

Universidad Técnica Nacional

Ingeniería del software

Aplicación de base de datos

Curso - 413

Tema

Diccionario de datos

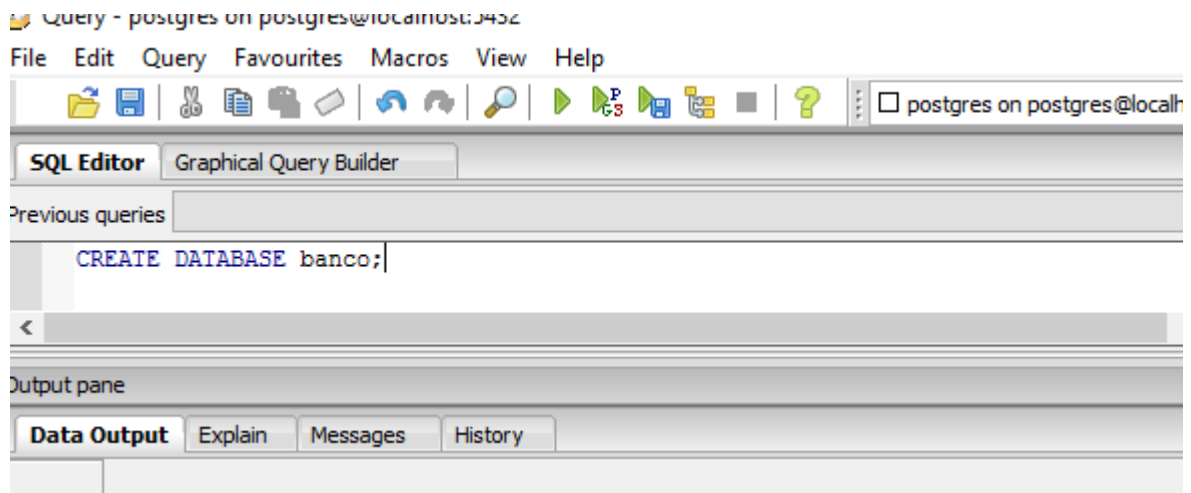
Prof.

