## 

### **Functional Dependency in PostgreSQL**

horizontal line

# 

|  |  |  |
| --- | --- | --- |
| **TEAM MEMBER DETAILS** | | |
| **S. No.** | **Name** | **Roll Number** |
| **1** | **K Anand Naik.** | **213110031** |

## 

# 

# Project Title - Functional Dependency Support in PostgreSQL

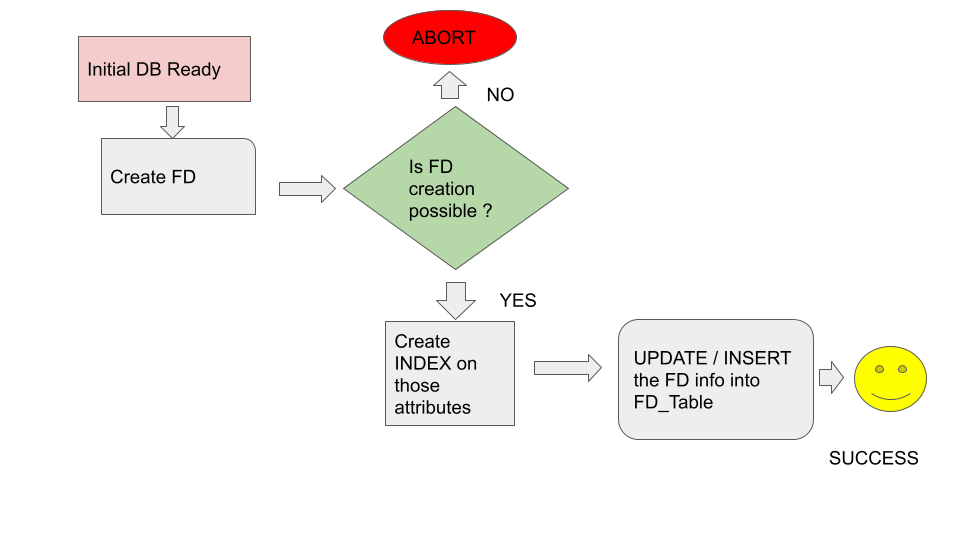
# Introduction

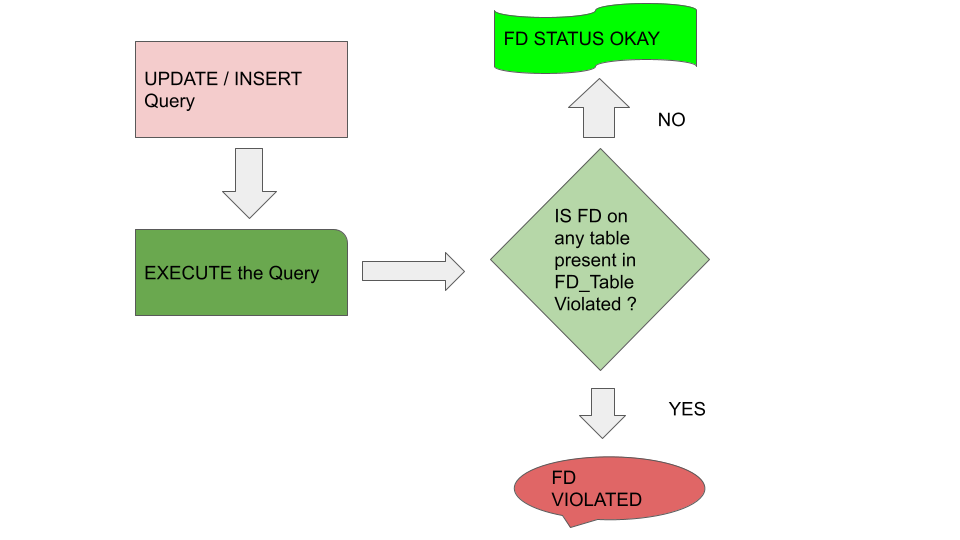
Functional dependencies are not supported by pretty much any database today. Given an FD A->B we need to create an index on (A,B) which stores a count of the number of tuples with that (A,B) value, and flags an error if there is more than one B value associated with a given A value.

