**II2250 Manajemen Basis Data**
**Semester II 2022/2023**

# Praktikum IV: Constraints and Trigger

Dipersiapkan oleh:
Asisten Laboratorium Basis Data

Pelaksanaan:
Waktu Mulai: Senin, 20 Maret 2023 – 16.15 WIB
Waktu Selesai: Senin, 20 Maret 2023 – 18.00 WIB

## I. Ketentuan

Peserta kuliah akan mengerjakan praktikum mengenai **Constraints and Trigger** pada DBMS PostgreSQL menggunakan data hospital.sql yang tersedia di folder praktikum ini. Praktikum ini dikerjakan secara **INDIVIDU**.

Peserta dapat memanfaatkan file yang terdapat pada folder Untuk Peserta dan dokumentasi PostgreSQL dalam pengerjaan soal praktikum.

*Deliverables* yang harus dikumpulkan untuk Praktikum ini adalah sebagai berikut.

1. File .docx berisikan daftar query beserta screenshot eksekusi tiap query dengan format nama: **K<Kelas>_P04_<NIM>.docx**
   **Contoh: K01_P04_18221000.docx**

Isi dari dokumen adalah sebagai berikut.

a. Untuk setiap soal.
   i. Perintah atau langkah yang dilakukan untuk menyelesaikan persoalan. Sediakan semua kode dalam bentuk **teks**.
   ii. Query pengecekan dalam bentuk **teks** (jika diminta pada soal).
   iii. Hasil perintah dari *query* tersebut

# *Deliverables* dikumpulkan pada pranala berikut.

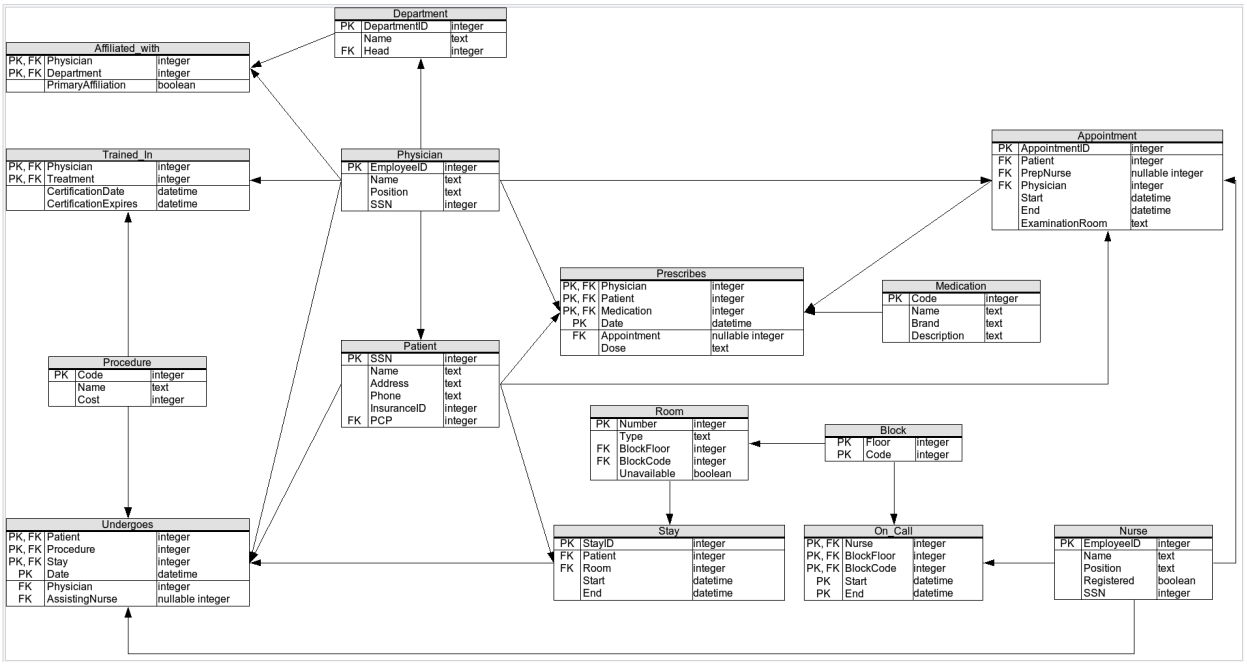
https://s.id/uploader-prak4

**Keterlambatan pengumpulan *deliverables* akan mengakibatkan pengurangan nilai. Segala tindak kecurangan akan ditindaklanjuti dan diikuti konsekuensi serius.**

**~ Selamat mengerjakan! :)) ~**

## II. Skema Basis Data

Berikut ini adalah skema dari basis data Rumah Sakit Basdat dari file hospital.sql.




