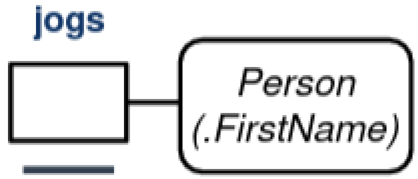
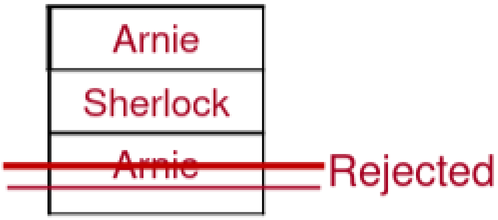
1. CSDP Step 4

> Step 4: Add uniqueness constraints and check arity of fact types.

Uniqueness Constraints

A uniqueness constraint means that the data values must be unique in a unary role. For example, if we have "Annie Sherlock", we can't have a second occurrence of "Annie Sherlock". This is because there is a uniqueness constraint. If there wasn't a uniqueness constraint, then we could have a second occurrence.

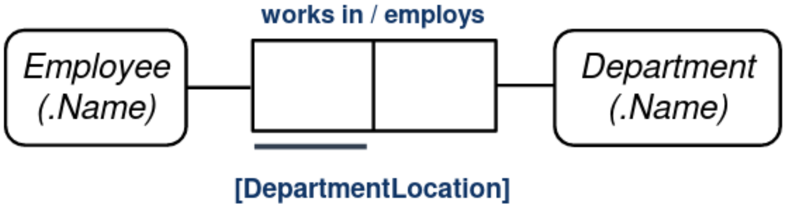




Binary Fact Types

In a binary fact type, we can have a uniqueness constraint over one end of the role or both parts of the role.

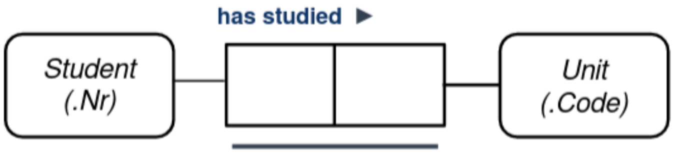
- If the uniqueness constraint is over one end of the role (e.g., the "works in" end of the role closest to "employee"), it means that an employee name may only occur once. However, a department name is not necessarily unique.



Mary Smith	Sales
Sally Wong	Accounts
David Hicks	Sales

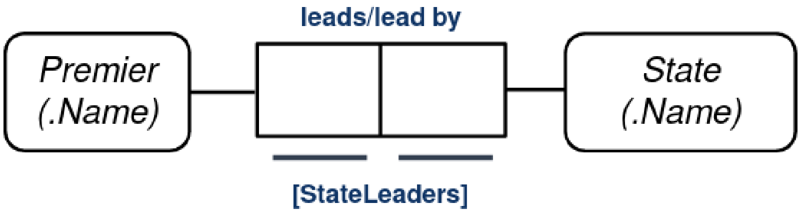
- Every **Employee** can occur only once and, therefore, each instance is unique.
- Every **Department** can occur more than once and, therefore, each instance is NOT unique.

- If the uniqueness constraint is over both parts of the role, it means that the combination of the two parts must be unique. For example, in a binary fact type involving "student" and "unit", the combination of student and unit must be unique.



0023456	ITN100
0023456	ITN200
0034567	ITN100
0034567	ITN300
0034567	ITN400

- To achieve uniqueness for all possible instances we ‘concatenate’ both roles.
- The uniqueness constraint spans both roles, i.e., the combination of **Student** and **Unit** is unique for this fact type.



Gladys Berejiklian	NSW
Annastacia Palaszczuk	QLD
Steven Marshall	SA
Will Hodgman	TAS
Daniel Andrews	VIC
Mark McGowan	WA

