EMPLOYEE DATABASE MANAGEMENT

Group Members –

1. CIJIN JOSEPH JOHN
2. MITHUN BABU
3. ATHUL ASHOK
4. VIVEK KUMAR CHAUBEY

AIM

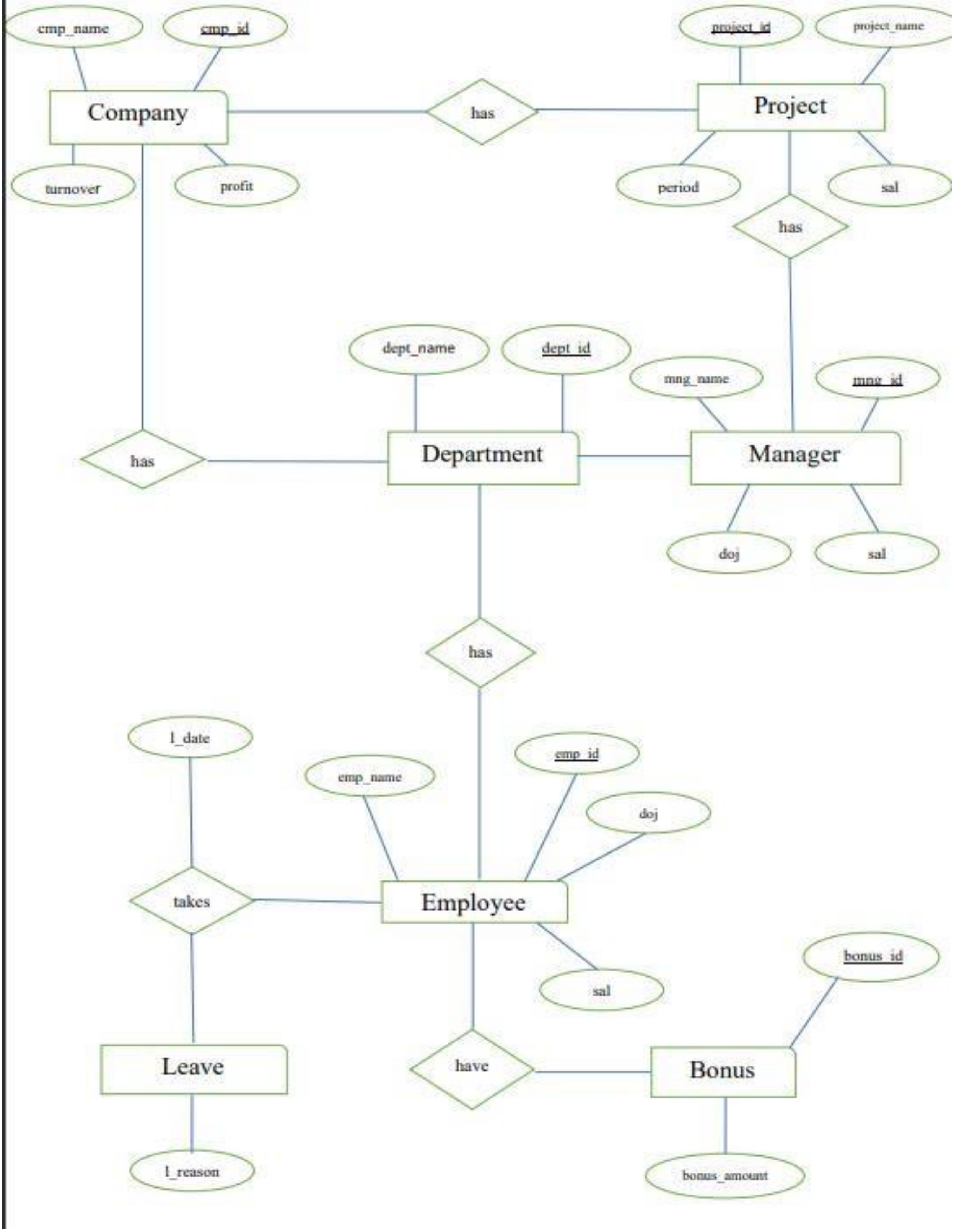
Database to demonstrate Employee Management System. It is developed to maintain the details of employee, managers working for any organisation. It maintains the information about the personal and professional details of their employees.

ABSTRACT

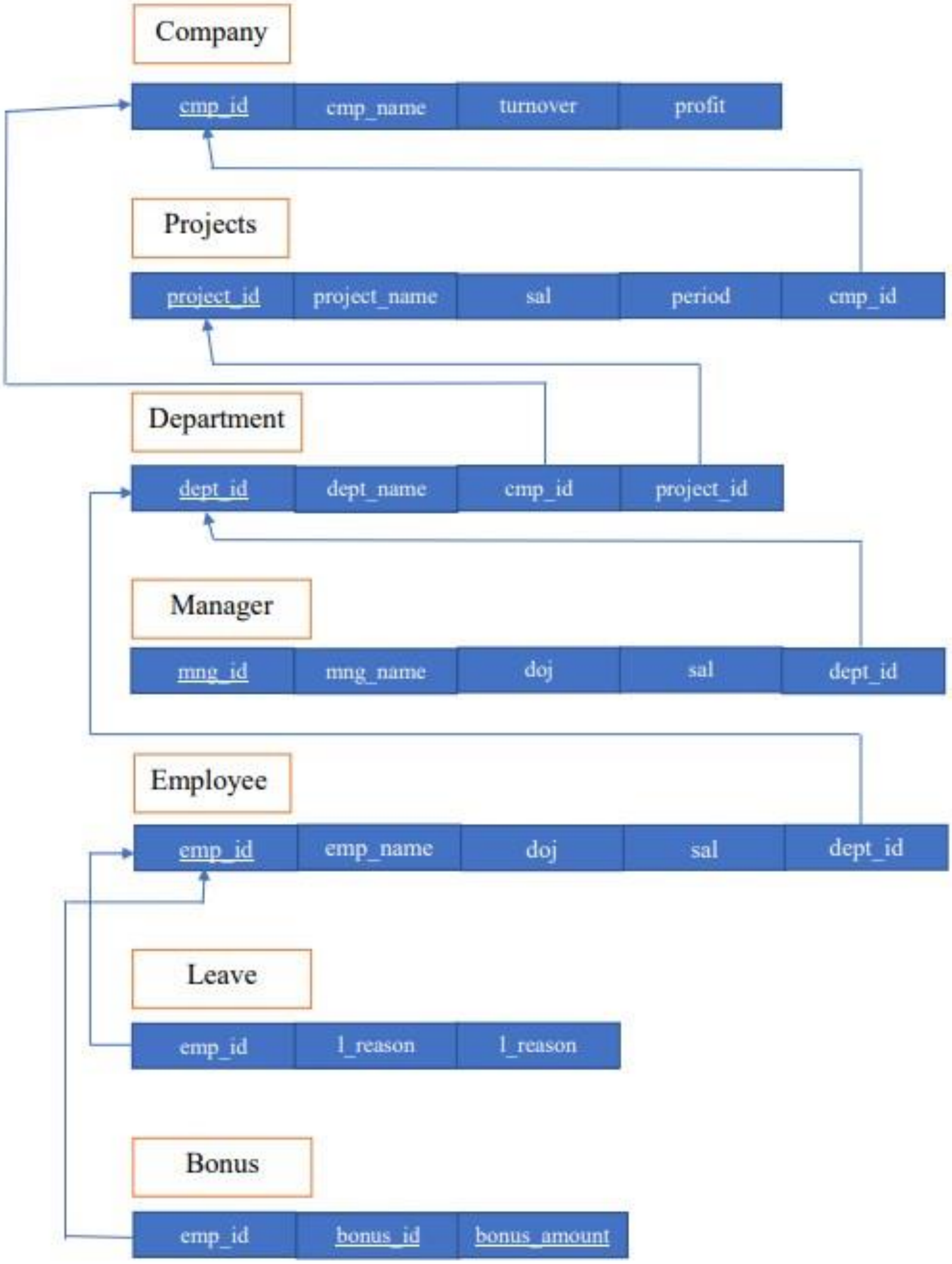
Employee database, which keeps the records of all the details pertaining to employees in a company. Along with that it stores the data regarding managers and their projects details. The database also contains data about employee bonus, tax, grants and also the leaves taken. Functions are written to calculate the turnover of company and also the company profits.

The project describes how to manage user data to good performance.

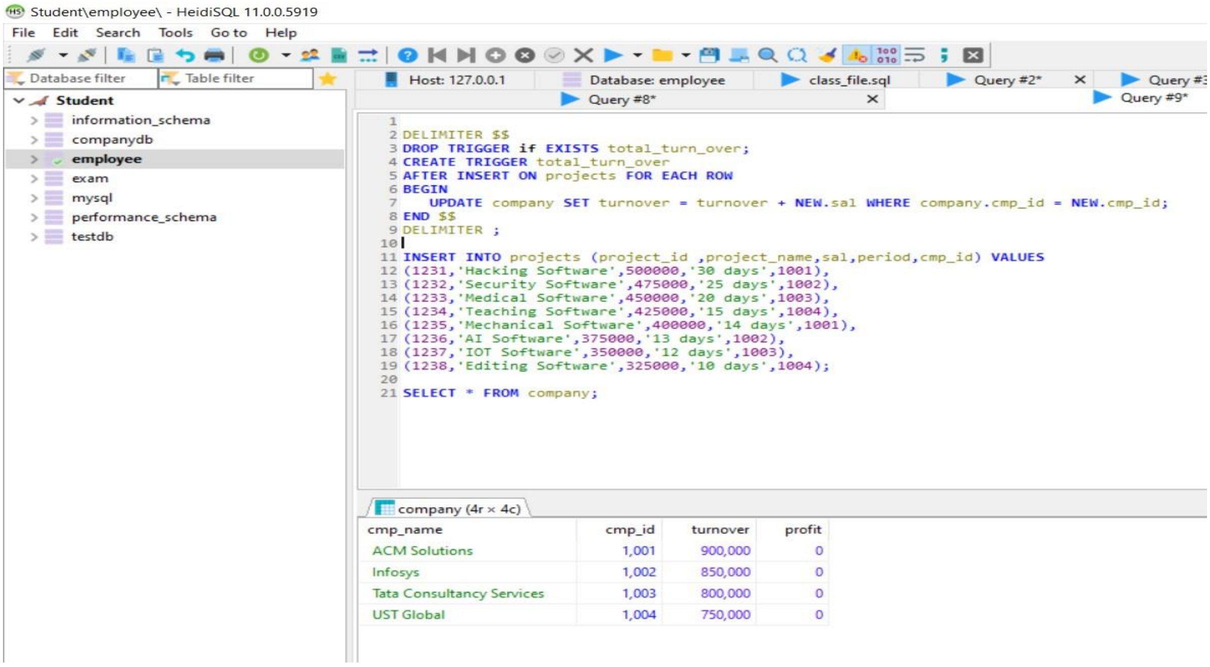
ER DIAGRAM


cmp_name
cmp_id
Company
turnover
profit
has
project_id
project_name
Project
period
sal
has
dept_name
dept_id
mng_name
mng_id
has
Department
Manager
doj
sal
has
l_date
emp_name
emp_id
doj
takes
Employee
sal
bonus_id
Leave
have
Bonus
l_reason
bonus_amount

SCHEMA DIAGRAM



```
CREATE TABLE company(cmp_name VARCHAR(50) NOT NULL , cmp_id INTEGER NOT
NULL , turnover DECIMAL , profit DECIMAL);

CREATE TABLE projects (project_id INTEGER PRIMARY KEY ,project_name
VARCHAR(50),sal DECIMAL, period VARCHAR(50),cmp_id INT REFERENCES
company(cmp_id));

CREATE TABLE department (dept_id INTEGER PRIMARY KEY
AUTO_INCREMENT,dept_name VARCHAR(30),cmp_id INT REFERENCES company(cmp_id),
project_id INT REFERENCES projects(project_id));

CREATE TABLE manager (mng_name VARCHAR(30) NOT NULL,mng_id INTEGER PRIMARY
KEY,doj DATE,sal DECIMAL,dept_id INT REFERENCES department(dept_id));

CREATE TABLE employee (emp_name VARCHAR(30) NOT NULL,emp_id INTEGER PRIMARY
KEY,doj DATE,sal DECIMAL,dept_id INT REFERENCES department(dept_id));

CREATE TABLE lEAVES (l_date DATE NOT NULL ,l_reason VARCHAR(30), emp_id INT
REFERENCES employee(emp_id));

CREATE TABLE bonus (bonus_id VARCHAR(6) primary key,emp_id INT REFERENCES
employee(emp_id), bonus_amount DECIMAL(7,2));

CREATE TABLE emp_backup (e_id INT PRIMARY KEY ,e_name VARCHAR(30) NOT NULL
, doj DATE ,sal DECIMAL(10,2), d_id INT );



DELIMITER $$
DROP TRIGGER if EXISTS total_turn_over;
CREATE TRIGGER total_turn_over AFTER
INSERT ON projects FOR EACH ROW
BEGIN
      UPDATE company SET turnover = turnover + NEW.sal WHERE company.cmp_id
= NEW.cmp_id;
END $$
DELIMITER ;
```

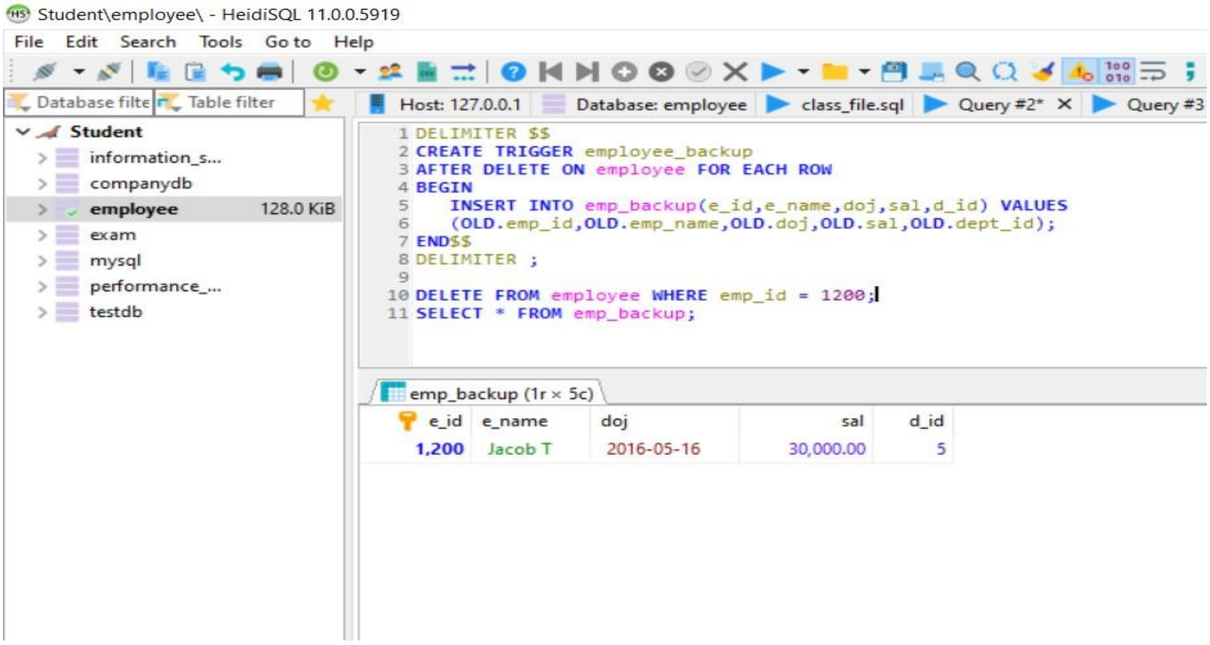


```
DELIMITER $$
DROP TRIGGER if EXISTS total_turn_over;
CREATE TRIGGER total_turn_over
AFTER INSERT ON projects FOR EACH ROW
BEGIN
   UPDATE company SET turnover = turnover + NEW.sal WHERE company.cmp_id = NEW.cmp_id;
END $$
DELIMITER ;

INSERT INTO projects (project_id ,project_name,sal,period,cmp_id) VALUES
(1231,'Hacking Software',500000,'30 days',1001),
(1232,'Security Software',475000,'25 days',1002),
(1233,'Medical Software',450000,'20 days',1003),
(1234,'Teaching Software',425000,'15 days',1004),
(1235,'Mechanical Software',400000,'14 days',1001),
(1236,'AI Software',375000,'13 days',1002),
(1237,'IOT Software',350000,'12 days',1003),
(1238,'Editing Software',325000,'10 days',1004);

SELECT * FROM company;
```

company (4r × 4c)

cmp_name	cmp_id	turnover	profit
ACM Solutions	1,001	900,000	0
Infosys	1,002	850,000	0
Tata Consultancy Services	1,003	800,000	0
UST Global	1,004	750,000	0

```
DELIMITER $$
CREATE TRIGGER employee_backup AFTER
DELETE ON employee FOR EACH ROW
BEGIN
       INSERT INTO emp_backup(e_id,e_name,doj,sal,d_id) VALUES
       (OLD.emp_id,OLD.emp_name,OLD.doj,OLD.sal,OLD.dept_id);
END$$
DELIMITER ;
```



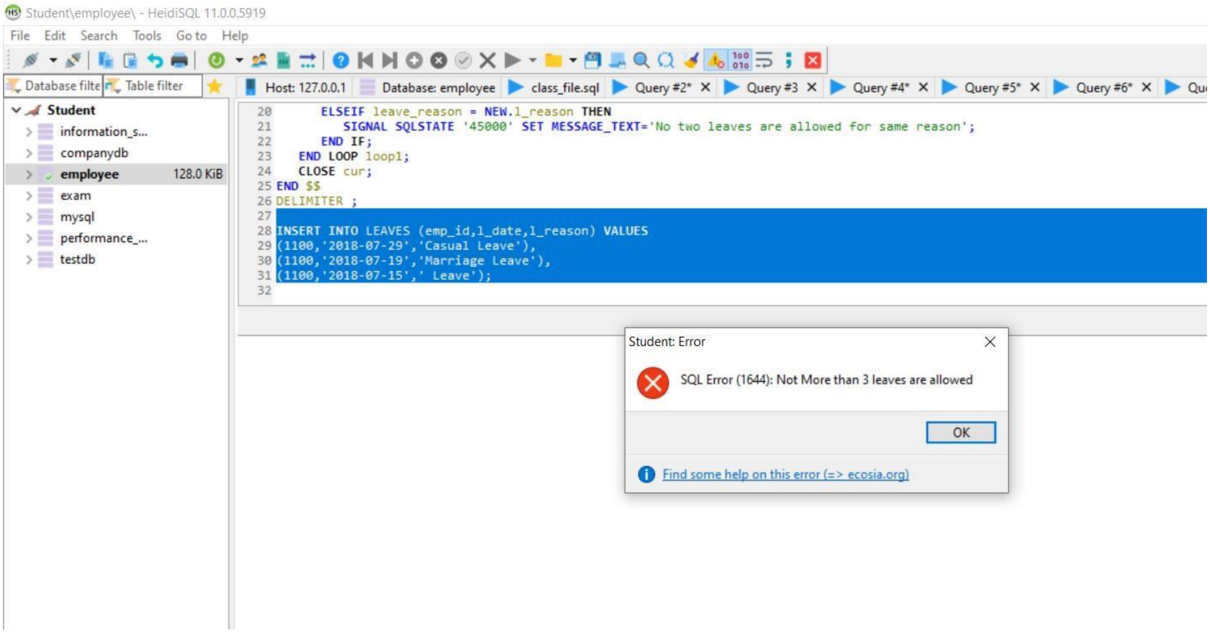
```
DELIMITER $$
CREATE TRIGGER employee_backup
AFTER DELETE ON employee FOR EACH ROW
BEGIN
    INSERT INTO emp_backup(e_id,e_name,doj,sal,d_id) VALUES
    (OLD.emp_id,OLD.emp_name,OLD.doj,OLD.sal,OLD.dept_id);
END$$
DELIMITER ;

DELETE FROM employee WHERE emp_id = 1200;
SELECT * FROM emp_backup;
```

emp_backup (1r × 5c)

e_id	e_name	doj	sal	d_id
1,200	Jacob T	2016-05-16	30,000.00	5

```
DELIMITER $$
DROP TRIGGER IF EXISTS check_for_leave;
CREATE TRIGGER check_for_leave BEFORE
INSERT ON leaves FOR EACH ROW
BEGIN
       DECLARE c INT ;
```

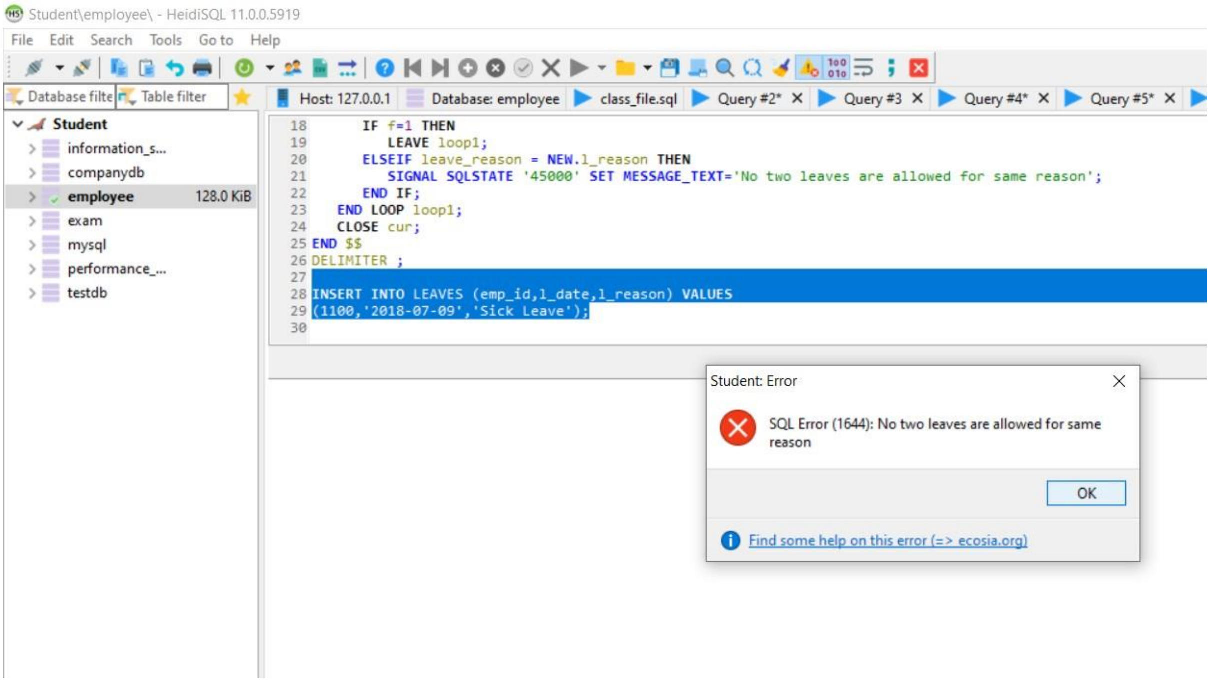
```
      DECLARE f INT DEFAULT 0;
      DECLARE leave_reason VARCHAR(30);
      DECLARE cur CURSOR FOR SELECT l_reason FROM leaves WHERE emp_id =
NEW.emp_id;
      DECLARE CONTINUE HANDLER FOR NOT FOUND SET f=1;
      SELECT COUNT(emp_id) FROM leaves WHERE emp_id=NEW.emp_id INTO c;
      IF c>=3 THEN
   SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='Not More than 3 leaves are
allowed';
      END IF;
OPEN cur;   loop1
: LOOP
            FETCH cur INTO leave_reason;
            IF f=1 THEN
                  LEAVE loop1;
            ELSEIF leave_reason = NEW.l_reason THEN
                  SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='No two leaves
are allowed for same reason';
            END IF;
      END LOOP loop1;
      CLOSE cur;
END $$
DELIMITER ;
```



```
      ELSEIF leave_reason = NEW.l_reason THEN
         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='No two leaves are allowed for same reason';
      END IF;
   END LOOP loop1;
   CLOSE cur;
END $$
DELIMITER ;

INSERT INTO LEAVES (emp_id,l_date,l_reason) VALUES
(1100,'2018-07-29','Casual Leave'),
(1100,'2018-07-19','Marriage Leave'),
(1100,'2018-07-15',' Leave');
```

Student: Error

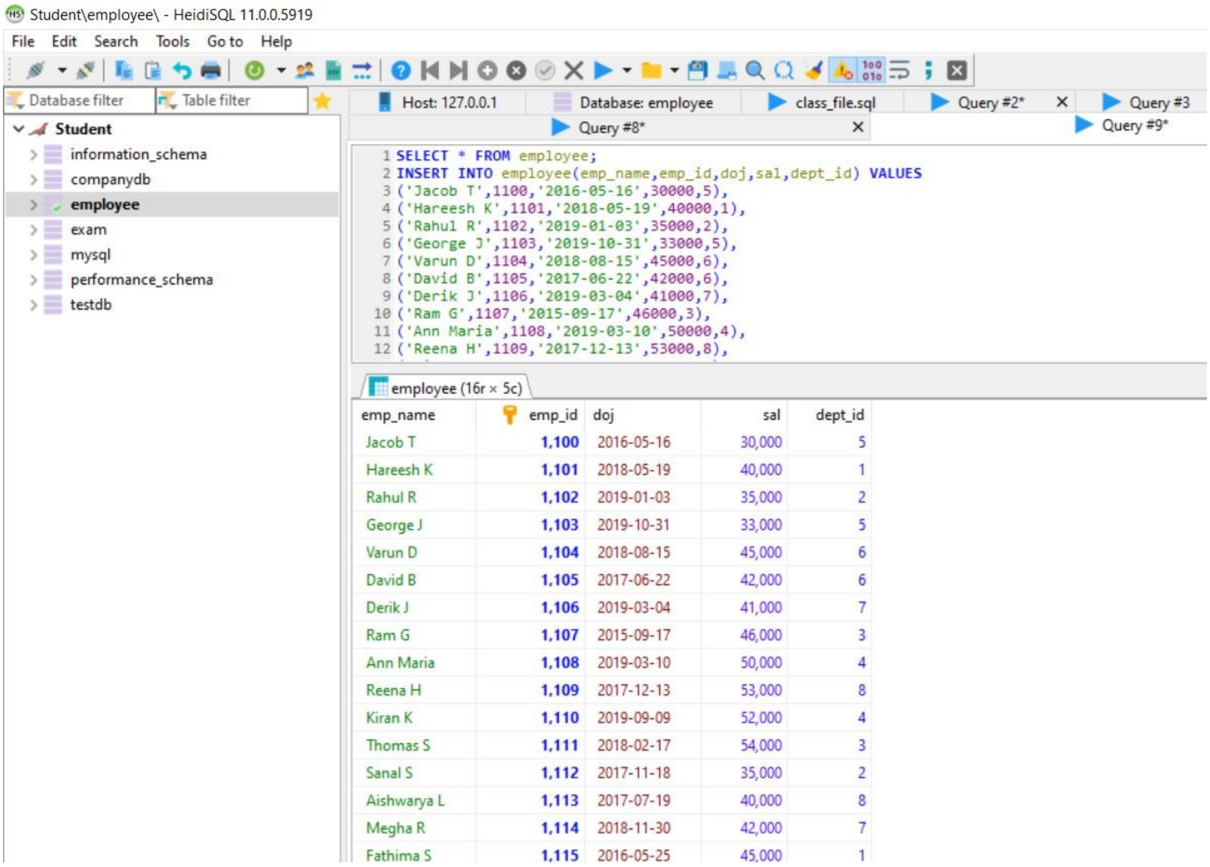
SQL Error (1644): Not More than 3 leaves are allowed



```
        IF f=1 THEN
           LEAVE loop1;
        ELSEIF leave_reason = NEW.l_reason THEN
           SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='No two leaves are allowed for same reason';
        END IF;
     END LOOP loop1;
     CLOSE cur;
  END $$
  DELIMITER ;

  INSERT INTO LEAVES (emp_id,l_date,l_reason) VALUES
  (1100,'2018-07-09','Sick Leave');
```

SQL Error (1644): No two leaves are allowed for same reason

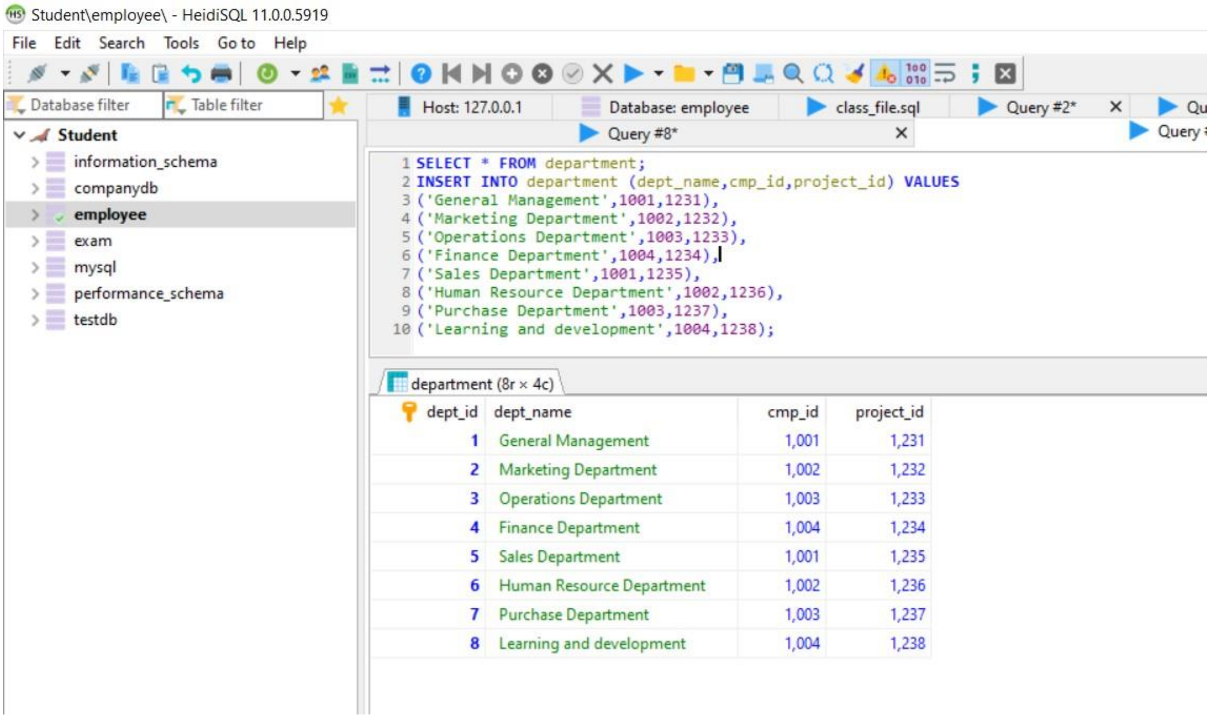
```
INSERT INTO employee(emp_name,emp_id,doj,sal,dept_id) VALUES
('Jacob T',1100,'2016-05-16',30000,5),
('Hareesh K',1101,'2018-05-19',40000,1),
('Rahul R',1102,'2019-01-03',35000,2),
('George J',1103,'2019-10-31',33000,5),
('Varun D',1104,'2018-08-15',45000,6),
('David B',1105,'2017-06-22',42000,6),
('Derik J',1106,'2019-03-04',41000,7),
('Ram G',1107,'2015-09-17',46000,3),
('Ann Maria',1108,'2019-03-10',50000,4),
('Reena H',1109,'2017-12-13',53000,8),
('Kiran K',1110,'2019-09-09',52000,4),
('Thomas S',1111,'2018-02-17',54000,3),
('Sanal S',1112,'2017-11-18',35000,2),
('Aishwarya L',1113,'2017-07-19',40000,8),
('Megha R',1114,'2018-11-30',42000,7),
('Fathima S',1115,'2016-05-25',45000,1);
```



```sql
SELECT * FROM employee;
INSERT INTO employee(emp_name,emp_id,doj,sal,dept_id) VALUES
('Jacob T',1100,'2016-05-16',30000,5),
('Hareesh K',1101,'2018-05-19',40000,1),
('Rahul R',1102,'2019-01-03',35000,2),
('George J',1103,'2019-10-31',33000,5),
('Varun D',1104,'2018-08-15',45000,6),
('David B',1105,'2017-06-22',42000,6),
('Derik J',1106,'2019-03-04',41000,7),
('Ram G',1107,'2015-09-17',46000,3),
('Ann Maria',1108,'2019-03-10',50000,4),
('Reena H',1109,'2017-12-13',53000,8),
```

emp_name	emp_id	doj	sal	dept_id
Jacob T	1,100	2016-05-16	30,000	5
Hareesh K	1,101	2018-05-19	40,000	1
Rahul R	1,102	2019-01-03	35,000	2
George J	1,103	2019-10-31	33,000	5
Varun D	1,104	2018-08-15	45,000	6
David B	1,105	2017-06-22	42,000	6
Derik J	1,106	2019-03-04	41,000	7
Ram G	1,107	2015-09-17	46,000	3
Ann Maria	1,108	2019-03-10	50,000	4
Reena H	1,109	2017-12-13	53,000	8
Kiran K	1,110	2019-09-09	52,000	4
Thomas S	1,111	2018-02-17	54,000	3
Sanal S	1,112	2017-11-18	35,000	2
Aishwarya L	1,113	2017-07-19	40,000	8
Megha R	1,114	2018-11-30	42,000	7
Fathima S	1,115	2016-05-25	45,000	1

```sql
INSERT INTO department (dept_name,cmp_id,project_id) VALUES
('General Management',1001,1231),
('Marketing Department',1002,1232),
('Operations Department',1003,1233),
('Finance Department',1004,1234),
('Sales Department',1001,1235),
('Human Resource Department',1002,1236),
('Purchase Department',1003,1237),
('Learning and development',1004,1238);
```

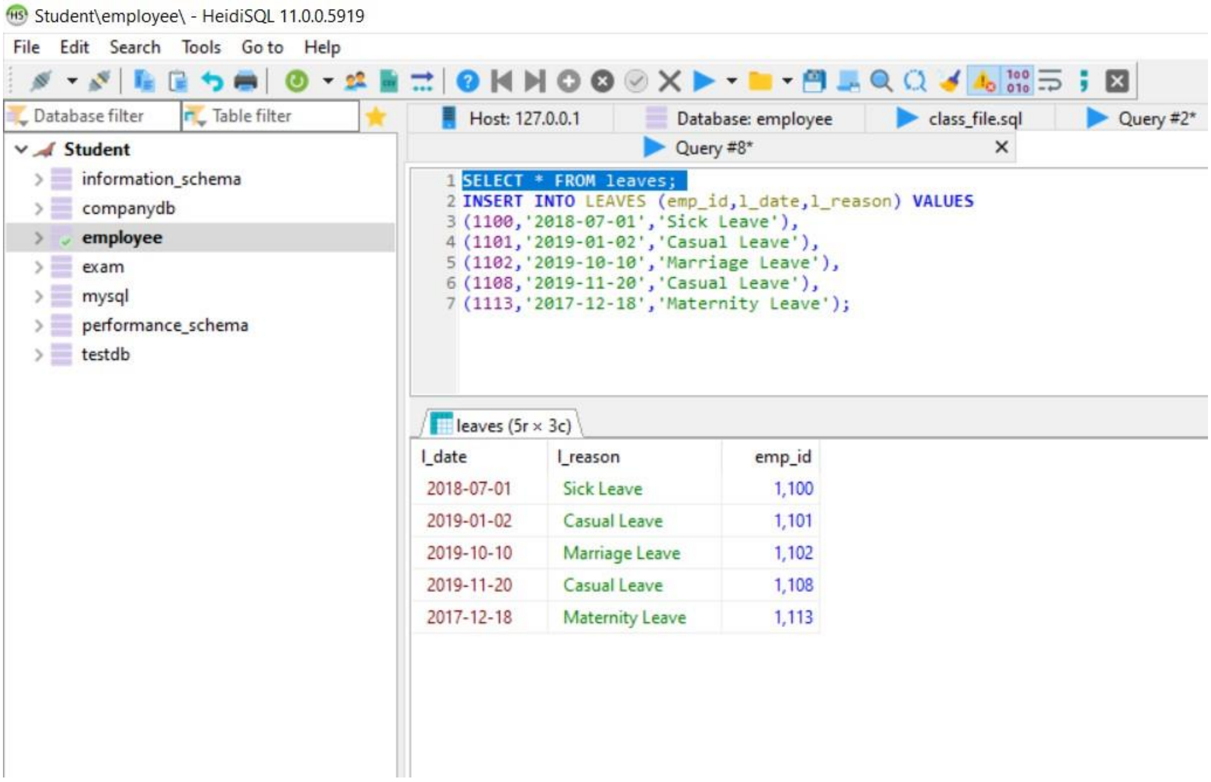


```sql
SELECT * FROM department;
INSERT INTO department (dept_name,cmp_id,project_id) VALUES
('General Management',1001,1231),
('Marketing Department',1002,1232),
('Operations Department',1003,1233),
('Finance Department',1004,1234),
('Sales Department',1001,1235),
('Human Resource Department',1002,1236),
('Purchase Department',1003,1237),
('Learning and development',1004,1238);
```

