7/27/2022

NAMES: NDAGIJIMANA Benoit

REG: 221011094

School of Business

DEPARTIMENT: BIT L2 GROUP 2

Module:

BUSINESS DEVELOPMRNT

DATABASE DEVELOPMENT CAT

TOPIC: *BUS O*NLINE TICKET RESERVATION SYSTEM

Section I Part1: Entities descriptions

This is system has a database called Ticket Reservation which composed of six tables namely:

1. Clients table
2. Transport company managers table
3. Tickets table
4. Bus tables
5. Drivers' tables
6. Transport companies table

Clients' tables

Clients table records all information related to the clients and has the following attributes:

- Client id
- Client names
- Client age
- Client_contacts
- Client gender
- Client location

Transport company managers table

Transport company managers table records information related to the managers of different transport company and has the following attributes:

- Manager id
- Manager names

- Manager location
- Manager office location
- Manager office room number

Tickets table

Tickets table records all ticket details of the tickets created and has the following attributes:

- Ticket id
- Created date
- Client id
- Manager id
- Bus_id
- Amount paid

Buses tables

Buses table record information related to the buses and has the following attributes:

- Buse id
- Bus model
- Bus plate
- Bus number of sits
- Bus manufactured year
- Company id

Drivers' table

Drivers' table record information related to the drivers and has the following attributes:

- Driver id
- Driver names
- Driver location

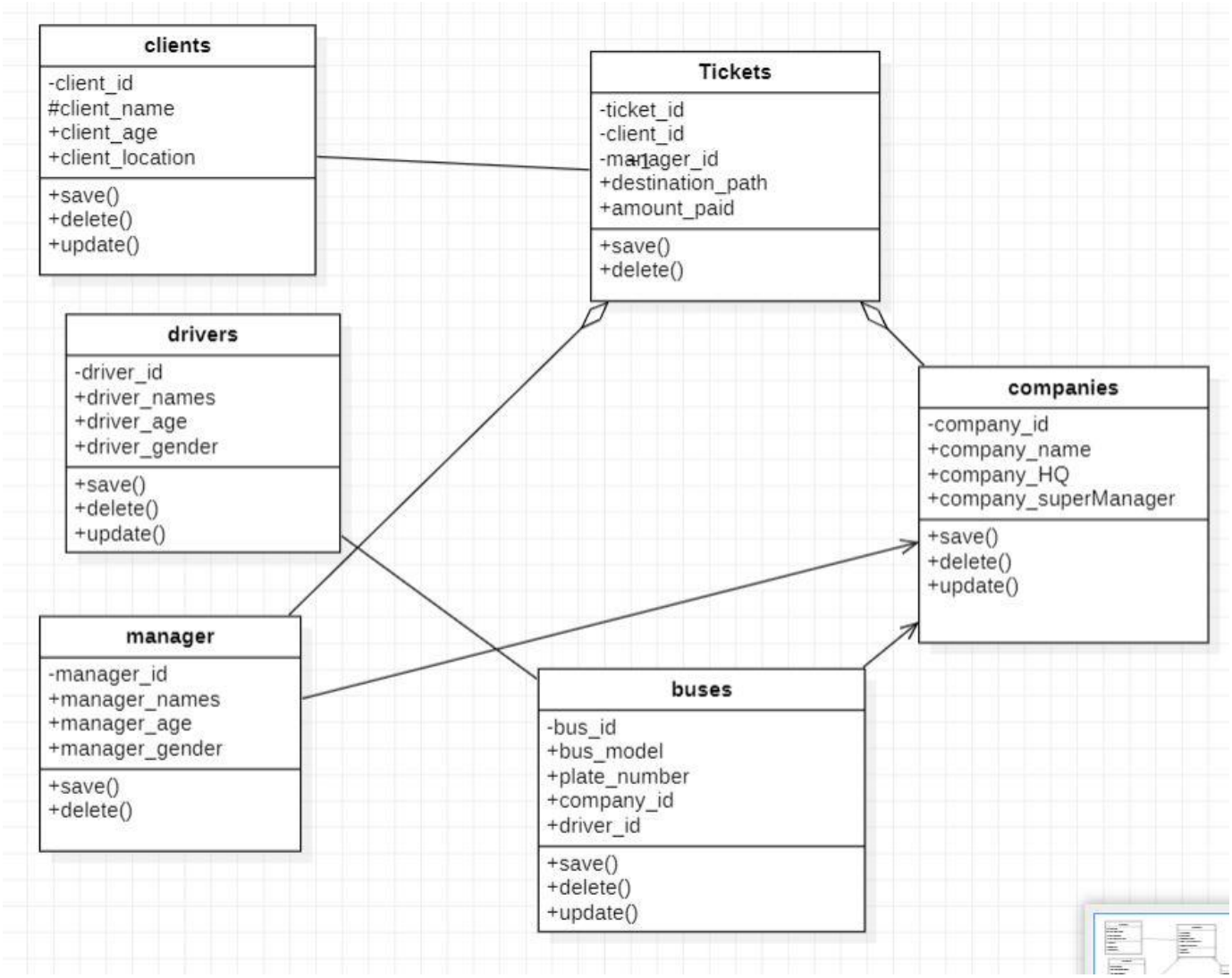
- Driver gender
- Driver age
- Company id

