

CS61A Lecture 35

Friday, November 22, 2019

Announcements

- Homework 10 due Thursday 10/5.

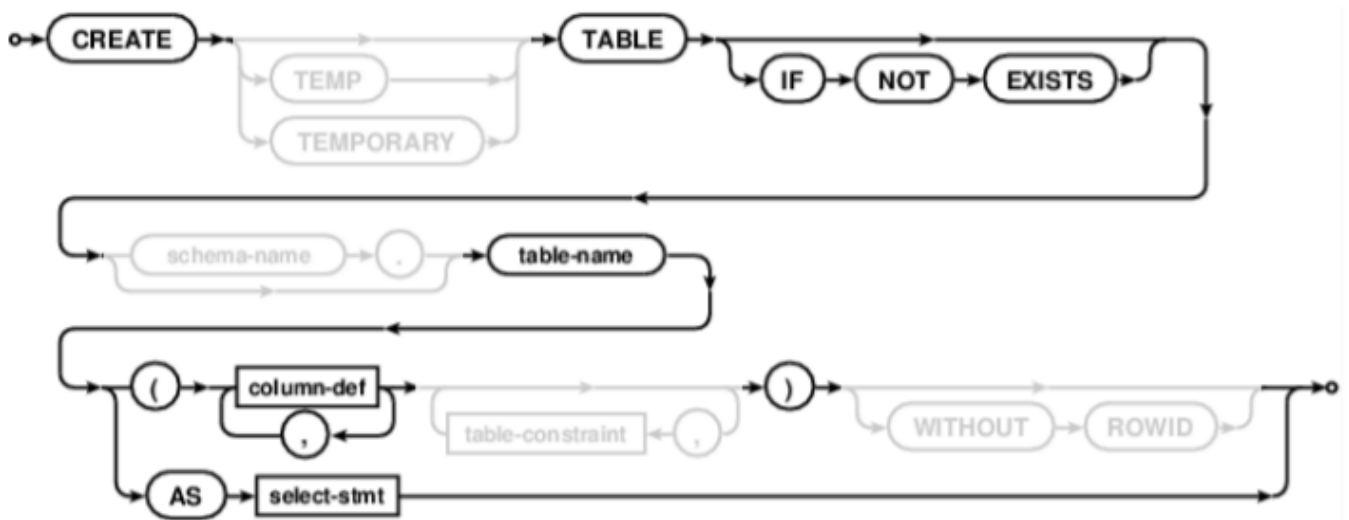
Making and Deleting Tables

Today is the fourth and final lecture on declarative programming. We know quite a lot about the `SELECT` statement, but what about all the other statements?

Create Table

The default `CREATE` statement has a lot of power. Here is a flowchart that describes all the different options in a `CREATE` statement. But you don't need to know all this! Here is everything you don't need to know, grayed out:

CREATE TABLE expression syntax:

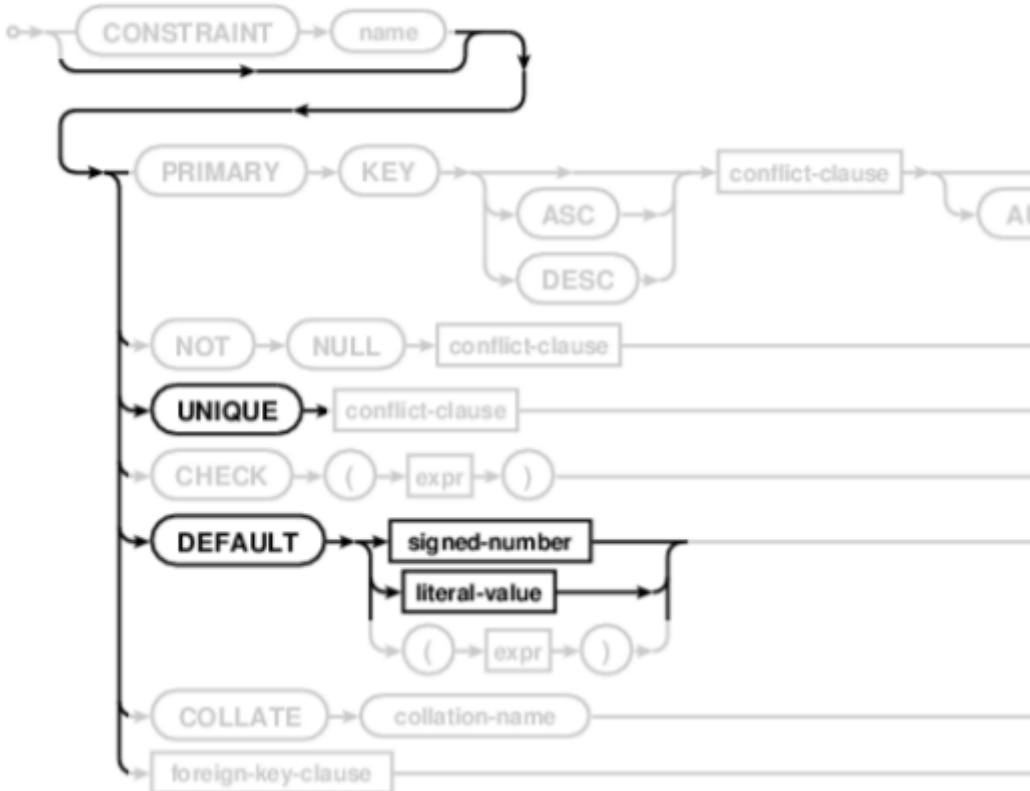


And here's everything you can put into the `column-def` and `column-constraint` statements:

column-def:

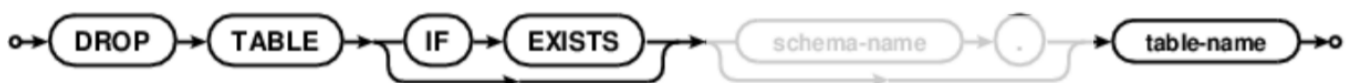


column-constraint:



Drop Table

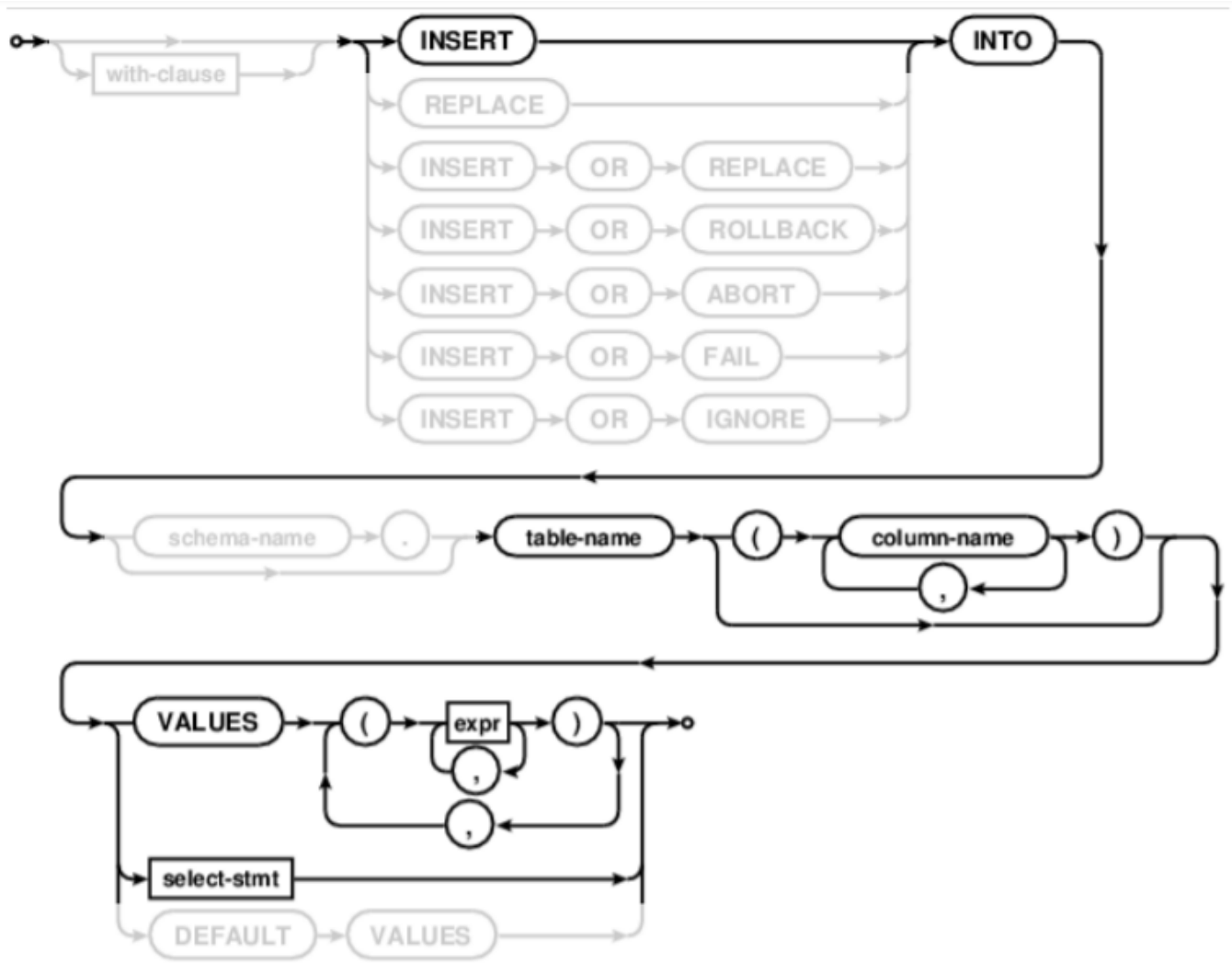
The `DROP` statement is pretty simple: it gets rid of a table. You can see everything it can do here:



Modifying Tables

Insert Statement

The `INSERT` statement lets you add new columns into an existing table.



For a table `t` with two columns, you can insert into only one column, or into both columns at once!

To insert into one column:

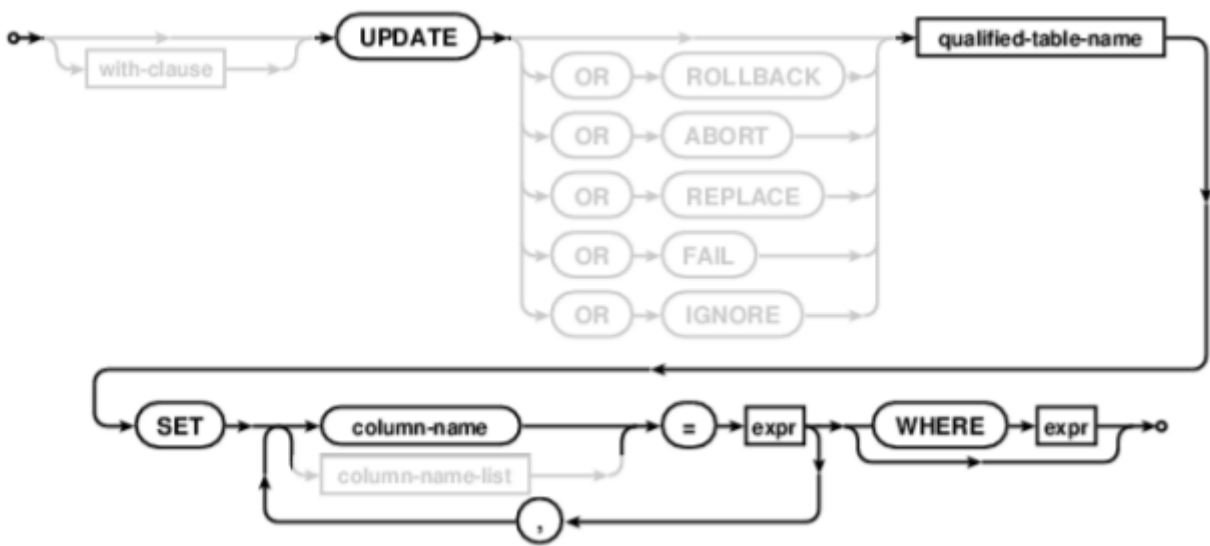
```
INSERT INTO t(column) VALUES (value);
```

