

-- Limit the number of catalogues user can borrow to 10 for books, magazines, and prints; and 5 for videos and musics.

CREATE TRIGGER before_insert_reservation_checkNumberOfBorrowedCatalogues

BEFORE INSERT ON Reservation

FOR EACH ROW

BEGIN

DECLARE totalCountBooks INT;

DECLARE totalCountVideosMusics INT;

SELECT COUNT(c.mediaTypeID) INTO totalCountBooks

FROM Reservation r

INNER JOIN Catalogue c

ON r.catalogueNumber = c.catalogueNumber

INNER JOIN MediaType m

ON c.mediaTypeID = m.mediaTypeID

WHERE userID = NEW.userID AND c.mediaTypeID IN (1,2,5)

GROUP BY userID;

SELECT COUNT(c.mediaTypeID) INTO totalCountVideosMusics

FROM Reservation r

INNER JOIN Catalogue c

ON r.catalogueNumber = c.catalogueNumber

INNER JOIN MediaType m

ON c.mediaTypeID = m.mediaTypeID

WHERE userID = NEW.userID AND c.mediaTypeID IN (3,4)

GROUP BY userID;

IF (totalCountVideosMusics >= 5 OR totalCountBooks >= 10) THEN

-- maximum number of documents

SIGNAL SQLSTATE '50002' SET MESSAGE_TEXT = 'MAXIMUM NUMBER OF ITEMS REACHED'; -
- EVERYTHING STOPS

END IF;

END //

-- View to show complete information of a catalogue

CREATE VIEW searchCatalogue AS

```
SELECT c.catalogueNumber, c.title, CONCAT(a.firstName, ' ', a.lastName) AS Author,
       m.mediaTypeName, c.edition, p.publisherName, c.ISBN, c.format, c.copiesAvailable,
       c.shelvingInfo, c.genre, c.description, c.ageGroup
FROM Catalogue c
INNER JOIN CatalogueAuthor ca
ON c.catalogueNumber = ca.catalogueNumber
INNER JOIN Author a
ON ca.authorID = a.authorID
INNER JOIN Publisher p
ON c.publisherID = p.publisherID
INNER JOIN MediaType m
on c.mediaTypeID = m.mediaTypeID;
```

-- Find a specific catalogue's ID by it's title

```
CREATE FUNCTION fnFindCatalogueNumberByTitle(catalogueName VARCHAR(100))
RETURNS int DETERMINISTIC
BEGIN
    DECLARE catalogueID int;
    SELECT catalogueNumber INTO catalogueID
    FROM Catalogue
    WHERE title like CONCAT('%',catalogueName,'%')
    LIMIT 1;
    RETURN catalogueID;
END//
```

-- Find a specific catalogue's complete information by it's title

```
CREATE PROCEDURE spFindCatalogueInfo(searchTitle VARCHAR(100))
BEGIN
```

```
SELECT c.title, CONCAT(a.firstName, ' ', a.lastName) AS Author,  
c.edition, p.publisherName, c.ISBN, c.catalogueFormat, c.copiesAvailable,  
c.shelvingInfo, c.genre, c.catalogueDescription, c.ageGroup  
FROM searchCatalogue  
WHERE c.catalogueNumber = fnFindCatalogueNumberByTitle(searchTitle);  
END //
```