

# Managing Partitions - List

Print to PDF ►

Let us understand how to manage partitions for a partitioned table using `users_part`.

- All users data with `user_role` as 'U' should go to one partition by name `users_part_u`.
- All users data with `user_role` as 'A' should go to one partition by name `users_part_a`.
- We can add partition to existing partitioned table using `CREATE TABLE partition_name PARTITION OF table_name`.
- We can have a partition for default values so that all the data that does not satisfy the partition condition can be added to it.
- We can have a partition for each value or for a set of values.
  - We can have one partition for `U` as well as `A` and default partition for all other values.
  - We can have individual partitions for `U`, `A` respectively and default partition for all other values.
  - We can use `FOR VALUES IN (val1, val2)` as part of `CREATE TABLE partition_name PARTITION OF table_name` to specify values for respective table created for partition.
- Once partitions are added, we can insert data into the partitioned table.
- We can detach using `ALTER TABLE` and drop the partition or drop the partition directly. To drop the partition we need to use `DROP TABLE` command.

## Note

Here is how we can create partition for default values for a list partitioned table `users_part`.

```
%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
```

```
%%sql
```

```
CREATE TABLE users_part_default
PARTITION OF users_part DEFAULT
```

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

```
[]
```

## Note

All the 3 records will go to default partition as we have not defined any partition for `user_role` 'U'.

```
%%sql
```

```
INSERT INTO users_part (user_first_name, user_last_name, user_email_id, user_role)
VALUES
('Scott', 'Tiger', 'scott@tiger.com', 'U'),
('Donald', 'Duck', 'donald@duck.com', 'U'),
('Mickey', 'Mouse', 'mickey@mouse.com', 'U')
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db
3 rows affected.
```

```
[]
```

%%sql

```
SELECT * FROM users_part_default
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
3 rows affected.

user_id	user_first_name	user_last_name	user_email_id	user_email_validated	user_passw
2	Scott	Tiger	scott@tiger.com	False	No
3	Donald	Duck	donald@duck.com	False	No
4	Mickey	Mouse	mickey@mouse.com	False	No

%%sql

```
CREATE TABLE users_part_a  
PARTITION OF users_part  
FOR VALUES IN ('A')
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
Done.

[]

%%sql

```
UPDATE users_part  
SET  
    user_role = 'A'  
WHERE user_email_id = 'scott@tiger.com'
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
1 rows affected.

[]

%%sql

```
SELECT * FROM users_part
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
3 rows affected.

user_id	user_first_name	user_last_name	user_email_id	user_email_validated	user_passw
2	Scott	Tiger	scott@tiger.com	False	No
3	Donald	Duck	donald@duck.com	False	No
4	Mickey	Mouse	mickey@mouse.com	False	No

%%sql

```
SELECT * FROM users_part_a
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
1 rows affected.

user_id	user_first_name	user_last_name	user_email_id	user_email_validated	user_password
2	Scott	Tiger	scott@tiger.com	False	None

```
%%sql
```

```
SELECT * FROM users_part_default
```

```
* postgresql://itversity_sms_user:***@localhost:5432/itversity_sms_db
2 rows affected.
```

user_id	user_first_name	user_last_name	user_email_id	user_email_validated	user_passw
3	Donald	Duck	donald@duck.com	False	No
4	Mickey	Mouse	mickey@mouse.com	False	No

## Error

This will fail as there are records with user\_role 'U' in default partition.

```
%%sql
```

```
CREATE TABLE users_part_u
PARTITION OF users_part
FOR VALUES IN ('U')
```

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

```

-----
CheckViolation                                Traceback (most recent call last)
/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/engine/base.py in
_execute_context(self, dialect, constructor, statement, parameters, *args)
    1276         self.dialect.do_execute(
-> 1277             cursor, statement, parameters, context
    1278         )

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/engine/default.py in
do_execute(self, cursor, statement, parameters, context)
    592     def do_execute(self, cursor, statement, parameters, context=None):
-> 593         cursor.execute(statement, parameters)
    594

CheckViolation: updated partition constraint for default partition "users_part_default" would
be violated by some row

The above exception was the direct cause of the following exception:

IntegrityError                                Traceback (most recent call last)
<ipython-input-35-fbb5e14aecbd> in <module>
----> 1 get_ipython().run_cell_magic('sql', '', "\nCREATE TABLE users_part_u \nPARTITION OF
users_part \nFOR VALUES IN ('U')\n")

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/IPython/core/interactiveshell.py in
run_cell_magic(self, magic_name, line, cell)
    2369         with self.builtin_trap:
    2370             args = (magic_arg_s, cell)
-> 2371             result = fn(*args, **kwargs)
    2372         return result
    2373

<decorator-gen-135> in execute(self, line, cell, local_ns)

