Nama : Andre

NIM : 220411100028

1. Stored procedure: sp\_countMemberBookTitle (IN nama\_anggota varchar(50))

function: f\_countMemberBookTitle (nama\_anggota varchar(50))

Menampilkan jumlah judul buku yang sudah pernah dipinjam anggota tersebut dengan menggunakan data pada v\_GetAllMemberBookTitle

* Membuat Stored Procedure

DELIMITER //

CREATE PROCEDURE sp\_countMemberBookTitle (IN nama\_anggota VARCHAR(50))

BEGIN

SELECT COUNT(\*) AS 'Jumlah Judul Buku yang Sudah Dipinjam'

FROM v\_GetAllMemberBookTitle

WHERE `nama\_anggota` = nama\_anggota;

END//

DELIMITER ;

* Memanggil Stored Procedure

CALL sp\_countMemberBookTitle('Andre');



* Membuat Function

DELIMITER //

CREATE FUNCTION f\_countMemberBookTitle (nama\_anggota VARCHAR(50))

RETURNS INT

BEGIN

DECLARE jumlah INT;

SELECT COUNT(\*) INTO jumlah

FROM v\_GetAllMemberBookTitle

WHERE `nama\_anggota` = nama\_anggota;

RETURN jumlah;

END//

DELIMITER ;

* Memanggil Function

SELECT `f\_countMemberBookTitle`('Andre') AS 'Jumlah Judul Buku yang Sudah Dipinjam';



1. Stored procedure: sp\_countMemberBookTitle (IN nama\_anggota varchar(50))

function: f\_countMemberBookTitle (nama\_anggota varchar(50))

Menampilkan jumlah judul buku yang belum dikembalikan anggota tersebut dengan menggunakan data pada v\_GetMemberBookTitleUnreturned

* Membuat Stored Procedure

DELIMITER //

CREATE PROCEDURE sp\_countMemberBookTitleUnreturned (IN nama\_anggota VARCHAR(50))

BEGIN

SELECT COUNT(\*) AS 'Jumlah Judul Buku yang Belum Dikembalikan'

FROM v\_GetMemberBookTitleUnreturned

WHERE nama\_anggota = nama\_anggota;

END//

DELIMITER ;

* Memanggil Stored Procedure

CALL sp\_countMemberBookTitleUnreturned('Andre');



* Membuat Function

DELIMITER //

CREATE FUNCTION f\_countMemberBookTitleUnreturned (nama\_anggota VARCHAR(50))

RETURNS INT

BEGIN

DECLARE jumlah INT;

SELECT COUNT(\*) INTO jumlah

FROM v\_GetMemberBookTitleUnreturned

WHERE nama\_anggota = nama\_anggota;

RETURN jumlah;

END//

DELIMITER ;

* Memanggil Function

SELECT f\_countMemberBookTitleUnreturned('Andre') AS 'Jumlah Judul Buku yang Belum Dikembalikan';



1. Stored procedure: sp\_GetMemberAmountFine (IN nama\_anggota varchar(50))

function: f\_GetMemberAmountFine (nama\_anggota varchar(50))

Menampilkan jumlah denda yang penah dibayarkan anggota tersebut dengan menggunakan data pada v\_GetMemberAmountFine

* Membuat Stored Procedure

DELIMITER //

CREATE PROCEDURE sp\_GetMemberAmountFine (IN nama\_anggota VARCHAR(50))

BEGIN

SELECT `Jumlah\_Denda`

FROM v\_GetMemberAmountFine

WHERE `Nama` = nama\_anggota;

END//

DELIMITER ;

* Memanggil Stored Procedure

CALL sp\_GetMemberAmountFine('andre');



* Membuat Function

DELIMITER //

CREATE FUNCTION f\_GetMemberAmountFine (nama\_anggota VARCHAR(50))

RETURNS DECIMAL(10,2)

BEGIN

DECLARE total\_denda DECIMAL(10,2);

SELECT `Jumlah\_Denda` INTO total\_denda

FROM v\_GetMemberAmountFine

WHERE `Nama` = nama\_anggota;

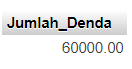
RETURN total\_denda;

END//

DELIMITER ;

* Memanggil Function

SELECT f\_GetMemberAmountFine('andre') AS 'Total Denda';



1. Stored procedure: sp\_GetTotalFinePerMonth (IN nama\_bulan varchar(50))

function: f\_GetTotalFinePerMonth (nama\_bulan varchar(50))

Menampilkan jumlah denda yang diterima perpustakaan pada bulan tersebut dengan menggunakan data pada v\_GetTotalFinePerMonth

* Membuat Stored Procedure

DELIMITER //

CREATE PROCEDURE sp\_GetTotalFinePerMonth (IN nama\_bulan VARCHAR(50))

BEGIN

SELECT Jumlah\_Denda

FROM v\_GetTotalFinePerMonth

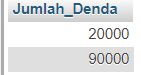
WHERE bulan = bulan;

END//

DELIMITER ;

* Memanggil Stored Procedure

CALL sp\_GetTotalFinePerMonth('2024-02');



