheuser-Busch
. . .	. . .

Now, the result has each of the attributes of Beers.

34

Renaming Attributes

35

- □ If you want the result to have different attribute names, use “AS <new name>” to rename an attribute.
- □ Example: Using Beers(name, manf):

```
SELECT name AS beer, manf
FROM Beers
WHERE manf = 'Anheuser-Busch'
```

35

Result of Query:

36

beer	manf
Bud	Anheuser-Busch
Bud Lite	Anheuser-Busch
Michelob	Anheuser-Busch
. . .	. . .

36

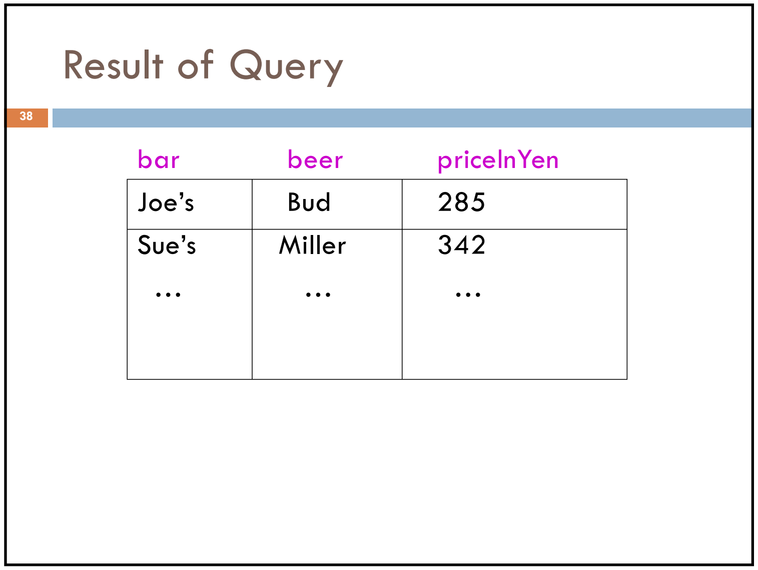
Expressions in SELECT Clauses

37

- □ Any valid expression can appear as an element of a SELECT clause.
- □ Example: Using Sells(bar, beer, price):

```
SELECT bar, beer,
       price*95 AS priceInYen
FROM Sells;
```

