Relational Databases

## Overview

An RDBMS is supposed to implement the relational model and provide the means to store, manage, enforce the integrity of, and query data.

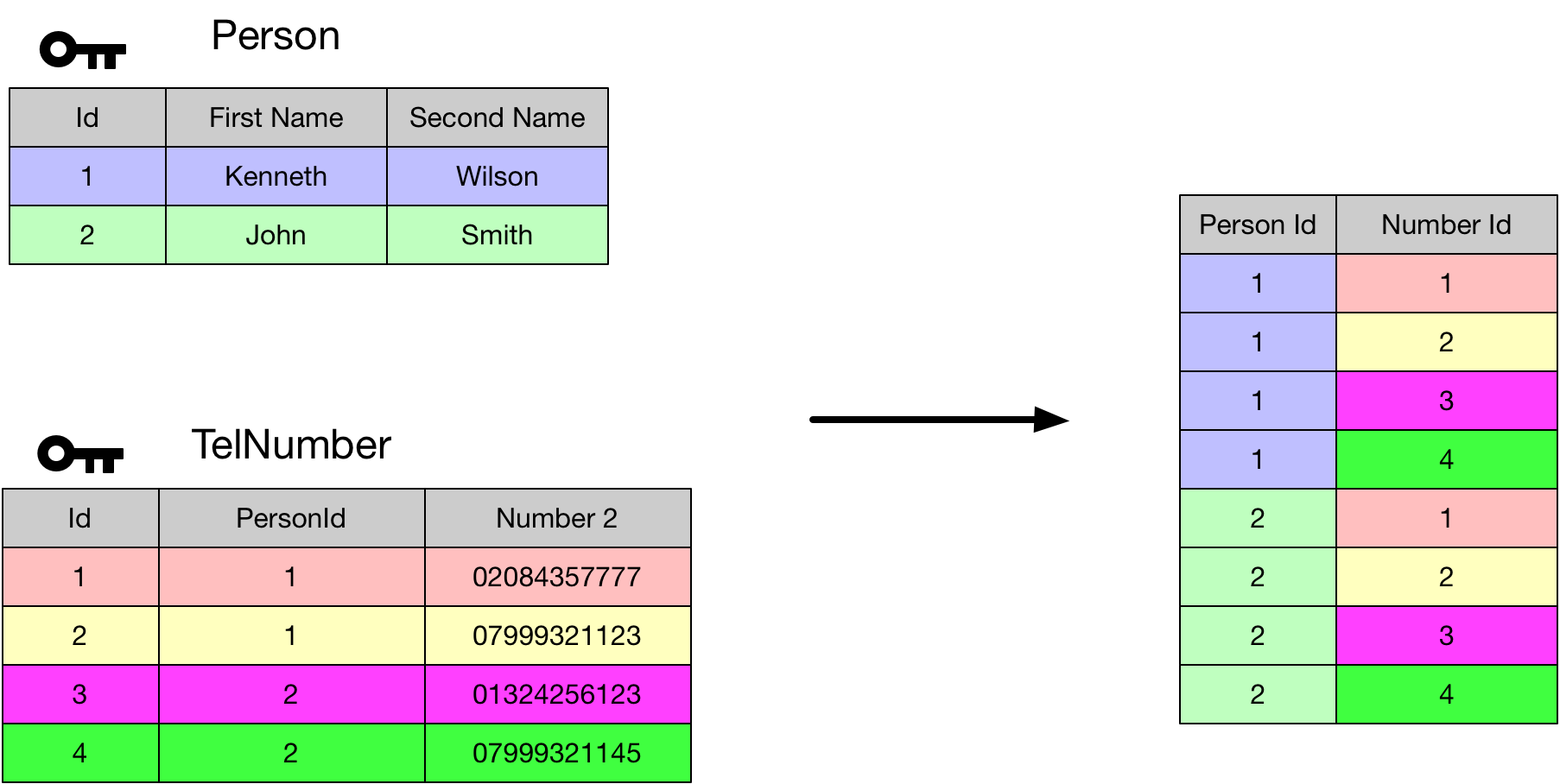
### Joins

#### Cross Join

SELECT p.id AS 'Person Id', T.id AS 'Number Id'