Affiliated_with = (Physician, Department, PrimaryAffiliation)

Appointment = (AppointmentID, Patient, PrepNurse, Physician, StartDate, EndDate, ExaminationRoom)

Block = (BlockFloor, BlockCode)

Department = (DepartmentID, Name, Head)

Medication = (Code, Name, Brand, Description)

Nurse = (EmployeeID, Name, Position, Registered, SSN)

On_call = (Nurse, BlockFloor, BlockCode, OnCallStart, OnCallEnd)

Patient = (SSN, Name, Address, Phone, InsuranceID, PCP)

Physician = (EmployeeID, Name, Position, SSN)

Prescribes = (Physician, Patient, Medication, Date, Appointment, Dose)

Procedures = (Code, Name, Cost)

Room = (RoomNumber, RoomType, BlockFloor, BlockCode, Unavailable)

Stay = (StayID, Patient, Room, StayStart, StayEnd)

Trained_in = (Physician, Treatment, CertificationDate, CertificationExpires)

Undergoes = (Patient, Procedures, Stay, DateUndergoes, Physician, AssistingNurse)

## III. Soal

(**Note**: Pastikan bahwa pada komputer yang digunakan, telah terdapat sebuah database bernama hospital. Jika belum, buatlah sebuah database bernama **hospital** dan import hospital.sql ke dalam database tersebut!)

1. Buatlah sebuah database bernama **rumahsakit** yang merupakan duplikat dari database **hospital**.
2. Rumah Sakit Basdat memiliki beberapa tipe ruangan. Tipe-tipe ruangan yang ada adalah sebagai berikut:

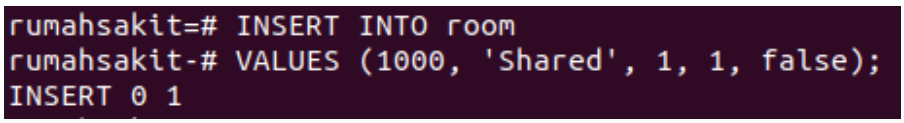
```
roomtype: ['Single', 'Shared', 'ICU', 'Recovery']
```

Untuk memastikan data yang dimasukan ke dalam basisdata benar. Buatlah:

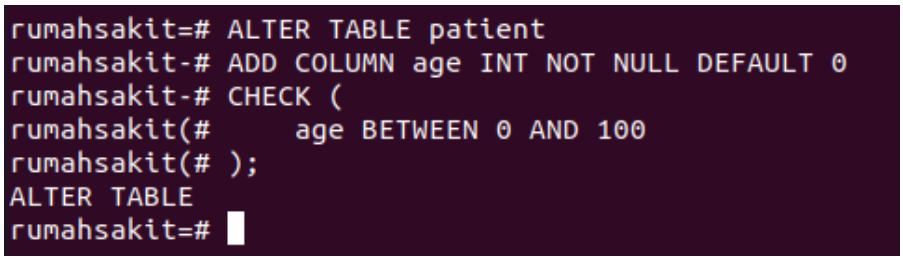
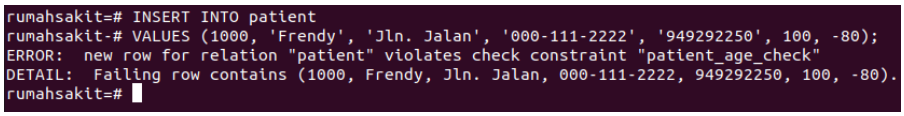
A. Buatlah sebuah *check constraint* bernama *valid_room_type* yang akan memeriksa apakah *record* baru untuk roomtype yang dimasukan adalah salah satu diantara nama ruangan yang ada di rumah sakit Basdat.
B. Buatlah contoh penambahan *record* pada tabel Room yang melanggar *check constraint.*
C. Buatlah contoh penambahan *record* pada tabel Room yang tidak melanggar *check constraint.*
D. Hapus kembali *record* yang baru saja dimasukan.

