KIỂM TRA GIỮA KỲ MÔN: THỰC HÀNH CƠ SỞ DỮ LIỆU 20232

Họ tên sinh viên: Lương Văn Khanh MSSV: 20225728

**Câu 1.**

*SQL:*

CREATE TABLE Book (  
    book\_id CHAR(10) PRIMARY KEY,  
    title CHAR(50) NOT NULL,  
    publisher CHAR(20) NOT NULL,  
    published\_year INTEGER CHECK (published\_year > 1900),  
    total\_number\_of\_copies INTEGER CHECK (total\_number\_of\_copies >= 0),  
    current\_number\_of\_copies INTEGER CHECK (current\_number\_of\_copies >= 0),  
    CHECK (total\_number\_of\_copies >= current\_number\_of\_copies)  
);  
CREATE TABLE Borrower (  
    borrower\_id CHAR(10) PRIMARY KEY,  
    name CHAR(50) NOT NULL,  
    address TEXT,  
    telephone\_number CHAR(12)  
);  
CREATE TABLE BorrowCard (  
    card\_id SERIAL PRIMARY KEY,  
    borrower\_id CHAR(10),  
    borrow\_date DATE NOT NULL,  
    expected\_return\_date DATE NOT NULL,  
    actual\_return\_date DATE,  
    FOREIGN KEY (borrower\_id) REFERENCES Borrower(borrower\_id)  
);  
CREATE TABLE BorrowCardItem (  
    card\_id INT,  
    book\_id CHAR(10),  
    PRIMARY KEY (card\_id, book\_id),  
    FOREIGN KEY (card\_id) REFERENCES BorrowCard(card\_id),  
    FOREIGN KEY (book\_id) REFERENCES Book(book\_id)  
);

**Câu 2.**

*SQL:*

CREATE VIEW BookShortInfo AS

(

SELECT book\_id, title, publisher, published\_year

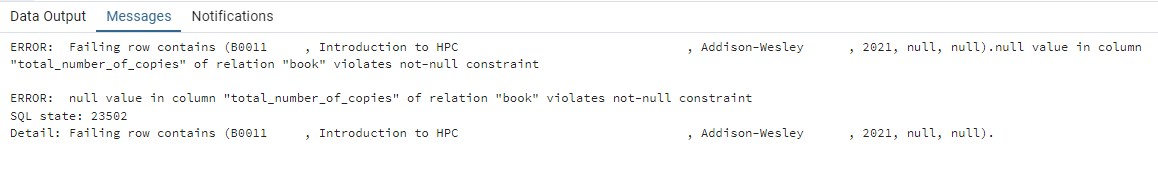
FROM book

);

INSERT INTO BookShortInfo VALUES('B0011', 'Introduction to HPC', 'Addison-Wesley', 2021);

SELECT \* FROM book;

*Kết quả:*

**

**Câu 3.**

*SQL:*

*CREATE OR REPLACE VIEW BorrowerStat AS*

*SELECT be.borrower\_id, be.name, COUNT(bi.book\_id) AS number\_of\_borrows*

*FROM borrower be*

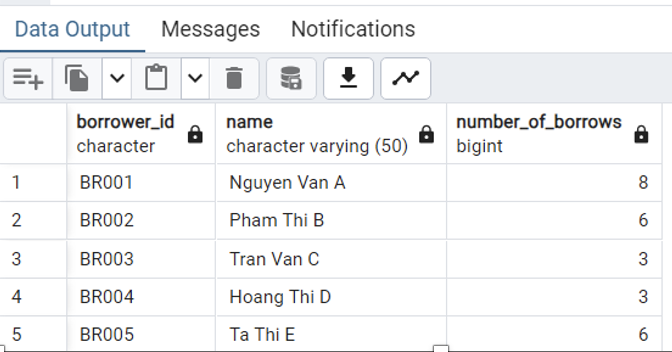
*LEFT JOIN borrowcard bc ON bc.borrower\_id=be.borrower\_id*

*LEFT JOIN borrowcarditem bi ON bc.card\_id=bi.card\_id*

*GROUP BY be.borrower\_id;*

*SELECT \* FROM BorrowerStat;*

*Kết quả:*



**Câu 4.**

*SQL:*

CREATE OR REPLACE FUNCTION number\_of\_borrowed\_copies(id char(5)) RETURNS integer AS

$$

DECLARE cnt integer;

        new\_table record;

BEGIN

   SELECT INTO new\_table book\_id, COUNT(card\_id) AS num

   FROM borrowcarditem

   WHERE book\_id = id

   GROUP BY book\_id;

   IF new\_table.num > 0 THEN cnt = new\_table.num; RETURN cnt;

   ELSE cnt = 0; RETURN cnt;

   END IF;

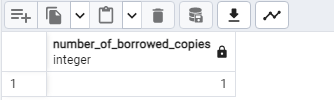
END;

$$

LANGUAGE plpgsql;

SELECT number\_of\_borrowed\_copies('B0001');

*Kết quả:*

**

**Câu 5.**

*SQL:*

CREATE OR REPLACE FUNCTION number\_of\_borrowed\_books\_by\_publisher(publisher CHAR(20), year INT)

RETURNS INT AS $$

DECLARE

   countr INT;