FROM Person AS P

CROSS JOIN TelNumber as T

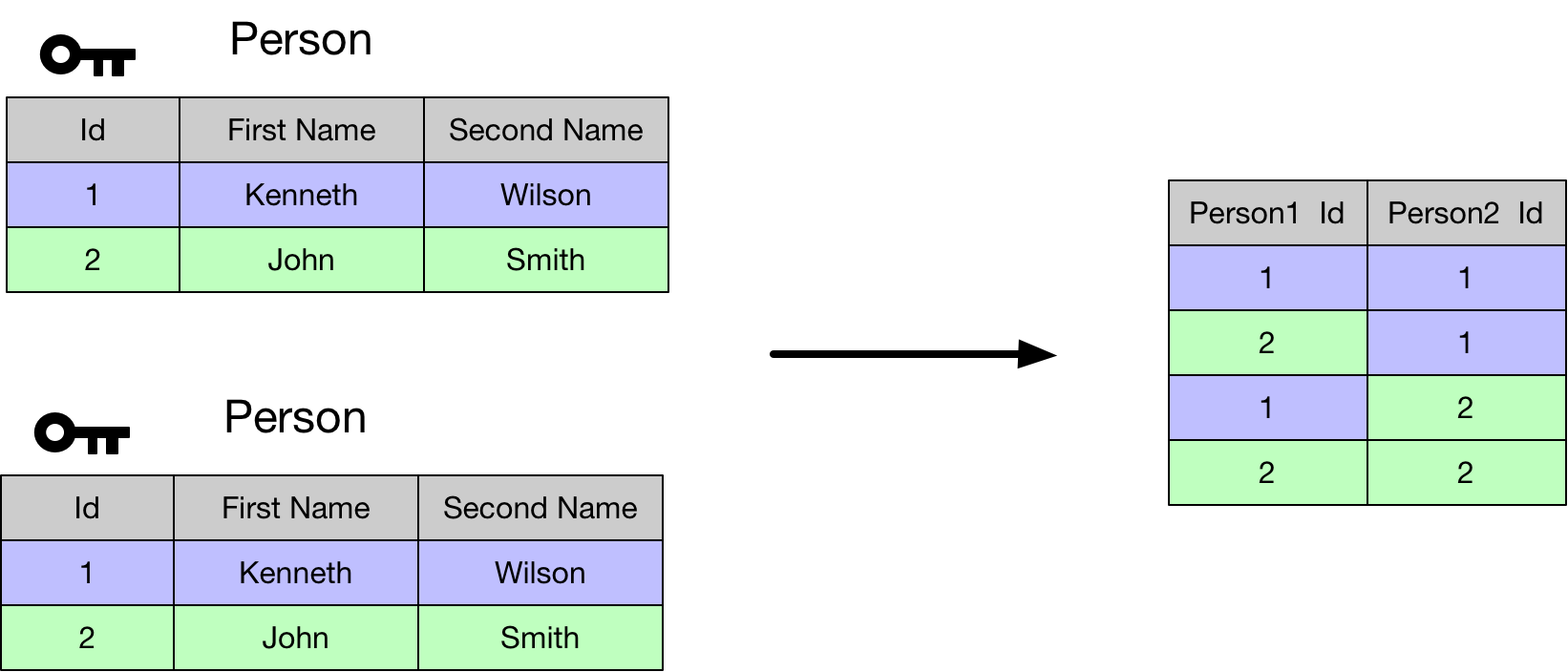


#### Self-Cross Join

SELECT P1.id AS 'Person1 Id', P2.id AS 'Person2 Id'

