Name: Ayush Patel   Enrolment: 22162171038   Class b batch 35

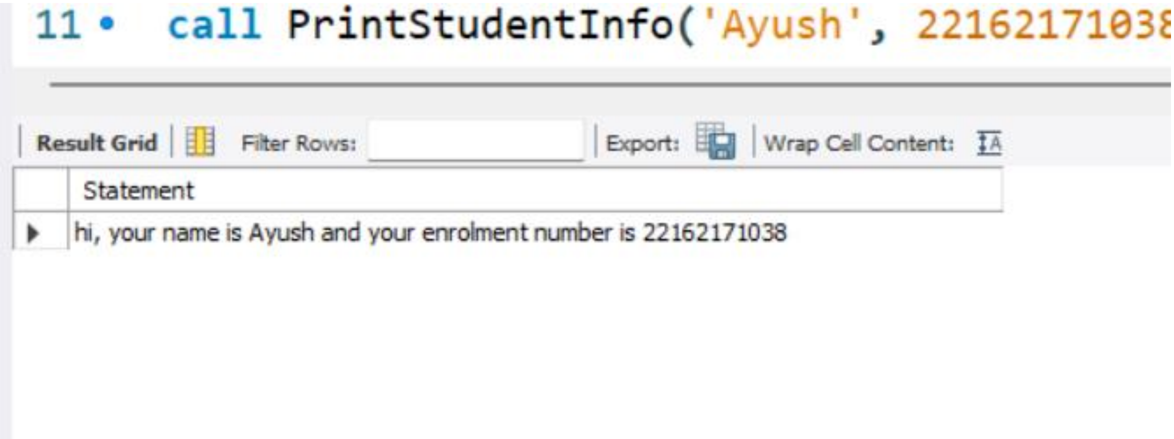
Institute of Computer Technology
Ganpat University
(2CSE301) DATABASE MANAGEMENT SYSTEM

Practical 10 MySQL Stored Procedures

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

```
DELIMITER //
create procedure PrintStudentInfo (in your_name varchar(255), in
enrolment_number bigint )
begin
   select CONCAT('hi, your name is ', your_name, ' and your enrolment number is
', enrolment_number) as Statement;
end//
DELIMITER ;

call PrintStudentInfo('Ayush', 22162171038);
```



```
call PrintStudentInfo('Ayush', 22162171038
```

Statement
hi, your name is Ayush and your enrolment number is 22162171038

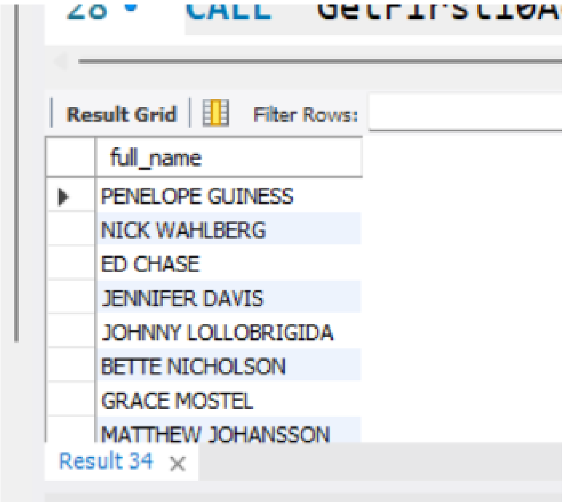
2. Create a stored procedure that displays the full name (e.g. PENELOPE GUINESS) of the first 10 actors.

```
DELIMITER //

CREATE PROCEDURE GetFirst10ActorNames()
BEGIN
   SELECT CONCAT(first_name, ' ', last_name) AS full_name
   FROM actor
```

```
    LIMIT 10;
    -- WHERE actor_id BETWEEN 1 AND 10;
END //

DELIMITER ;
CALL GetFirst10ActorNames();
```



```
CALL GetFirst10A
```

