DBMS ASSIGNMENT 2
5TH SEM SECTION I

PES1UG19CS579
VISHWAS R

PES1UG19CS548
UTHPAL P

PES1UG19CS534
T R SUDHARSHAN

PROJECT TITILE

G ITHUB
IT

List of reasons/constraints to justify the choice of DBMS. [RDBMS]

- Our data is -
  - structured
  - Tables had fixed rows and columns
- Data retrieval is simple in RDBMS compared to nosql
- Our data in database changes frequently so it would have been difficult if we had used nosql.
- RDBMS is best suited for complex queries.
- RDBMS focuses on vertical scaling .

- We used postgresql to implement our RDBMS
- As postgresql is very familiar to us and easy to connect to python using psycopg2.

Relational Table

Project

P_RID (pk) (fk from Repository (RID))	Description varchar(200)	Progress_Bar [int] check(Progress_Bar >= 0) && check(Progress_Bar<=100)	P_Company_ID (fk from Company)

Repository

RID(pk)[int]	Remote_RID (fk from Project) (not null)	Owns (fk from User)	hours [int]
		If null then remote	
		Else local	

Branch

BID(pk)[int]	B_RID (pk) (fk from Repository (RID))	Time stamp [time] (not null)	parent_BID (fk from Branch)	Parent_B_RID (fk from Branch)

Employs

E_User_ID (fk from User) [pk]	E_Company_ID (fk from Company)	Salary Numeric(10,2) NOT NULL

Company

Company_ID (pk)	Company Description varchar(200)	Location varchar(200)

Developer

User_ID (pk)	Password NOT NULL char_length(Password) >=8

Holds

BID [pk] [fk from branch]	B_RID [pk] [fk from branch]	VID [pk] [fk from version]	V_remo_repo_ID [pk] [fk from version]	Rank [int]

Version

VID (pk)[int]	V_Rem_repo_ID [pk] (fk from Branch)	V_User_id (fk from user)

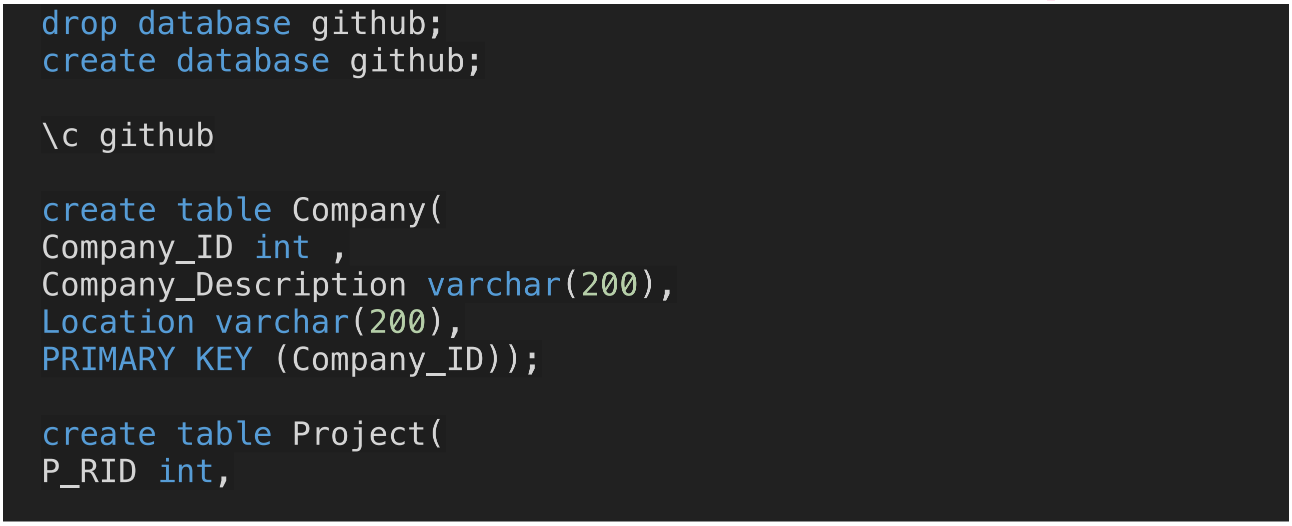
Contains

Contains_VID (pk) (fk from version)	Contains_ V_Rem_repo_ID [pk] (fk from version)	Contains_FID (fk from file) [pk]

