Lab 5 Report_Group6

Student_ID : 20151692, 20181071, 20181144, 20192006, 20192012, 20202008

We created a database using MySQL based on the database schema created in lab3.

The following parts were modified in the schema created in lab3.

1. There is a slight modification of the name of the table.
2. A new table is defined to represent the AD manager.

   AD_manager(staff_id)

3. Decompose the 'Equipment' table.

   Equipment(Lab_Name, Department_Name, Equipment_ID, Model_No,Date_purchased)

   and

   Model(model_no, model_name)

First, We creat a table based on lab3.

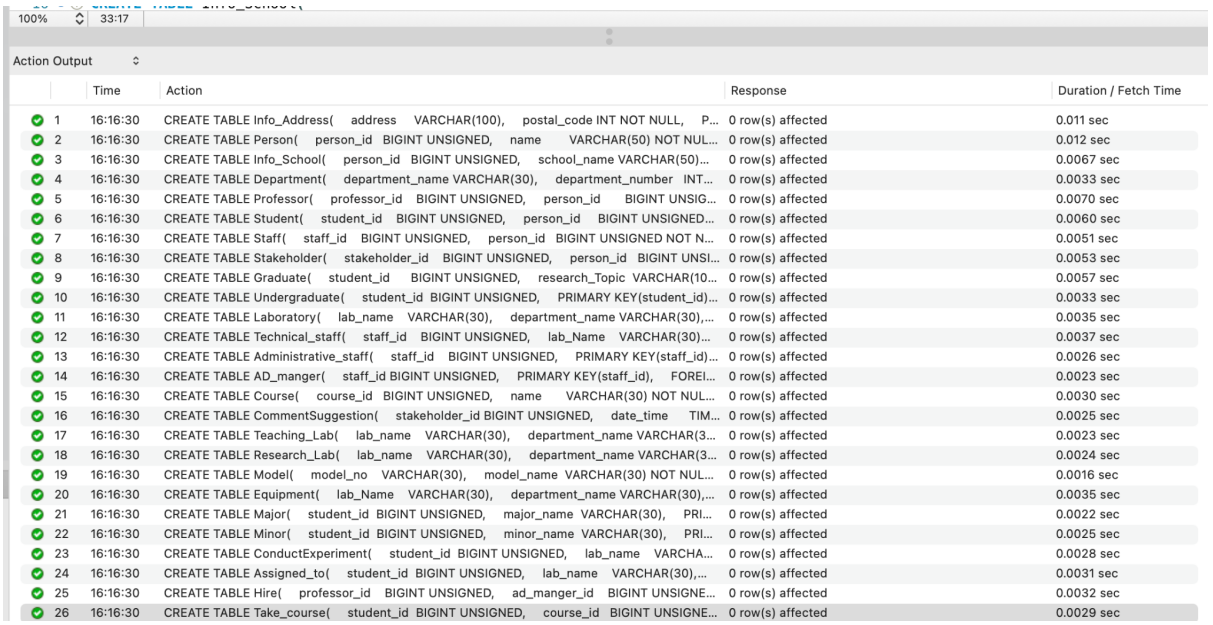
We made a total of 26 tables, and let's analyze the Graduate table as an example.

```
58 • CREATE TABLE Graduate(
59       student_id      BIGINT UNSIGNED,
60       research_Topic  VARCHAR(100) NOT NULL,
61       professor_id    BIGINT UNSIGNED NOT NULL,
62       PRIMARY KEY(student_id),
63       FOREIGN KEY(professor_id) REFERENCES Professor(professor_id) ON UPDATE CASCADE ON DELETE RESTRICT,
64       FOREIGN KEY(student_id) REFERENCES Student(student_id) ON UPDATE CASCADE ON DELETE CASCADE
65   );
```

The data type of each attribute is set and various constraints are specified.

ex)
1.For PK, use student_id.
2. The professor_id and the student_id refer to the Professor and Student tables in FK, respectively.

3.The ON DELETE RESTRICT condition was set so that the profesor_id of the Profesor table referenced by the profesor_id of the Graduate table could not be deleted. This will be explained in more detail later in the trigger section.



