

Chapter 13

Normalization

Purpose of Normalization

- Reduce data redundancy
- *Avoid update anomalies*
- Reduced data inconsistencies
- Creation of well-formed relations

Data redundancy and update anomalies

- Problems associated with data redundancy and update anomalies are illustrated by comparing the *Staff* and *Branch* tables with the *StaffBranch* table given on next slides

Data redundancy and update anomalies

Staff

staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Branch

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

Staff (staffNo, sName, position, salary, branchNo)

Branch (branchNo, bAddress)

Data redundancy and update anomalies

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

StaffBranch (staffNo, sName, position, salary, branchNo, bAddress)

StaffBranch table has redundant data; the details of a branch are repeated for every member of staff

Data redundancy and update anomalies

- **Tables that contain redundant information may potentially suffer from update anomalies**
- **Types of update anomalies include**
 - **insertion**
 - **deletion**
 - **modification**

Insertion Anomalies

- To insert details of a new branch that currently has no members of staff into the StaffBranch relation, it is necessary to enter nulls into the staffNo attribute which is a primary key and violates the entity integrity

Deletion Anomalies

- To delete a Staff member details from the StaffBranch relation, we will have to delete the Branch details and vice versa

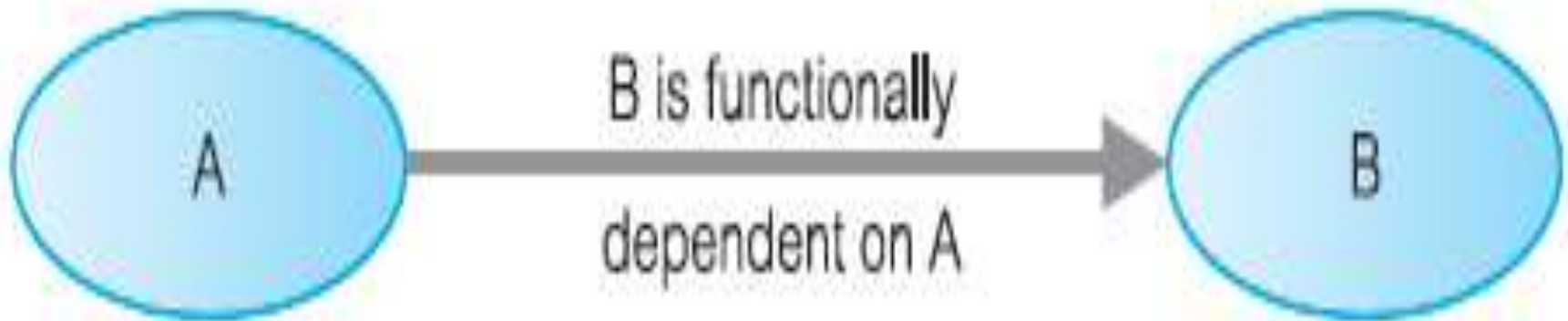
Modification Anomalies

- If we want to change the value of one of the attributes of a particular branch in the StaffBranch relation, for example the address for branch number B003, we must update the tuples of all staff located at that branch
- If this modification is not carried out on all the appropriate tuples of the StaffBranch relation, the database will become inconsistent
- In this example, branch number B003 may appear to have different addresses in different staff tuples

Functional Dependency

- Describes the relationship between attributes in a relation
- For example, if A and B are attributes of relation R, B is functionally dependent on A (denoted by $A \rightarrow B$), if for a given value of A there is exactly one value of B. (A and B may each consist of one or more attributes.)

Functional Dependency Diagram and Determinant



Determinant

Refers to the attribute, or group of attributes, on the left-hand side of the arrow of a functional dependency.

