Nama : Rima Natalia

Kelas : XII RPL 2

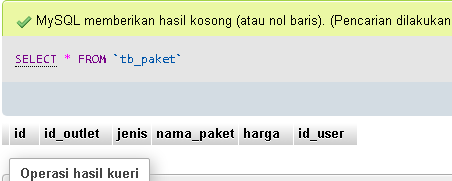
Doc. : Database

**1. Membuat Database**

****

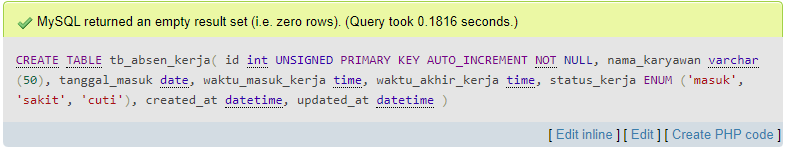
CREATE DATABASE db\_laundry\_192010068;

**2. Menambahkan field id\_user pada table paket**

****

ALTER TABLE tb\_paket ADD id\_user int;

**3. Menambahkan table tb\_absen\_kerja dengan field yang sudah ditentukan**

****

CREATE TABLE tb\_absen\_kerja(

id int UNSIGNED PRIMARY KEY AUTO\_INCREMENT NOT NULL,

nama\_karyawan varchar (50),

tanggal\_masuk date,

waktu\_masuk\_kerja time,

waktu\_akhir\_kerja time,

status\_kerja ENUM ('masuk', 'sakit', 'cuti'),

created\_at datetime,

updated\_at datetime

);

**4. Buat Foreign Key (Indexing) antar table**

ALTER TABLE tb\_user ADD CONSTRAINT FOREIGN KEY (id\_outlet) REFERENCES tb\_outlet(id);

ALTER TABLE tb\_paket

ADD CONSTRAINT FOREIGN KEY (id\_outlet) REFERENCES tb\_outlet(id),

ADD CONSTRAINT FOREIGN KEY (id\_user) REFERENCES tb\_user(id);

ALTER TABLE tb\_transaksi

ADD CONSTRAINT FOREIGN KEY (id\_user) REFERENCES tb\_user(id),

ADD CONSTRAINT FOREIGN KEY (id\_outlet) REFERENCES tb\_outlet(id),

ADD CONSTRAINT FOREIGN KEY (id\_member) REFERENCES tb\_member(id);

ALTER TABLE tb\_detail\_transaksi ADD CONSTRAINT FOREIGN KEY (id\_transaksi) REFERENCES tb\_transaksi(id);

**5. Menambahkan 10 data pada table tb\_absen\_kerja**

INSERT INTO tb\_absen\_kerja

(nama\_karyawan, tanggal\_masuk, waktu\_masuk\_kerja, waktu\_akhir\_kerja, status\_kerja, created\_at, updated\_at)

VALUES

("Isti", "2022-03-22", "07:00", "16:00", "sakit", NOW(), NOW()),

("Ara", "2022-03-25", "07:00", "16:00", "masuk", NOW(), NOW()),

("Jae", "2022-03-23", "07:00", "16:00", "cuti", NOW(), NOW()),

("Jeno", "2022-03-22", "07:00", "16:00", "sakit", NOW(), NOW()),

("Sisi", "2022-03-25", "07:00", "16:00", "masuk", NOW(), NOW()),

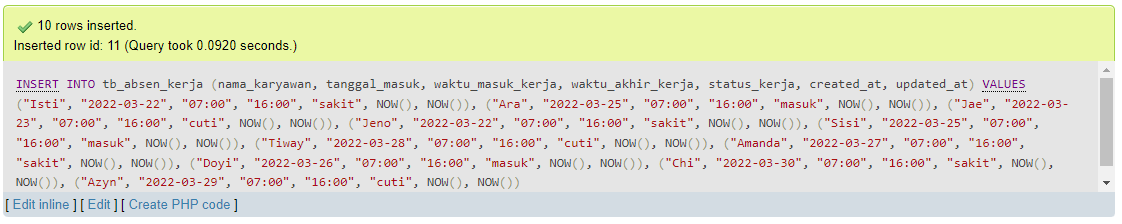
("Tiway", "2022-03-28", "07:00", "16:00", "cuti", NOW(), NOW()),

("Amanda", "2022-03-27", "07:00", "16:00", "sakit", NOW(), NOW()),

("Doyi", "2022-03-26", "07:00", "16:00", "masuk", NOW(), NOW()),

("Chi", "2022-03-30", "07:00", "16:00", "sakit", NOW(), NOW()),

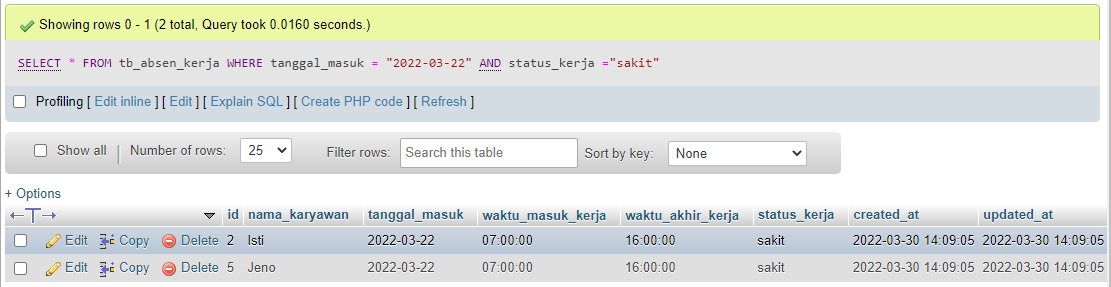
("Azyn", "2022-03-29", "07:00", "16:00", "cuti", NOW(), NOW());