100% 33:17

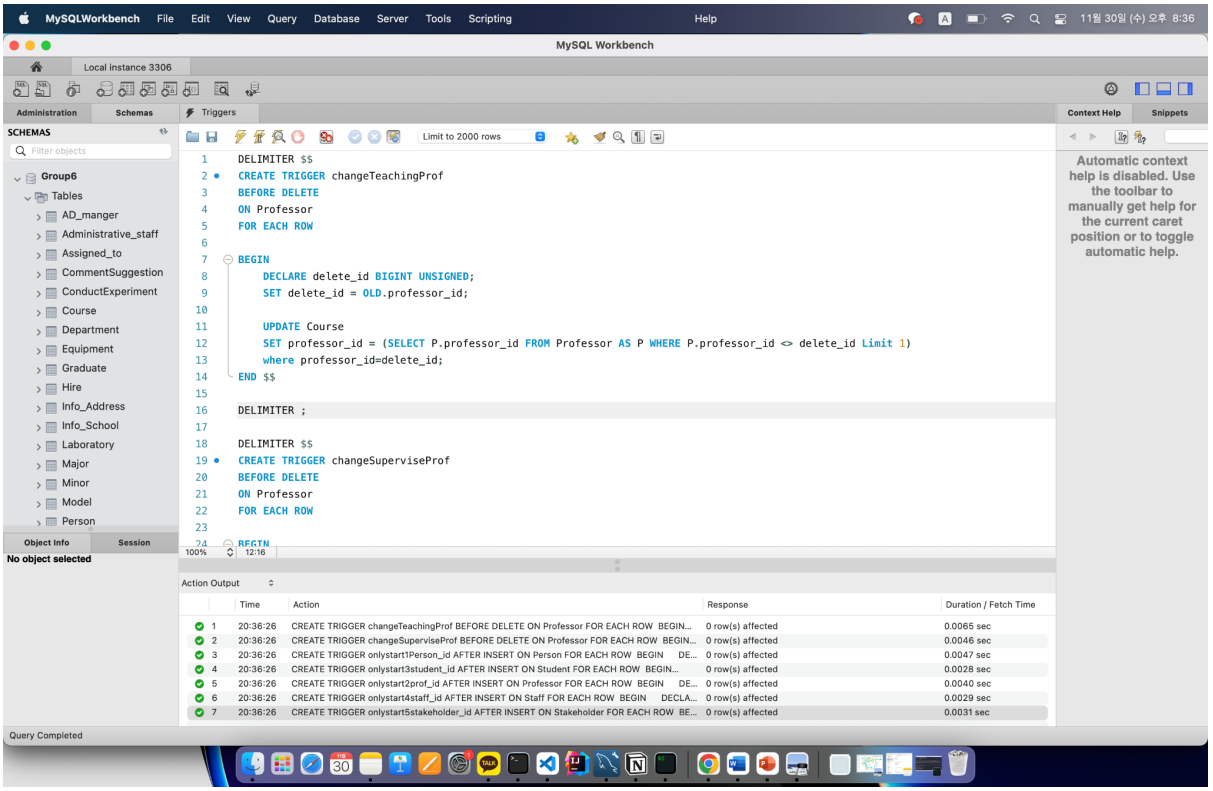
Action Output

	Time	Action	Response	Duration / Fetch Time
1	16:16:30	CREATE TABLE Info_Address(address VARCHAR(100), postal_code INT NOT NULL, P...	0 row(s) affected	0.011 sec
2	16:16:30	CREATE TABLE Person(person_id BIGINT UNSIGNED, name VARCHAR(50) NOT NUL...	0 row(s) affected	0.012 sec
3	16:16:30	CREATE TABLE Info_School(person_id BIGINT UNSIGNED, school_name VARCHAR(50)...	0 row(s) affected	0.0067 sec
4	16:16:30	CREATE TABLE Department(department_name VARCHAR(30), department_number INT...	0 row(s) affected	0.0033 sec
5	16:16:30	CREATE TABLE Professor(professor_id BIGINT UNSIGNED, person_id BIGINT UNSIG...	0 row(s) affected	0.0070 sec
6	16:16:30	CREATE TABLE Student(student_id BIGINT UNSIGNED, person_id BIGINT UNSIGNED...	0 row(s) affected	0.0060 sec
7	16:16:30	CREATE TABLE Staff(staff_id BIGINT UNSIGNED, person_id BIGINT UNSIGNED NOT N...	0 row(s) affected	0.0051 sec
8	16:16:30	CREATE TABLE Stakeholder(stakeholder_id BIGINT UNSIGNED, person_id BIGINT UNSI...	0 row(s) affected	0.0053 sec
9	16:16:30	CREATE TABLE Graduate(student_id BIGINT UNSIGNED, research_Topic VARCHAR(10...	0 row(s) affected	0.0057 sec
10	16:16:30	CREATE TABLE Undergraduate(student_id BIGINT UNSIGNED, PRIMARY KEY(student_id)...	0 row(s) affected	0.0033 sec
11	16:16:30	CREATE TABLE Laboratory(lab_name VARCHAR(30), department_name VARCHAR(30),...	0 row(s) affected	0.0035 sec
12	16:16:30	CREATE TABLE Technical_staff(staff_id BIGINT UNSIGNED, lab_Name VARCHAR(30)...	0 row(s) affected	0.0037 sec
13	16:16:30	CREATE TABLE Administrative_staff(staff_id BIGINT UNSIGNED, PRIMARY KEY(staff_id)...	0 row(s) affected	0.0026 sec
14	16:16:30	CREATE TABLE AD_manger(staff_id BIGINT UNSIGNED, PRIMARY KEY(staff_id), FOREI...	0 row(s) affected	0.0023 sec
15	16:16:30	CREATE TABLE Course(course_id BIGINT UNSIGNED, name VARCHAR(30) NOT NUL...	0 row(s) affected	0.0030 sec
16	16:16:30	CREATE TABLE CommentSuggestion(stakeholder_id BIGINT UNSIGNED, date_time TIM...	0 row(s) affected	0.0025 sec
17	16:16:30	CREATE TABLE Teaching_Lab(lab_name VARCHAR(30), department_name VARCHAR(3...	0 row(s) affected	0.0023 sec
18	16:16:30	CREATE TABLE Research_Lab(lab_name VARCHAR(30), department_name VARCHAR(3...	0 row(s) affected	0.0024 sec
19	16:16:30	CREATE TABLE Model(model_no VARCHAR(30), model_name VARCHAR(30) NOT NUL...	0 row(s) affected	0.0016 sec
20	16:16:30	CREATE TABLE Equipment(lab_Name VARCHAR(30), department_name VARCHAR(30),...	0 row(s) affected	0.0035 sec
21	16:16:30	CREATE TABLE Major(student_id BIGINT UNSIGNED, major_name VARCHAR(30), PRI...	0 row(s) affected	0.0022 sec
22	16:16:30	CREATE TABLE Minor(student_id BIGINT UNSIGNED, minor_name VARCHAR(30), PRI...	0 row(s) affected	0.0025 sec
23	16:16:30	CREATE TABLE ConductExperiment(student_id BIGINT UNSIGNED, lab_name VARCHA...	0 row(s) affected	0.0028 sec
24	16:16:30	CREATE TABLE Assigned_to(student_id BIGINT UNSIGNED, lab_name VARCHAR(30),...	0 row(s) affected	0.0031 sec
25	16:16:30	CREATE TABLE Hire(professor_id BIGINT UNSIGNED, ad_manger_id BIGINT UNSIGNE...	0 row(s) affected	0.0032 sec
26	16:16:30	CREATE TABLE Take_course(student_id BIGINT UNSIGNED, course_id BIGINT UNSIGNE...	0 row(s) affected	0.0029 sec

I will submit SQL DDL queries including constraints as a separate file.(CreateTable.sql)

Second, TRIGGERs.

We made a total of seven TRIGGERS.



MySQLWorkbench File Edit View Query Database Server Tools Scripting Help

MySQL Workbench

Local instance 3306

Administration Schemas Triggers Context Help Snippets

SCHEMAS

Filter objects

Group6
Tables
AD_manger
Administrative_staff
Assigned_to
CommentSuggestion
ConductExperiment
Course
Department
Equipment
Graduate
Hire
Info_Address
Info_School
Laboratory
Major
Minor
Model
Person

Object Info Session

No object selected

Limit to 2000 rows

```
1  DELIMITER $$
2  CREATE TRIGGER changeTeachingProf
3  BEFORE DELETE
4  ON Professor
5  FOR EACH ROW
6
7  BEGIN
8      DECLARE delete_id BIGINT UNSIGNED;
9      SET delete_id = OLD.professor_id;
10
11     UPDATE Course
12     SET professor_id = (SELECT P.professor_id FROM Professor AS P WHERE P.professor_id <> delete_id Limit 1)
13     where professor_id=delete_id;
14 END $$
15
16 DELIMITER ;
17
18 DELIMITER $$
19 CREATE TRIGGER changeSuperviseProf
20 BEFORE DELETE
21 ON Professor
22 FOR EACH ROW
23
24 BEGIN
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

100% 12:16

Action Output

	Time	Action	Response	Duration / Fetch Time
1	20:36:26	CREATE TRIGGER changeTeachingProf BEFORE DELETE ON Professor FOR EACH ROW BEGIN...	0 row(s) affected	0.0065 sec
2	20:36:26	CREATE TRIGGER changeSuperviseProf BEFORE DELETE ON Professor FOR EACH ROW BEGIN...	0 row(s) affected	0.0046 sec
3	20:36:26	CREATE TRIGGER onlystart1Person_id AFTER INSERT ON Person FOR EACH ROW BEGIN DE...	0 row(s) affected	0.0047 sec
4	20:36:26	CREATE TRIGGER onlystart3student_id AFTER INSERT ON Student FOR EACH ROW BEGIN...	0 row(s) affected	0.0028 sec
5	20:36:26	CREATE TRIGGER onlystart2prof_id AFTER INSERT ON Professor FOR EACH ROW BEGIN DE...	0 row(s) affected	0.0040 sec
6	20:36:26	CREATE TRIGGER onlystart4staff_id AFTER INSERT ON Staff FOR EACH ROW BEGIN DECLA...	0 row(s) affected	0.0029 sec
7	20:36:26	CREATE TRIGGER onlystart5stakeholder_id AFTER INSERT ON Stakeholder FOR EACH ROW BE...	0 row(s) affected	0.0031 sec

Query Completed

1.changeTeachingProf

When a professor teaching a particular course is deleted from the table, it is a trigger that causes another professor to teach the course.
Basically, the professor ID, which is a foreign key of the Course table, is set to ON DELETE RESTRICT, so the referenced professor_id cannot be deleted from the Professor table.
However, since there may be cases where it must be deleted, it is possible to refer to other professors before deleting and then delete them.

2.changeSuperviseProf

This is a trigger that allows another professor to supervise a student when a professor who supervises a Graduate is deleted from the table.
It is similar to the above case.

3.onlystart1Person_id

This is a trigger that allows only integers beginning with 1 as person_id.

4.onlystart3student_id

This is a trigger that allows only integers beginning with 1 as person_id.

5.onlystart2prof_id

This is a trigger that allows only integers beginning with 1 as person_id.

6.onlystart4staff_id

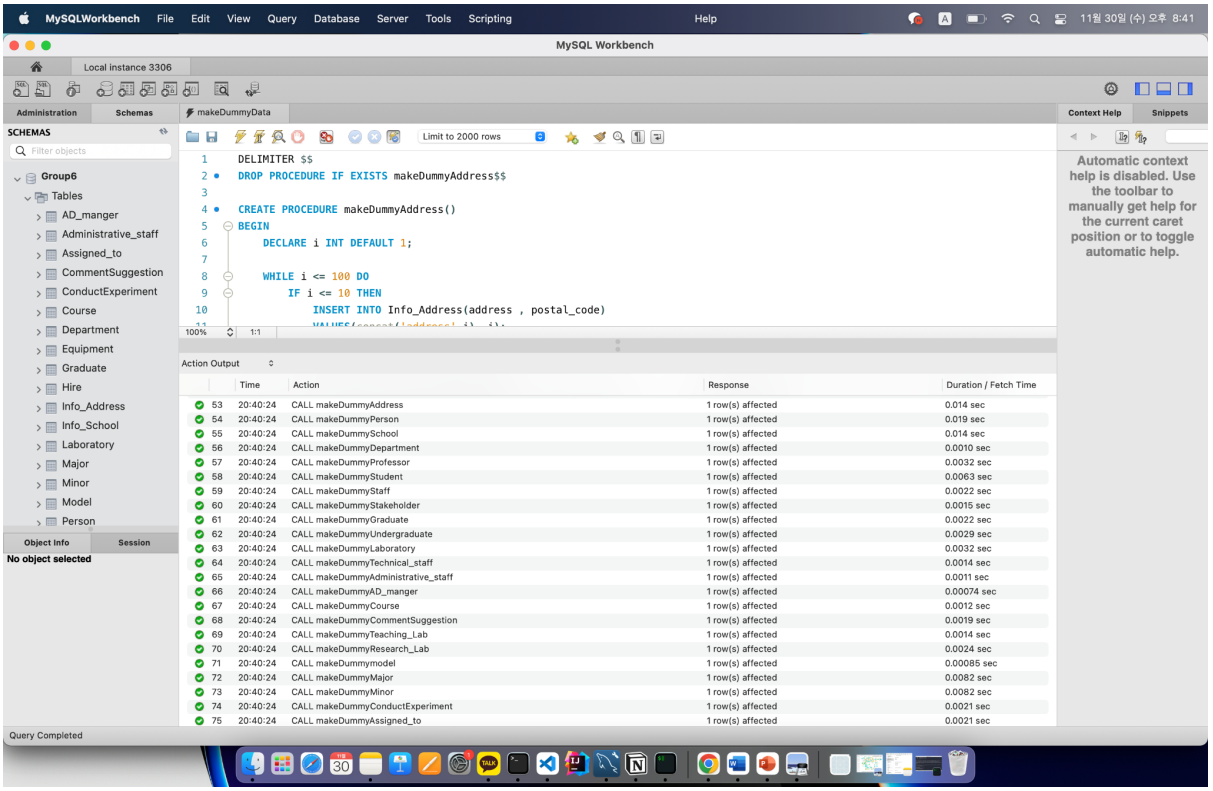
This is a trigger that allows only integers beginning with 1 as person_id.

7.onlystart5stakeholder_id

This is a trigger that allows only integers beginning with 1 as person_id.

I will submit the trigger creation query separately. (Triggers.sql)

Third, Dummy data.



We stored dummy data to verify that the triggers, sample queries, and additional queries we created worked well.

Dummy data was stored in a total of 26 tables, and data was generated using PROCEDURE.

I will also submit a query that generates Dummy data separately. (makeDummyData.sql)

Fourth, sample & additional queries

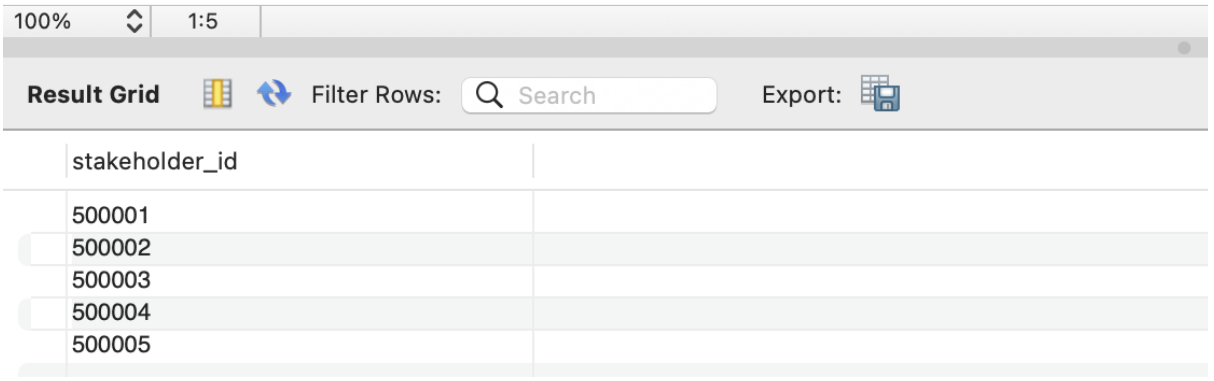
I will attach only the results of the query to the report, and submit the contents of the query as a separate file.

(additionalQuery.sql and sampleQuery.sql)

1) Sample queries

a) Identify all Stakeholders who have provided at least two comments or suggestions.

