SQL – Final Project Group No. 13

ID-316384130
ID-317975118

```
#########################################################################################################
########################## Part A – SCHEMA of NYC_GW – DB ################################################
#########################################################################################################
Create Database  nyc_gw;
USE NYC_GW;

CREATE TABLE `Stops` (
`stop_id` smallint unsigned not null auto_increment,
`stop_alias` varchar(4) not null,
`stop_description` varchar(50) not null,
`stop_address`varchar(50) not null,
primary key (`stop_id`));

CREATE TABLE `Line` (
  `line_idx` smallint unsigned not null auto_increment,
  `line_id` smallint unsigned not null,###blue/red/etc ---> 1...5
  `stop_id` smallint unsigned not null,### foreign key of bus_stops CLASS
  `line_alias` varchar(30)not null,
  PRIMARY KEY (`line_idx`),
  KEY `idx_fk_stop_line` (`stop_id`),
  Foreign Key `fk_stop_id`(`stop_id`) references `stops`(`stop_id`));

 Create TABLE `route`(
`route_id` int auto_increment,
`up_down_route` smallint unsigned not null, ###1 or o up[1](clockwise or down[0](counter-clockwise)
`Stop_Num` smallint unsigned not null,
`u_d_description` varchar(30),
`line_idx` smallint unsigned not null,
`line_id` smallint unsigned not null,
foreign key `fk_line_idx`(`line_idx`) references `line`(`line_idx`),
primary key (`route_id`)
);

CREATE TABLE `Bus` (
    `bus_idx` smallint unsigned not null auto_increment,
    `license_no` varchar(30) NOT NULL,
    `line_id` smallint unsigned not null,
    `line_idx` smallint unsigned null,
    `cap_id` smallint unsigned not null,
       KEY `idx_fk_bus` (`line_idx`),
    foreign key `fk_line_id`(`line_idx`) references `line`(`line_idx`),
    PRIMARY KEY (`bus_idx`));
    DELIMITER //
CREATE TRIGGER `bus_license_verifier`
       BEFORE INSERT
       ON `bus`
       FOR EACH ROW
BEGIN
       IF (NEW.license_no LIKE '__-___-__' or new.license_no like '___-__-___') THEN
              set NEW.license_no = NEW.license_no  ;
       else
       SIGNAL SQLSTATE VALUE '45000'
                     SET MESSAGE_TEXT = '[Table:Bus] – `License_No` column should consist of 7
charecters';
       END IF;
END //
DELIMITER ;
create table `capacity`(
`cap_id` int auto_increment,
`cap_desc` varchar(30) not null,
`max_cap` smallint unsigned not null,
`bus_idx` smallint unsigned NULL,
foreign key `fk_cap_bus`(`bus_idx`) references `bus`(`bus_idx`),
primary key(`cap_id`));
create table `tag` (
`tag_id` int auto_increment,
`tag_description` varchar(30),
`bus_idx` smallint unsigned NULL,
foreign key `fk_tag_bus`(`bus_idx`) references `bus`(`bus_idx`),
primary key (`tag_id`));
```

```
##########################################################################################
######################## Part B – Population Of Data Base##################################
##########################################################################################
Insert into `nyc_gw`.`stops`
(stop_alias,stop_description,stop_address)
Values
 ('A1','1st & Alex',"1 Alex St.– West Village,NY"), ('B1','1st & Beth',"1 Beth St. – West
Village,NY"), ('C1','1st & Charles',"1 Charles St. – West Village,NY"), ('D1','1st & David',"1 David
St. – West Village,NY"),
 ('A2','2nd & Alex',"2 Alex St. – West Village,NY"), ('B2','2nd & Beth',"2 Beth St. – West
Village,NY"), ('C2','2nd & Charles',"2 Charles St. – West Village,NY"), ('D2','2nd & David',"2 David
St. – West Village,NY"),
 ('A3','3rd & Alex',"3 Alex St. – West Village,NY"), ('B3','3rd & Beth',"3 Beth St. – West
Village,NY"), ('C3','3rd & Charles',"3 Charles St. – West Village,NY"),
                                                ('C4','4th & Charlie',"4 Charlie St. – West
Village,NY"), ('D4','4th & David',"4 David St. – West Village,NY"),('E4','4th & Ethan',"4 Ethan St.
– West Village,NY"),
                                                                                        ('E5','5th
& Ethan',"5 Ethan St. – West Village,NY");

INSERT INTO `nyc_gw`.`Line`
(`line_id`,`stop_id`,`line_alias`)
VALUES
(1,3,"(R)obin Ex."),(1,7,"(R)obin Ex."),(1,6,"(R)obin Ex."),(1,2,"(R)obin Ex."),
(2,9,"(G)arred Ex."),(2,5,"(G)arred Ex."),(2,6,"(G)arred Ex."),(2,10,"(G)arred Ex."),
(3,1,"(B)runo Ex."),(3,6,"(B)runo Ex."),(3,11,"(B)runo Ex."),(3,13,"(B)runo Ex."),
(4,4,"(Y)anis Ex."),(4,8,"(Y)anis Ex."),(4,7,"(Y)anis Ex."),(4,3,"(Y)anis Ex."),
(5,14,"(P)ablo Ex."),(5,15,"(P)ablo Ex."),(5,12,"(P)ablo Ex."),(5,13,"(P)ablo Ex.");
insert into route(line_id,stop_num,up_down_route,u_d_description,line_idx)
 VALUES
 (1,1,1,"Up-Route",1),(1,2,1,"Up-Route",2),(1,3,1,"Up-Route",3),(1,4,1,"Up-Route",4),
 (1,1,0,"Down-Route",4),(1,2,0,"Down-Route",3),(1,3,0,"Down-Route",2),(1,4,0,"Down-Route",1),

(2,1,1,"Up-Route",5),(2,2,1,"Up-Route",6),(2,3,1,"Up-Route",7),(2,4,1,"Up-Route",8),
(2,1,0,"Down-Route",8),(2,2,0,"Down-Route",7),(2,3,0,"Down-Route",6),(2,4,0,"Down-Route",5),

(3,1,1,"Up-Route",9),(3,2,1,"Up-Route",10),(3,3,1,"Up-Route",11),(3,4,1,"Up-Route",12),
(3,1,0,"Down-Route",12),(3,2,0,"Down-Route",11),(3,3,0,"Down-Route",10),(3,4,0,"Down-Route",9),

(4,1,1,"Up-Route",13),(4,2,1,"Up-Route",14),(4,3,1,"Up-Route",15),(4,4,1,"Up-Route",16),
(4,1,0,"Down-Route",16),(4,2,0,"Down-Route",15),(4,3,0,"Down-Route",14),(4,4,0,"Down-Route",13),

(5,1,1,"Up-Route",17),(5,2,1,"Up-Route",18),(5,3,1,"Up-Route",19),(5,4,1,"Up-Route",20),
(5,1,0,"Down-Route",20),(5,2,0,"Down-Route",19),(5,3,0,"Down-Route",18),(5,4,0,"Down-Route",17);

INSERT INTO `bus`
(license_no,line_id,cap_id)
VALUES
('10-000-01',1,1),
('10-000-02',1,1),
('10-000-03',2,2),
('10-000-04',2,2),
('10-000-05',3,3),
('10-000-06',3,3),
('10-000-07',4,4),
('10-000-08',4,4),
('10-000-09',5,5),
('10-000-10',5,5);

Insert into `capacity`(cap_desc,max_cap) Values
('Small',8),
('Medium',16),
('Large',32),
('Extra Large',50),
('Double',64);

Insert into tag(tag_description,bus_idx) Values
('Electric Motor',1),
('Gas Motor',2),
('Hybrid Motor',3),
('Diesel Motor',4),
('Rotary Motor',5);
```

