MySQL Stored Procedures Exercises

-GURU ABBISHEIK S

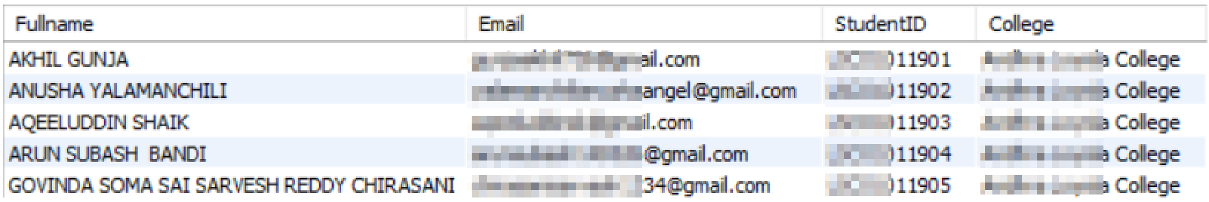
1. SIMPLE STORED PROCEDURES

```
DELIMITER $$

CREATE PROCEDURE id()

BEGIN

SELECT fullname AS Fullname, email AS Email, id AS StudentID, college AS College FROM students;

END$$

DELIMITER ;
```

#Procedure Call / Execution

```
CALL id();
```

#Output



Fullname	Email	StudentID	College
AKHIL GUNJA	[redacted]ail.com	[redacted]11901	[redacted]a College
ANUSHA YALAMANCHILI	[redacted]angel@gmail.com	[redacted]11902	[redacted]a College
AQEELUDDIN SHAIK	[redacted]il.com	[redacted]11903	[redacted]a College
ARUN SUBASH BANDI	[redacted]@gmail.com	[redacted]11904	[redacted]a College
GOVINDA SOMA SAI SARVESH REDDY CHIRASANI	[redacted]34@gmail.com	[redacted]11905	[redacted]a College

2. STORED PROCEDURES WITH USING VARIABLES

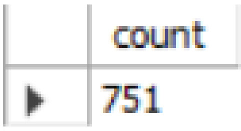
#Use this DB and drop the procedure name if already exists

```
USE `course_dedication_fortnite`;

DROP procedure IF EXISTS `course_dedication`;


DELIMITER $$

CREATE PROCEDURE `course_dedication` ()

BEGIN

DECLARE count INT DEFAULT 0;

SELECT COUNT(*) INTO count FROM ff_1;

SELECT count;

END$$
```

#Procedure Call / Execution

```
CALL course_dedication();
```

#Output



	count
▶	751

3. STORED PROCEDURES WITH PARAMETERS (1)

```
USE `course_dedication_fortnite`;

DROP procedure IF EXISTS `course_dedication_w_para`;


DELIMITER $$

CREATE PROCEDURE `course_dedication_w_para` (
```

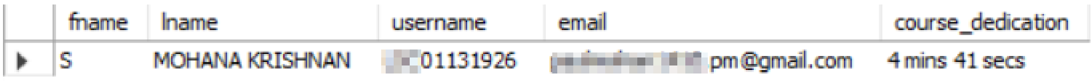
#Incoming Data with data type

```
IN user varchar(255)

)

BEGIN

SELECT * FROM ff_1 WHERE username = user;

END$$
```

#Procedure Call / Execution

```
CALL course_dedication_w_para('XXX01131926');
```

#Output



	fname	lname	username	email	course_dedication
▶	S	MOHANA KRISHNAN	[redacted]01131926	[redacted]pm@gmail.com	4 mins 41 secs

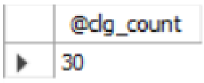
4. STORED PROCEDURES WITH PARAMETERS (2)

```
USE `student_id`;
DROP procedure IF EXISTS `college_count`;
DELIMITER $$
CREATE PROCEDURE `college_count` (
#Incoming Data with data type
IN clg varchar(255),
#Outgoing / Return Data with data type
OUT clg_count INT
)
BEGIN
SELECT COUNT(*) INTO clg_count FROM lscid WHERE college = clg;
END$$
```

#Procedure Call / Execution

```
CALL college_count('XXXXX College',@clg_count); #('IN Data', '@Return_Data')
SELECT @clg_count;
```

#Output



```
@clg_count
30
```

5. STORED PROCEDURES WITH CONDITIONAL STATEMENTS (IF / ELSE)

```
USE `workbook`;
DROP PROCEDURE IF EXISTS `bank_ifloop`;

DELIMITER $$
CREATE PROCEDURE `bank_ifloop`(
        IN customernumber INT,
  OUT customerlevel varchar(20),
  OUT customerage varchar(20)
)
BEGIN
        DECLARE credit DECIMAL(10,2) DEFAULT 0;
        DECLARE age INT DEFAULT 0;
  SELECT balance INTO credit FROM bank WHERE id = customernumber;
  SELECT age INTO age FROM bank WHERE id = customernumber;
```

