SQL Practice 2

1. Create database university;
2. Show databases;
3. Use university;
4. source DDL.sql;
5. source smallRelations.sql;

6. Nested subquery

(1) Nested subquery (check exists)

比較以下兩者寫法上的差別, 其結果是否相同? 對應查詢意義為何?

**select distinct** *course\_id* **from** *section* **as** *S* **where** *S.semester* = 'Fall' **and** *S.year* = 2009 **and   
 exists** (**select** \*  
 **from** *section* **as** *T* **where** *T.semester* = 'Spring' **and** *T.year*= 2010   
 **and** *S*.*course\_id* = *T*.*course\_id*) ;

**select distinct** *course\_id*

**from** *section*

**where** *semester* = **'**Fall**'** **and** *year*= 2009 **and** *course\_id* **in** (**select** *course\_id*

**from** *section*

**where** *semester* = **'**Spring**'** **and** *year*= 2010);

(2) Nested subquery (check unique)

比較以下兩者結果上的差異? 其個別對應的查詢意義為何?

**select distinct** *T*.*course\_id***from** *course* **as** *T***where** 1 = (**select count(***R*.*course\_id)* **from** *section* **as** *R* **where** *T*.*course\_id*= *R*.*course\_id* **and** *R*.*year* = 2009);

**select distinct S**.*course\_id*

**from** *section* **as S,** *section* **as****T****where S**.*course\_id*= **T**.*course\_id*

**and S**.*year* = 2009**and T**.*year* = 2009

**and** ((**S**.*semester*<> **T**.*semester*) or (**S**.*sec\_id*<> **T**.*sec\_id*));

(3) a subquery in from clause

**select** *dept\_name*, *avg\_salary***from** (**select** *dept\_name*, **avg** (*salary*) **as** *avg\_salary* **from** *instructor* **group by** *dept\_name*) as Temp  
**where** *avg\_salary* > 42000;

**select** *dept\_name*, **avg** (*salary*)

**from** *instructor***group by (***dept\_name*)  
**having avg** (*salary*)> 42000;

(4) a subquery in select clause

**select** *dept\_name*,   
 (**select count**(\*)   
 **from** *instructor* **where** *department*.*dept\_name* = *instructor*.*dept\_name*)  
 **as** *num\_instructors* **from** *department*;

(5) delete

**delete from** *instructor* **where** *dept\_name*= **'**Finance**'**;

**delete from** *instructor*

**where** *salary* < (**select avg** (*salary*) **from** *instructor*);

觀察對 instructor 的改變

(6) insert

**insert into** *course* **values** (**'**CS-437**'**, **'**Database Systems**'**, **'**Comp. Sci. **'**, 4);

**insert into** *student* **select** *ID, name, dept\_name, 0* **from**  *instructor*;

觀察對 instructor 的改變

(7) update

**update** *instructor* **set** *salary* = *salary* \* 1.03  
 **where** *salary* > 80000;

**update** *instructor* **set** *salary* = *salary* \* 1.05  
 **where** *salary* <= 80000;

觀察對 instructor 的改變

1. Write the following queries in SQL

(1) Find the name and department of the instructors who have the highest salary of instructors. (用nested subquery)

(2) Find the IDs of all instructors who didn’t teach any course. (用subqueries)

(3) Find the IDs of all instructors who didn’t teach any course. (use subqueries and not exists)

(4) Increase the salary of each instructor in the Comp. Sci. department by 10%.

(5) Delete all courses that have never been offered (that is, do not occur in the section relation).

(6) Insert every student whose *tot\_cred* attribute is greater than 100as an instructor in the same department, with a salary of $10,000.