

A herd of elephants is silhouetted against a bright, golden sunset sky. The elephants are standing in a grassy field, and their long shadows are cast on the ground. The overall mood is peaceful and majestic.

Wireless Software  
Laboratory

# PostgreSQL

## SQL Fundamental 1

wisoft

# 차례

Transaction

SQL Statement

Table Control

Data Manipulation Language

INSERT, UPDATE, DELETE, SELECT



[http://www.sourcehov.com/sites/default/files/Financial-Transaction-Svcs\\_0.jpg](http://www.sourcehov.com/sites/default/files/Financial-Transaction-Svcs_0.jpg)

# Transaction

---

## About

## Transaction

- About Transaction

- 시스템에서 사용되는 '더 이상 쪼갤 수 없는' 업무처리의 단위
- DB의 무결성이 보장되는 상태에서 요청된 작업을 완수하기 위한 작업의 기본 단위

- Example

- 은행에서 A(송신)와 B(수신) 사용자 사이의 계좌이체 - A가 B에게 100만원 송금
- [STEP 01] 은행 데이터베이스에서 A의 잔액 확인
- [STEP 02] B의 계좌가 있는지 확인 후, 계좌가 있으면 100만원 송금
- [STEP 03] 송금이 정상적으로 이루어졌는지 확인하기(A계좌 -100만원, B계좌 +100만원)
- [STEP 04] STEP 03 단계가 정상 완료되었으면 종료하고, 그렇지 않으면 모든 작업을 취소한 후 처음부터 다시 처리



# Commit Mode 설정 및 확인

## Commit Mode

```
\echo :AUTOCOMMIT
```

## Set Commit Mode (Session)

```
\set AUTOCOMMIT on/off
```

## Statement Level

```
BEGIN Command
```

### Statement Level

```
SQL> \echo :AUTOCOMMIT
SQL> BEGIN;
SQL> CREATE TABLE COMMIT_TEST
      (no BIGSERIAL, name VARCHAR(10));
SQL> INSERT INTO COMMIT_TEST(name) VALUES ('test');
SQL> SELECT * FROM COMMIT_TEST;
SQL> ROLLBACK;
```

### Session Level

```
SQL> \echo :AUTOCOMMIT
SQL> \set AUTOCOMMIT on/off
```

### Global Session Level

```
vim %APPDATA%\Roaming\postgresql\psqlrc.conf
\set AUTOCOMMIT off
```



[http://www.bigfootcmms.com/wp-content/uploads/SQL\\_query.jpg](http://www.bigfootcmms.com/wp-content/uploads/SQL_query.jpg)

## SQL Statement

## Statement

- Data Definition Language

### DDL - 데이터베이스 객체/데이터 구조 정의

CREATE	데이터베이스 객체 등을 생성
ALTER	존재하는 데이터베이스 객체 등을 다시 정의
DROP	데이터베이스 객체 등을 제거

- Data Manipulation Language

### DML - 데이터의 검색과 조작(삽입, 갱신, 그리고 삭제) 처리

INSERT	데이터베이스 객체에 데이터 삽입
DELETE	데이터베이스 객체에 저장되어 있는 데이터 삭제
UPDATE	존재하는 데이터베이스 객체에 저장되어 있는 데이터 수정
SELECT	데이터베이스 객체에 저장되어 있는 데이터 검색

- Data Control Language

### DCL - 데이터베이스 사용자/객체의 권한 제어

GRANT	데이터베이스 객체에 권한 부여
REVOKE	데이터베이스 객체에 허가된 권한 취소

---

## How to run SQL Statement

- Using PSQL Console
- Using Source Command with SQL Script files
  - Using Source Command

```
SQL> \i filepath/filename.sql;
```

