**COMP 440 FINAL PROJECT REPORT HARISH PINDI**

**Stored Procedures**

**1)Stored Procedure-1(INSERTION)**

**For feature Insertion**

USE [s16guest04]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[feature\_data] Script Date: 5/2/2016 10:40:50 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create PROCEDURE [dbo].[feature\_data]

@feature\_id int ,

@feature\_name varchar(50),

@feature\_description varchar(80)

AS

BEGIN

Begin Try

SET NOCOUNT ON

Begin Transaction

INSERT INTO [dbo].[feature]

([feature\_id],

[feature\_name],

[feature\_description]

)

VALUES

(

@feature\_id ,

@feature\_name,

@feature\_description

)

Commit Transaction

End Try

Begin Catch

Rollback Transaction

Select

ERROR\_NUMBER() as ErrorNumber,

ERROR\_MESSAGE() as ErrorMessage,

ERROR\_PROCEDURE() as ErrorProcedure,

ERROR\_STATE() as ErrorState,

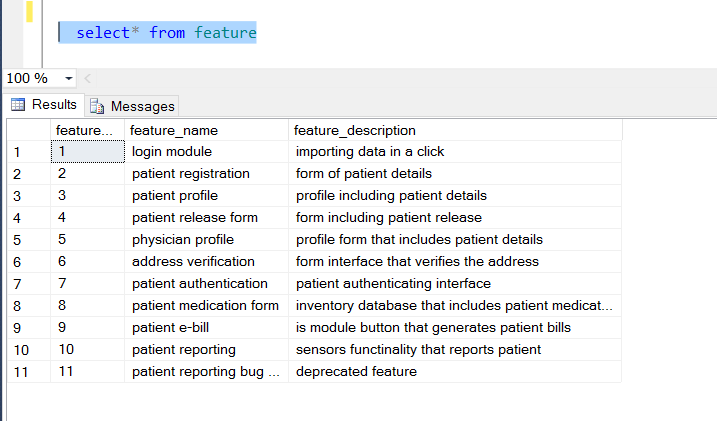
ERROR\_SEVERITY() as ErrorSeverity,

ERROR\_LINE() as ErrorLine

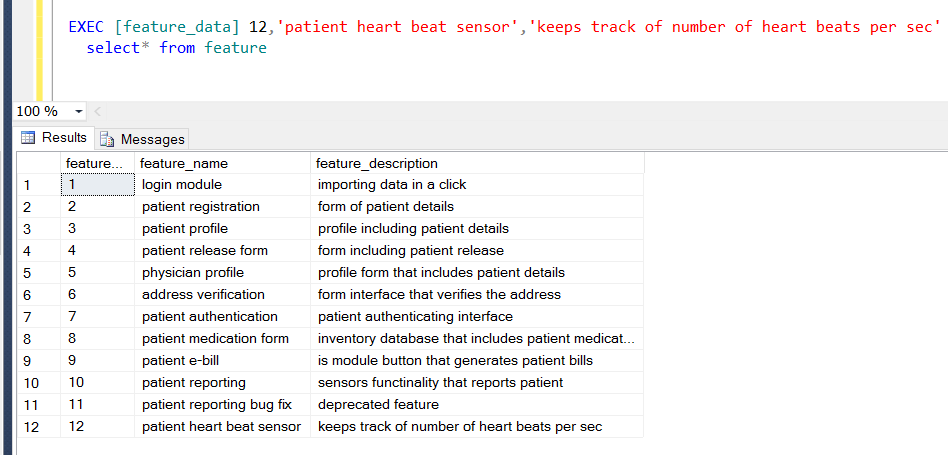
END Catch

END

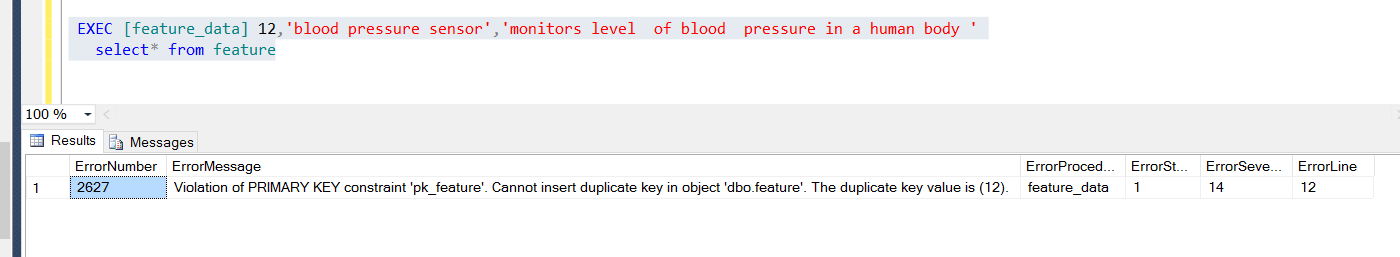
**Before insertion of feature data**



**After insertion of a record in to a feature table by means of stored procedure**



**Displaying error message while inserting duplicate data**



**2. Update product version stored procedure**

Here in these when ever product gets updated its version gets updated corresponding to its version id

**Code:**

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[sp\_product\_version] Script Date: 5/2/2016 3:13:15 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE procedure [dbo].[sp\_product\_version]

