## Data Normalization

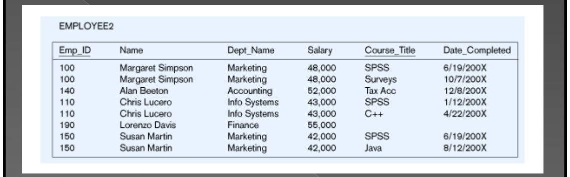
- Primarily a tool to validate and improve a logical design so that it satisfies certain constraints that ***avoid unnecessary duplication of data***
- The process of decomposing relations with anomalies to produce smaller, ***well-structured*** relations

## Well-Structured Relations

- A relation that contains minimal data redundancy and allows users to insert, delete, and update rows without causing data inconsistencies
- Goal is to avoid anomalies
  - Insertion Anomaly – adding new rows forces user to create duplicate data
  - Deletion Anomaly – deleting rows may cause a loss of data that would be needed for other future rows
  - Modification Anomaly – changing data in a row forces changes to other rows because of duplication

**General rule of thumb: a table should not pertain to more than one entity type**

## Example



EMPLOYEE2

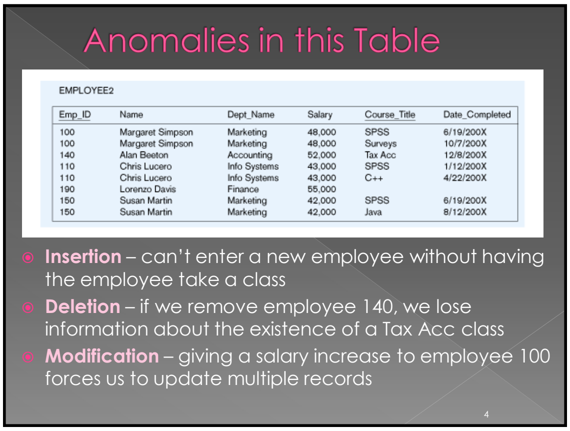
| Emp_ID | Name | Dept_Name | Salary | Course_Title | Date_Completed |
|---|---|---|---|---|---|
| 100 | Margaret Simpson | Marketing | 48,000 | SPSS | 6/19/200X |
| 100 | Margaret Simpson | Marketing | 48,000 | Surveys | 10/7/200X |
| 140 | Alan Beeton | Accounting | 52,000 | Tax Acc | 12/8/200X |
| 110 | Chris Lucero | Info Systems | 43,000 | SPSS | 1/12/200X |
| 110 | Chris Lucero | Info Systems | 43,000 | C++ | 4/22/200X |
| 190 | Lorenzo Davis | Finance | 55,000 | | |
| 150 | Susan Martin | Marketing | 42,000 | SPSS | 6/19/200X |
| 150 | Susan Martin | Marketing | 42,000 | Java | 8/12/200X |

Question – Is this a relation?

Answer – Yes: unique rows and no multivalued attributes

Question – What's the primary key?

Answer – Composite: Emp_ID, Course_Title

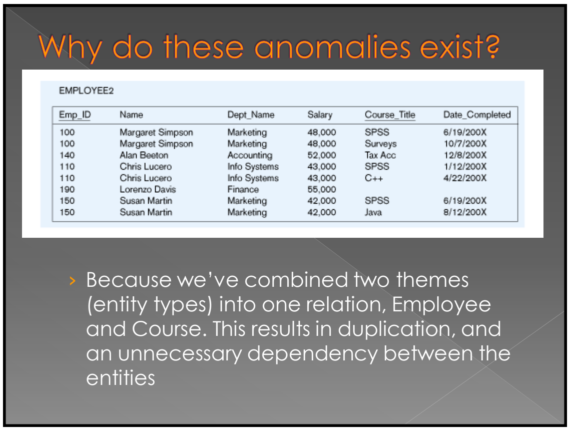


## Anomalies in this Table

EMPLOYEE2

| Emp_ID | Name | Dept_Name | Salary | Course_Title | Date_Completed |
|---|---|---|---|---|---|
| 100 | Margaret Simpson | Marketing | 48,000 | SPSS | 6/19/200X |
| 100 | Margaret Simpson | Marketing | 48,000 | Surveys | 10/7/200X |
| 140 | Alan Beeton | Accounting | 52,000 | Tax Acc | 12/8/200X |
| 110 | Chris Lucero | Info Systems | 43,000 | SPSS | 1/12/200X |
| 110 | Chris Lucero | Info Systems | 43,000 | C++ | 4/22/200X |
| 190 | Lorenzo Davis | Finance | 55,000 | | |
| 150 | Susan Martin | Marketing | 42,000 | SPSS | 6/19/200X |
| 150 | Susan Martin | Marketing | 42,000 | Java | 8/12/200X |