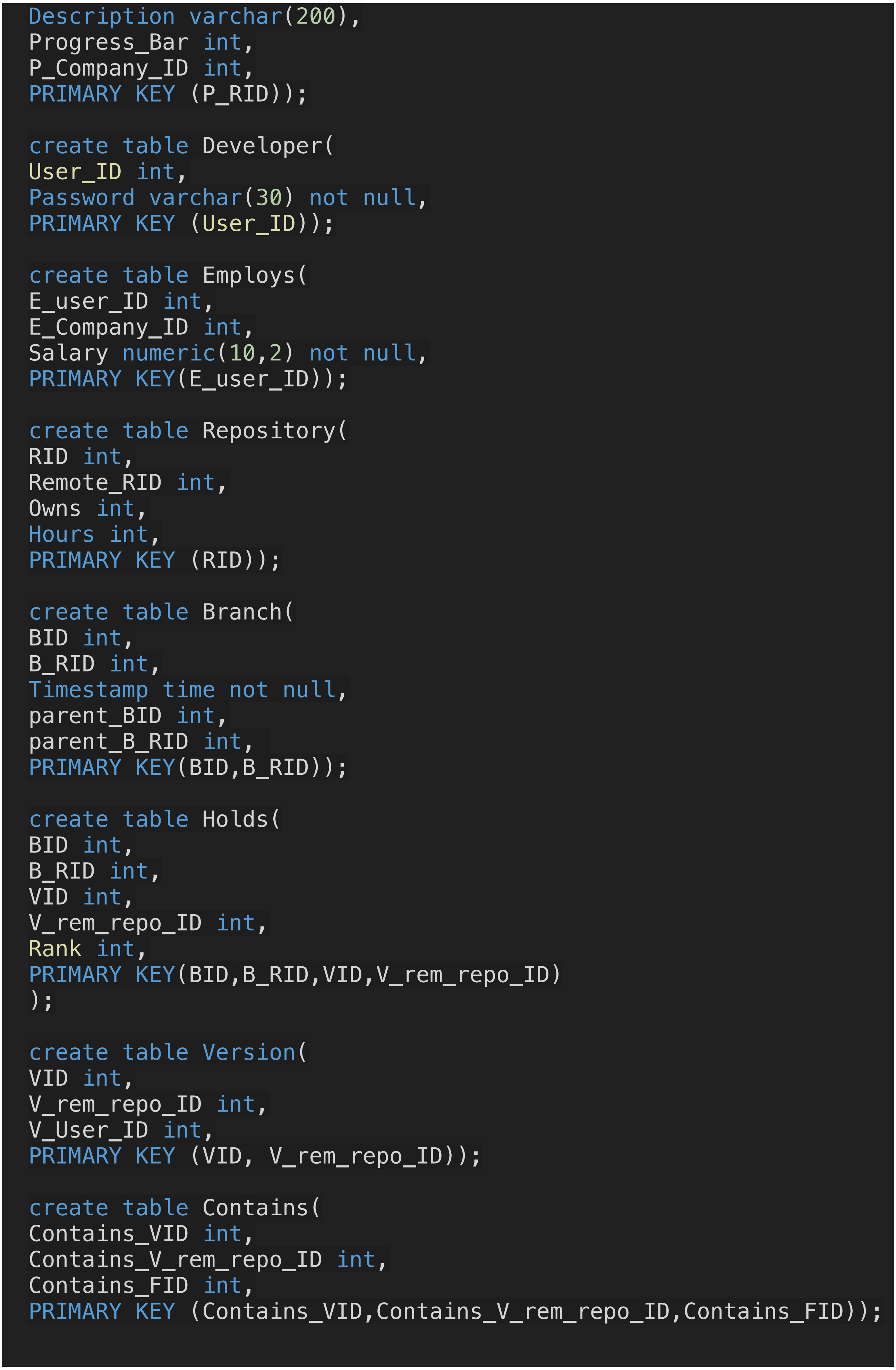
File

FID (pk)(not null) [int]	file_name(not null) varchar(30)	Type varchar(10) DEFAULT 'txt'	file_contains (text type)	file_size (not null) [int]

