**IS 441 Week 8 Class Summary and Highlights**

Dr. Yue Zhang; revised October 30, 2016

Class outline:

|  |  |
| --- | --- |
| 1. Normalization solution process;
2. Analysis of func dependencies;
 | 1. Correction of 2NF/3NF violations;
2. Exam review: ERD; Chap 4 HW problem designation.
 |

1. Normalization Solution Process (“Big Steps”)
2. Read, understand, examine data.

Result: Functional dependencies in 1NF (“Big Table”)

Details of this step is in the Main Contents Part II - “Analysis of functional dependencies”.

1. Apply normalization procedure on the relation with the functional dependencies identified in Part I.

There is ALWAYS ***ONE*** FD;

there does not have to be PD or TD

Result:

* + - * 1. Removal of PD - relations in 2NF
				2. Removal of TD – relations in 3NF – separate, smaller, “clean” relations/tables.

Normalization procedure: details in Part III – “Correction of 2NF/3NF violations”.

1. Analysis of Functional Dependencies (Take “Normalization Example 1” as example)

1. Use common sense, identify “easy” functional dependencies, such as EmpID 🡪 EmpLName, and ProjID 🡪 PName, etc.

* + - 1. In the remaining possible dependencies, first find repetitive values among a POSSIBLE determinant (“Determinant”, usually the determinants found in “1” above), examine the values of a POSSIBLE determinee (“Determinee”), to see whether the same value of the Determinant leads to the same value of a Determinee, and different Determinant values lead to different Determinee values. If yes, then there is a functional dependency between this pair of Determinant and Determinee; if not, move on to the next possible Determinee.

Repeat the procedure for all POSSIBLE Determinants and Determinees.

Result: PD (partial dependencies) and TD (transitive dependencies) are usually found directly in this step.

1. Identifying full dependency in the case of composite primary key

In conducting “2” above, \*IF\* there is a composite primary key (key that consists of two or more fields), pay attention to the special kind of Determinees who

* Does not depend on Determinant 1, AND
* Does not depend on Determinant 2.

In this case, verify that Determinant 1 and Determinant 2 **jointly determine** specific determinee(s), if so, that functional dependency is the Full Dependency – the determine(s) depend(s) on the full key.

\*\*\*NOTE 1: ALL relations have ONE full dependency – cannot be more than one, cannot be no full dependency.

There is ALWAYS ***ONE*** and ***ONLY one*** FD

\*\*\* Note 2: A full dependency does NOT have to have a composite key: a single field as the primary key of course fully determines the non-key fields depending on it.

There does NOT have to be a PD

1. Normalization Procedure: Correction of 2NF/3NF violations
	* + 1. Give every functional dependency its own “independence”: EVERY functional dependency becomes its own relation/table;
			2. Remove from the original relation the determinees in those violating functional dependencies.
				1. This applies to both 2NF and 3NF.

Normalization to reach 2NF:

**F**

**E**

**D**

**C**

**B**

**A**

In the above relation,

* The purple is FD;
* The red is PD (since it establishes a relationship between the determinant A – which is PART of the COMPOSITE KEY - and the non-key fields D and E);
* The blue determinant is also PD (since it establishes a relationship between the determinant B – which is PART of the COMPOSITE KEY - and the non-key field C);

Apply the above procedure:

1. Give every functional dependency its own “independence”: EVERY functional dependency becomes its own relation/table;

Red:

**E**

**D**

**A**

So the 2NF consists the following relations – the Full, and the Partial dependencies:

Blue:

**C**

**B**

1. Remove from the original relation the determinees in those violating functional dependencies.

**F**

**E**

**D**

**C**

**B**

**A**

becomes

**F**

**B**

**A**

Normalization to reach 3NF (a NEW example, unrelated with the above one for 2NF):

**R**

**Q**

**P**

**O**

**N**

**M**

In the above relation,

* The purple func dep is Full;
* The red func dep is Transitive (since it establishes a relationship between the determinant **R [Non-key!!]** and the non-key fields O, P and Q).

Apply the above procedure:

1. Give every functional dependency its own “independence”: EVERY functional dependency becomes its own relation/table;

Red:

**R**

**Q**

**P**

**O**

1. Remove from the original relation the determinees in those violating functional dependencies.

**R**

**Q**

**P**

**O**

**N**

**M**

Becomes

3NF:

**R**

**N**

**M**

Referential integrity constraint

**R**

**Q**

**P**

**O**