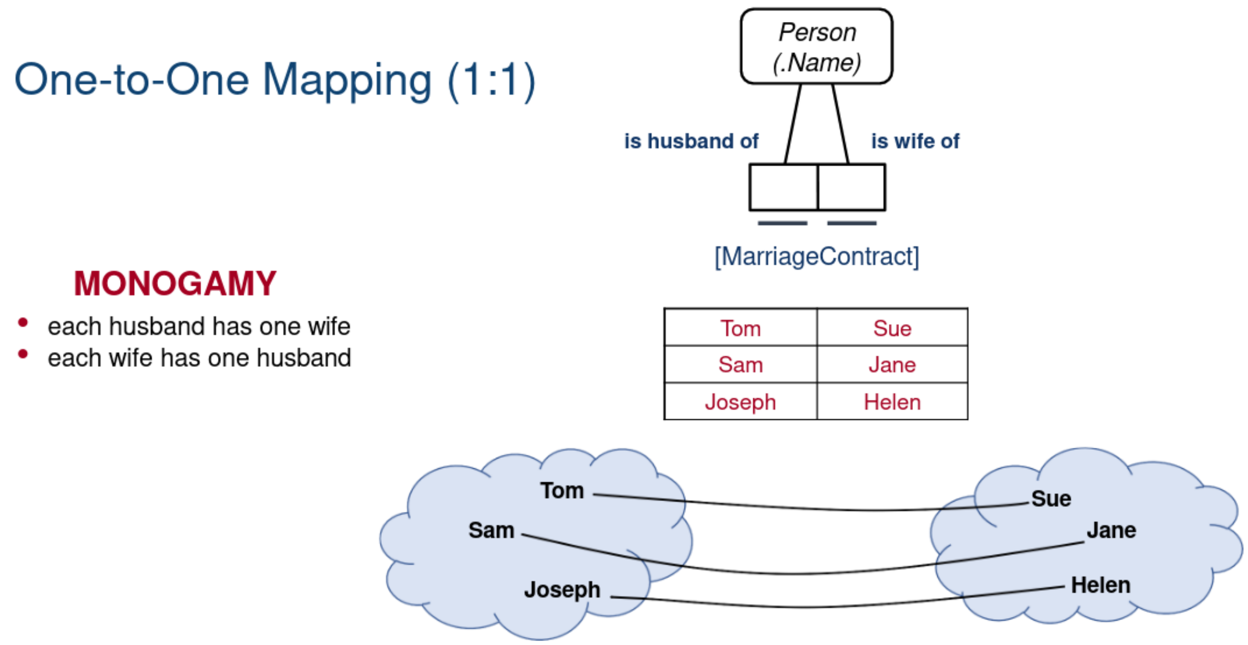
- A uniqueness constraint exists on each role: A **State** can have only one current **Premier**.
- A current **Premier** can be premier of only one **State**.

Mappings in Binary Fact Types

The uniqueness constraints in binary fact types lead to different types of mapping:

1. One-to-one mapping: Each entity on one side is uniquely related to an entity on the other side (e.g., each husband has one wife and each wife has one husband, representing

monogamy).



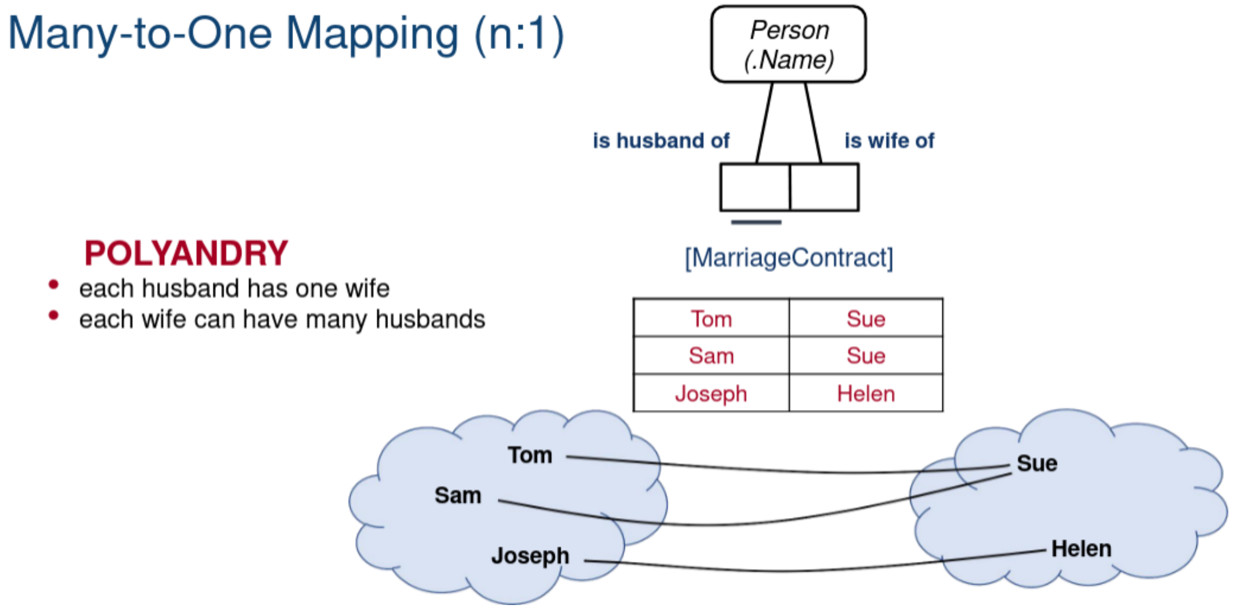
One-to-One Mapping (1:1)