- **Insertion** – can't enter a new employee without having the employee take a class
- **Deletion** – if we remove employee 140, we lose information about the existence of a Tax Acc class
- **Modification** – giving a salary increase to employee 100 forces us to update multiple records

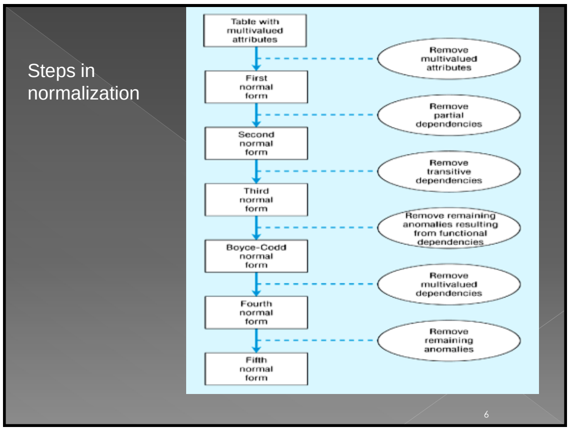


## Why do these anomalies exist?

EMPLOYEE2

| Emp_ID | Name | Dept_Name | Salary | Course_Title | Date_Completed |
|---|---|---|---|---|---|
| 100 | Margaret Simpson | Marketing | 48,000 | SPSS | 6/19/200X |
| 100 | Margaret Simpson | Marketing | 48,000 | Surveys | 10/7/200X |
| 140 | Alan Beeton | Accounting | 52,000 | Tax Acc | 12/8/200X |
| 110 | Chris Lucero | Info Systems | 43,000 | SPSS | 1/12/200X |
| 110 | Chris Lucero | Info Systems | 43,000 | C++ | 4/22/200X |
| 190 | Lorenzo Davis | Finance | 55,000 | | |
| 150 | Susan Martin | Marketing | 42,000 | SPSS | 6/19/200X |
| 150 | Susan Martin | Marketing | 42,000 | Java | 8/12/200X |

- Because we've combined two themes (entity types) into one relation, Employee and Course. This results in duplication, and an unnecessary dependency between the entities



Steps in normalization

Table with multivalued attributes

Remove multivalued attributes

First normal form

Remove partial dependencies

Second normal form

Remove transitive dependencies

Third normal form

Remove remaining anomalies resulting from functional dependencies

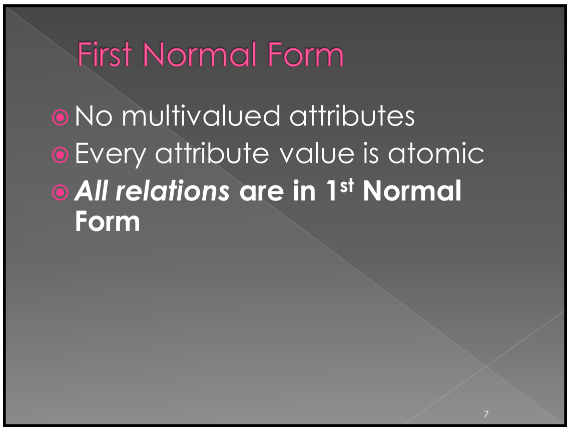
Boyce-Codd normal form

Remove multivalued dependencies

Fourth normal form

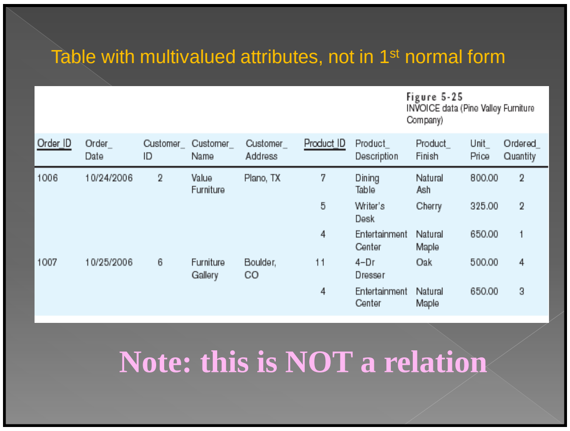
Remove remaining anomalies

Fifth normal form



## First Normal Form

- No multivalued attributes
- Every attribute value is atomic
- ***All relations* are in 1st Normal Form**

