Armstrong axioms are a complete set of inference rules or axioms, introduced and developed by William W. Armstrong in 1974. The inference rules are sound which is used to test logical inferences of functional dependencies. The axiom which also refers to as sound is used to infer all the functional dependencies on a relational database. The Axioms are a set of rules, that when applied to a specific set, generates a closure of functional dependencies.

Armstrong's Axioms has two different set of rules,

1. Axioms or primary rules
   a. Axiom of Reflexivity
   b. Axiom of Augmentation
   c. Axiom of Transitivity
2. Additional rules or Secondary rules
   a. Union
   b. Composition
   c. Decomposition
   d. Pseudo Transitivity

1) Axioms or primary rules

Let suppose **T (k)** with the set of attributes **k** be a relation scheme. Subsequently, we will represent subsets of **k** as **A**, **B**, **C**. The standard notation in database theory for the set of attributes is **AB** rather than **A∪B**.

a. **Axiom of Reflexivity**:
   If a set of attributes is P and its subset is **Q**, then **P** holds **Q**. If **Q ⊆ P**, then **P** → **Q**. This property is called as Trivial functional dependency.
   Where **P** holds **Q** (**P** → **Q**) denote **P** functionally decides **Q**.
b. **Axiom of Augmentation**:
   If **P** holds **Q** (**P** → **Q**) and **R** is a set of attributes, then **PR** holds **QR** (**PR** → **QR**). It means that a change in attributes in dependencies does not create a change in basic dependencies. If **P** → **Q**, then **PR** → **QR** for any **R**.
c. **Axiom of Transitivity**:
   If **P** holds **Q** (**P** → **Q**) and **Q** holds **R** (**Q** → **R**), then **P** hold **R** (**P** → **R**). Where **P** holds **R** (**P** → **R**) denote **P** functionally decides **R**, same with **P** holds **Q** and **Q** holds **R**.

2) Additional rules or secondary rules

These rules can be derived from the above axioms.

a. **Union**:
   If **P** holds **Q** (**P** → **Q**) and **P** holds **R** (**P** → **R**), then **P** → **QR**. If **X** → **Y** and **X** → **Z**, then **X** → **YZ**.
b. **Composition**:
   If **P** holds **Q** (**P** → **Q**) and **A** holds **B** (**A** → **B**), then **PA** → **QB**.
   proof,
   1. P → Q (Given)
   2. A → B (Given)
   3. PA → QA (Augmentation of 1 and A)
   4. PA → Q (Decomposition of 3)
   5. PA → PB (Augmentation of 2 and P)
   6. PA → B (Decomposition of 5)
   7. PA → QB (Union 4 and 6)
c. **Decomposition**:
   This rule is contrary of union rule. If **P** → **QR**, then **P** holds **Q** (**P** → **Q**) and **P** holds **R** (**P** → **R**). If **X** → **YZ**, then **X** → **Y** and **X** → **Z**.
   proof,
   1. P → QR (Given)
   2. QR → Q (Reflexivity)
   3. P → Q (Transitivity of 1 and 2)
d. **Pseudo Transitivity**:
   If **P** → **RQ** and **Q** → **S**, then **P** → **RS**.
   proof,
   1. P → RQ (Given)
   2. Q → S (Given)
   3. RQ → RS (Augmentation of 2 and R)
   4. P → RS (Transitivity of 1 and 3)

Trivial Functional Dependency

Trivial	If **P** holds **Q** (**P** → **Q**), where **P** is a subset of **Q**, then it is called a Trivial Functional Dependency. Trivial always holds Functional Dependency.
Non-Trivial	If **P** holds **Q** (**P** → **Q**), where **Q** is not a subset of **P**, then it is called as a Non-

	Trivial Functional Dependency.
Completely Non-Trivial	If **P** holds **Q** (**P** → **Q**), where **P** intersect **Y = Φ**, it is called as a Completely Non-Trivial Functional Dependency.