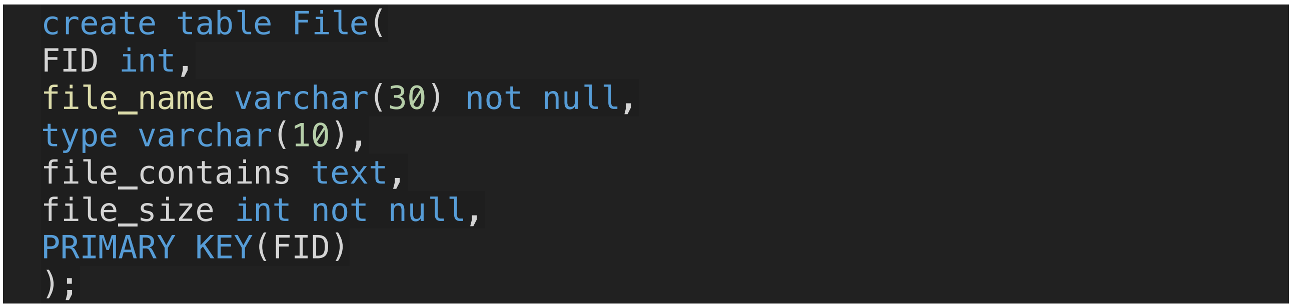
createstatements.sql



```
drop database github;
create database github;

\c github

create table Company(
Company_ID int ,
Company_Description varchar(200),
Location varchar(200),
PRIMARY KEY (Company_ID));

create table Project(
P_RID int,
```

