

Functional Dependency Support for Postgresql

Somil Swapnil Chandra (22m0809)

Sandeep Kumar (22m0790)

Shashank Kate (22m0756)

November 29, 2022

1 Project Description

1.1 Abstract

The Goal of the project is to implement the Functional Dependency support for databases. Given an FD $A \rightarrow B$, we have to check that there should not be 2 different values of B for the same value of A, and Flag error if such a case arises. Such issue can arise on insertion or updation of a tuple. So we have to check the condition that the tuple being inserted/updated follows the FD rules on every insert/update.

1.2 Problem Description

We need to check for FD violations in our given relation. In a given relation of a given FD set say $(A \rightarrow B)$ we need to check in R, at the time of insertion if there is already a distinct value associated with A in the relation. If that's the case then we interrupt the insertion and prompt an error at the screen saying "Functional Dependency violation".

2 Proposed Methodology

2.1 Approach

We will take FDs as input from the user and store them in a table(for our example table "func_dep"). Then for every insertion/update we will check if the value of B that is being

inserted matches with the value of B that is already in the table, for the same value of A.

If the value of A is not there in the index, we need not to check FD, as it will definitely follow the FD rule(as first time that value of A is inserted) We edit **PostgreSQL.c** and **nodeModifytable.c** for this project. On every insertion we will run a function to check if the given value of A exist in table or not, if not then simply allow the insertion or if value of A exist in table then check the value of B, if it matches with value of B which is being inserted ,then only we will allow the insertion otherwise raise an error.

We have to create a new relation for Functional dependencies and an index for both columns in FD. Then whenever an INSERT query is invoked we will retrieve all FDs from the relation, and for each FD $A \rightarrow B$, we will look for the value of A in the index, now we need to check the value of B existing in the table to the value of B that is being inserted. INSERT is allowed only if both the values are the same, as they follow the FD rule otherwise we will raise an ERROR, showing that the insertion is failed as tuple did not follow the FD rule.

2.2 Working

The working of our method is as follows:

- First we extracted the Table name from query string.
- If there is a functional dependency given in func_dep table, we call the function **check_for_fd**
- The function extracts the two columns(Let them be A,B where $A \rightarrow B$) on which FD is defined (multiple times if multiple FD on same table are given).
- After getting required column name, it calls **exec()** which gets index of required column from **exec2()**.
- After getting the index it checks if there are any values of A which are equal to value being inserted.If not it simply returns without any error.
- If there exists any value of A, which are same as value being inserted. It gets the corresponding B value using SPI command,from the table.
- If obtained value of B is equal to value being inserted,it returns otherwise it throws an error, saying "Functional Dependency Violated".

Here we use some global variables, in main.c which are then included in a header file HEAD.h. This file is then included in postgresql.c and nodeModifyTable.c, to make parsing query_string easy.Which then also helps in obtaining table name.

3 Result Analysis

For testing we created a relation "students", with columns roll,name,dept,grad and then pushed FDs roll->name,dept->grad into func_dep table. Then we inserted some tuples in "students" relation, some values follows the FD rules, hence they are inserted in table.

```
PostgreSQL stand-alone backend 14.6
backend> create table func_dep(table_name varchar(255),a varchar(255),b varchar(255));
backend> create table students(roll varchar(255),name varchar(255),dept varchar(255),grad varchar(255));
backend> insert into func_dep values('students','roll','name');
backend> insert into func_dep values('students','dept','grad');
backend> select * from func_dep;
 1: table_name (typeid = 1043, len = -1, typmod = 259, byval = f)
 2: a (typeid = 1043, len = -1, typmod = 259, byval = f)
 3: b (typeid = 1043, len = -1, typmod = 259, byval = f)
----
 1: table_name = "students" (typeid = 1043, len = -1, typmod = 259, byval = f)
 2: a = "roll" (typeid = 1043, len = -1, typmod = 259, byval = f)
 3: b = "name" (typeid = 1043, len = -1, typmod = 259, byval = f)
----
 1: table_name = "students" (typeid = 1043, len = -1, typmod = 259, byval = f)
 2: a = "dept" (typeid = 1043, len = -1, typmod = 259, byval = f)
 3: b = "grad" (typeid = 1043, len = -1, typmod = 259, byval = f)
----
backend>
```