An example of a functional dependency

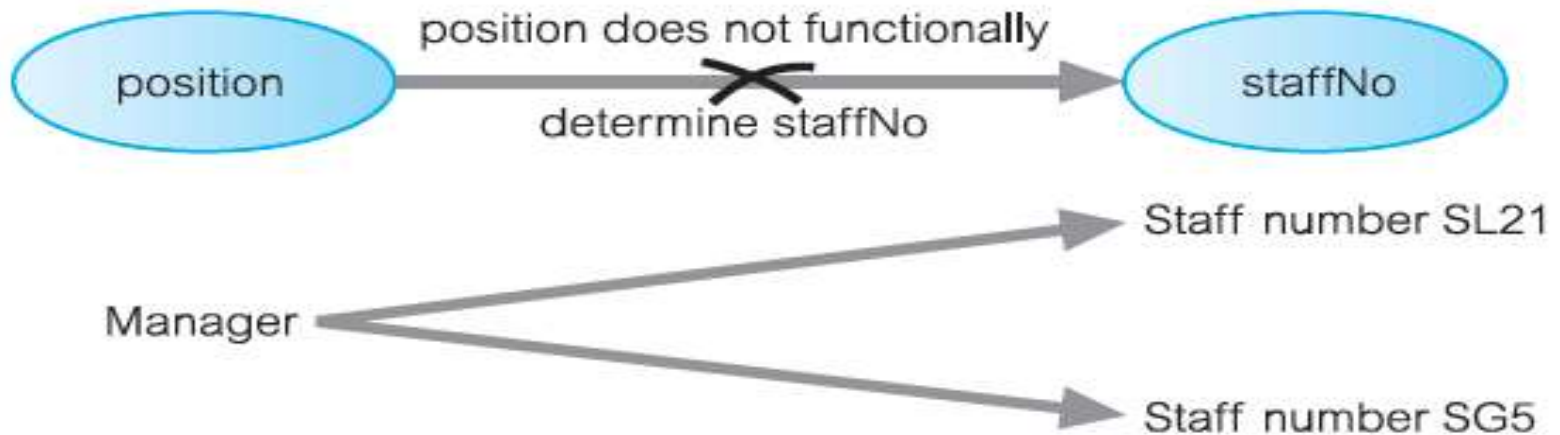
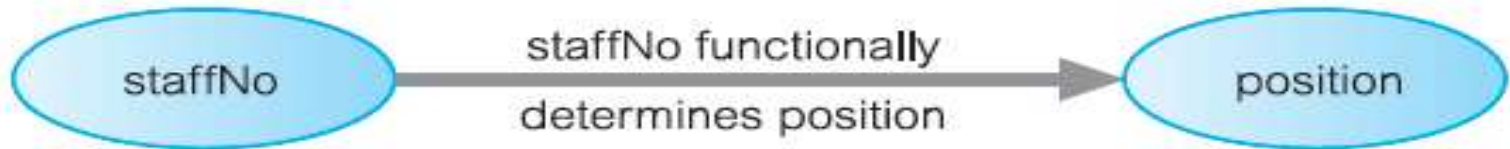
- Consider the attributes staffNo and position of the Staff relation
- For a specific staffNo, for example SL21, we can determine the position of that member of staff as Manager
- In other words, staffNo functionally determines position, as shown in Figure on next slide
- However, the opposite is not true, as position does not functionally determine staffNo
- A member of staff holds one position; however, there may be several members of staff with the same position

(a) staffNo functionally determines position

(staffNo \rightarrow position)

(b) position does *not* functionally determine staffNo

(position \nrightarrow staffNo)



An example of a functional dependency

- The relationship between staffNo and position is one-to-one (1:1): for each staff number there is only one position
- On the other hand, the relationship between position and staffNo is one-to-many (1:*): there are several staff numbers associated with a given position
- In this example, staffNo is the determinant of this functional dependency
- For the purposes of normalization we are interested in identifying functional dependencies between attributes of a relation that have a one-to-one relationship between the attribute(s) that makes up the **determinant on the left-hand side and the attribute(s) on the right-hand side** of a dependency

Full Functional Dependency

- Indicates that if A and B are attributes of a relation, B is fully functionally dependent on A if B is functionally dependent on A, but not on any proper subset of A

Transitive Dependency

- A condition where A, B, and C are attributes of a relation such that if $A \rightarrow B$ and $B \rightarrow C$, then C is transitively dependent on A via B
- Consider the following functional dependencies within the StaffBranch relation
staffNo \rightarrow sName, position, salary, branchNo, bAddress
branchNo \rightarrow bAddress
- The transitive dependency branchNo \rightarrow bAddress exists on staffNo via branchNo.
- In other words, the staffNo attribute functionally determines the bAddress via the branchNo attribute