--declaring the product attributes and version attributes

@product\_id int,

@product\_name varchar(50),

@version\_id int,

@version\_name varchar(90)

AS

BEGIN

Begin Try

SET NOCOUNT ON

Begin Transaction

----Updating the product table A by means of inner join with version table B

update A

set A.product\_name= @product\_name

from product A inner join version B

on B.product\_id = A.product\_id

and A.product\_id = @product\_id

----Updating the version table B by means of inner join with product table A

update B

set B.version\_name = @version\_name,B.version\_id=@version\_id

from version B inner join product A

on B.product\_id = A.product\_id

and A.product\_id = @product\_id

Commit Transaction

End Try

Begin Catch

Rollback Transaction

Select

ERROR\_NUMBER() as ErrorNumber,

ERROR\_MESSAGE() as ErrorMessage ,

ERROR\_PROCEDURE() as ErrorProcedure,

ERROR\_STATE() as ErrorState,

ERROR\_SEVERITY() as ErrorSeverity,

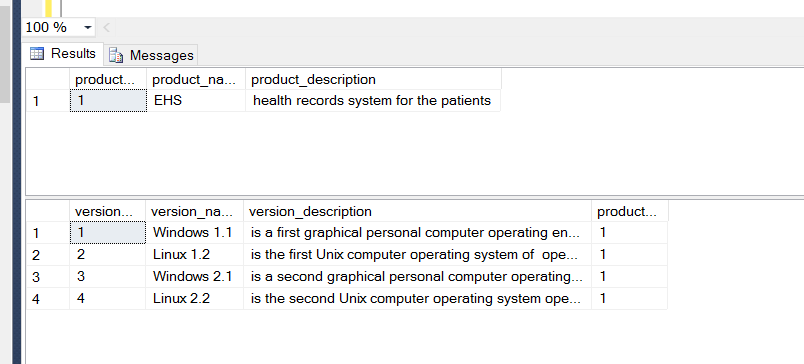
ERROR\_LINE() as ErrorLine

END Catch

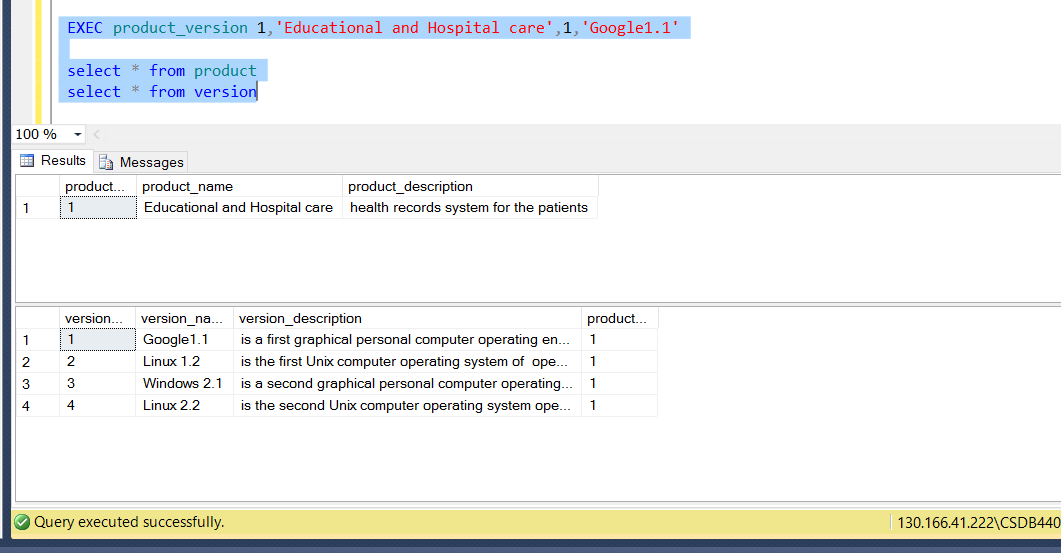
END

GO

**Before the stored procedure**



**After the update stored procedure**



**3) Download Report Stored procedure**

USE [s16guest04]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[download\_report] Script Date: 5/2/2016 3:36:36 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create procedure [dbo].[download\_report]

