Normalisation: Example 4 (Case 2)

Example 4: Normalisation

A traditional data file – sales data

Date	*Product*	*Price*	*Client*	*Phone*	*Address*
11 Jan	Widget	100	Nurk Inc.	666-999	11 Bush Ave
12 Jan	Gizmo	120	Klutz & Co	131-313	13 Luck Rd
12 Jan	Widget	100	Bloggs Ltd	123-456	12 High St
13 Jan	Widget	100	Klutz Coy.	131-323	13 Luck Rd
14 Jan	Gizmo	120	F. Nurk Inc.	666-999	11 Bushy Ave

Case 2

Suppose that in any single day

- Clients sometimes make several orders
- Each order is for a different product

What do these rule mean?

- Then, the combination of DATE and CLIENT no longer provides a unique identifier for each record,
- We need to include PRODUCT to uniquely identify each record
- Our primary key is thus a composite key consisting of DATE, CLIENT and PRODUCT
- So, this variation in rules leads to a slightly different normalization process, but a similar end result.

Example 4: Normalisation

UNF

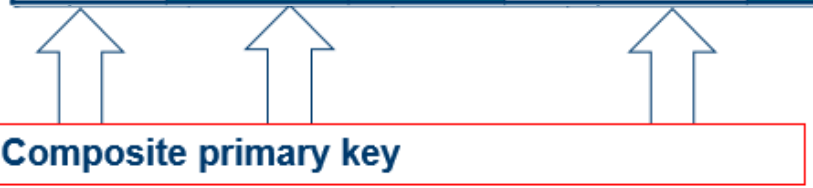
Sale

Date	Product	Price	Client	Phone	Address

Example 4: Normalisation

- **Convert to 1NF**
- Step 1: No null entries in the table now, so nothing to do!
- Step 2: Identify the primary key

Date	*Product*	*Price*	*Client*	*Phone*	*Address*
11 Jan	Widget	100	Nurk Inc.	666-999	11 Bush Ave
12 Jan	Gizmo	120	Klutz & Co	131-313	13 Luck Rd
12 Jan	Widget	100	Bloggs Ltd	123-456	12 High St
13 Jan	Widget	100	Klutz Coy.	131-323	13 Luck Rd
14 Jan	Gizmo	120	F. Nurk Inc.	666-999	11 Bushy Ave