FROM Person AS P1

CROSS JOIN Person AS P2



#### Inner Join

SELECT

P.firstName AS 'First Name',

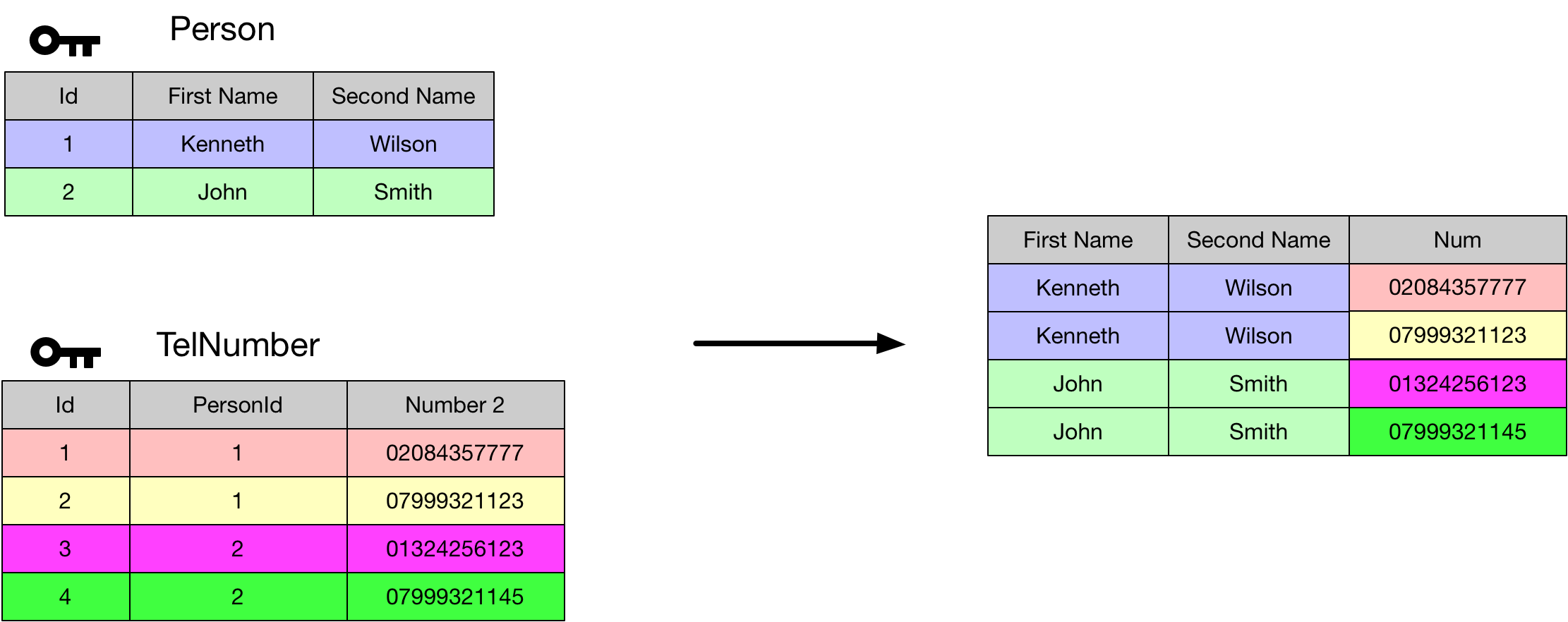
P.secondName AS 'Second Name',

T.telNumber AS 'Num'

FROM Person AS P

INNER JOIN TelNumber AS T

ON P.id = T.personId



#### Non Equi-Join

SELECT

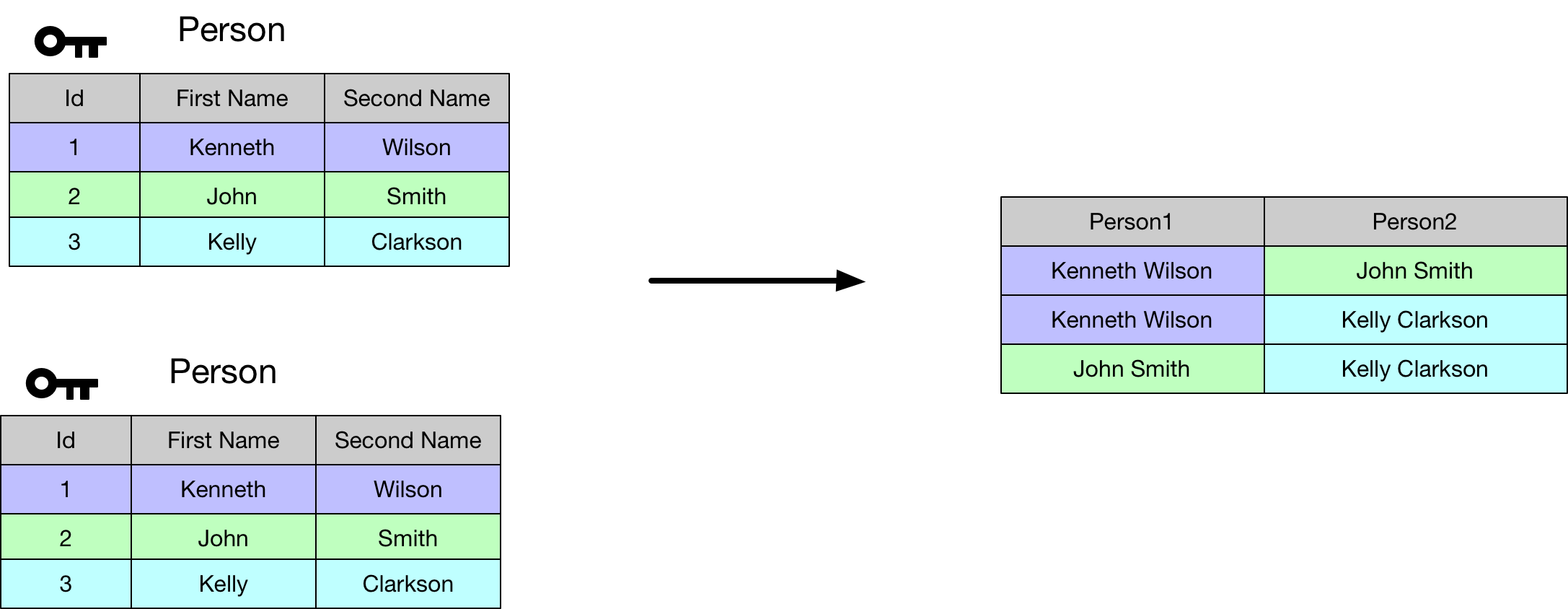
P1.firstName + ' ' + P1.secondName AS 'Person1',