**Jawaban:**

| | |
|---|---|
| **Query Pembuatan Constraint** | `ALTER TABLE room`<br>`ADD CONSTRAINT valid_room_type`<br>`CHECK (`<br>`roomtype IN ('Single', 'Shared', 'ICU', 'Recovery')`<br>`);` |
| **SS Pembuatan Constraint** | `rumahsakit=# ALTER TABLE room`<br>`rumahsakit-# ADD CONSTRAINT valid_room_type`<br>`rumahsakit-# CHECK (`<br>`rumahsakit(# roomtype IN ('Single', 'Shared', 'ICU', 'Recovery')`<br>`rumahsakit(# );`<br>`ALTER TABLE` |
| **SS Pelanggaran Constraint** | `rumahsakit=# INSERT INTO room`<br>`rumahsakit-# VALUES (1000, 'Basdat', 1, 1, false);`<br>`ERROR: new row for relation "room" violates check constraint "valid_room_type"`<br>`DETAIL: Failing row contains (1000, Basdat, 1, 1, f).` |

| | |
|---|---|
| **SS Tidak Melanggar Constraint** |  |

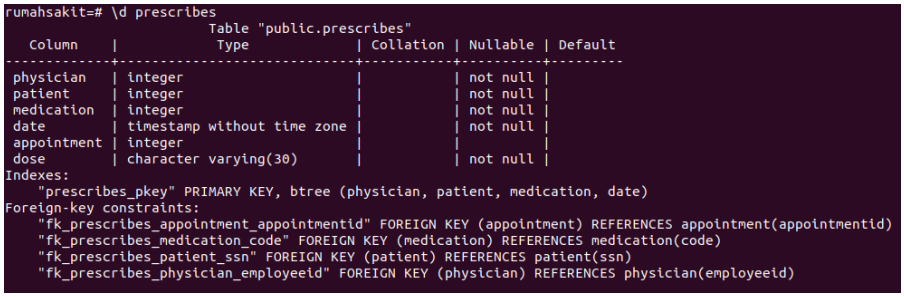
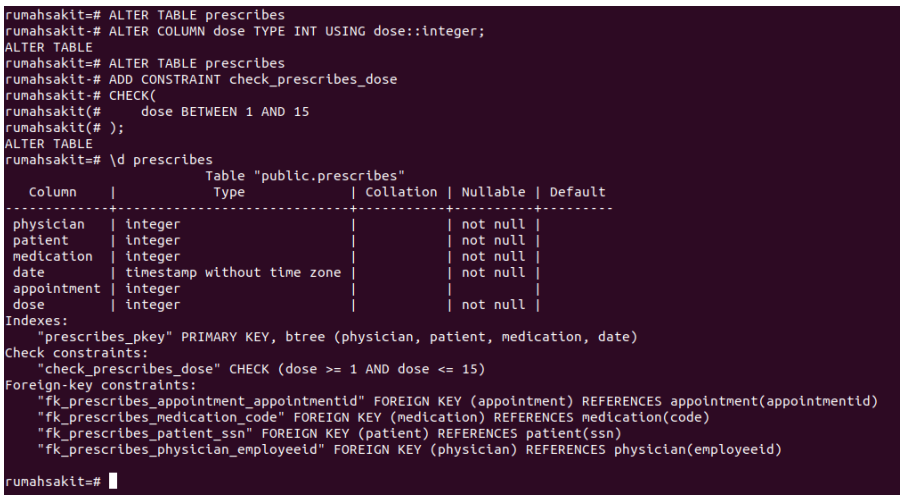
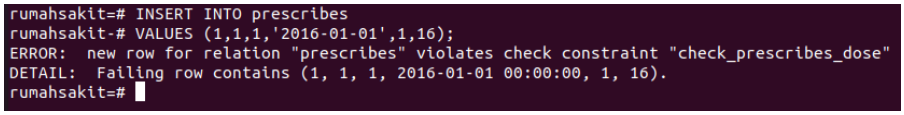
3. Rumah Sakit Basdat perlu melakukan pencatatan data pasien sebagai salah satu dasar untuk menentukan pemberian perawatan yang tepat. Salah satu data yang belum tercatat saat ini adalah data umur dari pasien.
    A. Buatlah sebuah kolom baru bernama *age* pada tabel *patient* yang akan menyimpan data umur dari setiap *record* pasien. Kolom tersebut akan menyimpan tipe data integer dan harus menerapkan *constraint* NOT NULL serta memiliki nilai *default* sebesar 0.
    B. Selanjutnya, buatlah sebuah *check constraint* yang akan memeriksa apakah nilai umur yang dimasukkan bernilai valid. Data umur bernilai valid apabila berada pada rentang nilai 0 hingga 100.
    C. Terakhir, buatlah sebuah contoh penambahan data pada tabel pasien yang melanggar *check constraint* yang telah dibuat pada bagian sebelumnya.