MONOGAMY

- each husband has one wife
- each wife has one husband

Tom	Sue
Sam	Jane
Joseph	Helen

2. Many-to-one mapping: Each entity on one side can be related to multiple entities on the other side, but each entity on the other side is uniquely related to an entity on the first side (e.g., each husband has only one wife, but each wife can have many husbands, representing polyandry).



Many-to-One Mapping (n:1)

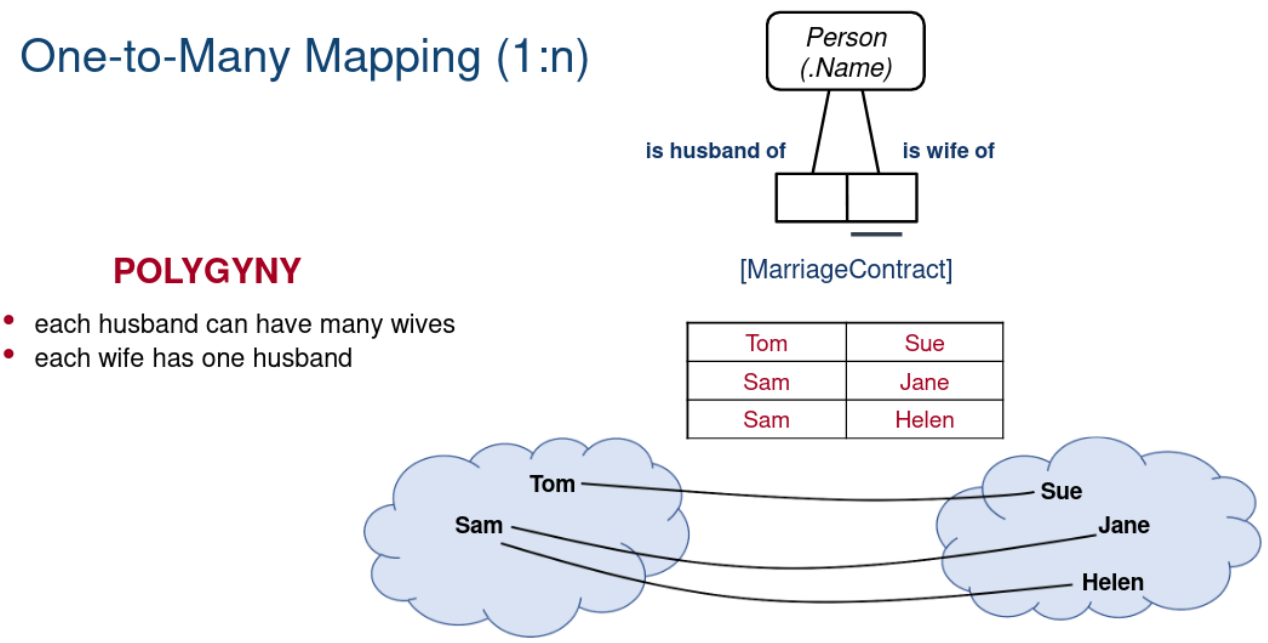
POLYANDRY

- each husband has one wife
- each wife can have many husbands

Tom	Sue
Sam	Sue
Joseph	Helen

3. One-to-many mapping: Each entity on one side is uniquely related to an entity on the other side, but each entity on the other side can be related to multiple entities on the first side (e.g., each husband can have many wives, but each wife has only one husband,

representing polygamy).