### 주의

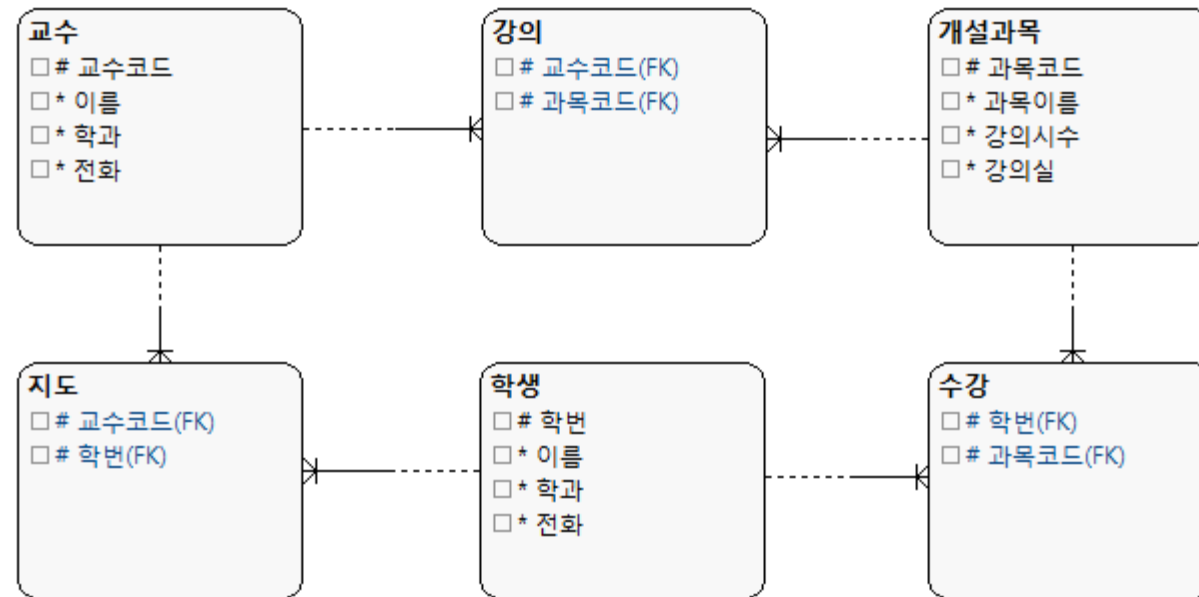
윈도우 플랫폼에서는 파일 경로에서 ‘\’ 대신 ‘/’를 사용해야 함.



# Sample Data

## Data Model

- Course registration system
  - Key Entity: 교수(Professor), 학생(Student), 개설과목(Course)
  - Association Entity: 강의(Lecture), 지도(Advise), 수강(Register)



## Sample Data (1/2)

- Professor Table

교수코드	이름	학과	전화
P001	안기홍	컴퓨터공학과	821-1101
P002	김정호	컴퓨터공학과	821-1102
P003	이창석	전파공학과	821-1201
P004	박현주	전파공학과	821-1202
P005	김응규	정보통신공학과	821-1301
P006	김은기	정보통신공학과	821-1302

- Student Table

학번	이름	학과	전화번호
S001	이정혁	전파공학과	123-4567
S002	오창세	전파공학과	234-5678
S003	마리아	컴퓨터공학과	345-6789
S004	김정은	컴퓨터공학과	456-7890
S005	김다인	정보통신공학과	567-8901
S006	박선영	정보통신공학과	678-9012

## Sample Data (2/2)

- Course Table

과목코드	과목명	강의시수	강의실
C001	TCP/IP	3	R001
C002	HTML	3	R002
C003	OS	3	R003
C004	Database	2	R004
C005	Java	2	R005
C006	C	2	R006

- Lecture

교수코드	과목코드
P002	C002
P002	C006
P003	C005
P004	C003
P004	C004
P006	C001

- Advise

교수코드	학번
P001	S001
P002	S002
P003	S003
P004	S004
P004	S005
P006	S006

- Register

학번	과목코드
S005	C001
S005	C003
S005	C004
S006	C001
S006	C003
S006	C004

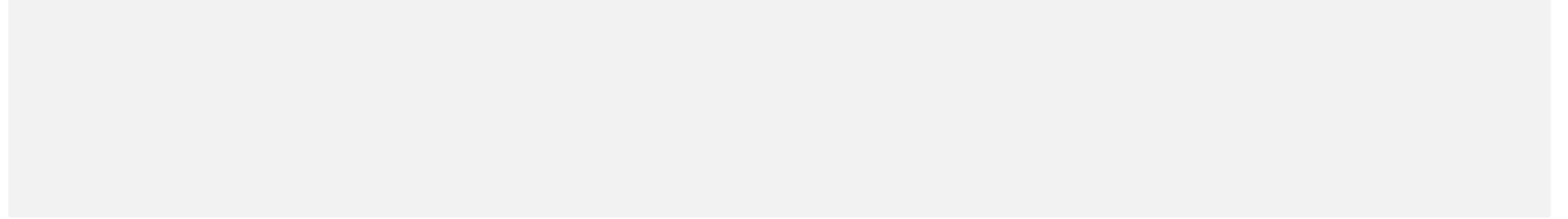
# 돈금없이 QUIZ

‘Education’ Database를 생성하시오.

