// Question 1
SELECT CONCAT(s1.Fname, " ", s1.Lname) as Name, CONCAT(s2.Fname, " ", s2.Lname) as Name
FROM Student s1, Student s2
WHERE s1.StuID <> s2.StuID AND
  s1.city_code = s2.city_code AND s1.StuID < s2.StuID AND
  (s1.StuID, s2.StuID) IN (SELECT * FROM Likes) AND
  (s2.StuID, s1.StuID) IN (SELECT * FROM Likes) AND
  (s1.StuID, s2.StuID) NOT IN (SELECT * FROM Loves) OR
  (s2.StuID, s1.StuID) NOT IN (SELECT * FROM Loves));

// Question 2
SELECT CONCAT(Fname, " ", Lname) as Name, CarManufacturer, CarModel, miles_per_gallon
FROM Car, Car_Ownership, Student
WHERE Car.CarID = Car_Ownership.CarID AND
  Student.StuID = Car_Ownership.StuID AND
  miles_per_gallon = (SELECT MIN(miles_per_gallon) FROM Car);

// Question 3
SELECT DISTINCT CONCAT(Fname, " ", Lname) as Name, Age, Major
FROM Car_Ownership, Student
WHERE NOT EXISTS ((
  SELECT CarID
  FROM Car
  WHERE CarManufacturer = "Nissan")
  EXCEPT (
  SELECT CarID
  FROM Car_Ownership
  WHERE Student.StuID = Car_Ownership.StuID));

// Question 4
SELECT CONCAT(Fname, " ", Lname) as Name
FROM Student
WHERE StuID IN
  (SELECT stuid
   FROM Lives_in
   INTERSECT
   (SELECT DISTINCT c1.StuID
    FROM Car_Ownership as c1, Car_Ownership as c2

// Question 5
SELECT CONCAT(Fname, " ", Lname) as Name
FROM Student
WHERE StuID IN (SELECT stuid FROM Lives_in) AND
   StuID NOT IN (SELECT StuID FROM Car_Ownership) AND
   StuID IN (SELECT StuID FROM Has_Pet);

// Question 6
SELECT *
FROM Student
WHERE StuID IN (SELECT stuid FROM Lives_in) AND
   StuID IN (SELECT StuID
             FROM Car_Ownership
             GROUP BY(StuID)
             HAVING COUNT(CarID) = 2) AND
   StuID IN (SELECT StuID
             FROM Has_Pet
             GROUP BY(PetID)
             HAVING COUNT(PetID) >= 2);

// Question 7
select min(miles_per_gallon), max(miles_per_gallon), avg(miles_per_gallon) from Car where
CarManufacturer="Porsche";

// Question 8
select min(age), max(age), avg(age) from Student, Lives_in where Student.StuID = Lives_in.stuid
and Student.StuID not in (select StuID from Car_Ownership);

// Question 9
select Fname, Lname, Age from Student where StuID not in (select stuid from Lives_in);

// Question 10
select avg(Age) from (select Age, count(*) as c from Participates_in, Student where Student.S
   tuID = Participates_in.stuid group by Student.StuID having c > 2) as a;

// Question 11
select C.c, Activity.activity_name from (select max(c) as m from (select actid, count(*) as c from
Participates_in group by actid) as A) as B, (select actid, count(*) as c from Participates_in group
by actid) as C, Activity where C.c = B.m and Activity.actid = C.actid;
Question 12
select distinct activity_name from Faculty_Participates_in, Activity where
Faculty_Participates_in.actid not in (select distinct actid from Participates_in) and Activity.actid =
Faculty_Participates_in.actid;

Question 13
SELECT distinct CONCAT(s.Fname, " ", s.Lname) as Name
From Student as s, Enrolled_in as e
Where
e.StuID = s.StuID and
e.CID in (
    Select e.CID
    FROM Student as s, Enrolled_in as e, Lives_in as l
    Where
e.StuID = s.StuID and
s.stuID = l.StuID and
(l.DormID, l.Room_Number) in (
    Select l.DormID, l.Room_number
    FROM
        VotedForElectioninUS as v,
        USCandidate as u,
        Student as s,
        City as c,
        Lives_in as l
    WHERE
        CandidateName = 'Donald Trump' and
s.StuID = v.StuID and
s.city_code = c.city_code and
c.state = "PA" and
l.StuID = s.StuID
);

Question 14
Select Distinct Concat(s1.Fname, " ", s1.LName) as Student Name, Concat(f1.Fname, " ",
f1.LName)
From
    Student as s1, Faculty as f1, Faculty_Participates_in as fp1, Participates_in as p1,
Where
s1.Advisor = f1.FacID and
p1.StuID = s1.StuID and
fp1.FacID = f1.FacID and
p1.ActID = fp1.ActID and
f1.FacID in (select Instructor from Course);
// Question 15
SELECT distinct Concat(s1.Fname, " ", s1.Lname) as Name1, Concat(s2.Fname, " ", s2.Lname) as Name2
FROM Student as s1, Student as s2, Lives_in as l1, Lives_in as l2, City as c1, City as c2
Where
  l1.Room_number = l2.Room_number and
  l1.stuID != l2.stuID and
  l1.StuID = s1.stuID and
  l1.stuID < l2.stuID and
  l2.stuID = s2.stuID and
  s1.City_Code = c1.City_Code and
  s2.City_Code = c2.City_Code and