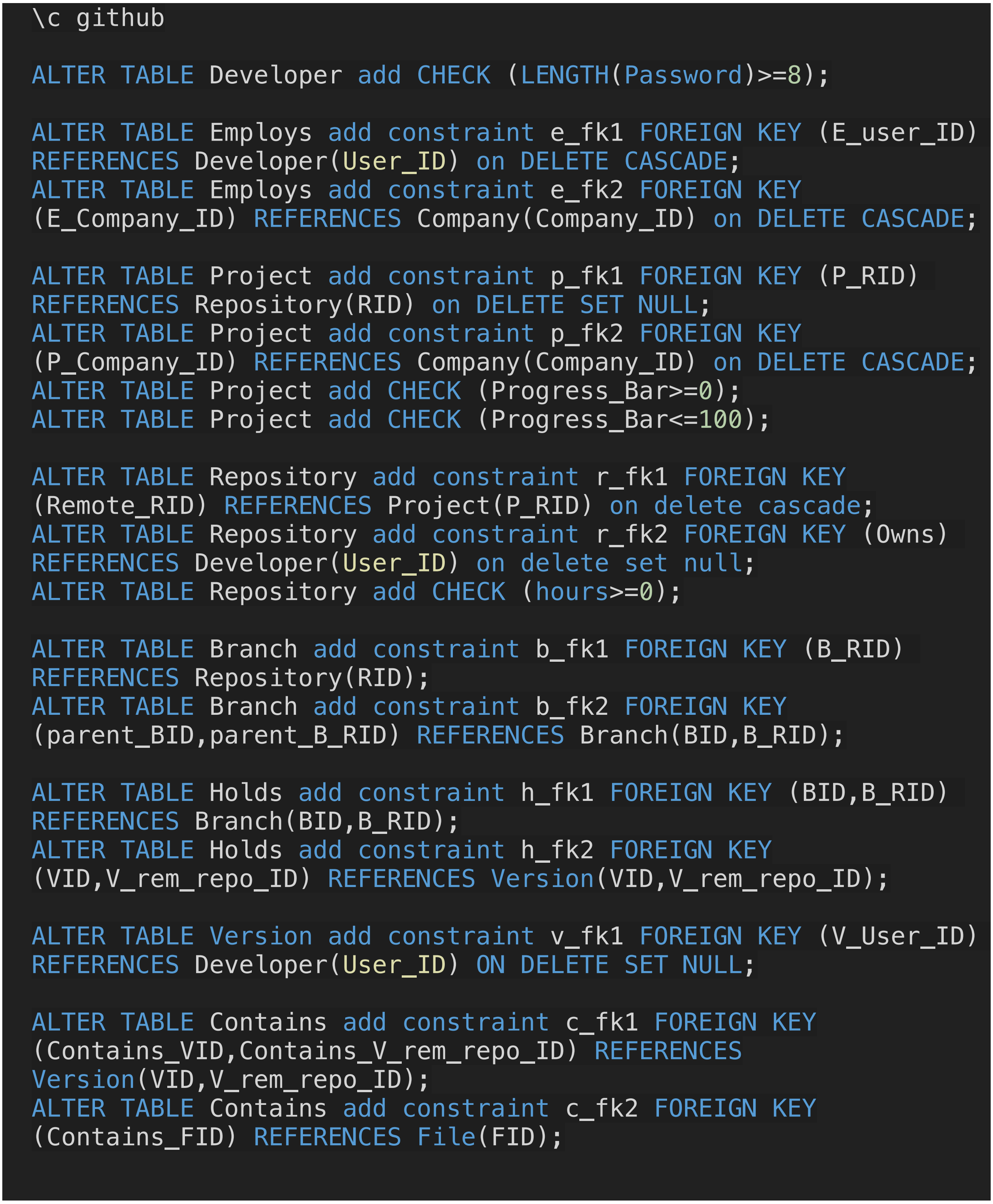


```
Description varchar(200),
Progress_Bar int,
P_Company_ID int,
PRIMARY KEY (P_RID));

create table Developer(
User_ID int,
Password varchar(30) not null,
PRIMARY KEY (User_ID));

create table Employs(
E_user_ID int,
E_Company_ID int,
Salary numeric(10,2) not null,
PRIMARY KEY(E_user_ID));

create table Repository(
RID int,
Remote_RID int,
Owns int,
Hours int,
PRIMARY KEY (RID));

create table Branch(
BID int,
B_RID int,
Timestamp time not null,
parent_BID int,
parent_B_RID int,
PRIMARY KEY(BID,B_RID));

create table Holds(
BID int,
B_RID int,
VID int,
V_rem_repo_ID int,
Rank int,
PRIMARY KEY(BID,B_RID,VID,V_rem_repo_ID)
);

create table Version(
VID int,
V_rem_repo_ID int,
V_User_ID int,
PRIMARY KEY (VID, V_rem_repo_ID));

create table Contains(
Contains_VID int,
Contains_V_rem_repo_ID int,
Contains_FID int,
PRIMARY KEY (Contains_VID,Contains_V_rem_repo_ID,Contains_FID));
```



```
create table File(
FID int,
file_name varchar(30) not null,
type varchar(10),
file_contains text,
file_size int not null,
PRIMARY KEY(FID)
);
```