# Abstract

* To obtain the functional dependency information from the User and create a special index on attributes in the functional dependency such that the index is not available to regular queries.
* To store the functional dependency metadata in a table and check whether the FD holds after insertion/deletion.

Algorithm





# Modifications in Postgres

Step 1: First DEBUG the command “CREATE INDEX A,B ON TABLE\_NAME”

Debugger goes into **index.c, execIndexing.c and indexing.c** files for CREATE INDEX clause. Once we find the function where the index is being created, we modify these files to enable us to create a special index for the relation of our choice.

Step 2: We create a table ( FD-TABLE) which holds information about the Functional Dependency ( Special FD-INDEX )

* Name of the table
* Columns on which there is FD

UPDATE COLUMN A FROM TABLE STUDENT / INSERTION

* Debug where actually the updates / insertions take place ( C FUNCTION )
* Check for Index lookup if it happens

Once you find that , you modify that C function to check if the columns we earlier added to the FD table still hold the functional dependency criteria or not ??

If they still hold, there is no problem. If not, we report an error.

# Python Implementation

We have implemented the above Algorithm in Python and using SQLite DB.

* To run the code, python functional\_dependency.py
* Code is well documented with print statements.
* Required Libraries
  + Tabulate
  + SQLite

## Initial DB state



We have three tables.

Instructors Table with following fields

* Instructor
* Department

Students Table with following fields

* Roll\_Number ( PRIMARY KEY )
* First\_Name
* Surname
* Department

Functional Dependency Table ( FD ) to store functional dependencies

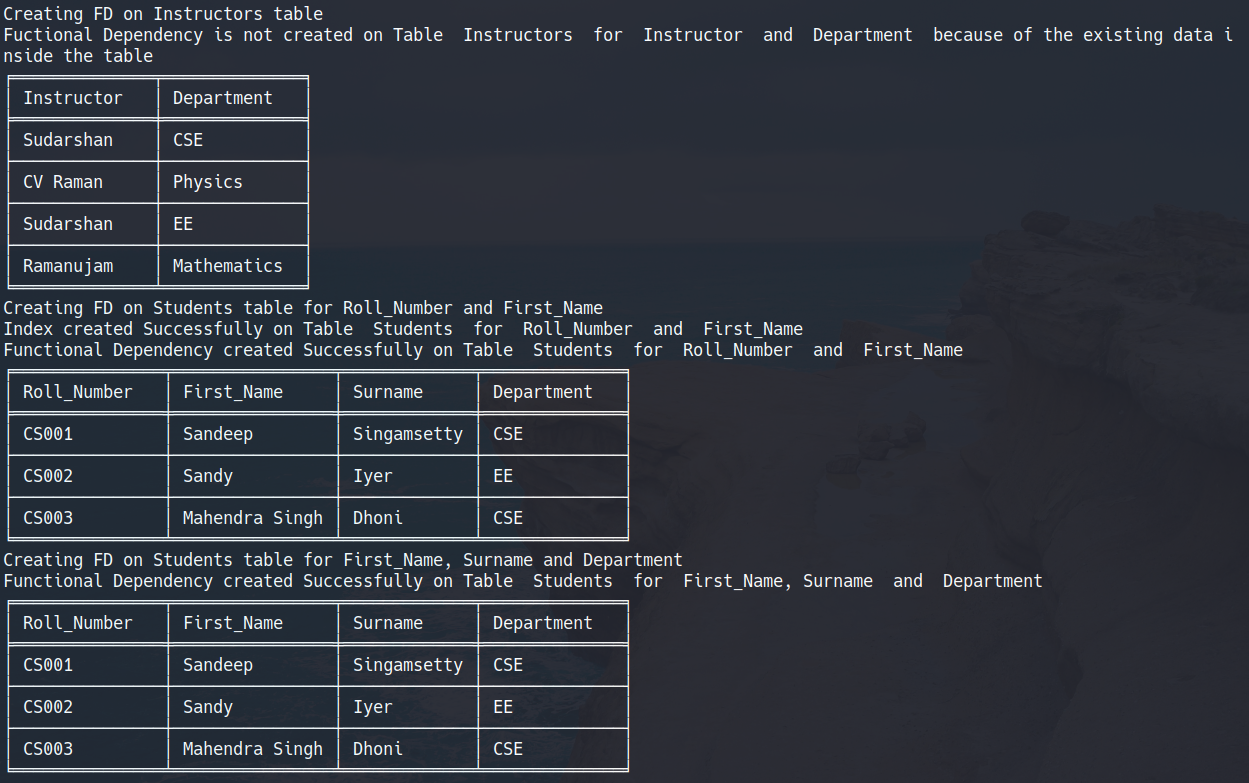
* Table\_Name
* Determinant
* Dependant

The FD table is initially empty.

## Adding Functional Dependencies

We have taken care of A → B and also AB→ C kind of Functional Dependencies.  
A →BC can be split as A→B and A→C.

Examples :



* Creating FD on Instructor table for Instructor → Department

create\_functional\_dependency('Instructors', 'Instructor' , 'Department')

As we can see, the functional dependency is not created, as it is not satisfied by the existing rows in the table.

The value of Instructor ‘Sudarshan’ is being mapped to two different Departments - ‘CSE’ and ‘EE’.

* Creating FD on Student table for Roll\_Number → First\_Name

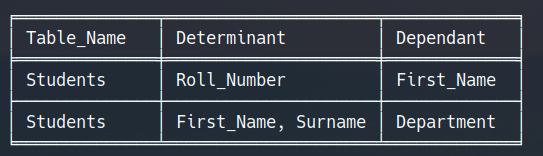
create\_functional\_dependency('Student', 'Roll\_Number' , 'First\_Name')

This FD is successfully created, and we insert the data into our FD table.

* Creating FD on Student table for ( First\_Name, Surname ) → Department

create\_functional\_dependency('Student', 'First\_Name, Surname' , 'Department')

This FD is also successfully created and inserted into the FD table.



These are the contents of the FD table - two rows for the two FDs.

## Test Cases ( Insertion / Update )

We explain five insert queries, and one update query to show how the FD checking behaves. We only look for the tables which have functional dependencies stated in the FD table, and if the insert or update has affected it.

## INSERT INTO Instructors VALUES('Marvel', 'Science Fiction')

No FDs will be violated by this insert, as we do not have a FD defined in the table for Instructor relation.

1. INSERT INTO Students VALUES('CS005', 'Virat', 'Kohli', 'CSE')

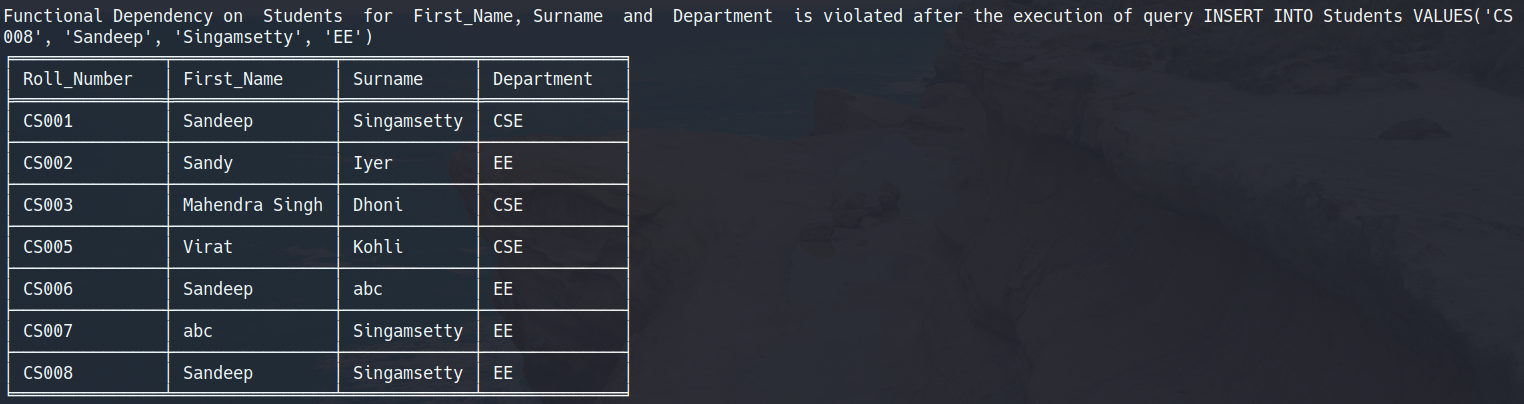
No FDs will be violated by this insert, as the new tuple satisfies the existing FDs.

1. INSERT INTO Students VALUES('CS006', 'Sandeep', 'abc', 'EE')
2. INSERT INTO Students VALUES('CS007', 'abc', 'Singamsetty', 'EE')

These do not cause violations, as our FD states that both First name and surname determine the department, and not either alone.

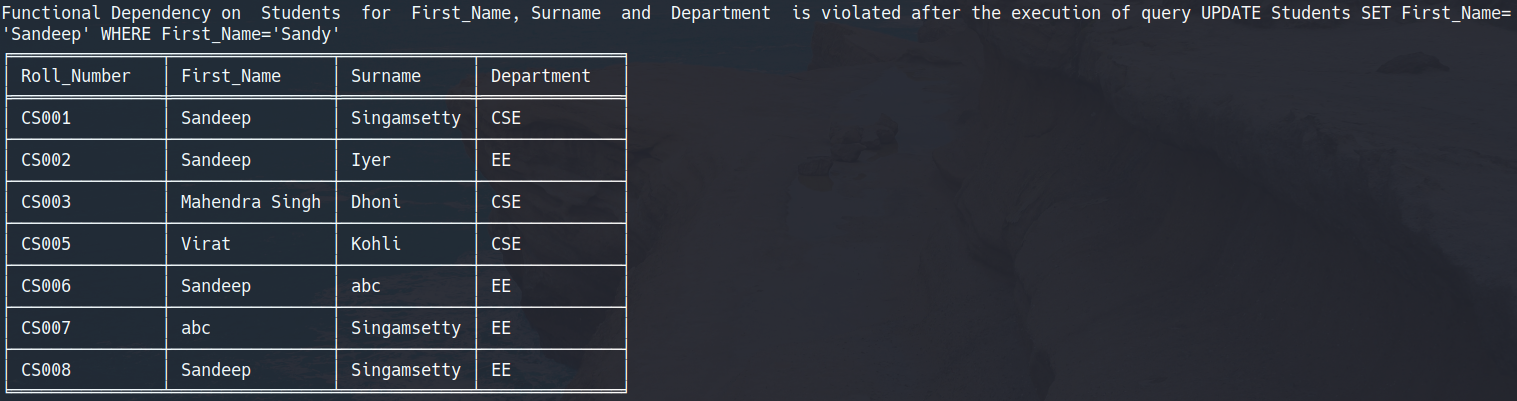
1. INSERT INTO Students VALUES('CS008', 'Sandeep', 'Singamsetty', 'EE')

This is a violation of both the functional dependencies. We already have a tuple with the same First name and surname, but with a different department.



1. UPDATE Students SET First\_Name='Sandeep' WHERE First\_Name='Sandy'

This is a violation, as there are two departments associated with the same first name.



# Difficulties Faced:

We used debugging in Eclipse on the PostgreSQL code to find the files for editing indexes. But we faced problems in editing those files. We had to deal with a lot of errors and abnormal terminations. Also, due to the vastness of the codebase, we could not figure out where to add the code for functionalities as per our idea given above.