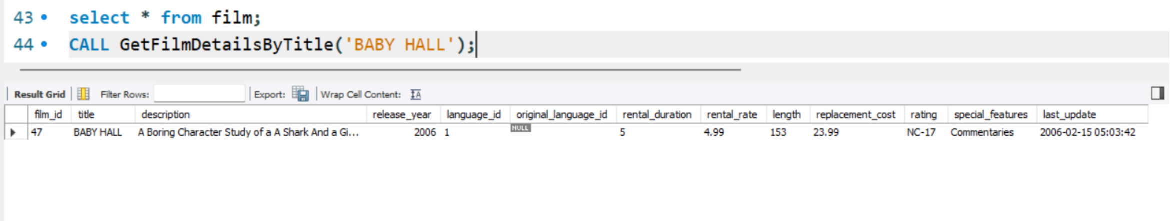
full_name
PENELOPE GUINESS
NICK WAHLBERG
ED CHASE
JENNIFER DAVIS
JOHNNY LOLLOBRIGIDA
BETTE NICHOLSON
GRACE MOSTEL
MATTHEW JOHANSSON

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

```
DELIMITER //

CREATE PROCEDURE GetFilmDetailsByTitle(IN film_title VARCHAR(255))
BEGIN
    SELECT *
    FROM sakila.film
    WHERE title = film_title;
END//

DELIMITER ;
```



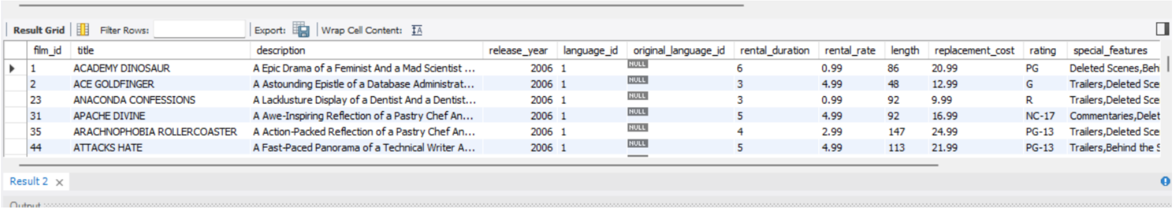
```
select * from film;
CALL GetFilmDetailsByTitle('BABY HALL');
```

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features	last_update
47	BABY HALL	A Boring Character Study of a A Shark And a Gi...	2006	1	NULL	5	4.99	153	23.99	NC-17	Commentaries	2006-02-15 05:03:42

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

```
delimiter $$
CREATE procedure Q4(IN A VARCHAR(20))
BEGIN
```

```
SELECT * FROM film WHERE title LIKE (CONCAT('%',A,'%'));
END;
$$
call Q4("AC");
```



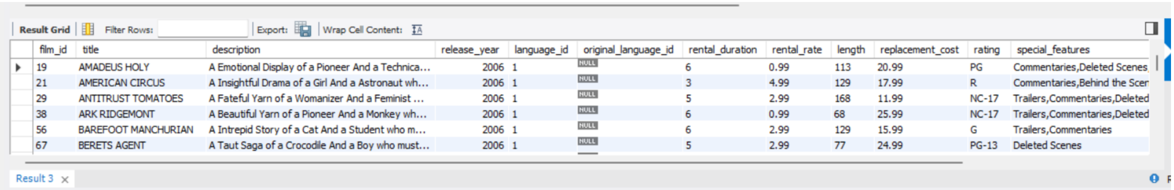
film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features
1	ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Mad Scientist ...	2006	1	NULL	6	0.99	86	20.99	PG	Deleted Scenes,Beh
2	ACE GOLDFINGER	A Astounding Epistle of a Database Administrat...	2006	1	NULL	3	4.99	48	12.99	G	Trailers,Deleted Sce
23	ANACONDA CONFESSIONS	A Lackluster Display of a Dentist And a Dentist...	2006	1	NULL	3	0.99	92	9.99	R	Trailers,Deleted Sce
31	APACHE DIVINE	A Awe-Inspiring Reflection of a Pastry Chef An...	2006	1	NULL	5	4.99	92	16.99	NC-17	Commentaries,Delet
35	ARACHNOPHOBIA ROLLERCOASTER	A Action-Packed Reflection of a Pastry Chef An...	2006	1	NULL	4	2.99	147	24.99	PG-13	Trailers,Deleted Sce
44	ATTACKS HATE	A Fast-Paced Panorama of a Technical Writer A...	2006	1	NULL	5	4.99	113	21.99	PG-13	Trailers,Behind the !

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

```
delimiter $$

create procedure Q5(IN A numeric)
BEGIN
SELECT * FROM film where film_id IN (select film_id from film_category where
    category_id = A);
END;
$$

call Q5('1');
```

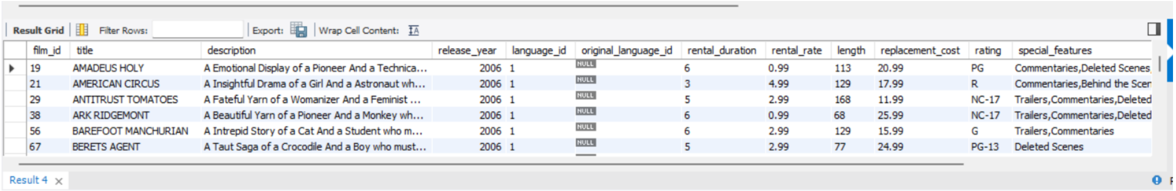


film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features
19	AMADEUS HOLY	A Emotional Display of a Pioneer And a Technica...	2006	1	NULL	6	0.99	113	20.99	PG	Commentaries,Deleted Scenes,
21	AMERICAN CIRCUS	A Insightful Drama of a Girl And a Astronaut wh...	2006	1	NULL	3	4.99	129	17.99	R	Commentaries,Behind the Scer
29	ANTITRUST TOMATOES	A Fateful Yarn of a Womanizer And a Feminist ...	2006	1	NULL	5	2.99	168	11.99	NC-17	Trailers,Commentaries,Deleted
38	ARK RIDGEMONT	A Beautiful Yarn of a Pioneer And a Monkey wh...	2006	1	NULL	6	0.99	68	25.99	NC-17	Trailers,Commentaries,Deleted
56	BAREFOOT MANCHURIAN	A Intrepid Story of a Cat And a Student who m...	2006	1	NULL	6	2.99	129	15.99	G	Trailers,Commentaries
67	BERETS AGENT	A Taut Saga of a Crocodile And a Boy who must...	2006	1	NULL	5	2.99	77	24.99	PG-13	Deleted Scenes

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

```
delimiter $$
CREATE procedure Q6(IN A VARCHAR(15))
BEGIN
SELECT * FROM film where film_id IN (select film_id from film_category where
    category_id IN (SELECT category_id FROM category WHERE name = A));
END;
$$

call Q6("Action");
```

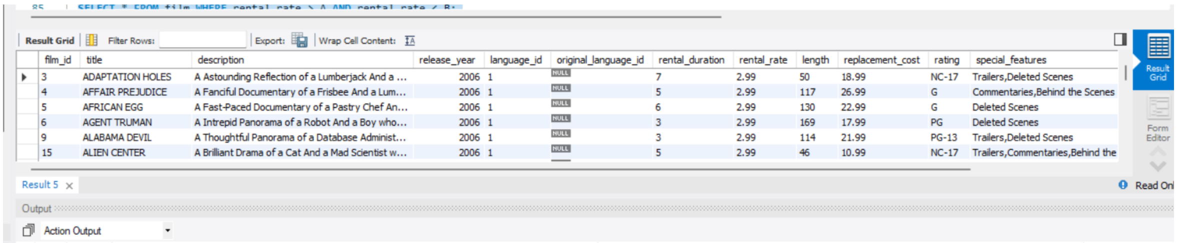


	film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features
▶	19	AMADEUS HOLY	A Emotional Display of a Pioneer And a Technica...	2006	1	NULL	6	0.99	113	20.99	PG	Commentaries,Deleted Scenes
	21	AMERICAN CIRCUS	A Insightful Drama of a Girl And a Astronaut wh...	2006	1	NULL	3	4.99	129	17.99	R	Commentaries,Behind the Scen
	29	ANTITRUST TOMATOES	A Fateful Yarn of a Womanizer And a Feminist ...	2006	1	NULL	5	2.99	168	11.99	NC-17	Trailers,Commentaries,Deleted
	38	ARK RIDGEMONT	A Beautiful Yarn of a Pioneer And a Monkey wh...	2006	1	NULL	6	0.99	68	25.99	NC-17	Trailers,Commentaries,Deleted
	56	BAREFOOT MANCHURIAN	A Intrepid Story of a Cat And a Student who m...	2006	1	NULL	6	2.99	129	15.99	G	Trailers,Commentaries
	67	BERETS AGENT	A Taut Saga of a Crocodile And a Boy who must...	2006	1	NULL	5	2.99	77	24.99	PG-13	Deleted Scenes

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

```
delimiter $$
create procedure Q7(IN A numeric(10,2),IN B numeric(10,2))
BEGIN
SELECT * FROM film WHERE rental_rate > A AND rental_rate < B;
END;
$$

call Q7(1.99,3.99);
```



