**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();

1. **Create a stored procedure that displays the full name (e.g. PENELOPE GUINESS) of the first 10 actors.**

**Code :**

SELECT

CONCAT(a.first\_name, " ", a.last\_name)AS `Full\_Name`

FROM actor a ;

Call fullname();

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

**Code :**

**SELECT \***

**FROM**

**film ;**

**CALL Details();**

1. **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();

1. **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);

1. **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');

1. **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;

1. **Create a stored procedure that displays the full name (e.g. PENELOPE GUINESS) 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();

1. **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);

1. **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);

1. **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);

1. **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);

1. **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);