****

****

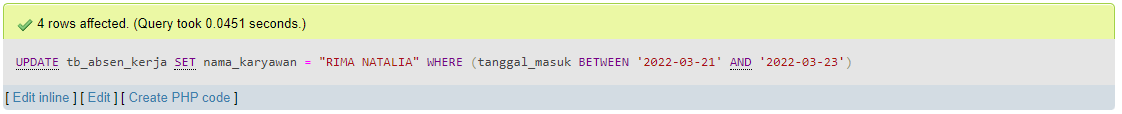
**6. Membuat instruksi SQL untuk menampilkan data di tb\_absen\_kerja yang statusnya “sakit”**

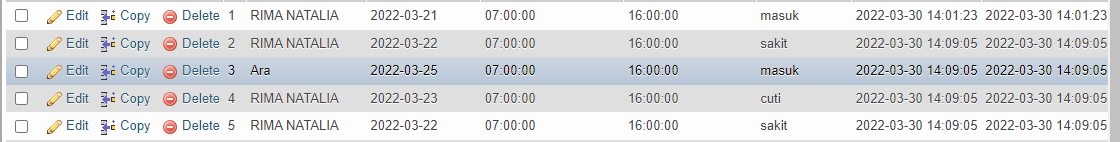
SELECT \* FROM tb\_absen\_kerja WHERE tanggal\_masuk = "2022-03-22" AND status\_kerja ="sakit";

****

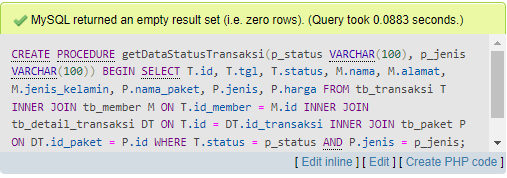
**7. Buat satu intruksi SQL untuk mengupdate data di tb\_absen\_kerja dengan nama kalian yang tanggal\_masuk antara 21-03-2022 sampai 23-03-2022**

UPDATE tb\_absen\_kerja SET nama\_karyawan = "RIMA NATALIA" WHERE (tanggal\_masuk BETWEEN '2022-03-21' AND '2022-03-23');

****

****

**8. Membuat Stored Procedure yang memiliki status selesai dan jenis paket kaos**



//

CREATE PROCEDURE getDataStatusTransaksi(p\_status VARCHAR(100), p\_jenis VARCHAR(100))

BEGIN

SELECT T.id, T.tgl, T.status, M.nama, M.alamat, M.jenis\_kelamin, P.nama\_paket, P.jenis, P.harga

FROM tb\_transaksi T

INNER JOIN tb\_member M ON T.id\_member = M.id

INNER JOIN tb\_detail\_transaksi DT ON T.id = DT.id\_transaksi

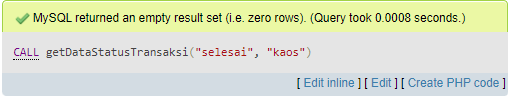
INNER JOIN tb\_paket P ON DT.id\_paket = P.id

WHERE T.status = p\_status

AND P.jenis = p\_jenis;

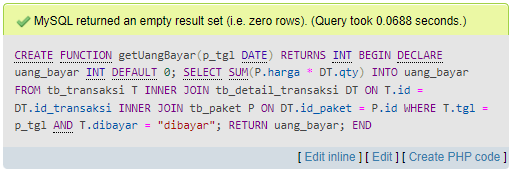
END

//



CALL getDataStatusTransaksi("selesai", "kaos");

**9. Buat Function Menghasilkan Total\_Uang\_bayar**

****

//

CREATE FUNCTION getUangBayar(p\_tgl DATE) RETURNS INT

BEGIN

DECLARE uang\_bayar INT DEFAULT 0;

SELECT SUM(P.harga \* DT.qty) INTO uang\_bayar FROM tb\_transaksi T

INNER JOIN tb\_detail\_transaksi DT ON T.id = DT.id\_transaksi

INNER JOIN tb\_paket P ON DT.id\_paket = P.id

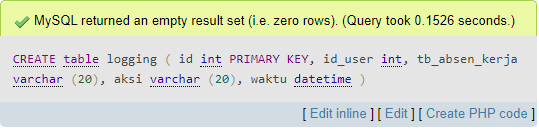
WHERE T.tgl = p\_tgl AND T.dibayar = "dibayar";

RETURN uang\_bayar;