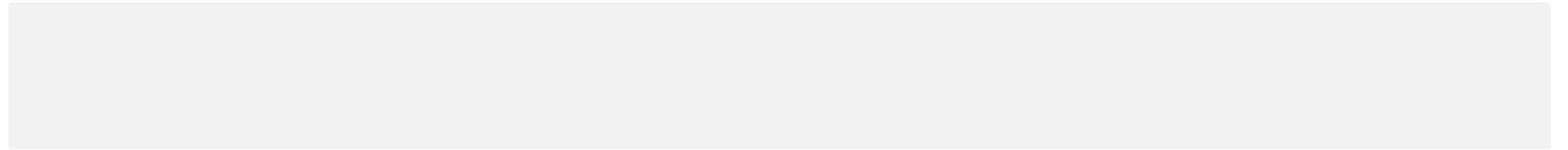
---

## Solution

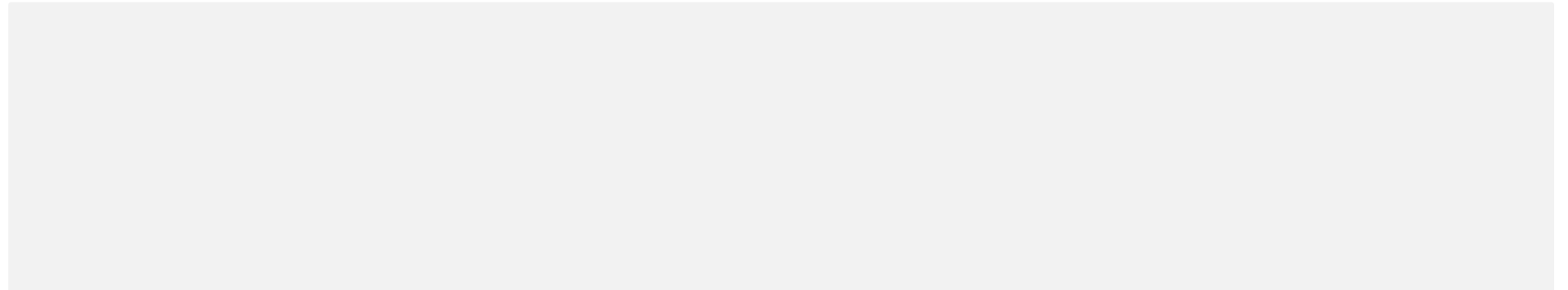
- 데이터베이스 생성 권한이 있는 계정으로 로그인을 한다.



