PL SQL

* <https://www.postgresqltutorial.com/postgresql-plpgsql/>
* <https://www.postgresqltutorial.com/postgresql-triggers/>
* <https://www.postgresql.org/docs/current/plpgsql.html>
* <https://www.postgresql.org/docs/current/server-programming.html>

Maak deze oefeningen op je lokale databank of de probeer databank. De vereisten voor het oplossen van de oefeningen breng je zelf in orde (bv lokaal de ruimtereizen databank draaien, ..)

|  |  |
| --- | --- |
| **TIP** | * try catch wordt BEGIN .. EXCEPTION .. END * syntax : de $$ delim kan je gebruiken ipv ' (escapen niet meer nodig ea) * taal: SQL : zoals in bash wordt er met $1, $2 enzoverder naar je parameters verwezen * taal: plpgsql : parameters hebben ook namen, meer mogelijkheden dan in sql * triggers: NEW en OLD sleutelwoorden |

Functies

* Schrijf een functie die de grootte van gegeven hemelobject teruggeeft (op basis van de tabel hemelobjecten uit de ruimtereizen databank)

Afbeelding met tekst, schermopname, software

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

* Schrijf een functie die drie getallen optelt en de som teruggeeft.

Afbeelding met tekst, elektronica, schermopname, software

Automatisch gegenereerde beschrijving

* Schrijf een functie waarbij je gegevens uit een transactie tabel verwijdert die ouder zijn dan 10 dagen (of een andere “archiverings” functie op 1 van jou eigen tabellen).

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving 

* Zou het mogelijk zijn een aggregatie functie te schrijven? Hoe zou je dit dan moeten doen en is dit sql standaard

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving



* Schrijf een functie met als input parameter een tabel, die als output een overzicht geeft van alle kolommen. Bv stel er bestaan een table A (x text, y int, z char(4)), dan zou *geef\_kolommen(A)* dit als uitvoer hebben:
* x text;
* y int;

z char(4);

tip: check table information about pg\_attribute

Afbeelding met tekst, schermopname, software

Automatisch gegenereerde beschrijving

SELECT \* FROM geef\_kolommen('A');

Procedures

* Schrijf een procedure waarmee een hemelobject kan worden verhuisd naar een andere reis. Dus gegeven twee reisnummers en een hemelobjecten, verplaats je het hemelobject van de ene reis naar de reis. De info van bezoek blijft verder hetzelfde, alleen wordt het verplaatste bezoek het laatste van de nieuwe reis. Een voorbeeld.
  + Reisnr 1 bezoekt de Aarde, Maan en Zon
  + Reisnr 2 bezoekt de Aarde, Maan en Mars
  + *verplaats(1,2,'Maan')* zorgt ervoor:
    - Reisnr 1 bezoekt de Aarde, Zon
    - Reisnr 2 bezoekt de Aarde, Maan, Mars en Maan

Afbeelding met tekst, schermopname, software

Automatisch gegenereerde beschrijving

* Uitbreiding:
  + Zorg ervoor de volgorde nummer blijven kloppen van de reis met een bezoek minder. Bv Reisnr 1 bezoekt de Aarde 1, Zon 2 ipv Reisnr 1 bezoekt de Aarde 1, Zon 3
  + Zorg ervoor dat enkel een commit gebeurt zolang de reis bezoeken blijft hebben. Vermijd dus dat een reis zonder bezoeken zou zitten. Gebruik hiervoor een oplossing met transacties.
* Welke van de bovenstaande functies kan je als procedure schrijven? Probeer dit ook.

Triggers

* Schrijf een functie waarbij je zinnige uitvoer probeert terug te geven naar gebruikers wanneer ze gegevens verkeerd/vergeten in voeren in een zelf gekozen tabel en zorg ervoor dat dit vanzelf gecontroleerd wordt.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

* Zorg ervoor dat er een tabel met een simpele statistiek van een tabel naar keuze wordt bijgehouden, namelijk het aantal inserts dat er reeds is op uitgevoerd.
* In welke situaties kan je een trigger schrijven? Bv bij een select die door een gebruiker word uitgevoerd?
  + Integriteitscontroles
* Schrijf een trigger die de 10 oudste boetes verwijdert als er meer dan 50 rijen in de boetes tabel zit (tennis databank)
* Afbeelding met tekst, elektronica, schermopname, software

  Automatisch gegenereerde beschrijving
* Als men een nieuwe boete ingeeft, zorg er dan voor dat een boodschap (notice) wordt teruggegeven met het totaal aantal boetes en de totale som aan boetebedragen (tennis databank)

Combinaties

Tellen

Zoals je misschien herinnerd, kan het tellen van al de rijen soms langs duren. Dus het gebruik van de count() functie.

Maak op je lokale databank een tabel met minstens 1 miljoen rijen.

View

Maak een view waarin de rijen van deze tabel telt. Hoe kan je ervoor zorgen dat deze view niet steeds opnieuw moet worden uitgevoerd, ie de onderliggende code. Maak zo een view aan.