Efrén Jiménez

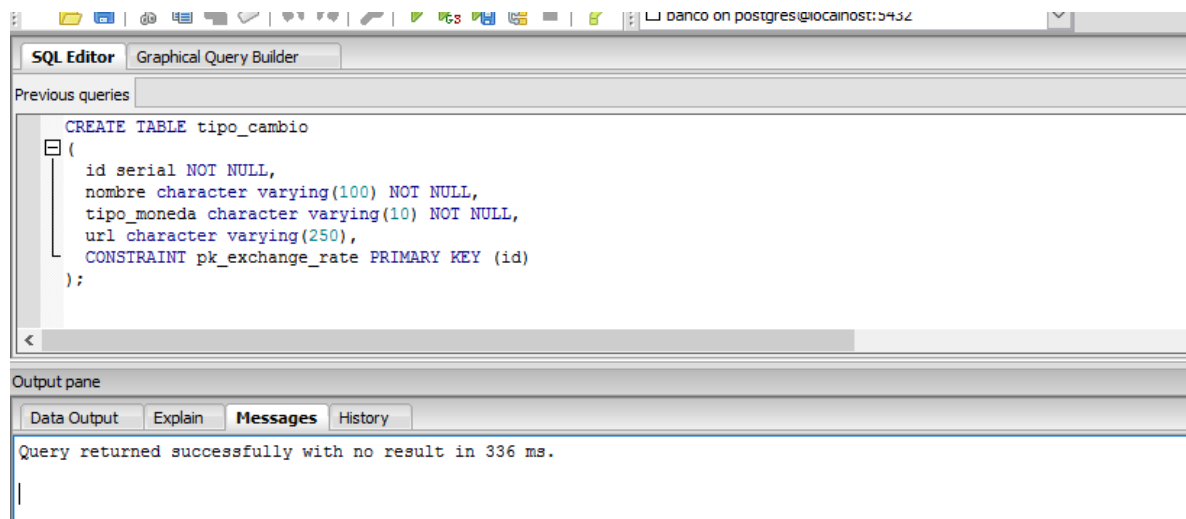
Enero 2019

Diccionario de Postgres

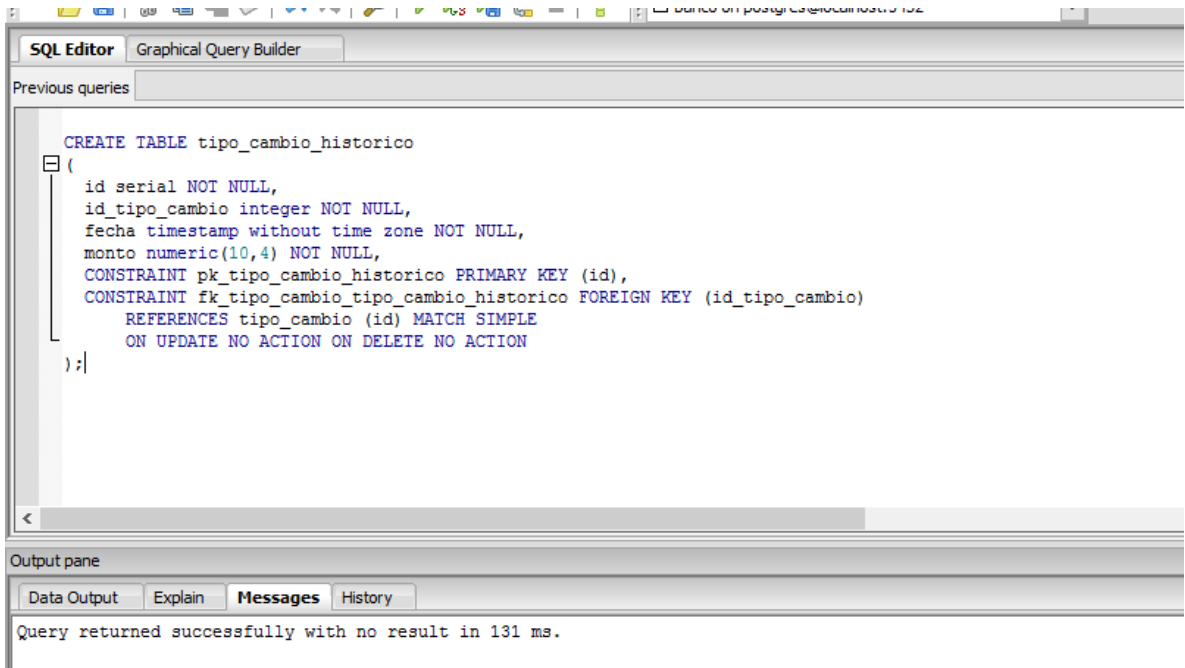
1. Crear una base de datos



2. Crear la tabla tipo_cambio



3. Crear la tabla tipo_cambio_historico



The screenshot shows the SQL Editor interface with the 'SQL Editor' tab selected. The 'Previous queries' section is empty. The main editor contains the following SQL code:

```
CREATE TABLE tipo_cambio_historico
(
  id serial NOT NULL,
  id_tipo_cambio integer NOT NULL,
  fecha timestamp without time zone NOT NULL,
  monto numeric(10,4) NOT NULL,
  CONSTRAINT pk_tipo_cambio_historico PRIMARY KEY (id),
  CONSTRAINT fk_tipo_cambio_tipo_cambio_historico FOREIGN KEY (id_tipo_cambio)
    REFERENCES tipo_cambio (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);|
```

The 'Output pane' at the bottom shows the 'Messages' tab selected, displaying the message: 'Query returned successfully with no result in 131 ms.'

4. Ejecutar el script de inserción 1



The screenshot shows the SQL Editor interface with the 'SQL Editor' tab selected. The 'Previous queries' section is empty. The main editor contains the following SQL code:

```
INSERT INTO tipo_cambio (nombre,tipo_moneda) values ('Dólar de los Estados Unidos','USD');
INSERT INTO tipo_cambio (nombre,tipo_moneda,url) values ('Colones Costa Rica','CRC','http://indicadoreseconomicos.bocr.fi.cr/indicadoreseconomicos/Cuadros/fmVerCatCuadro.aspx?idioma=1&CodCuadro=420370');
INSERT INTO tipo_cambio (nombre,tipo_moneda,url) values ('Dólar australiano','AUD','http://indicadoreseconomicos.bocr.fi.cr/indicadoreseconomicos/Cuadros/fmVerCatCuadro.aspx?idioma=1&CodCuadro=420768');
INSERT INTO tipo_cambio (nombre,tipo_moneda,url) values ('Libra Esterlina','GBP','http://indicadoreseconomicos.bocr.fi.cr/indicadoreseconomicos/Cuadros/fmVerCatCuadro.aspx?idioma=1&CodCuadro=420381');
INSERT INTO tipo_cambio (nombre,tipo_moneda,url) values ('Euro','EUR','http://indicadoreseconomicos.bocr.fi.cr/indicadoreseconomicos/Cuadros/fmVerCatCuadro.aspx?idioma=1&CodCuadro=42012');
INSERT INTO tipo_cambio (nombre,tipo_moneda,url) values ('Dólar Canadiense','CAD','http://indicadoreseconomicos.bocr.fi.cr/indicadoreseconomicos/Cuadros/fmVerCatCuadro.aspx?idioma=1&CodCuadro=420384');
```