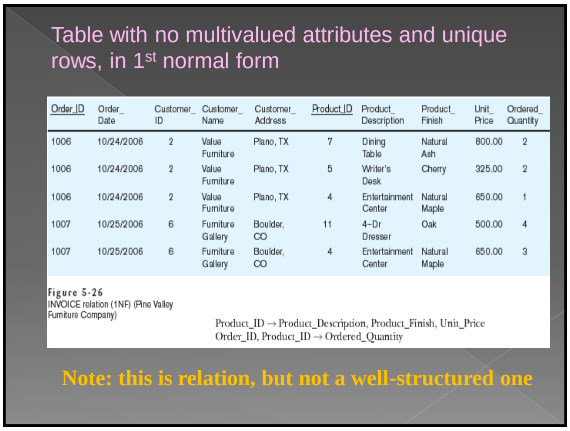


## Table with multivalued attributes, not in 1st normal form

**Figure 5-25**
INVOICE data (Pine Valley Furniture Company)

| Order_ID | Order_ Date | Customer_ ID | Customer_ Name | Customer_ Address | Product_ID | Product_ Description | Product_ Finish | Unit_ Price | Ordered_ Quantity |
|---|---|---|---|---|---|---|---|---|---|
| 1006 | 10/24/2006 | 2 | Value Furniture | Plano, TX | 7 | Dining Table | Natural Ash | 800.00 | 2 |
| | | | | | 5 | Writer's Desk | Cherry | 325.00 | 2 |
| | | | | | 4 | Entertainment Center | Natural Maple | 650.00 | 1 |
| 1007 | 10/25/2006 | 6 | Furniture Gallery | Boulder, CO | 11 | 4–Dr Dresser | Oak | 500.00 | 4 |
| | | | | | 4 | Entertainment Center | Natural Maple | 650.00 | 3 |

**Note: this is NOT a relation**



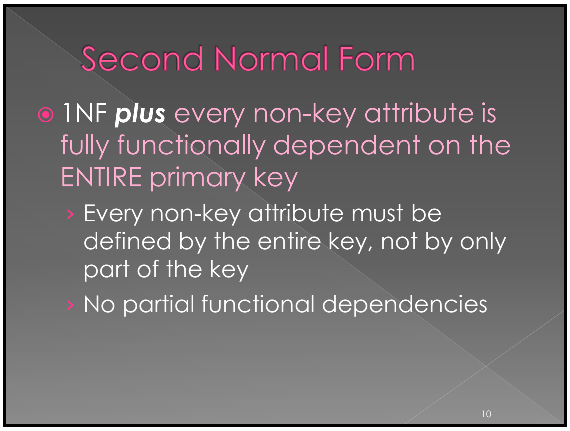
## Table with no multivalued attributes and unique rows, in 1st normal form

| Order_ID | Order_ Date | Customer_ ID | Customer_ Name | Customer_ Address | Product_ID | Product_ Description | Product_ Finish | Unit_ Price | Ordered_ Quantity |
|---|---|---|---|---|---|---|---|---|---|
| 1006 | 10/24/2006 | 2 | Value Furniture | Plano, TX | 7 | Dining Table | Natural Ash | 800.00 | 2 |
| 1006 | 10/24/2006 | 2 | Value Furniture | Plano, TX | 5 | Writer's Desk | Cherry | 325.00 | 2 |
| 1006 | 10/24/2006 | 2 | Value Furniture | Plano, TX | 4 | Entertainment Center | Natural Maple | 650.00 | 1 |
| 1007 | 10/25/2006 | 6 | Furniture Gallery | Boulder, CO | 11 | 4–Dr Dresser | Oak | 500.00 | 4 |
| 1007 | 10/25/2006 | 6 | Furniture Gallery | Boulder, CO | 4 | Entertainment Center | Natural Maple | 650.00 | 3 |