// Question 16
Select dormid, Dorm_Name, GPA
From (
  Select d.dormid, d.Dorm_Name, sum(g.gradepoint * c.Credits) / sum(c.Credits) as GPA
  From Enrolled_in as e, Course as c, Lives_in as l, Gradeconversion as g, Dorm as d
  Where
    e.CID = c.CID and
    e.Grade = g.lettergrade and
    e.StuID = l.stuid and
    l.dormid = d.dormid
  Group by d.dormid
  Order by GPA desc
) as temp
having GPA = max(GPA);

// Question 17
Part 1:
Drop table if exists Baltimore_Distance;
Create table Baltimore_Distance (  
city1_code varchar(3),
city2_code varchar(3),
distance int(11)
);
Insert into Baltimore_Distance
Select city1_code, city2_code, NULL
From Direct_distance
Where city1_code != city2_code;
Insert into Baltimore_Distance
Select city1_code, city2_code, 0
From Direct_distance
Where city1_code = city2_code;

Update Baltimore_Distance as b, Direct_distance as d
Set b.distance = d.distance
Where
  b.city1_code = d.city1_code and
  b.city2_code = d.city2_code and
  b.city1_code = "BAL";

Update Baltimore_Distance as b, Direct_distance as d
Set b.distance = d.distance
Where
  b.city1_code = d.city1_code and
  b.city2_code = d.city2_code and
  b.city2_code = "BAL";

Update
  Baltimore_Distance as b,
  Direct_distance as d1,
  Direct_distance as d2,
  Direct_distance as d3
Set b.distance = d2.distance + d3.distance
Where
  b.city1_code = d1.city1_code and
  b.city2_code = d1.city2_code and
  d1.city1_code != 'BAL' and
  d1.city2_code != 'BAL' and
  d1.city1_code != d1.city2_code and
  d2.city1_code != 'BAL' and
  d2.city2_code = 'BAL' and
  d2.city1_code = d1.city1_code and
  d3.city1_code = d1.city2_code and
  d3.city1_code != 'BAL' and
  d3.city2_code = 'BAL';

Part 2:
Drop table if exists Rectangular_Distance;
Create table Rectangular_Distance (  
city1_code      varchar(3),
city2_code      varchar(3),
)
distance int(11)
);

Insert into Rectangular_Distance
Select c1.city_code, c2.city_code, sqrt( pow( (70 * c1.latitude) - (70 * c2.latitude), 2 ) + pow( (70 *
c1.longitude) - (70 * c2.longitude), 2 ) )
From City as c1, City as c2
Where c1.city_code != c2.city_code;

Insert into Rectangular_Distance
Select c1.city_code, c2.city_code, 0
From City as c1, City as c2
Where c1.city_code = c2.city_code;

Part 3:
Drop table if exists All_Distances;
Create table All_Distances (city1_code varchar(3),
city2_code varchar(3),
direct_distance int(11),
baltimore_distance int(11),
rectangular_distance int(11)
);

Insert into All_Distances
Select r.city1_code, r.city2_code, NULL, NULL, r.distance
From Rectangular_Distance as r;

Update All_Distances as a, Baltimore_Distance as b
Set a.baltimore_distance = b.distance
Where b.city1_code = a.city1_code and b.city2_code = a.city2_code;

Update All_Distances as a, Direct_distance as d
Set a.direct_distance = d.distance
Where d.city1_code = a.city1_code and d.city2_code = a.city2_code;

Part 4:
Drop table if exists Best_Distance;
Create table Best_Distance (city1_code varchar(3),
city2_code varchar(3),
distance int(11))
);
Insert into Best_Distance
Select city1_code, city2_code,
   Least (  
       IFNULL (direct_distance, rectangular_distance),  
       IFNULL (baltimore_distance, rectangular_distance),  
       rectangular_distance 
   )
From All_Distances;

// Question 18
Select c.City_name, Count(s.StuID)
From Student as s, City as c
Where s.City_Code = c.City_Code
Group by s.City_Code
Having count(s.StuID) > 1;

// Question 19
select * from (select Dorm.dormid,Student.StuID,city_code from Dorm, Lives_in, Student where  
Dorm.dormid = Lives_in.dormid and student_capacity < 300 and Student.StuID = Lives_in.stuid) as A,(select Dorm.dormid,Student.StuID,city_code from Dorm, Lives_in, Student where  
Dorm.dormid = Lives_in.dormid and student_capacity < 300 and Student.StuID = Lives_in.stuid) as B,Best_Distance, City c1, City c2 where A.dormid = B.dormid and A.StuID != B.StuID and  
A.city_code = city1_code and B.city_code = city2_code and distance < 100 and A.StuID <  
B.StuID and A.city_code = c1.city_code and B.city_code = c2.city_code;

// Question 20
select C.country, StuID,Fname,Lname from (select City.city_code, country, distance from (select  
City.city_code from City where city_name = "Baltimore") as A, City, Best_Distance where  
city1_code = A.city_code and City.city_code = city2_code) as B, (select country, max(distance)  
as d from (select city_code from City where city_name = "Baltimore") as A, City, Best_Distance  
where city1_code = A.city_code and City.city_code = city2_code group by country) as C,  
Student where C.country = B.country and C.d = B.distance and Student.city_code =  
B.city_code;

// Question 21
Select Activity_name
From Activity
Where actid in (  
Select a.actid,  
From Participates_in as p, Activity as a, Student as s, Best_Distance as b  
Where  
a.actid = p.actid and
s.StuID = p.stuid and
b.city1_code = 'BAL' and
b.city2_code = s.city_code

