SQL Data Definition

Credit: Dr. Bruns

What is SQL?

- SQL = “Structured Query Language”
- Originally developed at IBM in the early 70’s
- Pronounced “sequel” or S-Q-L
- A special-purpose language that lets you express questions about your relational data in a precise way

Parts of SQL

SQL is not just for writing queries.

It has several parts, including:

- Data-definition language (DDL)
  - create/delete/modify schemas
  - define integrity constraints
  - define views
  - drop tables
- Data-manipulation language (DML)
  - define queries
  - modify tables (insert/delete/modify tuples)

Defining a relation schema

```
create table department (
  dept_name  varchar(20),
  building   varchar(15),
  budget     numeric(12,2),
  primary key (dept_name)
);
```

For each attribute we give the attribute name and type.

We also give the primary key.

Another ‘create table’ example

```
create table student (
   ID          varchar(20),
   name        varchar(20) not null,
   dept_name   varchar(20),
   tot_cred    numeric(3,0),
   primary key (ID),
   foreign key (dept_name) references department
);
```

The foreign key constraint says:

every dept_name value in the student table must equal the key of some row in the department table.

The not-null constraint says:

no name value in the student table can be null

Attribute types

Recall that attributes can only be simple values, like integers, and strings.

Here are some of the main types allowed by SQL:

`varchar(n)`

a variable-length string of length at most n

`int`

an integer – 'integer' is also okay

`numeric(p,d)`

a fixed-point number of p digits, with d digits to right of the decimal point. Example: 123.45 is numeric(5,2)

`float(n)`

a floating-point number with at least n digits precision

Yet another example

```
create table takes (
  ID          varchar(5),
  course_id   varchar(8),
  sec_id      varchar(8),
  semester    varchar(6),
  year        numeric(4,0),
  grade       varchar(2),
  primary key (ID,course_id,sec_id,semester,year),
  foreign key (course_id,sec_id, semester, year)
     references section on delete cascade,
  foreign key (ID)
     references student on delete cascade
);
```

- primary key has multiple attributes
- two foreign keys

What's wrong with this?

```
create table department {
  dept_name varchar(20) primary key;
  building varchar(15) not null;
  budget integer not null;
};
```

corrected version:

```
create table department (
  dept_name varchar(20) primary key,
  building varchar(15) not null,
  budget integer not null
);
```

Creating a table

patient_no	last_name	first_name	sex	date_of_birth	ward
454	Smith	John	M	14.08.78	6
223	Jones	Peter	M	07.12.85	8
597	Brown	Brenda	F	17.06.61	3
234	Jenkins	Alan	M	29.01.72	7
244	Wells	Chris	F	25.02.95	6

```
create table patient (
  patient_no integer primary key,
  last_name varchar(64) not null,
  first_name varchar(64) not null,
  sex varchar(1),
  date_of_birth varchar(8),
  ward integer
);

insert into patient values(454, "Smith",   "John",   "M", "14.08.78", 6);
insert into patient values(223, "Jones",   "Peter",  "M", "07.12.85", 8);
insert into patient values(597, "Brown",   "Brenda", "F", "17.06.61", 3);
insert into patient values(234, "Jenkins", "Alan",   "M", "29.01.72", 7);
insert into patient values(244, "Wells",   "Chris",  "F", "25.02.95", 6);
```

Deleting or Modifying a table

```
drop instructor;
```

delete the instructor schema and its tuples

```
alter table instructor add office varchar(5);
```

adds attribute ‘office’ to the instructor schema

value of office is initialized to ‘null’

Lab: Schema Definition

From memory, write an SQL ‘create table’ statement for a schema with name ‘contributor’ and 3 attributes:

1. name, a variable length string with at most 20 characters
2. candidate, a variable length string with at most 20 characters
3. amount, a fixed point number with a total of 8 digits, with two to the right of the decimal point

Make ‘name’ the primary key of the schema, and require that amount not be null

Lab: Schema Definition

```
create table contributor (
   name varchar(20),
   candidate varchar(20),
   amount  numeric(8,2) not null,
   primary key (name)
)
```