**Figure 5-26**
INVOICE relation (1NF) (Pine Valley Furniture Company)

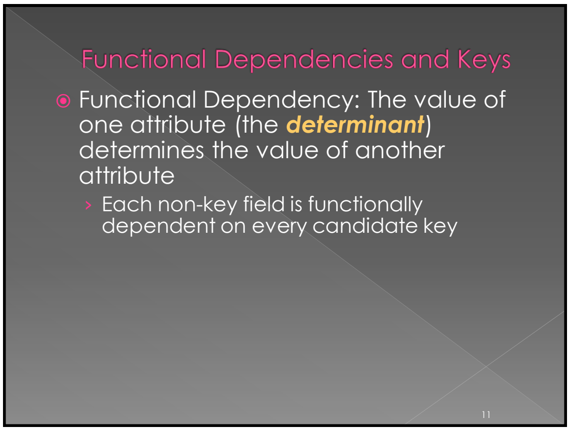
Product_ID → Product_Description, Product_Finish, Unit_Price
Order_ID, Product_ID → Ordered_Quantity

**Note: this is relation, but not a well-structured one**



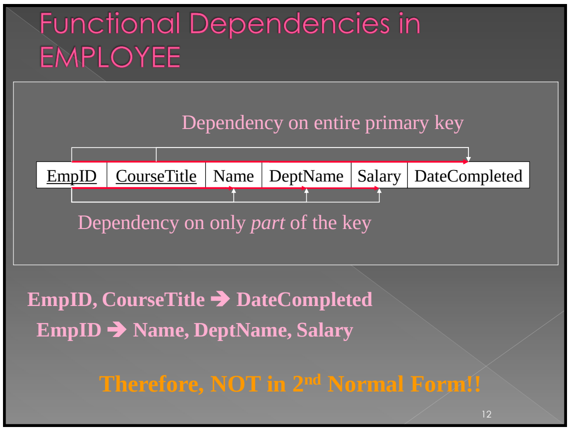
## Second Normal Form

- 1NF ***plus*** every non-key attribute is fully functionally dependent on the ENTIRE primary key
  - Every non-key attribute must be defined by the entire key, not by only part of the key
  - No partial functional dependencies



## Functional Dependencies and Keys

- Functional Dependency: The value of one attribute (the ***determinant***) determines the value of another attribute
  - Each non-key field is functionally dependent on every candidate key



## Functional Dependencies in EMPLOYEE

Dependency on entire primary key

| EmpID | CourseTitle | Name | DeptName | Salary | DateCompleted |
|---|---|---|---|---|---|

Dependency on only *part* of the key

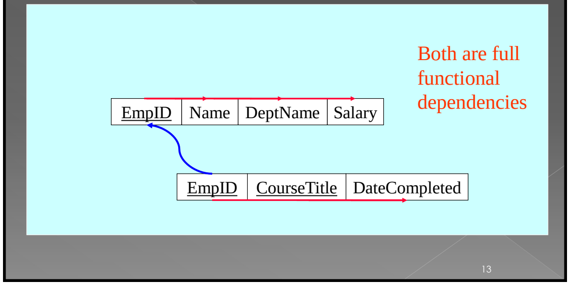
**EmpID, CourseTitle → DateCompleted**

**EmpID → Name, DeptName, Salary**

**Therefore, NOT in 2nd Normal Form!!**

## Getting it into 2nd Normal Form

- decomposed into two separate relations



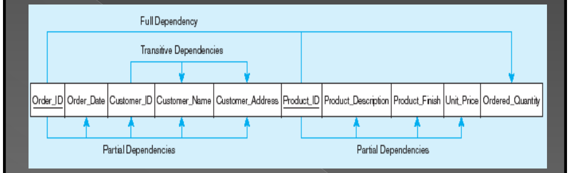
| <u>EmpID</u> | Name | DeptName | Salary |
|---|---|---|---|

| <u>EmpID</u> | <u>CourseTitle</u> | DateCompleted |
|---|---|---|

Both are full functional dependencies

13

## Functional dependency diagram for INVOICE



| <u>Order_ID</u> | Order_Date | Customer_ID | Customer_Name | Customer_Address | <u>Product_ID</u> | Product_Description | Product_Finish | Unit_Price | Ordered_Quantity |
|---|---|---|---|---|---|---|---|---|---|