constriants.sql



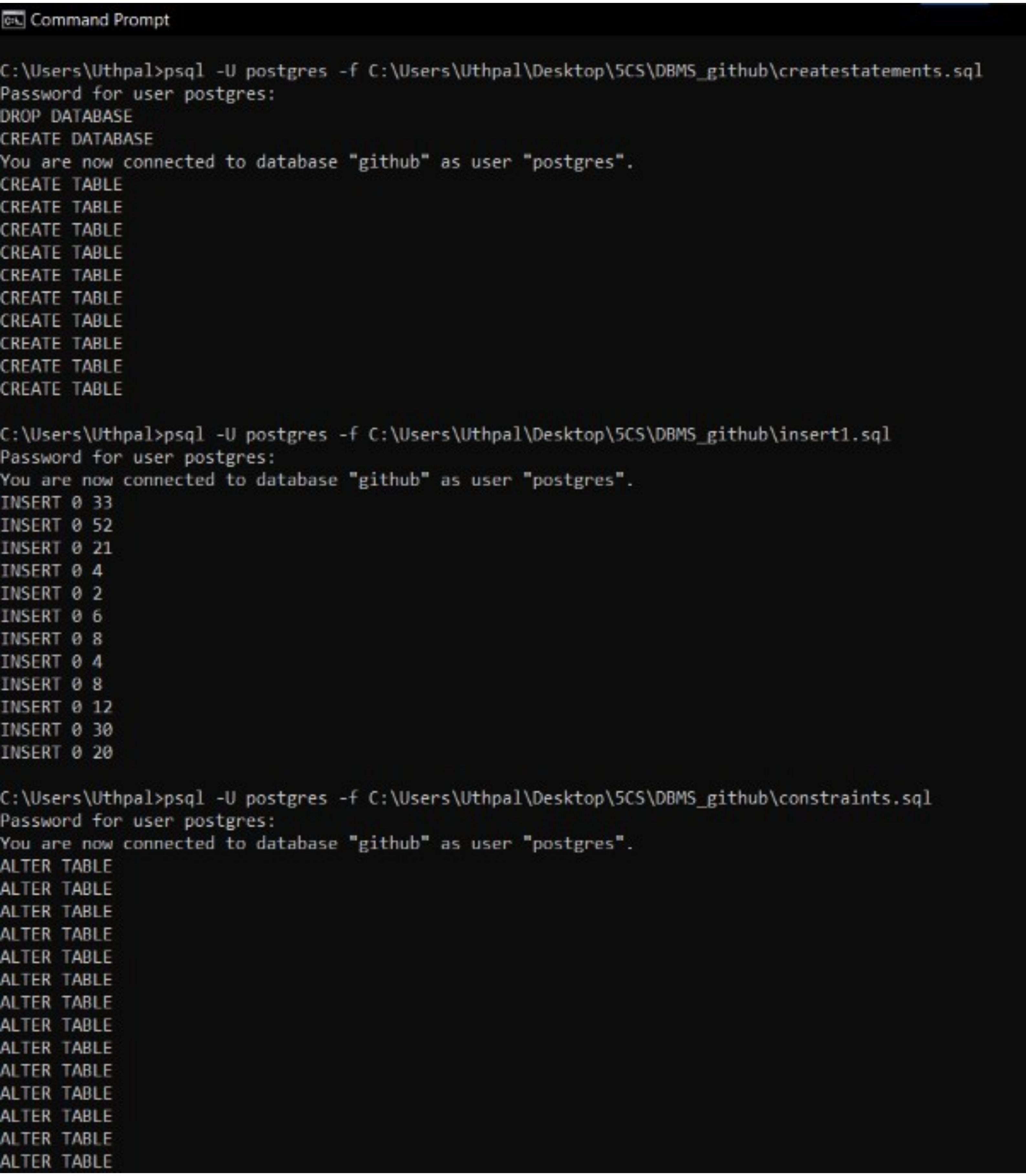
```
\c github

ALTER TABLE Developer add CHECK (LENGTH(Password)>=8);

ALTER TABLE Employs add constraint e_fk1 FOREIGN KEY (E_user_ID)
REFERENCES Developer(User_ID) on DELETE CASCADE;
ALTER TABLE Employs add constraint e_fk2 FOREIGN KEY
(E_Company_ID) REFERENCES Company(Company_ID) on DELETE CASCADE;

ALTER TABLE Project add constraint p_fk1 FOREIGN KEY (P_RID)
REFERENCES Repository(RID) on DELETE SET NULL;
ALTER TABLE Project add constraint p_fk2 FOREIGN KEY
(P_Company_ID) REFERENCES Company(Company_ID) on DELETE CASCADE;
ALTER TABLE Project add CHECK (Progress_Bar>=0);
ALTER TABLE Project add CHECK (Progress_Bar<=100);

ALTER TABLE Repository add constraint r_fk1 FOREIGN KEY
(Remote_RID) REFERENCES Project(P_RID) on delete cascade;
ALTER TABLE Repository add constraint r_fk2 FOREIGN KEY (Owns)
REFERENCES Developer(User_ID) on delete set null;
ALTER TABLE Repository add CHECK (hours>=0);

ALTER TABLE Branch add constraint b_fk1 FOREIGN KEY (B_RID)
REFERENCES Repository(RID);
ALTER TABLE Branch add constraint b_fk2 FOREIGN KEY
(parent_BID,parent_B_RID) REFERENCES Branch(BID,B_RID);

ALTER TABLE Holds add constraint h_fk1 FOREIGN KEY (BID,B_RID)
REFERENCES Branch(BID,B_RID);
ALTER TABLE Holds add constraint h_fk2 FOREIGN KEY
(VID,V_rem_repo_ID) REFERENCES Version(VID,V_rem_repo_ID);

ALTER TABLE Version add constraint v_fk1 FOREIGN KEY (V_User_ID)
REFERENCES Developer(User_ID) ON DELETE SET NULL;

ALTER TABLE Contains add constraint c_fk1 FOREIGN KEY
(Contains_VID,Contains_V_rem_repo_ID) REFERENCES
Version(VID,V_rem_repo_ID);
ALTER TABLE Contains add constraint c_fk2 FOREIGN KEY
(Contains_FID) REFERENCES File(FID);
```

```
ALTER TABLE File ALTER COLUMN type SET DEFAULT '.txt';
ALTER TABLE Contains add constraint c_fk2 FOREIGN KEY
(Contains_FID) REFERENCES File(FID);
ALTER TABLE File ALTER COLUMN type SET DEFAULT '.txt';
```