AS

BEGIN

Begin Try

SET NOCOUNT ON

Begin Transaction

select [product\_name],

[version\_name],

--here below the month of download date is so called as Month

MONTH(download\_date) AS [Month],

---For counting the number of months in a date we use count operation

COUNT(MONTH(download\_date)) AS [Count]

FROM version

INNER JOIN customer\_download

ON version.version\_id=customer\_download.downloaded\_version

INNER JOIN product

ON product.product\_id=version.product\_id

---the records for a particular manner to display we use group by

GROUP BY [product\_name],[version\_name],MONTH(download\_date)

Commit Transaction

End Try

Begin Catch

Rollback Transaction

Select

ERROR\_NUMBER() as ErrorNumber,

ERROR\_MESSAGE() as ErrorMessage ,

ERROR\_PROCEDURE() as ErrorProcedure,

ERROR\_STATE() as ErrorState,

ERROR\_SEVERITY() as ErrorSeverity,

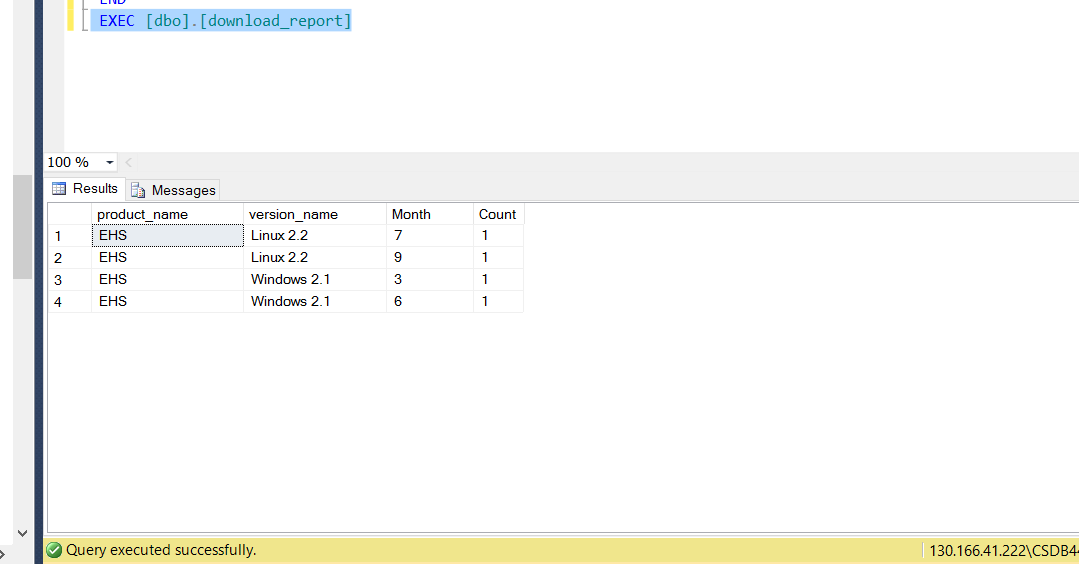
ERROR\_LINE() as ErrorLine

END Catch

END

GO

**After executing the stored procedure**

****

4)Feature report stored procedure

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[feature\_report] Script Date: 5/2/2016 11:34:27 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE procedure [dbo].[feature\_report]

@version\_name varchar(90),

@release\_name varchar(20)

AS

Begin

----retrieving the number of new features for the particular version by using inner join between version and feature by means of cross reference table Version\_feature

declare @result int

SELECT @result=COUNT (feature\_name )

FROM version\_feature

INNER JOIN feature

ON version\_feature.feature\_id=feature.feature\_id

INNER JOIN version

ON version\_feature.version\_id=version.version\_id

WHERE version\_name = @version\_name

if(@result<>0)

print 'For '++ @version\_name ++ ',There are '++ CAST(@result AS VARCHAR) ++ ' new features';

else

print 'There are bugs'

declare @Fresult int;

