

parlamentares que já realizaram todos os tipos de gastos	
1) SELECT p.codigo FROM parlamentar p WHERE (SELECT COUNT(codigo) FROM tipogasto) = (SELECT COUNT(DISTINCT g.codtipo) FROM gastorealizado g WHERE g.codpar = p.codigo)	$\Pi_{CODPAR, CODTIPO}(GASTOREALIZADO) /$ $\Pi_{CODIGO}(TIPOGASTO)$
nome do parlamentar e os códigos dos tipos de gastos realizados entre os dias e parlamentares que não tiveram gastos	
SELECT p.nome, g.codtipo FROM parlamentar p LEFT JOIN gastorealizado g ON g.codpar = p.codigo WHERE datagasto > '2017-01-01' AND datagasto < '2017-12-01'	2 -> PARLAM. $\neg_{CODPAR=CODIGO}$ GASTOREA. 1 -> $(\sigma_{DATAGASTO > '2017-01-01'}$ and $DATAGASTO < '2017-12-01'$ (2) RESULT -> $\Pi_{NOME, CODIGO}$ (1)
soma do valor gasto por tipo de gasto (código do tipo) considerando o mês x e os tipos de gastos com soma > 10.000	
SELECT SUM(valor), codtipo FROM gastorealizado WHERE datagato > 'aaaa-mm-aa' AND datagato < '2020-01-01' GROUP BY (codtipo) HAVING SUM(valor) > 99	1 -> $CODTIPO \ G_{SUM(V.ALOR)} > 1000$ (GASTOR.) RESULT -> $\sigma_{DATAGASTO > '2017-10-01'$ and $DATAGASTO < '2017-10-31'$ (1)
nome do parlamentar que gastou mais que a média do valor gasto no restaurante cujo CNPJ = X	
SELECT p.nome FROM parlamentar p JOIN gastorealizado g ON g.codpar = p.codigo WHERE g.valor > (SELECT avg(g.valor) FROM gastorealizado g JOIN fornecedor f ON g.codfomec = f.codigo WHERE f.cnpj = '42345')	3 -> $(\sigma_{CNPJ = '42345'}$ ($G_{avg(valor)}(GASTOREALIZ)$ 2 -> GASTOREALIZ $\neg_{codfomec=codigo}$ FORNECEDOR)(3) 1 -> $(\sigma_{VALOR > (2)}$ (GASTOREALIZADO) RESULT -> $\Pi_{NOME} (PARLAMENTAR \neg_{codfomec=codigo}$ GASTOREALIZADO)(1)
nome do parlamentar, o título do tipo de gasto, a razão social do fornecedor e o valor dos gastos realizados	
SELECT p.nome, t.titulo, f.razaosocial, g.valor FROM parlamentar p JOIN gastorealizado g ON p.codigo = g.codpar JOIN fornecedor f ON f.codigo = g.codfomec JOIN tipogasto t ON t.codigo = g.codtipo	3 -> TIPOGASTO $\neg_{CODTIPO=CODIGO}$ GASTOREALIZADO 2 -> FORNECEDOR $\neg_{CODFORNEC=CODIGO}$ 3 1 -> PARLAMENTAR $\neg_{CODPAR=CODIGO}$ 2 RESULT. -> $\Pi_{NOME, TITULO, RAZAOSOCIAL, VALOR}$ (1)

2) Elabore uma visão (view) com os dados listados. Neste caso, o aspecto essencial é a performance da aplicação uma vez que essa consulta representa um custo de processamento considerável.

```
CREATE VIEW MATERIALIZED transparencia(parlamentar, tipogasto, gastorealizado, fornecedor) AS
SELECT p.nome, t.titulo, g.valor, g.datagasto, g.forma, f.razaosocial
FROM GASTOREALIZADO G
JOIN PARLAMENTAR P
ONG.CODPAR = P.CODIGO
JOIN TIPOGASTO T
ONG.CODTIPO = T.CODIGO
JOIN FORNECEDOR F
ONG.CODFORNEC = F.CODIGO
```

3) a cada alteração de gastos, o valor da cota do parlamentar associado ao gasto deve ser alterado. Elabore um gatilho de maneira que essa alteração seja devidamente realizada

```
CREATE OR REPLACE FUNCTION nomefuncaotriger()
RETURNS TRIGGER AS $$ BEGIN
IF(TG_OP == 'UPDATE') THEN
UPDATE parlamentar SET valorcota = valorcota - NEW.valor WHERE codigo = NEW.codigo;

UPDATE parlamentar SET valorcota = valorcota - OLD.valor WHERE codigo = OLD.codigo;
ELSE IF(TG_OP = 'INSERT') THEN
UPDATE parlamentar SET valorcota = valorcota - NEW.valor WHERE codigo = NEW.codigo;
ELSE IF(TG_OP = 'DELETE') THEN
UPDATE parlamentar SET valorcota = valorcota - OLD.valor WHERE codigo = OLD.codigo;
END IF;
RETURN NULL;
END; $$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER nometriger / AFTER UPDATE OR INSERT OR DELETE ON gastorealizado
FOR EACH ROW / EXECUTE PROCEDURE nomefuncaotriger();
```

4) o novo valor da cota deverá ser passado como parâmetro, caso o valor atual seja maior ou igual a 0, o mesmo deverá ser substituído. Contudo, caso o valor atual seja menor que 0, o valor negativo deve ser abatido no valor atual. Isso deve ser feito para todos os parlamentares cadastrados. Elabore o procedimento descrito

```
CREATE OR REPLACE FUNCTION atualizaCota(novoValor float)
RETURNS VOID AS $$ DECLARE p parlamentar
BEGIN
FOR p IN SELECT * FROM parlamentar LOOP
IF(p.valor > 0) THEN
UPDATE parlamentar SET valor = novoValor WHERE codigo = p.codigo;
ELSE IF(p.valor = 0) THEN
UPDATE parlamentar SET valor = novoValor WHERE codigo = p.codigo;
ELSE IF(p.valor > 0) THEN
UPDATE parlamentar SET valor = valor + novoValor WHERE codigo = p.codigo;
END IF;
END LOOP;
END; $$ LANGUAGE plpgsql;
```