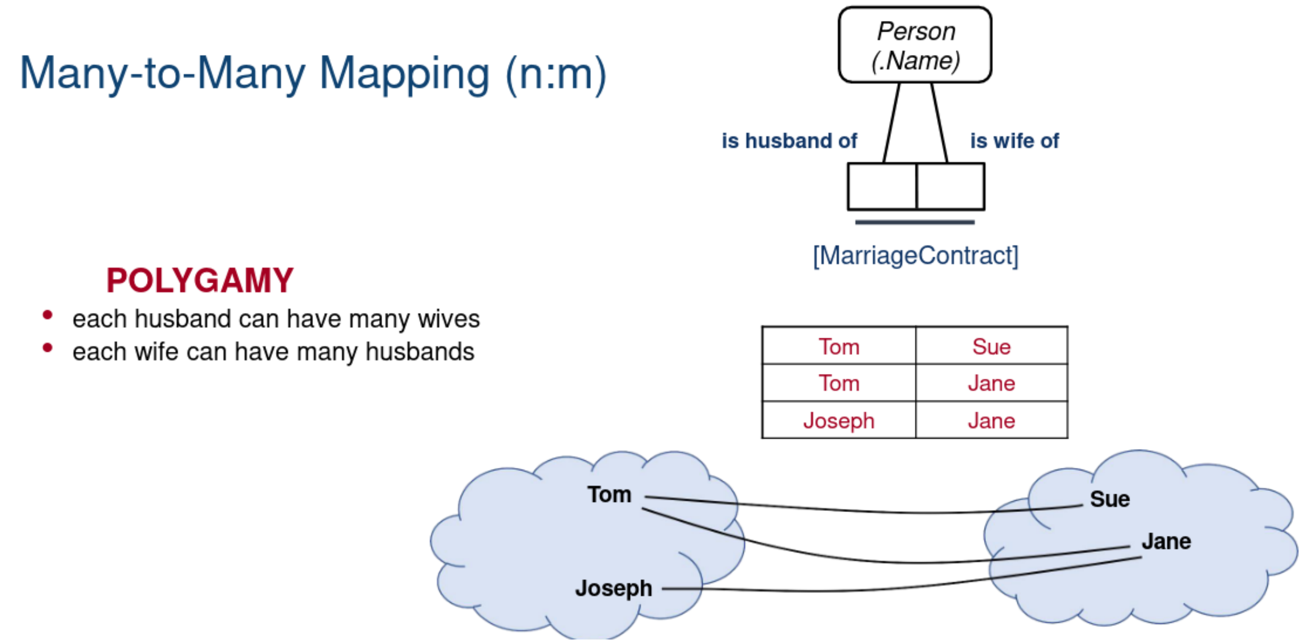
One-to-Many Mapping (1:n)

POLYGYNY

- each husband can have many wives
- each wife has one husband

Tom	Sue
Sam	Jane
Sam	Helen

4. Many-to-many mapping: Each entity on both sides can be related to multiple entities on the other side (e.g., each husband can have many wives and each wife can have many husbands, representing polygamy).



Many-to-Many Mapping (n:m)

POLYGAMY

- each husband can have many wives
- each wife can have many husbands

Tom	Sue
Tom	Jane
Joseph	Jane

Ternary Fact Types

In ternary fact types, the uniqueness constraint spans across multiple roles.

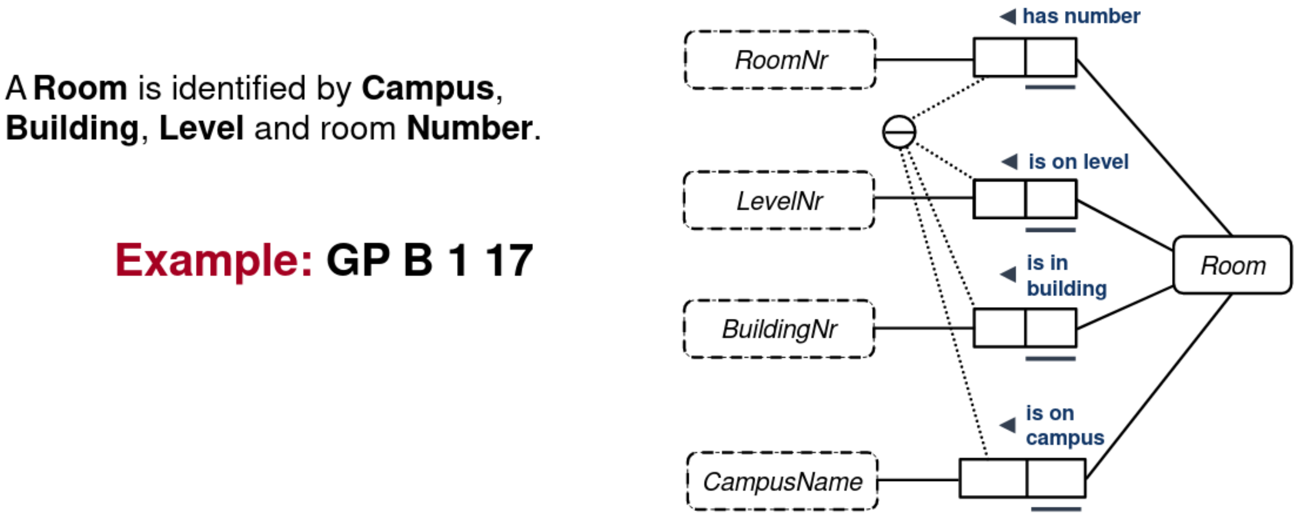
- The uniqueness constraint can span across two roles, meaning the combination of those two roles must be unique.
- The absence of a uniqueness constraint over a role is indicated by a dotted line.

