

# Indexes for Constraints

Let us understand details related to indexes for constraints.

- Constraints such as primary key and unique are supported by indexes.
- **Primary Key** - Unique and Not Null.
- **Unique** - Unique and can be null.
- Unless data is sorted, we need to perform full table scan to enforce uniqueness. Almost all the databases will create indexes implicitly for Primary Keys as well as Unique constraints.
- We cannot define Primary Key or Unique constraint without associated index.
- It is quite common that we explicitly create indexes on foreign key columns to improve the performance.

```
%load_ext sql
```

```
%env
```

```
DATABASE_URL=postgresql://itversity_retail_user:retail_password@localhost:5432/itversity_retail_db
```

```
env:
```

```
DATABASE_URL=postgresql://itversity_retail_user:retail_password@localhost:5432/itversity_retail_db
```

```
%sql DROP TABLE IF EXISTS users
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
Done.
```

```
[]
```

```
%sql DROP SEQUENCE IF EXISTS users_user_id_seq
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
Done.
```

```
[]
```

```
%%sql
```

```
CREATE TABLE users (
    user_id INT,
    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,
    user_password VARCHAR(200),
    user_role VARCHAR(1),
    is_active BOOLEAN,
    created_dt DATE DEFAULT CURRENT_DATE
)
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
Done.
```

```
[]
```

```
%%sql
```

```
SELECT table_catalog,
       table_name,
       constraint_type,
       constraint_name
FROM information_schema.table_constraints
WHERE table_name = 'users'
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
3 rows affected.
```

table_catalog	table_name	constraint_type	constraint_name
itversity_retail_db	users	CHECK	2200_17365_2_not_null
itversity_retail_db	users	CHECK	2200_17365_3_not_null
itversity_retail_db	users	CHECK	2200_17365_4_not_null

```
%%sql
```

```
SELECT * FROM pg_catalog.pg_indexes
WHERE schemaname = 'public'
      AND tablename = 'users'
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
0 rows affected.
```

schemaname	tablename	indexname	tablespace	indexdef
------------	-----------	-----------	------------	----------

```
%%sql CREATE SEQUENCE users_user_id_seq
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
Done.
```

```
[]
```

```
%%sql
```

```
ALTER TABLE users
  ALTER COLUMN user_id SET DEFAULT nextval('users_user_id_seq'),
  ADD PRIMARY KEY (user_id)
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
Done.
```

```
[]
```

```
%%sql
```

```
SELECT table_catalog,
       table_name,
       constraint_type,
       constraint_name
FROM information_schema.table_constraints
WHERE table_name = 'users'
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
5 rows affected.
```

table_catalog	table_name	constraint_type	constraint_name
itversity_retail_db	users	PRIMARY KEY	users_pkey
itversity_retail_db	users	CHECK	2200_17365_1_not_null
itversity_retail_db	users	CHECK	2200_17365_2_not_null
itversity_retail_db	users	CHECK	2200_17365_3_not_null
itversity_retail_db	users	CHECK	2200_17365_4_not_null

```
%%sql
```

```
SELECT * FROM pg_catalog.pg_indexes
WHERE schemaname = 'public'
      AND tablename = 'users'
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.
```

schemaname	tablename	indexname	tablespace	indexdef
public	users	users_pkey	None	CREATE UNIQUE INDEX users_pkey ON public.users USING btree (user_id)

```
%%sql
```

```
SELECT tc.table_catalog,
       tc.table_name,
       tc.constraint_name,
       pi.indexname
FROM information_schema.table_constraints tc JOIN pg_catalog.pg_indexes pi
      ON tc.constraint_name = pi.indexname
WHERE tc.table_schema = 'public'
      AND tc.table_name = 'users'
      AND tc.constraint_type = 'PRIMARY KEY'
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.
```

table_catalog	table_name	constraint_name	indexname
itversity_retail_db	users	users_pkey	users_pkey

```
%%sql
```

```
ALTER TABLE users
ADD UNIQUE (user_email_id)
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
Done.
```

```
[]
```

```
%%sql
```

```
SELECT table_catalog,
       table_name,
       constraint_type,
       constraint_name
FROM information_schema.table_constraints
WHERE table_name = 'users'
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
6 rows affected.
```

