Solution to Homework 4

1. Write relational algebra expressions to answer the following query statements based on the tables used in Project 1. It is okay to answer a query in multiple steps. You need to make sure that your relational algebra expressions are reasonably optimized as we discussed in class, i.e., conditions involving a single table should be specified on that table directly and Cartesian products should be replaced by joins whenever possible.

   (a) (10 points) For each TA from the CS department, find his/her B#, first name and birth date.

   $\pi_{\text{TAs.B\# first\_name bdate}}$ (TAs ∞ ($\sigma_{\text{deptname = 'CS'}}$ (Students)))

   (b) (10 points) For each class that has a PhD student TA, find its classid, dept_code and course# as well as the TA's first name and email.

   $\pi_{\text{classid dept\_code course\# first\_name, email}}$ (classes ∞ $_{\text{ta\_B\# = tas.B\#}}$ ($\sigma_{\text{ta\_level = 'PhD'}}$ (tas) ∞ students))

   (c) (10 points) Find the B# and first name of each MS student who did not take any course.

   $\pi_{\text{B\# first\_name}}$(Students ∞ ($\pi_{\text{B\#}}$ ($\sigma_{\text{status = 'MS'}}$(Students)) - $\pi_{\text{B\#}}$ (Enrollments ∞ $\sigma_{\text{status = 'MS'}}$(Students))))

   (d) (10 points) Find the classid, dept_code and course# of each class that has been taken by all juniors.

   $\pi_{\text{classid dept\_code course\#}}$ (Classes ∞ ($\pi_{\text{B\# classid}}$ (Enrollments) ÷ $\pi_{\text{B\#}}$ ($\sigma_{\text{status = 'junior'}}$(Students))))

2. (20 points) Consider the two tables Courses and Classes used in Project 1. Test the following with real SQL statements in your Oracle account and report the result of each test case (including the SQL statements used).

   (a) Try to drop table Courses using "drop table courses;". Report what happened.

   (b) Try to drop table Courses using "drop table courses cascade constraints;"). Did the table Courses get dropped? Did any tuples from table Classes get deleted?

   (c) Following the statement in (b), analyze what happened to the foreign key (dept_code, course#) in Classes that references the primary key of Courses? Use SQL statement(s) and the results to explain your answer.

   Answer:
   (a) The drop statement failed because the primary key of Courses was referenced by a foreign key

```
SQL> drop table courses;
drop table courses
           *
ERROR at line 1:
ORA-02449: unique/primary keys in table referenced by foreign keys

SQL> select * from courses;

DEPT_CODE    COURSE# TITLE
--------- ---------- --------------------
CS              432 database systems
Math             314 discrete math
CS              240 data structure
Math             221 calculus I
CS              532 database systems
```

```
CS          552 operating systems
Biol        425 molecular biology

7 rows selected.
```

(b) This time the drop statement succeeded as the Courses was dropped successfully. We show the tuples in Classes before and after table Courses is dropped. It is clear that no tuple in Classes was affected.

```
SQL> select * from classes;

CLASSID DEPT_CODE    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE ROOM
TA_B
------- ---------- ---------- ---------- ---------- -------- ---------- ---------- ----
c0001   CS            432         1      2017 Spring          3         2 LH 005     B005
c0002   Math          314         1       2016 Fall           3         2 LH 009     B012
c0003   Math          314         2       2016 Fall           4         1 LH 009     B009
c0004   CS            432         1      2016 Spring          3         3 SW 222     B005
c0005   CS            240         1      2017 Spring          4         3 LH 003     B010
c0006   CS            532         1      2017 Spring          4         3 LH 005     B011
c0007   Math          221         1       2017 Spring         1         1 WH 155

7 rows selected.

SQL> drop table courses cascade constraints
  2  /

Table dropped.

SQL> select * from courses;
select * from courses
              *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> select * from classes;

CLASSID DEPT_CODE    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE ROOM
TA_B
------- ---------- ---------- ---------- ---------- -------- ---------- ---------- ----
c0001   CS            432         1      2017 Spring          3         2 LH 005     B005
c0002   Math          314         1       2016 Fall           3         2 LH 009     B012
c0003   Math          314         2       2016 Fall           4         1 LH 009     B009
c0004   CS            432         1      2016 Spring          3         3 SW 222     B005
c0005   CS            240         1      2017 Spring          4         3 LH 003     B010
c0006   CS            532         1      2017 Spring          4         3 LH 005     B011
c0007   Math          221         1       2017 Spring         1         1 WH 155

7 rows selected.

SQL> spool off
```

(c) The fact that no tuple in Classes was deleted as a result of dropping Courses indicates that the foreign key reference has been either dropped or disabled. If delete on cascade was implemented, all tuples in Classes would have been deleted as they all reference the tuples in Courses.

3. (20 points) The Referential Integrity Constraint says that any value under a foreign key must either be a null value or a value matching a value under the corresponding primary (candidate) key. When a foreign key FK in table R1 has two attributes (A, B), there are two possible ways to define the meaning of a null value: (i) both A and B have null values, i.e., (null, null) and (ii) at least one of the values under A and B is a null value, i.e., (null, b), (a, null) or (null, null), where a and b are non-null values. In Project 1, the foreign key (dept_code, course#) in table Classes has two attributes that reference the primary key of table Courses. Use the two tables and design experiments in Oracle to answer the following questions. Report your queries for each experiment.

   (a) Which of the above two meanings, i.e., (i) and (ii), is adopted by Oracle? Note that for this experiment, the schema for table Classes needs to be modified to allow null values for both dept_code and course# (just remove "not null" for them for this table in the script file for Project 1).

   (b) Is it possible to insert a tuple in table Classes such that its values under (dept_code, course#) have one null value (say under dept_code) and a non-null value (say under course#) and the non-value does not match any value under course# in table Courses?

Answer:

(a) First remove "not null" for dept_code and class# for Classes in proj1_tables_script18.sql and re-run the script to generate the tables. We first show the contents of the two tables. We then attempt to insert a tuple into Classes with dept_code = 'CS' and class# = 666. The insertion fails because 666 does not match any value under Courses.class#, indicating that foreign key constraint is being enforced. Next we attempted to insert three tuples with (dept_code, class#) being (null, null), ('CS', null) and (null, 888), respectively. All insertions succeeded, indicating that in Oracle, (null, null), (a, null) and (null, b) are all considered as null.

(b) Yes it is possible. We successfully inserted (null, 888) while 888 has no matching value under Courses.course#. In this case, no matter what value a we could use to replace the null value in (null, 888), (a, 888) will violate the referential integrity constraint. The fact that (null, 888) is allowed indicates that Oracle simply treated it as a null value and did not bother to check if 888 matches any value under Courses.course#.

SQL> select * from courses;

DEPT_CODE	COURSE#	TITLE
CS	432	database systems
Math	314	discrete math
CS	240	data structure
Math	221	calculus I
CS	532	database systems
CS	552	operating systems
Biol	425	molecular biology

7 rows selected.

SQL> select * from classes;

CLASSID	DEPT_CODE	COURSE#	SECT#	YEAR	SEMESTER	LIMIT	CLASS_SIZE	ROOM	TA_B
c0001	CS	432	1	2017	Spring	3	2	LH 005	B005
c0002	Math	314	1	2016	Fall	3	2	LH 009	B012
c0003	Math	314	2	2016	Fall	4	1	LH 009	B009
c0004	CS	432	1	2016	Spring	3	3	SW 222	B005

```
c0005   CS          240        1      2017 Spring         4        3 LH 003    B010
c0006   CS          532        1      2017 Spring         4        3 LH 005    B011
c0007   Math         221        1      2017 Spring         1        1 WH 155

7 rows selected.

SQL> insert into classes values ('c0008', 'CS', 666, 2, null, null, null, null, null, null);
insert into classes values ('c0008', 'CS', 666, 2, null, null, null, null, null, null)
*
ERROR at line 1:
ORA-02291: integrity constraint (MENG.SYS_C00164824) violated - parent key not found


SQL> insert into classes values ('c0009', null, null, 1, 2018, null, null, null, null, null);

1 row created.

SQL> insert into classes values ('c0010', 'CS', null, 2, 2018, null, null, null, null, null);

1 row created.

SQL> insert into classes values ('c0011', null, 888, 1, 2018, null, null, null, null, null);

1 row created.

SQL> select * from classes;

CLASSID DEPT_CODE    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE ROOM
TA_B
------- --------- ---------- ---------- ---------- -------- ---------- ---------- ---------- ----
c0001   CS          432        1      2017 Spring         3        2 LH 005    B005
c0002   Math         314        1       2016 Fall          3        2 LH 009    B012
c0003   Math         314        2       2016 Fall          4        1 LH 009    B009
c0004   CS          432        1      2016 Spring         3        3 SW 222    B005
c0005   CS          240        1      2017 Spring         4        3 LH 003    B010
c0006   CS          532        1      2017 Spring         4        3 LH 005    B011
c0007   Math         221        1      2017 Spring         1        1 WH 155
c0009                          1      2018
c0010   CS                     2      2018
c0011               888        1      2018

10 rows selected.

SQL> spool off
```

