SQL NULLs and Integrity constraints

Credit: Dr. Bruns

The NULL Value

CustomerID	name	zip	phone
0	Bethany Mills	93933	831-582-1615
1	Dolores Lyons	93902	null
2	Marc Todd	93907	831-582-9821

"null" is used when:

- the value is unknown (we don't know Dolores' number)
- the value does not exist (Dolores doesn't have a phone)

Handling of NULLs

What should happen when you add 5 to null?

```
sqlite> create table temp (
 ...> a integer,
 ...> b integer
 ...> );
sqlite> insert into temp values(1,2);
sqlite> insert into temp values(3,null);
sqlite> select * from temp;
1,2
3,
sqlite> select a+b from temp;
3

sqlite> select a*b from temp;
2
```

temp

a	b
1	2
3	null

Handling of NULLs, cont'd.

What happens when you compare a value to null, etc.?

```
sqlite> select * from temp where b = 2;
1,2
sqlite> select * from temp where b > 0;
1,2
sqlite> select * from temp where b is null;
3,
sqlite> select * from temp where b is not null;
1,2
```

temp

a	b
1	2
3	null

Note that it is not "b = null".

Generally, be careful with NULL.

Select count(b) from temp?

Database integrity constraints

Find three problems with this table

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	90000
22222	Einstein	Physics	95000
32343		History	60000
33456	Gold	Physics	87000
33456	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	-60000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Find three problems with this table

teaches

10101	CS-347	1	Fall	2009
12121	FIN-101	1	Spring	2010
15151	MUS-199	0	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Autumn	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	210
98345	EE-181	1	Spring	2009

Integrity Constraints

An **integrity constraint** is a condition on data.

If an integrity constraint doesn't hold, there is a problem (or inconsistency) with the data.

Integrity constraints can concern:

- A problem in a single row
- A problem between rows in a table
- A problem between tables

Examples

- A problem with a single row
  - salary value is -80,000
- A problem between rows in a table
  - key constraint violation
- A problem between tables
  - a foreign key constraint violation

SQL check constraints

```
create table department
(
    dept_name     varchar(20),
    building      varchar(15),
    budget        numeric(12,2) check (budget > 0),
    primary key (dept_name)
);
```

A department's budget must be greater than 0

```
create table section
(
    course_id       varchar(8),
    sec_id varchar(8),
    semester        varchar(6)
                    check (semester in
                    ('Fall', 'Winter', 'Spring', 'Summer')),
    ...
);
```

Semester can be only 'Fall', 'Winter', 'Summer', etc.

More SQL check constraints

```
create table course
(
   course_id     varchar(8)
      check (course_id like “%-%”),
   title         varchar(50),
   dept_name     varchar(20),
   credits       numeric(2,0) check (credits > 0),
);
```

course_id must contain “-”

```
create table course
(
   course_id     varchar(8)
      check(dept_name != “Comp. Sci.” or
            substr(course_id,1,3) = “CS-”),
   title         varchar(50),
   dept_name     varchar(20),
   credits       numeric(2,0)
);
```

If dept_name is “Comp. Sci.” then course_id must start with “CS-”

SQL check constraints, cont'd.

Do check constraints concern:

- a problem with a row? 
- a problem between rows in a table?
- a problem across tables?

SQL unique constraints

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

No two rows can have the same non-null dept_name values

Things to remember:

- `unique(a,b,c)` means that attributes a,b,c form a superkey
- unique constraints look like primary key constraints
- a table can have many unique constraints
- null = x is always false, so unique only cares about non-null value

SQL unique constraints, cont'd.

Do unique constraints concern:

- a problem with a row?
- a problem between rows in a table? 
- a problem across tables?

SQL referential integrity constraints

These constraints say that:

- if a row in table T1 has certain attribute values,
- then some row in a table T2 must have some corresponding attribute values.

A foreign key constraint is a kind of referential integrity constraint.

Foreign key constraints are the only kind we'll use.

SQL referential integrity constraints, cont'd.

Do unique constraints concern:

- a problem with a row?
- a problem between rows in a table?
- a problem across tables? 

What does SQLite do with constraints?

```
sqlite> insert into instructor values("84930", "Swanson", "Physics", 77000);
sqlite> insert into instructor values("84930", "Gomez", "History", 72000);
Error: UNIQUE constraint failed: instructor.ID
sqlite> insert into instructor values("84931", "Gomez", "History", 72000);
sqlite> insert into instructor values("84999", "Moody", "History", 12000);
Error: CHECK constraint failed: instructor
sqlite> insert into instructor values("84999", "Moody", "History", 92000);
sqlite> insert into instructor values("65199", "Kim", "Math", 92000);
sqlite> select * from department;
Biology,Watson,90000
Comp. Sci.,Taylor,100000
Elec. Eng.,Taylor,85000
Finance,Painter,120000
History,Painter,50000
Music,Packard,80000
Physics,Watson,70000
```

```
create table instructor
(
   ID                   varchar(5),
   name                 varchar(20) not null,
   dept_name            varchar(20),
   salary               numeric(8,2) check (salary > 29000),
   primary key (ID),
   foreign key (dept_name) references department
                        on delete set null
);
```

cascade delete 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
);
```