P2.firstName + ' ' + P2.secondName AS 'Person2'

FROM Person AS P1

INNER JOIN Person AS P2

ON P1.id < P2.id



#### Left Outer Join

SELECT

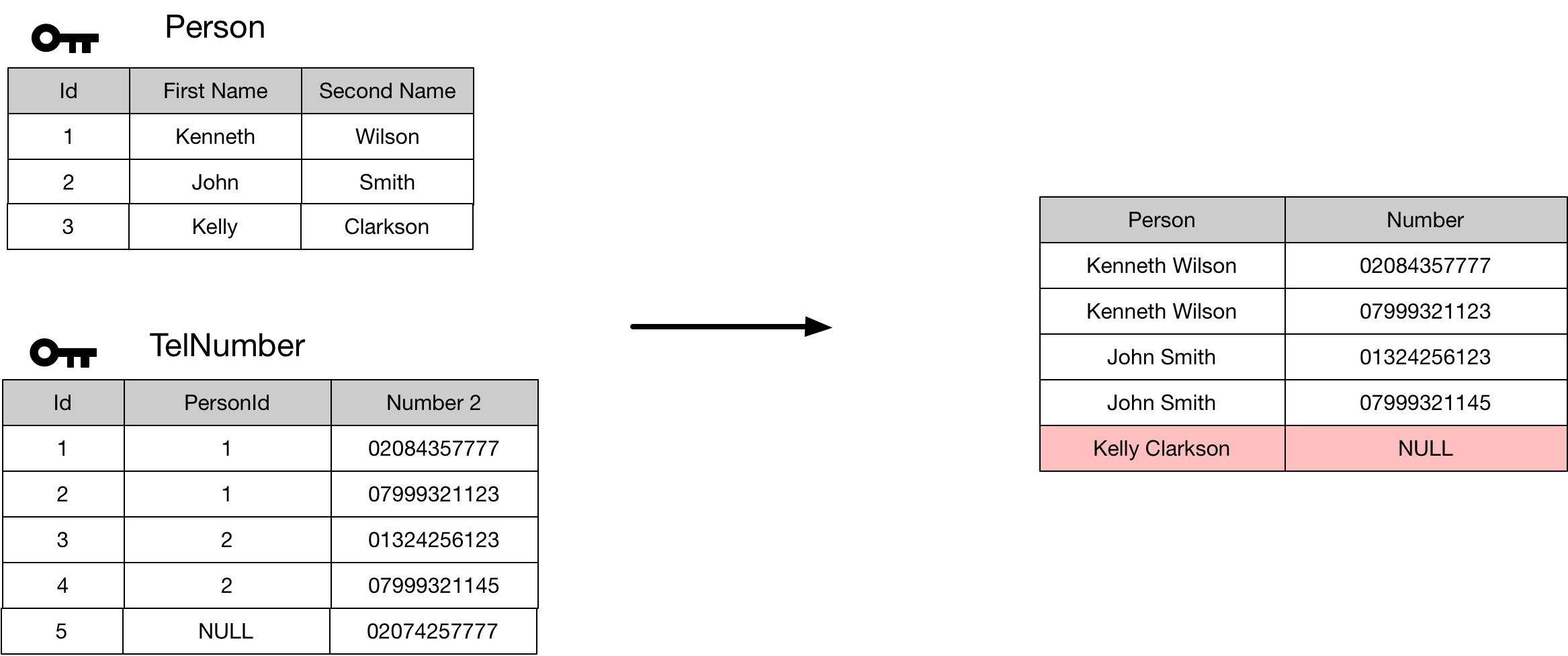
P.firstName + ' ' + P.secondName AS 'Person',