* Membuat Function

DELIMITER //

CREATE FUNCTION f\_GetTotalFinePerMonth (bulan VARCHAR(10))

RETURNS DECIMAL(10,2)

BEGIN

DECLARE Jumlah\_Denda DECIMAL(10,2) DEFAULT 0.00;

SELECT COALESCE(SUM(jmldenda), 0)

INTO Jumlah\_Denda

FROM transaksi\_pengembalian

WHERE DATE\_FORMAT(tanggal\_pengembalian, '%Y-%m') = bulan;

RETURN Jumlah\_Denda;

END//

DELIMITER ;

* Memanggil Function

SELECT f\_GetTotalFinePerMonth('2024-02') AS 'Total Denda';



1. Stored procedure: sp\_GetOfficerMemberBookTitleLoanDate (IN nama\_petugas varchar(100))

function: f\_GetOfficerMemberBookTitleLoanDate (nama\_petugas varchar(100))

Menampilkan jumlah transaksi peminjaman yang dilakukan oleh petugas perpustakaan tersebut dengan menggunakan data pada v\_GetOfficerMemberBookTitleLoanDate

* Membuat Stored Procedure

DELIMITER //

CREATE PROCEDURE sp\_GetOfficerMemberBookTitleLoanDate (IN nama\_petugas VARCHAR(100))

BEGIN

SELECT COUNT(\*) AS 'Jumlah Transaksi Peminjaman'

FROM v\_GetOfficerMemberBookTitleLoanDate

WHERE nama\_petugas = nama\_petugas;

END//

DELIMITER ;

* Memanggil Stored Procedure

CALL sp\_GetOfficerMemberBookTitleLoanDate('adit');



* Membuat Function

DELIMITER //

CREATE FUNCTION f\_GetOfficerMemberBookTitleLoanDate (nama\_petugas VARCHAR(100))

RETURNS INT

BEGIN

DECLARE jumlah INT;

SELECT COUNT(\*) INTO jumlah

FROM v\_GetOfficerMemberBookTitleLoanDate

WHERE nama\_petugas = nama\_petugas;

RETURN jumlah;

END//

DELIMITER ;

* Memanggil Function

SELECT f\_GetOfficerMemberBookTitleLoanDate('adit') AS 'jumlah Transaksi Peminjaman';



1. Buat trigger before insert (t\_BeforeInsertPengembalian) pada tabel pengembalian untuk mengecek stok buku yang akan dipinjam apakah tersedia atau tidak

* Membuat Trigger

DELIMITER //

CREATE TRIGGER t\_BeforeInsertPengembalian

BEFORE INSERT ON transaksi\_pengembalian

FOR EACH ROW

BEGIN

DECLARE stok INT;

SELECT stok\_buku INTO stok

FROM buku

WHERE IDBuku = NEW.IDBuku;

IF stok <= 0 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Stok buku tidak tersedia';

END IF;

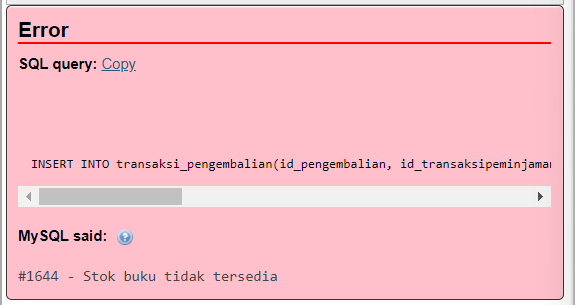
END//

DELIMITER ;

* Output

INSERT INTO transaksi\_pengembalian(id\_pengembalian, id\_transaksipeminjaman, id\_anggota, IDBuku, id\_petugas, tanggal\_pengembalian, tanggal\_aktual\_pengembalian, status, jmldenda) VALUES ('019','004','002','001','004','2024-02-10','2024-02-05','Kembali', 0)

\*Stok buku pada id 001 bernilai 0



1. Buat trigger after insert (t\_AfterInsertPengembalian) pada tabel pengembalian untuk mengupdate kuota peminjaman tiap anggota

* Membuat Trigger

DELIMITER //

CREATE TRIGGER t\_AfterInsertPengembalian

AFTER INSERT ON transaksi\_pengembalian

FOR EACH ROW

BEGIN

UPDATE anggota

SET kuota\_peminjaman = kuota\_peminjaman - 1

WHERE id\_anggota = NEW.id\_anggota;

END//

DELIMITER ;

* Output

Sebelum Melakukan Insert pada tabel pengembalian, kuota peminjaman anggota dengan id 001 bernilai 5



Lalu melakukan Insert

INSERT INTO transaksi\_pengembalian(id\_pengembalian, id\_transaksipeminjaman, id\_anggota, IDBuku, id\_petugas, tanggal\_pengembalian, tanggal\_aktual\_pengembalian, status, JmDenda) VALUES ('019','015','001','003','001','2024-06-01','2024-06-15','Kembali', 0)

Maka otomatis kuota peminjaman dengan id 001 akan berkurang menjadi 4

