

Usage Scenarios

Let us go through some of the usage scenarios with respect to partitioning.

- It is typically used to manage large tables so that the tables does not grow abnormally over a period of time.
- Partitioning is quite often used on top of log tables, reporting tables etc.
- If a log table is partitioned and if we want to have data for 7 years, partitions older than 7 years can be quickly dropped.
- Dropping partitions to clean up huge chunk of data is much faster compared to running delete command on non partitioned table.
- For tables like orders with limited set of statuses, we often use list partitioning based up on the status. It can be 2 partitions (CLOSED orders and ACTIVE orders) or separate partition for each status.
 - As most of the operations will be on **Active Orders**, this approach can significantly improve the performance.
- In case of log tables, where we might want to retain data for several years, we tend to use range partition on date column. If we use list partition, then we might end up in duplication of data unnecessarily.

```
%load_ext sql
```

The sql extension is already loaded. To reload it, use:
%reload_ext sql

```
%env DATABASE_URL=postgresql://itversity_sms_user:sms_password@localhost:5432/itversity_sms_db
```

```
env:  
DATABASE_URL=postgresql://itversity_sms_user:sms_password@localhost:5432/itversity_sms_db
```

Note

Monthly partition using list. We need to have additional column to store the month to use list partitioning strategy.

```
%%sql
```

```
DROP TABLE IF EXISTS users_mthly
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db  
Done.
```

```
[]
```

```
%%sql
```

```
CREATE TABLE users_mthly (  
    user_id SERIAL,  
    user_first_name VARCHAR(30) NOT NULL,  
    user_last_name VARCHAR(30) NOT NULL,  
    user_email_id VARCHAR(50) NOT NULL,  
    user_email_validated BOOLEAN DEFAULT FALSE,  
    user_password VARCHAR(200),  
    user_role VARCHAR(1) NOT NULL DEFAULT 'U', --U and A  
    is_active BOOLEAN DEFAULT FALSE,  
    created_dt DATE DEFAULT CURRENT_DATE,  
    created_mnth INT,  
    last_updated_ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (created_mnth, user_id)  
) PARTITION BY LIST(created_mnth)
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db  
Done.
```

```
[]
```

```
%%sql
```

```
CREATE TABLE users_mthly_201601  
PARTITION OF users_mthly  
FOR VALUES IN (201601)
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db  
Done.
```

```
[]
```

```
%%sql
```

```
CREATE TABLE users_mthly_201602  
PARTITION OF users_mthly  
FOR VALUES IN (201602)
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db  
Done.
```

```
[]
```

Note

Monthly partition using range. Partition strategy is defined on top of **created_dt**. No additional column is required.

```
%%sql
```

```
DROP TABLE IF EXISTS users_mthly
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db  
Done.
```

```
[]
```

```
%%sql
```

```
CREATE TABLE users_mthly (  
    user_id SERIAL,  
    user_first_name VARCHAR(30) NOT NULL,  
    user_last_name VARCHAR(30) NOT NULL,  
    user_email_id VARCHAR(50) NOT NULL,  
    user_email_validated BOOLEAN DEFAULT FALSE,  
    user_password VARCHAR(200),  
    user_role VARCHAR(1) NOT NULL DEFAULT 'U', --U and A  
    is_active BOOLEAN DEFAULT FALSE,  
    created_dt DATE DEFAULT CURRENT_DATE,  
    last_updated_ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (created_dt, user_id)  
) PARTITION BY RANGE(created_dt)
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db  
Done.
```

```
[]
```

```
%%sql
```

```
CREATE TABLE users_mthly_201601  
PARTITION OF users_mthly  
FOR VALUES FROM ('2016-01-01') TO ('2016-01-31')
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db  
Done.
```

```
[]
```

```
%%  
%%sql
```

```
CREATE TABLE users_mthly_201602  
PARTITION OF users_mthly  
FOR VALUES FROM ('2016-02-01') TO ('2016-02-29')
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db  
Done.
```

```
[]
```