The Process of Normalization

- Normalization is a formal technique for analyzing relations based on their primary key and functional dependencies (Codd, 1972b)

The Process of Normalization

- For the relational data model, it is important to recognize that it is only First Normal Form (1NF) that is critical in creating relations; all subsequent normal forms are optional
- However, to avoid the **update anomalies**, it is generally recommended that we proceed to at least Third Normal Form (3NF)

Unnormalized Form (UNF)

- A table that contains one or more **repeating groups**
- A repeating group is an attribute, or group of attributes, within a table that occurs with multiple values for a single occurrence of the nominated key attribute(s) for that table

ClientRental Unnormalized Table

ClientRental

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul-03	31-Aug-04	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-04	1-Sep-05	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-02	10-June-03	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-03	1-Dec-04	375	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

ClientRental Unnormalized Table

- We identify the key attribute for the ClientRental unnormalized table as clientNo
- Next, we identify the repeating group in the unnormalized table as the property rented details, which repeats for each client
- The structure of the repeating group is:
- Repeating Group = (propertyNo, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

ClientRental Unnormalized Table

- As a consequence, there are multiple values at the intersection of certain rows and columns
- For example, there are two values for propertyNo (PG4 and PG16) for the client named John Kay
- To transform an unnormalized table into 1NF, we ensure that there is a single value at the intersection of each row and column
- This is achieved by removing the repeating group

First normal form (1NF)

- A table in which there is no repeating group and the intersection of every column and record contains only *one* value

Another Example: Branch table is not in 1NF

branchNo	branchAddress	telNos
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618, 503-555-2727, 503-555-6534
B002	City Center Plaza, Seattle, WA 98122	206-555-6756, 206-555-8836
B003	14 - 8th Avenue, New York, NY 10012	212-371-3000
B004	16 - 14th Avenue, Seattle, WA 98128	206-555-3131, 206-555-4112

↑
Primary key

More than
one value, so
not in 1NF

Converting Branch table to 1NF

Branch (Not 1NF)

branchNo	branchAddress	telNos
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618, 503-555-2727, 503-555-6534
B002	City Center Plaza, Seattle, WA 98122	206-555-6756, 206-555-8836
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131, 206-555-4112

Take copy of branchNo column to new table to become foreign key

Remove telNos column and create new column called telNo in the new table

Branch (1NF)

branchNo	branchAddress
B001	8 Jefferson Way, Portland, OR 97201
B002	City Center Plaza, Seattle, WA 98122
B003	14 – 8th Avenue, New York, NY 10012
B004	16 – 14th Avenue, Seattle, WA 98128

Primary key

BranchTelephone (1NF)

branchNo	telNo
B001	503-555-3618
B001	503-555-2727
B001	503-555-6534
B002	206-555-6756
B002	206-555-8836
B003	212-371-3000
B004	206-555-3131
B004	206-555-4112

