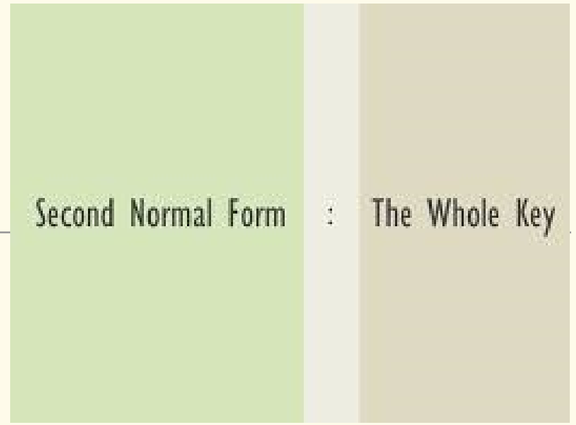
Second Normal Form



Second Normal Form (2NF)

- 2NF is based on the concept of full functional dependency.
- Recall, full functional dependency indicates that if A, B and C are attributes of a relation, and C is fully functionally dependent on the composition of A and B, then C is functionally dependent on A,B but not on any proper subset of A,B.
- 2NF ONLY applies to tables with composite primary keys (i.e. more than one attribute constitutes the primary key).

Second Normal Form (2NF)

- **A table is in second normal form (2NF) if and only if it is in 1NF and every non-key attribute is functionally dependent on the entire primary key and not just on a subset of the primary key (i.e. fully functionally dependent on the primary key).**

1NF to 2NF

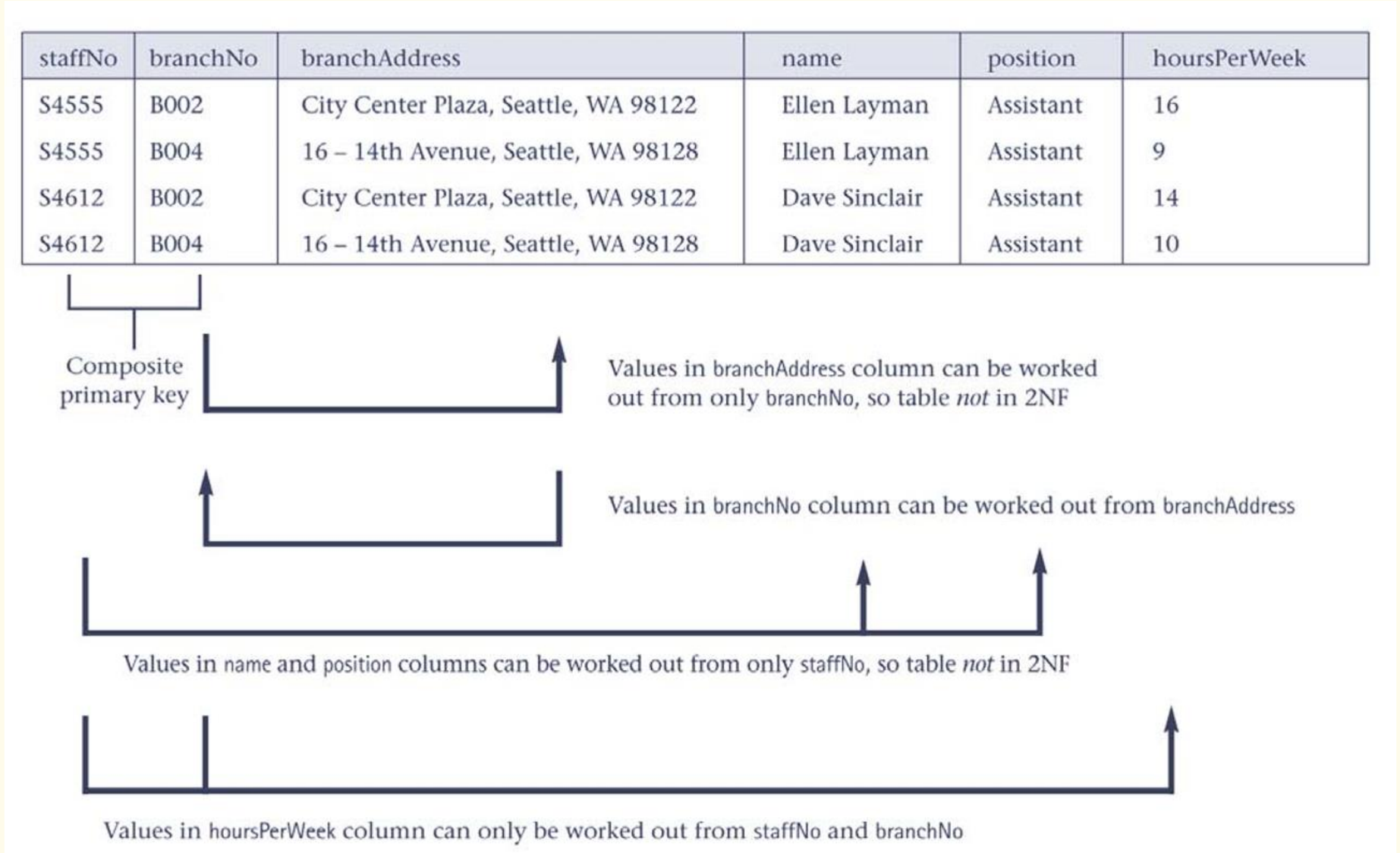
- Identify the primary key for the 1NF relation.
- Identify the functional dependencies in the relation.
- If partial dependencies exist on the primary key remove them by placing them in a new relation along with a copy of their determinant.
- The following table (TempStaffAllocation) is in 1NF and not in 2NF as some of the non-key attributes are functionally dependent on only a subset of the primary key.