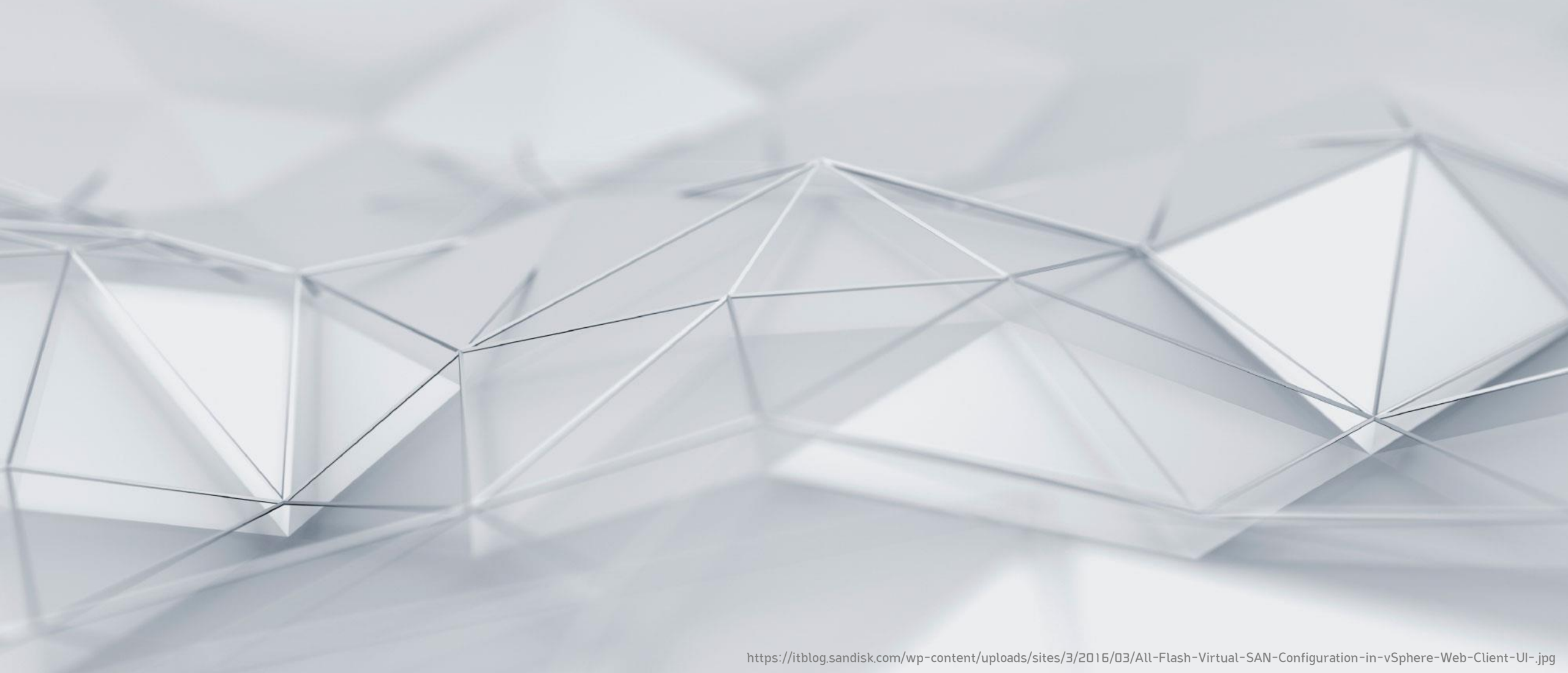
- 데이터베이스 'education'을 생성한다.



- 데이터베이스가 생성되었는지 확인하고 활성화한다. (권한은 scott에게 부여)







<https://itblog.sandisk.com/wp-content/uploads/sites/3/2016/03/All-Flash-Virtual-SAN-Configuration-in-vSphere-Web-Client-UI-.jpg>

# Build Environment

---

## Processing

## Build Environment

- Script Download
  - [https://git.wisoft.io/hyeonsig/material-public/blob/master/seminar/2017\\_postgresql/201707-getting-started-with-pgsql.zip](https://git.wisoft.io/hyeonsig/material-public/blob/master/seminar/2017_postgresql/201707-getting-started-with-pgsql.zip)
- FILE Moving
  - C:\WSQL
- Decompression
- FILE Check

# 테이블 생성

---

## 테이블 생성하기 (1/3)

- Syntax

```
CREATE TABLE [IF NOT EXISTS] table_name (<column-definition>);
```

- table\_name

생성할 테이블 이름을 입력

- IF NOT EXISTS

기존에 동일한 이름의 테이블 이름이 있으면 생성하지 않음

- column-definition

테이블을 구성하는 컬럼 이름과 컬럼 도메인 정보 등을 지정

- Whitepaper

<https://www.postgresql.org/docs/9.5/static/sql-createtable.html>

## 테이블 생성하기 (2/3)

- Syntax <column-definition>

```
<column_name> <datatype>  
    [NOT NULL | NULL]  
    [CHECK <expression>]  
    [DEFAULT <default_value>]  
    [UNIQUE [KEY] | [PRIMARY] KEY]  
...
```

- NOT NULL | NULL - NULL 값 허용 지정
- CHECK - 컬럼의 제약조건 지정
- DEFAULT - 컬럼의 기본값 지정
- UNIQUE KEY | PRIMARY KEY - 유일 값, 기본 키 지정