Becomes foreign key

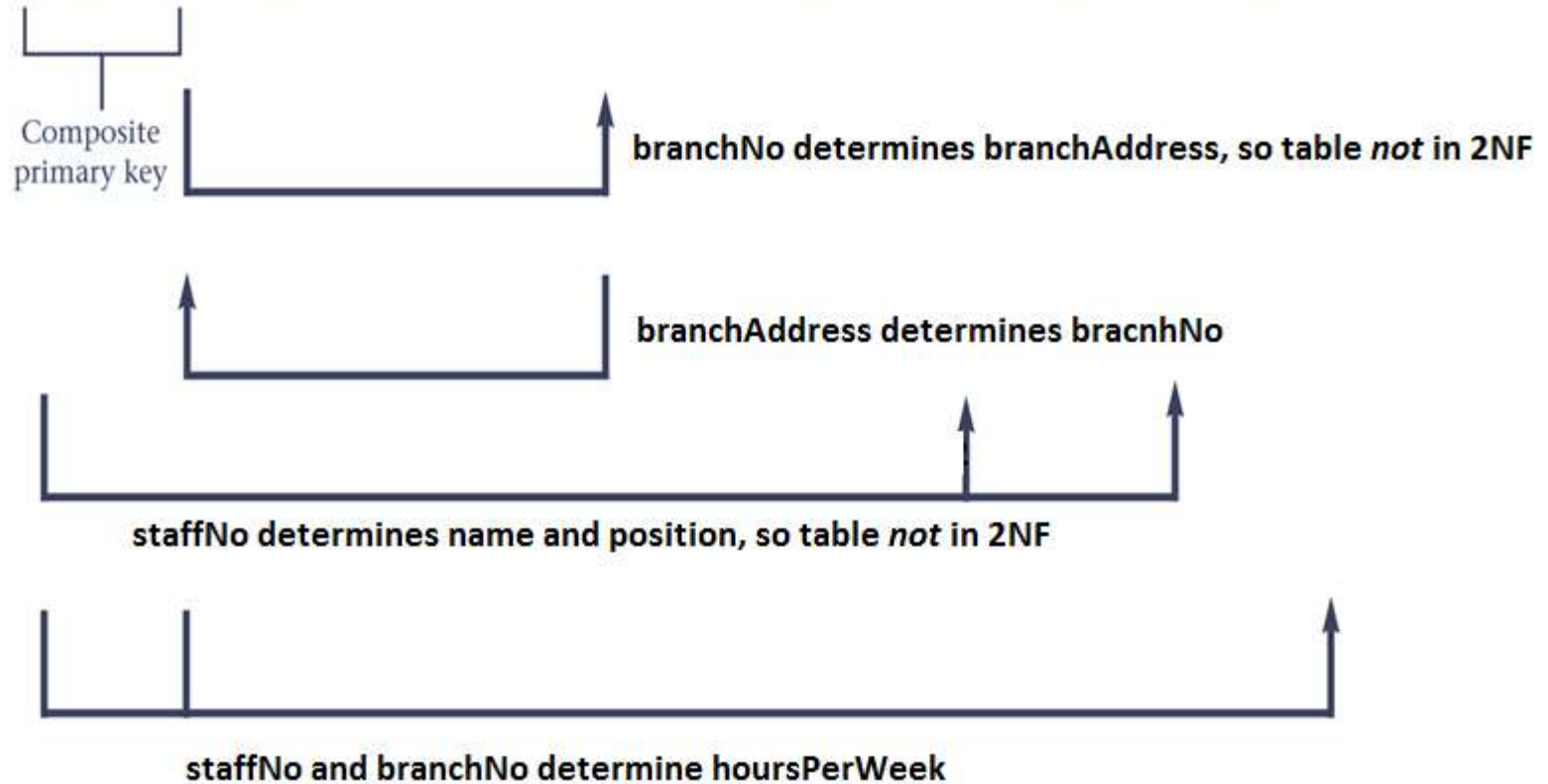
Becomes primary key

Second normal form (2NF)