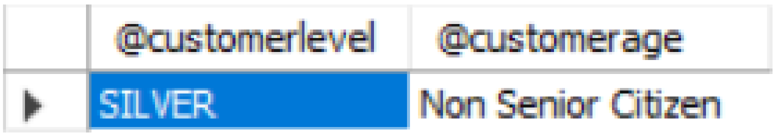
#IF, ELSE, ELSEIF Conditional Statement

```
        IF credit < 500 THEN
                SET customerlevel = 'SILVER';
        ELSEIF credit < 5000 THEN
                SET customerlevel = 'GOLD';
        ELSE
                SET customerlevel = 'PLATINUM';
        END IF;

        IF age >= 60 THEN
                SET customerage = 'Senior Citizen';
        ELSE
                SET customerage = 'Non Senior Citizen';
        END IF;
END$$
DELIMITER ;
```

#Procedure Call / Execution

```
call bank_ifloop('5', @customerlevel, @customerage);   #('IN Data',@Return1,#Return2)
SELECT @customerlevel,@customerage;
```

#Output



	@customerlevel	@customerage
▶	SILVER	Non Senior Citizen

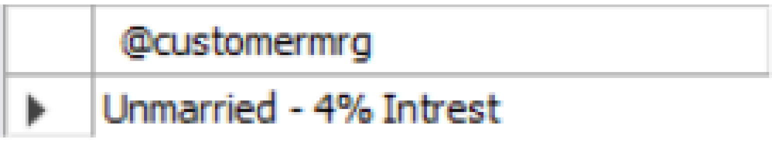
5. STORED PROCEDURES WITH CONDITIONAL STATEMENTS (CASE / SWITCH)

```
USE `workbook`;
DROP PROCEDURE IF EXISTS `bank_case`;


DELIMITER $$
CREATE PROCEDURE `bank_case`(
IN customernumber INT,
OUT customermrg varchar(100)
)
BEGIN
DECLARE mrg VARCHAR(20) DEFAULT 0;
SELECT marital INTO mrg FROM bank WHERE id = customernumber;


  CASE mrg
        WHEN 'married' THEN
                SET customermrg='Married - 8% Intrest';
        WHEN 'single' THEN
                SET customermrg='Unmarried - 4% Intrest';
        ELSE
                SET customermrg='Others - 5% Intrest';
        END CASE;
END$$
DELIMITER ;
```

#Procedure Call / Execution

```
call bank_case('88', @customermrg);
SELECT @customermrg;
```

#Output



```
   @customermrg
▶ Unmarried - 4% Intrest
```

5. STORED PROCEDURES WITH LOOPs (WHILE)

#Creating a new Table namely Calendars

```
CREATE TABLE calendars(
   id INT AUTO_INCREMENT,
   fulldate DATE UNIQUE,
   day TINYINT NOT NULL,
   month TINYINT NOT NULL,
   quarter TINYINT NOT NULL,
   year INT NOT NULL,
   PRIMARY KEY(id)
);
```

#Creating a new Procedure to Insert a Date from the another Procedure into the table we Created

```
DELIMITER $$
CREATE PROCEDURE insertcalendar(
IN dt DATE
)
BEGIN
         INSERT INTO calendars(fulldate,
     day,
     month,
     quarter,
     year) VALUES (dt,
     EXTRACT(DAY FROM dt),
     EXTRACT(MONTH FROM dt),
     EXTRACT(QUARTER FROM dt),
     EXTRACT(YEAR FROM dt)
     );
END$$
DELIMITER ;
```

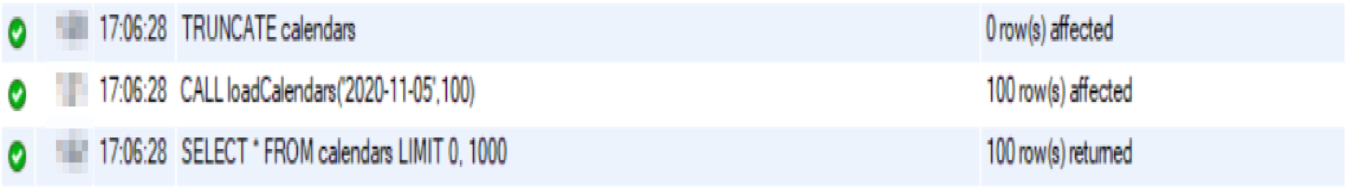
#Creating a Procedure with While Loop to Create List of Dates.

```
DELIMITER %%
CREATE PROCEDURE loadcalendars(
           IN startDate DATE,
   IN day INT
   )
BEGIN
           DECLARE counter INT DEFAULT 1;
   DECLARE dt DATE DEFAULT startDate;

   WHILE counter <= day DO
                      CALL insertcalendar(dt);
      SET counter = counter + 1;
      set dt = DATE_ADD(dt,INTERVAL 1 day);
           END WHILE;

END%%

DELIMITER ;
```