```sql
SELECT stakeholder_id
FROM CommentSuggestion
GROUP BY stakeholder_id
HAVING COUNT(*) >= 2;
```



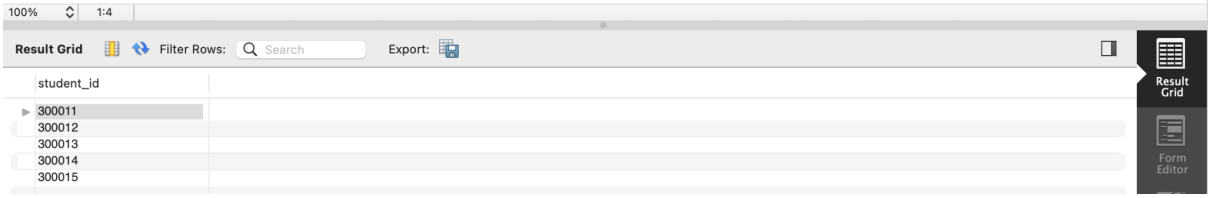
100% 1:5

Result Grid Filter Rows: Search Export:

stakeholder_id
500001
500002
500003
500004
500005

b) Identify Undergraduate students who take over 15 credits and conduct experiments in three different labs.

```sql
select Y.student_id
from (select student_id from Take_course as T, Course as C where T.course_id = C.course_id Group by student_id Having SUM(credit) >
where X.student_id = Y.student_id;
```



100% 1:4

Result Grid Filter Rows: Search Export:

student_id
300011
300012
300013
300014
300015

Result Grid

Form Editor

c) Find all the graduate students who take research labs of the CS department.

```
1 •  select student_id
2    from Assigned_to as X
3    where X.department_name='CS';
4
```

100% 30:3

Result Grid | Filter Rows: Search | Export:

student_id
300001
300002
300003
300004
300005
300006
300007
300008
300009
300010

d) List all the Equipment belonging to a particular Lab. (Change lab_Name and department_name as needed.)

```
1 •  select lab_Name , department_name , equipment_id, model_no
2    from Equipment AS E
3    where lab_Name = 'lab1' AND department_name = 'BIO';
4
```

100% 51:3

Result Grid | Filter Rows: Search | Edit: | Export/Import:

lab_Name	department_na...	equipment_id	model_no
lab1	BIO	800001	model-no-1
lab1	BIO	800002	model-no-2
lab1	BIO	800003	model-no-3

e) Find all Professors who address at least 5 different research topics.

```sql
select professor_id
from Graduate as G
Group by professor_id
Having Count(DISTINCT research_Topic )>=5;
```

100% 1:5

Result Grid Filter Rows: Search Export:

professor_id
200001

2) Additional queries

a). Find all the students who have taken at least 4 courses and attended at least 3 different labs.

```sql
SELECT student_id
FROM Student
WHERE student_id IN (
    SELECT student_id
    FROM Take_course
    GROUP BY student_id
    HAVING COUNT(*) >= 4
) AND student_id IN (
    SELECT student_id
    FROM ConductExperiment
    GROUP BY student_id
    HAVING COUNT(*) >= 3
);
```

100% 3:13

Result Grid Filter Rows: Search Edit: Export/Import:

student_id
300011
300012
300013
300014
300015
NULL

b). Find the graduate students who is assigned to “lab1” lab in “CS” but did not take “course3” course.

```sql
SELECT student_id
FROM Graduate
WHERE student_id IN (
    SELECT student_id
    FROM Assigned_to
    WHERE lab_name = 'lab1' AND department_name = 'CS'
) AND student_id NOT IN (
    SELECT student_id
    FROM Take_course
    WHERE course_id IN (
        SELECT course_id
        FROM Course as C
        WHERE C.name = 'course3'
    )
);
```

100% 25:11

Result Grid | Filter Rows: Search | Edit: | Export/Import:

student_id
300001
300002
300003
300004
300005

c). Find the Phone number of graduate students belonging to the rich lab. (Rich Lab: Labs with 5 or more pieces of equipment)

```sql
SELECT phone
FROM Person
WHERE person_id IN (
    SELECT person_id
    From Student
    WHERE student_id IN (
        SELECT student_id
        From Assigned_to AS X
        WHERE student_id IN (
            SELECT student_id
            FROM (SELECT lab_Name , department_name FROM Equipment GROUP BY lab_Name, department_name HAVING COUNT(*) >= 5) AS Y
            WHERE X.lab_name=Y.lab_name AND X.department_name = Y.department_name
        )
    )
);
```

100% 3:15

Result Grid | Filter Rows: Search | Export:

phone
010-1234-1026
010-1234-1027
010-1234-1028
010-1234-1029
010-1234-1030

Query (It is included in the ‘lab5_Group6.zip’.)

1.createTable.sql