END

//

**10. Membuat table logging dan 3 trigger pada table tb\_absen\_kerja**

****

CREATE table logging (

id int PRIMARY KEY,

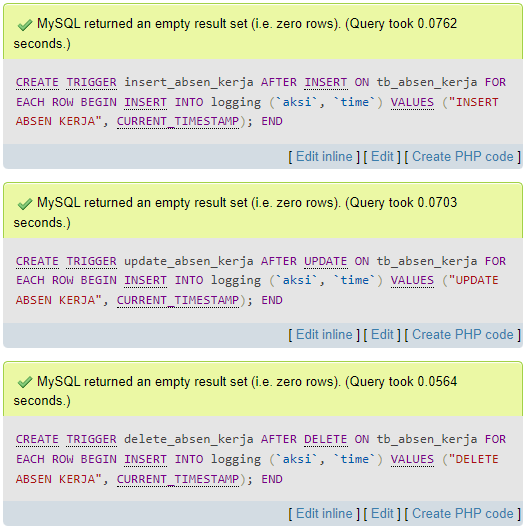
id\_user int,

tb\_absen\_kerja varchar (20),

aksi varchar (20),

waktu datetime

);

****

CREATE TRIGGER insert\_absen\_kerja AFTER INSERT ON tb\_absen\_kerja FOR EACH ROW

BEGIN

INSERT INTO logging (`aksi`, `time`) VALUES ("INSERT ABSEN KERJA", CURRENT\_TIMESTAMP);

END//

CREATE TRIGGER update\_absen\_kerja AFTER UPDATE ON tb\_absen\_kerja FOR EACH ROW

BEGIN

INSERT INTO logging (`aksi`, `time`) VALUES ("UPDATE ABSEN KERJA", CURRENT\_TIMESTAMP);

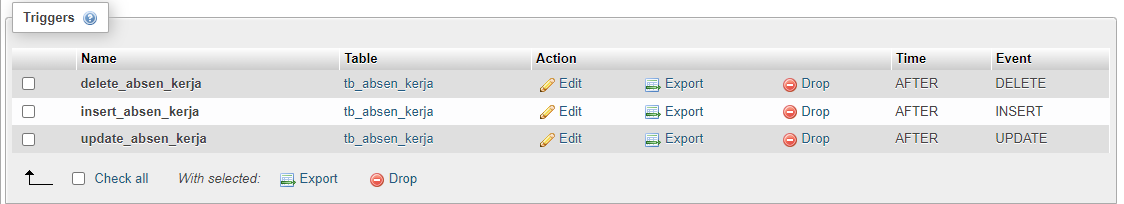
END//

CREATE TRIGGER delete\_absen\_kerja AFTER DELETE ON tb\_absen\_kerja FOR EACH ROW

BEGIN

INSERT INTO logging (`aksi`, `time`) VALUES ("DELETE ABSEN KERJA", CURRENT\_TIMESTAMP);

END//



**11. Membuat v\_transaksi\_selesai**

CREATE VIEW v\_transaksi\_selesai AS

SELECT T.id, T.tgl, T.status, M.nama, M.alamat, M.jenis\_kelamin, SUM(P.harga \* DT.qty) AS total\_bayar

FROM tb\_transaksi T

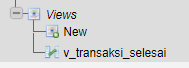
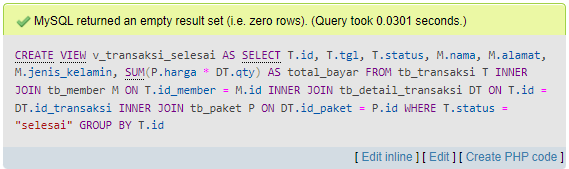
INNER JOIN tb\_member M ON T.id\_member = M.id

INNER JOIN tb\_detail\_transaksi DT ON T.id = DT.id\_transaksi

INNER JOIN tb\_paket P ON DT.id\_paket = P.id

WHERE T.status = "selesai"

GROUP BY T.id;

****

**12. Perintah COMMIT**



START TRANSACTION;

INSERT INTO tb\_outlet (nama, alamat, tlp)

VALUES ("Jamal", "Jl. Cahaya", "086355272");

COMMIT;

**13. Perintah ROLLBACK**

****

START TRANSACTION;

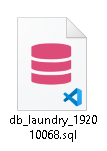
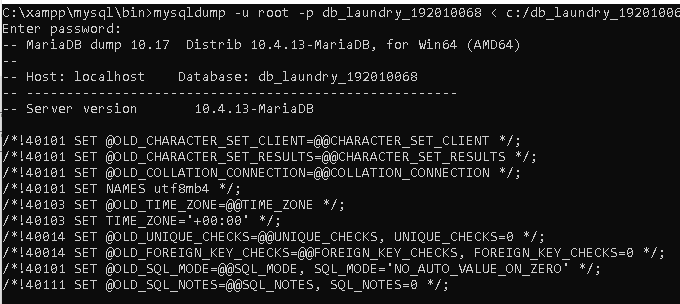
INSERT INTO tb\_outlet (nama, alamat, tlp)

VALUES ("Aminah", "Jl. Bumi", "086355272");

ROLLBACK;

**14. BACKUP & RESTORE**

****

****