Group by p.City

// Question 22
SELECT Student.Fname, Student.LName, Student.Age
FROM
  Student,
  Minor_in,
  Department AS StudentDepartment,
  Enrolled_in,
  Course,
  Faculty,
  Department AS FacultyDepartment,
  Member_of
WHERE
  Student.Sex='F' AND
  Student.StuID = Minor_in.StuID AND
  Minor_in.DNO = StudentDepartment.DNO AND
  StudentDepartment.Division = 'EN' AND
  Enrolled_in.StuID = Student.StuID AND
  Course.CID = Enrolled_in.CID AND
  Course.Instructor = Faculty.FacID AND
  Faculty.Building = FacultyDepartment.Building AND
  FacultyDepartment.Division = 'EN' AND Member_of.FacID = Faculty.FacID AND
  Appt_Type = "Primary" AND Member_of.DNO = FacultyDepartment.DNO AND
  FacultyDepartment.Division='EN';

// Question 23
SELECT Student.Fname, Student.LName, Student.StuID
FROM
  Student,
  Enrolled_in,
  Course,
  Faculty
WHERE
  Enrolled_in.StuID = Student.StuID AND
  Course.CID = Enrolled_in.CID AND
  Course.Instructor = Faculty.FacID AND
  Faculty.Fname = "Paul" AND
  Faculty.Lname = "Smolensky"
GROUP BY
HAVING COUNT(*) = ( 
    SELECT COUNT(*)
    FROM Course, Faculty
    WHERE
        Course.Instructor = Faculty.FacID AND
        Faculty.Fname = "Paul" AND
        Faculty.Lname = "Smolensky"
)
;
// Question 24
SELECT Student.Fname, Student.LName, Student.StuID
FROM
    Student,
    Enrolled_in,
    Student as OtherStudent,
    Enrolled_in as OtherEnrolledIn,
    Enrolled_in as OtherEnrolledInTwo,
    Student as LindaStudent,
    Enrolled_in as LindaEnrolledIn,
    City as OtherStudentCity,
    City as LindaCity,
    VotedForElectioninUS as OtherStudentVotedTwentySixteen,
    VotedForElectioninUS as OtherStudentVotedTwentyTwenty,
    VotedForElectioninUS as LindaVotedTwentySixteen,
    VotedForElectioninUS as LindaVotedTwentyTwenty
WHERE
    Enrolled_in.StuID = Student.StuID AND
    Enrolled_in.CID = OtherEnrolledIn.CID AND
    OtherEnrolledIn.StuID = OtherStudent.StuID AND
    OtherStudent.StuID = OtherEnrolledInTwo.StuID AND
    OtherEnrolledInTwo.CID = LindaEnrolledIn.CID AND
    LindaStudent.StuID = LindaEnrolledIn.StuID AND
    LindaStudent.Fname = "Linda" AND
    LindaStudent.Lname = "Smith" AND
    OtherStudent.city_code = OtherStudentCity.city_code AND
    LindaStudent.city_code = LindaCity.city_code AND
    OtherStudentCity.state = LindaCity.state AND
    OtherStudentVotedTwentySixteen.Year = 2016 AND
    OtherStudentVotedTwentyTwenty.Year = 2020 AND
    LindaVotedTwentySixteen.Year = 2016 AND
    LindaVotedTwentyTwenty.Year = 2020 AND
    OtherStudentVotedTwentySixteen.StuID = OtherStudent.StuID AND
    OtherStudentVotedTwentyTwenty.StuID = OtherStudent.StuID AND
    LindaVotedTwentySixteen.StuID = OtherStudent.StuID AND
    LindaVotedTwentyTwenty.StuID = OtherStudent.StuID AND
    OtherStudentVotedTwentySixteen.CandidateID = LindaVotedTwentySixteen.CandidateID AND
    OtherStudentVotedTwentyTwenty.CandidateID = LindaVotedTwentyTwenty.CandidateID
// Question 25
SELECT Course.CName
FROM
    Course,
    Enrolled_in,
    Student
WHERE
    Course.CID = Enrolled_in.CID AND
    Enrolled_in.StuID = Student.StuID AND
    0 = (SELECT COUNT(*) FROM Member_of_club WHERE Member_of_club.StuID = Student.StuID) AND
    0 = (SELECT COUNT(*) FROM Has_Allergy WHERE Has_Allergy.StuID = Student.StuID) AND
    NOT 0 = (SELECT COUNT(*) FROM Likes, Member_of_club, Has_Allergy
        WHERE
            Likes.WhoLikes = Student.StuID AND
            Likes.WhoIsLiked = Member_of_club.StuID AND
            Likes.WhoIsLiked = Has_Allergy.StuID)

// Question 26
SELECT Student.Fname, Student.Lname, Dorm.dorm_name, COUNT(*)
FROM
    Student,
    Lives_in,
    Dorm,
    ConductViolation
WHERE
    Student.StuID = Lives_in.stuid AND
    Lives_in.dormid = Dorm.dormid AND
    Student.StuID = ConductViolation.StuID
GROUP BY
    Student.StuID

// Question 27
SELECT Student.Fname, Student.Lname, Dorm.dorm_name, COUNT(*)
FROM
    Student,
    Lives_in,
    Dorm,
    ConductViolation
WHERE
    Student.StuID = Lives_in.stuid AND
    Lives_in.dormid = Dorm.dormid AND
    Student.StuID = ConductViolation.StuID
GROUP BY
    Student.StuID
HAVING
    COUNT(*) = (SELECT COUNT(*)
                FROM Student, ConductViolation
                WHERE Student.StuID = ConductViolation.StuID
                GROUP BY Student.StuID
                LIMIT 1
                ORDER BY COUNT(*) DESC)
;

