Database Systems Lab

SESSION 10

SQL Data Definition Language

SQL is a non-procedural relational database language. SQL mainly contains two main sub-components: Data Definition Language (DDL) and Data Manipulation Language (DML).

DDL is useful for implementing a relational schema in a given relational database. It is mainly comprise of:

- CREATE statements
- ALTER statements
- DROP statements

DML is useful working with the actual data. DML is comprised of the following:

- SELECT
- INSERT
- UPDATE
- DELETE

The focus for this lab session is DDL.

Tasks to be completed

A) Installation (to be completed before coming to class)

1. RDBMS Installation: Install the DBMS software as per the instructions specified in the product document. For MySQL, you can find the instructions here: http://dev.mysql.com/downloads/mysql/. On Ubuntu, you can run the following command to install and assign the password for your installation:

```
sudo apt-get install mysql-server
```

2. Client Installation: All RDBMS can work with a variety of client tools. Some client tools provide a graphical user interface to the database. You must NOT use any GUI clients for your activities. For MySQL, everyone must be familiar and be comfortable using the command line tool called "mysql". So there must not be any need for any separate client installation on your machines:

   You can login using the following command and confirm the version:

```
mysql -u root –p

System must take you to the mysql prompt

mysql> select version();
mysql> quit
```

B) DDL Scripts

Implement the Company Database schema ("company_handout.pdf") using the database definition guidelines and database manipulation guidelines given in this document. As part of the implementation, you must create and upload the following SQL scripts that follow the given guidelines:

1. companydb_create.sql
2. companydb_alter.sql
3. companydb_drop.sql
4. companydb_data.sql

C) Database Definition Guidelines

1. First create an empty database using the procedure appropriate for the respective DBMS. For example, in MySQL, you can use the following commands to inspect the schema:

```
mysql> create database companydb;
mysql> use companydb;
mysql> show tables;
```

2. Have the relational schema design ready on paper. Having it in tabular form that we discussed will make it easier.
3. Full documentation on SQL DDL can be found here: http://dev.mysql.com/
4. The relation schema design must be implemented using a combination of three DDL script files as noted below:
    a. "CREATE" scripts: Create ONE large script file called **dbname_create.sql** (where dbname is replaced by the short name of the database; e.g., companydb_create.sql). This script file should contain:
        i. Basic CREATE TABLE statements for each table in the schema
        ii. PK Constraint against the appropriate attribute
        iii. NOT NULL constraints as applicable attributes
        iv. Do NOT include any FK constraints yet.

    Example for one table:

```
create table department(
    dname varchar(30) NOT NULL,
    dnumber smallint,
    mgr_ssn char(6),
    mgr_start_date date,
    constraint pk_department PRIMARY KEY (dnumber)
);
```

Useful mysql data types:

Character data types	CHAR(20) / *fixed length */
	VARCHAR(20) /* variable length */
Numeric data types	TINYINT (1 byte)
	SMALLINT (2 bytes)
	MEDIUMINT (3 bytes)
	INT (4 bytes)
	BIGINT (8 bytes)
	DECIMAL(p,s) FLOAT(p,s) p → Total number of digits s → Number of digits after the decimal point
Temporal Data Types	Date (default format YYYY-MM-DD)
	Datetime (default format YYYY-MM-DD HH:MI:SS)

b. "ALTER" scripts: Create ONE large script file called dbname_alter.sql (e.g., companydb_alter.sql). This script file should contain:

   i. ALTER TABLE statements to add foreign key constraints for each table as applicable.

   ii. ALTER TABLE statements to add any further semantic constraints as applicable.

Example for one table:

```
alter table employee
    add constraint fk_super_ssn FOREIGN KEY (super_ssn)
REFERENCES employee(ssn);
```

c. "DROP" scripts: Create ONE large script file called dbname_drop.sql (e.g., companydb_drop.sql). This script file should contain:

   i. ALTER TABLE statements to drop each of the FKs that were added.

Example for one table:

```
alter table employee
    drop FOREIGN KEY fk_super_ssn;
```

   ii. DROP TABLE statements to drop each of the tables.

Example for one table:

```
drop table employee;
```

D) Database Manipulation Guidelines

1. Insert rows into the database as per the data present in the handout. For attributes pertaining to the FK, assign NULL values. Example for one row given below:

```
insert into employee(fname, minit, lname, ssn, bdate, address,
sex, salary, super_ssn, dno)
VALUES('John','B','Smith','123456789','1965-01-09','731,
Fondren, Houston, TX','M',30000,NULL,NULL);
```

2. Update rows with appropriate foreign keys as per the handout. For example:

```
update employee
set super_ssn = '333445555', dno=5
where ssn='123456789';
```

3. You can check the entered data using the following simple SQL select statements:

```
select * from employee;

select ssn, fname
from employee;

select ssn, fname
from employee
where ssn = '123456789';

select ssn "ID Number", fname "FirstName"
from employee
where ssn = '123456789';
```

What to upload

1. Zip file containing the following scripts:
    a. companydb_create.sql
    b. companydb_alter.sql
    c. companydb_drop.sql
    d. companydb_data.sql