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/IPython/core/magic.py in <lambda>(f,
*a, **k)
    185     # but it's overkill for just that one bit of state.
    186     def magic_deco(arg):
-> 187         call = lambda f, *a, **k: f(*a, **k)
    188
    189         if callable(arg):

<decorator-gen-134> in execute(self, line, cell, local_ns)

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/IPython/core/magic.py in <lambda>(f,
*a, **k)
    185     # but it's overkill for just that one bit of state.
    186     def magic_deco(arg):
-> 187         call = lambda f, *a, **k: f(*a, **k)
    188
    189         if callable(arg):

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sql/magic.py in execute(self, line,
cell, local_ns)
    215
    216     try:
-> 217         result = sql.run.run(conn, parsed["sql"], self, user_ns)
    218
    219         if (

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sql/run.py in run(conn, sql, config,
user_namespace)
    365         else:
    366             txt = sqlalchemy.sql.text(statement)
-> 367             result = conn.session.execute(txt, user_namespace)
    368             _commit(conn=conn, config=config)
    369             if result and config.feedback:

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/engine/base.py in
execute(self, object_, *multiparams, **params)
   1009         )
   1010     else:
-> 1011         return meth(self, multiparams, params)
   1012
   1013     def _execute_function(self, func, multiparams, params):

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/sql/elements.py in
_execute_on_connection(self, connection, multiparams, params)
    296     def _execute_on_connection(self, connection, multiparams, params):
    297         if self.supports_execution:
-> 298             return connection._execute_clauseelement(self, multiparams, params)
    299         else:
    300             raise exc.ObjectNotExecutableError(self)

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/engine/base.py in
_execute_clauseelement(self, elem, multiparams, params)
   1128         distilled_params,

```

```

1129         compiled_sql,
-> 1130         distilled_params,
1131     )
1132     if self._has_events or self.engine._has_events:

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/engine/base.py in
_execute_context(self, dialect, constructor, statement, parameters, *args)
1315     except BaseException as e:
1316         self._handle_dbapi_exception(
-> 1317             e, statement, parameters, cursor, context
1318         )
1319

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/engine/base.py in
_handle_dbapi_exception(self, e, statement, parameters, cursor, context)
1509     elif should_wrap:
1510         util.raise_(
-> 1511             sqlalchemy_exception, with_traceback=exc_info[2], from_=e
1512         )
1513     else:

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/util/compat.py in
raise_(*failed resolving arguments*)
180
181     try:
--> 182         raise exception
183     finally:
184         # credit to

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/engine/base.py in
_execute_context(self, dialect, constructor, statement, parameters, *args)
1275     if not evt_handled:
1276         self.dialect.do_execute(
-> 1277             cursor, statement, parameters, context
1278         )
1279

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/sqlalchemy/engine/default.py in
do_execute(self, cursor, statement, parameters, context)
591
592     def do_execute(self, cursor, statement, parameters, context=None):
--> 593         cursor.execute(statement, parameters)
594
595     def do_execute_no_params(self, cursor, statement, context=None):

IntegrityError: (psycopg2.errors.CheckViolation) updated partition constraint for default
partition "users_part_default" would be violated by some row

[SQL: CREATE TABLE users_part_u PARTITION OF users_part
FOR VALUES IN ('U')]
(Background on this error at: http://sqlalche.me/e/13/gkpj)

```

## Note

We can detach the partition, add partition for 'U' and load the data from detached partitione into the new partition created.

```
%%sql
```

```
ALTER TABLE users_part
DETACH PARTITION users_part_default
```

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

```
[]
```

```
%%sql
```

```
CREATE TABLE users_part_u
PARTITION OF users_part
FOR VALUES IN ('U')
```

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

```
[]
```

%%sql

```
INSERT INTO users_part
SELECT * FROM users_part_default
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
2 rows affected.

[]

%%sql

```
SELECT * FROM users_part_a
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
1 rows affected.

user_id	user_first_name	user_last_name	user_email_id	user_email_validated	user_password
2	Scott	Tiger	scott@tiger.com	False	None

%%sql

```
SELECT * FROM users_part_u
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
2 rows affected.

user_id	user_first_name	user_last_name	user_email_id	user_email_validated	user_passw
3	Donald	Duck	donald@duck.com	False	N
4	Mickey	Mouse	mickey@mouse.com	False	N

#### Note

We can drop and create partition for default or truncate and attach the existing default partition.

%%sql

```
DROP TABLE users_part_default
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
Done.

[]

%%sql

```
CREATE TABLE users_part_default
PARTITION OF users_part DEFAULT
```

\* postgresql://itversity\_sms\_user:\*\*\*@localhost:5432/itversity\_sms\_db  
Done.

[]