| | |
|---|---|
| **Query Penambahan Kolom & Pembuatan Constraint** | `ALTER TABLE patient`<br>`ADD COLUMN age INT NOT NULL DEFAULT 0`<br>`CHECK (`<br>`  age BETWEEN 0 AND 100`<br>`);` |
| **SS Query Penambahan Kolom dan Pembuatan Constraint** |  |
| **Query Contoh Insert** | `INSERT INTO patient`<br>`VALUES (1000, 'Frendy', 'Jln. Jalan',`<br>`'000-111-2222', '949292250', 100, -80);` |
| **SS Query Contoh Insert** |  |

4. Rumah Basdat memiliki sebuah data resep obat yang diberikan kepada pasien dalam tabel *Prescribes*, namun terdapat kesalahan skema pada tabel tersebut. Kolom *dose* pada tabel *Prescribes* memiliki tipe *varchar.*

    Untuk memastikan data yang diinput memiliki nilai yang benar, lakukanlah hal-hal berikut:

a. Ubah tipe data pada kolom *dose* menjadi *integer*.
b. Berikan *check constraint* untuk memastikan kolom *dose* memiliki nilai dalam rentang 1-15. (*1 ≤ dose ≤ 15*)
c. Buatlah sebuah contoh penambahan data pada tabel pasien yang melanggar check constraint yang telah dibuat pada bagian sebelumnya.

| | |
|---|---|
| **Detail Schema Tabel Prescribes (Before)** |  |
| **Query Perubahan Type Kolom** | `ALTER TABLE prescribes`<br>`ALTER COLUMN dose TYPE INT USING dose::integer;` |
| **Query Pembuatan Constraint** | `ALTER TABLE prescribes`<br>`ADD CONSTRAINT check_prescribes_dose`<br>`CHECK(`<br>`dose BETWEEN 1 AND 15`<br>`);` |
| **Detail Schema Tabel Prescribes (After)** |  |
| **SS Insert Pelanggaran Constraint** |  |

5. Saat ini data appointment yang dimiliki Rumah Sakit masih terdapat data yang belum konsisten, yaitu terdapat appointment yang berakhir sebelum appointment tersebut dimulai. Oleh karenanya pihak Rumah Sakit membutuhkan anda untuk memperbaikinya
    A. Hapuslah semua data appointment yang berakhir sebelum appointment tersebut dimulai.
    B. Buatlah sebuah *check constraint* yang akan memeriksa apakah data appointment yang diberikan bernilai benar
    C. Buatlah sebuah contoh penambahan data pada tabel appointment yang melanggar check constraint yang telah dibuat pada bagian sebelumnya.
    D. Hapuslah Constraint yang baru saja anda buat.

| | |
|---|---|
| **Query Hapus Data yang Tidak Konsisten** | `ALTER TABLE prescribes`<br>`DROP CONSTRAINT`<br>`fk_prescribes_appointment_appointmentid;`<br><br>`DELETE FROM appointment`<br>`WHERE startdate > enddate;` |
| **SS Query Pembuatan Constraint** | `rumahsakit=# ALTER TABLE appointment`<br>`rumahsakit-# ADD CONSTRAINT check_startdate_enddate`<br>`rumahsakit-# CHECK (`<br>`rumahsakit(#  startdate < enddate`<br>`rumahsakit(# );`<br>`ALTER TABLE`<br>`rumahsakit=#` |
| **Query Contoh Insert dan Berikan SS nya** | `INSERT INTO appointment`<br>`VALUES (1,1,1,1,'2018-01-01', '2017-01-01', 'A');`<br><br>`rumahsakit=# INSERT INTO appointment`<br>`rumahsakit-# VALUES (1,1,1,1,'2018-01-01', '2017-01-01', 'A');`<br>`ERROR:  new row for relation "appointment" violates check constraint "check_startdate_enddate"`<br>`DETAIL:  Failing row contains (1, 1, 1, 1, 2018-01-01 00:00:00, 2017-01-01 00:00:00, A).`<br>`rumahsakit=#` |
| **Hapus Constraint** | `ALTER TABLE appointment`<br>`DROP CONSTRAINT check_startdate_enddate;`<br><br>`rumahsakit=# ALTER TABLE appointment`<br>`rumahsakit-# DROP CONSTRAINT check_startdate_enddate;`<br>`ALTER TABLE` |