Trigger

Schrijf een trigger die bij een toevoeging of verwijdering van een rij in de tabel een teller in teltabel aanpast. Het doel van deze teltabel is het aantal rijen van een andere tabel bij te houden.

Performantie

Welk van de 3 bovenstaande methode is het meest performant?

* rechstreeks tellen
* view (optie 1 en 2)
* teltabel (via trigger)

Catalogus

* Toon de beschikbare talen die je voor een functie/procedure/.. kan gebruiken
* Toon de beschikbare functies die je kan gebruiken.

Constraints

* Voeg een beperking toe die ervoor zorgt dat boetes van meer dan 200 euro niet kunnen (tennis databank). Probeer dit op zoveel verschillende manieren te bewerkstelligen.

Last updated 2024-04-17 12:25:46 +0200

Code in tekst:

CREATE OR REPLACE FUNCTION bereken\_grootte(object\_naam VARCHAR)

RETURNS INT AS $$

DECLARE

diameter\_object INT;

BEGIN

-- Haal de diameter op van het gegeven hemelobject

SELECT diameter INTO diameter\_object

FROM hemelobjecten

WHERE objectnaam = bereken\_grootte.object\_naam;

-- Return de diameter

RETURN diameter\_object;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- Als het object niet wordt gevonden, return NULL

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

SELECT objectnaam, bereken\_grootte(objectnaam) AS diameter

FROM hemelobjecten;

CREATE OR REPLACE FUNCTION optellen(num1 INT, num2 INT, num3 INT)

RETURNS INT AS $$

DECLARE

som INT;

BEGIN

som := num1 + num2 + num3;

RETURN som;

END;

$$ LANGUAGE plpgsql;

SELECT optellen(5, 10, 15); -- Dit zal 30 retourneren

CREATE OR REPLACE FUNCTION archiveer\_transacties()

RETURNS VOID AS $$

DECLARE

archief\_datum DATE;

BEGIN

-- Bereken de archiefdatum als 10 dagen geleden vanaf vandaag

archief\_datum := CURRENT\_DATE - INTERVAL '10 days';

-- Verwijder gegevens uit de transactietabel die ouder zijn dan de archiefdatum

DELETE FROM transactietabel

WHERE transactiedatum < archief\_datum;

END;

$$ LANGUAGE plpgsql;

SELECT archiveer\_transacties();

CREATE OR REPLACE FUNCTION gemiddelde(numbers INT[])

RETURNS FLOAT AS $$

DECLARE

total INT := 0;

count INT := 0;

avg FLOAT := 0.0;

i INT;

BEGIN

-- Loop door de array en bereken de som en het aantal elementen

FOR i IN 1..array\_length(numbers, 1) LOOP

total := total + numbers[i];

count := count + 1;

END LOOP;

-- Bereken het gemiddelde, let op het gebruik van NULLIF om deling door nul te voorkomen

IF count > 0 THEN

avg := total / NULLIF(count, 0);

END IF;

RETURN avg;

END;

$$ LANGUAGE plpgsql;

SELECT gemiddelde(ARRAY[5, 10, 15, 20]);

CREATE OR REPLACE FUNCTION geef\_kolommen(tabel\_naam VARCHAR)

RETURNS TABLE (kolom\_naam VARCHAR, datatype VARCHAR) AS $$

DECLARE

rec RECORD;

BEGIN

FOR rec IN SELECT column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = tabel\_naam

ORDER BY ordinal\_position LOOP

-- Return elk kolomnaam en datatype

kolom\_naam := rec.column\_name;

datatype := rec.data\_type;

RETURN NEXT;

END LOOP;

RETURN;

END;

$$ LANGUAGE plpgsql;

SELECT \* FROM geef\_kolommen('A');

Trigger 1:

set search\_path to ruimtereizen;

-- Voorbeeldtabel klanten

CREATE TABLE klanten\_info (

klantnr SERIAL PRIMARY KEY,

naam VARCHAR(50),

voornaam VARCHAR(50),

geboortedatum DATE

);

-- Triggerfunctie om lege naam te controleren

CREATE OR REPLACE FUNCTION controleer\_naam\_invoer()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.naam IS NULL OR NEW.naam = '' THEN

RAISE EXCEPTION 'Naam mag niet leeg zijn.';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- Trigger om de triggerfunctie toe te passen op de tabel klanten

CREATE TRIGGER naam\_controle

BEFORE INSERT OR UPDATE ON klanten\_info

FOR EACH ROW EXECUTE FUNCTION controleer\_naam\_invoer();

set search\_path to ruimtereizen;

-- Correcte invoer

INSERT INTO klanten\_info (naam, voornaam, geboortedatum) VALUES ('Doe', 'John', '1990-05-15');

-- Ongeldige invoer (lege naam)

INSERT INTO klanten\_info (naam, voornaam, geboortedatum) VALUES ('', 'Jane', '1985-08-20');