**You will need 3 tables and 3 views as follows:**

**Tables :**

* 1 PK + 0 FK
* 1 PK + 1 FK
* 1 PK (2 or more columns)

**Views (on tables below):**

* On 1 table
* 2 tables with a JOIN
* 2 tables with a JOIN and a GROUP BY

**Download the script that creates the test tables from this link** [**http://sabina-cs.com/wp-content/uploads/2019/11/Script\_lab4.txt**](http://sabina-cs.com/wp-content/uploads/2019/11/Script_lab4.txt)

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Column Name** | **Description** |
| **Tables** | **TableID** | Key for the table |
| **Name** | Table name |
| **Views** | **ViewID** | Key for the view |
| **Name** | View name |
| **Tests** | **TestID** | Key for the test |
| **Name** | Name of the test. It can be the name of your procedures that have to be executed. |
| **TestViews** | **TestID** | FK to TestID from Tests. |
| **ViewID** | FK to ViewID from Views. |
| **TestTables** | **TestID** | FK to TestID from Tests. |
| **TableID** | FK to TableID from Tables. |
| **NoOfRows** | Number of rows that will be affected by operation. |
| **Position** | The order in witch the tables will be considered on insert and delete. |
| **TestRuns** | **TestRunID** | Key for the test run. |
| **Description** | A short description of the test. |
| **StartAt** | The date when the test start. |
| **EndAt** | The date when the test is done. |
| **TestRunTables** | **TestRunID** | FK to TestRunID from TestRuns. |
| **TableID** | FK to TableID from Tables. |
| **StartAt** | The date when the insert start. |
| **EndAt** | The date when the insert is done. |
| **TestRunViews** | **TestRunID** | FK to TestRunID from TestRuns. |
| **ViewID** | FK to ViewID from Views. |
| **StartAt** | The date when the select start. |
| **EndAt** | The date when the select is done. |

**Here are the values that I inserted in tables:**

|  |  |
| --- | --- |
| **Tables** |  |
| **Views** |  |
| **Tests** |  |
| **TestViews** |  |
| **TestTables** | Order for insert: PK, PK+FK, multicolumn PK  Order for delete: multicolumn PK, PK+FK, PK |

Sugestions for insert procedures:

* Table with PK: create a column with negative value (for numeric columns) or with value Test+@i (for varchar columns)
* Table with PK and FK: take a valid value FK into a variable (@fk =select top 1 id from ... ) and insert test data (as below) with that FK
* Table with multicolumn PK: if your columns are FKs take all of posible combination of keys using cross join (you can use a cursor on it too)

How a complete test should look (what stepts it have to make):

Declare @t0 datetime = GETDATE()

Delete TeaShops

Delete Teas

Delete TeaTypes

Declare @t1 datetime = GETDATE()

Insert into TeaType

Declare @t2 datetime = GETDATE()

Insert into TestRunTables (testid, tableid, @t1, @t2)

@t1 datetime = GETDATE()

Insert into Teas

@t2 datetime = GETDATE()

Insert into TestRunTables (testid, tableid, @t1, @t2)

@t1 datetime = GETDATE()

Insert into TeaShops

@t2 datetime = GETDATE()

Insert into TestRunTables (testid, tableid, @t1, @t2)

@t1 datetime = GETDATE()

Select from View1

@t2 datetime = GETDATE()

Insert into TestRunViews(testid, viewid, @t1, @t2)

@t1 datetime = GETDATE()

Select from View2

@t2 datetime = GETDATE()

Insert into TestRunViews(testid, viewid, @t1, @t2)

@t1 datetime = GETDATE()

Select from View3

@t2 datetime = GETDATE()

Insert into TestRunViews(testid, viewid, @t1, @t2)

Insert into TestRuns(description, @t0, @t2)

**The way you implement the stored procedures and / or functions is up to you. If you make all the calls manually you will get a 9. If you make it to take (automatically) everything from tables you will get a 10. If you get only the NoOfRows and/or Position you will get a 9.5.**