TempStaffAllocation(staffNo, branchNo, branchAddress, name, position, hoursPerWeek)

Primary key staffNo, branchNo

1NF to 2NF

staffNo	branchNo	branchAddress	name	position	hoursPerWeek
S4555	B002	City Center Plaza, Seattle, WA 98122	Ellen Layman	Assistant	16
S4555	B004	16 – 14th Avenue, Seattle, WA 98128	Ellen Layman	Assistant	9
S4612	B002	City Center Plaza, Seattle, WA 98122	Dave Sinclair	Assistant	14
S4612	B004	16 – 14th Avenue, Seattle, WA 98128	Dave Sinclair	Assistant	10



1NF to 2NF

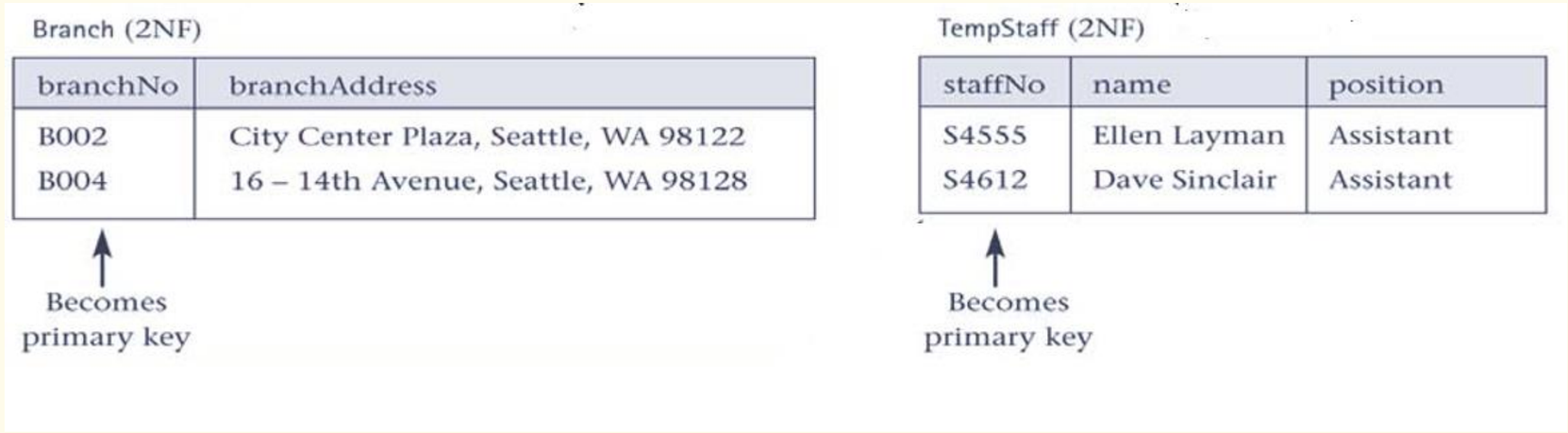
- The steps involved in transforming a table in first normal form into a set of second normal form tables are as follows:

    1. We must separate out the dependent attribute(s) and the determinant into a table of their own. The determinant becomes the primary key of this table.