6. Rumah sakit basdat perlu memastikan bahwa ruangan yang digunakan untuk rawat inap terutama dengan tipe single tidak terjadi bentrok. Oleh karena itu, rumah sakit basdat ingin menerapkan constraint untuk melakukan pengecekan sebelum ruangan di booking, yaitu dengan memastikan ruangan yang dibooking berstatus available.
    A. Buatlah query pengaplikasian constraint tersebut.
       HINT:
       - Gunakan Trigger
       - Gunakan EXISTS untuk mengecek apakah subquery mereturn row atau tidak

       HINT Ref :
       - https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-exists/
    B. Buatlah contoh insert pada table stay dengan kasus valid dan tidak valid, ss query untuk memperlihatkan room available dan unavailable dan lakukan insert dengan roomnumber yang available dan roomnumber yang available
    C. (Optional) Jelaskan algoritma pembuatan trigger tersebut in case pembuatan trigger tidak sesuai dengan harapan.

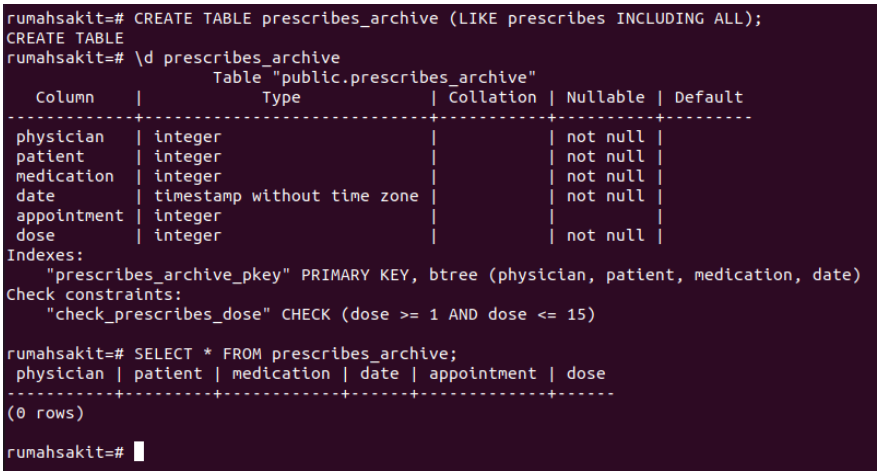
| **Query Pengaplikasian Constraint** | |
|---|---|

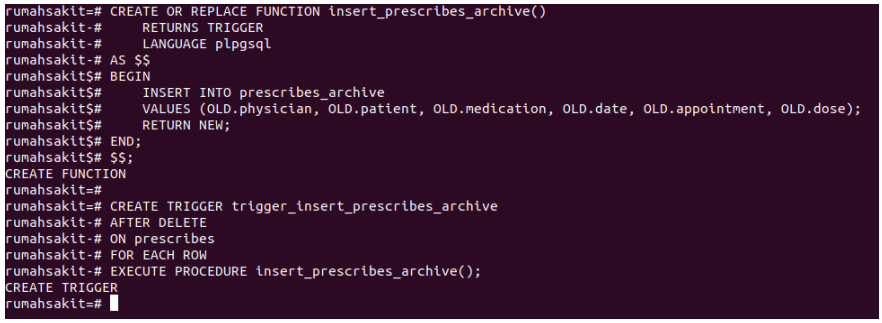
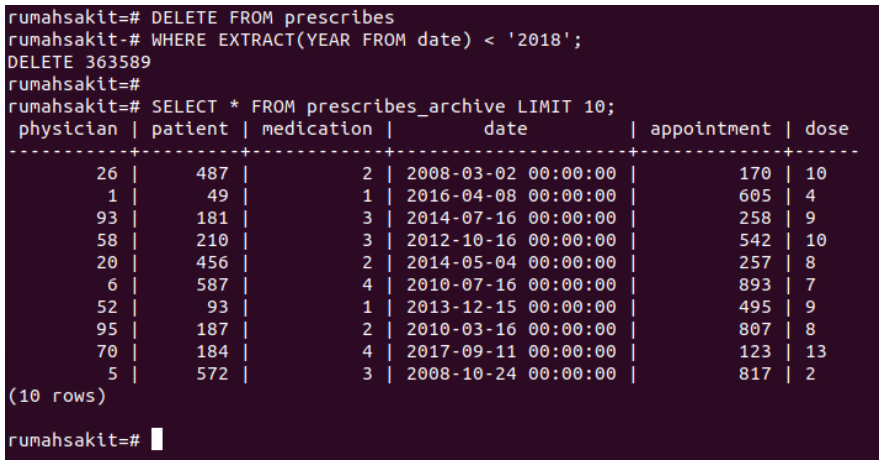
```
CREATE OR REPLACE FUNCTION
check_room_availability()
      RETURNS TRIGGER
      LANGUAGE plpgsql
AS $$
BEGIN
      IF EXISTS (
      SELECT *
      FROM room
      WHERE NEW.room = roomnumber AND roomnumber
IN ( SELECT roomnumber FROM room WHERE roomtype =
'Single' AND unavailable = false)
    ) THEN INSERT INTO stay VALUES(NEW.stayid,
NEW.patient, NEW.room, NEW.staystart,
NEW.stayend);
    END IF;

      RETURN NEW;
END;
$$;

CREATE TRIGGER trigger_check_room_availability
BEFORE INSERT
ON stay
```