T.telNumber AS 'Number'

FROM Person AS P

LEFT OUTER JOIN TelNumber AS T

On P.id = T.personId



#### Right Outer Join

SELECT

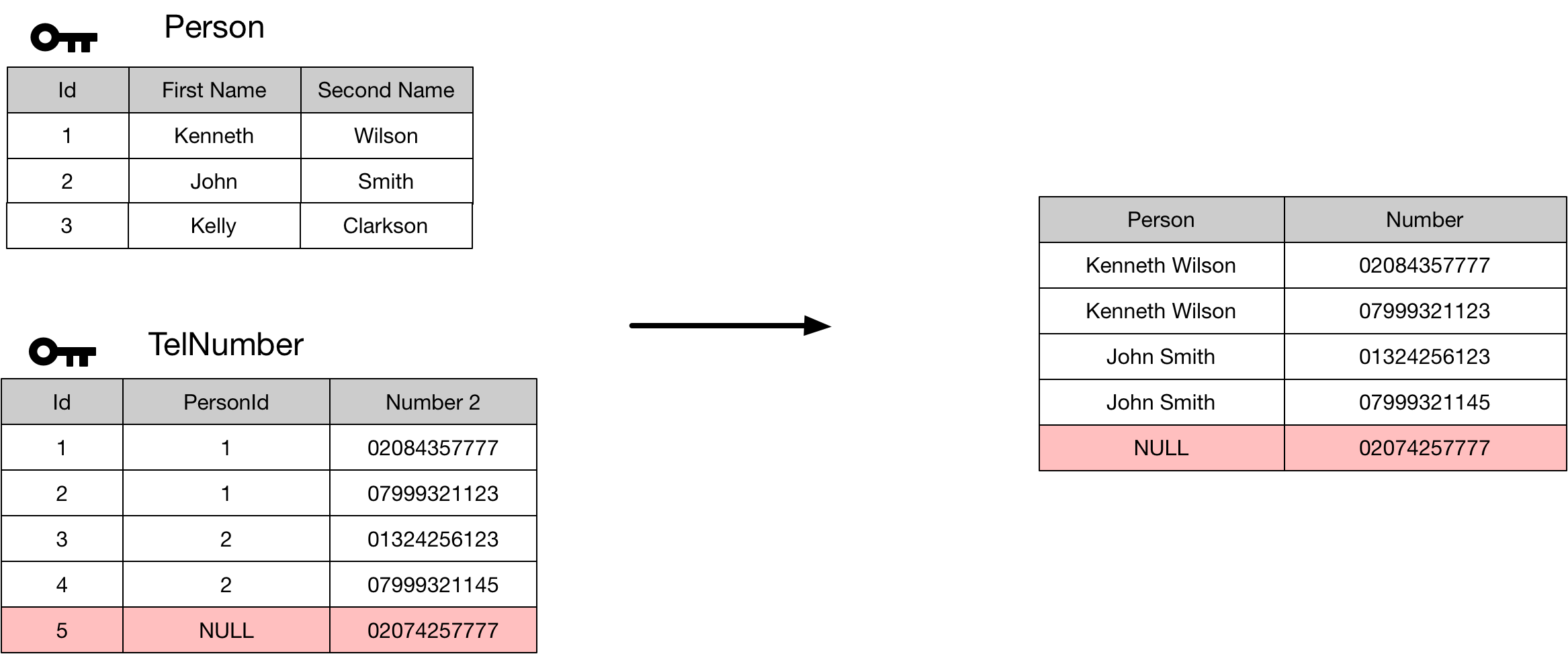
P.firstName + ' ' + P.secondName AS 'Person',