    2. The determinant(s) remains as part of the composite key in the original table.

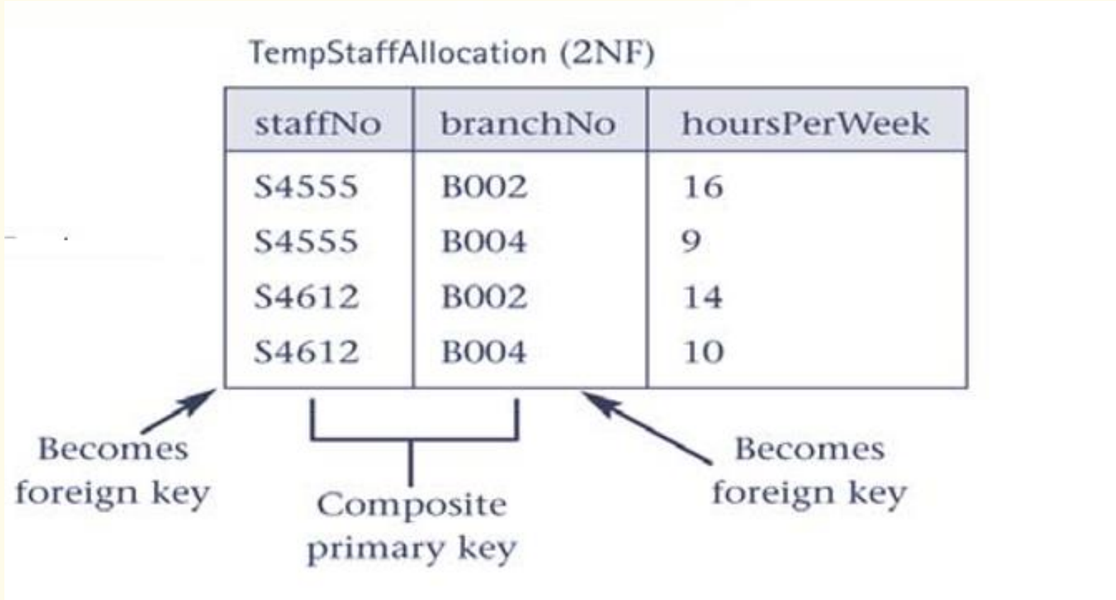
Example One

1. We must separate out the dependent attribute(s) and the determinant into a table of their own. The determinant becomes the primary key of this table.



Example One

2. The determinant(s) remains as part of the composite key in the original table.



Example One – Full set of relations

Branch(branchNo, branchAddress)
Primary key branchNo

TempStaff(staffNo, name, position)
Primary key staffNo

TempStaffAllocation(staffNo, branchNo, hoursPerWeek)
Primary key staffNo, branchNo
Foreign key branchNo references Branch(branchNo)
Foreign key staffNo references TempStaff(staffNo)

Example Two

EmployeeProject(PPS, ProjCode, Hours, Name, Title)

Primary key PPS, ProjCode

PPS	ProjCode	Hours	Name	Title
123456789	ABC	32.5	Smith, John	Galaxy
123456789	PQR	7.5	Smith, John	Romeo
333456781	ABC	20	English, Joyce	Galaxy
333456781	XYZ	14	English, Joyce	Alpha
333456781	JKL	6	English, Joyce	Sapphire
345123876	PQR	23	Ryan, Melanie	Romeo
345123876	XYZ	17	Ryan, Melanie	Alpha

EmployeeProject

Example Two

- The relation (EmployeeProject) is not in Second Normal Form because there are partial key dependencies (i.e. some of the non key attributes are functionally dependent on only a subset of the primary key).

  PPS → Name

  ProjCode → Title

Example Two

1. We must separate out the dependent attribute(s) and the determinant into a table of their own. The determinant becomes the primary key of this table.

   Employee(PPS, Name)

   Primary key PPS

   Project(ProjCode, Title)

   Primary key ProjCode

Example Two

2. The determinant(s) remains as part of the composite key in the original table.

   EmployeeProject(PPS, ProjCode, Hours)
   Primary key PPS, ProjCode
   Foreign key PPS references Employee(PPS)
   Foreign key ProjCode references Project(ProjCode)

Example Two – Full set of relations

Employee(PPS, Name)
Primary key PPS

Project(ProjCode, Title)
Primary key ProjCode

EmployeeProject(PPS, ProjCode, Hours)
Primary key PPS, ProjCode
Foreign key PPS references Employee(PPS)
Foreign key ProjCode references Project(ProjCode)