----retrieving the number of new features for particular release by using inner join between development\_release and feature by means of cross reference table release\_feature

SELECT @Fresult=COUNT (feature\_name )

FROM release\_feature

INNER JOIN feature

ON release\_feature.feature\_id=feature.feature\_id

INNER JOIN [development\_release]

ON release\_feature.release\_id=development\_release.release\_id

WHERE release\_name = @release\_name

if(@Fresult<>0)

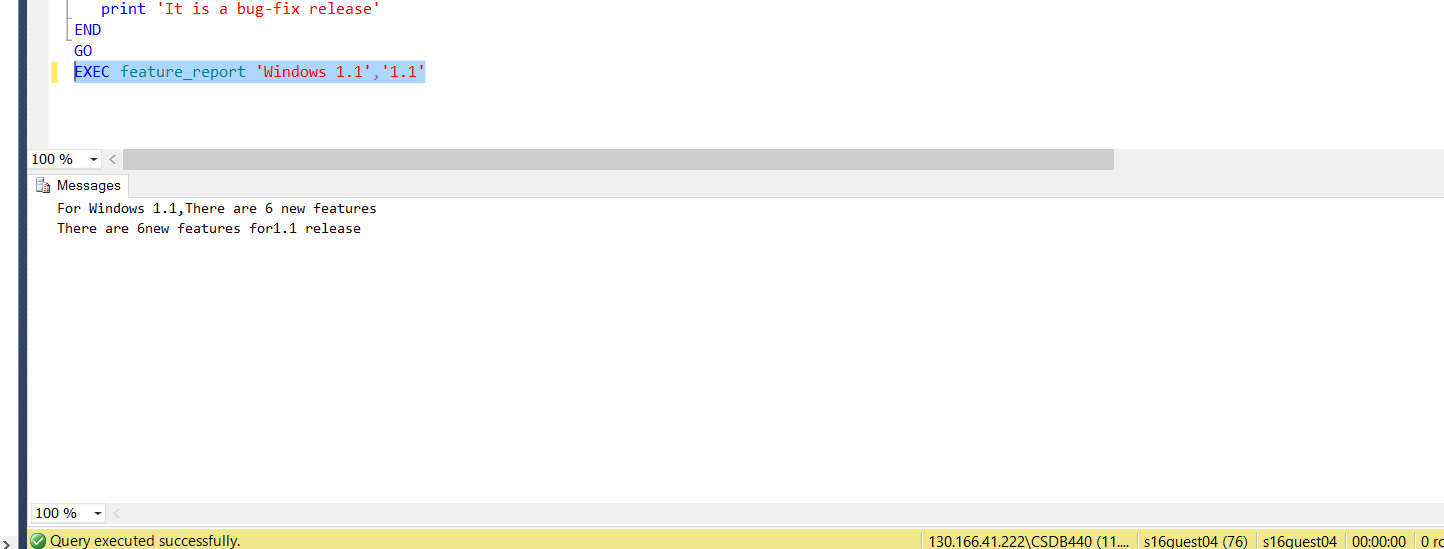
print 'There are '+ CAST(@result AS VARCHAR) + 'new features for' ++ @release\_name ++' release' ;