T.telNumber AS 'Number'

FROM Person AS P

RIGHT OUTER JOIN TelNumber AS T

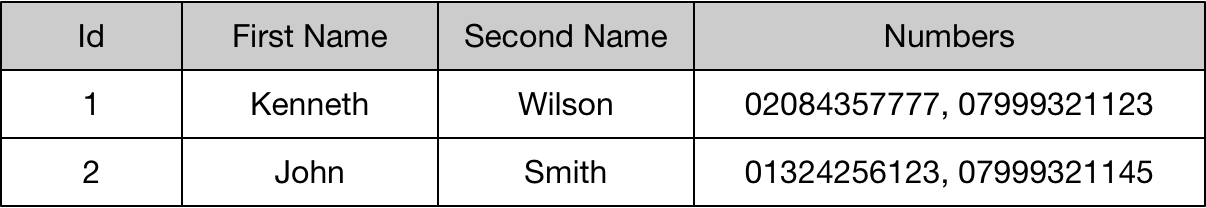
On P.id = T.personId



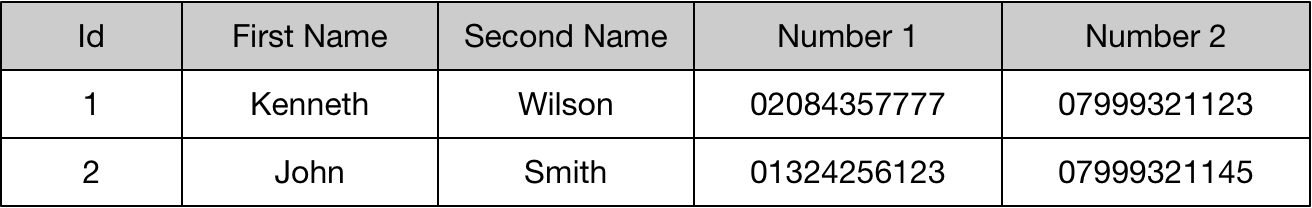
### Normal Forms

#### 1st Normal Form

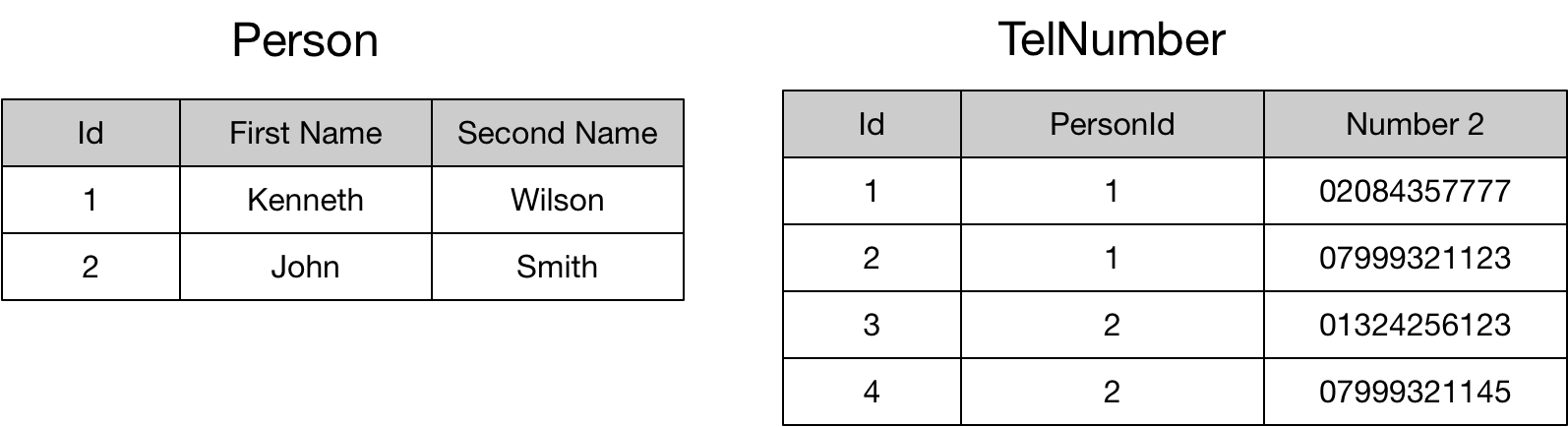
For a table to be in first normal form each cell must contain only one value from the domain. The following table violates this restriction



