Relational Algebra/SQL Tutorial

1. Consider the following relations:

   Student (stuNum: integer, stuName: string, Program: string, level: integer, age: integer)
   Class (claName: string, meetsAt: time, room: string, facID: integer)
   Enrolled (stuNum: integer, claName: string)
   Faculty (facID: integer, facName: string, deptID: integer)

   The meaning of these relations is straightforward. For example, Enrolled has one record for each instance of a student enrolling in a class. Write the following queries in relational algebra/SQL. No duplicates should be included in each of the answers.

1.1. Find the names of all students in year 2 (level = 2) who are enrolled in a class taught by Vincent.

```
SELECT      DISTINCT S.stuName
FROM        Student S, Class C, Enrolled E, Faculty F
WHERE       S.stuNum = E.stuNum AND E.claName = C.claName AND C.facID =
F.facID AND
            F.facName = 'Vincent' AND S.level = 2
```

```
PROJECT[stuName]
(SELECT[level=2]Student)    JOIN    Enrolled    JOIN    Class    JOIN
(SELECT[facname=Vincent]Faculty)
```

1.2. Find the names of all classes that either meet in room TU107 or have 100 or more students enrolled.

```
SELECT      C.claName
FROM        Class C
WHERE       C.room = 'TU107'
            OR C.claName IN    (SELECT      E.claName
                                    FROM    Enrolled E
                                    GROUP BY    E.claName
                                    HAVING      COUNT(*) >= 100)
```

1.3. Find the names of all students who are enrolled in two classes that meet at the same time.

```
SELECT      DISTINCT S.stuName
FROM        Student S
WHERE       S.stuNum IN  (SELECT       E1.stuNum
                          FROM         Enrolled E1, Class C1, Enrolled
                                       E2, Class C2
                          WHERE        E1.stuNum = E2.stuNum
                                   AND E1.claName <> E2.claName
                                  AND E1.claName = C1.claName
```

```
                                                    AND E2.claName = C2.claName
                                                    AND C1.meetsAt = c2.meetsAt)
```

```
PROJECT[stuName]
(Student JOIN SELECT[A.meetsAt = B.meetsAt
AND A.stuNum=B.studNum AND A.claName>B.claName]
(RENAME[Enrolled JOIN Class]A TIMES RENAME[Enrolled JOIN Class]B))
```

1.4. Find the names of students who are not enrolled in any class.

```
SELECT      DISTINCT S.stuName
FROM        Student S
WHERE       S.stuNum NOT IN    (SELECT       E.stuNum
                                 FROM         Enrolled E)
```

```
PROJECT[studName]Student –
PROJECT[studName](Student JOIN Enrolled)
```

2. Consider the following schema:

    Vendors (venID: integer, venName, address: string)
    Software (swID: integer, swName: string, type: string)
    Catalog (venID: integer, swID: integer, cost: real)

The Catalog relation lists the prices charged for software by vendors. Write the following queries in relational algebra/SQL:

2.1 Find the swNames of softwares for which there is at least one vendor.

```
SELECT     S.swName
FROM       Software S, Catalog C
WHERE      S.swID = C.swID

PROJECT[swName](Software JOIN Catalog)
```

2.2 Find the venNames of vendors who supply every software.

```
SELECT     V.venName
FROM  Vendors V
WHERE      NOT EXISTS   (    (SELECT     *
                              FROM       Software S)
                              EXCEPT
                              (SELECT    C.swID
                              FROM       Catalog C
                              WHERE      C.venID = V.venID) )

PROJECT[venName](Vendors JOIN (Catalog DIVIDEBY PROJECT[swID]Software))
```

2.3 Find the venIDs of vendors who supply both operating system software and web browser software.

```
SELECT     DISTINCT C1.venID
FROM       Catalog C1, Software S
WHERE      C1.swID = S.swID AND S.type = ‘operating system’
INTERSECT
SELECT     DISTINCT C2.venID
FROM       Catalog C2, Software S
WHERE      C2.swID = S.swID AND S.type = ‘web browser’
```