Example 4: Normalisation

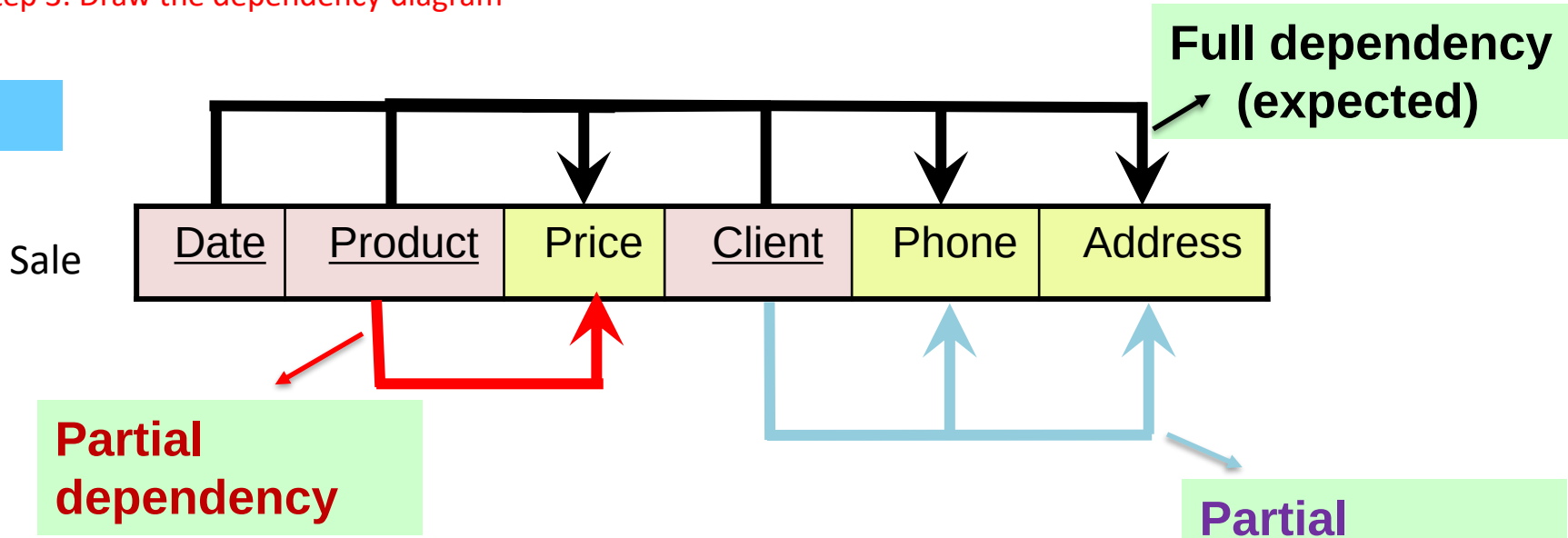
1NF

Sale

Date	Product	Price	Client	Phone	Address

- Step 3: Draw the dependency diagram

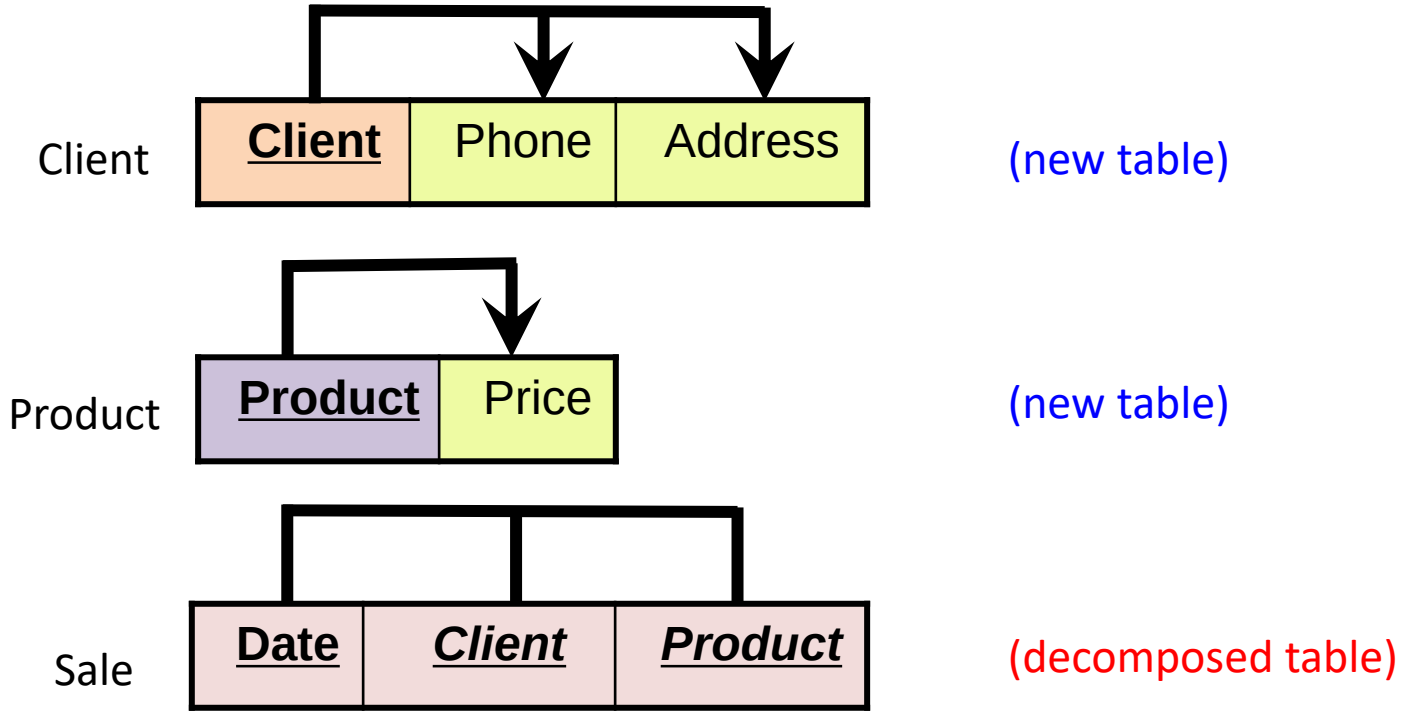
1NF



Example 4: Normalisation

- **Convert to 2NF**
  - Step 4: Remove partial dependency

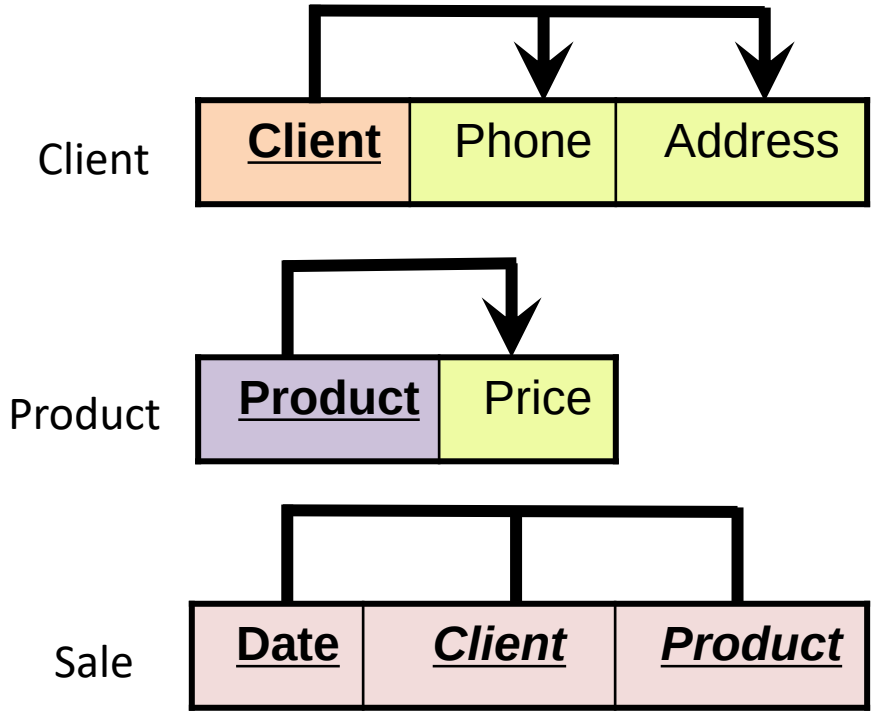
2NF



Example 4: Normalisation

- **Convert to 3NF**
  - Step 5: Remove transitive dependency

3NF



(no change)

(no change)

(no change)

Example 4: Relation schema

UNF

Sale (date, product, price, client, phone, address)

1NF

Sale (**date**, **product**, price, **client**, phone, address)

2NF

Product (**product**, price)
Customer (**client**, phone, address)
Sale (**date**, ***product***, ***client***)

3NF

Product (**product**, price)
Customer (**client**, phone, address)
Sale (**date**, ***product***, ***client***)

Normalisation: Example 5 (Case 3)

Example 5: Normalisation

A traditional data file – sales data

Date	*Product*	*Price*	*Client*	*Phone*	*Address*
11 Jan	Widget	100	Nurk Inc.	666-999	11 Bush Ave
12 Jan	Gizmo	120	Klutz & Co	131-313	13 Luck Rd
12 Jan	Widget	100	Bloggs Ltd	123-456	12 High St
13 Jan	Widget	100	Klutz Coy.	131-323	13 Luck Rd
14 Jan	Gizmo	120	F. Nurk Inc.	666-999	11 Bushy Ave

Case 3

Suppose that

- Clients sometimes order the same product several times in a single day

What does it mean?