Transport companies' table

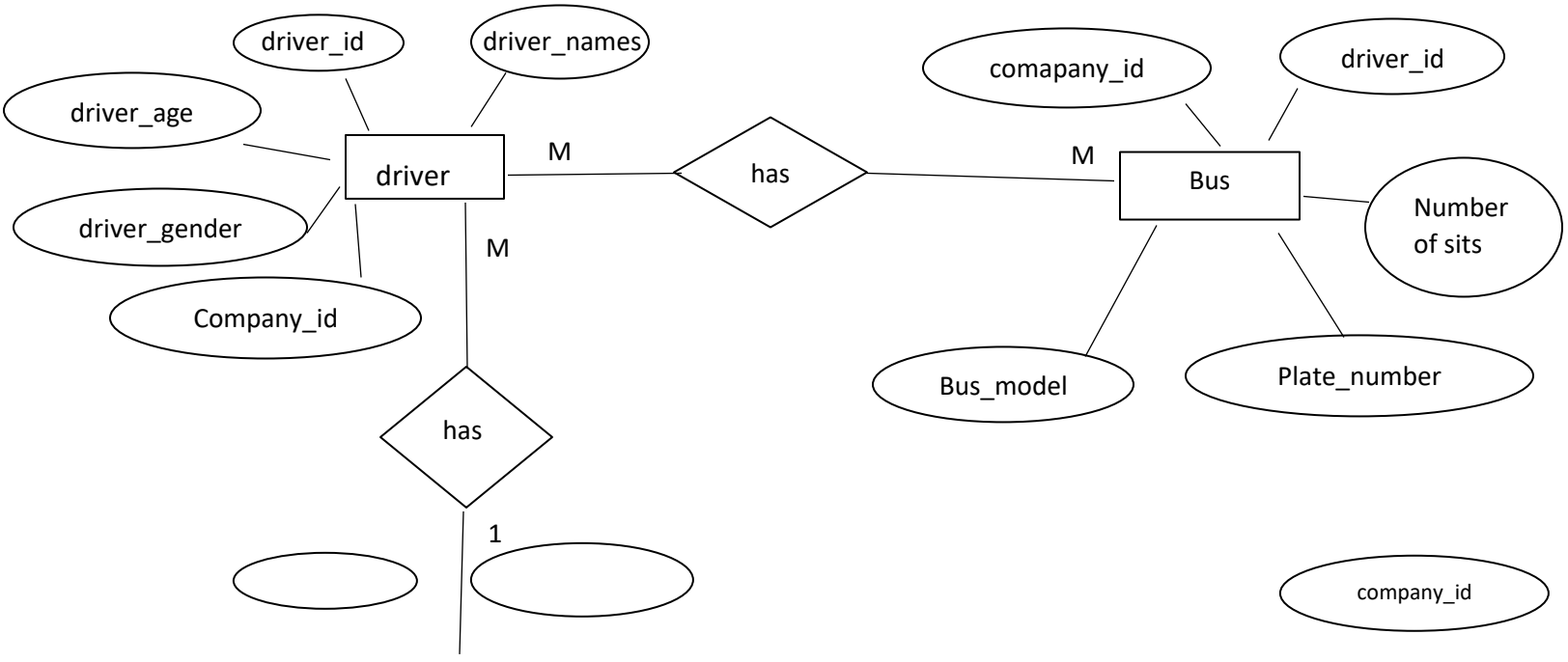
This table records information of the transport companies which eligible to serve their clients using this system and has the following attributes:

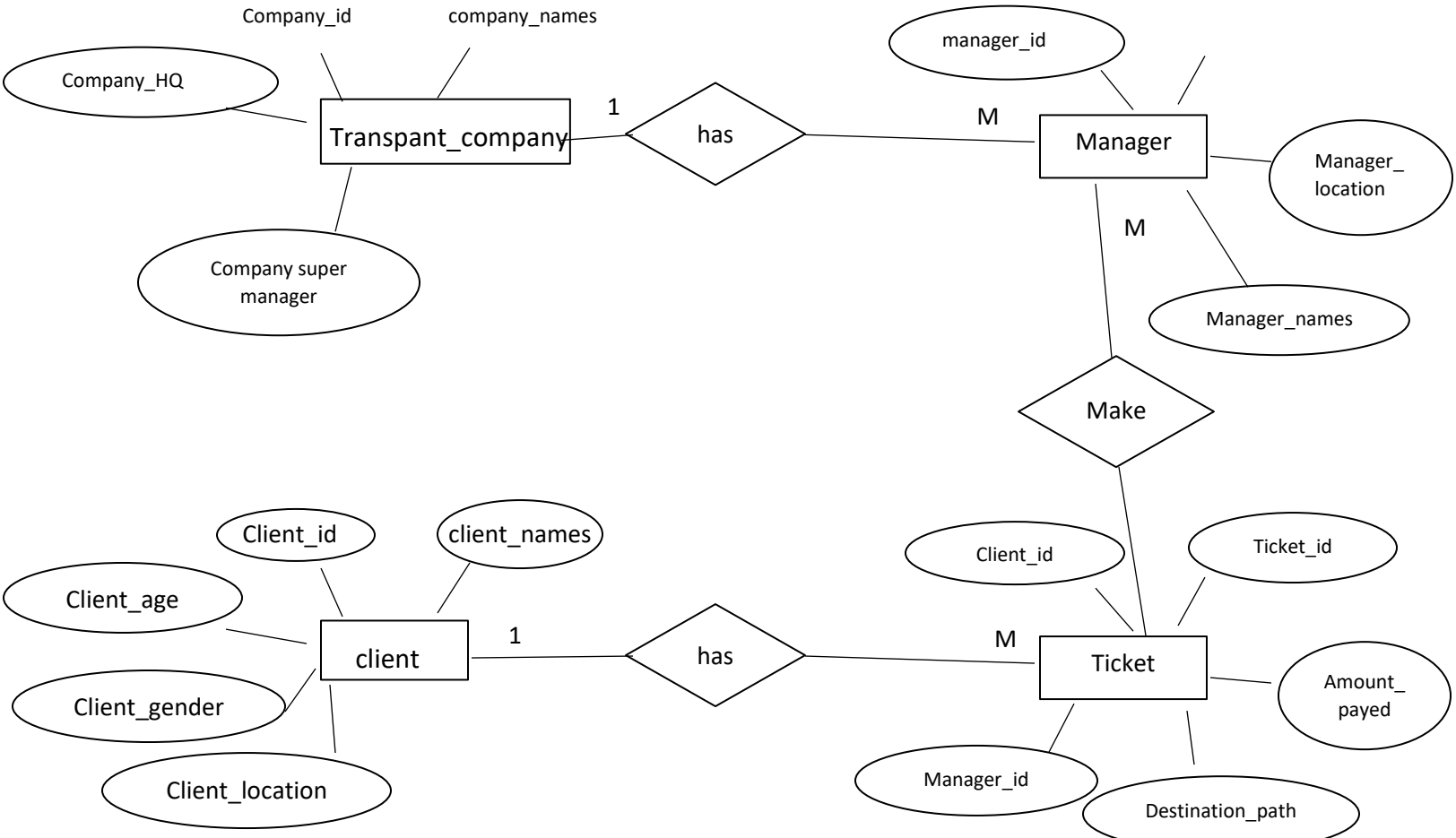
- Company id
- Bus id
- Company name
- Company heard quarter
- Company super manger

Section I part 2: Logical data model diagram


clients
-client_id
#client_name
+client_age
+client_location
+save()
+delete()
+update()
Tickets
-ticket_id
-client_id
-manager_id
+destination_path
+amount_paid
+save()
+delete()
drivers
-driver_id
+driver_names
+driver_age
+driver_gender
+save()
+delete()
+update()
companies
-company_id
+company_name
+company_HQ
+company_superManager
+save()
+delete()
+update()
manager
-manager_id
+manager_names
+manager_age
+manager_gender
+save()
+delete()
buses
-bus_id
+bus_model
+plate_number
+company_id
+driver_id
+save()
+delete()
+update()

Part3: Entity Relationship Diagram





SECTION II: SQL QUERIES

Database: CREATE DATABASE ticket_reservation;

i. Create table command:

1. **<u>Clients</u>**

```
CREATE TABLE clients (
client_id INT,
client_names varchar(30),
client_location varchar (40),
client_age int,
client_gender varchar (10),
PRIMARY KEY (client_id))
```

