



KOREA UNIVERSITY
DATABASE LAB

Chapter 4 - Lab

SQL Practice 4

Database Setup

- Useful PostgreSQL commands
 - `\h`: help, `\h command`: help on the command
 - `\d`: list tables, `\d table_name`: describe table
 - `\i file_name`: import SQL script
 - `\c database_name`: connect to the database
 - `\q`: quit PostgreSQL
 - History 기능 제공 (위, 아래 화살표 사용)
- Reuse 'practice2' database of the previous lab – SQL 2.

DDL Statements for University Database

“university.sql”

- create table classroom(building varchar(15), room_number varchar(7), capacity numeric(4,0), primary key (building, room_number));
- create table department(dept_name varchar(20), building varchar(15), budget numeric(12,2) check (budget > 0), primary key (dept_name));
- create table course(course_id varchar(8), title varchar(50), dept_name varchar(20), credits numeric(2,0) check (credits > 0), primary key (course_id), foreign key (dept_name) references department (dept_name) on delete set null);
- create table instructor(ID varchar(5), name varchar(20) not null, dept_name varchar(20), salary numeric(8,2) check (salary > 9000), primary key (ID), foreign key (dept_name) references department (dept_name) on delete set null);
- create table section(course_id varchar(8), sec_id varchar(8), semester varchar(6) check (semester in ('Fall', 'Winter', 'Spring', 'Summer')), year numeric(4,0) check (year > 1701 and year < 2100), building varchar(15), room_number varchar(7), time_slot_id varchar(4), primary key (course_id, sec_id, semester, year), foreign key (course_id) references course (course_id) on delete cascade, foreign key (building, room_number) references classroom (building, room_number) on delete set null);
- create table teaches(ID varchar(5), course_id varchar(8), sec_id varchar(8), semester varchar(6), year numeric(4,0), primary key (ID, course_id, sec_id, semester, year), foreign key (course_id, sec_id, semester, year) references section (course_id, sec_id, semester, year) on delete cascade, foreign key (ID) references instructor (ID) on delete cascade);
- create table student(ID varchar(5), name varchar(20) not null, dept_name varchar(20), tot_cred numeric(3,0) check (tot_cred >= 0), primary key (ID), foreign key (dept_name) references department (dept_name) on delete set null);
- 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 (course_id, sec_id, semester, year) on delete cascade, foreign key (ID) references student (ID) on delete cascade);
- create table advisor(s_ID varchar(5), i_ID varchar(5), primary key (s_ID), foreign key (i_ID) references instructor (ID) on delete set null, foreign key (s_ID) references student (ID) on delete cascade);
- create table time_slot(time_slot_id varchar(4), day varchar(1), start_hr numeric(2) check (start_hr >= 0 and start_hr < 24), start_min numeric(2) check (start_min >= 0 and start_min < 60), end_hr numeric(2) check (end_hr >= 0 and end_hr < 24), end_min numeric(2) check (end_min >= 0 and end_min < 60), primary key (time_slot_id, day, start_hr, start_min));
- create table prereq(course_id varchar(8), prereq_id varchar(8), primary key (course_id, prereq_id), foreign key (course_id) references course (course_id) on delete cascade, foreign key (prereq_id) references course (course_id));

Exercise

- Make examples for the followings
 1. The university database schema contains various integrity constraints. Execute some SQL statements violating them.
 - Primary key constraints, foreign key constraints, not null constraints, etc.
 - Hint: `\d table_name`
 2. Make two or more concurrently executed transactions, and show they are executed in an isolated manner.
 - Hint: Make two (or more) windows (i.e., terminals) and use 'begin transaction' commands
 3. Make users and set up a few authorization rules; Show some non-authorized accesses.
 - Hint: `\h create user`, `\h grant`, `\h revoke`
 - Hint: Use two or more windows for different users
 4. Create some views and show how view maintenance works and how view update is processed.
 - Update of source relation vs. update of view.

Homework

- Complete today's practice exercise
- Take some screenshots containing the execution results
- Submit your report on blackboard
 - 10:29:59, May 17th, 2022
 - **Only PDF files** are accepted
 - **No late submission**



KOREA UNIVERSITY
DATABASE LAB

End of Lab