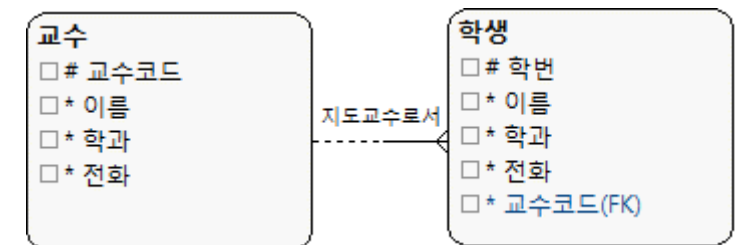
## 테이블 생성하기 (3/3)

- Syntax FOREIGN KEY

```
FOREIGN KEY <Table Constraint> ::=  
  
[ CONSTRAINT <Constraint name> ]  
FOREIGN KEY (referencing <Column name> [ {,<Column name>}... ])  
    REFERENCES referenced <Table name>  
    [ (referenced <Column name> [ {,<Column name>}... ] ) ]  
[ <constraint attributes> ]
```

- Example

```
CREATE TABLE 학생 (  
    SPCODE VARCHAR(4) NOT NULL PRIMARY KEY,  
    ...  
    CONSTRAINT student_pcode_fk  
    FOREIGN KEY (SPCODE)  
    REFERENCES 교수(PCODE)  
);
```





## 연습문제 (1/6)

- 다음 테이블을 생성하시오.

컬럼이름	데이터 타입	제약조건	비고
PCODE	VARCHAR(4)	PRIMARY KEY	교수코드
PNAME	VARCHAR(10)		이름
PDEPT	VARCHAR(12)		학과
PPHONE	VARCHAR(8)		전화

## 연습문제 (2/6)

- 다음 테이블을 생성하시오.

컬럼이름	데이터 타입	제약조건	비고
SCODE	VARCHAR(4)	PRIMARY KEY	학번
SNAME	VARCHAR(10)		이름
SDEPT	VARCHAR(12)		학과
SPHONE	VARCHAR(8)		전화

## 연습문제 (3/6)

- 다음 테이블을 생성하시오.

컬럼이름	데이터 타입	제약조건	비고
CCODE	VARCHAR(4)	PRIMARY KEY	과목코드
CNAME	VARCHAR(10)		과목이름
CTIME	INTEGER		강의시수
CROOM	VARCHAR(8)		강의실

## 연습문제 (4/6)

- 다음 테이블을 생성하시오.

컬럼이름	데이터 타입	제약조건	비고
LPCODE	VARCHAR(4)	PRIMARY KEY	교수코드
LCCODE	VARCHAR(4)	PRIMARY KEY	과목코드

## 연습문제 (5/6)

- 다음 테이블을 생성하시오.

컬럼이름	데이터 타입	제약조건	비고
APCODE	VARCHAR(4)	PRIMARY KEY	교수코드
ASCODE	VARCHAR(4)	PRIMARY KEY	학번

## 연습문제 (6/6)

- 다음 테이블을 생성하시오.

컬럼이름	데이터 타입	제약조건	비고
RSCODE	VARCHAR(4)	PRIMARY KEY	학번
RCCODE	VARCHAR(4)	PRIMARY KEY	과목코드



# 데이터 등록

## 데이터 삽입

- Syntax

```
INSERT [INTO] <table_name> [(column_name>[, column_name, ...])]
VALUES ({expression | DEFAULT }, ...) [, (...), ...];
```

- table\_name

데이터를 삽입할 테이블을 지정

- VALUES

삽입할 데이터 값

- expression

컬럼에 입력할 값을 나열하며, 기본 값을 사용하고 싶을 때에는 DEFAULT 키워드 사용

- Whitepaper

<https://www.postgresql.org/docs/9.5/static/sql-insert.html>

---

## 데이터 등록 방법

- Using SQL Command
- Using Source Command with SQL Script files

- Using Load Data Infile

```
COPY table_name FROM 'file_path'  
( FORMAT CSV, DELIMITER(',') );
```

- 대용량의 TEXT 파일이나 CSV 파일 등을 활용하여 대용량의 데이터를 삽입할 때  
사용함

## 연습문제 (1/4)

- ‘Professor’ 테이블에 데이터를 등록하시오.

