

TABELA STATES

1FN

Como todos os atributos desta tabela já são atômicos, a primeira forma normal já está aplicada.

2FN

States

(idState, stateName, stateAbbv, idRegion <FK>, stateFlag)

Com a primeira forma normal aplicada, pode-se verificar na tabela se existem colunas que não são funcionalmente dependentes da(s) chave(s) primária(s) desta tabela.

A tabela em questão já se encontra na segunda forma normal, uma vez que todos seus atributos são funcionalmente dependentes do idState, que rege todas as outras informações do estado, como abreviação, nome e bandeira. O outro atributo presente nessa tabela é uma chave estrangeira para a tabela de regiões.

3FN

Analisando a tabela para ver se a mesma se apresenta na terceira forma normal, devemos “ignorar” a chave primária e ver se algum dos atributos é dependente de um outro atributo desta mesma tabela.

States

(____, stateName, stateAbbv, idRegion <FK>, stateFlag)

Como todos os atributos dessa tabela são definidos por idState, não há necessidade de gerar nenhuma tabela a partir dessa, ou seja, essa tabela já se encontra na terceira forma normal.

TABELA CITIES

1FN

Como todos os atributos desta tabela já são atômicos, a primeira forma normal já está aplicada.

2FN

Cities

(idCity, cityName, idState <FK>)

Com a primeira forma normal aplicada, pode-se verificar na tabela se existem colunas que não são funcionalmente dependentes da(s) chave(s) primária(s) desta tabela.

A tabela em questão já se encontra na segunda forma normal, uma vez que todos seus atributos são funcionalmente dependentes do idCity, que rege todas as outras informações da cidade, como seu nome. O outro atributo presente nessa tabela é uma chave estrangeira para a tabela de estados.

3FN

Analisando a tabela para ver se a mesma se apresenta na terceira forma normal, devemos “ignorar” a chave primária e ver se algum dos atributos é dependente de um outro atributo desta mesma tabela.

Cities

(___, cityName, idState <FK>)

Como todos os atributos dessa tabela são definidos por idCity, não há necessidade de gerar nenhuma tabela a partir dessa, ou seja, essa tabela já se encontra na terceira forma normal.

TABELA ADDRESSES

1FN

Como todos os atributos desta tabela já são atômicos, a primeira forma normal já está aplicada.

2FN

Addresses

(idAddress, addressName, idCity <FK>)

Com a primeira forma normal aplicada, pode-se verificar na tabela se existem colunas que não são funcionalmente dependentes da(s) chave(s) primária(s) desta tabela.

A tabela em questão já se encontra na segunda forma normal, uma vez que todos seus atributos são funcionalmente dependentes do idAddress, que rege todas as outras informações da tabela, como seu nome (que no caso é a string que contém o endereço propriamente dito). O outro atributo presente nessa tabela é uma chave estrangeira para a tabela de cidades.

3FN

Analisando a tabela para ver se a mesma se apresenta na terceira forma normal, devemos “ignorar” a chave primária e ver se algum dos atributos é dependente de um outro atributo desta mesma tabela.

Addresses

(_____, addressName, idCity <FK>)

Como todos os atributos dessa tabela são definidos por idAddresses, não há necessidade de gerar nenhuma tabela a partir dessa, ou seja, essa tabela já se encontra na terceira forma normal.

TABELA PEOPLE

1FN

Como todos os atributos desta tabela já são atômicos, a primeira forma normal já está aplicada.

2FN

People

(idPerson, personFirstName, personLastName, personBirthDate, personSex, idAddress <FK>, idEducation <FK>)

Com a primeira forma normal aplicada, pode-se verificar na tabela se existem colunas que não são funcionalmente dependentes da(s) chave(s) primária(s) desta tabela.

A tabela em questão já se encontra na segunda forma normal, uma vez que todos seus atributos são funcionalmente dependentes do idPerson, além de suas chaves estrangeiras para as tabelas de endereços e educação.

3FN

Analisando a tabela para ver se a mesma se apresenta na terceira forma normal, devemos “ignorar” a chave primária e ver se algum dos atributos é dependente de um outro atributo desta mesma tabela.

People

(____, personFirstName, personLastName, personBirthDate, personSex, idAddress <FK>, idEducation <FK>)

Assim como nas outras tabelas descritas acima, fica evidente que essa tabela está completamente normalizada.

TABELA CARGO_STATES

1FN

Como todos os atributos desta tabela já são atômicos, a primeira forma normal já está aplicada.

2FN

Cargo_States

(idCargo <FK>, idState <FK>, cargoAmount)

Com a primeira forma normal aplicada, pode-se verificar na tabela se existem colunas que não são funcionalmente dependentes da(s) chave(s) primária(s) desta tabela.

A tabela em questão já se encontra na segunda forma normal, uma vez que todos seus atributos são funcionalmente dependentes simultaneamente de idCargo e idState.

3FN

Analisando a tabela para ver se a mesma se apresenta na terceira forma normal, devemos “ignorar” a chave primária e ver se algum dos atributos é dependente de um outro atributo desta mesma tabela.

Cargo_States

(_____, _____, cargoAmount)

Como só sobrou um atributo que não é chave primária, a tabela está, automaticamente, normalizada.