// Question 28
SELECT D.dorm_name, D.student_capacity, amenity_name
FROM
    (SELECT Dorm.dormid, Dorm.dorm_name, Dorm.student_capacity
     FROM Dorm, ConductViolation
     WHERE
         Dorm.dormid = ConductViolation.Dormid AND
         ConductViolation.Reason = "ALCOHOL"
     GROUP BY
         Dorm.dormid
     HAVING
         Count(*) = (select max(a.c) from
                     (SELECT COUNT(*) as c
                      FROM ConductViolation
                      WHERE ConductViolation.Reason = "ALCOHOL"
                      GROUP BY ConductViolation.dormid
                     ) AS a
                    )
    ) D,
    Has_amenity,
    Dorm_amenity
WHERE
    D.dormid = Has_amenity.dormid AND
    Has_amenity.amenid = Dorm_amenity.amenid
;
// Question 29
SELECT Course.CName, Department.DName
FROM
  Student,
  Enrolled_in,
  Course,
  Department,
  ConductViolation
WHERE
  Student.StuID = Enrolled_in.StuID AND
  Enrolled_in.CID = Course.CID AND
  Course.DNO = Department.DNO AND
  ConductViolation.StuID = Student.StuID
GROUP BY
  Enrolled_in.CID
HAVING
  Count(*) = ( SELECT a.c
               FROM
                ( SELECT COUNT(*) as c
                  FROM
                    Student,
                    Enrolled_in,
                    ConductViolation
                  WHERE
                    Student.StuID = Enrolled_in.StuID AND
                    ConductViolation.StuID = Student.StuID
                  GROUP BY
                    Enrolled_in.CID
                ) as a
               LIMIT 1
          ORDER BY a.c DESC)
;

/* Question 30 */
SELECT activity_name
FROM Activity,
  (SELECT Participates_in.actid, ConductViolation.Dormid, ConductViolation.Date,
       COUNT(actid) as count
    FROM Student, Participates_in, ConductViolation
    WHERE Student.StuID = Participates_in.stuid
    AND Student.StuID = ConductViolation.StuID
    GROUP BY
      Participates_in.actid
  ) as a
WHERE a.count = 1
ORDER BY a.activity_name DESC
;
GROUP BY actid
ORDER BY actid) AS A
WHERE A.actid = Activity.actid AND A.count > 3;

/* Question 31 */
SELECT K.FName, K.LName, K.stuid, J.FName, J.LName, J.stuid
FROM
(SELECT Student.FName, Student.LName, Lives_in.stuid, Lives_in.dormid,
Lives_in.room_number, A.CandidateId, A.CandidateName
FROM Lives_in, Student,
(SELECT VotedForElectioninUS.StuID, VotedForElectioninUS.Year,
USCandidate.CandidateID, USCandidate.CandidateName, USCandidate.Party
FROM VotedForElectioninUS, USCandidate
WHERE VotedForElectioninUS.CandidateID = USCandidate.CandidateID
AND VotedForElectioninUS.Year = '2020') as A
WHERE Student.StuID = Lives_in.stuid and Student.StuID = A.StuID) as K,
(SELECT Student.FName, Student.LName, Lives_in.stuid, Lives_in.dormid,
Lives_in.room_number, A.CandidateId, A.CandidateName
FROM Lives_in, Student,
(SELECT VotedForElectioninUS.StuID, VotedForElectioninUS.Year,
USCandidate.CandidateID, USCandidate.CandidateName, USCandidate.Party
FROM VotedForElectioninUS, USCandidate
WHERE VotedForElectioninUS.CandidateID = USCandidate.CandidateID
AND VotedForElectioninUS.Year = '2020') as A
WHERE Student.StuID = Lives_in.stuid and Student.StuID = A.StuID) as J
WHERE K.dormid = J.dormid and K.room_number = J.room_number and K.stuid < J.stuid and
K.CandidateID != J.CandidateID;

/* Question 32 */
SELECT dorm_name
FROM Dorm, (SELECT MAX(DormVoterCount) as max
FROM (SELECT Student.StuID, Lives_in.dormid, COUNT(dormid) as DormVoterCount
FROM Student, VotedForElectioninUS, Lives_in, USCandidate
WHERE Student.StuID = VotedForElectioninUS.StuID
AND Student.StuID = Lives_in.stuid
AND USCandidate.CandidateName = 'Donald Trump'
AND VotedForElectioninUS.year = '2020'
AND VotedForElectioninUS.CandidateID = USCandidate.CandidateID
GROUP BY Lives_in.dormid ORDER BY Lives_in.dormid ASC) AS A) AS B,
(SELECT Student.StuID, Lives_in.dormid, COUNT(dormid) as DormVoterCount
FROM Student, VotedForElectioninUS, Lives_in, USCandidate
WHERE Student.StuID = VotedForElectioninUS.StuID
AND Student.StuID = Lives_in.stuid
AND USCandidate.CandidateName = 'Donald Trump'
AND VotedForElectioninUS.Year = '2020'
AND VotedForElectioninUS.CandidateID = USCandidate.CandidateID
GROUP BY Lives_in.dormid ORDER BY Lives_in.dormid ASC) AS C
WHERE B.max = C.DormVoterCount
AND Dorm.dormid = C.dormid;

