University of Toronto
CSC343, Fall 2017

SQL Data Definition Language: Solutions

1. Which of the following table definitions are valid? Where invalid, explain why.

```
create table Stuff1 (
   name text primary key,
   number int,
   rating float not null
);
```

```
create table Stuff2 (
   name varchar(25) primary key,
   number int primary key,
   rating float
);
```

```
create table Stuff3 (
   name text primary key,
   number int unique default 0,
   rating float
);
```

```
create table Stuff4 (
   name char(30) unique,
   number int unique,
   rating real
);
```

Solution: All but Stuff2 are valid. It generates this error:

```
ERROR:  multiple primary keys for table "stuff2" are not allowed
LINE 3:    number int primary key,
```

2. Suppose we have defined this table:

```
create table Fluff (
   this int,
   that int,
   other text unique,
   primary key (this, that)
);
```

Which of the following is valid? (Consider each as if it were being applied to any empty instance of the table.) For each that is invalid, identify the problem.

```
insert into Fluff values (1, 2, ’my’), (1, 2, ’night’);
insert into Fluff values (11, 22, ’twinkle’), (33, 44, ’twinkle’);
insert into Fluff values (100, 5, ’night’), (100, 10, ’my’);
insert into Fluff values (null, null, ’oh’);
insert into Fluff values (5, null, ’uh’);
insert into Fluff values (null, 20, ’a’), (null, 21, ’b’);
insert into Fluff values (80, 81, null);
insert into Fluff values (90, 91, null), (92, 93, null);
```

Solution: Here’s what each of the insert statements yields:

```
csc343h-dianeh=> insert into Fluff values (1, 2, ’my’), (1, 2, ’night’);
ERROR:  duplicate key value violates unique constraint "fluff_pkey"
DETAIL:  Key (this, that)=(1, 2) already exists.

csc343h-dianeh=> insert into Fluff values (11, 22, ’twinkle’), (33, 44, ’twinkle’);
ERROR:  duplicate key value violates unique constraint "fluff_other_key"
DETAIL:  Key (other)=(twinkle) already exists.

csc343h-dianeh=> insert into Fluff values (100, 5, ’night’), (100, 10, ’my’);
INSERT 0 2

csc343h-dianeh=> insert into Fluff values (null, null, ’oh’);
ERROR:  null value in column "this" violates not-null constraint
DETAIL:  Failing row contains (null, null, oh).

csc343h-dianeh=> insert into Fluff values (5, null, ’uh’);
ERROR:  null value in column "that" violates not-null constraint
DETAIL:  Failing row contains (5, null, uh).

csc343h-dianeh=> insert into Fluff values (null, 20, ’a’), (null, 21, ’b’);
ERROR:  null value in column "this" violates not-null constraint
DETAIL:  Failing row contains (null, 20, a).

csc343h-dianeh=> insert into Fluff values (80, 81, null);
INSERT 0 1

csc343h-dianeh=> insert into Fluff values (90, 91, null), (92, 93, null);
INSERT 0 2

csc343h-dianeh=> select * from Fluff;
 this | that | other
------+------+-------
  100 |    5 | night
  100 |   10 | my
   80 |   81 |
   90 |   91 |
   92 |   93 |
(5 rows)
```

3. Again, suppose we have defined this table:

```
create table Fluff (
   this int,
   that int,
   other text unique,
   primary key (this, that)
);
```

Which of these table definitions is valid, given the definition of table `Fluff`? Where invalid, explain why.

error, since either unique, or ALL or primary key

```
create table Nonsense1 (
    a int,
    b int,
    foreign key (b) references Fluff(this)
);
```

```
create table Nonsense2 (
    a int,
    b text references Fluff(other)
);
```

```
create table Nonsense3 (
    a int,
    b int,
    c int,
    foreign key (b, c) references Fluff
);
```

OK, implies primary key

```
create table Nonsense4 (
    a int references Fluff(blah),
    b int
);
```

blah not an attribute of Fluff

Solution: Only Nonsense2 and Nonsense3 are valid. Here's what each of the table definitions yields:

```
csc343h-dianeh=> create table Nonsense1 (
csc343h-dianeh(>     a int,
csc343h-dianeh(>     b int,
csc343h-dianeh(>     foreign key (b) references Fluff(this)
csc343h-dianeh(> );
ERROR:  there is no unique constraint matching given keys for referenced table "fluff"

csc343h-dianeh=> create table Nonsense2 (
csc343h-dianeh(>     a int,
csc343h-dianeh(>     b text references Fluff(other)
csc343h-dianeh(> );
CREATE TABLE

csc343h-dianeh=> create table Nonsense3 (
csc343h-dianeh(>     a int,
csc343h-dianeh(>     b int,
csc343h-dianeh(>     c int,
csc343h-dianeh(>     foreign key (b, c) references Fluff
csc343h-dianeh(> );
CREATE TABLE

csc343h-dianeh=> create table Nonsense4 (
csc343h-dianeh(>     a int references Fluff(blah),
csc343h-dianeh(>     b int
csc343h-dianeh(> );
ERROR:  column "blah" referenced in foreign key constraint does not exist
```

4. Can you think of any other ways that an attempt to define a foreign key could fail?