

HW3 Solution ¹

March 5, 2018

Problem 1

1. What is the age of the oldest student?

```
1 SELECT MAX(age) AS max_age
2 FROM student;
```

max_age
99

2. Find the names and gpas of the students who have enrolled in course 302.

```
1 SELECT s.sname, s.gpa
2 FROM student s, enroll e
3 WHERE s.sid = e.sid AND e.cno = 302;
```

sname	gpa
Pierson, E.	3.5
Scott, Kim J.	3.8
Sather, Roberto B.	2.2
Stanley, Leotha T.	3.6
Jones, David S.	3.5
Paul, Mary W.	3.6
Soong, V.	3.5
Kellerman, S.	2.9
Korpel, E.	3.5

sname	gpa
Emile, R.	2
Carter, Jimmy	3.5
Kissinger, Henry	3.4
Andrews, R.	2.8
Auen, B.	2.7
Shoemaker, A.	3.5
Fy, Clara I.	2
Heilskov, G.	2.5
Baskett, Wayse T.	2.1

3. Find the names and majors of students who have taken an advanced course.

```
1 SELECT distinct s.sname, m.dname
2 FROM student s, major m, enroll e,
3     course c
4 WHERE s.sid = m.sid and e.sid = m.
    sid and e.cno = c.cno
    and c.cname LIKE "%Advanced%"
    ;
```

sname	dname
Bomber, C.	Computer Sciences
Hamilton, S.	Chemical Engineering
Liu, Huihusan	Chemical Engineering
Kasten, Norman L.	Chemical Engineering
Roger, Blotter N.	Civil Engineering
Natividad, A.	Civil Engineering
Calcinity, J.	Civil Engineering
Kennedy, Ed	Civil Engineering
Fred, Edwin B.	Industrial Engineering
Caucutt, B.	Industrial Engineering
Smith, Ike Z.	Industrial Engineering
Birch, M.	Mathematics

4. Find the names of students who have enrolled in both a course offered by the “Computer Sciences” department and a course offered by the ”Mathematics” department.

```

1  SELECT s.sname FROM student s, enroll e1,
    enroll e2 WHERE s.sid = e1.sid AND e1.
    dname = 'Computer Sciences' AND s.sid =
    e2.sid AND e2.dname = 'Mathematics';

```

sname
Zappa, F.

5. For each department, find the average gpa of the students majoring in that department along with the gpa difference between the highest and the lowest gpa.

```

1  SELECT m.dname, AVG (s.gpa) AS aveGpa, (MAX(s.gpa) - MIN(s.gpa)) AS GpaDiff
2  FROM student s, major m
3  WHERE s.sid = m.sid
4  GROUP BY m.dname;

```

dname	aveGpa	GpaDiff
Chemical Engineering	3.299	1.4
Civil Engineering	2.914	4
Computer Sciences	3.004	3.200
Industrial Engineering	2.769	3.799
Mathematics	3.242	2.899
Sanitary Engineering	2.799	0

6. How many students have only one major?

```

1  SELECT count(*) as total_students
2  FROM student s
3  WHERE EXISTS (SELECT m.sid
4                FROM major m
5                WHERE s.sid = m.sid
6                GROUP BY m.sid
7                HAVING count(*) = 1);

```

total_students
95

7. Find the name(s) of the student(s) who have taken the least number of courses.

```

1 SELECT s.sname, count(distinct e.cno) as numberofcourse
2 FROM student s LEFT OUTER JOIN enroll e ON s.sid=e.sid
3 GROUP BY s.sid, s.sname HAVING numberofcourse <= ALL
4     (SELECT count(distinct e1.cno) FROM student s1
5      LEFT OUTER JOIN enroll e1 ON s1.sid=e1.sid
6      GROUP BY s1.sid);

```

sname	numberofcourse
Ripper, Jack T.	0
Grzlbtlz, Q.	0

8. Find the name(s) of the oldest 3rd year student(s).

```

1 SELECT s.sname, s.age
2 FROM student s
3 WHERE s.age = (SELECT MAX(s2.age)
4                FROM student s2
5                WHERE s2.year = 3)
6 AND s.year = 3;

```

sname	age
Kennedy, ED	55

9. Find the ids, names, and gpas of the students who have taken all Computer Sciences courses.

```

1 SELECT s.sid, s.sname, s.gpa
2 FROM student s
3 WHERE NOT EXISTS (SELECT c.cno
4                   FROM course c
5                   WHERE c.dname = 'Computer Sciences'
6                   AND NOT EXISTS (SELECT e.cno
7                                   FROM enroll e
8                                   WHERE e.dname = c
9                                   .dname AND e.cno = c.cno
                                   AND e.sid = s.sid));

```

sid	sname	gpa
-----	-------	-----

```

1 SELECT s.sid, s.sname, s.gpa
2 FROM student s, enroll e
3 WHERE s.sid=e.sid and e.dname = 'Computer Sciences'
4 GROUP BY s.sid, s.sname, s.gpa
5 HAVING count(DISTINCT cno)=
6 (SELECT count(DISTINCT cno) FROM course c where c.dname =
   'Computer Sciences');

```

sid	sname	gpa
-----	-------	-----

10. Find the departments with none of their majors taking "Computer Sciences" courses.

```

1 SELECT *
2 FROM dept d
3 WHERE NOT EXISTS (SELECT m.sid
4                     FROM major m, enroll e
5                     WHERE m.sid = e.sid
6                     AND m.dname = d.dname
7                     AND e.dname = 'Computer
                        Sciences');