교수코드	이름	학과	전화
P001	안기흥	컴퓨터공학과	821-1101
P002	김정호	컴퓨터공학과	821-1102
P003	이창석	전파공학과	821-1201
P004	박현주	전파공학과	821-1202
P005	김응규	정보통신공학과	821-1301
P006	김은기	정보통신공학과	821-1302

## 연습문제 (2/4)

- ‘Student’ 테이블에 데이터를 등록하시오.

학번	이름	학과	전화번호
S001	이정혁	전파공학과	123-4567
S002	오창세	전파공학과	234-5678
S003	오라클	컴퓨터공학과	345-6789
S004	전현식	컴퓨터공학과	456-7890
S005	신동성	정보통신공학과	567-8901
S006	이정민	정보통신공학과	678-9012

## 연습문제 (3/4)

- 'Course' 테이블에 데이터를 등록하시오.

과목코드	과목명	강의시수	강의실
C001	TCP/IP	3	R001
C002	HTML	3	R002
C003	OS	3	R003
C004	Database	4	R004
C005	Java	2	R005
C006	C	2	R006



## 연습문제 (4/4)

- 다음 테이블에 데이터를 등록하시오.

Lecture

교수코드	과목코드
P002	C002
P002	C006
P003	C005
P004	C003
P004	C004
P006	C001

Advise

교수코드	학번
P001	S001
P002	S002
P003	S003
P004	S004
P004	S005
P006	S006

Register

학번	과목코드
S005	C001
S005	C003
S005	C004
S006	C001
S006	C003
S006	C004

# 데이터 수정

## 데이터 수정

- Syntax

```
UPDATE <table_name> SET column_name={expression | DEFAULT}  
                        [, columnmn_name={expression | DEFAULT }]  
...  
[WHERE <query_condition>]
```

- table\_name

갱신을 수행할 대상 테이블을 지정

- column\_name={expression | DEFAULT}

해당 컬럼의 값을 표현식으로 변경하거나 기본 값으로 변경

- query\_condition

변경하고자 하는 테이블 내의 대상을 지정하기 위해 사용하므로 매우 중요

- Whitepaper

<https://www.postgresql.org/docs/9.5/static/sql-update.html>

## 연습문제 (1/3)

- ‘Professor’ 테이블에서 안기홍 교수님을 안기홍 교수님으로 변경하시오.

변경 전

교수코드	이름	학과	전화
P001	안기홍	컴퓨터공학과	821-1101
P002	김정호	컴퓨터공학과	821-1102
P003	이창석	전파공학과	821-1201
P004	박현주	전파공학과	821-1202
P005	김응규	정보통신공학과	821-1301
P006	김은기	정보통신공학과	821-1302

변경 후

교수코드	이름	학과	전화
P001	안기홍	컴퓨터공학과	821-1101
P002	김정호	컴퓨터공학과	821-1102
P003	이창석	전파공학과	821-1201
P004	박현주	전파공학과	821-1202
P005	김응규	정보통신공학과	821-1301
P006	김은기	정보통신공학과	821-1302

## 연습문제 (2/3)

- 'Advise' 테이블에서 교수코드가 P004를 모두 P003으로 변경하시오.

변경 전

교수코드	학번
P001	S001
P002	S002
P003	S003
P004	S004
P004	S005
P006	S006

변경 후

교수코드	학번
P001	S001
P002	S002
P003	S003
P003	S004
P003	S005
P006	S006

## 연습문제 (3/3)

- 강의 시수가 2인 과목의 강의 시수를 1 증가하고, 강의실을 Lab1로 변경하라.

변경 전

과목코드	과목명	강의시수	강의실
C001	TCP/IP	3	R001
C002	HTML	3	R002
C003	OS	3	R003
C004	Database	4	R004
C005	Java	2	R005
C006	C	2	R006

변경 후

과목코드	과목명	강의시수	강의실
C001	TCP/IP	3	R001
C002	HTML	3	R002
C003	OS	3	R003
C004	Database	4	R004
C005	Java	3	Lab1
C006	C	3	Lab1

# 데이터 삭제

---

## 데이터 삭제

- Syntax

```
DELETE FROM <table_name> [WHERE <query_condition>];
```

- table\_name

테이블 내용을 삭제할 대상 테이블 이름을 입력

- query\_condition

이 조건을 생략하는 테이블의 전체 내용을 제거

삭제하고자 하는 테이블 내의 대상을 지정하기 위해 사용하므로 매우 중요

- Whitepaper

<https://www.postgresql.org/docs/9.5/static/sql-delete.html>



# 데이터 조회

---

## 데이터 조회

- Syntax

```
SELECT <target_column> FROM <table_name>  
[WHERE <query_condition>]  
[ORDER BY <column_name>];
```

