1 Administrivia

Announcements

Recommendation: Use a special section of your notebook to collect SQL statement details. It will grow in terms of number of statements and aspects of statements; plan accordingly.

Assignment

Have a look through the PostgreSQL Documentation on the course web site to familiarize yourself with PostgreSQL.

From Last Time

Intro to relational data model.

Outline

1. SQL, psql info.

2. SQL DDL: creating tables, primary and candidate keys, constraints, and foreign keys.

3. Exercise.
2 SQL Variants, Useful psql Meta-Commands

1. SQL-92.

   Most DBMSs support most of this, with extensions.

2. SQL:99.

   Some DBMSs support parts of this.

3. SQL 3.

   Some DBMSs support parts of this?

Some useful psql meta-commands:

1. `\i <filename>` — Read and execute a list of SQL commands in the text file `<filename>.

2. `\d <table>` — List `<table>`’s description.

3. `\l` — List available databases on the system.

4. `\dS` — Display list of system tables. Example queries on two system tables:

   ```
   SELECT * FROM pg_user;
   SELECT * FROM pg_tables;
   ```

5. `\dp` — List permissions in current database.

6. `\dt` — List tables in current database.

Run `\?` in psql for the entire list and see the man page for psql.
3 SQL DDL

What we need to do:

1. Create tables.

2. Create assertions within tables.
   
   (a) Domain constraints: NOT NULL, DEFAULT.
   
   (b) Semantic constraints (cannot order a negative number of widgets).
   
   (c) Keys: primary, candidate, and foreign.

3. Create assertions between (external to) tables.

4. Create data types (or restrict existing types).

3.1 Creating Tables

CREATE TABLE Course (  
  CrsCode CHAR(6),  
  DeptId CHAR(4),  
  CrsName CHAR(20),  
  Descr CHAR(100) ); -- Use text?

Some data types in PostgreSQL:

1. Numeric:
   
   (a) INTEGER — four bytes. Also SMALLINT and BIGINT.
   
   (b) REAL — single precision. Also DOUBLE PRECISION.
   
   (c) SERIAL — unique serial number. Some caveats — see manual.
2. Character:

(a) **CHAR(n)** — fixed length string of \(n\) characters.

(b) **VARCHAR(n)** — variable length string of at most \(n\) characters.

(c) **TEXT** — variable length string without limit. Not SQL standard.

3. Others: Monetary, date/time, boolean, etc.

See manual.

(Refer to the online documentation for details.

### 3.2 Primary and Candidate Keys

```sql
CREATE TABLE Course (  
    CrsCode CHAR(6),  
    DeptId CHAR(4),  
    CrsName CHAR(20),  
    Descr CHAR(100),  
    PRIMARY KEY (CrsCode),  
    UNIQUE (DeptId, CrsName) );
```

1. One **PRIMARY KEY**. Table usually indexed on this.

2. One **UNIQUE** constraint for each remaining candidate key.

### 3.3 Specifying Constraints

1. Domain and semantic constraint examples:

```sql
CREATE TABLE Course (  
    CrsCode CHAR(6) NOT NULL,  
    Limit INTEGER DEFAULT '10',  
    Division CHAR(5),
```


(a) Use for intra-table constraints.

(b) Chicken and egg problem: use ALTER TABLE to add checks once tables are populated.

2. Assertion example:

```sql
CREATE ASSERTION CoursesShallNotBeEmpty
    CHECK (NOT EXISTS (
        SELECT * FROM Teaching Te
        WHERE NOT EXISTS (
            SELECT * FROM Transcript Tr
            WHERE Te.CrsCode = Tr.CrsCode
            AND Te.Semester = Tr.Semester )))
```

(a) NOT EXISTS: empty set test. EXISTS?

Briefly explain the correlated sub-query: for loop similarity.

(b) Note: this will fail before the registration period!

(c) Use for inter-table constraints.

(d) Tables must first exist!

(e) PostgreSQL does not have assertions! Use triggers?

### 3.4 Foreign Keys

```sql
CREATE TABLE Teaching (
    ProfId INTEGER,
    CrsCode CHAR(6),
```
1. In PostgreSQL the referenced key must be a primary key!

4 Exercise

See handout.