BEGIN

   SELECT COUNT(\*)

   INTO countr

   FROM BorrowCard BC

   JOIN BorrowCardItem BCI ON BC.card\_id = BCI.card\_id

   JOIN Book B ON BCI.book\_id = B.book\_id

   WHERE B.publisher = number\_of\_borrowed\_books\_by\_publisher.publisher

     AND EXTRACT(YEAR FROM BC.borrow\_date) = number\_of\_borrowed\_books\_by\_publisher.year;

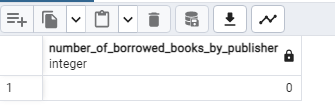
   RETURN countr;

END;

$$ LANGUAGE plpgsql;

SELECT number\_of\_borrowed\_books\_by\_publisher('Wiley', 2012);

*Kết quả:*

**

**Câu 6.**

*SQL:*

CREATE OR REPLACE FUNCTION books\_by\_publisher( publisher CHARACTER)

RETURNS TABLE ( title CHAR) AS $$

BEGIN

  RETURN QUERY

  SELECT title

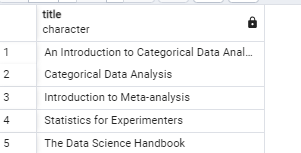
  FROM library.Book

  WHERE book.publisher = books\_by\_publisher.publisher;

END;

$$ LANGUAGE plpgsql;

*Kết quả:*



**Câu 7.**

*SQL:*

CREATE OR REPLACE FUNCTION set\_default\_values()

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.total\_number\_of\_copies IS NULL THEN

        NEW.total\_number\_of\_copies := 0;

    END IF;

    IF NEW.current\_number\_of\_copies IS NULL THEN

        NEW.current\_number\_of\_copies := 0;

    END IF;

    IF NEW.total\_number\_of\_copies < NEW.current\_number\_of\_copies THEN

        RAISE EXCEPTION 'total\_number\_of\_copies must be greater than or equal to current\_number\_of\_copies';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER set\_default\_values\_trigger

BEFORE INSERT OR UPDATE ON book

FOR EACH ROW

EXECUTE FUNCTION set\_default\_values();

*Kết quả:*

**Câu 8.**

*SQL:*

CREATE OR REPLACE FUNCTION update\_current\_copies\_trigger() RETURNS TRIGGER AS

$$

BEGIN

   UPDATE book

   SET current\_number\_of\_copies = current\_number\_of\_copies - 1

   WHERE book\_id = NEW.book\_id;

   RETURN NEW;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER after\_insert AFTER INSERT ON borrowcarditem

FOR EACH ROW

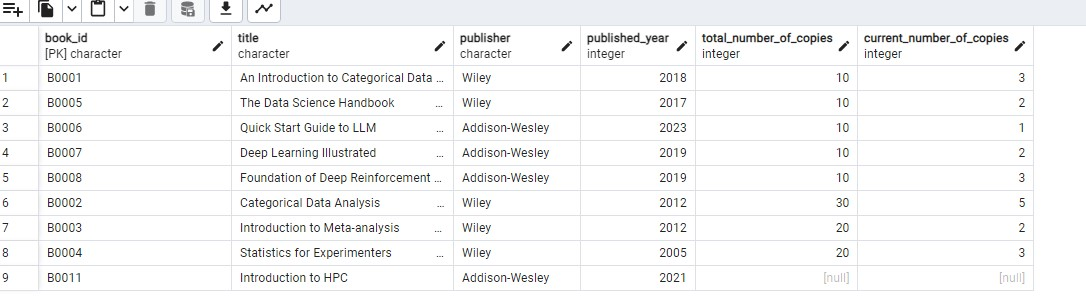
EXECUTE FUNCTION update\_current\_copies\_trigger();

SELECT \* FROM book;

INSERT INTO borrowcarditem VALUES('6', 'B0004');

*Kết quả:*

Trước khi INSERT thì current\_number\_of\_copies của sách có mã ‘B0004’ là 3



Sau khi INSERT thì current\_number\_of\_copies của sách có mã ’B0004’ giảm xuống còn 2



**Câu 9.**

*SQL:*

SELECT b1.card\_id, book\_id, actual\_return INTO query91

FROM borrowcard b1, borrowcarditem b2

WHERE b1.card\_id = b2.card\_id;

CREATE OR REPLACE FUNCTION update\_return\_date\_trigger() RETURNS TRIGGER AS

$$

BEGIN

   UPDATE book

   SET current\_number\_of\_copies = current\_number\_of\_copies + 1

   WHERE book\_id = NEW.book\_id;

   RETURN NEW;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER after\_return\_v1 AFTER UPDATE ON query91

FOR EACH ROW

EXECUTE FUNCTION update\_return\_date\_trigger();

SELECT \* FROM query91;

SELECT \* FROM borrowcard;

SELECT \* FROM book ORDER BY book\_id;

UPDATE query91

SET actual\_return = '2024-05-05'

WHERE card\_id = '8';

*Kết quả:*

Trước khi khách hàng có card\_id là 8 trả sách: (khách này mượn các quyển B0002, B0003, B0008)

Sau khi khách hàng có card\_id là 8 trả sách:

