## STORED PROCEDURE 1

- Name: add\_person
- Purpose: After registration, this stored procedure adds a new person to the database. Based on the
  person type, it adds their details to their respective tables. For eg: If the new person is a student, then
  their details are added in both the PERSON table and the STUDENT table, and the same applies for
  STAFF and FACULTY tables.

### • Code:

```
use niner dining;
drop procedure if exists add person;
DELIMITER $$
create PROCEDURE add_person (in person_name varchar(300), in person_email varchar(150),
in person_type_param varchar(10), in phone_number bigint (15),in date_of_birth datetime,
in graduation_year int(20), in major varchar(75), in academic_level varchar(75),
in title varchar(75), in highest_degree varchar(75), in degree_college varchar(75),
in position varchar(75), in is_admin boolean, out result boolean)
BEGIN
declare personType int(20) default 1;
declare personID int(20);
declare sql_error TINYINT DEFAULT FALSE;
declare continue handler for sqlexception
set sql error = true;
set personType = (SELECT person_type_id FROM person_type where person_type = person_type_param);
insert into person (person_name, person_email,date_of_birth, phone_number,person_type_id,enrolled_date)
values(person_name, person_email, date_of_birth,phone_number, personType,now());
set personID = (SELECT LAST INSERT ID());
if(person_type_param = 'STUDENT') then
insert into student (person_id, graduation_year, major, academic_level) values
(personID, graduation_year, major, academic_level);
elseif(person_type_param = 'FACULTY') then
insert into faculty (person_id, title, highest_degree, degree_college) values
(personID, title, highest_degree, degree_college);
elseif(person type param = 'STAFF') then
insert into staff (person id, position, is admin) values
(personID, position, is_admin);
end if:
if sql\_error = false then
set result = true;
commit;
else
set result = false;
rollback;
end if;
END$$
DELIMITER:
```

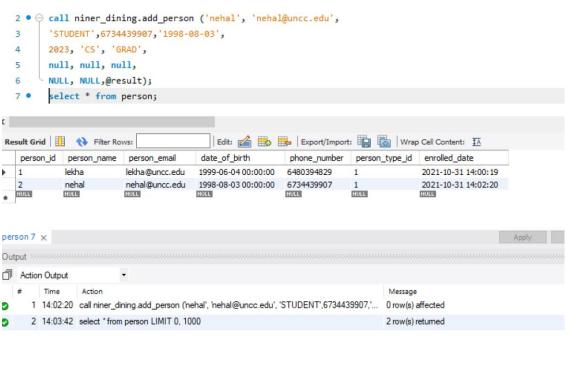
### • Screenshot of execution:

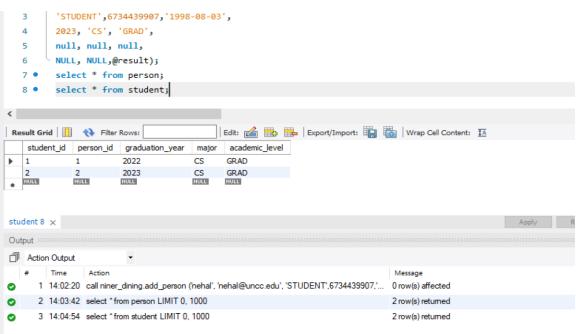
```
□ □ □ | \( \frac{\nagger}{R} \) \( \frac{\nagger}{
      1 • use niner_dining;
      2 • drop procedure if exists add_person;
                      DELIMITER $$
      5
                     in person_type_param varchar(10), in phone_number bigint (15),in date_of_birth datetime,
                      in graduation_year int(20), in major varchar(75), in academic_level varchar(75),
      6
                      in title varchar(75), in highest_degree varchar(75), in degree_college varchar(75),
                  in position varchar(75), in is_admin boolean, out result boolean)
      8
      9 ⊝ BEGIN
   10
                      declare personType int(20) default 1;
   11
                    declare personID int(20);
   12
                   declare sql_error TINYINT DEFAULT FALSE;
   13
                     declare continue handler for sqlexception
                      set sql_error = true ;
   14
                       set personType = (SELECT person type id FROM person type where person type = person type param);
    15
Output :::::
Action Output
           1 13:46:22 use niner_dining
                                                                                                                                                                                                           0 row(s) affected
2 13:46:22 drop procedure if exists add_person
                                                                                                                                                                                                           0 row(s) affected
              3 13:46:22 create PROCEDURE add_person (in person_name varchar(300), in person_email var... 0 row(s) affected, 4 warning(s): 1681 Integer display width
```

### • Screenshot of call:



### • Screenshot of results:





# **STORED PROCEDURE 2**

- Name: add\_restaurant
- **Purpose:** Adds details of a new restaurant as well as its social media details to the database.

# • Code:

```
use niner_dining;
drop procedure if exists add_restaurant;
DELIMITER $$
create PROCEDURE add_restaurant (in location varchar(200), in restaurant_name varchar(75),
in working_hours varchar(20), in website varchar(200), in email varchar(200), in instagram_id
varchar(200),
in facebook_id varchar(200), out result boolean)
BEGIN
declare restaurantID int(20) default 1;
declare sql_error TINYINT DEFAULT FALSE;
declare continue handler for sqlexception
set sql_error = true;
insert into restaurant (location, restaurant_name, working_hours)
values (location, restaurant_name, working_hours);
set restaurantID = (SELECT LAST_INSERT_ID());
insert into social_media_details ( restaurant_id, website, email, instagram_id, facebook_id )
values (restaurantID, website, email, instagram_id, facebook_id);
if sql error = false then
set result = true;
commit;
else
set result = false;
rollback;
end if;
END$$
DELIMITER;
```

## • Screenshot of execution:

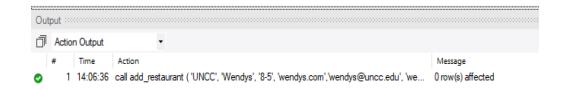
```
use niner_dining;
        drop procedure if exists add_restaurant;
         DELIMITER $$
  4 • ⊖ create PROCEDURE add_restaurant (in location varchar(200), in restaurant_name varchar(75),
         in working_hours varchar(20), in website varchar(200), in email varchar(200), in instagram_id varchar(20
         in facebook_id varchar(200), out result boolean)

⊖ BEGIN

         declare restaurantID int(20) default 1;
        declare sql_error TINYINT DEFAULT FALSE;
 10
        declare continue handler for sqlexception
 11
        set sal error = true ;
 12
         insert into restaurant ( location, restaurant_name, working_hours )
 13
         values (location, restaurant_name, working_hours);
         set restaurantID = (SELECT LAST_INSERT_ID());
 14
        insert into social_media_details ( restaurant_id, website, email, instagram_id, facebook_id )
 15
         values ( restaurantID, website, email, instagram_id, facebook_id);
Output :
Action Output
                                                                                Message
     1 13:49:59 use niner_dining
                                                                                0 row(s) affected
2 13:49:59 drop procedure if exists add_restaurant
                                                                               0 row(s) affected
     3 13:49:59 create PROCEDURE add_restaurant (in location varchar(200), in restaurant_name v... 0 row(s) affected, 1 warning(s): 1681 Integer display width
```

# • Screenshot of Call:

```
1 •  call add_restaurant ( 'UNCC', 'Wendys', '8-5', 'wendys.com', 'wendys@uncc.edu',
2  'wendysIG', 'wendysFB', @result);
3
```



## • Screenshot of results:

