

Portland PostgreSQL User Group

Creating an Auto-Partition Strategy

Ed Snajder (/ˈsnīdər/)
@edinor @edthadba
Jive Software

Table Partitioning

- Typical for analytic system or eternal storage
- Often partitioned on, but not limited to, date
 - Ranges or Lists
- Implements table inheritance
- Query performance can benefit from constraint exclusion
 - requires at least default setting of `constraint_exclusion=partition`
- Easy to implement
- PITA to maintain
- Ideal at design time
 - Never happens

Creating a Partitioned Table

--Parent Table No Indexes

```
CREATE TABLE fact_table(PK INTEGER, TimeStamp  
TIMESTAMP, eventdata VARCHAR(255);
```

--y2014q3 Partition

```
CREATE TABLE fact_table_y2014q3  
(CHECK (creationDate>='2014-07-01' AND  
creationDate<'2014-10-01')  
INHERITS fact_table;
```

```
CREATE UNIQUE INDEX ft_y2014q3_PK  
ON fact_table_y2014q3 (PK);
```

```
CREATE INDEX ft_y2014q3_cdate  
ON fact_table_y2014q3 (creationDate);
```

PK	creationDate	eventData
1	2014-08-05	Fact 156354
200	2014-11-11	Fact 11234767
3000	2015-02-18	Fact 69384790

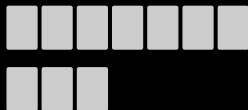
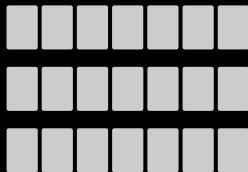
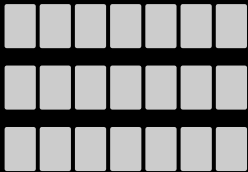
```
CREATE TRIGGER  
ft_insert_trigger BEFORE  
INSERT ON fact_table  
EXECUTE ft_insert_trigger();
```

fact_table
PK
creationDate
eventData

fact_table_y2014q3
PK
creationDate
eventData

fact_table_y2014q4
PK
creationDate
eventData

fact_table_y2015q1
PK
creationDate
eventData



The Trigger Function

Dumber Function

CREATE OR REPLACE FUNCTION

ft_insert_trigger() RETURNS TRIGGER AS \$\$

BEGIN

IF (NEW.creationDate>='2014-07-01' AND NEW.creationDate <'2014-10-01') THEN

INSERT INTO fact_table_y2014q3() VALUES (NEW.*);

ELSEIF (NEW.creationDate>='2014-10-01' AND NEW.creationDate <'2015-01-01') THEN

INSERT INTO fact_table_y2014q4() VALUES (NEW.*);

ELSEIF (NEW.creationDate>='2015-01-01' AND NEW.creationDate <'2015-04-01') THEN

INSERT INTO fact_table_y2015q1() VALUES (NEW.*);

ELSE

RAISE EXCEPTION 'Date out of range';

END IF;

RETURN NULL;

END;

\$\$

LANGUAGE 'plpgsql';



Smarter Function

CREATE OR REPLACE FUNCTION

ft_insert_trigger() RETURNS TRIGGER AS \$\$

DECLARE newdate TIMESTAMP WITH TIME ZONE;

DECLARE inserttbl VARCHAR;

BEGIN

newdate:=NEW.creationDate;

inserttbl:='fact_table_y' || EXTRACT(YEAR FROM newdate) || 'q' ||
EXTRACT(QUARTER from newdate);

EXECUTE 'INSERT INTO ' || inserttbl || 'SELECT (\$1).* USING NEW;

RETURN NULL;

EXCEPTION WHEN undefined_table THEN

PERFORM('SELECT tableautopartadd_date' || tblname || colname ||
partitioninterval || NEW.creationDate;

EXECUTE 'INSERT INTO ' || inserttbl || 'SELECT (\$1).* USING NEW;

RETURN NULL;

END;

\$\$

LANGUAGE 'plpgsql';



Implementing an Auto-Partition Strategy

Function for new partition creation

- Make a copy of the master table, without indexes (new parent)
 - *CREATE TABLE fact_table_part (AS FACT TABLE EXCLUDING ALL);*
- Make a copy of the master table, with index (partition template)
 - *CREATE TABLE fact_table_template (AS FACT TABLE INCLUDING ALL);*
- Create partition tables for range of existing data
 - *CREATE TABLE fact_table_y2014q1 (CHECK part_col) INHERITS fact_table_template;*
- Create trigger function
 - Trigger function will include exception handling if correct table not found
- Add trigger to parent table
 - *CREATE trigger triggername BEFORE INSERT ON fact_table EXECUTE triggerfunction();*
- Create new partition function
 - This trigger will be called when a new partition needs to be created
- Create insert scripts
- Create final transfer scripts
- Create rename table scripts
- Finally, after verification, remove original fact table

Demo

Visit https://github.com/edinor/pg_autopart_simple