Internal and External Uniqueness Constraints

There are two types of uniqueness constraints:

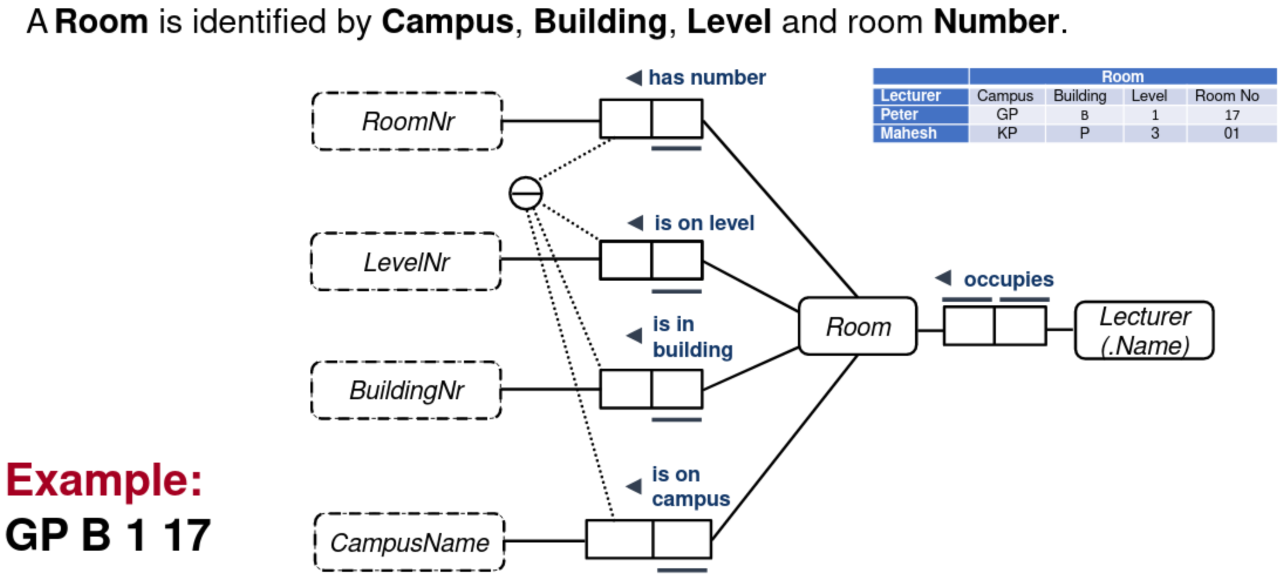
1. Internal uniqueness constraints: These apply to roles in any one predicate or one fact type. All the constraints discussed so far are internal uniqueness constraints.
2. External uniqueness constraints: These apply to roles from different predicates (two or more facts) and are called inter-predicate uniqueness constraints.

- Example: If we have a room entity type with room number, level number, building number, and campus name as value types, we can have an external uniqueness constraint spanning across all these roles, meaning the combination of room number, level number, building number, and campus name must be unique.



A **Room** is identified by **Campus**, **Building**, **Level** and room **Number**.