table_catalog	table_name	constraint_type	constraint_name
itversity_retail_db	users	PRIMARY KEY	users_pkey
itversity_retail_db	users	UNIQUE	users_user_email_id_key
itversity_retail_db	users	CHECK	2200_17365_1_not_null
itversity_retail_db	users	CHECK	2200_17365_2_not_null
itversity_retail_db	users	CHECK	2200_17365_3_not_null
itversity_retail_db	users	CHECK	2200_17365_4_not_null

```
%%sql
```

```
SELECT * FROM pg_catalog.pg_indexes
WHERE schemaname = 'public'
      AND tablename = 'users'
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
2 rows affected.
```

schemaname	tablename	indexname	tablespace	indexdef
public	users	users_pkey	None	CREATE UNIQUE INDEX users_pkey ON public.users USING btree (user_id)
public	users	users_user_email_id_key	None	CREATE UNIQUE INDEX users_user_email_id_key ON public.users USING btree (user_email_id)

```
%%sql
```

```
SELECT tc.table_catalog,
       tc.table_name,
       tc.constraint_name,
       pi.indexname
FROM information_schema.table_constraints tc JOIN pg_catalog.pg_indexes pi
     ON tc.constraint_name = pi.indexname
WHERE tc.table_schema = 'public'
      AND tc.table_name = 'users'
      AND tc.constraint_type = 'UNIQUE'
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
1 rows affected.
```

table_catalog	table_name	constraint_name	indexname
itversity_retail_db	users	users_user_email_id_key	users_user_email_id_key

#### Note

Query to get all the primary key and unique constraints along with indexes.

```
%%sql
```

```
SELECT tc.table_catalog,
       tc.table_name,
       tc.constraint_type,
       tc.constraint_name,
       pi.indexname
FROM information_schema.table_constraints tc JOIN pg_catalog.pg_indexes pi
     ON tc.constraint_name = pi.indexname
WHERE tc.table_catalog = 'itversity_retail_db'
      AND tc.constraint_type IN ('PRIMARY KEY', 'UNIQUE')
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
8 rows affected.
```

table_catalog	table_name	constraint_type	constraint_name	indexname
itversity_retail_db	departments	PRIMARY KEY	departments_pkey	departments_pkey
itversity_retail_db	categories	PRIMARY KEY	categories_pkey	categories_pkey
itversity_retail_db	products	PRIMARY KEY	products_pkey	products_pkey
itversity_retail_db	customers	PRIMARY KEY	customers_pkey	customers_pkey
itversity_retail_db	orders	PRIMARY KEY	orders_pkey	orders_pkey
itversity_retail_db	order_items	PRIMARY KEY	order_items_pkey	order_items_pkey
itversity_retail_db	users	PRIMARY KEY	users_pkey	users_pkey
itversity_retail_db	users	UNIQUE	users_user_email_id_key	users_user_email_id_key

#### Error

It is not possible to drop the indexes that are automatically created to enforce primary key or unique constraints.

```
%sql DROP INDEX users_user_email_id_key
```

```
* postgresql://itversity_retail_user:***@localhost:5432/itversity_retail_db
```

```

-----
DependentObjectsStillExist                                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

DependentObjectsStillExist: cannot drop index users_user_email_id_key because constraint
users_user_email_id_key on table users requires it
HINT: You can drop constraint users_user_email_id_key on table users instead.

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

InternalError                                            Traceback (most recent call last)
<ipython-input-89-7b38c07068a1> in <module>
----> 1 get_ipython().run_line_magic('sql', 'DROP INDEX users_user_email_id_key')

/opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/IPython/core/interactiveshell.py in
run_line_magic(self, magic_name, line, _stack_depth)
    2324         kwargs['local_ns'] = sys._getframe(stack_depth).f_locals
    2325         with self.builtin_trap:
-> 2326             result = fn(*args, **kwargs)
    2327         return result
    2328

<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):

InternalError: (psycopg2.errors.DependentObjectsStillExist) cannot drop index
users_user_email_id_key because constraint users_user_email_id_key on table users requires it
HINT:  You can drop constraint users_user_email_id_key on table users instead.

[SQL: DROP INDEX users_user_email_id_key]
(Background on this error at: http://sqlalche.me/e/13/2j85)

```