CSC 430/530 – DBMS/DT

Lecture 8: Basics of SQL and ER – Modeling
Exercise

261

Relational Query Languages

- Two sublanguages:
  - DDL – Data Definition Language
    - Define and modify schema
  - DML – Data Manipulation Language
    - Queries can be written intuitively.
- DBMS is responsible for efficient evaluation.
  - The key: precise semantics for relational queries.
  - Optimizer can re-order operations, without affecting query answer.
  - Choices driven by "cost model"

262

The SQL Query Language

- The most widely used relational query language.
- Standardized
  (although most systems add their own "special sauce" -- including PostgreSQL)
- We will study SQL92 -- a basic subset

263

Data Definition Language

The SQL data-definition language (DDL) allows the specification of information about relations, including:

1. The domain of values associated with each attribute.
2. The schema for each relation.
3. Integrity constraints
4. And as we will see later, also other information such as
   - The set of indices to be maintained for each relations.
   - Security and authorization information for each relation.
   - The physical storage structure of each relation on disk.

264

Domain Types in SQL

- **`CHAR(N)`**: Fixed length character string, with user-specified length *n*.
- **`VARCHAR(N)`**: Variable length character strings, with user-specified maximum length *n*.
- **`int`**: Integer (a finite subset of the integers that is machine-dependent).
- **`smallint`**: Small integer (a machine-dependent subset of the integer domain type).

265

Domain Types in SQL

- **`NUMERIC(p,d)`**: Fixed point number, with user-specified precision of *p* digits, with *d* digits to the right of decimal point. (ex., **`NUMERIC`**`(3,1)`, allows 44.5 to be stored exactly, but not 444.5 or 0.32)
- **`REAL, DOUBLE PRECISION`**: Floating point and double-precision floating point numbers, with machine-dependent precision.
- **`FLOAT(n)`**: Floating point number, with user-specified precision of at least *n* digits.

266

Create Table Construct

- A SQL relation is defined using the **create table** command:

```
CREATE TABLE r (A1 D1, A2 D2, ..., An Dn,
                (integrity-constraint1),
                ...,
                (integrity-constraintk))
```

  - *r* is the name of the relation
  - each A_i is an attribute name in the schema of relation *r*
  - D_i is the data type of values in the domain of attribute A_i

- Example:

```
CREATE TABLE instructor (
    ID            char(5),
    name          varchar(20),
    dept_name     varchar(20),
    salary        numeric(8,2))
```

267

Integrity Constraints in Create Table

- **not null**
- **primary key** (A_1, ..., A_n)
- **foreign key** (A_m, ..., A_n) **references** *r*

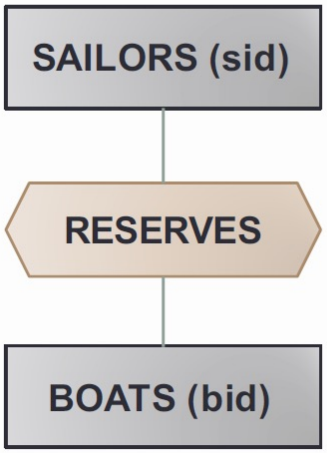
Example:

```
CREATE TABLE INSTRUCTOR (
        ID            CHAR(5),
        NAME          VARCHAR(20) NOT NULL,
        DEPT_NAME     VARCHAR(20),
        SALARY        NUMERIC(8,2),
        PRIMARY KEY (ID),
        FOREIGN KEY (DEPT_NAME) REFERENCES DEPARTMENT);
```

NOTE: primary key declaration on an attribute automatically ensures **not null**

268

Example Database



Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Reserves

sid	bid	day
1	102	9/12
2	102	9/13

Boats

bid	bname	color
101	Nina	red
102	Pinta	blue
103	Santa Maria	red

269

The SQL DDL

```
• CREATE TABLE SAILORS (
        SID     INT,
        SNAME   VARCHAR(20) NOT NULL,
        RATING  INT,
        AGE     REAL,
       PRIMARY KEY (SID))
• CREATE TABLE BOATS (
        BID     INT,
        BNAME   VARCHAR(20) NOT NULL,
        COLOR   VARCHAR(10),
       PRIMARY KEY (BID))
• CREATE TABLE RESERVES (
        SID     INT,
        BID     INT,
        DAY     DATE,
        PRIMARY KEY (SID, BID)
        FOREIGN KEY SID REFERENCES SAILORS,
        FOREIGN KEY BID REFERENCES BOATS)
```

270

Updates to tables

- **Insert:** Insert a tuple into the *instructor* relation
  - `INSERT INTO INSTRUCTOR VALUES ('10211', 'Smith', 'Biology', 66000);`
- **Delete:** Remove all tuples from the *student* relation
  - `DELETE FROM STUDENT`
- **Drop Table**
  - `DROP TABLE R`
- **Alter**
  - `ALTER TABLE R ADD A D`
    - where *A* is the name of the attribute to be added to relation *r* and *D* is the domain of *A.*
    - All exiting tuples in the relation are assigned *null* as the value for the new attribute.
  - `ALTER TABLE R DROP A`
    - where *A* is the name of an attribute of relation *r*
    - Dropping of attributes not supported by many databases.