```

dname	numphds
Sanitary Engineering	3

11. Find the student names for each year with the maximum gpa.

```

1 SELECT s.year, s.sname, s.gpa
2 FROM student s
3 WHERE NOT EXISTS (SELECT * FROM student s1
4                   WHERE s.year=s1.year AND s.sid != s1.sid AND s.gpa<s1.gpa)
5 ORDER BY s.year;

```

year	sname	gpa
1	Scott, Kim J.	3.8
2	Quarnty, G.	4
3	Andrus, J.	3.7
3	Davis, Scott P.	3.7
4	Zappa, F.	4
5	Natividad, A.	4
5	Fred, Edwin B.	4
5	Altenhaus, Gloria	4
5	Longlastname, A.	4

12. Find the name(s) of the professor(s) who has (have) taught the least number of courses.

```

1 SELECT pname, count(*) AS work
2 FROM section
3 GROUP BY pname
4 HAVING count(*) <= ALL
5      ( SELECT count(*)
6        FROM section GROUP BY pname);

```

pname	work
Brian, C.	1
Brown, S.	1
Bucket, T.	1
Clark, E.	1
Edison, L.	1
Smith, S.	1
Walter, A.	1

13. For each department, find the student name (along with the departname) with the maximum average grade.

```

1 SELECT m.dname, s.sname, avg(e.grade) as maxgrade FROM major m, student s, enroll e
2 WHERE m.sid=s.sid and e.sid=s.sid GROUP BY s.sid, s.sname, m.dname
3 HAVING avg(grade) >= ALL (SELECT avg(grade) FROM major m1, enroll e1
4                           WHERE m1.sid=e1.sid and m1.dname=m.dname GROUP BY e1.sid);

```

dname	sname	maxgrade
Computer Sciences	Sulfate, Barry M.	4
Sanitary Engineering	Sulfate, Barry M.	4
Computer Sciences	Sather, Roberto B.	4
Computer Sciences	Stanley, Leotha T.	4
Computer Sciences	Bomber, C.	4
Computer Sciences	Carter, Jimmy	4
Chemical Engineering	Baker, C.	4
Civil Engineering	Moeri, S.	4
Civil Engineering	Micheal, Zadicki T.	4
Civil Engineering	Moomchi, B.	4
Civil Engineering	Atny, Mary H.	4
Industrial Engineering	Evert, Chris	4
Industrial Engineering	Connors, Jimmy	4
Mathematics	Mueller, D.	4
Mathematics	Ghandi, I.	4
Mathematics	Kirk, J.	4
Mathematics	Andermanthenol, K.	4
Mathematics	Taylor, R.	4

14. Find the sections (dname, cno, sectno) with the highest enrollment.

```

1 SELECT dname, cno, sectno, count(*) as enrolled FROM enroll e
2 GROUP BY dname, cno, sectno
3 HAVING count(*)
4 >= ALL (SELECT count(*) FROM enroll e1
5         GROUP BY e1.dname, e1.cno, e1.sectno);

```

dname	cno	sectno	enrolled
Computer Sciences	726	1	17

15. Find the department with more than 5 studnets.

```

1 SELECT dname, count(sid) FROM major
2 GROUP BY dname HAVING count(sid)>5;

```

dname	count(sid)
Chemical Engineering	11
Civil Engineering	28
Computer Sciences	24
Industrial Engineering	20
Mathematics	20

16. For all departments that offer the same number of courses, print the department name that has the least number of professors

```

1 SELECT s.dname, count(distinct s.cno) as numberofcourses,
2         count(distinct p.pname) as numberofprofs
3 FROM prof p, section s
4 WHERE p.dname=s.dname GROUP BY s.dname HAVING count(distinct p.pname)
5 <= ALL (SELECT count(distinct p1.pname) FROM prof p1, section s1
6         WHERE p1.dname=s1.dname
7         GROUP BY s1.dname HAVING count(distinct s1.cno) = count(distinct s.cno))
8 ORDER BY numberofcourses, numberofprofs;

```

dname	numberofcourses	numberofprofs
Chemical Engineering	1	1
Sanitary Engineering	1	1
Mathematics	2	1
Civil Engineering	3	3
Computer Sciences	4	2

Problem 2

1) Use a column constraint:

```
CREATE TABLE Emp(eid INTEGER,
                  ename VARCHAR(30),
                  age INTEGER,
                  salary FLOAT CHECK ( salary < 200000),
                  PRIMARY KEY (eid))
```

2) This is enforced by the referential integrity constraint when declaring managerid as a foreign key referencing back to the Emp table. However, when a DB server doesn't enforce foreign key constraint, you can also create an assertion as follows:

```
CREATE ASSERTION ManagerIsEmployee
CHECK (( SELECT COUNT (*)
        FROM Dept D
        WHERE D.managerid NOT IN (SELECT eid FROM Emp)) = 0)
```

3) Create an assertion as follows:

```
CREATE ASSERTION ManagerHigherSalary
CHECK ( NOT EXISTS (SELECT E.eid
                   FROM Emp E, Emp M, Works W, Dept D
                   WHERE E.eid = W.eid AND W.did = D.did
                   AND D.managerid = M.eid AND E.salary > M.salary))
```

4) Add an assertion as follows,

```
CREATE ASSERTION employee-appointment-constraint
CHECK (NOT EXISTS (
    SELECT Emp.eid
    FROM Emp, Works
    WHERE Emp.eid = Works.eid
    GROUP BY Emp.eid
    HAVING sum(pct_time) != 100))
```

5) Add an assertion as follows,

```
CREATE ASSERTION ManageConstraint
CHECK (NOT EXISTS (
    SELECT managerid
    FROM Dept
    GROUP BY managerid
    HAVING count(did) >2))
```