* + 1. **Using the INVOICE table structure shown in Table P6.3, do the following.**

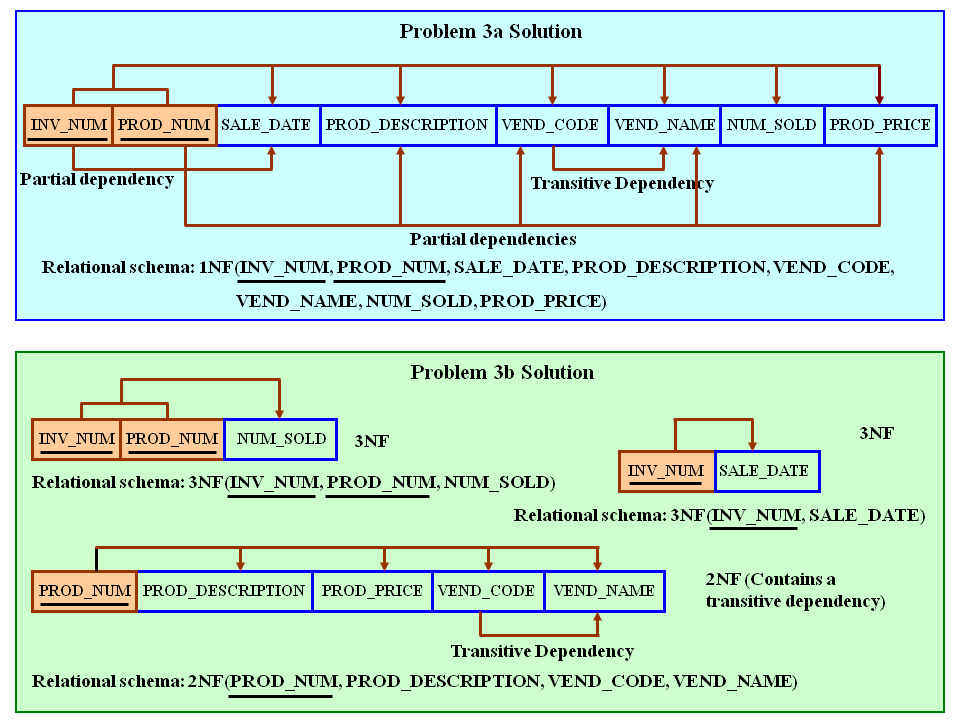
**Table P6.3 Sample INVOICE Records**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Sample Value** | **Sample Value** | **Sample Value** | **Sample Value** | **Sample Value** |
| INV\_NUM | 211347 | 211347 | 211347 | 211348 | 211349 |
| PROD\_NUM | AA-E3422QW | QD-300932X | RU-995748G | AA-E3422QW | GH-778345P |
| SALE\_DATE | 15-Jan-2010 | 15-Jan-2010 | 15-Jan-2010 | 15-Jan-2010 | 16-Jan-2010 |
| PROD\_LABEL | Rotary sander | 0.25-in. drill bit | Band saw | Rotary sander | Power drill |
| VEND\_CODE | 211 | 211 | 309 | 211 | 157 |
| VEND\_NAME | NeverFail, Inc. | NeverFail, Inc. | BeGood, Inc. | NeverFail, Inc. | ToughGo, Inc. |
| QUANT\_SOLD | 1 | 8 | 1 | 2 | 1 |
| PROD\_PRICE | $49.95 | $3.45 | $39.99 | $49.95 | $87.75 |

**a. Write the relational schema, draw its dependency diagram, and identify all dependencies, including all partial and transitive dependencies. You can assume that the table does not contain repeating groups and that any invoice number may reference more than one product. (*Hint*: This table uses a composite primary key.)**

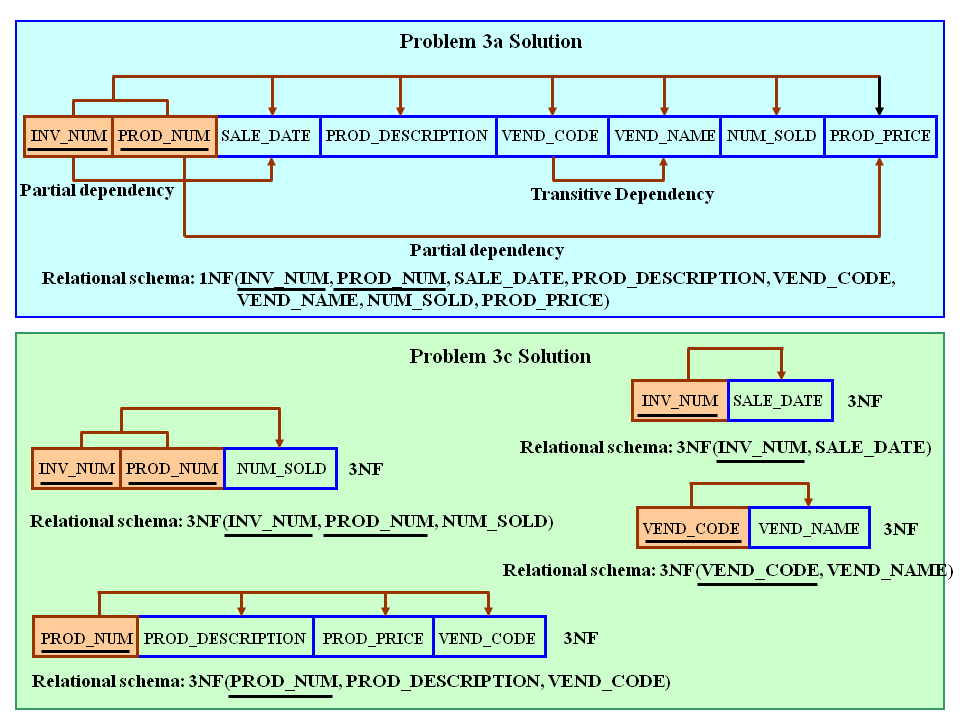
**b. Remove all partial dependencies, draw the new dependency diagrams, and identify the normal forms for each table structure you created.**

|  |
| --- |
| **NOTE**  **You can assume that any given product is supplied by a single vendor, but a vendor can supply many products. Therefore, it is proper to conclude that the following dependency exists:**  **PROD\_NUM → PROD\_DESCRIPTION, PROD\_PRICE, VEND\_CODE, VEND\_NAME**  **(*Hint*: Your actions should produce three dependency diagrams.)** |

**Figure P6.3a The Dependency Diagrams for Problems 3a and 3b**

**c. Remove all transitive dependencies, and draw the new dependency diagrams. Also identify the normal forms for each table structure you created.**

**Figure P6.3c The Dependency Diagram for Problem 3c**



**d. Draw the Crow’s Foot ERD.**



**Note:**

***Whilst the Relational Digram above, shows the entities exactly as they fell out of the***

***Normalization process, the ERD does not show the INV\_LINE entity in quite the same***

***way. If it did, then the the PK would be composite: (INV\_NUM+PROD\_CODE).***

***However, the ERD implementation shown, follows normal business implementation***

***of invoicing systems and technically allows a product to appear several times in the same***

***invoice (Perhaps for open orders with different prices on quantities).***

***If we wanted to ensure that a product can only appear once on an invoice,***

***then a cardinality of 0,1 would be implied on the INV\_LINE side in the relationship***

***between PRODUCT and INV\_LINE. (Reads: a given product may appear only once or not***

***at all on a given invoice)***

***In the internal model, this would be implemented by having a UNIQUE constraint on***

***the PROD\_CODE FK attribute in the INV\_LINE table.***