**Order_ID ➔ Order_Date, Customer_ID, Customer_Name, Customer_Address**

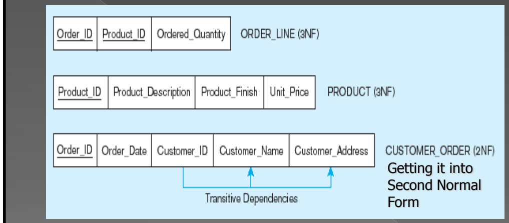
**Customer_ID ➔ Customer_Name, Customer_Address**

**Product_ID ➔ Product_Description, Product_Finish, Unit_Price**

**Order_ID, Product_ID ➔ Order_Quantity**

**Therefore, NOT in 2nd Normal Form**

## Removing partial dependencies



| <u>Order_ID</u> | <u>Product_ID</u> | Ordered_Quantity |
|---|---|---|

ORDER_LINE (3NF)

| <u>Product_ID</u> | Product_Description | Product_Finish | Unit_Price |
|---|---|---|---|

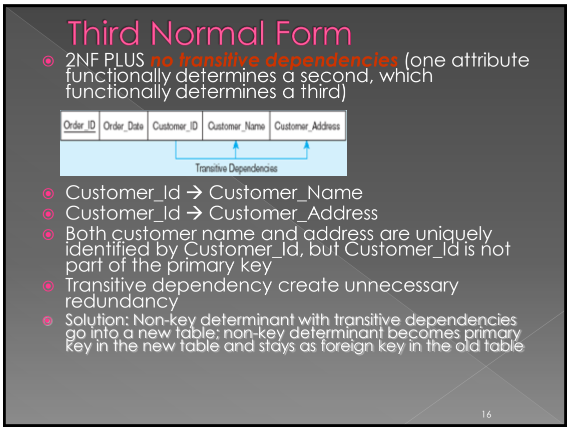
PRODUCT (3NF)

| <u>Order_ID</u> | Order_Date | Customer_ID | Customer_Name | Customer_Address |
|---|---|---|---|---|

CUSTOMER_ORDER (2NF)

Getting it into Second Normal Form

Partial dependencies are removed, but there are still **transitive dependencies**



## Third Normal Form

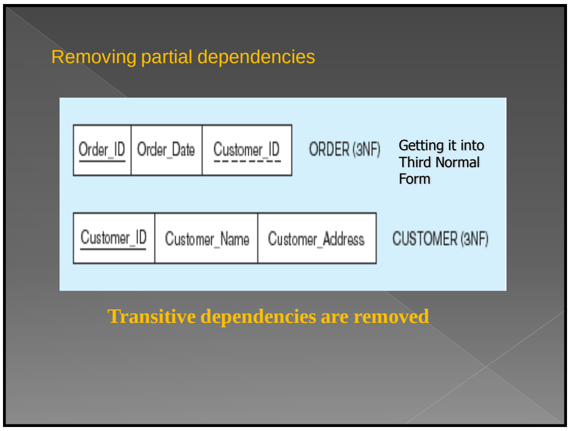
- 2NF PLUS *no transitive dependencies* (one attribute functionally determines a second, which functionally determines a third)

| Order_ID | Order_Date | Customer_ID | Customer_Name | Customer_Address |
|---|---|---|---|---|

Transitive Dependencies

- Customer_Id → Customer_Name
- Customer_Id → Customer_Address
- Both customer name and address are uniquely identified by Customer_Id, but Customer_Id is not part of the primary key
- Transitive dependency create unnecessary redundancy
- Solution: Non-key determinant with transitive dependencies go into a new table; non-key determinant becomes primary key in the new table and stays as foreign key in the old table

16



Removing partial dependencies

| Order_ID | Order_Date | Customer_ID |
|---|---|---|

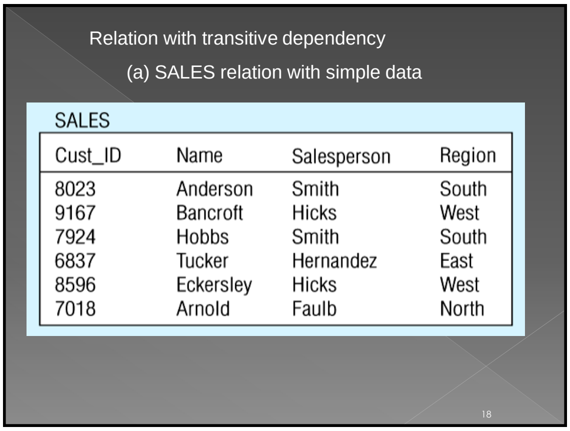
ORDER (3NF)