/* Question 33 */
SELECT J.dormid, J.trump_voter_count, J.num_students_in_dorm,
MAX(J.trump_voter_count/J.num_students_in_dorm) AS percentage_trump_voters
FROM
(SELECT VotedForElectioninUS.StuID, Lives_in.dormid, COUNT(Lives_in.dormid) as trump_voter_count, A.num_students_in_dorm
FROM VotedForElectioninUS, Lives_in, USCandidate,
(SELECT dormid, count(dormid) as num_students_in_dorm
FROM Lives_in
GROUP BY dormid) AS A
WHERE VotedForElectioninUS.StuID = Lives_in.stuid
AND USCandidate.CandidateName = 'Donald Trump'
AND USCandidate.CandidateID = VotedForElectioninUS.CandidateID
AND VotedForElectioninUS.Year = '2020'
AND Lives_in.dormid = A.dormid
GROUP BY Lives_in.dormid) AS J;

/* Question 34 */
SELECT LName, FName, Age, J.2020CandidateName, J.2020Party, J.2016CandidateName,
J.2016Party
FROM Student
JOIN
(SELECT A.StuID, A.CandidateID as '2020CandidateID', A.CandidateName as '2020CandidateName', A.Party as '2020Party', B.CandidateID as '2016Candidate',
B.CandidateName as '2016CandidateName', B.Party as '2016Party'
FROM
(SELECT StuID, VotedForElectioninUS.CandidateID,
USCandidate.CandidateName, USCandidate.Party, VotedForElectioninUS.Year
FROM VotedForElectioninUS, USCandidate
WHERE Year = '2020' AND VotedForElectioninUS.CandidateID =
USCandidate.CandidateID) AS A,
(SELECT StuID, VotedForElectioninUS.CandidateID,
USCandidate.CandidateName, USCandidate.Party, VotedForElectioninUS.Year
FROM VotedForElectioninUS, USCandidate
WHERE Year = '2016' AND VotedForElectioninUS.CandidateID =
USCandidate.CandidateID) AS B
WHERE A.StuID = B.StuID AND A.CandidateID != B.CandidateID) AS J
WHERE J.StuID = Student.StuID;

/* Question 35 */
SELECT DISTINCT LName, FName, state as 'HomeState'
FROM
  Student, City,
  (SELECT StuID, VotedForElectioninUS.CandidateID, CandidateName, Party, Year
   FROM VotedForElectioninUS
   JOIN USCandidate
   WHERE USCandidate.CandidateID = VotedForElectioninUS.CandidateID) AS A,
  (SELECT StuID, VotedForElectioninUS.CandidateID, CandidateName, Party, Year
   FROM VotedForElectioninUS
   JOIN USCandidate
   WHERE USCandidate.CandidateID = VotedForElectioninUS.CandidateID) AS B
WHERE A.Year != B.Year
  AND A.Party != B.Party
  AND A.StuID = B.StuID
  AND A.StuID = Student.StuID
  AND City.city_code = Student.city_code;

/* Question 36 */
SELECT Student.FName, Student.LName
FROM Studied_Abroad, Student,
  (SELECT *
   FROM Worked_at
   WHERE Position LIKE '%Intern%') AS A
WHERE Studied_Abroad.StuID = A.StuID
  AND Student.StuID = Studied_Abroad.StuID
  AND (Studied_Abroad.End_Date >= A.Start_Date)
    OR (Studied_Abroad.Start_Date <= A.End_Date)
    OR (Studied_Abroad.Start_Date <= A.Start_Date AND Studied_Abroad.End_Date >= A.End_Date)
    OR (A.Start_Date <= Studied_Abroad.Start_Date AND A.End_Date >= Studied_Abroad.End_Date);

/* Question 37 */
SELECT Student.FName, Student.LName
FROM Student,
  (SELECT *
   FROM Worked_at
   WHERE Position LIKE '%Intern%') AS A,
  (SELECT *
/* Question 38 */
SELECT DISTINCT LName, FName, country
FROM City,
(SELECT DISTINCT A.StuID, Location, LName, FName, City.city_code
FROM City,
(SELECT DISTINCT Student.StuID, Location, LName, FName, city_code
FROM Studied_Abroad
INNER JOIN Student
WHERE Student.StuID = Studied_Abroad.StuID) AS A
WHERE City.city_code = A.city_code AND UPPER(B.Location) = country;

/* Question 39 */
SELECT * FROM Worked_at, Student, Department
WHERE Student.StuID = Worked_at.StuID AND Student.Major = Department.DNO AND Department.Dname = "Computer Science";

/* Question 40 */
SELECT *, DATEDIFF(End_Date, Start_Date) FROM Worked_at, Student
WHERE Student.StuID = Worked_at.StuID AND Position like "%Intern%";

/* Question 41 */
SELECT msdd, FName, LName, Company
FROM (SELECT max(B.sdd) as msdd
FROM (SELECT A.StuID, sum(dd) as sdd
FROM (SELECT Student.StuID, Company, DATEDIFF(End_date, Start_Date) as dd
FROM Worked_at, Student
WHERE Student.StuID = Worked_at.StuID AND Position like "%Intern%") as A
GROUP BY Company, A.StuID) as B
GROUP BY Company)
AS C, (SELECT * FROM (SELECT A.StuID, sum(dd) as sdd
FROM (SELECT Student.StuID, Company, DATEDIFF(End_date, Start_Date) as dd
FROM Worked_at, Student
WHERE Student.StuID = Worked_at.StuID AND Position like "%Intern%") as A
GROUP BY Company, A.StuID) as B) as D,
Student, Worked_at
WHERE msdd = sdd AND Student.stuid = D.StuID AND Worked_at.StuID = W
tuID = D.StuID and Student.StuID = Worked_at.StuID;