insert.sql

```
\c github;

INSERT into Holds values
(1,1,1,1,1),
(2,1,2,1,1),
(1,1,3,1,2),
(1,4,1,1,1),
(2,4,2,1,1),
(1,6,1,1,1),
(2,6,2,1,1),
(3,6,5,1,1),
(1,2,1,2,1),
(2,2,3,2,1),
(1,7,1,2,1),
(1,7,2,2,2),
(2,7,3,2,1),
(4,7,4,2,1),
(1,8,1,2,1),
(2,8,3,2,1),
(3,8,5,2,1),
(1,3,1,3,1),
(1,9,1,3,1),
(2,9,2,3,1),
(3,9,3,3,1),
(1,10,1,3,1),
(2,10,4,3,1),
(4,10,5,3,1),
(1,11,1,11,1),
(2,11,2,11,1),
(4,11,4,11,1),
(1,12,1,11,1),
(2,12,2,11,1),
(3,12,3,11,1),
(4,12,4,11,1),
(5,12,5,11,1),
(1,4,3,1,1);

INSERT into Contains values
(1,1,1),
(1,1,2),
(1,1,4),
(2,1,1),
(2,1,2),
(2,1,5),
(3,1,1),
(3,1,3),
(3,1,4),
(4,1,1),
(4,1,3),
(4,1,5),
(5,1,1),
(5,1,2),
(1,2,6),
(1,2,7),
(1,2,9),
(2,2,6),
(2,2,8),
(2,2,9),
(3,2,6),
(3,2,7),
(3,2,10),
(4,2,6),
(4,2,8),
(4,2,10),
(5,2,6),
(5,2,9),
(1,3,11),
(1,3,12),
(1,3,14),
(2,3,11),
(2,3,12),
(2,3,15),
(3,3,11),
(3,3,13),
(3,3,14),
(4,3,11),
(4,3,13),
(4,3,15),
(5,3,1),
(5,3,15),
(1,11,16),
(1,11,17),
(2,11,16),
(2,11,18),
(3,11,16),
(3,11,19),
(4,11,16),
(4,11,20),
(5,11,16),
(5,11,21);

INSERT into File values
(1,'read','md','file_desc1',100),
(2,'sub','c','file_desc2',500),
(3,'sub','c','file_desc3',540),
(4,'subh','h','file_desc4',300),
(5,'subh','h','file_desc5',310),
(6,'read','md','file_desc6',120),
(7,'solve','py','file_desc7',600),
(8,'solve','py','file_desc8',610),
(9,'test','py','file_desc9',450),
(10,'test','py','file_desc10',410),
(11,'read','md','file_desc11',150),
(12,'ddl','sql','file_desc12',300),
(13,'ddl','sql','file_desc13',350),
(14,'dml','sql','file_desc14',700),
(15,'dml','sql','file_desc15',750),
(16,'read','md','file_desc16',70),
(17,'report','doc','file_desc17',500),
(18,'report','doc','file_desc18',550),
(19,'report','doc','file_desc19',570),
(20,'report','doc','file_desc20',600),
(21,'report','doc','file_desc21',620);

INSERT into Project values
(1,'proj_desc_1',50,1),
(2,'proj_desc_2',40,2),
(3,'proj_desc_3',60,2),
(11,'proj_desc_4',70,null);

