Normalization Convertor and SQL Schema Generator (NC SSG)

NC SSG is a tool which normalizes a relation up to BCNF and generate its SQL schema. The process of normalization starts from Minimal cover result in our tool as its result makes functional dependencies set irreducible by removing trivial dependencies.

1. Minimal Cover:

It is the irreducible set of functional dependencies. It is actually the minimal set of dependencies that cannot be further reduced. Its process is as follow:

a. Convert RHS attribute into singleton attribute.

b. Remove the extra LHS attribute.

c. Remove the redundant FDs.

Example:

Current functional dependencies:

Current Functional Dependencies

1. ssn → pnum
2. Name → ssn
3. ssn → Name
4. ssn → Email1
5. Name → Email1
6. ssn → Email2
7. Name → Email2
8. Email1 → Email2
9. Name, Email1 → Email2
10. ssn → Address1
11. ssn → Address2
12. ssn → DId
13. ssn → ploc
14. ssn → pname
15. ssn → dnum
16. ssn → dname

Minimal cover result:

Minimal Cover Result

1. ssn → pnum
2. Name → ssn
3. ssn → Name
4. ssn → Email1
5. ssn → Email2
6. ssn → Address1
7. ssn → Address2
8. ssn → DId
9. ssn → ploc
10. ssn → pname
11. ssn → dnum
12. ssn → dname

2. First Normal Form:

It states that the domain of an attribute must include only atomic (simple, indivisible) values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. Hence, 1NF disallows having a set of values, a tuple of values, or a combination of both as an attribute value for a single tuple.

Example:

Minimal Cover Result:

1. A, E --> B
2. A --> C
3. A --> D
4. C --> E

E is a multivalued Attribute so 1NF will make it the part of Primary Key

FULL Dependent Relations

Example:	A	B	C	D	E

Reason:
Multi-value Attribute => {E}.
The relation contains multi-valued attribute, so those attributes will become the part of primary key to create a composite primary key to make sure that all the tuples are unique.

3. **Second Normal Form:**
A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.

1NF Result:

FULL Dependent Relations

Example:	A	B	C	D	E

After checking for partial dependency the above table will be dived into two tables:

FULL Dependent Relations

Example:	A	C	D

MULTI-VALUED Relations

Exa_E:	C	E

Reason:
No partial dependency in the relation, so relation would be the same.
Steps:
Round 1: The FD **B** --> **A** is a partial dependency As LHS **B** is a proper subset of **A** which is a PK , so we split it
Round 2: The FD **A** --> **B** is already in 2NF

Round 3: The FD **B** --> **C** is a partial dependency As LHS **B** is a proper subset of **A** which is a PK , so we split it
Round 4: The FD **A** --> **D** is already in 2NF
Round 5: The FD **C** --> **E** is a partial dependency As LHS **C** is a proper subset of **A** which is a PK , so we split it

4. **Third Normal Form:**
According to Codd's original definition, a relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.

2NF Result:

FULL Dependent Relations

Example:

<u>A</u>	C	D

MULTI-VALUED Relations

Exa_E:

<u>C</u>	<u>E</u>

After Checking for Transitive dependency the result of 3rd Normal form is:

FULL Dependent Relations

Example:

A	B	D

MULTI-VALUED Relations

Exa_E:

C	E

TRANSITIVE Dependent Relations

Exa_B:

B	C

Reason:
3rd-NF works on transitive dependency. As attribute set => {C} is transitive dependent on set => {B}. So, it would be separated as new relation.
Steps:
Round 1: The FD **B** --> **A** is a transitive dependency As LHS **B** is not Primary Key, so we split it into a new relation.
Round 2: The FD **A** --> **B** is already in 3NF.
Round 3: The FD **B** --> **C** is a transitive dependency As LHS **B** is not Primary Key, so we split it into a new relation
Round 4: The FD **A** --> **D** is already in 3NF.
Round 5: The FD **C** --> **E** is a transitive dependency As LHS **C** is not Primary Key, so we split it into a new relation.

5. **Boyce-Codd normal form**
A relation schema R is in BCNF if whenever a nontrivial functional dependency X → A holds in R, then X is a super key of R.

3rd Normal Form Result:

FULL Dependent Relations

Example:

A	B	D

MULTI-VALUED Relations

Exa_E:

C	E

TRANSITIVE Dependent Relations

Exa_B:

B	C

Non trivial Functional Dependency:

In **Non-trivial functional dependency**, the dependent is strictly not a subset of the determinant.

If **X → Y** and **Y is not a subset of X**, then it is called Non-trivial functional dependency.

After checking of non trivial Functional Dependency:

FULL Dependent Relations

Example:	A	B	D

MULTI-VALUED Relations

Exa_E:	C	E

PRIMEDEPENDENCY Dependent Relations

Exa_B:	B	A

TRANSITIVE Dependent Relations

Exa_B:	B	C

Reason:
BC-NF works on Primary key dependency. As attribute set
=> {A} is dependent on set => {B}. So, it would be separated as new relation.

Steps:
Round 1: BC-NF works on Primary key dependency. As in FD **B** --> **A**, LHS **B** is PK so it would be separated as new relation.
Round 2: The FD **A** --> **B** is already in BCNF
Round 3: The FD **B** --> **C** is already in BCNF
Round 4: The FD **A** --> **D** is already in BCNF
Round 5: The FD **C** --> **E** is already in BCNF