Fall 2022

Comp2411 Tutorial No. 3:
****

1. For the chosen application you had modeled using ER during the tutorial class last time (*in Week 3*), please convert the ER diagram into relational schema (*table structures*) by following the "rules" described in the lecturing class.

Answer:

It'd depend on your group's ER diagram from the last time's tutorial class. Pay particular attention on converting the weak entity set into a table, which should be done together with the identifying relationship set – the double diamond! Also, when converting a ternary relationship set ***Tr*** into a table, we need to include all the primary keys from the three rectangles (*entity sets*) which are connected to ***Tr*** into the same table, along with the attributes of ***Tr*** (*if any*). Also note that for a *many-to-many* relationship set (*diamond*), we must convert it into a separate table. But for a *many-to-one*, or *one-to-one* diamond ***Td*** between two rectangles ***E1*** and ***E2***, it is possible for us to skip the table for ***Td*** if we incorporate the primary key of ***E2*** into the table for ***E1*** (*of course, plus the attributes of **Td** if any*). (*Fundamental DB – Chapter 9*: *ER-to-Relational Mapping Algorithm*)

2. Consider the following SQL query:

```
SELECT R.a1, R.a2
FROM   R, R1, R2
WHERE  R.a1 = R1.a1
AND    R.a1 = R2.a1;
```

Under what situations does the above query select tuples of the form (R.a1, R.a2) when R.a1 appears in both R1 and R2?

(***Hint***: *Examine carefully the cases where R1 and/or R2 may contain empty/null data*.)

Answer:

As hinted, we need to examine the cases of tuples from **R**, **R1**, and **R2** containing "**null**" value on attribute ***a1***, in which case **R.a1 = R1.a1** (*also **R.a1 = R2.a1***) would be false, thus no result for the pair (***R.a1***, ***R.a2***). In particular, for a tuple ***t*** of **R**, ***t1*** of **R1** and ***t2*** of **R2**, if any of the value **ϵ{t.a1, t1.a1, t2.a1}** is **null**, then the result will be **null**. Reversely, the pair of (***R.a1***, ***R.a2***) will appear in the result only if none of the values **ϵ{t.a1, t1.a1, t2.a1}** is null, and the condition **R.a1 = R1.a1** and **R.a1 = R2.a1** holds.

3. A database is to be set up to maintain the pool of lecture theatres and to assist in their allocation to courses. Consider the following relation/table with the set of functional dependencies **F** defined on its attributes:

```
CourseRmAlloc(CourseId, CourseName, Year, Lecturer, Enrollment, RoomId,
              RoomCapacity, Day, Time)

F = { CourseId -> CourseName,          CourseName -> CourseId,
      CourseId, Year -> Lecturer,      CourseId, Year -> Enrollment,
      RoomId -> RoomCapacity,          RoomId, Year, Day, Time -> CourseId,
      CourseId, Year, Day, Time -> RoomId }
```

   a) Find all the candidate keys of `CourseRmAlloc`. Demonstrate that they are indeed candidate keys.

Solution: There are 3 candidates keys, each has four attributes:

`(CourseId, Year, Day, Time),`

`(RoomId, Year, Day, Time),`

`(CourseName, Year, Day, Time).`

*** Need to show two aspects in order to prove some set of attributes is a candidate key:

(i) they can decide all the other attributes in the table, and

(ii) removing any attribute from the set will make the remaining set not powerful enough!

   b) Determine the highest normal form that the relation `CourseRmAlloc` is in, and justify your answer. What problems will arise with this relation?

Solution: `CourseRmAlloc` is in 1NF because there is a partial dependency `RoomId -> RoomCapacity` which makes it not in 2NF.

   c) Considering the following decomposition, give all the candidate keys for the relations `Course` and `RoomAlloc`. State what normal form each relation is in.

```
Course(CourseId, CourseName, Year, Lecturer, Enrollment)
RoomAlloc(RoomId, RoomCapacity, Day, Time, CourseId)
```

Solution: `Course` is in the 3NF but `RoomAlloc` is in the 1NF only.