271

DML: Basic Query Structure

- A typical SQL query has the form:

$$\textbf{SELECT}\ A_1,\ A_2,\ \ldots,\ A_n$$
$$\textbf{FROM}\ r_1,\ r_2,\ \ldots,\ r_m$$
$$\textbf{WHERE}\ P$$

  - A_i represents an attribute
  - r_i represents a relation
  - P is a predicate (or condition).

- The result of an SQL query is a relation.

272

The SQL DML

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

- Find all 18-year-old sailors:

```
SELECT *
FROM   Sailors S
WHERE  S.age=18
```

- To find just names and ratings, replace the first line:

```
SELECT S.sname, S.rating
FROM   Sailors S
WHERE  S.age=18
```

273

Try your self:

- List the names of sailors who reserved the boat with ID 102

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Reserves

sid	bid	day
1	102	9/12
2	102	9/13

Boats

bid	bname	color
101	Nina	red
102	Pinta	blue
103	Santa Maria	red

274

Querying Multiple Relations

Sailors

sid	sname	rating	age
1	Fred	7	22
2	Jim	2	39
3	Nancy	8	27

Reserves

sid	bid	day
1	102	9/12
2	102	9/13

```
SELECT S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid=R.sid AND R.bid=102
```

275

Basic SQL Query

```
SELECT    [DISTINCT] <attribute list>
FROM      <table list>
WHERE     <condition>;
```

- *table list*: List of relation names
- *Attribute list* : List of attributes of tables in *table-list*
- *condition* : >, <, >=, <=, Comparisons combined using AND, OR and NOT.
- *DISTINCT* : optional keyword indicating that the answer should not contain duplicates.

276

EXERCISE

277

Exercise

- The Prescriptions-R-X chain of pharmacies was recently “ghosted” by their “loyal” DBA. Their recent “hire” needs help to verify existing conceptual diagram. He wants varied perspectives so that he could debug the design and has gathered information as follows:
  - Patients are identified by an SSN, and their names, addresses, and ages must be recorded.
  - Doctors are identified by an SSN. For each doctor, the name, specialty, and years of experience must be recorded.
  - Each pharmaceutical company is identified by name and has a phone number.
  - For each drug, the trade name and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.

278

Furthermore…

- Each pharmacy has a name, address, and phone number.
- Every patient has a primary physician.
- Every doctor has at least one patient.
- Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.
- Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.

279

Finally…

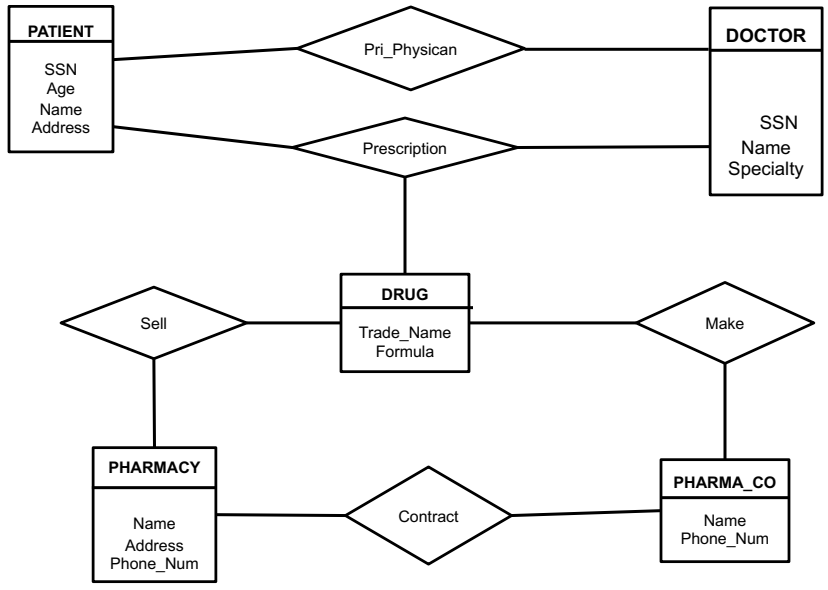
- Pharmaceutical companies have long-term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract.
- Pharmacies appoint a supervisor for each contract. There must always be a supervisor for each contract, but the contract supervisor can change over the lifetime of the contract.

280

Your Task

- As we are waiting for the Prescriptions-R-X DBA to arrive!!!
- Play the role of a consulting team to verify these requirements to spend no more than 15 mins to conceptualize your own little conceptual ER diagram

281



282