- target\_column

조회할 컬럼 리스트

- table\_name

조회할 대상 테이블

- query\_condition

조회하고자 하는 테이블 내의 대상을 지정

- Whitepaper

<https://www.postgresql.org/docs/9.5/static/sql-select.html>

## 연습문제 (1/3)

- 전체 교수의 목록을 보여주는 질의를 작성하라.

### 질의 결과

교수코드	이름	학과	전화
P001	안기홍	컴퓨터공학과	821-1101
P002	김정호	컴퓨터공학과	821-1102
P003	이창석	전파공학과	821-1201
P004	박현주	전파공학과	821-1202
P005	김응규	정보통신공학과	821-1301
P006	김은기	정보통신공학과	821-1302

## 연습문제 (2/3)

- 전체 교수의 목록을 보여주는 질의를 작성하라. 단, 이름 순으로 출력하라.

### 질의 결과

교수코드	이름	학과	전화
P006	김은기	정보통신공학과	821-1302
P005	김응규	정보통신공학과	821-1301
P002	김정호	컴퓨터공학과	821-1102
P004	박현주	전파공학과	821-1202
P001	안기홍	컴퓨터공학과	821-1101
P003	이창석	전파공학과	821-1201

## 연습문제 (3/3)

- 전체 교수의 목록을 보여주는 질의를 작성하라.

단, 학과별로 출력해야 하며, 학과 이름이 동일할 때는 이름 순으로 출력하라.

### 질의 결과

교수코드	이름	학과	전화
P004	박현주	전파공학과	821-1202
P003	이창석	전파공학과	821-1201
P006	김은기	정보통신공학과	821-1302
P005	김응규	정보통신공학과	821-1301
P002	김정호	컴퓨터공학과	821-1102
P001	안기홍	컴퓨터공학과	821-1101

계산된 컬럼

---

## Computed column

- Overview
  - Computed column does not exist in the database but is computed using data in existing columns
- Arithmetic operators
  - + for addition
  - - for subtraction
  - \* for multiplication
  - / for division

## 연습문제

- 과목코드, 과목명, 과목의 원 시수와 시수에 한 시간씩 더한 시수를 출력하는 질의를 작성하시오.

```
SELECT CCODE, CNAME, CTIME, CTIME+1 AS "PLUS1"  
FROM COURSE;
```

### 질의 결과

과목코드	과목이름	과목시수	과목시수+1
C001	TCP/IP	3	4
C002	HTML	3	4
C003	OS	3	4
C004	Database	4	5
C005	Java	3	4
C006	C	3	4



# WHERE Clause

---

## Overview

- WHERE clause:
  - Used to retrieve rows that satisfy some condition
  - What is the name of professor code number 'P004'?

```
SELECT PNAME  
FROM PROFESSOR  
WHERE PCODE='P004';
```

- Simple condition
  - column name, comparison operator followed by either a column name or a value

## 비교 연산자

- Comparison Operator

Operator	Description
=	Equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
<>	Not equal to
!=	Not equal to

- SELECT command to find all courses with lectures that exceed 2 time per week.

```
SELECT *  
FROM COURSE  
WHERE CTIME > 2;
```

---

## 복합 연산자

- Compound condition
  - connects two or more simple conditions with AND, OR, and NOT operators
- AND operator
  - all simple conditions are true
- OR operator
  - any simple condition is true
- NOT operator
  - reverses the truth of the original condition

---

## BETWEEN

- Between Operator
  - Use instead of AND operator  
BETWEEN 2 and 5 ( $\geq 2$  AND  $\leq 5$ )
  - Use when searching a range of values
  - Makes SELECT commands simpler to construct
  - Inclusive  
when using BETWEEN 2000 and 5000, values of 2000 or 5000 would be true