- So, the combination of DATE, CLIENT and PRODUCT no longer provide a candidate key
- We have no choice but to add a new field (call it "SALE") that uniquely identifies each record and can therefore serve as a primary key
- Once again, this case leads to a slightly different normalization process, but a similar end result

Example 5: Normalisation

Sale ID	Date	Product	Price	Client	Phone	Address
001	11 Jan	Widget	100	Nurk Inc.	666-999	11 Bush Ave
002	12 Jan	Gizmo	120	Klutz & Co	131-313	13 Luck Rd
003	12 Jan	Widget	100	Bloggs Ltd	123-456	12 High St
004	13 Jan	Widget	100	Klutz Coy.	131-323	13 Luck Rd
005	14 Jan	Gizmo	120	F. Nurk Inc.	666-999	11 Bushy Ave

UNF

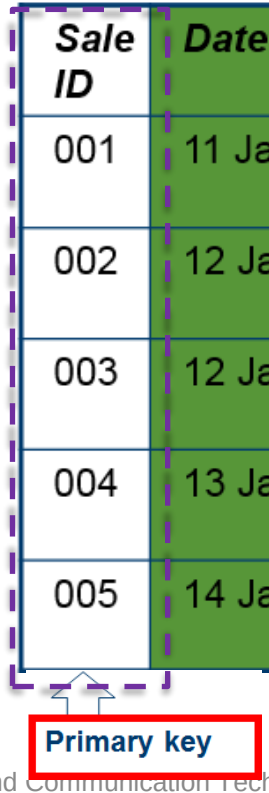
Sale

SaleID	Date	Product	Price	Client	Phone	Address

Example 5: Normalisation

- **Convert to 1NF**
- Step 1: No null entries in the table now, so nothing to do!
- Step 2: Identify the primary key

Sale ID	Date	Product	Price	Client	Phone	Address
001	11 Jan	Widget	100	Nurk Inc.	666-999	11 Bush Ave
002	12 Jan	Gizmo	120	Klutz & Co	131-313	13 Luck Rd
003	12 Jan	Widget	100	Bloggs Ltd	123-456	12 High St
004	13 Jan	Widget	100	Klutz Coy.	131-323	13 Luck Rd
005	14 Jan	Gizmo	120	F. Nurk Inc.	666-999	11 Bushy Ave