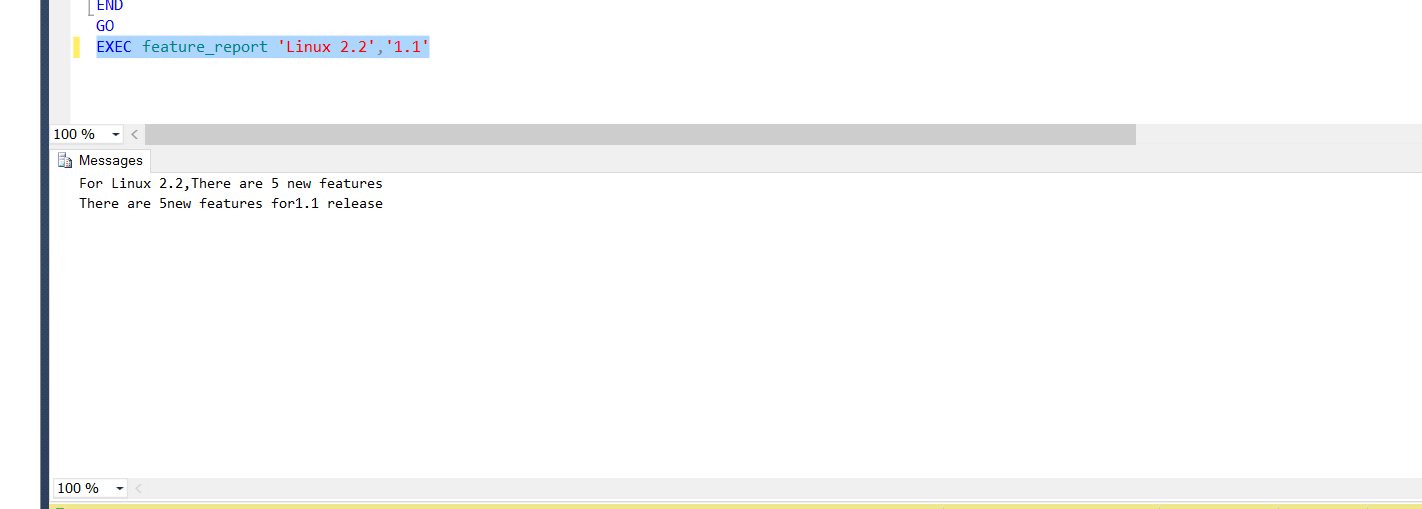
else

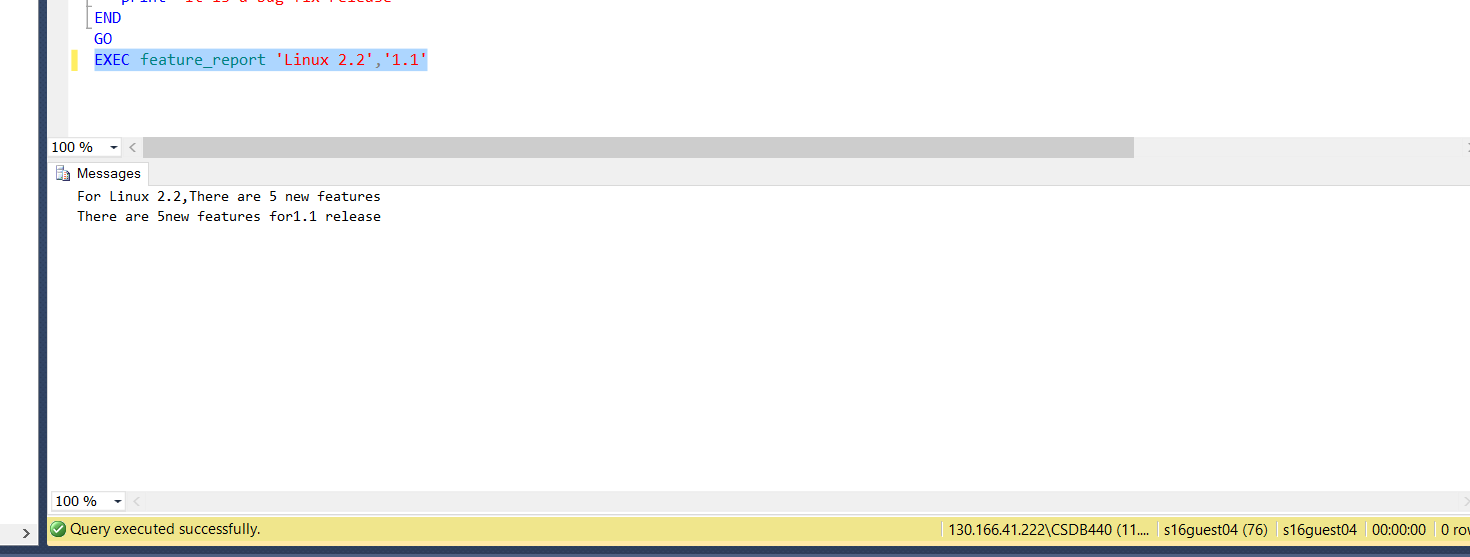
print 'It is a bug-fix release'