2. **<u>Companies</u>**

```
CREATE TABLE comapanies (
company_id INT,
company_name varchar (30),
company_HQ varchar (40),
company_superManager varchar (40),
```

PRIMARY KEY (comapany_id))

3. Managers

CREATE TABLE managers (
manager_id INT,
manager_name varchar (30),
manager_location varchar (40),
company_id INT,
PRIMARY KEY (manager_id),
FOREIGN KEY (company_id) REFERENCES companies(company_id))

4. Driver

CREATE TABLE drivers (
driver_id INT,
driver_name varchar (30),
driver_age INT,
driver_gender varchar (10),
company_id INT,
PRIMARY KEY (driver_id),
FOREIGN KEY (company_id) REFERENCES companies(company_id))

5. Buses

CREATE TABLE buses (
bus_id INT,
bus_model varchar (30),
plate_number INT,
manufactured_date date,
numberOfSits INT,
driver_id INT,
company_id INT,
PRIMARY KEY (driver_id),
FOREIGN KEY (company_id) REFERENCES companies(company_id),
FOREIGN KEY (driver_id) REFERENCES drivers(company_id))

6. Tickets

CREATE TABLE tickets (

ticket_id INT,

created_date varchar (30),

client_id INT,

manager_id INT,

bus_id INT,

amount_paid INT,

PRIMARY KEY (ticket_id),

FOREIGN KEY (manager_id) REFERENCES managers(manager_id),

FOREIGN KEY (client_id) REFERENCES clients(client_id),

FOREIGN KEY (bus_id) REFERENCES buses(bus_id))

ii. Insert into table commands

1. clients

INSERT INTO `clients`(`client_id`, `client_names`, `client_location`, `client_age`, `client_gender`) VALUES ('1','ben gray','huye',25,'male'),

('2','jeiden james','kigali',26,'male'),

('3','bryan luiz','huye',25,'female')

2. companies

INSERT INTO `companies` (`company_id`, `company_name`, `company_HQ`, `company_superManager`) VALUES (1,'volcano','kigali','olivier'),

(2,'horizon','kigali','cloude'*)*

3. Managers

INSERT INTO `managers`(`manager_id`, `manager_name`, `manager_location`, `company_id`) VALUES (1,'jelome karangwa','huye',1),
(2,'viateur mugabo','huye',2)

4. Drivers

INSERT INTO `drivers`(`driver_id`, `driver_name`, `driver_age`, `driver_gender`, `company_id`) VALUES (1,'peter kagabo',30,'male','1'),
(2,'john kagabo',30,'male','2')

5. Buses

INSERT INTO `buses`(`bus_id`, `bus_model`, `plate_number`, `manufactured_date`, `numberOfSits`, `driver_id`, `company_id`) VALUES (1,'vox wagen v2334','RAE302','12-JAN-2022',60,1,1)

6. Tickets

INSERT INTO `tickets`(`ticket_id`, `created_date`, `client_id`, `manager_id`, `bus_id`, `amount_paid`) VALUES (1,10-02-2022,1,1,1,2500)

iii. SELECT QUERY

Tables:

1. **Clients**
   SELECT * FROM `clients`
2. **Companies**
   SELECT * FROM `companies`
3. **Managers**
   SELECT * FROM `managers`
   SELECT * FROM `managers`JOIN companies ON manager_id=companies.company_id
4. **Drivers**
   SELECT * FROM `drivers` WHERE drivers.driver_id =drivers.company_id
   SELECT * FROM `drivers`JOIN companies ON driver_id=companies.company_id
5. **Buses**
   SELECT * FROM `buses`
   SELECT * FROM `buses` JOIN companies ON bus_id=companies.company_id
6. **Tickets**
   SELECT * FROM `tickets`

iv. Update query

Tables: 1. clients

UPDATE `clients` SET `client_names`='steven jei',
`client_location`='musanze',`client_age`=32,`client_gender`='male' WHERE client_id =1

2. managers

UPDATE `managers` SET `manager_name`='anne kwizera',`manager_location`='nyanza'
WHERE manager_id= 1

Views to display all information:

create view ticketview AS SELECT
clients.client_names,managers.manager_name,tickets.amount_paid FROM
clients,managers,tickets

create view busesview AS SELECT
buses.bus_model,managers.manager_name,tickets.amount_paid FROM
buses,managers,tickets

INSERT INTO `busesview`(`bus_id`, `bus_model`, `plate_number`, `manufactured_date`, `numberOfSits`, `driver_id`, `company_id`) VALUES (5,'vox wagen v2334','RAE302','12-JAN-2022',60,1,1)

