DEMO ANSWERS

A1-

```
SELECT DISTINCT P.Name
FROM Professor P, Teaching T
WHERE T.ProfessorId = P.ProfessorId
AND T.CourseId LIKE 'CENG%'
AND P.DeptId <> 'CENG'
ORDER BY P.Name;
```

A2-

```
SELECT S.Name
FROM Student S, Transcript T, Course C
WHERE T.StudentId = S.StudentId
AND T.CourseId = C.CourseId
AND C.CourseName = 'Introduction to C Programming'
INTERSECT
SELECT S.Name
FROM Student S, Transcript T, Course C
WHERE T.StudentId = S.StudentId
AND T.CourseId = C.CourseId
AND C.CourseName = 'Electrical Circuits Laboratory';
ORDER BY S.Name;
```

A3-

```
SELECT P.Name , P.age
FROM Professor P, Teaching T
WHERE P.ProfessorId = T.ProfessorId
AND T.CourseId LIKE 'CENG%'
UNION
SELECT P.Name , P.age
FROM Professor P
WHERE P.DeptId = 'CENG'
ORDER BY P.Name;
```

A4-

```
SELECT S.Name, S.Gpa
FROM Student S
WHERE S.Gpa > (SELECT MAX (S1.Gpa)
FROM Student S1
WHERE S1.Status = 'Senior')
ORDER BY S.Name;
```

A5-

```
SELECT T.StudentId , AVG(T.Score)
FROM Transcript T
WHERE T.Semester IN ('F2013', 'S2014')
GROUP BY T.StudentId
HAVING AVG(T.Score)>50
ORDER BY T.StudentId;
```

A6-

```
SELECT S.Name
FROM Student S
WHERE NOT EXISTS (SELECT C.CourseId
FROM Course C
WHERE C.CourseId LIKE 'EE%'
EXCEPT
SELECT T.CourseId
FROM Transcript T
WHERE T.StudentId = S.StudentId )
ORDER BY S.Name;
```