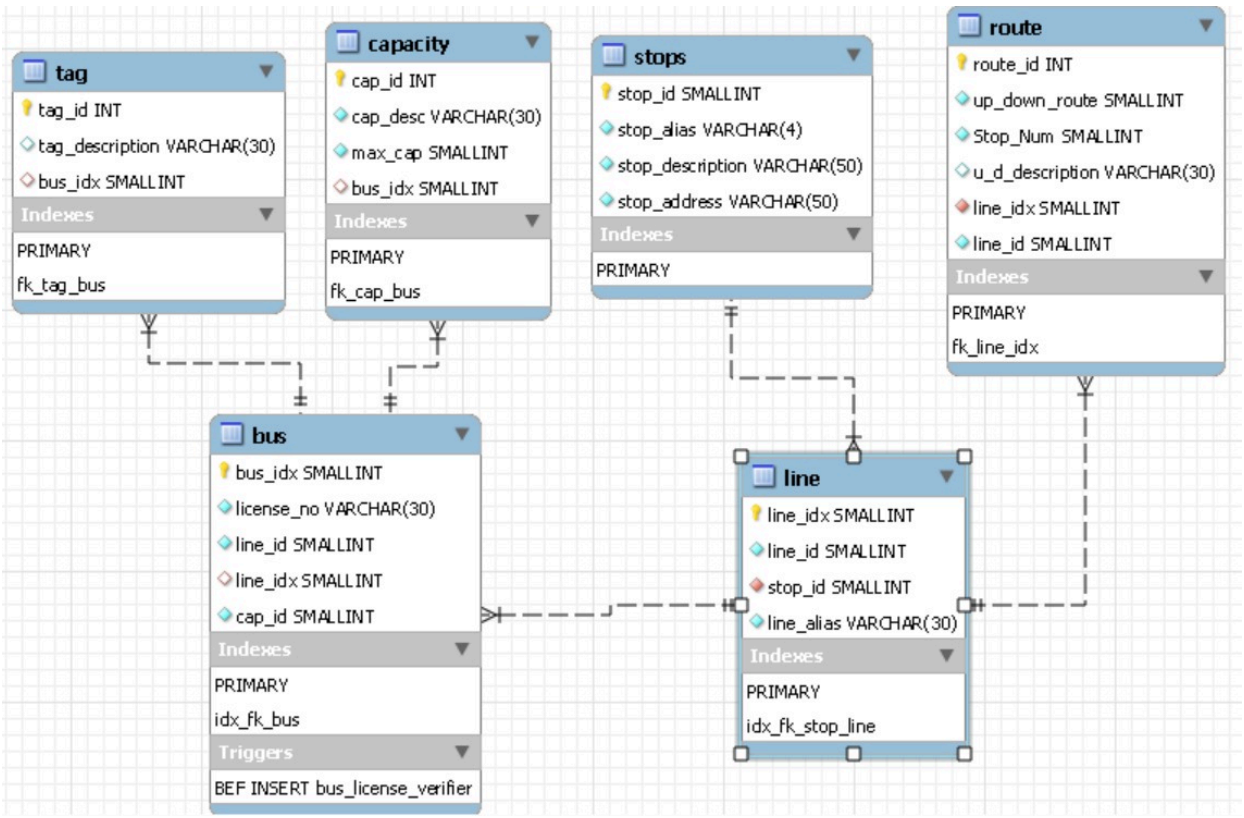
```
#########################################################################################################
######################### Part – C –Select Statements Queries ###########################################
#########################################################################################################
######################################## First Query ####################################################
Use nyc_gw;
Select case when (select R.Stop_Num from route R join line L  on R.line_id=L.line_id where
L.line_id =1 and R.up_down_route=1
Order by R.line_idx Desc Limit 1)=(Select R.Stop_Num from route R join line L  on
R.line_id=L.line_id where  L.line_id =1 and R.up_down_route=0
Order by R.line_idx Asc LIMIT 1)
And
(Select R.Stop_Num From route R Join line L  on R.line_id=L.line_id where  L.line_id =1 and
R.up_down_route=1
Order by R.line_idx Asc Limit 1)=(Select R.Stop_Num from route R join line L  on R.line_id=L.line_id
where  L.line_id =1 and R.up_down_route=0
Order by R.line_idx Desc Limit 1)
Then 1 Else 0 End As Proper_Loop;
```