Example: **GP B 1 17**



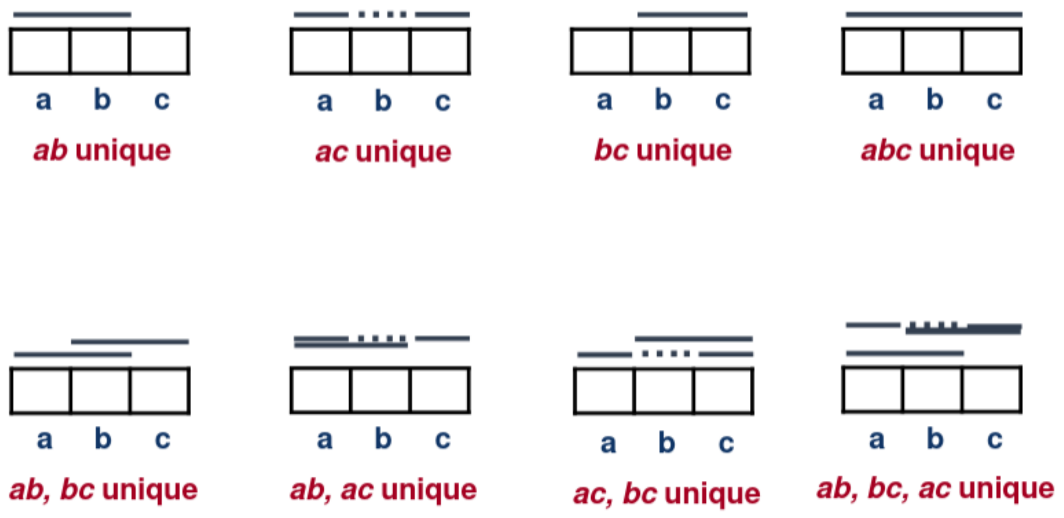
A **Room** is identified by **Campus**, **Building**, **Level** and room **Number**.