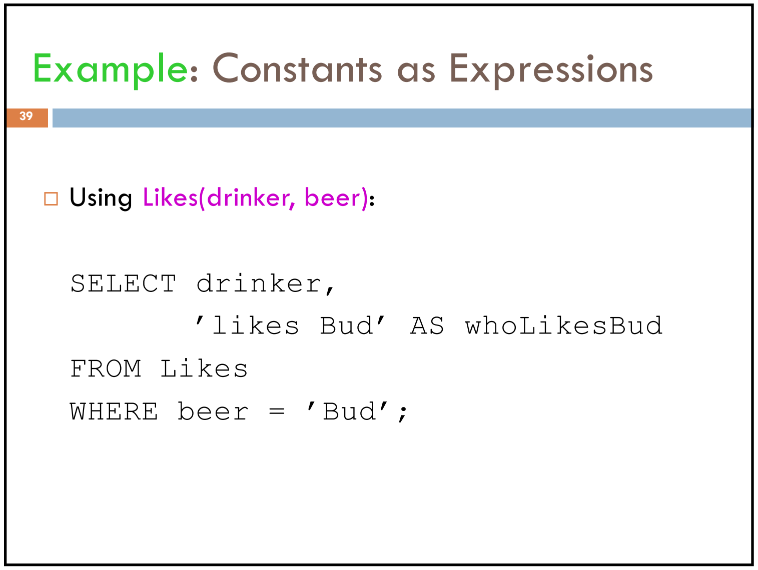
37



Result of Query

bar	beer	priceInYen
Joe’s	Bud	285
Sue’s	Miller	342
…	…	…

38

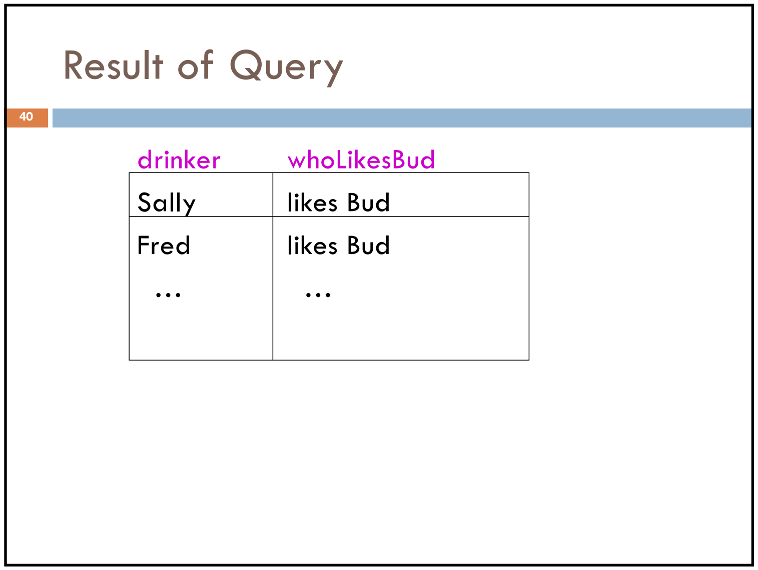


Example: Constants as Expressions

- □ Using Likes(drinker, beer):

```
SELECT drinker,
       ’likes Bud’ AS whoLikesBud
FROM Likes
WHERE beer = ’Bud’;
```

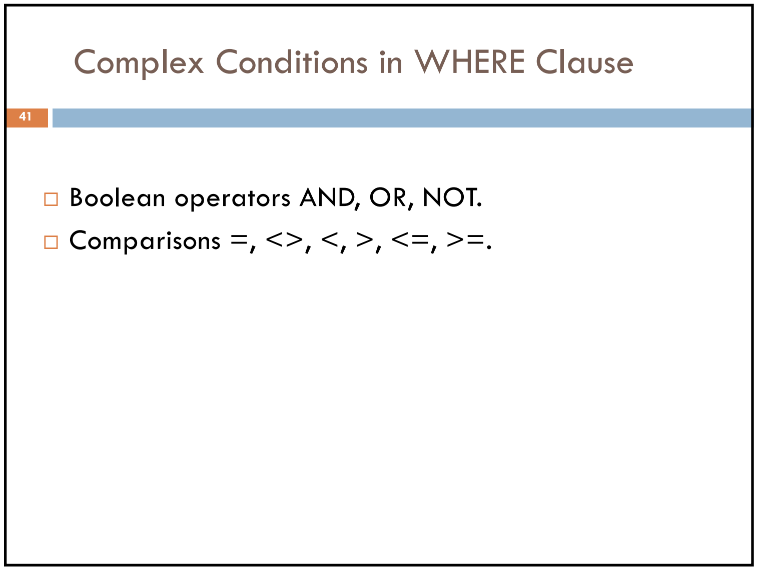
39



Result of Query

drinker	whoLikesBud
Sally	likes Bud
Fred	likes Bud
…	…

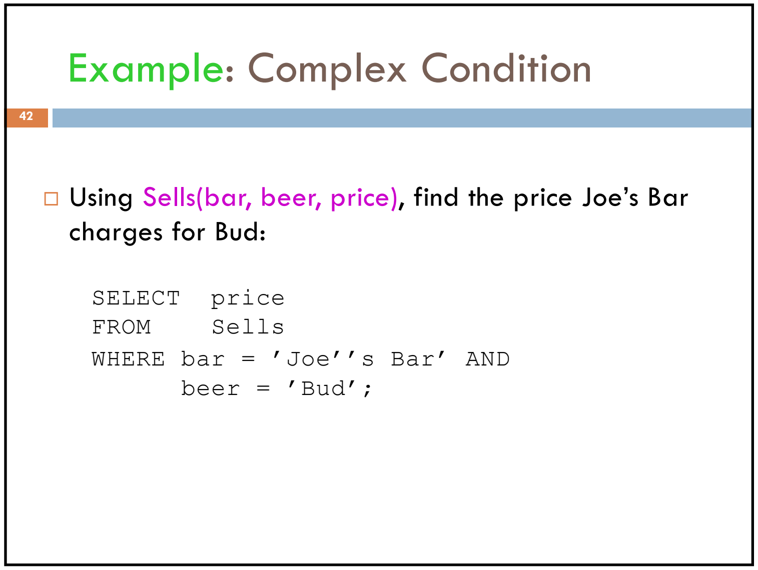
40



Complex Conditions in WHERE Clause

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

41



Example: Complex Condition

- Using Sells(bar, beer, price), find the price Joe’s Bar charges for Bud:

```
SELECT  price
FROM    Sells
WHERE bar = ’Joe’’s Bar’ AND
      beer = ’Bud’;
```