insert into company values
(1,'Comp_desc1','loc1'),
(2,'Comp_desc2','loc2');

insert into employs values
(1,1,1500),
(2,1,1400),
(3,1,1200),
(5,2,1700),
(6,2,1600),
(7,2,1900);

insert into Developer values
(1,'password1'),
(2,'password2'),
(3,'password3'),
(4,'password4'),
(5,'password5'),
(6,'password6'),
(7,'password7'),
(8,'password8');

insert into repository values
(1),
(2),
(3),
(11);

insert into repository values
(4,1,1,10),
(5,1,2,12),
(6,1,3,4),
(7,2,5,11),
(8,2,7,6),
(9,3,6,12),
(10,3,5,11),
(12,11,8,14);

insert into branch values
(1, 1,  '04:05:06'),
(1, 4,  '04:05:06'),
(1, 5,  '04:05:06'),
(1, 6,  '04:05:06'),
(1, 2,  '01:01:01'),
(1, 7,  '01:01:01'),
(1, 8,  '01:01:01'),
(1, 3,  '02:01:01'),
(1, 9,  '02:01:01'),
(1, 10, '02:01:01'),
(1, 11, '03:00:59'),
(1, 12, '03:01:04');

insert into branch values
(2, 1,  '04:05:07', 1,  1),
(3, 1,  '04:05:08', 1,  1),
(4, 1,  '04:05:09', 2,  1),
(2, 4,  '04:05:07', 1,  4),
(2, 5,  '04:05:07', 1,  5),
(4, 5,  '04:05:09', 2,  5),
(2, 6,  '04:05:07', 1,  6),
(3, 6,  '04:05:08', 1,  6),
(2, 2,  '01:01:02', 1,  2),
(3, 2,  '01:01:03', 1,  2),
(4, 2,  '01:01:04', 1,  2),
(2, 7,  '01:01:02', 1,  7),
(4, 7,  '01:01:04', 1,  7),
(2, 8,  '01:01:02', 1,  8),
(3, 8,  '01:01:03', 1,  8),
(2, 3,  '02:01:02', 1,  3),
(3, 3,  '02:01:03', 2,  3),
(4, 3,  '02:01:04', 2,  3),
(2, 9,  '02:01:02', 1,  9),
(3, 9,  '02:01:03', 2,  9),
(2, 10, '02:01:02', 1,  10),
(4, 10, '02:01:04', 2,  10),
(2, 11, '02:01:09', 1,  11),
(3, 11, '03:01:01', 2,  11),
(4, 11, '03:01:02', 1,  11),
(5, 11, '03:01:03', 4,  11),
(2, 12, '03:01:05', 1,  12),
(3, 12, '03:01:06', 2,  12),
(4, 12, '03:01:07', 1,  12),
(5, 12, '03:01:08', 4,  12);

insert into version values
(1, 1,  1),
(2, 1,  2),
(3, 1,  1),
(4, 1,  2),
(5, 1,  3),
(1, 2,  5),
(2, 2,  5),
(3, 2,  7),
(4, 2,  5),
(5, 2,  7),
(1, 3,  6),
(2, 3,  6),
(3, 3,  6),
(4, 3,  5),
(5, 3,  5),
(1, 11, 8),
(2, 11, 8),
(3, 11, 8),
(4, 11, 8),
(5, 11, 8);
```