Getting it into Third Normal Form

| Customer_ID | Customer_Name | Customer_Address |
|---|---|---|

CUSTOMER (3NF)

**Transitive dependencies are removed**



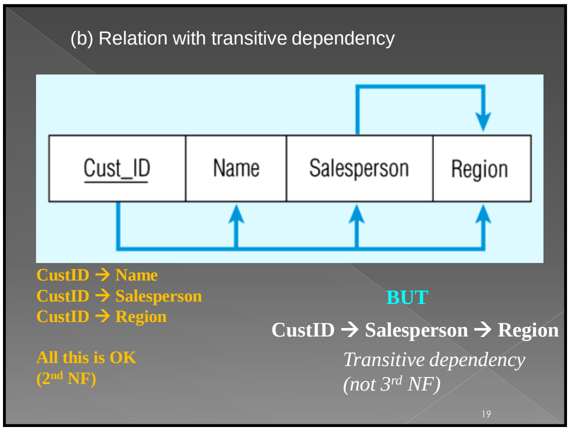
Relation with transitive dependency

(a) SALES relation with simple data

SALES

| Cust_ID | Name | Salesperson | Region |
|---|---|---|---|
| 8023 | Anderson | Smith | South |
| 9167 | Bancroft | Hicks | West |
| 7924 | Hobbs | Smith | South |
| 6837 | Tucker | Hernandez | East |
| 8596 | Eckersley | Hicks | West |
| 7018 | Arnold | Faulb | North |

18



(b) Relation with transitive dependency

| Cust_ID | Name | Salesperson | Region |
|---|---|---|---|

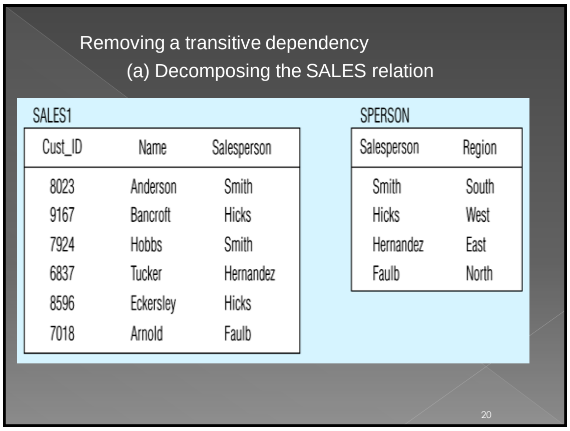
CustID → Name
CustID → Salesperson
CustID → Region

All this is OK
(2nd NF)

BUT

**CustID → Salesperson → Region**

*Transitive dependency*
*(not 3rd NF)*



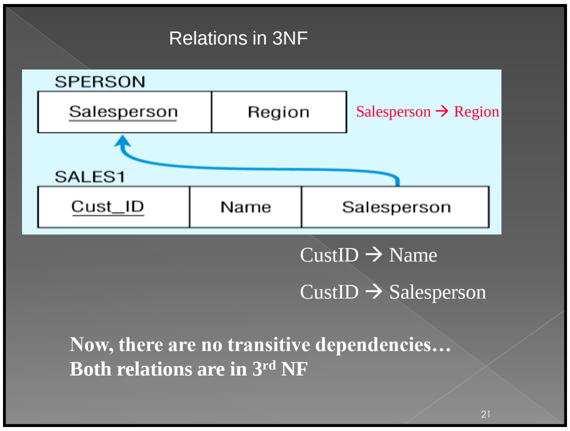
Removing a transitive dependency
(a) Decomposing the SALES relation

SALES1

| Cust_ID | Name | Salesperson |
|---|---|---|
| 8023 | Anderson | Smith |
| 9167 | Bancroft | Hicks |
| 7924 | Hobbs | Smith |
| 6837 | Tucker | Hernandez |
| 8596 | Eckersley | Hicks |
| 7018 | Arnold | Faulb |

SPERSON

| Salesperson | Region |
|---|---|
| Smith | South |
| Hicks | West |
| Hernandez | East |
| Faulb | North |



Relations in 3NF

SPERSON

| Salesperson | Region |
|---|---|

Salesperson → Region

SALES1

| Cust_ID | Name | Salesperson |
|---|---|---|

CustID → Name

CustID → Salesperson

**Now, there are no transitive dependencies…**
**Both relations are in 3rd NF**