1.

DELIMITER $$

CREATE PROCEDURE displayFilmInfo (IN p\_category\_id INT, IN p\_language\_id INT)

BEGIN

IF(p\_category\_id = 0 AND p\_language\_id != 0) THEN

SELECT \*FROM language

WHERE language\_id = p\_language\_id;

ELSEIF(p\_language\_id = 0 AND p\_category\_id = 0) THEN

# khong co space

SELECT \*FROM category

WHERE category\_id = p\_category\_id;

ELSE

SELECT \*FROM category, language1

WHERE category\_id = p\_category\_id AND language\_id = p\_language\_id;

END IF;

END$$

//

2.

DELIMITER $$

DROP PROCERDURE sumInventory $$

CREATE PROCEDURE sumInventory(in storeid INT, in month VARCHAR(2), in year VARCHAR(4))

BEGIN

SELECT a.address\_id, a.address, COUNT(\*) as rented FROM store s

JOIN address a ON s.address\_id = a.address\_id

JOIN inventory I ON s.store\_id = i.store\_id

JOIN rental r ON i.inventory\_id = r.inventory\_id

WHERE MONTH(r.rental\_date) = month AND year(r.rental\_date) = year AND s.store\_id = storeid;

END $$

DELIMITER;

//FUNCTION

DELIMITER $$

CREATE FUNCTION getTotal(storeid INT, month INT, year INT) RETURNS INT

BEGIN

DECLARE sumTotal INT;

SELECT COUNT(\*) INTO sumTotal FROM rental r

JOIN inventory i ON r.inventory\_id = i.inventory\_id

JOIN store s ON i.store\_id = s.store\_id

WHERE MONTH(r.rental\_date) = month AND year(r.rental\_date) = year AND s.store\_id = storeid;

RETURN sumTotal;

END $$

3.

DELIMITER $$

DROP PROCEDURE IF EXISTS rental\_film $$

CREATE PROCEDURE rental\_film()

BEGIN

SELECT c.customer\_id, c.first\_name, c.last\_name FROM rental r

JOIN customer c ON r.customer\_id = c.customer\_id

WHERE DATEDIFF(r.last\_update, r.rental\_date) >= 30 AND r.return\_date is null ;

END$$

DELIMITER;

Call rental\_film();

4.

DELIMITER $$

DROP PROCEDURE IF EXISTS upadate\_price$$

CREATE PROCEDURE update\_price(IN p double, IN n INT)

BEGIN

DECLARE cost double;

DECLARE finish INT DEFAULT 0;

DECLARE point\_cursor CURSOR

FOR SELECT i.film\_id, f.replacement\_cost, COUNT(\*) as total FROM inventory i

JOIN film f ON i.film\_id = f.film\_id

GROUP BY film\_id

ORDER BY total ASC

LIMIT n;

OPEN point\_cursor;

get\_cost: LOOP

FETCH point\_cursor INTO cost;

SET cost = cost - cost\*p/100;

END LOOP get\_cost;

CLOSE point\_cursor;

END$$

DELIMITER;