/*Question 42*/
select * from Dorm NATURAL JOIN Lives_in NATURAL Join Student NATURAL JOIN Has_Pet
NATURAL JOIN Pet NATURAL JOIN Has_Allergy where AllergyName = PetType;

/*Question 43*/
select * from (select * from (select L1.WhoLoves,L1.WhoIsLoved from Loves L1, Loves L2
Lives_in on A.WhoLoves = Lives_in.StuID) as B JOIN Lives_in on Lives_in.room_number =
B.room_number and Lives_in.dormid = B.dormid JOIN Has_Pet on Lives_in.stuid =
B.WhoisLoved;

/*Question 44*/
select * from (select max(PetAge) as mpa from Pet) as p1, Pet as p2 NATURAL JOIN Has_Pet
NATURAL JOIN Student where p2.PetAge = mpa;

/*Question 45*/
select A.FName, A.LName,A.room_number, A.dorm_name, B.FName,
B.LName,B.room_number, B.dorm_name from (select * from Has_Pet NATURAL JOIN Pet
NATURAL JOIN Student NATURAL JOIN Lives_in NATURAL JOIN Dorm) as A, (select * from
Has_Pet NATURAL JOIN Pet NATURAL JOIN Student NATURAL JOIN Lives_in NATURAL
JOIN Dorm) as B where (A.PetType = "Dog" and B.PetType = "Cat") or (A.PetType = "Cat" and
B.PetType = "Dog") or (A.PetType = "Parrot" and B.PetType = "Cat") or (A.PetType = "Cat"
and B.PetType = "Parrot") and A.stuid < B.stuid;

/*Question 46*/
select count(*) as rooms_occupied,dorm_name, student_capacity from Lives_in NATURAL
JOIN Dorm group by dormid;

/*Question 47*/
select dorm_name, count(*) from (select dorm_name,dormid from Has_Pet natural join Lives_in
natural join Dorm group by dormid, room_number) as A group by dormid;

/*Question 48*/
select C.dormid, c1/c2 from (select dormid,count(*) as c1 from (select dormid from Lives_in
Natural JOIN Has_Pet group by dormid,room_number) as A group by dormid) as B right join
(select dormid,count(*) as c2 from Lives_in group by dormid) as C on B.dormid = C.dormid;
### QBE

#### 1)

<table>
<thead>
<tr>
<th>STUDENT</th>
<th>StuID</th>
<th>Lname</th>
<th>Fname</th>
<th>Age</th>
<th>Sex</th>
<th>Major</th>
<th>Advisor</th>
<th>City_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td></td>
<td>P._Lname1</td>
<td>P._Fname1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>_city</td>
</tr>
<tr>
<td>s2</td>
<td></td>
<td>P._Lname2</td>
<td>P._Fname2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>_city</td>
</tr>
</tbody>
</table>

**LOVES**

<table>
<thead>
<tr>
<th>Wholoves</th>
<th>WhoIsLoved</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>_s2</td>
</tr>
<tr>
<td>s2</td>
<td>_s1</td>
</tr>
</tbody>
</table>

**LIKES**

<table>
<thead>
<tr>
<th>Wholikes</th>
<th>WhoIsLiked</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>_s3</td>
</tr>
<tr>
<td>s2</td>
<td>_s4</td>
</tr>
</tbody>
</table>

**Conditions**

\[-s_3 = -s_2 \text{ OR } -s_4 = -s_1\]

#### 4)

<table>
<thead>
<tr>
<th>STUDENT</th>
<th>StuID</th>
<th>Lname</th>
<th>Fname</th>
<th>Age</th>
<th>Sex</th>
<th>Major</th>
<th>Advisor</th>
<th>City_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>_s1</td>
<td></td>
<td>P._Lname1</td>
<td>P._Fname1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIVES_IN</th>
<th>StuID</th>
<th>DormID</th>
<th>Room_number</th>
</tr>
</thead>
<tbody>
<tr>
<td>_s1</td>
<td>_dorm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR_OWNERSHIP</th>
<th>StuID</th>
<th>CarID</th>
</tr>
</thead>
<tbody>
<tr>
<td>_s1</td>
<td>_c1</td>
<td></td>
</tr>
<tr>
<td>_s1</td>
<td></td>
<td>NOT_c1</td>
</tr>
<tr>
<td>STUDENT</td>
<td>StuID</td>
<td>Lname</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>_s1</td>
<td>P. Lname1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIVES_IN</th>
<th>StuID</th>
<th>DormID</th>
<th>Room_number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_s1</td>
<td>_dorm</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HAS_PET</th>
<th>StuID</th>
<th>PetID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_s1</td>
<td>_pet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR_OWNERSHIP</th>
<th>StuID</th>
<th>CarID</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT_EXIST</td>
<td>_s1</td>
<td>_c1</td>
</tr>
<tr>
<td>STUDENT</td>
<td>StuID</td>
<td>Lname</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>_s1</td>
<td>P._Lname1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIVES_IN</th>
<th>StuID</th>
<th>DormID</th>
<th>Room_number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_s1</td>
<td>dorm</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HAS_PET</th>
<th>StuID</th>
<th>PetID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_s1</td>
<td>CNT.UNQ.ALL._pet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR_OWNERSHIP</th>
<th>StuID</th>
<th>CarID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_s1</td>
<td>CNT.UNQ.ALL._car</td>
</tr>
</tbody>
</table>