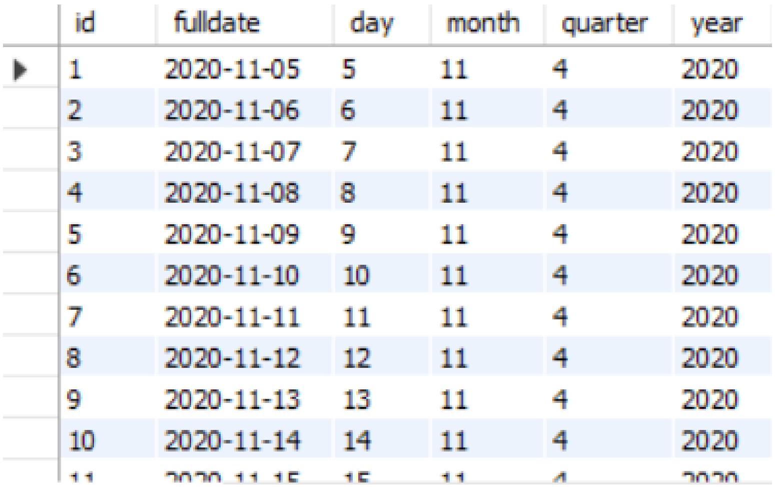
#Procedure Call / Execution

```
TRUNCATE calendars;
CALL loadCalendars('2020-11-05',100);
SELECT * FROM calendars;
```

#Output



```
✓  17:06:28  TRUNCATE calendars                          0 row(s) affected
✓  17:06:28  CALL loadCalendars('2020-11-05',100)        100 row(s) affected
✓  17:06:28  SELECT * FROM calendars LIMIT 0, 1000       100 row(s) returned
```



id	fulldate	day	month	quarter	year
1	2020-11-05	5	11	4	2020
2	2020-11-06	6	11	4	2020
3	2020-11-07	7	11	4	2020
4	2020-11-08	8	11	4	2020
5	2020-11-09	9	11	4	2020
6	2020-11-10	10	11	4	2020
7	2020-11-11	11	11	4	2020
8	2020-11-12	12	11	4	2020
9	2020-11-13	13	11	4	2020
10	2020-11-14	14	11	4	2020
11	2020-11-15	15	11	4	2020

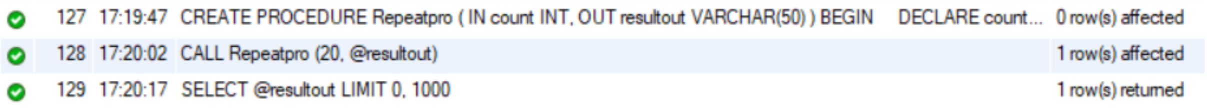
5. STORED PROCEDURES WITH LOOPs (REPEAT)

```
DELIMITER $$
CREATE PROCEDURE Repeatpro (
IN count INT,
OUT resultout VARCHAR(50)
)
BEGIN
  DECLARE counter INT DEFAULT 1;
  DECLARE result VARCHAR(100) DEFAULT '';

  REPEAT
    SET result = CONCAT(result,counter,',');
    SET counter = counter + 1;
  UNTIL counter >= count
  END REPEAT;

   SELECT result INTO resultout;
END$$
DELIMITER ;
```

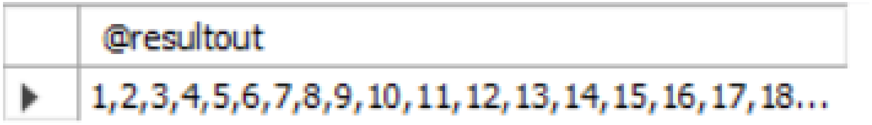
#Procedure Call / Execution

```
CALL Repeatpro (20, @resultout);

SELECT @resultout;
```

#Output



```
✓ 127 17:19:47 CREATE PROCEDURE Repeatpro ( IN count INT, OUT resultout VARCHAR(50) ) BEGIN    DECLARE count... 0 row(s) affected
✓ 128 17:20:02 CALL Repeatpro (20, @resultout)                                                                  1 row(s) affected
✓ 129 17:20:17 SELECT @resultout LIMIT 0, 1000                                                                  1 row(s) returned
```



	@resultout
▶	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18...