```
CREATE TABLE Info_Address(
   address     VARCHAR(100),
   postal_code INT NOT NULL,
   PRIMARY KEY(address)
);
CREATE TABLE Person(
   person_id   BIGINT UNSIGNED,
   name        VARCHAR(50) NOT NULL,
   address     VARCHAR(100),
   nation      VARCHAR(30) NOT NULL,
   email       VARCHAR(50) NOT NULL,
   phone       VARCHAR(30) ,
   PRIMARY KEY(person_id),
   FOREIGN KEY(address) REFERENCES Info_Address(address) ON UPDATE CASCADE ON
DELETE SET NULL
);
CREATE TABLE Info_School(
   person_id   BIGINT UNSIGNED,
   school_name VARCHAR(50) NOT NULL,
   PRIMARY KEY(person_id,school_name),
   FOREIGN KEY(person_id) REFERENCES Person(person_id) ON UPDATE CASCADE ON DELETE
CASCADE
);
CREATE TABLE Department(
   department_name VARCHAR(30),
   department_number   INT NOT NULL,
   PRIMARY KEY(department_name)
);
CREATE TABLE Professor(
   professor_id    BIGINT UNSIGNED,
   person_id       BIGINT UNSIGNED NOT NULL,
   department_name VARCHAR(30),
   major           VARCHAR(30) NOT NULL,
   PRIMARY KEY(professor_id),
   FOREIGN KEY(person_id) REFERENCES Person(person_id) ON UPDATE CASCADE ON DELETE
CASCADE,
   FOREIGN KEY(department_name) REFERENCES Department(department_name) ON UPDATE
CASCADE ON DELETE SET NULL
);
CREATE TABLE Student(
   student_id    BIGINT UNSIGNED,
   person_id     BIGINT UNSIGNED NOT NULL,
```

```
   year          INT NOT NULL,
   PRIMARY KEY(student_id),
   FOREIGN KEY(person_id) REFERENCES Person(person_id) ON UPDATE CASCADE ON DELETE
CASCADE
);
CREATE TABLE Staff(
   staff_id    BIGINT UNSIGNED,
   person_id   BIGINT UNSIGNED NOT NULL,
   date_hired  DATE NOT NULL,
   position    VARCHAR(30) NOT NULL,
   PRIMARY KEY(staff_id),
   FOREIGN KEY(person_id) REFERENCES Person(person_id) ON UPDATE CASCADE ON DELETE
CASCADE
);
CREATE TABLE Stakeholder(
   stakeholder_id    BIGINT UNSIGNED,
   person_id   BIGINT UNSIGNED NOT NULL,
   domain    VARCHAR(30) NOT NULL,
   PRIMARY KEY(stakeholder_id),
   FOREIGN KEY(person_id) REFERENCES Person(person_id) ON UPDATE CASCADE ON DELETE
CASCADE
);
CREATE TABLE Graduate(
   student_id      BIGINT UNSIGNED,
   research_Topic  VARCHAR(100) NOT NULL,
   professor_id    BIGINT UNSIGNED NOT NULL,
   PRIMARY KEY(student_id),
   FOREIGN KEY(professor_id) REFERENCES Professor(professor_id) ON UPDATE CASCADE
ON DELETE RESTRICT,
   FOREIGN KEY(student_id) REFERENCES Student(student_id) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE Undergraduate(
   student_id  BIGINT UNSIGNED,
   PRIMARY KEY(student_id),
   FOREIGN KEY(student_id) REFERENCES Student(student_id) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE Laboratory(
   lab_name    VARCHAR(30),
   department_name VARCHAR(30),
   capacity    INT NOT NULL,
   location    VARCHAR(100) NOT NULL,
   PRIMARY KEY(lab_name,department_name),
```

```
   FOREIGN KEY(department_name) REFERENCES Department(department_name) ON UPDATE
CASCADE ON DELETE RESTRICT
);
CREATE TABLE Technical_staff(
   staff_id    BIGINT UNSIGNED,
   lab_Name    VARCHAR(30) NOT NULL,
   department_name VARCHAR(30) NOT NULL,
   PRIMARY KEY(staff_id),
   FOREIGN KEY(staff_id) REFERENCES Staff(staff_id) ON UPDATE CASCADE ON DELETE
CASCADE,
   FOREIGN KEY(lab_name, department_name) REFERENCES
Laboratory(lab_name,department_name) ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE Administrative_staff(
   staff_id    BIGINT UNSIGNED,
   PRIMARY KEY(staff_id),
   FOREIGN KEY(staff_id) REFERENCES Staff(staff_id) ON UPDATE CASCADE ON DELETE
CASCADE
);
CREATE TABLE AD_manger(
   staff_id BIGINT UNSIGNED,
   PRIMARY KEY(staff_id),
   FOREIGN KEY(staff_id) REFERENCES Administrative_staff(staff_id) ON UPDATE
CASCADE ON DELETE CASCADE
);
CREATE TABLE Course(
   course_id   BIGINT UNSIGNED,
   name       VARCHAR(30) NOT NULL,
   credit      INT NOT NULL,
   date_time   TIME NOT NULL,
   professor_id   BIGINT UNSIGNED NOT NULL,
   PRIMARY KEY(course_id),
   FOREIGN KEY(professor_id) REFERENCES Professor(professor_id) ON UPDATE CASCADE
ON DELETE RESTRICT
);
CREATE TABLE CommentSuggestion(
   stakeholder_id BIGINT UNSIGNED,
   date_time      TIMESTAMP,
   topic          VARCHAR(100),
   PRIMARY KEY(stakeholder_id, date_Time, topic),
   FOREIGN KEY(stakeholder_id) REFERENCES Stakeholder(stakeholder_id) ON UPDATE
CASCADE ON DELETE CASCADE
);
CREATE TABLE Teaching_Lab(
   lab_name    VARCHAR(30),
```

```
   department_name VARCHAR(30),
   PRIMARY KEY(lab_name,department_name),
   FOREIGN KEY(lab_name,department_name) REFERENCES
Laboratory(lab_name,department_name) ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE Research_Lab(
   lab_name    VARCHAR(30),
   department_name VARCHAR(30),
   PRIMARY KEY(Lab_Name,Department_Name),
   FOREIGN KEY(lab_name,department_name) REFERENCES
Laboratory(lab_name,department_name) ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE Model(
   model_no    VARCHAR(30),
   model_name  VARCHAR(30) NOT NULL,
   PRIMARY KEY(model_no)
);
CREATE TABLE Equipment(
   lab_Name    VARCHAR(30),
   department_name VARCHAR(30),
   equipment_id    BIGINT UNSIGNED,
   model_no    VARCHAR(30) NOT NULL,
   date_purchased  DATE NOT NULL,
   PRIMARY KEY(lab_name,department_name,equipment_id),
   FOREIGN KEY(lab_name,department_name) REFERENCES
Laboratory(lab_name,department_name) ON UPDATE CASCADE ON DELETE CASCADE,
   FOREIGN KEY(model_no) REFERENCES Model(model_no) ON UPDATE CASCADE ON DELETE
CASCADE
);
CREATE TABLE Major(
   student_id  BIGINT UNSIGNED,
   major_name  VARCHAR(30),
   PRIMARY KEY(student_id, major_name),
   FOREIGN KEY(student_id) REFERENCES Student(student_id) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE Minor(
   student_id  BIGINT UNSIGNED,
   minor_name  VARCHAR(30),
   PRIMARY KEY(student_id, minor_name),
   FOREIGN KEY(student_id) REFERENCES Student(student_id) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE ConductExperiment(
   student_id  BIGINT UNSIGNED,
```