Conditions

CNT.UNQ.ALL._car = 2
CNT.UNQ.ALL._pet >= 2
<table>
<thead>
<tr>
<th>STUDENT</th>
<th>StuID</th>
<th>Lname</th>
<th>Fname</th>
<th>Age</th>
<th>Sex</th>
<th>Major</th>
<th>Adviser</th>
<th>City_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>UNO.7</td>
<td>P.</td>
<td>P.</td>
<td></td>
<td></td>
<td></td>
<td>_c</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ENROLLED_IN</th>
<th>StuID</th>
<th>CID</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>_e</td>
<td>_e</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIVES_IN</th>
<th>StuID</th>
<th>DormID</th>
<th>Room_number</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>_d</td>
<td>_a</td>
<td>_b</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VotedForElection</th>
<th>StuID</th>
<th>Candidate_ID</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>_x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>US_Candidate</th>
<th>Candidate_ID</th>
<th>Candidate_Name</th>
<th>Party</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>'Donald Trump'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CITY</th>
<th>City_code</th>
<th>City_name</th>
<th>State</th>
<th>Country</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>'PA'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STUDENT</td>
<td>StuID</td>
<td>Lname</td>
<td>Fname</td>
<td>Age</td>
<td>Sex</td>
<td>Major</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>_a</td>
<td>P.</td>
<td>P.</td>
<td>P.</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MINOR_IN</th>
<th>StuID</th>
<th>DNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>_a</td>
<td>_b</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPARTMENT</th>
<th>DNO</th>
<th>Division</th>
<th>DName</th>
<th>Room</th>
<th>Building</th>
<th>DPhone</th>
</tr>
</thead>
<tbody>
<tr>
<td>_b</td>
<td>EN</td>
<td>EN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ENROLLED_IN</th>
<th>StuID</th>
<th>CID</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>_a</td>
<td>_c</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE</th>
<th>CID</th>
<th>CName</th>
<th>Credits</th>
<th>Instructor</th>
<th>Days</th>
<th>Hours</th>
<th>DNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>_c</td>
<td></td>
<td></td>
<td></td>
<td>_d</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FACULTY</th>
<th>FacID</th>
<th>Lname</th>
<th>Fname</th>
<th>Rank</th>
<th>Sex</th>
<th>Phone</th>
<th>Room</th>
<th>Building</th>
</tr>
</thead>
<tbody>
<tr>
<td>_d</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MEMBER_OF</th>
<th>FacID</th>
<th>DNO</th>
<th>Appt_Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>_d</td>
<td>_e</td>
<td></td>
<td>Primary</td>
</tr>
</tbody>
</table>
### CITY
<table>
<thead>
<tr>
<th>City_code</th>
<th>City_name</th>
<th>State</th>
<th>Country</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>_a</td>
<td>_b</td>
<td>_d</td>
<td>_d</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### VotedForElection
<table>
<thead>
<tr>
<th>VotedForElection</th>
<th>StuID</th>
<th>Candidate_ID</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>_x</td>
<td>_e</td>
<td></td>
<td>2016</td>
</tr>
<tr>
<td>_y</td>
<td>_e</td>
<td></td>
<td>2016</td>
</tr>
<tr>
<td>_x</td>
<td>_f</td>
<td></td>
<td>2020</td>
</tr>
<tr>
<td>_y</td>
<td>_f</td>
<td></td>
<td>2020</td>
</tr>
</tbody>
</table>

### ENROLLED_IN
<table>
<thead>
<tr>
<th>ENROLLED_IN</th>
<th>StuID</th>
<th>CID</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>_x</td>
<td>_c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_y</td>
<td>_c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_y</td>
<td>_g</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_z</td>
<td>_g</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### STUDENT
<table>
<thead>
<tr>
<th>STUDENT</th>
<th>StuID</th>
<th>Lname</th>
<th>Fname</th>
<th>Age</th>
<th>Sex</th>
<th>Major</th>
<th>Advisor</th>
<th>City_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>_x</td>
<td>_a</td>
<td>Smith</td>
<td>Linda</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>_b</td>
</tr>
<tr>
<td>_y</td>
<td></td>
<td>P.</td>
<td>_z</td>
<td>_j</td>
<td>_k</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CONDUCT_VIOLATION
<table>
<thead>
<tr>
<th>CONDUCT_VIOLATION</th>
<th>StuID</th>
<th>DormID</th>
<th>Reason</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>G._x</td>
<td></td>
<td></td>
<td>P.CNT.ALL.</td>
<td></td>
</tr>
</tbody>
</table>

### LIVES_IN
<table>
<thead>
<tr>
<th>LIVES_IN</th>
<th>StuID</th>
<th>DormID</th>
<th>Room_number</th>
</tr>
</thead>
<tbody>
<tr>
<td>_x</td>
<td>_d</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### DORM
<table>
<thead>
<tr>
<th>DORM</th>
<th>DormID</th>
<th>Dorm_name</th>
<th>Student_capacity</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>_d</td>
<td>P.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STUDENT</td>
<td>StuID</td>
<td>Lname</td>
<td>FName</td>
<td>Age</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>r1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VotedForElection</th>
<th>StuID</th>
<th>Candidate_ID</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>c1</td>
<td>2016</td>
<td></td>
</tr>
<tr>
<td>r1</td>
<td>c2</td>
<td>2020</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>US_Candidate</th>
<th>Candidate_ID</th>
<th>Candidate_Name</th>
<th>Party</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>c1Name</td>
<td>c1Party</td>
<td></td>
</tr>
<tr>
<td>c2</td>
<td>c2Name</td>
<td>c2Party</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>US_Candidate_For</th>
<th>Candidate_ID</th>
<th>Office</th>
<th>Location</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>President</td>
<td></td>
<td></td>
<td>2016</td>
</tr>
<tr>
<td>c2</td>
<td>President</td>
<td></td>
<td></td>
<td>2020</td>
</tr>
</tbody>
</table>