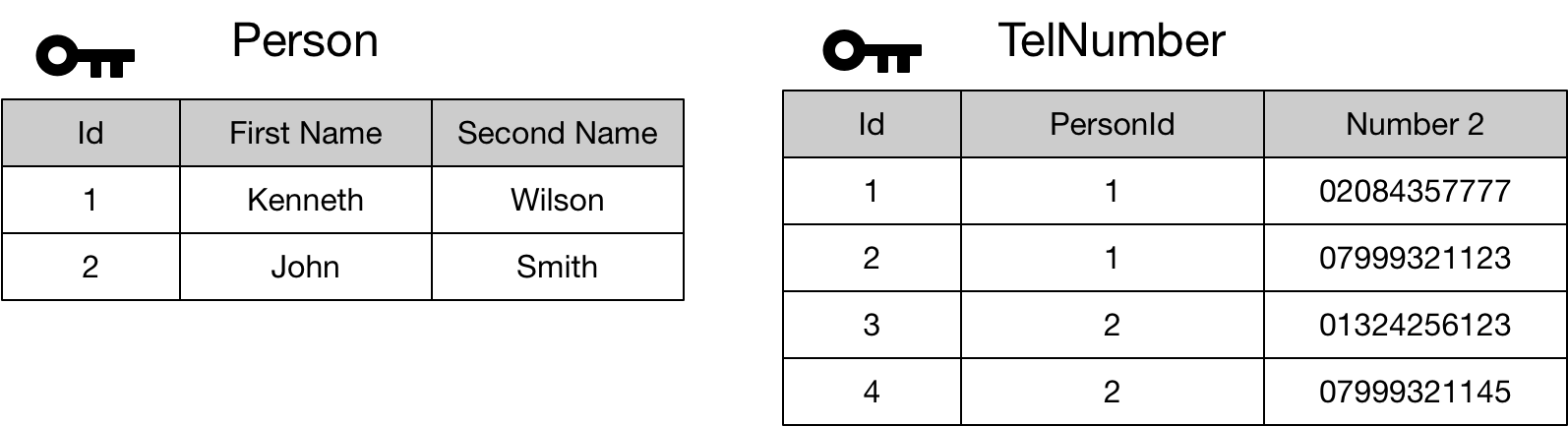
We should note the following table is still in violation of 1st normal form as repeating columns groups are also disallowed.



We can fix this using a schema something like the following. The PersonId field acts as the foreign key that indexes into the person table

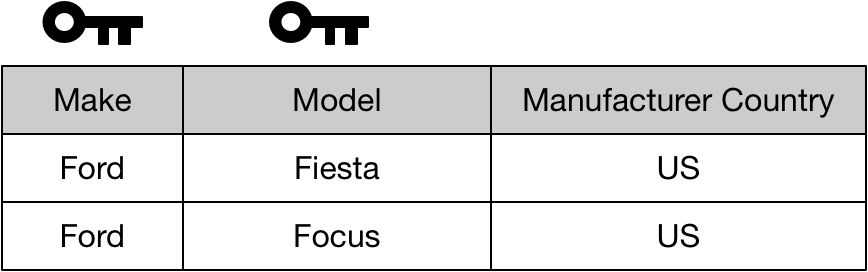


The second restriction on 1st normal form is that every row is unique. We can ensure uniqueness of rows by applying a candidate key to the row.