```
    lab_name    VARCHAR(30),
    department_name VARCHAR(30),
    attendance_date DATE NOT NULL,
    PRIMARY KEY(student_id,lab_name,department_name),
    FOREIGN KEY(student_id) REFERENCES Undergraduate(student_id) ON UPDATE CASCADE
ON DELETE CASCADE,
    FOREIGN KEY(lab_name,department_name) REFERENCES
Teaching_Lab(lab_name,department_name) ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE Assigned_to(
    student_id  BIGINT UNSIGNED,
    lab_name    VARCHAR(30),
    department_name VARCHAR(30),
    PRIMARY KEY(student_id,lab_name,department_name),
    FOREIGN KEY(student_id) REFERENCES Graduate(student_id) ON UPDATE CASCADE ON
DELETE CASCADE,
    FOREIGN KEY(lab_name,department_name) REFERENCES
Research_Lab(lab_name,department_name) ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE Hire(
    professor_id    BIGINT UNSIGNED,
    ad_manger_id    BIGINT UNSIGNED,
    PRIMARY KEY(professor_id,ad_manger_id),
    FOREIGN KEY(professor_id) REFERENCES Professor(professor_id) ON UPDATE CASCADE
ON DELETE CASCADE,
    FOREIGN KEY(ad_manger_id) REFERENCES AD_manger(staff_id)ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE Take_course(
    student_id  BIGINT UNSIGNED,
    course_id   BIGINT UNSIGNED,
    PRIMARY KEY(student_id,course_id),
    FOREIGN KEY(student_id) REFERENCES Student(student_id) ON UPDATE CASCADE ON
DELETE CASCADE,
    FOREIGN KEY(course_id) REFERENCES Course(course_id) ON UPDATE CASCADE ON DELETE
CASCADE
);
```

2. Triggers.sql

```
DELIMITER $$
CREATE TRIGGER changeTeachingProf
BEFORE DELETE
ON Professor
FOR EACH ROW
```

```
BEGIN
   DECLARE delete_id BIGINT UNSIGNED;
   SET delete_id = OLD.professor_id;

   UPDATE Course
   SET professor_id = (SELECT P.professor_id FROM Professor AS P WHERE
P.professor_id <> delete_id Limit 1)
   where professor_id=delete_id;
END $$

DELIMITER ;

DELIMITER $$
CREATE TRIGGER changeSuperviseProf
BEFORE DELETE
ON Professor
FOR EACH ROW

BEGIN
   DECLARE delete_id BIGINT UNSIGNED;
   SET delete_id = OLD.professor_id;

   UPDATE Graduate
   SET professor_id = (SELECT P.professor_id FROM Professor AS P WHERE
P.professor_id <> delete_id Limit 1)
   where professor_id=delete_id;
END $$

DELIMITER ;


DELIMITER $$
CREATE TRIGGER onlystart1Person_id
AFTER INSERT
ON Person
FOR EACH ROW

BEGIN
   DECLARE new_person_id BIGINT UNSIGNED;
   DECLARE new_person_id_to_str VARCHAR(30);
   SET new_person_id = NEW.person_id;
   SET new_person_id_to_str = concat(new_person_id);

   IF new_person_id_to_str NOT LIKE '1%' THEN
       DELETE FROM Person WHERE person_id=new_person_id;
```

```
   END IF;
END $$

DELIMITER ;



DELIMITER $$
CREATE TRIGGER onlystart3student_id
AFTER INSERT
ON Student
FOR EACH ROW

BEGIN
   DECLARE new_id BIGINT UNSIGNED;
   DECLARE new_id_to_str VARCHAR(30);
   SET new_id = NEW.student_id;
   SET new_id_to_str = concat(new_id);

   IF new_id_to_str NOT LIKE '3%' THEN
       DELETE FROM Student WHERE student_id=new_id;
   END IF;
END $$

DELIMITER ;



DELIMITER $$
CREATE TRIGGER onlystart2prof_id
AFTER INSERT
ON Professor
FOR EACH ROW

BEGIN
   DECLARE new_id BIGINT UNSIGNED;
   DECLARE new_id_to_str VARCHAR(30);
   SET new_id = NEW.professor_id;
   SET new_id_to_str = concat(new_id);

   IF new_id_to_str NOT LIKE '2%' THEN
       DELETE FROM Professor WHERE professor_id=new_id;
   END IF;
END $$

DELIMITER ;
```

```
DELIMITER $$
CREATE TRIGGER onlystart4staff_id
AFTER INSERT
ON Staff
FOR EACH ROW

BEGIN
   DECLARE new_id BIGINT UNSIGNED;
   DECLARE new_id_to_str VARCHAR(30);
   SET new_id = NEW.staff_id;
   SET new_id_to_str = concat(new_id);

   IF new_id_to_str NOT LIKE '4%' THEN
       DELETE FROM Staff WHERE staff_id=new_id;
   END IF;
END $$

DELIMITER ;


DELIMITER $$
CREATE TRIGGER onlystart5stakeholder_id
AFTER INSERT
ON Stakeholder
FOR EACH ROW

BEGIN
   DECLARE new_id BIGINT UNSIGNED;
   DECLARE new_id_to_str VARCHAR(30);
   SET new_id = NEW.stakeholder_id;
   SET new_id_to_str = concat(new_id);

   IF new_id_to_str NOT LIKE '5%' THEN
       DELETE FROM Stakeholder WHERE stakeholder_id=new_id;
   END IF;
END $$

DELIMITER ;
```

3.makeDummyData.sql

```
DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyAddress$$
CREATE PROCEDURE makeDummyAddress()
BEGIN
   DECLARE i INT DEFAULT 1;
```

```
WHILE i <= 100 DO
    IF i <= 10 THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i);

    ELSEIF (i <=20 AND i>=11) THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i);

    ELSEIF (i <=30 AND i>=21) THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i-20);

    ELSEIF (i <=40 AND i>=31) THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i-20);

    ELSEIF (i <=50 AND i>=41) THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i-40);

    ELSEIF (i <=60 AND i>=51) THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i-40);

    ELSEIF (i <=70 AND i>=61) THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i-60);

    ELSEIF (i <=80 AND i>=71) THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i-60);

    ELSEIF (i <=90 AND i>=81) THEN
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i-80);

    ELSE
        INSERT INTO Info_Address(address , postal_code)
        VALUES(concat('address',i), i-80);

    END IF;
        SET i = i + 1;
END WHILE;
```

```
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyPerson$$
CREATE PROCEDURE makeDummyPerson()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 100 DO
       IF i <= 10 THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i),concat('address',i),'KOREA',concat('email',i),conc
at('010-','1234-',1000+i));

       ELSEIF (i <=20 AND i>=11) THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i),concat('address',i),'KOREA',concat('email',i),conc
at('010-','1234-',1000+i));

       ELSEIF (i <=30 AND i>=21) THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i-20),concat('address',i),'KOREA',concat('email',i),c
oncat('010-','1234-',1000+i));


       ELSEIF (i <=40 AND i>=31) THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i),concat('address',i-10),'USA',concat('email',i),con
cat('010-','1234-',1000+i));


       ELSEIF (i <=50 AND i>=41) THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i),concat('address',i),'USA',concat('email',i),concat
('010-','1234-',1000+i));

       ELSEIF (i <=60 AND i>=51) THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)
```

```
VALUES(i+100000,concat('name',i),concat('address',i),'USA',concat('email',i),concat
('010-','1234-',1000+i));


       ELSEIF (i <=70 AND i>=61) THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i),concat('address',i),'GANA',concat('email',i),conca
t('010-','1234-',1000+i));


       ELSEIF (i <=80 AND i>=71) THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i),concat('address',i),'GANA',concat('email',i),conca
t('010-','1234-',1000+i));

       ELSEIF (i <=90 AND i>=81) THEN
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i),concat('address',i),'GANA',concat('email',i),conca
t('010-','1234-',1000+i));

       ELSE
           INSERT INTO Person(person_id , name, address, nation, email, phone)

VALUES(i+100000,concat('name',i),concat('address',i),'JAPAN',concat('email',i),conc
at('010-','1234-',1000+i));

       END IF;
           SET i = i + 1;
   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummySchool$$
CREATE PROCEDURE makeDummySchool()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 140 DO
       IF i <= 30  THEN
           INSERT INTO Info_School(person_id , school_name)
```

```
            VALUES(100000+i,'school_1');

        ELSEIF (i <=60 AND i>=31) THEN
            INSERT INTO Info_School(person_id , school_name)
            VALUES(100000+i-30,'school_2');


        ELSEIF (i <=90 AND i>=61) THEN
            INSERT INTO Info_School(person_id , school_name)
            VALUES(100000+i-30,'school_3');


        ELSEIF (i <=100 AND i>=91) THEN
            INSERT INTO Info_School(person_id , school_name)
            VALUES(100000+i-30,'school_3');

        ELSEIF (i <=110 AND i>=101) THEN
            INSERT INTO Info_School(person_id , school_name)
            VALUES(100000+i-40,'school_1');

        ELSE
            INSERT INTO Info_School(person_id , school_name)
            VALUES(100000+i-40,'school_4');