	Room			
Lecturer	Campus	Building	Level	Room No
Peter	GP	B	1	17
Mahesh	KP	P	3	01

Example:
GP B 1 17

Checking Arity Length of Fact Types

All Allowed Constraints for Ternary Facts



ab unique

ac unique

bc unique

abc unique

ab, bc unique

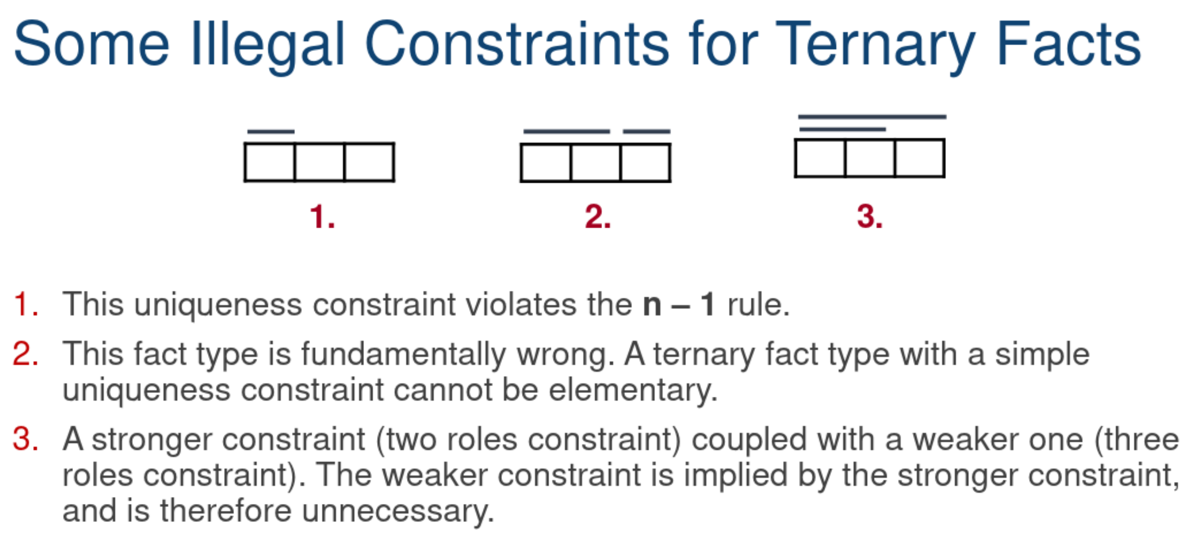
ab, ac unique

ac, bc unique

ab, bc, ac unique

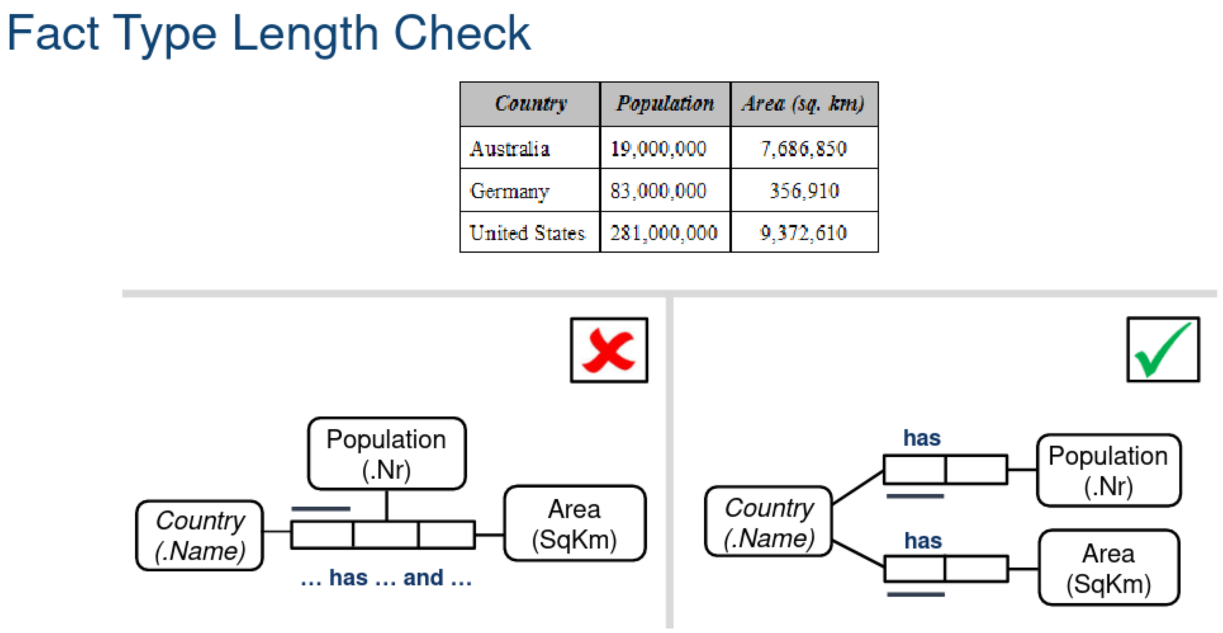
When checking the arity length of fact types:

- For an n-ary fact type, we should have exactly one uniqueness constraint that spans the whole fact type or at least one uniqueness constraint which spans at least n-1 roles (the "n-1 spanning rule").
- For ternary fact types, there are specific valid uniqueness constraints. Any constraint that violates the n-1 rule is considered illegal.



Some Illegal Constraints for Ternary Facts

1. This uniqueness constraint violates the **n – 1** rule.
2. This fact type is fundamentally wrong. A ternary fact type with a simple uniqueness constraint cannot be elementary.
3. A stronger constraint (two roles constraint) coupled with a weaker one (three roles constraint). The weaker constraint is implied by the stronger constraint, and is therefore unnecessary.



Fact Type Length Check

Country	*Population*	*Area (sq. km)*
Australia	19,000,000	7,686,850
Germany	83,000,000	356,910
United States	281,000,000	9,372,610

Nested Fact Types

If we have a nested fact type, the spanning rule must apply, and the uniqueness constraint must span all the roles within the nested fact type.

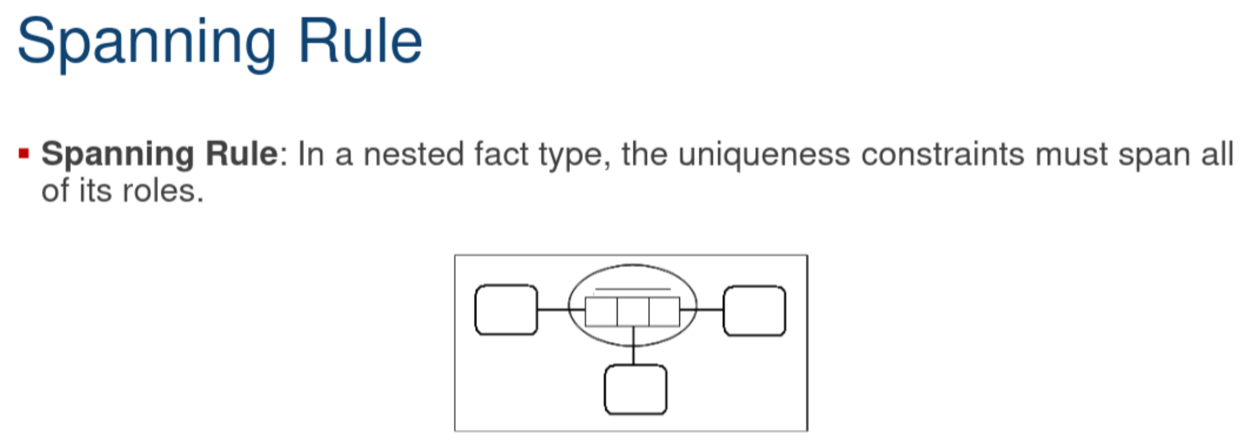
- Example: If we have a nested fact type that includes "subject" and "credit" with a binary fact type connected to "lecturer", the uniqueness constraint must span across all roles within the nested fact type to meet the n-1 rule.