	Proper_Loop
▶	1

```
######################################## Second Query ###################################################
Use nyc_gw;
Select stops.stop_description,stops.stop_address,cap_desc,avg(max_cap) from capacity join bus on
bus.cap_id=capacity.cap_id
Join line on line.line_id = bus.line_id
Join stops on stops.stop_id = line.stop_id where stops.stop_alias like "c2";
```

	stop_description	stop_address	cap_desc	avg(max_cap)
▶	2nd & Charles	2 Charles St. - West Village,NY	Small	29.0000

```
#########################################################################################################
################################## Part – D –Appendix ###################################################
#########################################################################################################
```



line_id	line_alias	stop_id	stop_1	stop_2	stop_3	stop_4
1	red	1	c1	c2	b2	b1
2	green	2	a3	a2	b2	b3
3	blue	3	a1	b2	c3	d4
4	yellow	4	d1	d2	c2	c1
5	purple	5	e4	e5	c4	d4

LINES

line_idx(PK)	line_id	stop_alias	stop_id
1	1	c1	3
2	1	c2	7
3	1	b2	6
4	1	b1	2
5	2	a3	9
6	2	a2	5
7	2	b2	6
8	2	b3	10
9	3	a1	1
10	3	b2	6
11	3	c3	11
12	3	d4	13
13	4	d1	4
14	4	d2	8
15	4	c2	7
16	4	c1	3
17	5	e4	14
18	5	e5	15
19	5	c4	12
20	5	d4	13

(R)obin Ex.	1
(G)arred Ex.	2
(B)runo Ex.	3
(Y)anis Ex.	4
(P)ablo Ex.	5