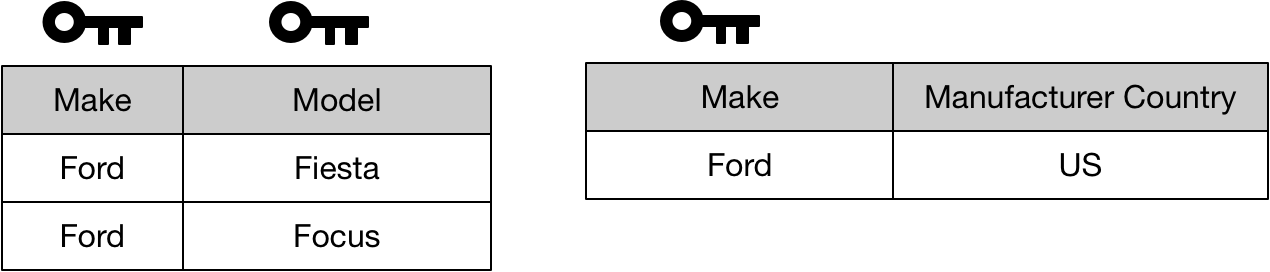


#### 2nd Normal Form

Second normal for applies to relations with composite keys. Where there is a composite key, we should not be able to locate the value of any non-key attribute using only part of the composite key. The following table is in violation of 2nd normal form.



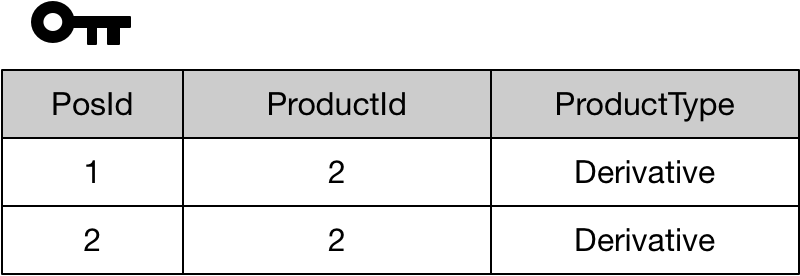
We can fix this as follows



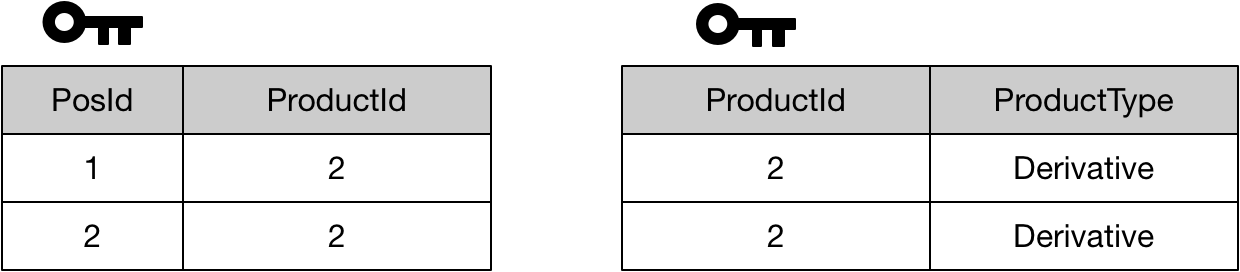
A relation with a single attribute primary key in 1st normal form is automatically in 2nd normal form.

#### 3rd Normal Form

To be in third normal form the relation must first be in second normal form. The second rule is that no non-key attribute can be identified by another non-key attribute. This table is in violation of 3rd normal form



We can fix this as follows.



We can summarise the second and third forms as meaning than in order to identify the value of any non-key attribute we need to use the full primary key. Furthermore, we cannot identify the value of any non-key field using another non-key field.

### Constraints

#### Primary Key Constraint

A primary key has the following properties

* Each table can have only one primary key
* The fields making up the primary key cannot be null
* The server creates an index to efficiently enforce uniqueness and retrieval

ALTER TABLE dbo.Products

ADD CONSTRAINT PK\_Products

PRIMARY KEY(prodId);

### SQL

#### Phases

➎ SELECT empid, YEAR(orderdate) AS orderyear, COUNT(\*) AS numorders

➊ FROM Sales.Orders

➋ WHERE custid = 71

➌ GROUP BY empid, YEAR(orderdate)

➍ HAVING COUNT(\*) > 1

➏ ORDER BY empid, orderyear;

|  |  |
| --- | --- |
| ➊ FROM | Specify the table we want to query |
| ➋ WHERE | Uses a predicate to filter the rows returned. Where clauses enable the use of indices and reduce network client compared to filtered whole tables on the client. |
| ➌ GROUPBY | Produce a group for each unique combination of values specified in this clause. |
| ➍ HAVING | Uses a predicate to filter the groups returned. Can utilise aggregate functions in the predicate. |
| ➎ SELECT | Specify the columns we want to see in the result |
| ➏ ORDER BY | Sort the rows for presentation purposes |

### Indices