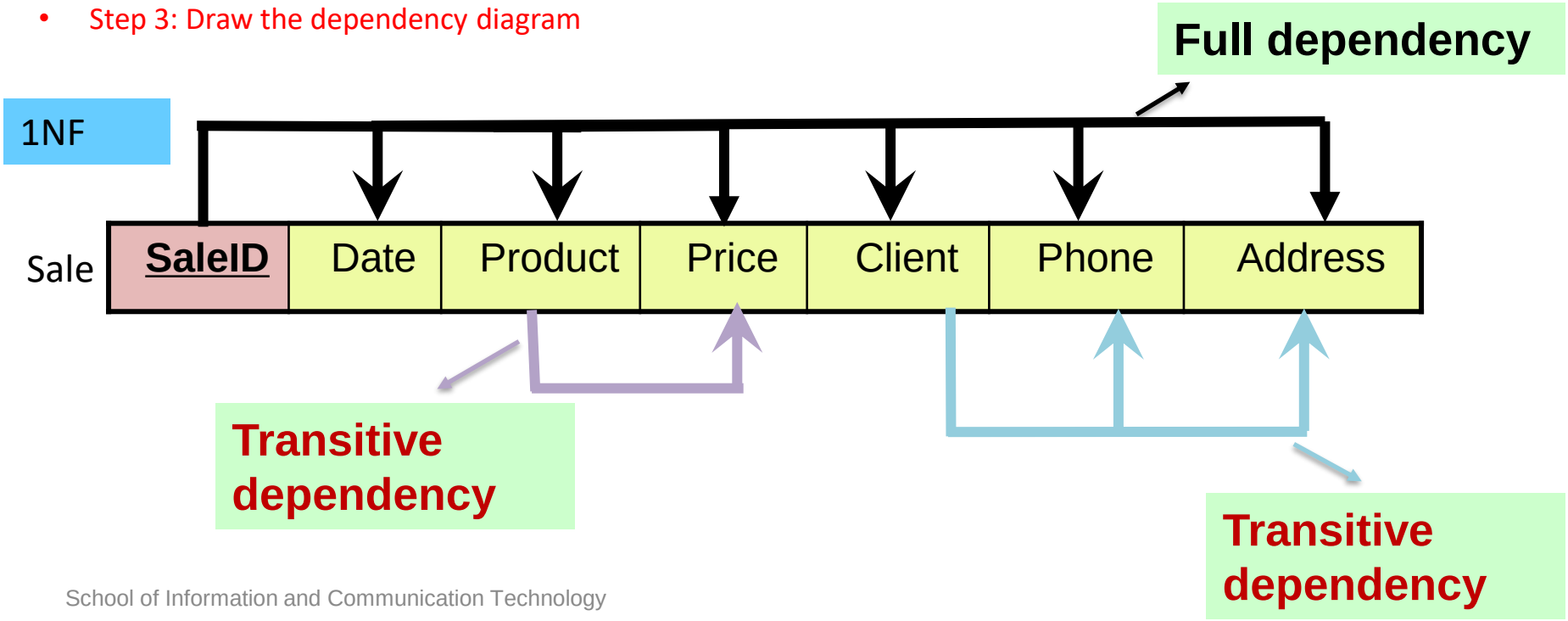


Example 5: Normalisation

1NF

Sale	**SaleID**	Date	Product	Price	Client	Phone	Address

- Step 3: Draw the dependency diagram