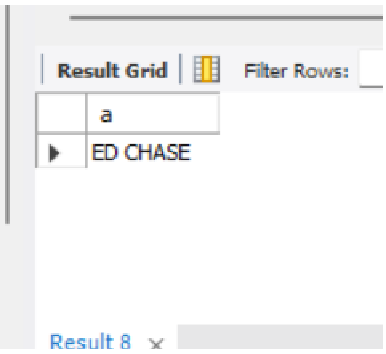
```
SELECT * FROM film WHERE rental_rate > A AND rental_rate < B;
```

	film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features
▶	3	ADAPTATION HOLES	A Astounding Reflection of a Lumberjack And a ...	2006	1	NULL	7	2.99	50	18.99	NC-17	Trailers,Deleted Scenes
	4	AFFAIR PREJUDICE	A Fanciful Documentary of a Frisbee And a Lum...	2006	1	NULL	5	2.99	117	26.99	G	Commentaries,Behind the Scenes
	5	AFRICAN EGG	A Fast-Paced Documentary of a Pastry Chef An...	2006	1	NULL	6	2.99	130	22.99	G	Deleted Scenes
	6	AGENT TRUMAN	A Intrepid Panorama of a Robot And a Boy who...	2006	1	NULL	3	2.99	169	17.99	PG	Deleted Scenes
	9	ALABAMA DEVIL	A Thoughtful Panorama of a Database Administ...	2006	1	NULL	3	2.99	114	21.99	PG-13	Trailers,Deleted Scenes
	15	ALIEN CENTER	A Brilliant Drama of a Cat And a Mad Scientist w...	2006	1	NULL	5	2.99	46	10.99	NC-17	Trailers,Commentaries,Behind the

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

```
delimiter $$
create procedure Q8 (IN A numeric)
BEGIN
SELECT CONCAT(first_name, " " , last_name) AS a FROM actor WHERE actor_id = A;
END;
$$

call Q8(3);
```

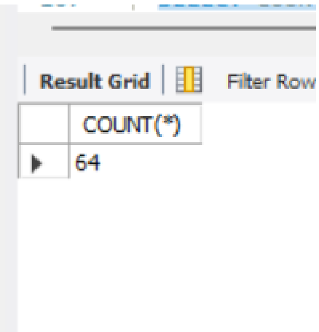


	a
▶	ED CHASE

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

```
delimiter $$
CREATE procedure Q9(IN A VARCHAR(15))
BEGIN
SELECT COUNT(*) FROM film where film_id IN (select film_id from film_category
    where category_id IN (SELECT category_id FROM category WHERE name = A));
END;
$$

call Q9("Action");
```

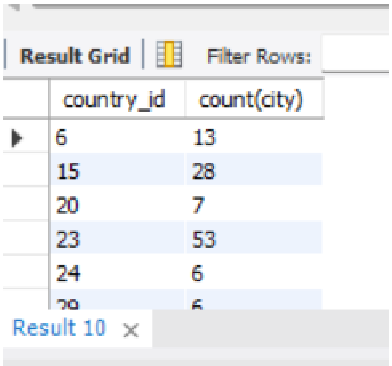


COUNT(*)
64

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.

```
delimiter $$
CREATE procedure Q10(IN A numeric)
BEGIN
select country_id,count(city) from city group by country_id HAVING count(city)>A;
END;
$$

CALL Q10(5);
```

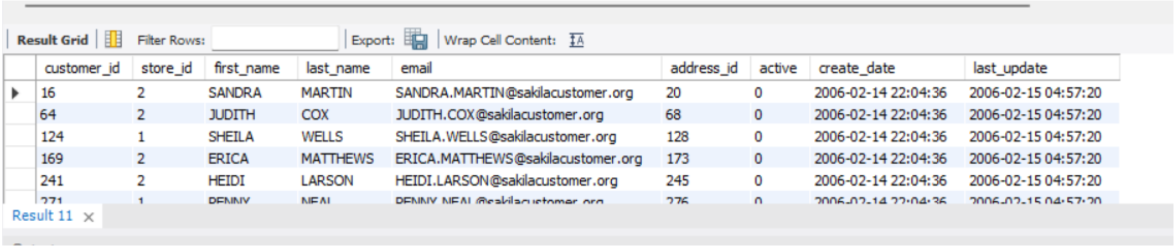


country_id	count(city)
6	13
15	28
20	7
23	53
24	6
29	6

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

```
delimiter $$
CREATE procedure Q11(IN A numeric)
BEGIN
SELECT * FROM customer WHERE active=A;
END;
$$
```

```
call Q11(0);
```