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyDepartment$$
CREATE PROCEDURE makeDummyDepartment()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 8 DO
        IF i =1  THEN
            INSERT INTO Department(department_name , department_number)
            VALUES('CS',101);

        ELSEIF (i=2) THEN
            INSERT INTO Department(department_name , department_number)
            VALUES('EE',102);
```

```
       ELSEIF (i=3) THEN
           INSERT INTO Department(department_name , department_number)
           VALUES('IE',103);

       ELSEIF (i=4) THEN
           INSERT INTO Department(department_name , department_number)
           VALUES('ME',104);

       ELSEIF (i=5) THEN
           INSERT INTO Department(department_name , department_number)
           VALUES('BIO',105);

       ELSEIF (i=6) THEN
           INSERT INTO Department(department_name , department_number)
           VALUES('ECHE',106);

       ELSEIF (i=7) THEN
           INSERT INTO Department(department_name , department_number)
           VALUES('BME',107);

       ELSE
           INSERT INTO Department(department_name , department_number)
           VALUES('MTH',108);

       END IF;
           SET i = i + 1;
   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyProfessor$$
CREATE PROCEDURE makeDummyProfessor()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 20 DO
       IF i <= 3  THEN
           INSERT INTO Professor(professor_id , person_id, department_name, major)
           VALUES(200000+i,100000+i,'CS','Computer science');

       ELSEIF (i=4) THEN
           INSERT INTO Professor(professor_id , person_id, department_name, major)
           VALUES(200000+i,100000+i,'CS','Mathematics');
```

```
      ELSEIF (i=5) THEN
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'CS','Electronic engineering');

      ELSEIF (i <=7 AND i>=6) THEN
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'EE','Electronic engineering');

      ELSEIF (i=8) THEN
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'EE','Computer science');


      ELSEIF (i <=10 AND i>=9) THEN
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'IE','Industrial engineering');


      ELSEIF (i=11) THEN
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'ME','Mechanical engineering');

      ELSEIF (i <=13 AND i>=12) THEN
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'BIO','Biotechnology');

      ELSEIF (i <=16 AND i>=14) THEN
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'ECHE','Chemical engineering');

      ELSEIF (i <=18 AND i>=17) THEN
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'BME','Biotechnology');

      ELSE
          INSERT INTO Professor(professor_id , person_id, department_name, major)
          VALUES(200000+i,100000+i,'MTH','Mathematics');

      END IF;
          SET i = i + 1;
   END WHILE;
END$$
DELIMITER ;
```

```
DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyStudent$$
CREATE PROCEDURE makeDummyStudent()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 50 DO
       IF i <= 20  THEN
           INSERT INTO Student(student_id , person_id, year)
           VALUES(300000+i,100020+i,1);

       ELSEIF (i <=35 AND i>=21) THEN
           INSERT INTO Student(student_id , person_id, year)
           VALUES(300000+i,100020+i,2);

       ELSEIF (i <=45 AND i>=36) THEN
           INSERT INTO Student(student_id , person_id, year)
           VALUES(300000+i,100020+i,3);

       ELSE
           INSERT INTO Student(student_id , person_id, year)
           VALUES(300000+i,100020+i,4);

       END IF;
           SET i = i + 1;
   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyStaff$$
CREATE PROCEDURE makeDummyStaff()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 20 DO
       IF i <= 5  THEN
           INSERT INTO Staff(staff_id , person_id, date_hired, position)
           VALUES(400000+i,100070+i,'2022-10-01','position1');

       ELSEIF (i <=10 AND i>=6) THEN
           INSERT INTO Staff(staff_id , person_id, date_hired, position)
           VALUES(400000+i,100070+i,'2020-05-01','position1');

       ELSEIF (i <=15 AND i>=11) THEN
```

```
            INSERT INTO Staff(staff_id , person_id, date_hired, position)
            VALUES(400000+i,100070+i,'2020-10-01','position2');

        ELSE
            INSERT INTO Staff(staff_id , person_id, date_hired, position)
            VALUES(400000+i,100070+i,'2021-07-01','position3');

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyStakeholder$$
CREATE PROCEDURE makeDummyStakeholder()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 10 DO
        IF i <= 5  THEN
            INSERT INTO Stakeholder(stakeholder_id , person_id, domain)
            VALUES(500000+i,100090+i,'domain1');

        ELSEIF (i <=8 AND i>=6) THEN
            INSERT INTO Stakeholder(stakeholder_id , person_id, domain)
            VALUES(500000+i,100090+i,'domain2');

        ELSE
            INSERT INTO Stakeholder(stakeholder_id , person_id, domain)
            VALUES(500000+i,100090+i,'domain3');

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyGraduate$$
CREATE PROCEDURE makeDummyGraduate()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 20 DO
```

```
        IF i <= 3  THEN
            INSERT INTO Graduate(student_id , research_Topic, professor_id)
            VALUES(300000+i,'research_topic1',200001);

        ELSEIF (i <=5 AND i>=4) THEN
            INSERT INTO Graduate(student_id , research_Topic, professor_id)
            VALUES(300000+i,'research_topic2',200002);

        ELSEIF (i <=10 AND i>=6) THEN
            INSERT INTO Graduate(student_id , research_Topic, professor_id)
            VALUES(300000+i,'research_topic3',200003);

        ELSEIF (i <=12 AND i>=11) THEN
            INSERT INTO Graduate(student_id , research_Topic, professor_id)
            VALUES(300010+i,'research_topic4',200001);

        ELSEIF (i <=15 AND i>=13) THEN
            INSERT INTO Graduate(student_id , research_Topic, professor_id)
            VALUES(300010+i,'research_topic5',200001);

        ELSEIF (i <=19 AND i>=16) THEN
            INSERT INTO Graduate(student_id , research_Topic, professor_id)
            VALUES(300020+i,'research_topic6',200001);

        ELSE
            INSERT INTO Graduate(student_id , research_Topic, professor_id)
            VALUES(300046,'research_topic7',200001);

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyUndergraduate$$
CREATE PROCEDURE makeDummyUndergraduate()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 30 DO
        IF i <= 10  THEN
            INSERT INTO Undergraduate(student_id)
            VALUES(300010+i);
```

```
        ELSEIF (i <=20 AND i>=11) THEN
            INSERT INTO Undergraduate(student_id)
            VALUES(300015+i);

        ELSEIF (i <=26 AND i>=21) THEN
            INSERT INTO Undergraduate(student_id)
            VALUES(300019+i);

        ELSE
            INSERT INTO Undergraduate(student_id)
            VALUES(300020+i);

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyLaboratory$$
CREATE PROCEDURE makeDummyLaboratory()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 30 DO
        IF i <= 5  THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i),'CS',10+i*2,concat('location',i));

        ELSEIF (i <=8 AND i>=6) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-5),'EE',10+(i-5)*2,concat('location',i));


        ELSEIF (i <=10 AND i>=9) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-8),'IE',10+(i-8)*2,concat('location',i));


        ELSEIF (i <=11 AND i>=11) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-10),'ME',10+(i-10)*2,concat('location',i));

        ELSEIF (i <=13 AND i>=12) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
```

```
            VALUES(concat('lab',i-11),'BIO',10+(i-11)*2,concat('location',i));

        ELSEIF (i <=16 AND i>=14) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-13),'ECHE',10+(i-13)*2,concat('location',i));

        ELSEIF (i <=18 AND i>=17) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-16),'BME',10+(i-16)*2,concat('location',i));

        ELSEIF (i <=20 AND i>=19) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-18),'MTH',10+(i-18)*2,concat('location',i));

        ELSEIF (i <=23 AND i>=21) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-10),'BIO',10+(i-20)*2,concat('location',i));

        ELSEIF (i <=27 AND i>=24) THEN
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-13),'ECHE',10+(i-23)*2,concat('location',i));

        ELSE
            INSERT INTO Laboratory(lab_name, department_name, capacity, location)
            VALUES(concat('lab',i-17),'BME',10+(i-23)*2,concat('location',i));

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyTechnical_staff$$
CREATE PROCEDURE makeDummyTechnical_staff()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 10 DO
        IF i = 1  THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','CS');

        ELSEIF i=2 THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
```

```
            VALUES(400000+i,'lab1','EE');

        ELSEIF i=3 THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','IE');

        ELSEIF i=4 THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','ME');

        ELSEIF i=5 THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','BIO');

        ELSEIF i=6 THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','ECHE');

        ELSEIF i=7 THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','BME');

        ELSEIF i=8 THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','MTH');

        ELSEIF i=9 THEN
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','CS');