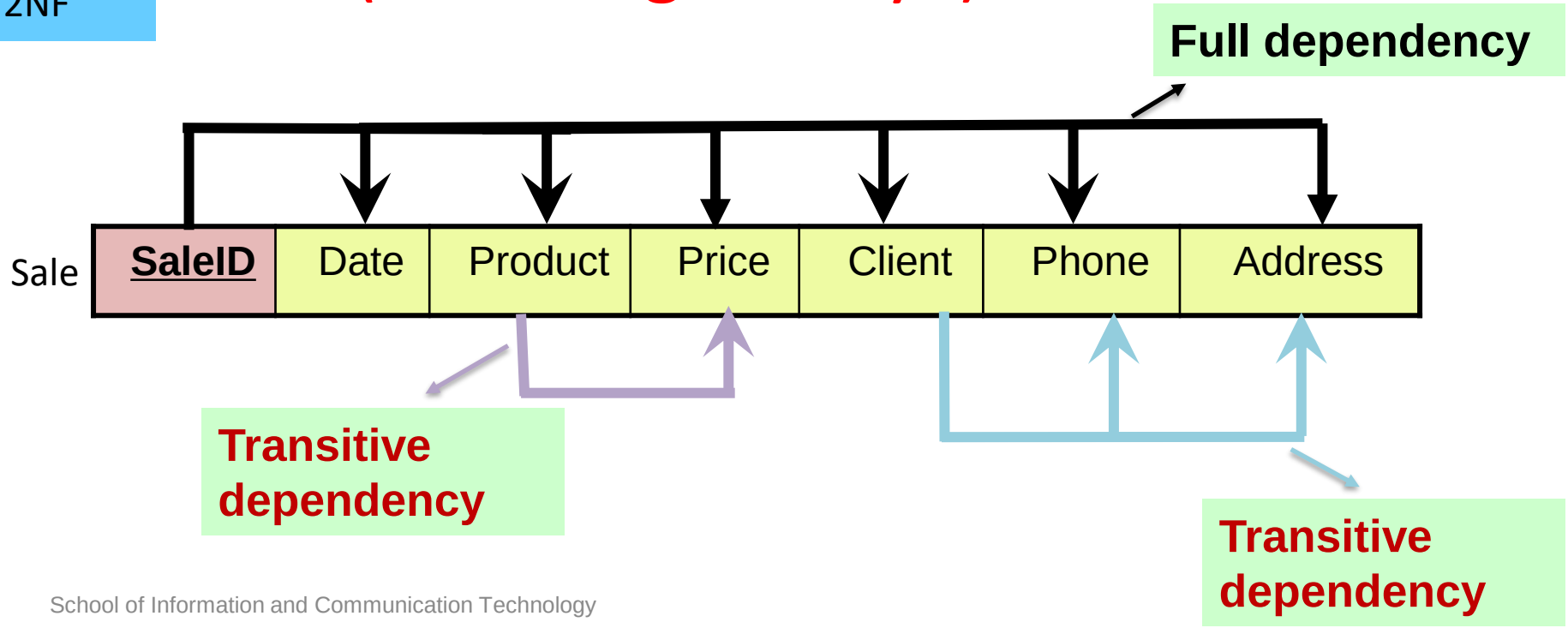


Example 5: Normalisation

- **Convert to 2NF**
  - Step 4: Remove partial dependency

(no change! Why?)