Section IV

Standard procedure to SHOW

```
CREATE PROCEDURE `selectClients`(IN `client_id` INT(11), IN `client_names` VARCHAR(30), IN `client_loaction` VARCHAR(30), IN `client_age` INT(11), IN `client_gender` INT(10)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT * FROM clients
```

Standard procedure to insert

```
CREATE PROCEDURE `Clients`(IN `client_id` INT(11) UNSIGNED, IN `client_names` VARCHAR(30), IN `client_loaction` VARCHAR(30), IN `client_age` INT(11), IN `client_gender` VARCHAR(10)) DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER INSERT INTO clients VALUES (1,'benkalisa','kenya',35,'male')
```

Standard procedure to update

TABLE:clients

```
CREATE PROCEDURE `updateClients`(IN `client_id` INT, IN `client_names` VARCHAR(30), IN `client_loaction` VARCHAR(30), IN `client_age` INT(11), IN `client_gender` VARCHAR(10)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER UPDATE clients SET clients_names='kalisa james',client_location='rwanda' WHERE client_id=2
```

Table:managers

```
CREATE PROCEDURE `updateClients`(IN `manager_id` INT, IN `manager_names` VARCHAR(30), IN `manager_loaction` VARCHAR(30), IN `manager_age` INT(11), IN `manager_gender` VARCHAR(10)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER UPDATE manager SET manager_names='kalisa james',manager_location='rwanda' WHERE manager_id=2
```

Standard procedure to delete

```
CREATE PROCEDURE `deleteclients`(IN `client_id` INT(11), IN `client_names` VARCHAR(30), IN `client_loaction` VARCHAR(30), IN `client_age` INT(11), IN `client_gender` VARCHAR(10)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER DELETE FROM clients WHERE client_id=1S
```

Stored procedure for view

```
CREATE PROCEDURE `viewprocedure`(IN `client_id` INT(11)
, IN `client_names` VARCHAR(30), IN `client_loaction` V
ARCHAR(30), IN `client_age` INT(11), IN `client_gender`
 VARCHAR(10)) NOT DETERMINISTIC CONTAINS SQL SQL SECURI
TY DEFINER CREATE VIEW clientView AS SELECT * FROM clie
nts
```

Section V

Triggers

I. Create trigger and after insert for buses table

```
CREATE TRIGGER `InsertIntoBuses` AFTER INSERT ON `buses
` FOR EACH ROW INSERT INTO buses VALUES(5,'HUNDAI NEW
MODEL','RAE305',1,1)
```

Create trigger and after insert for companies table

```
CREATE TRIGGER `companiesTriggerInsert` AFTER INSERT ON
 `companies` FOR EACH ROW INSERT INTO companies values(
6,'omega','karenzi christopher','kigali')
```

ii. **create trigger and update for buses table**

```
CREATE TRIGGER `updatebusesTrigger` AFTER INSERT ON `bu
ses` FOR EACH ROW UPDATE buses SET bus_model='benz' WHE
RE bus_id=2
```

create trigger and update for companies table

```
CREATE TRIGGER `companiesTriggerUpdate` AFTER INSERT ON
 `buses` FOR EACH ROW UPDATE companies SET company_name
='ritco' where company_id=2
```

iii. Create after delete triggers for companies' table

```
CREATE TRIGGER `busesDeleteTrigger` AFTER DELETE ON `bu
ses` FOR EACH ROW DELETE FROM buses WHERE bus_id=2
```

Create after delete triggers for companies' table

```
CREATE TRIGGER `deleteCompaniesTriggers` AFTER DELETE O
N `companies` FOR EACH ROW DELETE FROM companies WHERE
comapny_id=3
```

Section VI

CREATE USER ACCOUNTS AND GRANT PRIVILIGES

```
CREATE USER 'NDAGIJIAMANA BENOIT'@'%' GRANT SELECT,
INSERT, UPDATE, DELETE, CREATE, DROP, FILE, INDEX,
ALTER, CREATE TEMPORARY TABLES, CREATE VIEW, EVENT,
TRIGGER, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE,
EXECUTE ON *.* TO 'NDAGIJIAMANA BENOIT'@'%'
```

GRANT PRIVILIGES delete ,update and delete

```
CREATE USER 'NDAGIJIAMANA BENOIT'@'%' GRANT SELECT,
INSERT, UPDATE, DELETE , SHOW VIEW, CREATE ROUTINE,
ALTER ROUTINE, EXECUTE ON *.* TO 'NDAGIJIAMANA
BENOIT'@'%'
```