To insert into both columns:

```
INSERT INTO t VALUES (value0,value1);
```

Update Statement

You can also mutate existing values in a table using the `UPDATE` statement.



We use the `WHERE` clause to update only certain rows instead of all the rows. For example, let's imagine we had a two-column table called `ints`, where the first column is all the integers from 2 to 25, and the second is just a series of 1s.

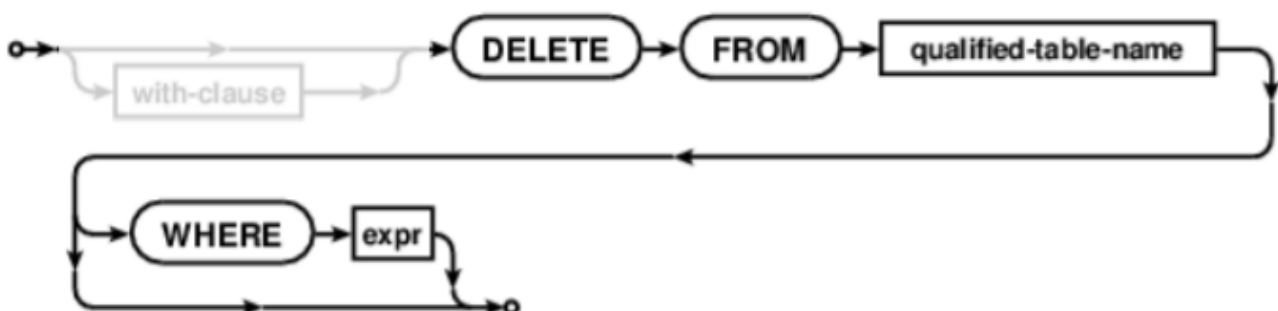
Let's say we wanted to update this table so that the second column tells us if a number is prime.

```
UPDATE ints SET prime=0 WHERE n>2 AND n%2=0;
UPDATE ints SET prime=0 WHERE n>3 AND n%3=0;
UPDATE ints SET prime=0 WHERE n>5 AND n%5=0;
```

Of course we have to continuously do this if we went beyond 25, but right now this solution works well enough to show you how the `UPDATE` statement works.

Delete Statements

The `DELETE` statement works just like the `UPDATE` statement, but instead of updating, it deletes rows from a table.



So if we wanted to delete all non-primes from our previous table:

```
DELETE FROM primes WHERE prime=0;
```

Python and SQL

You can also combine SQL tables with Python expressions:

```
$ python3
>>> import sqlite3
>>> db = sqlite3.Connection("n.db")
```

To execute SQL statements, we put the statement into a bound method call to `execute` :

```
>>> db.execute("CREATE TABLE nums SELECT 2 UNION SELECT 3;")
```

We can use Python expressions too:

```
>>> db.execute("INSERT INTO nums VALUES (?, ?), (?, ?)", range(4,7))
```

And finally, we can save our changes with the `commit` bound method.

Blackjack, CS61A Edition

Here's a program that simulates Blackjack using Python and SQL, where the cards are stored in SQL.

 WATCH VIDEO