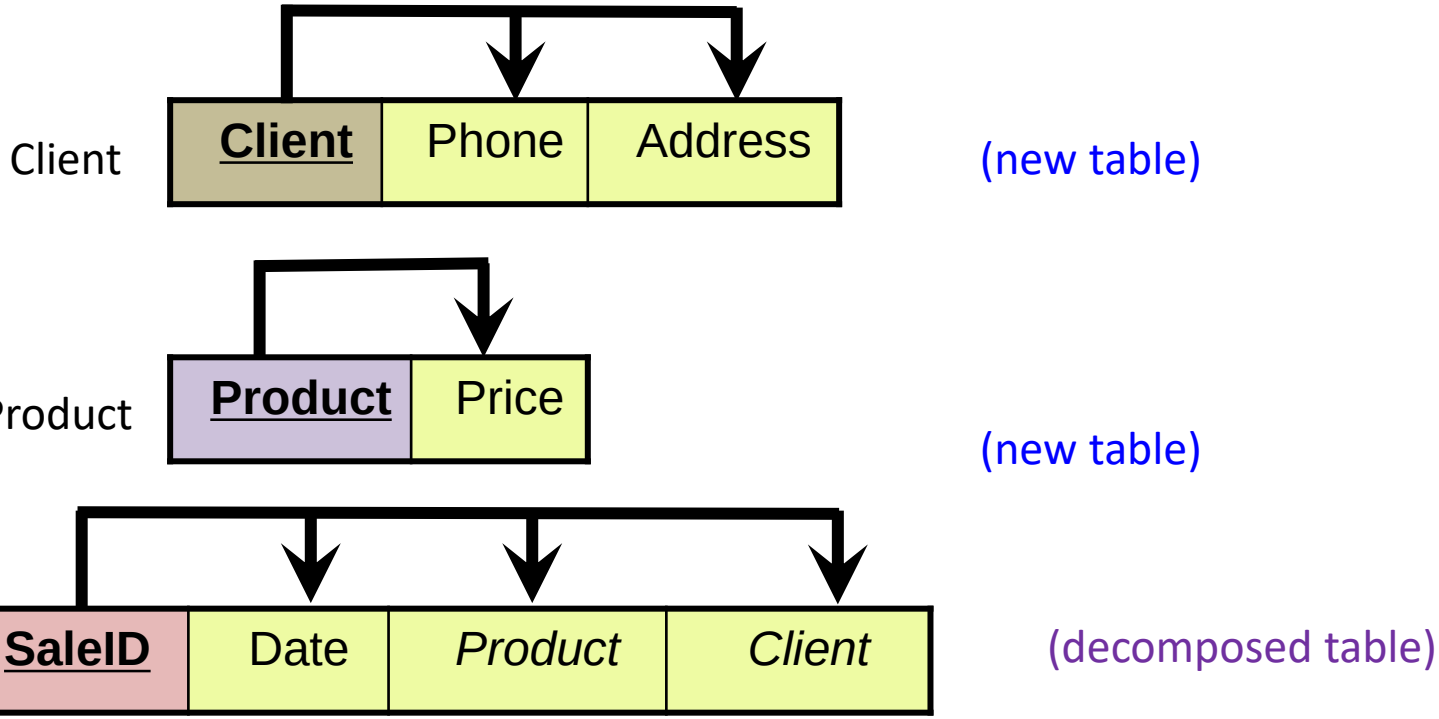
2NF



Example 5: Normalisation

- **Convert to 3NF**
  - Step 5: Remove transitive dependency

3NF



Example 5: Relation schema

UNF

Sale (SaleID, date, product, price, client, phone, address)

1NF

Sale (**SaleID**, date, product, price, client, phone, address)

2NF

Sale (**SaleID**, date, product, price, client, phone, address)

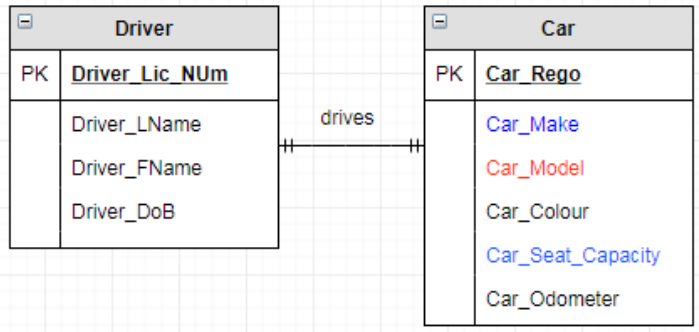
3NF

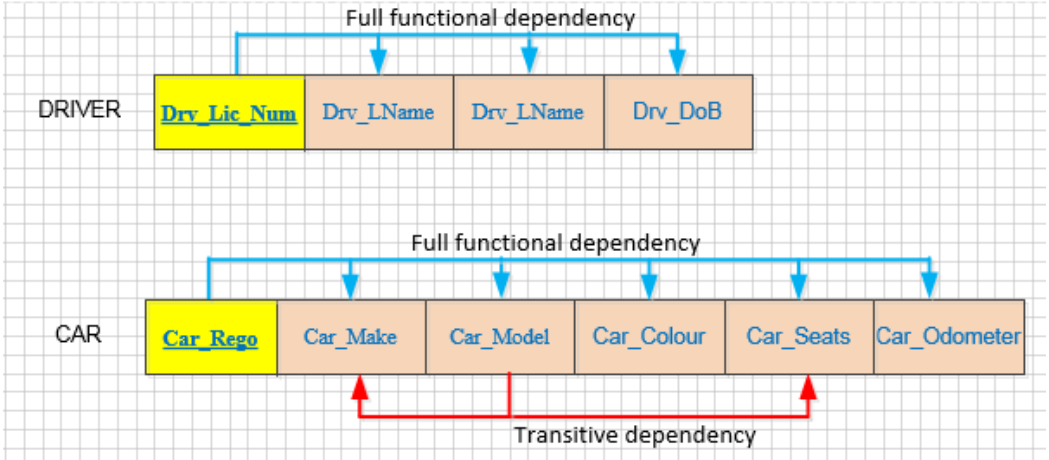
Product (**product**, price)
Customer (**client**, phone, address)
Sale (**SaleID**, date, *product*, *client*)

Relation not in 3NF

Example: Table not in 3NF

- In an ERD some entity (table) may not be in 3NF
- Depends on how far the normalisation is required by the business
- You can normalise further, if required





- Driver in 3NF
- Car is in 2NF, not in 3NF

- Toyota – Yaris – Ascent, SX, ZX
- Toyota – Corolla – Hatch, Sedan – Ascent, Sport, SX, Hybrid
- See https://www.toyota.com.au/cars

Thank you