

Fifth Normal Form (5NF) - Lossless join condition is one of the most important criteria for a good database design. Decomposing a relation into 4NF does not always produce relations that are dependency preserving.

5NF is related to join dependency which is the term used to ~~indicate~~ indicate the property of a relation scheme that can not be decomposed losslessly into fewer simpler relations but can be decomposed losslessly into n simpler relations.

4NF is by no means the ultimate normal form. MVDs help us to understand and tackle some form of repetition of information that can not be represented in terms of FDs. So there are types of constraints called join dependencies that generalize MVDs and lead to another normal form called PJNF (Project join normal form) or 5NF.

When we applied 4NF to the relation supply, we have two resulting relations supply & vendor-project. These relations still have a problem because it is not dependency preserving (FDE between item & project is not retained) and decomposition is a lossy decomposition (because we can not find which items were used by which project). Therefore we need another relation which can relate itemcode with project code.

This means when we split the relation supply into two new relations applying 4NF, it is not dependency preserving & not lossless. Therefore ~~the~~ the relation can not be decomposed losslessly into 2 simpler relations, but can be losslessly decomposed into 3 simpler relations:

Vendor-supply		Vendor-project		Supply-project	
Vendecode	Itemcode	Vendecode	Projectcode	Itemcode	Projectcode
V1	I1	V1	P1	I1	P1
V1	I2	V1	P2	I2	P1
V2	I2	V2	P2	I1	P2
V3	I3	V3	P3	I2	P2
				I3	P3

The practical problem associated with 5NF is the general constraints are not only hard to reason but there is also no set of sound and complete rules for reasoning about these constraints. Hence, PJNF/5NF is used rarely.

when we applied 4NF to the relation supply, we have two resulting relations supplies, vendor-supply & vendor-project. These relations still have a problem because it is not dependency preserving (FDE between item & project is not retained) and decomposition is a lossy decomposition (because we can not find which items were used by which project). Therefore we need another relation which can relate itemcode with project code.

This means when we split the relation supply into two new relations applying 4NF, it is not dependency preserving & not lossless. Therefore the relation can not be decomposed losslessly into 2 simpler relations, but can be losslessly decomposed into 3 simpler relations:

Vendor-Supply		Vendor-Project		Supply-Project	
Vendcode	Itemcode	Vendcode	Projectcode	Itemcode	Projectcode
V1	I1	V1	P1	I1	P1
V1	I2	V1	P2	I2	P1
V2	I2	V2	P2	I1	P2
V3	I3	V3	P3	I2	P2
				I3	P3

The practical problem associated with 5NF is the general constraints are not only hard to reason but there is also no set of sound and complete rules for reasoning about these constraints. Hence, PJNF/5NF is used rarely.