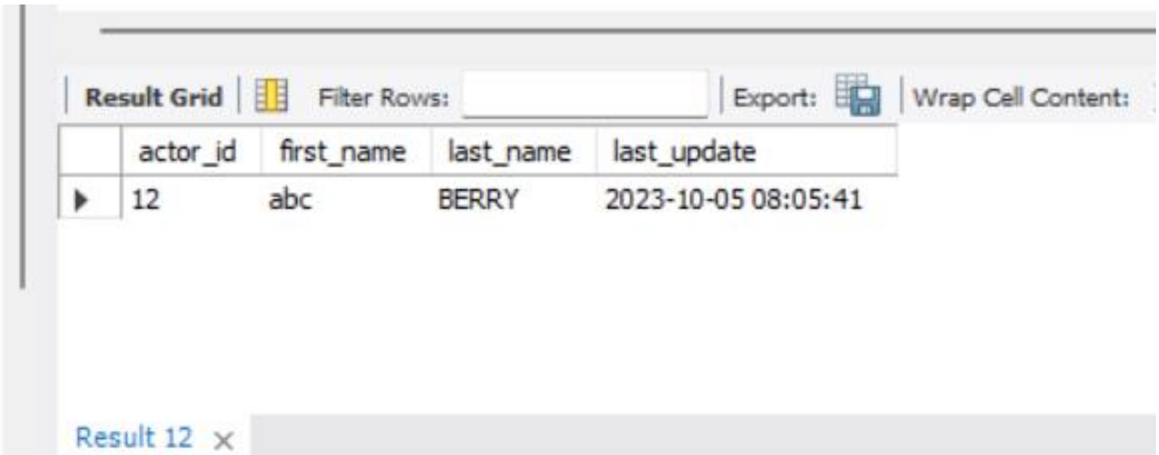


	customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
▶	16	2	SANDRA	MARTIN	SANDRA.MARTIN@sakilacustomer.org	20	0	2006-02-14 22:04:36	2006-02-15 04:57:20
	64	2	JUDITH	COX	JUDITH.COX@sakilacustomer.org	68	0	2006-02-14 22:04:36	2006-02-15 04:57:20
	124	1	SHEILA	WELLS	SHEILA.WELLS@sakilacustomer.org	128	0	2006-02-14 22:04:36	2006-02-15 04:57:20
	169	2	ERICA	MATTHEWS	ERICA.MATTHEWS@sakilacustomer.org	173	0	2006-02-14 22:04:36	2006-02-15 04:57:20
	241	2	HEIDI	LARSON	HEIDI.LARSON@sakilacustomer.org	245	0	2006-02-14 22:04:36	2006-02-15 04:57:20
	271	1	PENNY	NEAL	PENNY.NEAL@sakilacustomer.org	276	0	2006-02-14 22:04:36	2006-02-15 04:57:20

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.

```
delimiter $$
CREATE procedure Q12(IN A NUMERIC,IN B VARCHAR(15))
BEGIN
update actor set first_name = B where actor_id = A;
SELECT * FROM actor WHERE actor_id = A;
END;
$$
```

```
call Q12(12,"abc");
```

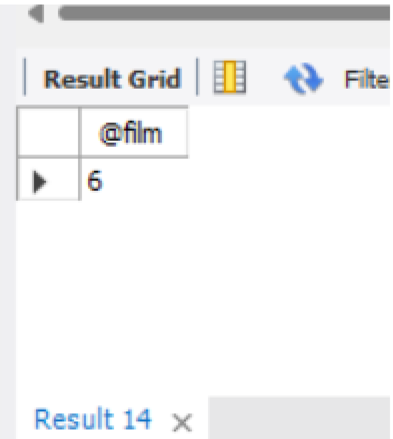


	actor_id	first_name	last_name	last_update
▶	12	abc	BERRY	2023-10-05 08:05:41

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

```
delimiter $$
create procedure Q13(INOUT A NUMERIC)
BEGIN
SELECT rental_duration into A FROM film where film_id = A;
END;
$$

set @film = 11;
call Q13(@film);
select @film;
```



@film
6