

CSE 180 - Lab 4

Stored Functions

Dev
November 2023, Section 9



Logistics

- Lab 4 Due 11:59 PM, Tuesday December 5
- Final in less than two weeks!



Functions in runNewspaperApplication

- `increaseSomeRates(PGconn *conn, int maxTotalRateIncrease) :`
 - Returns : The value returned by stored function `increaseSomeRatesFunction`
 - Arguments: `int maxTotalRateIncrease`
 - Calls the stored function

Stored Function

`increaseSomeRatesFunction(maxTotalRateIncrease INTEGER):`

iterates through the subscriptionKinds for rate increase

- increase 10 if popularity ≥ 5
- increase 5 if $3 \leq \text{popularity} < 5$
- increase 3 if $2 \leq \text{popularity} < 3$
- increase 0 if $0 \leq \text{popularity} < 2$
- total increase cannot exceed maxTotalRateIncrease

Updates only if the total does not exceed maximum allowed. Returns total increase.

Creating Functions in PL/pgSQL

```
[ <<label>> ]  
[ DECLARE  
    declarations ]  
BEGIN  
    statements  
END [ label ];
```

Assignment Operators

- In PostgreSQL, Assignment operator is := and equality is =

```
rating = 'h'; //checks if rating is h  
rating := 'h'; //sets rating to h
```

- Type is specified after variable name

```
profits INTEGER;
```

Cursors

- Cursors get the result of a query a few rows at a time, they iterate

```
DECLARE
```

```
    curs3 CURSOR FOR SELECT * FROM student WHERE studentID > 25;
```

```
    FETCH [ direction { FROM | IN } ] cursor INTO target;
```

<https://www.postgresql.org/docs/current/plpgsql-cursors.html>

Conditionals

```
IF boolean-expression THEN
    statements
[ ELSIF boolean-expression THEN
    statements
[ ELSIF boolean-expression THEN
    statements
    ...
]]
[ ELSE
    statements ]
END IF;
```


Loops

- Unconditional LOOP, FOR, WHILE, FOREACH

```
[ <<label>> ]
```

```
LOOP
```

```
statements
```

```
END LOOP [ label ];
```

```
[ <<label>> ]
```

```
WHILE boolean-expression LOOP
```

```
statements
```

```
END LOOP [ label ];
```

- For unconditional loops, you need to use EXIT, CONTINUE,

```
[ <<label>> ]
```

```
FOR name IN expression LOOP
```

```
statements
```

```
END LOOP [ label ];
```

Debugging

- RAISE NOTICE Syntax

```
RAISE NOTICE 'Increased Rates by %', rateIncrease;
```

Calling a Stored Function

```
\i increaseSomeRatesFunction.pgsql  
CREATE FUNCTION
```

```
SELECT increaseSomeRates(20);
```