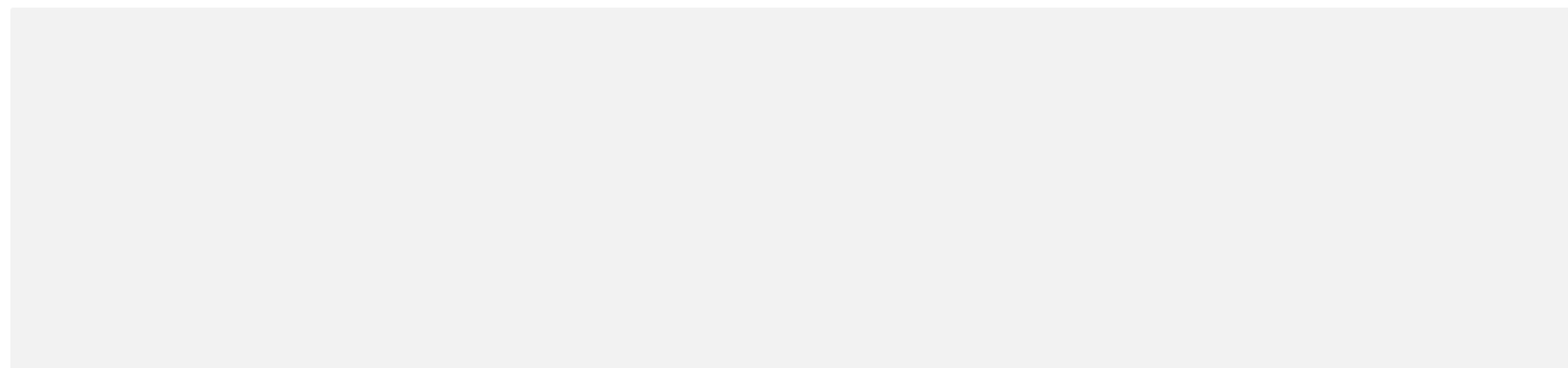
---

## LIKE

- LIKE Operator
  - Used for pattern matching
  - LIKE %Central% will retrieve data with those characters;  
e.g., “3829 Central” or “Centralia”
  - Underscore (\_) represents any single character;  
e.g., “T\_M” for TIM or TOM or T3M

## 연습문제 (1/2)

- 전파공학과 교수 목록을 이름 역순으로 출력하는 질의를 작성하라.

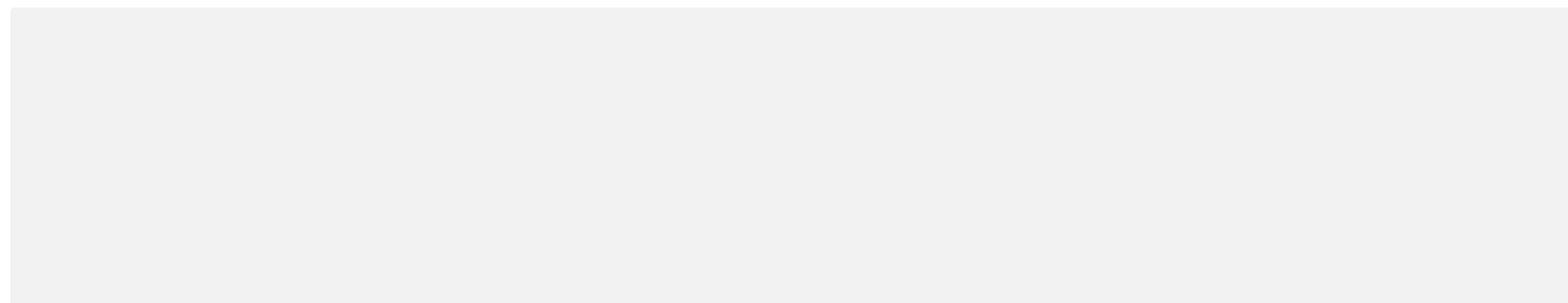


### 질의 결과

교수코드	이름	학과	전화
P003	이창석	전파공학과	821-1201
P004	박현주	전파공학과	821-1202

## 연습문제 (2/2)

- 교수 목록에서 성이 '김'씨인 교수 목록을 출력하라.



### 질의 결과

교수코드	이름	학과	전화
P002	김정호	컴퓨터공학과	821-1102
P005	김응규	정보통신공학과	821-1301
P006	김은기	정보통신공학과	821-1302



# QnA

hsjeon@hanbat.ac.kr

<http://blog.ngelmaum.org>

***Thank You***