CONDITIONS
_c1 != _c2
<table>
<thead>
<tr>
<th>STUDENT</th>
<th>StuID</th>
<th>Lname</th>
<th>FName</th>
<th>Age</th>
<th>Sex</th>
<th>Major</th>
<th>Advisor</th>
<th>City_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>_r1</td>
<td>_r1</td>
<td>_r1</td>
<td>_r1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_r2</td>
<td>_r2</td>
<td>_r2</td>
<td>_r2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIVES_IN</th>
<th>StuID</th>
<th>DormID</th>
<th>Room_number</th>
</tr>
</thead>
<tbody>
<tr>
<td>_r1</td>
<td>_d</td>
<td>_n</td>
<td></td>
</tr>
<tr>
<td>_r2</td>
<td>_d</td>
<td>_n</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VotedForElection</th>
<th>StuID</th>
<th>Candidate_ID</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>_r1</td>
<td>_c1</td>
<td>2020</td>
<td></td>
</tr>
<tr>
<td>_r2</td>
<td>_c2</td>
<td>2020</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>US_Candidate</th>
<th>Candidate_ID</th>
<th>Candidate_Name</th>
<th>Party</th>
</tr>
</thead>
<tbody>
<tr>
<td>_c1</td>
<td>P._c1Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_c2</td>
<td>P._c2Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>US_Candidate_For</th>
<th>Candidate_ID</th>
<th>Office</th>
<th>Location</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>_c1</td>
<td>President</td>
<td></td>
<td></td>
<td>2020</td>
</tr>
<tr>
<td>_c2</td>
<td>President</td>
<td></td>
<td></td>
<td>2020</td>
</tr>
</tbody>
</table>

CONDITIONS

_c1 ! = _c2 AND _r1 < _r2
<table>
<thead>
<tr>
<th>STUDENT</th>
<th>StuID</th>
<th>Lname</th>
<th>Fname</th>
<th>Age</th>
<th>Sex</th>
<th>Major</th>
<th>Advisor</th>
<th>City_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>_r1</td>
<td>_r1</td>
<td>_r1Lname</td>
<td>_r1Fname</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>_city</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VotedForElection</th>
<th>StuID</th>
<th>Candidate_ID</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>_r1</td>
<td>_r1</td>
<td>_c1</td>
<td>_y1</td>
</tr>
<tr>
<td>_r1</td>
<td>_r1</td>
<td>_c2</td>
<td>_y2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>US_Candidate</th>
<th>Candidate_ID</th>
<th>Candidate_Name</th>
<th>Party</th>
</tr>
</thead>
<tbody>
<tr>
<td>_c1</td>
<td>_c1Name</td>
<td>_c1Party</td>
<td></td>
</tr>
<tr>
<td>_c2</td>
<td>_c2Name</td>
<td>_c2Party</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>US_Candidate_For</th>
<th>Candidate_ID</th>
<th>Office</th>
<th>Location</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>_c1</td>
<td>President</td>
<td></td>
<td>_y1</td>
<td></td>
</tr>
<tr>
<td>_c2</td>
<td>President</td>
<td></td>
<td>_y2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CITY</th>
<th>City_code</th>
<th>City_name</th>
<th>State</th>
<th>Country</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>_city</td>
<td>P._s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CONDITIONS
_c1Party != _c2Party AND _y1 != _y2
<table>
<thead>
<tr>
<th>STUDENT</th>
<th>StuID</th>
<th>Lname</th>
<th>Fname</th>
<th>Age</th>
<th>Sex</th>
<th>Major</th>
<th>Advisor</th>
<th>City_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>_r1</td>
<td>P._r1Lname</td>
<td>P._r1Fname</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>_city</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUDIED_ABROAD</th>
<th>StuID</th>
<th>Location</th>
<th>Start_Date</th>
<th>End_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>_r1</td>
<td>_place</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CITY</th>
<th>City_code</th>
<th>City_name</th>
<th>State</th>
<th>Country</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>_city</td>
<td></td>
<td></td>
<td></td>
<td>P._country</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CONDITIONS
_country == _place
<table>
<thead>
<tr>
<th>HAS_PET</th>
<th>StuID</th>
<th>PetID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_c</td>
<td>_d</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PET</th>
<th>PetID</th>
<th>PetName</th>
<th>PetType</th>
<th>PetAge</th>
<th>PetSex</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_d</td>
<td>_e</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HAS_ALLERGY</th>
<th>StuID</th>
<th>AllergyName</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_a</td>
<td>_e</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DORM</th>
<th>DormID</th>
<th>Dorm_name</th>
<th>Student_capacity</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_b</td>
<td>P.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STUDENT</th>
<th>StuID</th>
<th>Lname</th>
<th>Fname</th>
<th>Age</th>
<th>Sex</th>
<th>Major</th>
<th>Advisor</th>
<th>City_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_a</td>
<td>F.</td>
<td>F.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIVES_IN</th>
<th>StuID</th>
<th>DORMID</th>
<th>Room_number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_a</td>
<td>_b</td>
<td>_c</td>
</tr>
</tbody>
</table>