        ELSE
            INSERT INTO Technical_staff(staff_id, lab_Name, department_name)
            VALUES(400000+i,'lab1','CS');

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyAdministrative_staff$$
CREATE PROCEDURE makeDummyAdministrative_staff()
BEGIN
    DECLARE i INT DEFAULT 1;
```

```
   WHILE i <= 10 DO

       INSERT INTO Administrative_staff(staff_id)
       VALUES(400010+i);

       SET i = i + 1;
   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyAD_manger$$
CREATE PROCEDURE makeDummyAD_manger()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 5 DO

       INSERT INTO AD_manger(staff_id)
       VALUES(400010+i);

       SET i = i + 1;
   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyCourse$$
CREATE PROCEDURE makeDummyCourse()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 10 DO
       IF i <= 3  THEN
           INSERT INTO Course(course_id, name, credit, date_time,professor_ID)
           VALUES(700000+i,concat('course',i),1,'09:00:00',200004+i);

       ELSEIF (i <=6 AND i>=4) THEN
           INSERT INTO Course(course_id, name, credit, date_time,professor_ID)
           VALUES(700000+i,concat('course',i-3),5,'10:30:00',200003+i);

       ELSEIF (i <=9 AND i>=7) THEN
           INSERT INTO Course(course_id, name, credit, date_time,professor_ID)
           VALUES(700000+i,concat('course',i-6),5,'13:00:00',200010+i);
```

```
        ELSE
            INSERT INTO Course(course_id, name, credit, date_time,professor_ID)
            VALUES(700000+i,concat('course',i-9),5,'16:00:00',200005+i);

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyCommentSuggestion$$
CREATE PROCEDURE makeDummyCommentSuggestion()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 20 DO
        IF i <= 10  THEN
            INSERT INTO CommentSuggestion(stakeholder_id,date_time,topic)
            VALUES(500000+i,'2022-01-01 12:00:00',concat('topic',i));

        ELSEIF (i<=15 AND i>=11) THEN
            INSERT INTO CommentSuggestion(stakeholder_id,date_time,topic)
            VALUES(500000+i-10,'2022-01-01 13:00:00',concat('topic',i));

        ELSE
            INSERT INTO CommentSuggestion(stakeholder_id,date_time,topic)
            VALUES(500000+i-15,'2022-01-01 14:00:00',concat('topic',i));

        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyTeaching_Lab$$
CREATE PROCEDURE makeDummyTeaching_Lab()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 10 DO
        IF i = 1  THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
```

```
            VALUES(concat('lab',i+10),'BIO');

        ELSEIF i = 2 THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+9),'BME');

        ELSEIF i = 3 THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+8),'ECHE');

        ELSEIF i = 4 THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+8),'BIO');

        ELSEIF i = 5 THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+7),'BME');

        ELSEIF i = 6 THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+6),'ECHE');

        ELSEIF i = 7 THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+6),'BIO');

        ELSEIF i = 8 THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+5),'BME');

        ELSEIF i = 9 THEN
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+4),'ECHE');

        ELSE
            INSERT INTO Teaching_Lab(lab_name,department_name)
            VALUES(concat('lab',i+4),'ECHE');


        END IF;
            SET i = i + 1;
    END WHILE;
END$$
DELIMITER ;
```

```
DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyResearch_Lab$$
CREATE PROCEDURE makeDummyResearch_Lab()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 20 DO
       IF i <=5  THEN
           INSERT INTO Research_Lab(lab_name,department_name)
           VALUES(concat('lab',i),'CS');

       ELSEIF (i<=8 AND i>=6) THEN
           INSERT INTO Research_Lab(lab_name,department_name)
           VALUES(concat('lab',i-5),'EE');

       ELSEIF (i<=10 AND i>=9) THEN
           INSERT INTO Research_Lab(lab_name,department_name)
           VALUES(concat('lab',i-8),'IE');

       ELSEIF (i<=11 AND i>=11) THEN
           INSERT INTO Research_Lab(lab_name,department_name)
           VALUES(concat('lab',i-10),'ME');

       ELSEIF (i<=13 AND i>=12) THEN
           INSERT INTO Research_Lab(lab_name,department_name)
           VALUES(concat('lab',i-11),'BIO');

       ELSEIF (i<=16 AND i>=14) THEN
           INSERT INTO Research_Lab(lab_name,department_name)
           VALUES(concat('lab',i-13),'ECHE');

       ELSEIF (i<=18 AND i>=17) THEN
           INSERT INTO Research_Lab(lab_name,department_name)
           VALUES(concat('lab',i-16),'BME');

       ELSE
           INSERT INTO Research_Lab(lab_name,department_name)
           VALUES(concat('lab',i-18),'MTH');


       END IF;
           SET i = i + 1;
   END WHILE;
END$$
DELIMITER ;
```

```
DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummymodel$$
CREATE PROCEDURE makeDummymodel()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 5 DO

       INSERT INTO Model(model_no, model_name)
       VALUES(concat('model-no-',i),concat('model-name-',i));

       SET i = i + 1;
   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyMajor$$
CREATE PROCEDURE makeDummyMajor()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 90 DO
       IF i<= 20 THEN
           INSERT INTO Major(student_id,major_name)
           VALUES(300000+i,concat('major',i));

       ELSEIF (i<=40 AND i>=21) THEN
           INSERT INTO Major(student_id,major_name)
           VALUES(300000+i-20,concat('major',i));

       ELSEIF (i<=60 AND i>=41) THEN
           INSERT INTO Major(student_id,major_name)
           VALUES(300000+i-20,concat('major',i-40));

       ELSEIF (i<=80 AND i>=61) THEN
           INSERT INTO Major(student_id,major_name)
           VALUES(300000+i-40,concat('major',i-40));

       ELSE
           INSERT INTO Major(student_id,major_name)
           VALUES(300000+i-40,concat('major',i-80));

       END IF;
```

```
            SET i = i + 1;

    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyMinor$$
CREATE PROCEDURE makeDummyMinor()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 90 DO
        IF i<= 20 THEN
            INSERT INTO Minor(student_id,minor_name)
            VALUES(300000+i,concat('minor',i));

        ELSEIF (i<=40 AND i>=21) THEN
            INSERT INTO Minor(student_id,minor_name)
            VALUES(300000+i-20,concat('minor',i));

        ELSEIF (i<=60 AND i>=41) THEN
            INSERT INTO Minor(student_id,minor_name)
            VALUES(300000+i-20,concat('minor',i-40));

        ELSEIF (i<=80 AND i>=61) THEN
            INSERT INTO Minor(student_id,minor_name)
            VALUES(300000+i-40,concat('minor',i-40));

        ELSE
            INSERT INTO Minor(student_id,minor_name)
            VALUES(300000+i-40,concat('minor',i-80));

        END IF;
            SET i = i + 1;

    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyConductExperiment$$
CREATE PROCEDURE makeDummyConductExperiment()
BEGIN
    DECLARE i INT DEFAULT 1;
```

```
   WHILE i <= 20 DO
       IF i<= 5 THEN
           INSERT INTO ConductExperiment(student_id,lab_name, department_name,
attendance_date)
           VALUES(300010+i,'lab11','BIO','2022-11-01');

       ELSEIF (i<=10 AND i>=6) THEN
           INSERT INTO ConductExperiment(student_id,lab_name, department_name,
attendance_date)
           VALUES(300005+i,'lab12','ECHE','2022-11-02');

       ELSEIF (i<=15 AND i>=11) THEN
           INSERT INTO ConductExperiment(student_id,lab_name, department_name,
attendance_date)
           VALUES(300000+i,'lab13','BME','2022-11-03');

       ELSE
           INSERT INTO ConductExperiment(student_id,lab_name, department_name,
attendance_date)
           VALUES(300025+i,'lab14','ECHE','2022-11-04');

       END IF;
           SET i = i + 1;

   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyAssigned_to$$
CREATE PROCEDURE makeDummyAssigned_to()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 20 DO
       IF i<= 5 THEN
           INSERT INTO Assigned_to(student_id,lab_name, department_name)
           VALUES(300000+i,'lab1','CS');

       ELSEIF (i<=10 AND i>=6) THEN
           INSERT INTO Assigned_to(student_id,lab_name, department_name)
           VALUES(300000+i,'lab2','CS');
```

```
       ELSEIF (i<=15 AND i>=11) THEN
           INSERT INTO Assigned_to(student_id,lab_name, department_name)
           VALUES(300010+i,'lab3','ECHE');