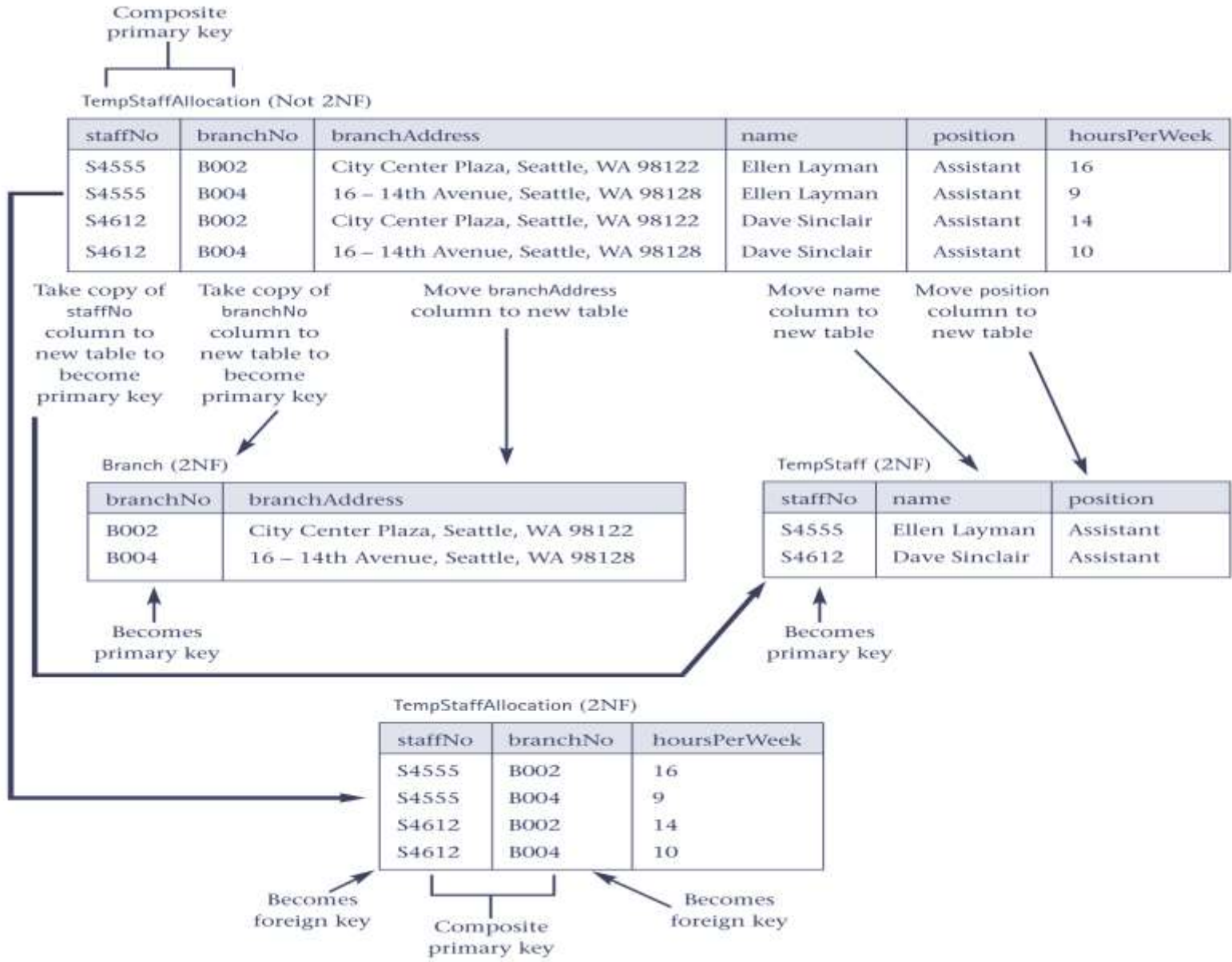
- Must be in 1NF and all non-primary-key attributes are *fully functionally dependent* on *primary key*
- It is only relevant when the key is **composite** i.e., a relation with a single **primary key is automatically in at least 2nd NF**

TempStaffAllocation table is not in 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



Converting TempStaffAllocation table to 2NF



Third normal form (3NF)

- If a transitive dependency exists on the primary key, the table is not in 3NF

StaffBranch table is not in 3NF

StaffBranch (Not 3NF)

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 - 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 - 8th Avenue, New York, NY 10012	212-371-3000



branchNo determines branchAddress and telNo, so table *not* in 3NF



branchAddress determines branchNo and telNo, so table *not* in 3NF



telNo determines branchNo and branchAddress, so table *not* in 3NF



Third normal form (3NF)

- The formal definition of 3NF is a table that is in 1NF and 2NF and in which no non-primary-key column is *transitively dependent* on the primary key

Converting the StaffBranch table to 3NF

StaffBranch (Not 3NF)

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 – 8th Avenue, New York, NY 10012	212-371-3000

Take copy of branchNo column to new table to become primary key

Move branchAddress column to new table

Move telNo column to new table

Branch (3NF)

branchNo	branchAddress	telNo
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
B002	City Center Plaza, Seattle, WA 98122	206-555-6756
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131

↑
Becomes primary key

↑
Becomes candidate key

↑
Becomes candidate key

Staff (3NF)

staffNo	name	position	salary	branchNo
S1500	Tom Daniels	Manager	46000	B001
S0003	Sally Adams	Assistant	30000	B001
S0010	Mary Martinez	Manager	50000	B002
S3250	Robert Chin	Supervisor	32000	B002
S2250	Sally Stern	Manager	48000	B004
S0415	Art Peters	Manager	41000	B003

↑
Primary key

↑
Becomes foreign key