Running 3 files



```
C:\Users\Uthpal>psql -U postgres -f C:\Users\Uthpal\Desktop\5CS\DBMS_github\createstatements.sql
Password for user postgres:
DROP DATABASE
CREATE DATABASE
You are now connected to database "github" as user "postgres".
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE

C:\Users\Uthpal>psql -U postgres -f C:\Users\Uthpal\Desktop\5CS\DBMS_github\insert1.sql
Password for user postgres:
You are now connected to database "github" as user "postgres".
INSERT 0 33
INSERT 0 52
INSERT 0 21
INSERT 0 4
INSERT 0 2
INSERT 0 6
INSERT 0 8
INSERT 0 4
INSERT 0 8
INSERT 0 12
INSERT 0 30
INSERT 0 20

C:\Users\Uthpal>psql -U postgres -f C:\Users\Uthpal\Desktop\5CS\DBMS_github\constraints.sql
Password for user postgres:
You are now connected to database "github" as user "postgres".
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
```

Contribution of each member

Designing DATABASE for GitHub is complex as it involved understanding GitHub in detail .

We regularly had team meetings almost every week though Microsoft Teams and finalised the entities and operations in our DBMS

We together made lot of sketches like about how a repository should look and how we perform operations like pull push merge etc…

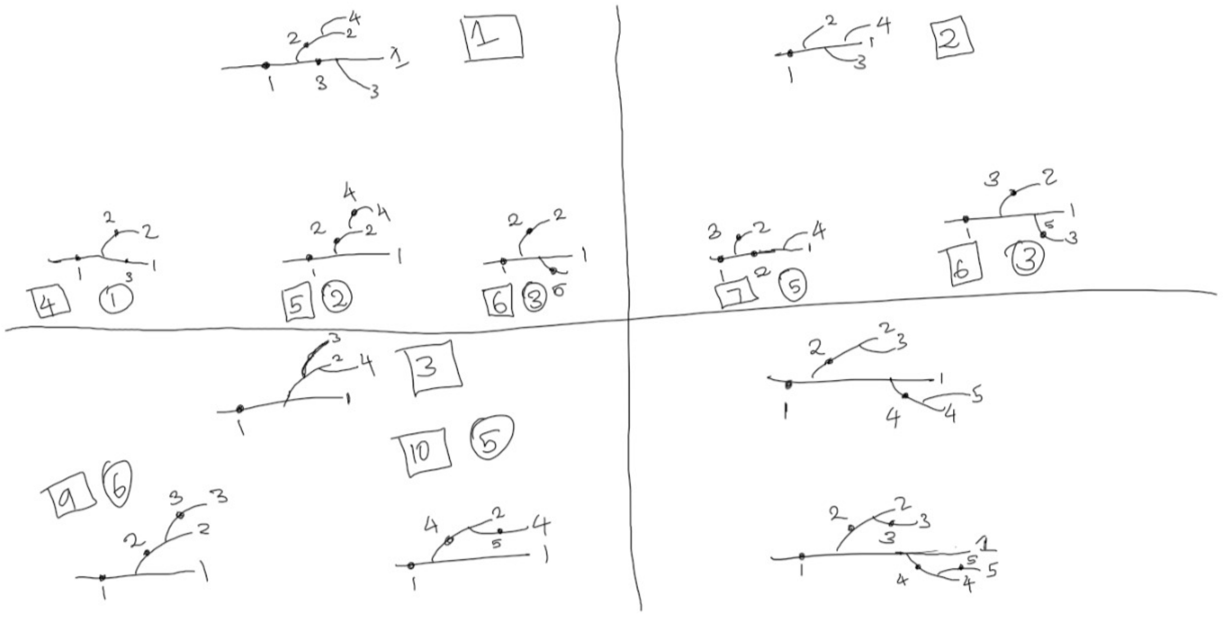
PES1UG19CS579 VISHWAS R
Converted ER diagram to Relational Schema and defined constraints.

PES1UG19CS548 UTHPAL P
Wrote the DLL statements.

PES1UG19CS534 T R SUDHARSHAN
Inserting data to the database.

Rough idea how our RDBMS works

Number in box is repo id, Number in circle is user id


1
2
4
1
5
2
6
3
7
5
6
3
3
10
5
9
6