department (8r × 4c)

dept_id	dept_name	cmp_id	project_id
1	General Management	1,001	1,231
2	Marketing Department	1,002	1,232
3	Operations Department	1,003	1,233
4	Finance Department	1,004	1,234
5	Sales Department	1,001	1,235
6	Human Resource Department	1,002	1,236
7	Purchase Department	1,003	1,237
8	Learning and development	1,004	1,238

```sql
INSERT INTO LEAVES (emp_id,l_date,l_reason) VALUES
(1100,'2018-07-01','Sick Leave'),
(1101,'2019-01-02','Casual Leave'),
(1102,'2019-10-10','Marriage Leave'),
(1108,'2019-11-20','Casual Leave'),
(1113,'2017-12-18','Maternity Leave');
```

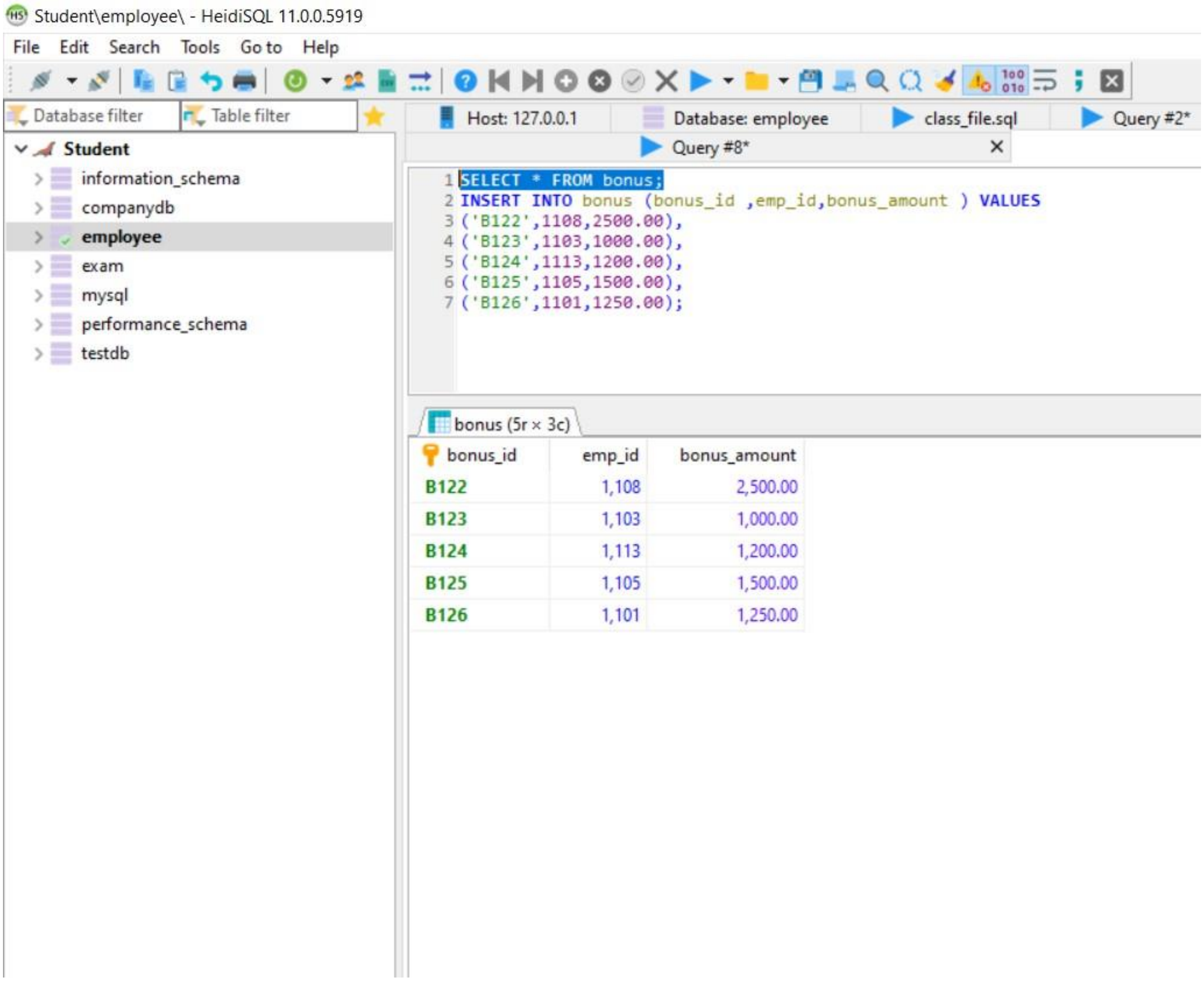


```sql
SELECT * FROM leaves;
INSERT INTO LEAVES (emp_id,l_date,l_reason) VALUES
(1100,'2018-07-01','Sick Leave'),
(1101,'2019-01-02','Casual Leave'),
(1102,'2019-10-10','Marriage Leave'),
(1108,'2019-11-20','Casual Leave'),
(1113,'2017-12-18','Maternity Leave');
```

leaves (5r × 3c)

l_date	l_reason	emp_id
2018-07-01	Sick Leave	1,100
2019-01-02	Casual Leave	1,101
2019-10-10	Marriage Leave	1,102
2019-11-20	Casual Leave	1,108
2017-12-18	Maternity Leave	1,113

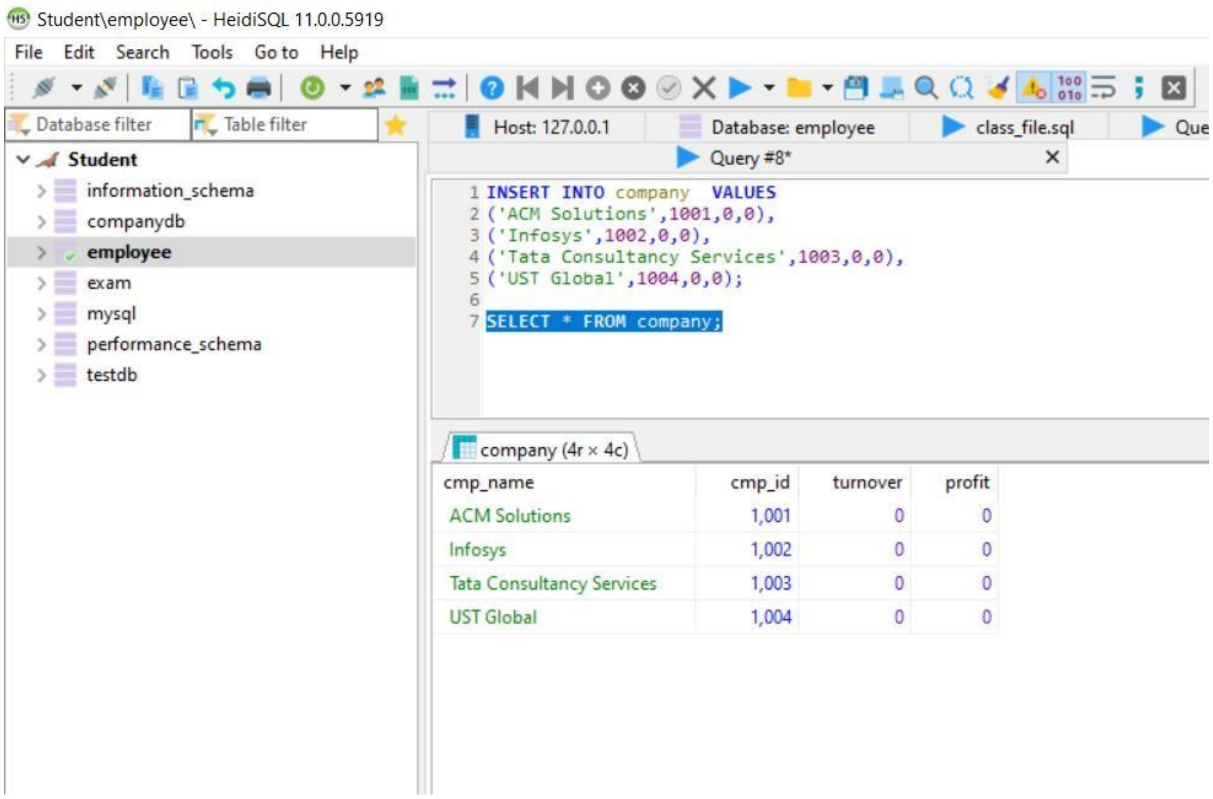
```
INSERT INTO bonus (bonus_id ,emp_id,bonus_amount ) VALUES
('B122',1108,2500.00),
('B123',1103,1000.00),
('B124',1113,1200.00),
('B125',1105,1500.00),
('B126',1101,1250.00);
```



```
SELECT * FROM bonus;
INSERT INTO bonus (bonus_id ,emp_id,bonus_amount ) VALUES
('B122',1108,2500.00),
('B123',1103,1000.00),
('B124',1113,1200.00),
('B125',1105,1500.00),
('B126',1101,1250.00);
```

bonus_id	emp_id	bonus_amount
B122	1,108	2,500.00
B123	1,103	1,000.00
B124	1,113	1,200.00
B125	1,105	1,500.00
B126	1,101	1,250.00