END

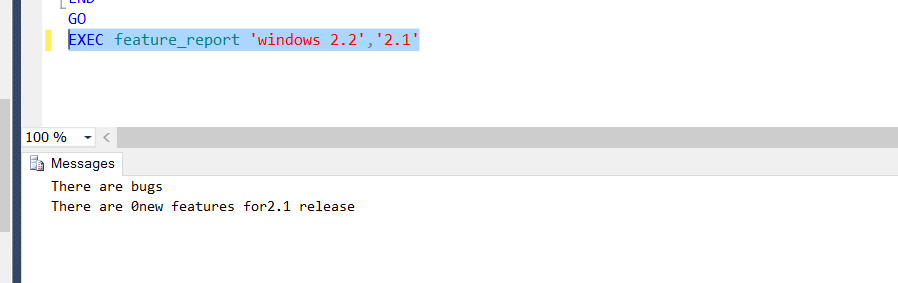
GO



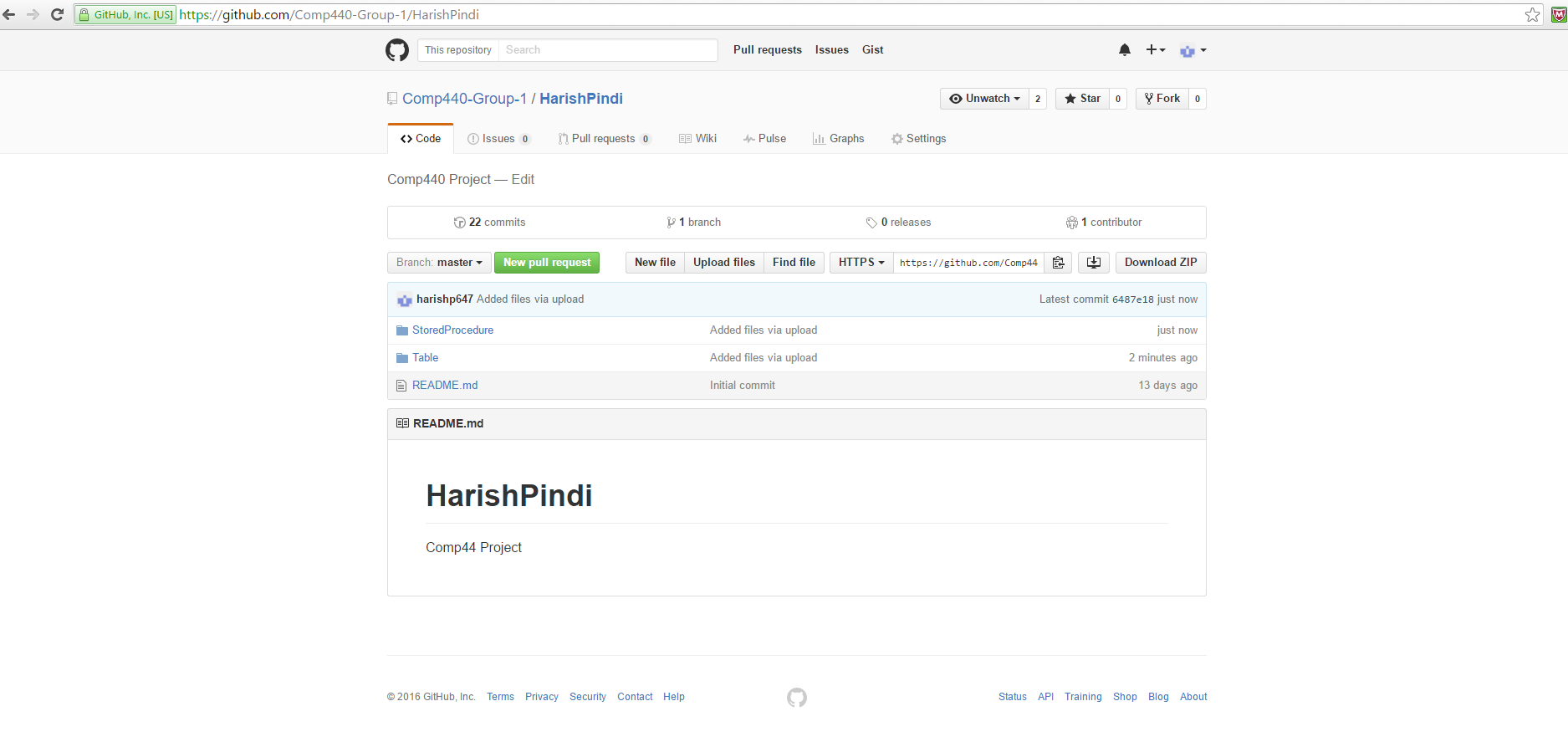




**Bugrelease**



5)Source Control



**6)Diagram**