| | |
|---|---|
| | `FOR EACH ROW`<br>`EXECUTE PROCEDURE check_room_availability();` |
| **SS Query Pengaplikasian Constraint** | |
| **SS Insert Berhasil dan Gagal** | `-- BERHASIL`<br>`INSERT INTO stay`<br>`VALUES (100000, 1, 1, '2018-01-01',`<br>`'2019-01-01');`<br><br>`-- GAGAL`<br>`INSERT INTO stay`<br>`VALUES (100000, 1, 10, '2018-01-01',`<br>`'2019-01-01');` |
| **Penjelasan Algoritma** | |

7. Rumah Sakit Basdat mencatat banyak data prescribes yang terjadi setiap hari pada setiap appointment. Untuk mempercepat waktu pengaksesan data prescribes, rumah sakit basdat ingin memisahkan data prescribes yang sudah lama pada sebuah tabel arsip.
    A. Buatlah sebuah query untuk membuat tabel baru dengan nama prescribes_archive dengan atribut yang sama dengan tabel prescribes.
    HINT Ref:
        - https://www.geeksforgeeks.org/postgresql-copy-table/
    B. Buatlah sebuah trigger untuk menambahkan data baru pada tabel prescribes_archive setiap menghapus data dari tabel prescribes.
    HINT:
        - Buat Function terlebih dahulu sebelum membuat Trigger
    HINT Ref:
        - https://www.postgresql.org/docs/current/sql-createfunction.html
        - https://www.postgresql.org/docs/current/sql-createtrigger.html
    C. Buatlah query untuk menghapus data pada tabel prescribes yang terjadi sebelum tahun 2018. Lalu perlihatkan 10 data arsip terbaru dari tabel prescribes_archive.

**Jawaban:**

| | |
|---|---|
| **Query Pembuatan Tabel Baru (A)** | `CREATE TABLE prescribes_archive (LIKE prescribes INCLUDING ALL);` |
| **SS Pembuatan Tabel Baru (A)** |  |
| **Query Pembuatan Trigger (B)** | `CREATE OR REPLACE FUNCTION insert_prescribes_archive()`<br>`RETURNS TRIGGER`<br>`LANGUAGE plpgsql`<br>`AS $$`<br>`BEGIN`<br>`INSERT INTO prescribes_archive`<br>`VALUES (OLD.physician, OLD.patient, OLD.medication, OLD.date, OLD.appointment, OLD.dose);`<br>`RETURN NEW;`<br>`END;`<br>`$$;`<br><br>`CREATE TRIGGER trigger_insert_prescribes_archive`<br>`AFTER DELETE`<br>`ON prescribes`<br>`FOR EACH ROW` |

| | |
|---|---|
| | `EXECUTE PROCEDURE insert_prescribes_archive();` |
| **SS Pembuatan Trigger (B)** |  |
| **Query Pengaplikasian Trigger (C)** | `DELETE FROM prescribes`<br>`WHERE EXTRACT(YEAR FROM date) < '2018';`<br><br>`SELECT * FROM prescribes_archive LIMIT 10;` |
| **SS Pengaplikasian Trigger (C)** |  |

## Referensi


[PostgreSQL Check Constraint (tutorialsteacher.com)]

[PostgreSQL - CONSTRAINTS (tutorialspoint.com)]

[PostgreSQL: COPY TABLE (geeksforgeeks.org)]

[PostgreSQL: Documentation: 15: CREATE FUNCTION]

[PostgreSQL: Documentation: 15: CREATE TRIGGER]


## Feedback Praktikum

bit.ly/feedback-praktikum