       ELSEIF (i<=19 AND i>=16) THEN
           INSERT INTO Assigned_to(student_id,lab_name, department_name)
           VALUES(300020+i,'lab2','EE');

       ELSE
           INSERT INTO Assigned_to(student_id,lab_name, department_name)
           VALUES(300046,'lab1','ME');


       END IF;
           SET i = i + 1;

   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyHire$$
CREATE PROCEDURE makeDummyHire()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 20 DO
       IF i<= 5 THEN
           INSERT INTO Hire(professor_id, ad_manger_id)
           VALUES(200000+i,400011);

       ELSEIF (i<=10 AND i>=6) THEN
           INSERT INTO Hire(professor_id, ad_manger_id)
           VALUES(200000+i-5,400012);

       ELSEIF (i<=15 AND i>=11) THEN
           INSERT INTO Hire(professor_id, ad_manger_id)
           VALUES(200000+i,400011);

       ELSE
           INSERT INTO Hire(professor_id, ad_manger_id)
           VALUES(200000+i,400011);

       END IF;
           SET i = i + 1;
```

```
   END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyTake_course$$
CREATE PROCEDURE makeDummyTake_course()
BEGIN
   DECLARE i INT DEFAULT 1;

   WHILE i <= 100 DO
       IF i <= 10 THEN
           INSERT INTO Take_course(student_id, course_id)
           VALUES(300000+i,700001);

       ELSEIF (i<=20 AND i>=11) THEN
           INSERT INTO Take_course(student_id, course_id)
           VALUES(300000+i-10,700002);

       ELSEIF (i<=30 AND i>=21) THEN
           INSERT INTO Take_course(student_id, course_id)
           VALUES(300000+i-10,700003);

       ELSEIF (i<=40 AND i>=31) THEN
           INSERT INTO Take_course(student_id, course_id)
           VALUES(300000+i-20,700004);

       ELSEIF (i<=50 AND i>=41) THEN
           INSERT INTO Take_course(student_id, course_id)
           VALUES(300000+i-30,700005);

       ELSEIF (i<=60 AND i>=51) THEN
           INSERT INTO Take_course(student_id, course_id)
           VALUES(300000+i-40,700006);

       ELSEIF (i<=70 AND i>=61) THEN
           INSERT INTO Take_course(student_id, course_id)
           VALUES(300000+i-30,700007);

       ELSEIF (i<=80 AND i>=71) THEN
           INSERT INTO Take_course(student_id, course_id)
           VALUES(300000+i-50,700008);

       ELSEIF (i<=90 AND i>=81) THEN
```

```
            INSERT INTO Take_course(student_id, course_id)
            VALUES(300000+i-50,700009);

        ELSE
            INSERT INTO Take_course(student_id, course_id)
            VALUES(300000+i-50,700010);

        END IF;
            SET i = i + 1;

    END WHILE;
END$$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS makeDummyEquipment$$
CREATE PROCEDURE makeDummyEquipment()
BEGIN
    DECLARE i INT DEFAULT 1;

    WHILE i <= 40 DO
        IF i<= 3 THEN
            INSERT INTO Equipment(lab_name, department_name, equipment_id, model_no,
date_purchased)
            VALUES('lab1','BIO',800000+i,concat('model-no-',i),'2009-10-18');

        ELSEIF (i<=6 AND i>=4) THEN
            INSERT INTO Equipment(lab_name, department_name, equipment_id, model_no,
date_purchased)
            VALUES('lab11','BIO',800000+i-3,concat('model-no-',i-1),'2007-10-18');

        ELSEIF (i<=20 AND i>=7) THEN
            INSERT INTO Equipment(lab_name, department_name, equipment_id, model_no,
date_purchased)
            VALUES('lab2','CS',800000+i-6,concat('model-no-',3),'2005-05-12');

        ELSEIF (i<=25 AND i>=21) THEN
            INSERT INTO Equipment(lab_name, department_name, equipment_id, model_no,
date_purchased)
            VALUES('lab3','CS',800000+i-20,concat('model-no-',i-20),'2015-05-05');

        ELSEIF (i<=30 AND i>=26) THEN
            INSERT INTO Equipment(lab_name, department_name, equipment_id, model_no,
date_purchased)
            VALUES('lab1','EE',800000+i-25,concat('model-no-',2),'2014-12-25');
```

```
       ELSEIF (i<=35 AND i>=31) THEN
           INSERT INTO Equipment(lab_name, department_name, equipment_id, model_no,
date_purchased)
           VALUES('lab13','BME',800000+i-30,concat('model-no-',i-30),'2022-01-04');

       ELSE
           INSERT INTO Equipment(lab_name, department_name, equipment_id, model_no,
date_purchased)
           VALUES('lab14','ECHE',800000+i-35,concat('model-no-',1),'2022-11-11');



       END IF;
           SET i = i + 1;

   END WHILE;
END$$
DELIMITER ;



CALL makeDummyAddress;
CALL makeDummyPerson;
CALL makeDummySchool;
CALL makeDummyDepartment;
CALL makeDummyProfessor;
CALL makeDummyStudent;
CALL makeDummyStaff;
CALL makeDummyStakeholder;
CALL makeDummyGraduate;
CALL makeDummyUndergraduate;
CALL makeDummyLaboratory;
CALL makeDummyTechnical_staff;
CALL makeDummyAdministrative_staff;
CALL makeDummyAD_manger;
CALL makeDummyCourse;
CALL makeDummyCommentSuggestion;
CALL makeDummyTeaching_Lab;
CALL makeDummyResearch_Lab;
CALL makeDummymodel;
CALL makeDummyMajor;
CALL makeDummyMinor;
CALL makeDummyConductExperiment;
CALL makeDummyAssigned_to;
CALL makeDummyHire;
CALL makeDummyTake_course;
```

```
CALL makeDummyEquipment;
```

4.sampleQuery.sql

```
--a) Identify all Stakeholders who have provided at least two comments or
suggestions.

SELECT stakeholder_id
FROM CommentSuggestion
GROUP BY stakeholder_id
HAVING COUNT(*) >= 2;

--b) Identify Undergraduate students who take over 15 credits and conduct
experiments in three different labs.

select Y.student_id
from (select student_id from Take_course as T, Course as C where T.course_id =
C.course_id Group by student_id Having SUM(credit) > 15) as X , (select student_id
from ConductExperiment as C group by student_id Having count(*)=3) as Y
where X.student_id = Y.student_id;

--c) Find all the graduate students who take research labs of the CS department.

select student_id
from Assigned_to as X
where X.department_name='CS';

--d) List all the Equipment belonging to a particular Lab.  (Change lab_Name and
department_name as needed.)

select lab_Name , department_name , equipment_id, model_no
from Equipment AS E
where lab_Name = 'lab1' AND department_name = 'BIO';

--e) Find all Professors who address at least 5 different research topics.

select professor_id
from Graduate as G
Group by professor_id
Having Count(DISTINCT research_Topic )>=5;
```

5. additionalQuery.sql

```
-- 1. Find all the students who have taken at least 4 courses and attended at least
3 different labs.
```

```sql
SELECT student_id
FROM Student
WHERE student_id IN (
   SELECT student_id
   FROM Take_course
   GROUP BY student_id
   HAVING COUNT(*) >= 4
) AND student_id IN (
   SELECT student_id
   FROM ConductExperiment
   GROUP BY student_id
   HAVING COUNT(*) >= 3
);

-- 2. Find the graduate students who is assigned to “lab1” lab in “CS” but did not take “course3” course.


SELECT student_id
FROM Graduate
WHERE student_id IN (
   SELECT student_id
   FROM Assigned_to
   WHERE lab_name = 'lab1' AND department_name = 'CS'
) AND student_id NOT IN (
   SELECT student_id
   FROM Take_course
   WHERE course_id IN (
       SELECT course_id
       FROM Course as C
       WHERE C.name = 'course3'
   )
);

-- 3. Find the Phone number of graduate students belonging to the rich lab. (Rich Lab: Labs with 5 or more pieces of equipment)
SELECT phone
FROM Person
WHERE person_id IN (
   SELECT person_id
   From Student
   WHERE student_id IN (
       SELECT student_id
       From Assigned_to AS X
       WHERE student_id IN (
```

```
            SELECT student_id
            FROM (SELECT lab_Name , department_name FROM Equipment GROUP BY
lab_Name, department_name HAVING COUNT(*) >= 5) AS Y
            WHERE X.lab_name=Y.lab_name AND X.department_name = Y.department_name
        )
    )
);
```