Remember, when flattening the fact type, ensure that the resulting schema has the correct uniqueness constraints and meets the n-1 rule.

Key points:

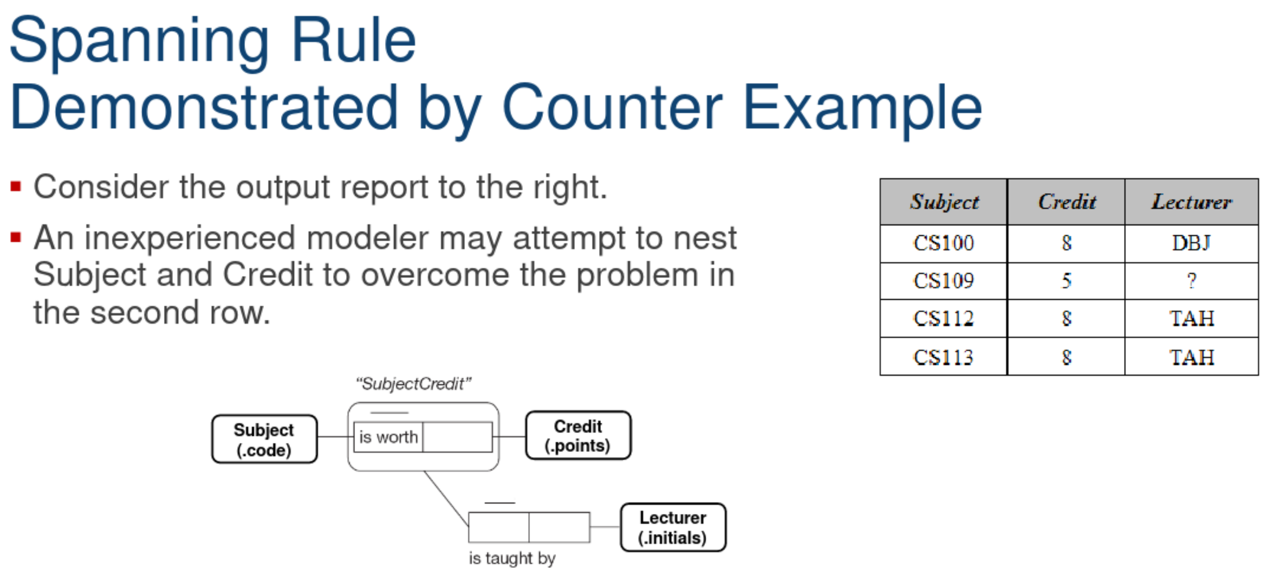
- Uniqueness constraints ensure data values are unique in a role.

- Different mappings in binary fact types: one-to-one, many-to-one, one-to-many, many-to-many.
- Internal uniqueness constraints apply to roles in one fact type, while external uniqueness constraints apply to roles from different predicates.
- For n-ary fact types, at least one uniqueness constraint should span n-1 roles (the "n-1 spanning rule").



Spanning Rule

- **Spanning Rule**: In a nested fact type, the uniqueness constraints must span all of its roles.



Spanning Rule
Demonstrated by Counter Example

- Consider the output report to the right.
- An inexperienced modeler may attempt to nest Subject and Credit to overcome the problem in the second row.

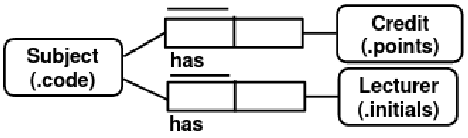
Subject	*Credit*	*Lecturer*
CS100	8	DBJ
CS109	5	?
CS112	8	TAH
CS113	8	TAH

- Nested fact types must have uniqueness constraints spanning all roles within the nested fact type.

The Correct Schema

- Our elementary facts are:
  - Subject (code) ‘CS100’ is worth Credit (points) 8
  - Subject (code) ‘CS100’ is taught by Lecturer (initials) ‘DBJ’

- So, our schema becomes:



See Also

2. CSDP Step 5