

Institute of Computer Technology
B. Tech. Computer Science and Engineering

Semester: III

Sub: Database Management System

Course Code: 2CSE301

Practical Number:9

Objective:

MySQL Stored Procedures

Queries:

1. Create a stored procedure that prints the following statement: hi, your name is your_name and your enrolment number is enrolment_number.

Code :

```
SELECT
CONCAT(
'Hi, Your Name is ',
'jagir shastri ',
'and your Enrollment Number is ',
'19162171014.'
) AS 'Line';
```

```
CALL name();
```

2. Create a stored procedure that displays the full name (e.g. PENELOPE GUINNESS) of the first 10 actors.

Code :

```
SELECT
CONCAT(a.first_name, " ", a.last_name)AS `Full_Name`
FROM actor a ;
```

```
Call fullname();
```

3. Create a stored procedure that displays all the details for the given film title.

Code :

```
SELECT *  
FROM  
    film ;
```

```
CALL Details();
```

4. Create a stored procedure that displays all the details of the film whose name contains the given string in its name.

Code :

```
SELECT *  
FROM  
    film where title like('AC%');  
Call id();
```

5. Create a stored procedure that displays all the films for the given category_id.

Code :

```
SELECT title FROM film,category,film_category WHERE  
category.category_id = ID AND  
category.category_id=film_category.category_id AND  
film_category.film_id=film.film_id;  
CALL d_films(50);
```

6. Create a stored procedure that displays all the films for the given category name.

Code :

```
SELECT title FROM film,film_category,category  
WHERE ategory.category_id=film_category.category_id AND  
film.film_id=film_category.film_id
```

```
AND category.name=cname;  
CALL c_name('Horror');
```

- 7. Create a procedure to display details of all the films whose rental_rate falls between specified range.**

Code :

```
CREATE VIEW rent_rate AS  
SELECT * FROM film  
WHERE film.rent_rate > 1  
AND film.rent_rate <= 3.99;  
SELECT * FROM rent_rate;
```

- 8. Create a stored procedure that displays the full name (e.g. PENELOPE GUINNESS) of the actor for given actor id.**

Code :

```
SELECT concat  
(actor.first_name,"",actor.last_name)as fullname FROM actor  
WHERE actor.actor_id > 5 AND actor.actor_id< 50;  
  
call full_name();
```

- 9. Create a procedure to display total number of films for the given category.**

Code :

```
SELECT `category_id`,count(`film_id`) FROM film_category  
WHERE `category_id`=a;  
  
CALL film_count(8);
```

- 10. Create a procedure to display the total number of cities for the countries in which number of cities are more than the given number.**

Code :

```
select country.country, count(`city`) FROM city,country  
WHERE country.country_id=city.country_id  
GROUP BY country  
HAVING COUNT(`city`)>a;  
  
call Country_count(4);
```

- 11. Create a stored procedure to Display total number of customers as per the active status given as an input.**

Code :

```
SELECT COUNT(*) FROM customer WHERE active = stat;
```

```
CALL total_customer(1);
```

- 12. Create a stored procedure to update an actor's first name with given name and actor id. Also display the updated name in the same procedure.**

Code :

```
UPDATE actor SET actor.first_name=fn,
```

```
actor.last_name=ln
```

```
WHERE actor.actor_id=ID;
```

```
SELECT * FROM actor WHERE actor.actor_id = id;
```

```
CALL u_name("jagir","shastri",1);
```

- 13. Display rental duration for the given film id (both film id and rental duration should be accessible from a single variable)**

Code :

```
SELECT film.rental_duration FROM film WHERE film.film_id=ID;
```

```
CALL r_duration(19);
```