Institute of computer technology

Ganpat university

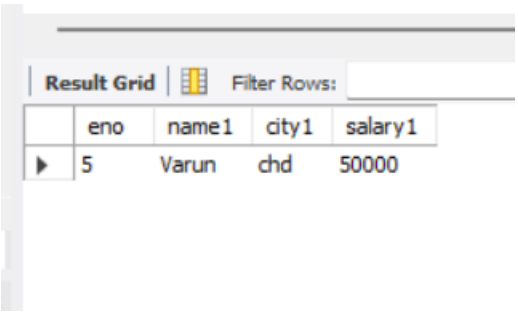
B.tech. CSE (CBA/BDA/CS/CSE)

(2CSE301) DATABASE MANAGEMENT SYSTEM

PRACTICAL 14 (Cursor)

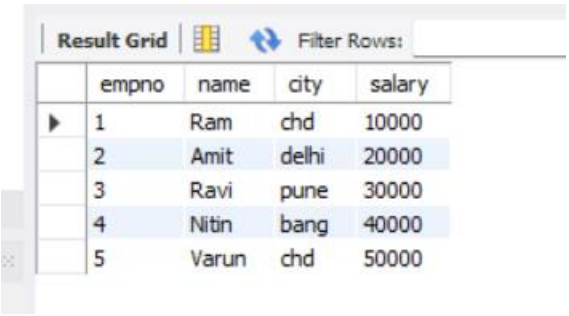
1. Write Cursor which display employee information from Employee table

```
delimiter //
create procedure cursor_proc2()
begin
        declare eno int;
  declare name text;
  declare city text;
  declare salary int;
  declare exit_loop boolean;
  declare done int default 0;
  declare c1 cursor for select eno , name, city, salary from employes;
  declare continue handler for not found set done = 1;
  open c1;
  L1: loop
                fetch c1 into eno, name,city, salary;
    if done = 1 then
                        leave L1;
                end if;
    select eno,name,city,salary;
        end loop;
  close c1;
end;
//
delimiter ;

call cursor_proc2();
```



2. Implements Cursor which reads one by one row from Employee table and insert data in EmployeeInfo table.

```
CREATE TABLE EmployeeInfo (
   empno INT,
   name TEXT,
   city TEXT,
   salary INT
);

DELIMITER //
CREATE PROCEDURE CopyEmployeeDataToEmployeeInfo()
BEGIN
   DECLARE eno INT;
   DECLARE name1 TEXT;
   DECLARE city1 TEXT;
   DECLARE salary1 INT;
   DECLARE done INT DEFAULT 0;
   DECLARE c1 CURSOR FOR SELECT eid, cname, address, salary FROM Employee;
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
   OPEN c1;
   L1: LOOP
      FETCH c1 INTO eno, name1, city1, salary1;
      IF done = 1 THEN
         LEAVE L1;
      END IF;
      INSERT INTO EmployeeInfo (empno, name, city, salary) VALUES (eno, name1, city1, salary1);
   END LOOP;

   CLOSE c1;
END;
//
DELIMITER ;

call CopyEmployeeDataToEmployeeInfo();
select * from EmployeeInfo;
```



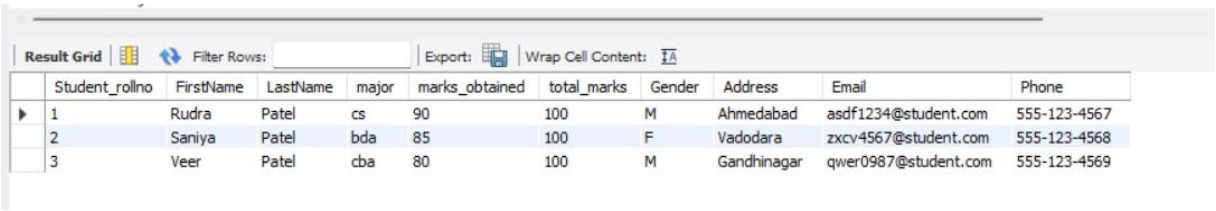
Result Grid | Filter Rows:

empno	name	city	salary
1	Ram	chd	10000
2	Amit	delhi	20000
3	Ravi	pune	30000
4	Nitin	bang	40000
5	Varun	chd	50000