while some tuples violate the FD rules mentioned, as a result of which they are not inserted in relation and insertion is aborted and an ERROR is shown.

```

backend>
backend> insert into students values('22m0809','Somil','CSE','A');
backend> insert into students values('22m0756','Shashank','CSE','A');
backend>
backend> select * from students;
      1: roll      (typeid = 1043, len = -1, typmod = 259, byval = f)
      2: name      (typeid = 1043, len = -1, typmod = 259, byval = f)
      3: dept      (typeid = 1043, len = -1, typmod = 259, byval = f)
      4: grad      (typeid = 1043, len = -1, typmod = 259, byval = f)
----
      1: roll = "22m0809"      (typeid = 1043, len = -1, typmod = 259, byval = f)
      2: name = "Somil"        (typeid = 1043, len = -1, typmod = 259, byval = f)
      3: dept = "CSE"          (typeid = 1043, len = -1, typmod = 259, byval = f)
      4: grad = "A"           (typeid = 1043, len = -1, typmod = 259, byval = f)
----
      1: roll = "22m0756"      (typeid = 1043, len = -1, typmod = 259, byval = f)
      2: name = "Shashank"     (typeid = 1043, len = -1, typmod = 259, byval = f)
      3: dept = "CSE"          (typeid = 1043, len = -1, typmod = 259, byval = f)
      4: grad = "A"           (typeid = 1043, len = -1, typmod = 259, byval = f)
----
backend> insert into students values('22m0809','Somil Chandra','CSE','A');
backend> 2022-11-29 12:30:52.207 IST [192617] ERROR: Functional Dependency Violated
2022-11-29 12:30:52.207 IST [192617] STATEMENT: insert into students values('22m0809','Somil Chandra','CSE','A');

backend> insert into students values('22m0790','Sandeep','CSE','B');
backend> 2022-11-29 12:31:05.129 IST [192617] ERROR: Functional Dependency Violated
2022-11-29 12:31:05.129 IST [192617] STATEMENT: insert into students values('22m0790','Sandeep','CSE','B');

backend> select * from students;
|      1: roll      (typeid = 1043, len = -1, typmod = 259, byval = f)
      2: name      (typeid = 1043, len = -1, typmod = 259, byval = f)
      3: dept      (typeid = 1043, len = -1, typmod = 259, byval = f)
      4: grad      (typeid = 1043, len = -1, typmod = 259, byval = f)
----
      1: roll = "22m0809"      (typeid = 1043, len = -1, typmod = 259, byval = f)
      2: name = "Somil"        (typeid = 1043, len = -1, typmod = 259, byval = f)
      3: dept = "CSE"          (typeid = 1043, len = -1, typmod = 259, byval = f)
      4: grad = "A"           (typeid = 1043, len = -1, typmod = 259, byval = f)
----
      1: roll = "22m0756"      (typeid = 1043, len = -1, typmod = 259, byval = f)
      2: name = "Shashank"     (typeid = 1043, len = -1, typmod = 259, byval = f)
      3: dept = "CSE"          (typeid = 1043, len = -1, typmod = 259, byval = f)
      4: grad = "A"           (typeid = 1043, len = -1, typmod = 259, byval = f)
----
backend>

```

4 Challenges and Future Scope

- Parsing the query string to get the relation name and the attribute values using C language was a challenging task
- Executing queries in Postgresql (using SPI) and handling the global variables was a bit tricky.
- Future scope for this project could be to implement multi-value functional dependency constraint in the relation as well.

5 Conclusion

Here we have proposed a check on the Functional Dependency constraint for Postgresql. We have considered simple functional dependency $A \rightarrow B$ and have reported FD violations if there are any. Given the potential of the project we can do a lot more work in it to provide functional dependency checks on multi-valued dependencies in a relation.

6 References

- <https://doxygen.postgresql.org/files.html>
- <https://www.postgresql.org/docs/8.2/spi-spi-connect.html>
- <https://www.postgresql.org/docs/9.1/spi-examples.html>
- <https://www.codingame.com/playgrounds/14213/how-to-play-with-strings-in-c/string-split>