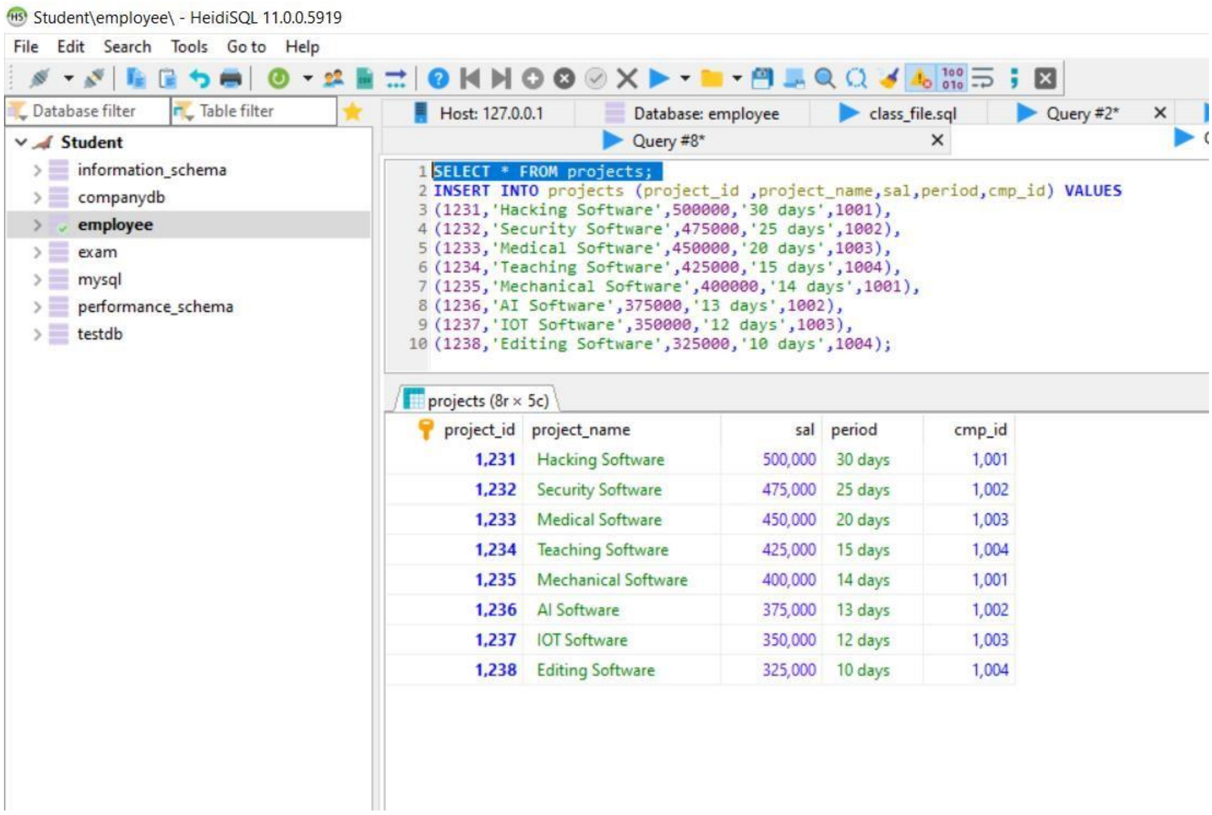
```
INSERT INTO company  VALUES
('ACM Solutions',1001,0,0),
('Infosys',1002,0,0),
('Tata Consultancy Services',1003,0,0),
('UST Global',1004,0,0);
```



```
INSERT INTO company  VALUES
('ACM Solutions',1001,0,0),
('Infosys',1002,0,0),
('Tata Consultancy Services',1003,0,0),
('UST Global',1004,0,0);

SELECT * FROM company;
```

cmp_name	cmp_id	turnover	profit
ACM Solutions	1,001	0	0
Infosys	1,002	0	0
Tata Consultancy Services	1,003	0	0
UST Global	1,004	0	0

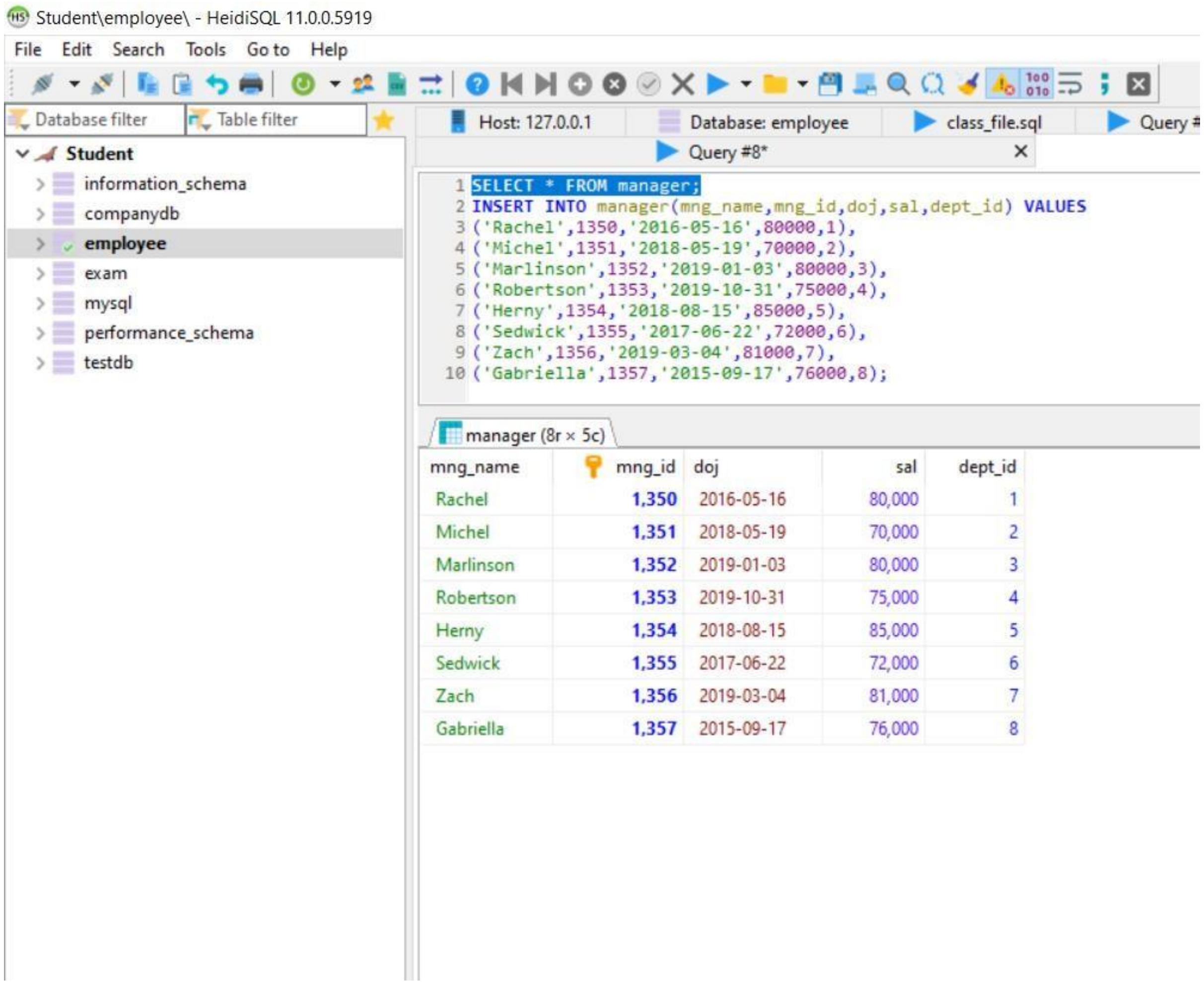
```
INSERT INTO projects (project_id ,project_name,sal,period,cmp_id) VALUES
(1231,'Hacking Software',500000,'30 days',1001),
(1232,'Security Software',475000,'25 days',1002),
(1233,'Medical Software',450000,'20 days',1003),
(1234,'Teaching Software',425000,'15 days',1004),
(1235,'Mechanical Software',400000,'14 days',1001),
(1236,'AI Software',375000,'13 days',1002),
(1237,'IOT Software',350000,'12 days',1003),
(1238,'Editing Software',325000,'10 days',1004);
```



```sql
SELECT * FROM projects;
INSERT INTO projects (project_id ,project_name,sal,period,cmp_id) VALUES
(1231,'Hacking Software',500000,'30 days',1001),
(1232,'Security Software',475000,'25 days',1002),
(1233,'Medical Software',450000,'20 days',1003),
(1234,'Teaching Software',425000,'15 days',1004),
(1235,'Mechanical Software',400000,'14 days',1001),
(1236,'AI Software',375000,'13 days',1002),
(1237,'IOT Software',350000,'12 days',1003),
(1238,'Editing Software',325000,'10 days',1004);
```

project_id	project_name	sal	period	cmp_id
1,231	Hacking Software	500,000	30 days	1,001
1,232	Security Software	475,000	25 days	1,002
1,233	Medical Software	450,000	20 days	1,003
1,234	Teaching Software	425,000	15 days	1,004
1,235	Mechanical Software	400,000	14 days	1,001
1,236	AI Software	375,000	13 days	1,002
1,237	IOT Software	350,000	12 days	1,003
1,238	Editing Software	325,000	10 days	1,004

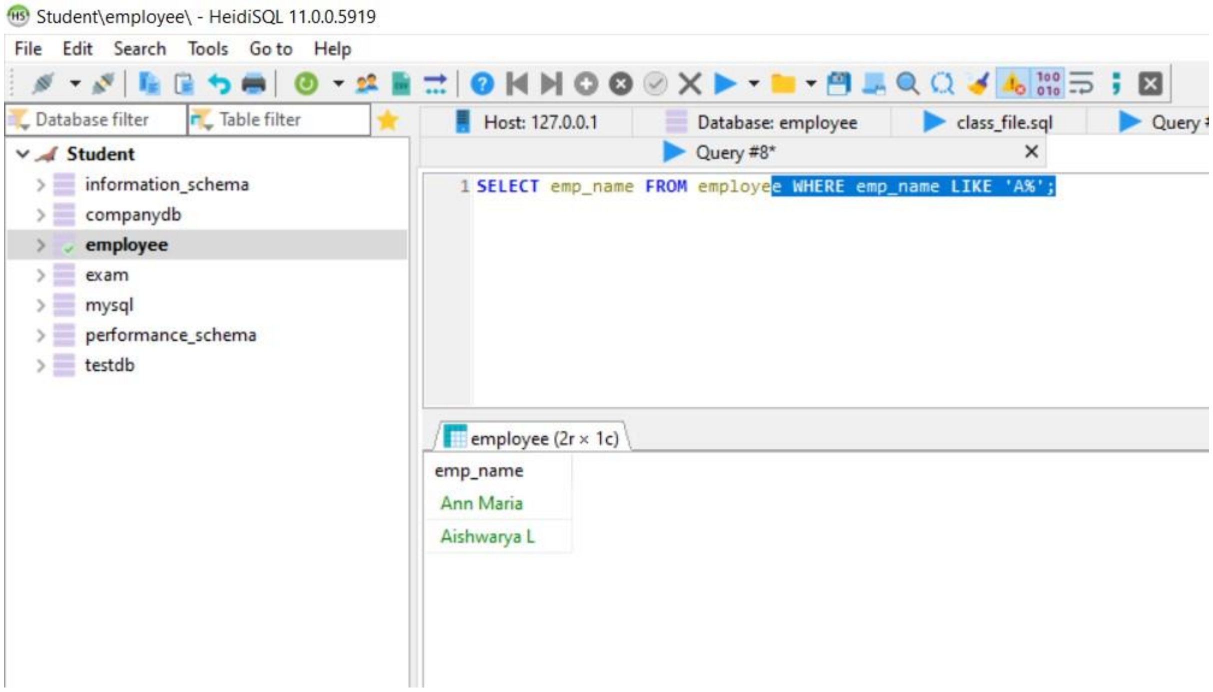
```sql
INSERT INTO manager(mng_name,mng_id,doj,sal,dept_id) VALUES
('Rachel',1350,'2016-05-16',80000,1),
('Michel',1351,'2018-05-19',70000,2),
('Marlinson',1352,'2019-01-03',80000,3),
('Robertson',1353,'2019-10-31',75000,4),
('Herny',1354,'2018-08-15',85000,5),
('Sedwick',1355,'2017-06-22',72000,6),
('Zach',1356,'2019-03-04',81000,7),
('Gabriella',1357,'2015-09-17',76000,8);
```



```
SELECT * FROM manager;
INSERT INTO manager(mng_name,mng_id,doj,sal,dept_id) VALUES
('Rachel',1350,'2016-05-16',80000,1),
('Michel',1351,'2018-05-19',70000,2),
('Marlinson',1352,'2019-01-03',80000,3),
('Robertson',1353,'2019-10-31',75000,4),
('Herny',1354,'2018-08-15',85000,5),
('Sedwick',1355,'2017-06-22',72000,6),
('Zach',1356,'2019-03-04',81000,7),
('Gabriella',1357,'2015-09-17',76000,8);
```

mng_name	mng_id	doj	sal	dept_id
Rachel	1,350	2016-05-16	80,000	1
Michel	1,351	2018-05-19	70,000	2
Marlinson	1,352	2019-01-03	80,000	3
Robertson	1,353	2019-10-31	75,000	4
Herny	1,354	2018-08-15	85,000	5
Sedwick	1,355	2017-06-22	72,000	6
Zach	1,356	2019-03-04	81,000	7
Gabriella	1,357	2015-09-17	76,000	8

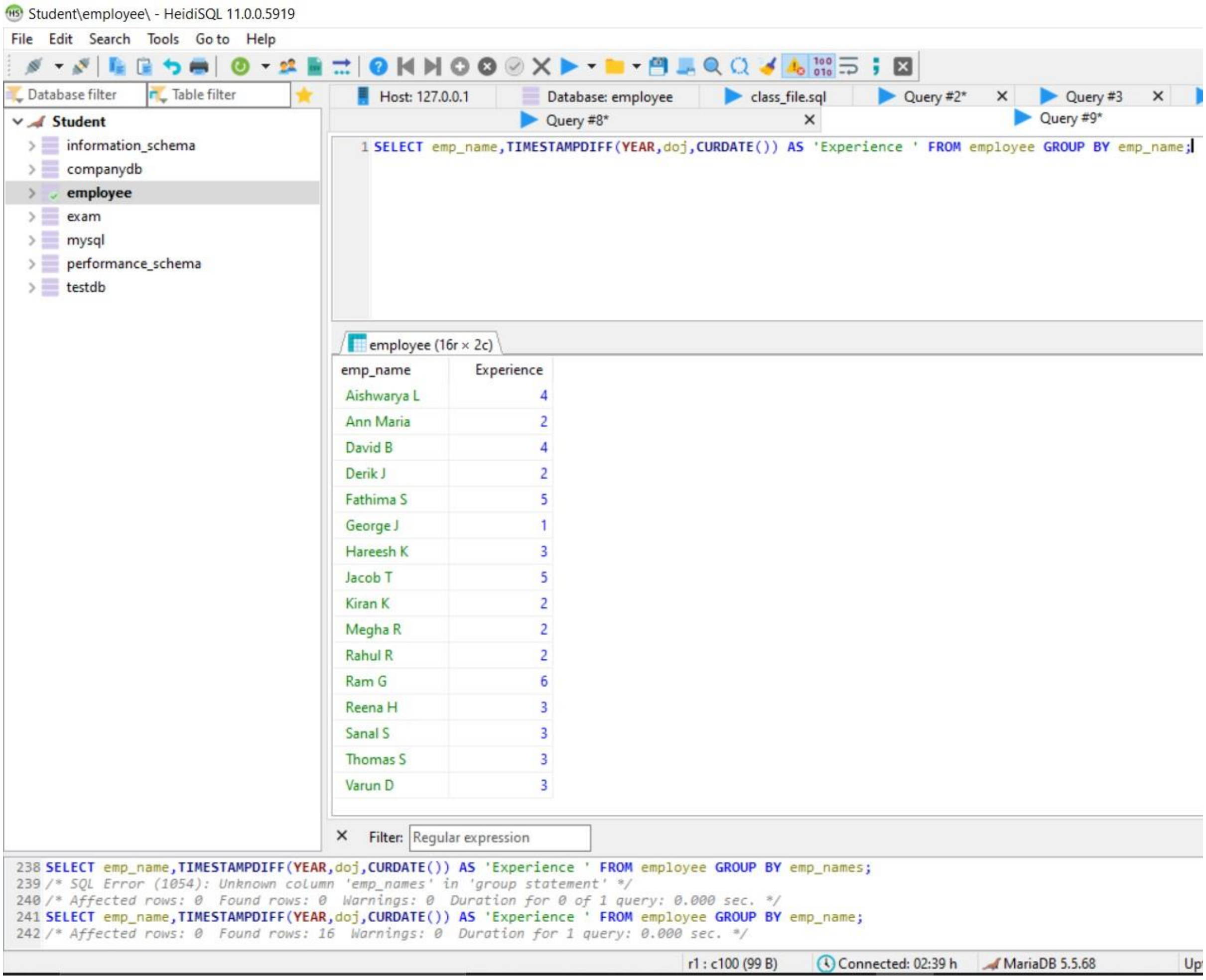
```
SELECT emp_name FROM employee WHERE emp_name LIKE 'A%';
```



```
SELECT emp_name FROM employee WHERE emp_name LIKE 'A%';
```

emp_name
Ann Maria
Aishwarya L

```
SELECT emp_name,TIMESTAMPDIFF(YEAR,doj,CURDATE()) AS 'Experience ' FROM
employee GROUP BY emp_name;
```

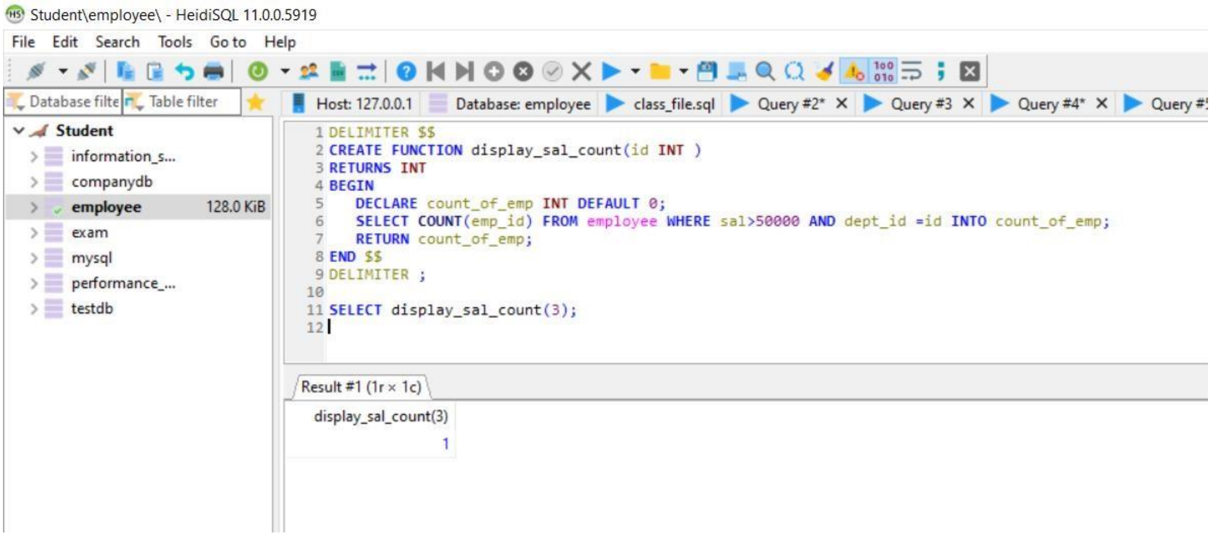


```
SELECT emp_name,TIMESTAMPDIFF(YEAR,doj,CURDATE()) AS 'Experience ' FROM employee GROUP BY emp_name;
```

emp_name	Experience
Aishwarya L	4
Ann Maria	2
David B	4
Derik J	2
Fathima S	5
George J	1
Hareesh K	3
Jacob T	5
Kiran K	2
Megha R	2
Rahul R	2
Ram G	6
Reena H	3
Sanal S	3
Thomas S	3
Varun D	3

```
SELECT emp_name,TIMESTAMPDIFF(YEAR,doj,CURDATE()) AS 'Experience ' FROM employee GROUP BY emp_names;
/* SQL Error (1054): Unknown column 'emp_names' in 'group statement' */
/* Affected rows: 0  Found rows: 0  Warnings: 0  Duration for 0 of 1 query: 0.000 sec. */
SELECT emp_name,TIMESTAMPDIFF(YEAR,doj,CURDATE()) AS 'Experience ' FROM employee GROUP BY emp_name;
/* Affected rows: 0  Found rows: 16  Warnings: 0  Duration for 1 query: 0.000 sec. */
```

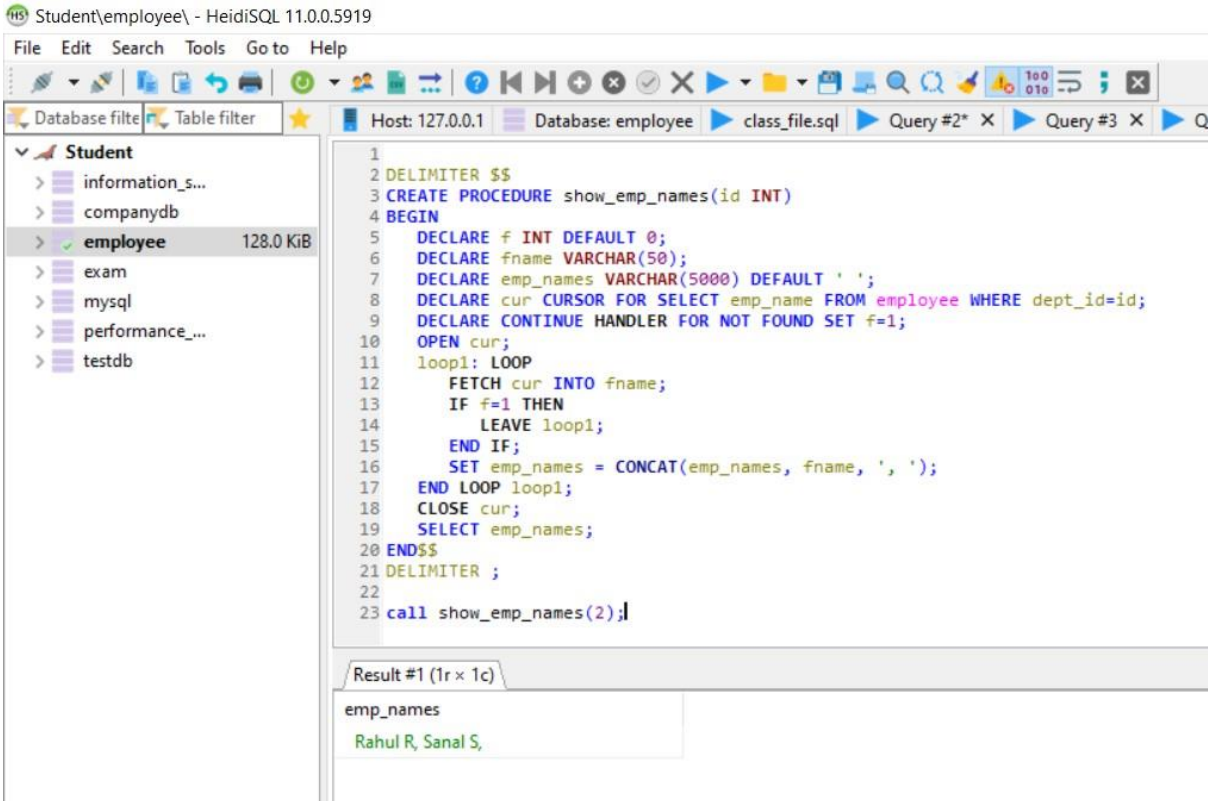
```
DELIMITER $$
CREATE FUNCTION display_sal_count(id INT )
RETURNS INT
BEGIN
	DECLARE count_of_emp INT DEFAULT 0;
	SELECT COUNT(emp_id) FROM employee WHERE sal>50000 AND dept_id =id
INTO count_of_emp;
	RETURN count_of_emp;
END $$
DELIMITER ;
```



```
DELIMITER $$
CREATE FUNCTION display_sal_count(id INT )
RETURNS INT
BEGIN
   DECLARE count_of_emp INT DEFAULT 0;
   SELECT COUNT(emp_id) FROM employee WHERE sal>50000 AND dept_id =id INTO count_of_emp;
   RETURN count_of_emp;
END $$
DELIMITER ;

SELECT display_sal_count(3);
```

```
display_sal_count(3)
                   1
```

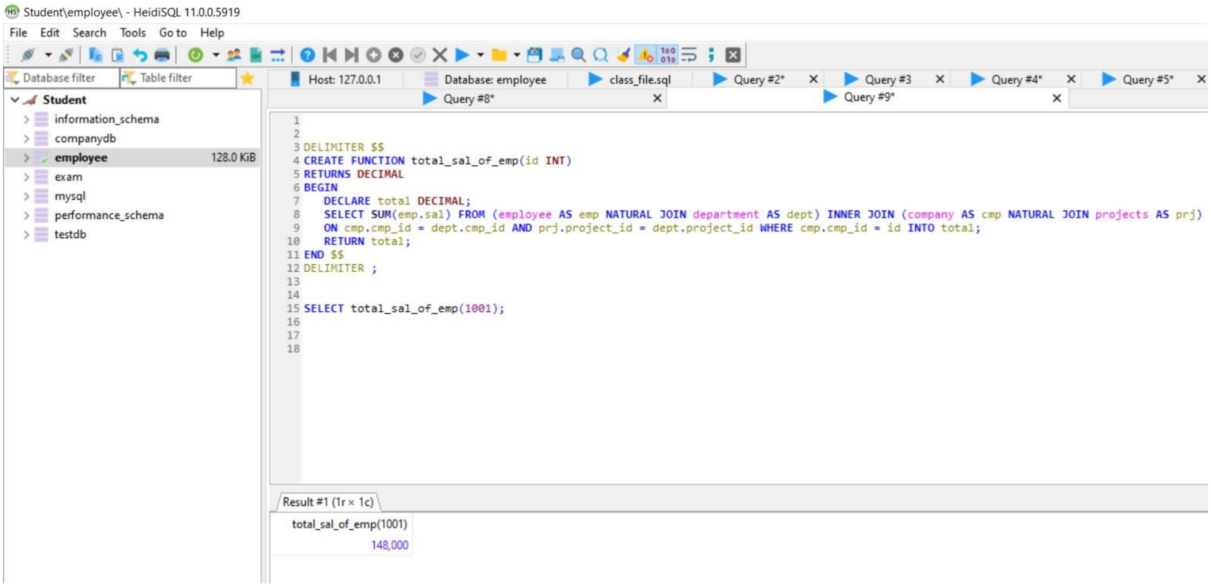
```
DELIMITER $$
CREATE PROCEDURE show_emp_names(id INT)
BEGIN
        DECLARE f INT DEFAULT 0;
        DECLARE fname VARCHAR(50);
        DECLARE emp_names VARCHAR(5000) DEFAULT ' ';
        DECLARE cur CURSOR FOR SELECT emp_name FROM employee WHERE
dept_id=id;
        DECLARE CONTINUE HANDLER FOR NOT FOUND SET f=1;
        OPEN cur;
loop1: LOOP
                FETCH cur INTO fname;
                IF f=1 THEN
                        LEAVE loop1;
                END IF;
                SET emp_names = CONCAT(emp_names, fname, ', ');
        END LOOP loop1;
        CLOSE cur;
        SELECT emp_names;
END$$
DELIMITER ;
```



```
DELIMITER $$
CREATE PROCEDURE show_emp_names(id INT)
BEGIN
    DECLARE f INT DEFAULT 0;
    DECLARE fname VARCHAR(50);
    DECLARE emp_names VARCHAR(5000) DEFAULT ' ';
    DECLARE cur CURSOR FOR SELECT emp_name FROM employee WHERE dept_id=id;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET f=1;
    OPEN cur;
    loop1: LOOP
        FETCH cur INTO fname;
        IF f=1 THEN
            LEAVE loop1;
        END IF;
        SET emp_names = CONCAT(emp_names, fname, ', ');
    END LOOP loop1;
    CLOSE cur;
    SELECT emp_names;
END$$
DELIMITER ;

call show_emp_names(2);
```

```
emp_names
Rahul R, Sanal S,
```

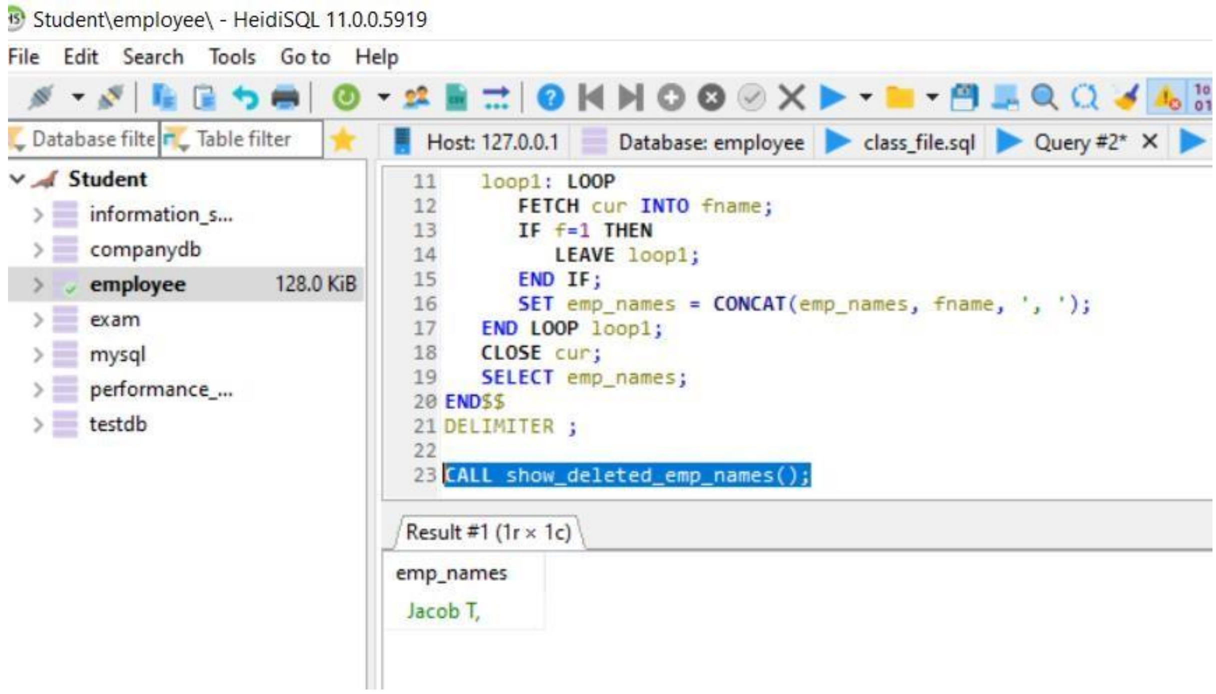
```
DELIMITER $$
CREATE FUNCTION total_sal_of_emp(id INT)
RETURNS DECIMAL
BEGIN
      DECLARE total DECIMAL;
      SELECT SUM(emp.sal) FROM (employee AS emp NATURAL JOIN department AS
dept) INNER JOIN (company AS cmp NATURAL JOIN projects AS prj)
      ON cmp.cmp_id = dept.cmp_id AND prj.project_id = dept.project_id
WHERE cmp.cmp_id = id INTO total;
      RETURN total;
END $$
DELIMITER ;
```



```


DELIMITER $$
CREATE FUNCTION total_sal_of_emp(id INT)
RETURNS DECIMAL
BEGIN
   DECLARE total DECIMAL;
   SELECT SUM(emp.sal) FROM (employee AS emp NATURAL JOIN department AS dept) INNER JOIN (company AS cmp NATURAL JOIN projects AS prj)
   ON cmp.cmp_id = dept.cmp_id AND prj.project_id = dept.project_id WHERE cmp.cmp_id = id INTO total;
   RETURN total;
END $$
DELIMITER ;


SELECT total_sal_of_emp(1001);
```

```
total_sal_of_emp(1001)
               148,000
```

```
DELIMITER $$
CREATE PROCEDURE show_deleted_emp_names()
BEGIN
       DECLARE f INT DEFAULT 0;
       DECLARE fname VARCHAR(50);
       DECLARE emp_names VARCHAR(5000) DEFAULT ' ';
       DECLARE cur CURSOR FOR SELECT e_name FROM emp_backup;
       DECLARE CONTINUE HANDLER FOR NOT FOUND SET f=1;
       OPEN cur;
loop1: LOOP
             FETCH cur INTO fname;
             IF f=1 THEN
                   LEAVE loop1;
             END IF;
             SET emp_names = CONCAT(emp_names, fname, ', ');
       END LOOP loop1;
       CLOSE cur;
       SELECT emp_names;
END$$
DELIMITER ;
```

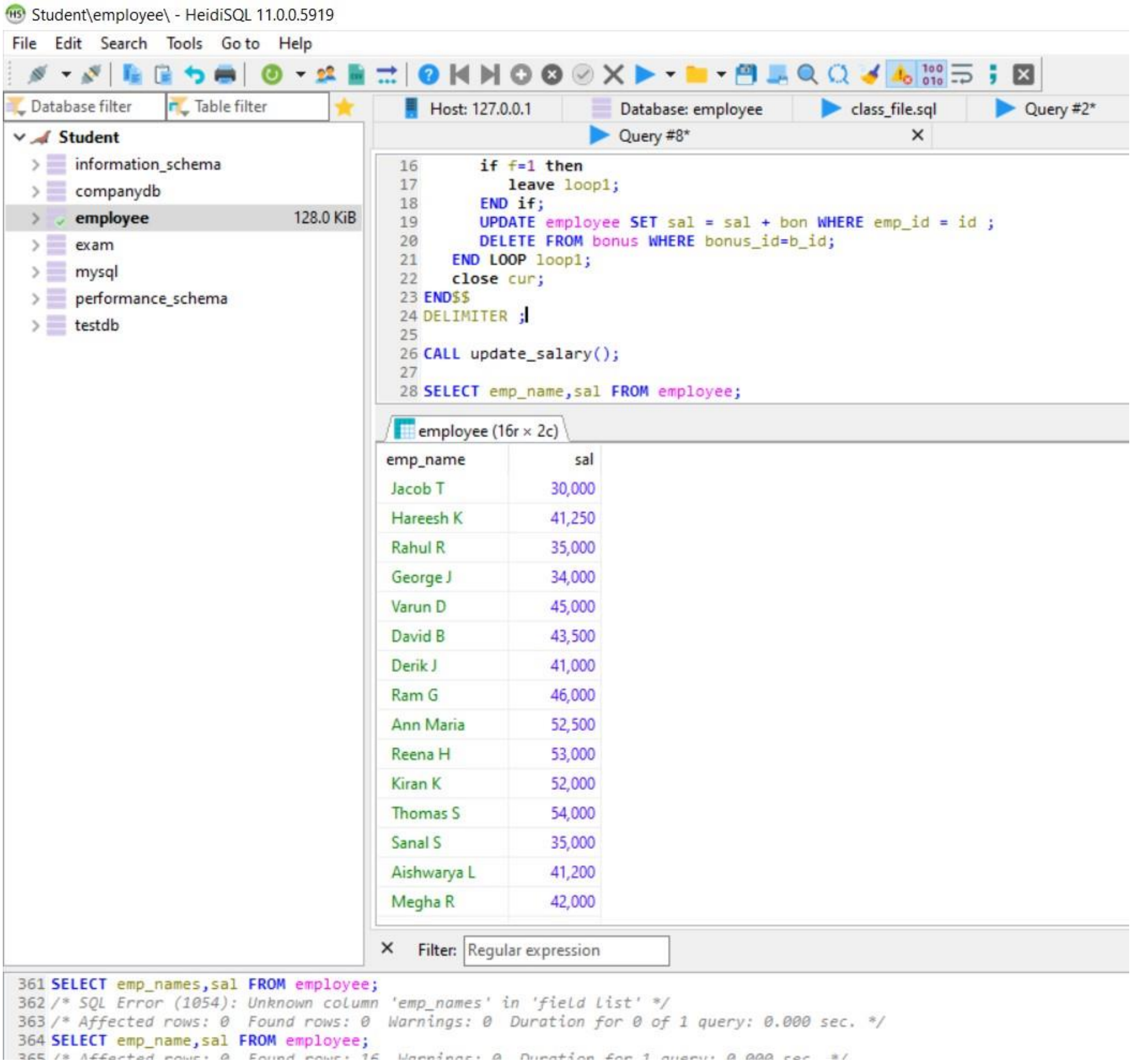


```
    loop1: LOOP
       FETCH cur INTO fname;
       IF f=1 THEN
          LEAVE loop1;
       END IF;
       SET emp_names = CONCAT(emp_names, fname, ', ');
    END LOOP loop1;
    CLOSE cur;
    SELECT emp_names;
END$$
DELIMITER ;

CALL show_deleted_emp_names();
```

emp_names
Jacob T,

```
DELIMITER $$
CREATE PROCEDURE update_salary()
BEGIN
       DECLARE f INT ;
       DECLARE id INT ;
       DECLARE bon DECIMAL;
       DECLARE b_id VARCHAR(8);
```

```
        DECLARE cur CURSOR  FOR SELECT bonus_id, emp_id,bonus_amount FROM
bonus;
        DECLARE CONTINUE handler FOR NOT FOUND SET f=1;
        OPEN cur;
        loop1 : loop            fetch
cur INTO b_id,id,bon;           if f=1
then                        leave loop1;
                END if;
                UPDATE employee SET sal = sal + bon WHERE emp_id = id ;
                DELETE FROM bonus WHERE bonus_id=b_id;
        END LOOP loop1;
        close cur;
END$$
DELIMITER ;
```



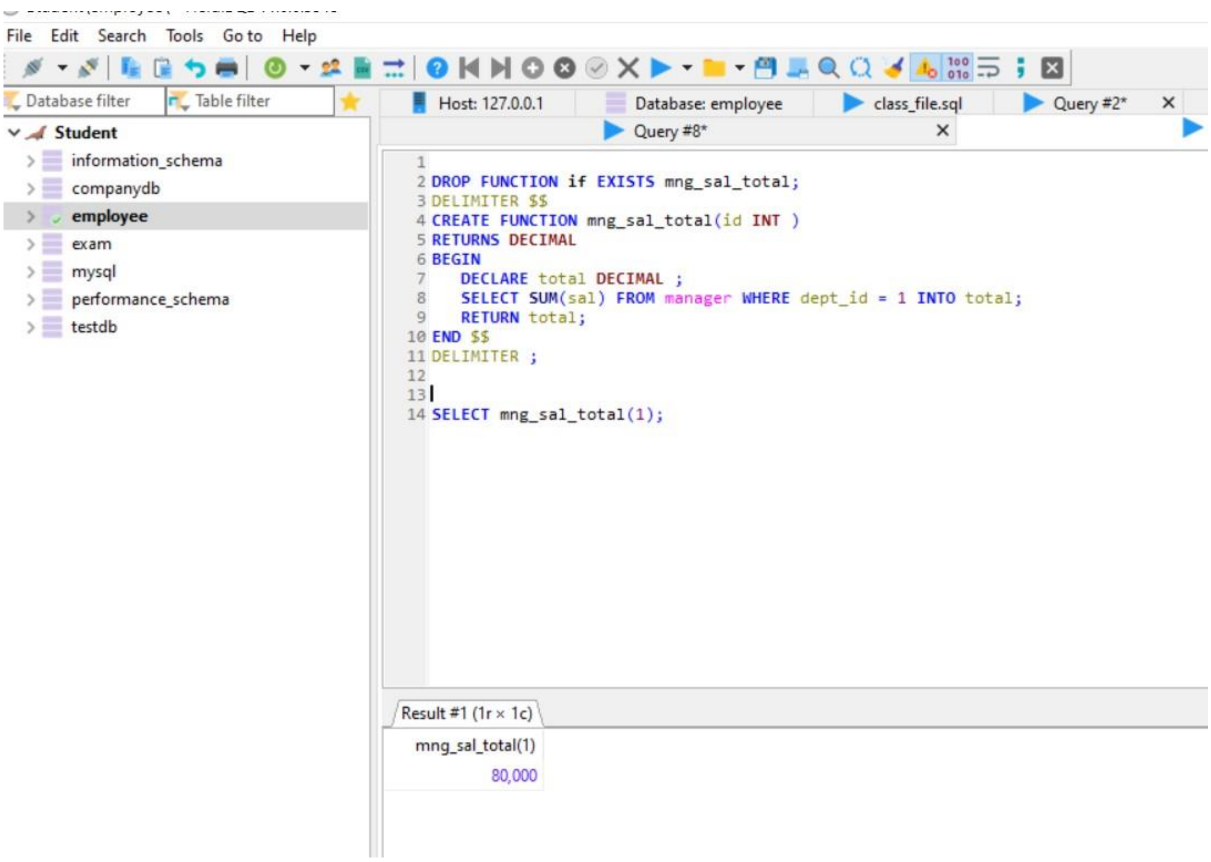
```
        if f=1 then
            leave loop1;
        END if;
        UPDATE employee SET sal = sal + bon WHERE emp_id = id ;
        DELETE FROM bonus WHERE bonus_id=b_id;
    END LOOP loop1;
    close cur;
END$$
DELIMITER ;

CALL update_salary();

SELECT emp_name,sal FROM employee;
```

emp_name	sal
Jacob T	30,000
Hareesh K	41,250
Rahul R	35,000
George J	34,000
Varun D	45,000
David B	43,500
Derik J	41,000
Ram G	46,000
Ann Maria	52,500
Reena H	53,000
Kiran K	52,000
Thomas S	54,000
Sanal S	35,000
Aishwarya L	41,200
Megha R	42,000

```
SELECT emp_names,sal FROM employee;
/* SQL Error (1054): Unknown column 'emp_names' in 'field list' */
/* Affected rows: 0  Found rows: 0  Warnings: 0  Duration for 0 of 1 query: 0.000 sec. */
SELECT emp_name,sal FROM employee;
/* Affected rows: 0  Found rows: 16  Warnings: 0  Duration for 1 query: 0.000 sec.  */
```

```
DELIMITER $$
CREATE FUNCTION mng_sal_total(id INT )
RETURNS DECIMAL
BEGIN
        DECLARE total DECIMAL ;
        SELECT SUM(m.sal) FROM (manager AS m NATURAL JOIN department AS dept)
```

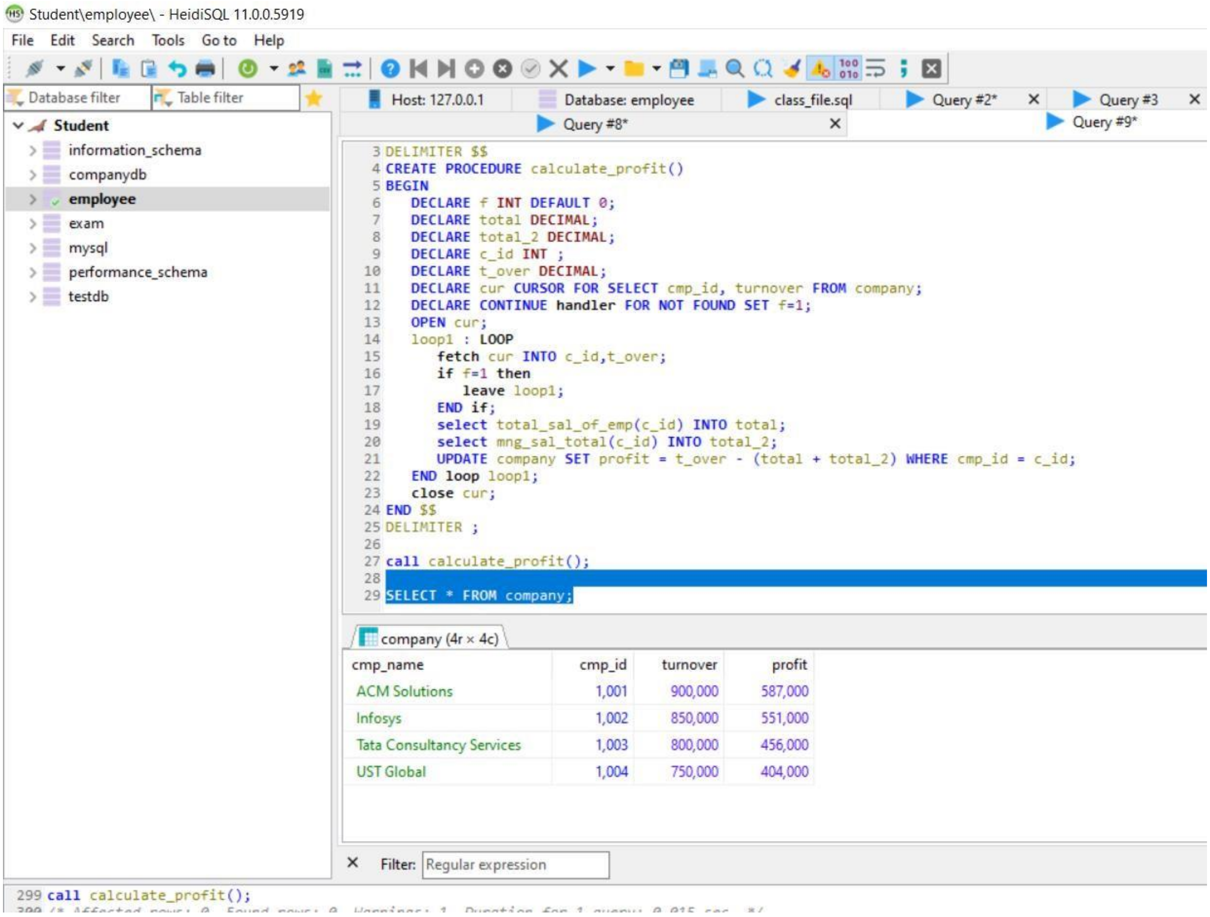
```
INNER JOIN (company AS cmp NATURAL JOIN projects AS prj)
      ON cmp.cmp_id = dept.cmp_id AND prj.project_id = dept.project_id
WHERE cmp.cmp_id = id INTO total;
      RETURN total;
END $$
DELIMITER ;
```



```

DROP FUNCTION if EXISTS mng_sal_total;
DELIMITER $$
CREATE FUNCTION mng_sal_total(id INT )
RETURNS DECIMAL
BEGIN
    DECLARE total DECIMAL ;
    SELECT SUM(sal) FROM manager WHERE dept_id = 1 INTO total;
    RETURN total;
END $$
DELIMITER ;


SELECT mng_sal_total(1);
```

```
mng_sal_total(1)
          80,000
```

```
DELIMITER $$
CREATE PROCEDURE calculate_profit()
BEGIN
      DECLARE f INT DEFAULT 0;
      DECLARE total DECIMAL;
      DECLARE total_2 DECIMAL;
      DECLARE c_id INT ;
      DECLARE t_over DECIMAL;
      DECLARE cur CURSOR FOR SELECT cmp_id, turnover FROM company;
      DECLARE CONTINUE handler FOR NOT FOUND SET f=1;
      OPEN cur;    loop1 : LOOP
fetch cur INTO c_id,t_over;           if
f=1 then                   leave loop1;
      END if;
            select total_sal_of_emp(c_id) INTO total;
select mng_sal_total(c_id) INTO total_2;
  UPDATE company SET profit = t_over - (total + total_2) WHERE cmp_id = c_id;
      END loop loop1;
      close cur;
END $$
DELIMITER ;
```



```
DELIMITER $$
CREATE PROCEDURE calculate_profit()
BEGIN
   DECLARE f INT DEFAULT 0;
   DECLARE total DECIMAL;
   DECLARE total_2 DECIMAL;
   DECLARE c_id INT ;
   DECLARE t_over DECIMAL;
   DECLARE cur CURSOR FOR SELECT cmp_id, turnover FROM company;
   DECLARE CONTINUE handler FOR NOT FOUND SET f=1;
   OPEN cur;
   loop1 : LOOP
      fetch cur INTO c_id,t_over;
      if f=1 then
         leave loop1;
      END if;
      select total_sal_of_emp(c_id) INTO total;
      select mng_sal_total(c_id) INTO total_2;
      UPDATE company SET profit = t_over - (total + total_2) WHERE cmp_id = c_id;
   END loop loop1;
   close cur;
END $$
DELIMITER ;

call calculate_profit();

SELECT * FROM company;
```

cmp_name	cmp_id	turnover	profit
ACM Solutions	1,001	900,000	587,000
Infosys	1,002	850,000	551,000
Tata Consultancy Services	1,003	800,000	456,000
UST Global	1,004	750,000	404,000

```
call calculate_profit();
/* Affected rows: 0  Found rows: 0  Warnings: 1  Duration for 1 query: 0.015 sec. */
```

Conclusion

The project is built in such a way that the database to store the details of employee, manager and also the department of certain companies. The main objective of this database is to save time, make the system cost effective and management records. The payment of employee is done after calculating the bonus for each month and after completing that purpose the bonus will be deleted from the database so it make the database more efficient in memory.