3. Implements Cursor which reads one by one row from Student table and insert all BDA student in data in table BDAStudent and insert all CBA student data in table CBAStudent and insert all CS student data in table CSStudent

```
ALTER TABLE Student CHANGE COLUMN branch major VARCHAR(50);

DELIMITER //
CREATE PROCEDURE CopyStudentDataToMajorTables()
BEGIN
  DECLARE student_id INT;
  DECLARE student_name VARCHAR(255);
  DECLARE major VARCHAR(50);
  DECLARE done INT DEFAULT 0;
  DECLARE student_cursor CURSOR FOR SELECT Student_rollno, Firstname, major FROM Student;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
  OPEN student_cursor;
  L1: LOOP
    FETCH student_cursor INTO student_id, student_name, major;
    IF done = 1 THEN
      LEAVE L1;
    END IF;
    IF major = 'BDA' THEN
      INSERT INTO BDAStudent (student_id, student_name, major) VALUES (student_id,
student_name, major);
    ELSEIF major = 'CBA' THEN
      INSERT INTO CBAStudent (student_id, student_name, major) VALUES (student_id,
student_name, major);
    ELSEIF major = 'CS' THEN
      INSERT INTO CSStudent (student_id, student_name, major) VALUES (student_id,
student_name, major);
    END IF;
  END LOOP;
  CLOSE student_cursor;
END;
//
DELIMITER ;
```

```
call CopyStudentDataToMajorTables();
select * from student;
```

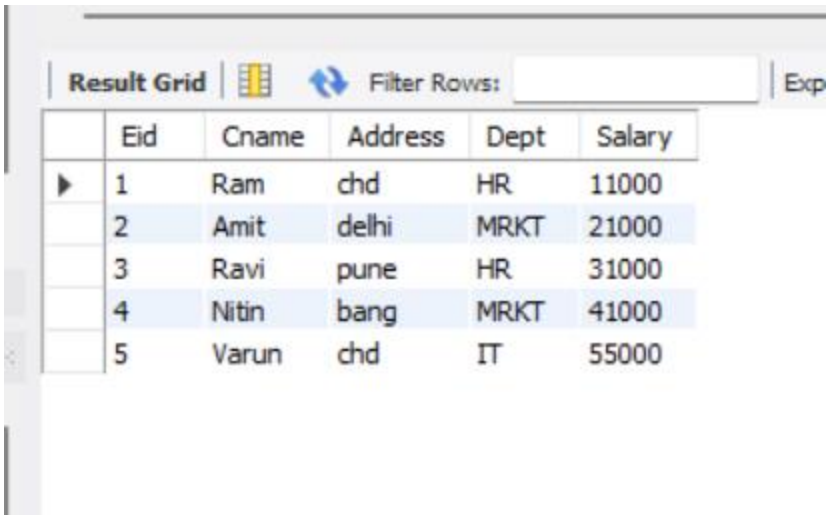


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Student_rollno	FirstName	LastName	major	marks_obtained	total_marks	Gender	Address	Email	Phone
1	Rudra	Patel	cs	90	100	M	Ahmedabad	asdf1234@student.com	555-123-4567
2	Saniya	Patel	bda	85	100	F	Vadodara	zxcv4567@student.com	555-123-4568
3	Veer	Patel	cba	80	100	M	Gandhinagar	qwer0987@student.com	555-123-4569

4. Employee Management system manage all employee information. Implements Cursor which reads data from Employee table and update salary. Increase salary by 5000 whose salary is more than 40000 else increase salary by 1000.

```
 delimiter //
create procedure UpdateEmployeeSalaries()
begin
declare empid int;
declare emp_salary int;
DECLARE exit_loop BOOLEAN default false;
declare c1 cursor for select eid,salary from employee;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET exit_loop = TRUE;
open c1;
L1:loop
fetch c1 into empid,emp_salary;
if exit_loop then
close c1;
 leave L1;
end if;
if emp_salary > 40000 then
update employee set salary = salary+4000 where eid = empid;
end if;
update employee set salary = salary+1000 where eid = empid;
end loop;
end;
//

call UpdateEmployeeSalaries();
select * from employee;
```



Result Grid | Filter Rows: | Exp

Eid	Cname	Address	Dept	Salary
1	Ram	chd	HR	11000
2	Amit	delhi	MRKT	21000
3	Ravi	pune	HR	31000
4	Nitin	bang	MRKT	41000
5	Varun	chd	IT	55000

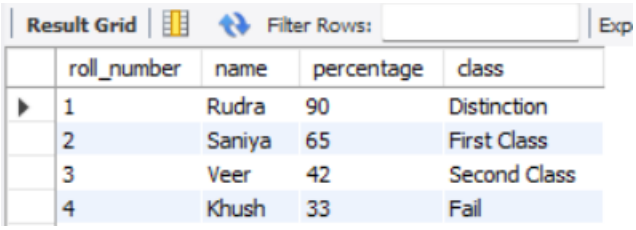
5. Implements Cursor which reads one by one row from Result table and insert row in Result1 table. Read percentage from result and based on percentage assign class to Students as per below

Percentage	**class**
100 – 70	Distinction
69 – 60	First Class
59 – 40	Second Class
< 40	Fail

```
CREATE TABLE Result (
    roll_number INT,
    name VARCHAR(255),
    percentage INT
);

-- Insert sample data into the Result table
INSERT INTO Result (roll_number, name, percentage) VALUES
(1, 'Rudra', 90),
(2, 'Saniya', 65),
(3, 'Veer', 42),
(4,'Khush',33);
-- Create the Result1 table
CREATE TABLE Result1 (
    roll_number INT,
    name VARCHAR(255),
    percentage INT,
    class VARCHAR(50)
```

```
);

DELIMITER //
CREATE PROCEDURE pr14()
BEGIN
   DECLARE done INT DEFAULT 0; -- Variable to track end of cursor

   DECLARE roll_number_val INT;
   DECLARE name_val VARCHAR(255);
   DECLARE percentage_val INT;
   DECLARE class_val VARCHAR(50);

   -- Declare cursor to fetch rows from Result table
   DECLARE cursor_result CURSOR FOR
      SELECT roll_number, name, percentage FROM Result;

   -- Declare handler for NOT FOUND condition
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

   -- Open the cursor
   OPEN cursor_result;

   -- Loop through the cursor
   read_loop: LOOP
      -- Fetch row from cursor into variables
      FETCH cursor_result INTO roll_number_val, name_val, percentage_val;

      -- Check if cursor fetch was successful
      IF done = 1 THEN
         LEAVE read_loop;
      END IF;

      -- Determine class based on percentage
      IF percentage_val >= 70 THEN
         SET class_val = 'Distinction';
      ELSEIF percentage_val >= 60 THEN
         SET class_val = 'First Class';
      ELSEIF percentage_val >= 40 THEN
         SET class_val = 'Second Class';
      ELSE
         SET class_val = 'Fail';
      END IF;

      -- Insert row into Result1 table
      INSERT INTO Result1 (roll_number, name, percentage, class)
      VALUES (roll_number_val, name_val, percentage_val, class_val);
   END LOOP;
```

```
    -- Close the cursor
    CLOSE cursor_result;
END;
//
DELIMITER ;


call pr14();
SELECT * FROM Result1;
```



Result Grid | Filter Rows: | Exp

roll_number	name	percentage	class
1	Rudra	90	Distinction
2	Saniya	65	First Class
3	Veer	42	Second Class
4	Khush	33	Fail