```
/* BuchananBrandonHW4.sql
-------------------------------------------------------------------------
Programmer's Name: Brandon Buchanan
Course: CSCI2020                          Section Number: 002
Creation Date: 10/22/2011        Date of Last Modification: 10/23/2011
E-mail address: zbwb8@goldmail.etsu.edu
-------------------------------------------------------------------------
Purpose -
 Homework 4
-------------------------------------------------------------------------
Notes on specifications, special algorithms, and assumptions:
Designed to work on the database created by the schooldb.sql
script
*/

/* 1. -  Display the highest paid instructor that teaches in Nicks.  The inner
         query finds the highest paid salary and the outer query finds every
         Nicks professor that also matches the highest paid salary returned
         from the inner query.*/

select INSTRUCTOR.FAMILY_NAME as HIGHEST_PAID
  from INSTRUCTOR
  join DEPARTMENT
    on INSTRUCTOR.DEPARTMENT = DEPARTMENT.NAME
  where INSTRUCTOR.TITLE = 'Professor'
    and DEPARTMENT.BUILDING = 'Nicks'
    and INSTRUCTOR.SALARY = (
    select max(INSTRUCTOR.SALARY) from INSTRUCTOR);


/* 2. - Displays a list of instructors that teach in the earliest founded
        department.  The inner query finds the earliest founded department
        while the outer query displays the instructor's name for those that
        work in the earliest department.  */

select INSTRUCTOR.GIVEN_NAME, INSTRUCTOR.FAMILY_NAME
  from INSTRUCTOR
  join DEPARTMENT
    on INSTRUCTOR.DEPARTMENT = DEPARTMENT.NAME
    where DEPARTMENT.FOUNDED = (
    select min(DEPARTMENT.FOUNDED) from DEPARTMENT);

/* 3. -  Displays students who took a class during the term with the lowest
         credit hour fees.  The last query finds the lowest credit hour fee
         term.  The second query finds the term code for the lowest credit
         hour fee.  The first query displays the student names for the
         lowest credit hour fee code.*/

select distinct STUDENT.GIVEN_NAME, STUDENT.FAMILY_NAME
  from STUDENT
  join TAKES
    on STUDENT.ID = TAKES.STUDENT
```

96

Do not use distinct unless called for - no point deduction

given_Name = 4 The inner query does not match on Professor and Nicks

indentation

```
  where TAKES.TERM = (

select TERM.CODE
  from TERM
  where TERM.CREDIT_HOUR_FEE = (

select min(TERM.CREDIT_HOUR_FEE)
  from TERM
));

/* 4. -  Display the student ID and major for those in a major with at least
         10 students.  The first query displays the ID and major in each case
         where a student's major has at least 10 students with the major, found
         by the second query.  */

select STUDENT.ID, STUDENT.MAJOR
  from STUDENT
  where STUDENT.MAJOR in (

select STUDENT.MAJOR
  from STUDENT
  group by STUDENT.MAJOR
  having count(STUDENT.MAJOR) >= 10
);

/* 5. -  Displays a when each department was founded in the 1950s and
         when the department was founded.  The query formats the output
         according to the month spelled out, two digit day, and four digit
         year.  */

select 'The ' || DEPARTMENT.NAME || ' department was founded on ' ||
       to_char(DEPARTMENT.FOUNDED, 'FMMONTH') || ' ' ||
       to_char(DEPARTMENT.FOUNDED, 'DD') || ', ' ||
       to_char(DEPARTMENT.FOUNDED, 'YYYY')
  from DEPARTMENT
  where extract(year from DEPARTMENT.FOUNDED) >= 1950 and
        extract(year from DEPARTMENT.FOUNDED) < 1960;

/* 6. -  Displays the date 3 months from today.  This query adds 3 months to
         the current date and then displays that in the format of 'YYYY-MM-DD'*/

select to_char(sysdate + numtoyminterval(3,'MONTH'),'YYYY-MM-DD') as Expiration
  from dual ;

/* 7. - Display terms that begin in August with the number of sections scheduled
         for that term.  This query displays the term ID and number of sections
         scheduled for that term where the month is equal to the month's number
         (8 for August) and when there are at least 23 sections scheduled.  */

select SECTION.TERM, count(SECTION.TERM) as NUM_OF_SECTIONS
  from SECTION
  where extract(month from START_DATE)=8
```

```
  having count(SECTION.TERM) >= 23
  group by SECTION.TERM;

/* 8. -  Displays the highest, lowest, and average ACT scores rounded to
         nearest integer using the round function along with max, min, and
         avg.  */

select round(max(STUDENT.ACT_SCORE)) as HIGHEST,
       round(min(STUDENT.ACT_SCORE)) as LOWEST,
       round(avg(STUDENT.ACT_SCORE)) as AVERAGE
  from STUDENT;

/* 9. -  Display the first initial and last name of students that attended an
         Academy who also have a higher than average GPA among all students.
         The inner query finds the average GPA of all students.  The outer
         query searches on all Academy and where a student's GPA is higher
         than average.  */

select substr(STUDENT.GIVEN_NAME,1,1) || ' ' || STUDENT.FAMILY_NAME
  from STUDENT
  where STUDENT.HIGH_SCHOOL like '%Academy'
    and STUDENT.HS_GPA > (

select avg(STUDENT.HS_GPA) from STUDENT
);
```

follow indentation properly and ~~correctly~~ consistently.