4. (20 points) Create a simple view CSCourses on table Courses from Project 1 in your Oracle account. CSCourses logically contains only CS courses from Courses.

   (a) Try to insert a tuple into, delete a tuple from, and update a value of Courses through CSCourses.

   (b) Try to insert a new math course into Course via CSCourses. Note that this new course does not logically belong to CSCourses. Report whether you succeeded.

   Save your work in one or more spool files and hand in the files.

Answer:

```
SQL> select * from courses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS            432 database systems
Math           314 discrete math
CS            240 data structure
Math           221 calculus I
CS            532 database systems
CS            552 operating systems
Biol          425 molecular biology

7 rows selected.

SQL> create view CSCourses as select * from Courses where dept_code = 'CS';

View created.

SQL> select * from CSCourses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS            432 database systems
CS            240 data structure
CS            532 database systems
CS            552 operating systems
```

(a) Insert into, delete from and update the view. The changes can also be seen in the base table Courses.

```
SQL> insert into CSCourses values ('CS', 666, 'simple view');

1 row created.

SQL> select * from CSCourses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS            432 database systems
CS            240 data structure
CS            532 database systems
CS            552 operating systems
CS            666 simple view

SQL> select * from courses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS            432 database systems
Math           314 discrete math
CS            240 data structure
Math           221 calculus I
CS            532 database systems
```

```
CS          552 operating systems
Biol        425 molecular biology
CS          666 simple view

8 rows selected.

SQL> update CSCourses set title = 'data analytics' where course# = 240;

1 row updated.

SQL> select * from CSCourses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS          432 database systems
CS          240 data analytics
CS          532 database systems
CS          552 operating systems
CS          666 simple view

SQL> select * from courses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS          432 database systems
Math         314 discrete math
CS          240 data analytics
Math         221 calculus I
CS          532 database systems
CS          552 operating systems
Biol        425 molecular biology
CS          666 simple view

8 rows selected.

SQL> delete from CSCourses where course# = 666;

1 row deleted.

SQL> select * from CSCourses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS          432 database systems
CS          240 data analytics
CS          532 database systems
CS          552 operating systems

SQL> select * from courses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS          432 database systems
Math         314 discrete math
```

```
CS          240 data analytics
Math         221 calculus I
CS          532 database systems
CS          552 operating systems
Biol        425 molecular biology

7 rows selected.
```

(b) Successfully inserted ('Math', 555, 'view insertion') into table courses through the view! Note that this tuple is not viewable from the view CSCourses.

```
SQL> insert into CSCourses values ('Math', 555, 'view insertion');

1 row created.

SQL> select * from CSCourses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS          432 database systems
CS          240 data analytics
CS          532 database systems
CS          552 operating systems

SQL> select * from courses;

DEPT_CODE    COURSE# TITLE
---------- ---------- --------------------
CS          432 database systems
Math         314 discrete math
CS          240 data analytics
Math         221 calculus I
CS          532 database systems
CS          552 operating systems
Biol        425 molecular biology
Math         555 view insertion

8 rows selected.

SQL> spool off
```