SQL Practice

1. create database university;
2. show databases;
3. use university;
4. create table tablename;

source D:\DB\SQL\_lab1\DDL.sql

source D:\DB\SQL\_lab1\smallRelations.sql

1. Show tables;
2. Describe course;

**INSERT** **INTO** table\_name (column1, column2, column3...)

**VALUES** (value1, value2, value3...);

1. **select** \* from course;
2. (1) 比較

**select** *dept\_name*  
 **from** *instructor;*

**select distinct** *dept\_name*

**from** *instructor;*

(2) **select** *ID, name, salary/12*  
 **from** *instructor;*

(3)比較

**select** *name, course\_id*  
 **from** *instructor, teaches;*

**select** *name, course\_id***from** *instructor, teaches***where** *instructor.ID = teaches.id*;

(4) 比較

**select** *\** **from** *instructor, teaches***where** *instructor.ID = teaches.id*;

**select \***

**from** instructor **natural join** teaches**;**

(5) 比較

**select** *T. name***from** *instructor* **as** *T, instructor* **as** *S***where** *T.salary > S.salary*;

**select distinct** *T. name***from** *instructor* **as** *T, instructor* **as** *S***where** *T.salary > S.salary*;

(6) Partial match

**select** *name* **from** *instructor* **where***name* **like '**%in%';

(7) Sort the result

**select distinct** *name, salary* **from** *instructor*

**where** *dept\_name* = '*Physics*' **order by** *salary* ***desc****, name* ***asc***;

(8) Aggregation Functions

**select avg** (*salary*) as avg\_salary  
**from** *instructor***where** *dept\_name*=**'**Comp. Sci. **'**;

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

(9) 比較

**select count(***ID*)  
**from** *teaches***where** *semester* = **'**Spring**' and** *year* = 2010;

**select count(distinct** *ID*)  
**from** *teaches***where** *semester* = **'**Spring**' and** *year* = 2010;

(10) 比較

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

**from** *instructor*

**group by** *dept\_name*;

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

**from** *instructor*

**group by** *dept\_name*

**having avg** (*salary*) > 70000;

(11) Nested subquery (intersection)

**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);

(12) Nested subquery (compare with any)

**select** *name*

**from** *instructor*

**where** *salary* > **some (select *salary***

**from *instructor***

**where *dept\_name* =** 'Comp. Sci. '**)**;

(13) Nested subquery (compare with all)

**select** *name*

**from** *instructor*

**where** *salary* > **all (select *salary***

**from *instructor***

**where *dept\_name* =** 'Comp. Sci. '**)**;

1. Write the following queries in SQL
2. Find the titles of courses in the Comp. Sci. department that have 3 credits.
3. Find the IDs of all students who were taught by an instructor named Einstein, make sure there are no duplicates in the result.
4. Find the highest salary of instructors.
5. Find the highest salary of the instructors in each department.
6. Find the enrollment of each section that was offered in Autumn 2009.
7. Find the maximum enrollment, across all sections, in Autumn 2009.
8. Find the IDs of all instructors who didn’t teach any course. (use subqueries and not in)
9. Find all instructors earning the highest salary.
10. Find the sections that had the maximum enrollment in Autumn 2009.