The 'Output pane' at the bottom shows the 'Messages' tab selected, displaying the message: 'Query returned successfully: one row affected, 708 ms execution time.'

5. Ejecutar el script de inserción 2

The screenshot shows a SQL Editor window with a large INSERT script. The script consists of 30 INSERT statements into the 'tipo_cambio_historico' table, each with a unique timestamp and a value. The output pane at the bottom shows the message: 'Query returned successfully: one row affected, 708 ms execution time.'

```
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-01 17:36:46.425878', 545.5300);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-02 17:36:49.202075', 545.5300);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-03 17:36:51.385941', 545.6500);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-04 17:36:53.413416', 545.6500);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-05 17:36:55.461239', 545.6500);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-06 17:36:57.444601', 545.5100);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-07 17:36:59.469338', 545.2800);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-08 17:37:01.523814', 545.4000);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-09 17:37:03.611198', 545.4000);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-10 17:37:05.647168', 545.5000);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-11 17:37:07.660447', 545.5000);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-12 17:37:09.670501', 545.5000);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-13 17:37:11.698904', 545.4000);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-14 17:37:13.710303', 545.3600);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-15 17:37:15.754466', 543.1300);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-16 17:37:17.762872', 542.4100);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-17 17:37:19.775093', 542.2900);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-18 17:37:21.816866', 542.2900);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-19 17:37:23.857529', 542.2900);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-20 17:37:25.917206', 542.2000);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-21 17:37:27.927216', 542.1000);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-22 17:37:29.99115', 542.1100);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-23 17:37:31.991421', 542.2300);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-24 17:37:33.983803', 542.0600);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-25 17:37:35.98359', 542.0600);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-26 17:37:38.001147', 542.0600);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-27 17:37:40.014138', 542.0800);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-28 17:37:42.033957', 542.0900);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-29 17:37:44.032123', 542.0800);
INSERT INTO tipo_cambio_historico (id_tipo_cambio, fecha, monto) values (2, '2015-01-30 17:37:46.029704', 542.6400);
```

Query returned successfully: one row affected, 708 ms execution time.

Diccionario de datos

6. Para saber que bases de datos hay en el sistema:

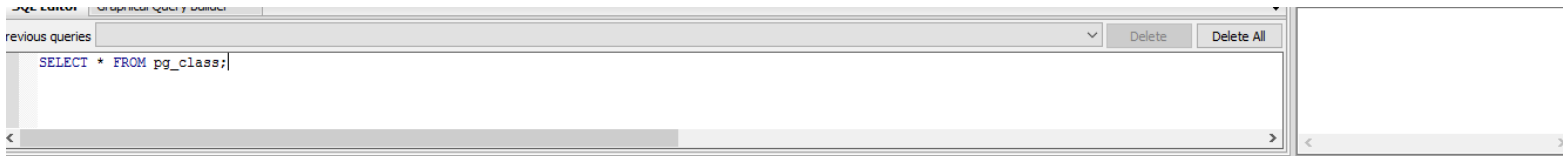
```
SELECT * FROM pg_database;
```

The screenshot shows the SQL Editor with the query 'SELECT * FROM pg_database;' and the resulting output table. The table has 14 columns: datname, name, datdba, encoding, collate, name, datatype, template, allowconn, connlimit, lastsysoid, frozenxid, minmxid, tablespace, and aclitem. The output shows 5 rows of data.

	datname	name	datdba	encoding	collate	name	datatype	template	allowconn	connlimit	lastsysoid	frozenxid	minmxid	tablespace	aclitem
1	template1		10	6	Spanish Costa Rica.1252	Spanish Costa Rica.1252	t	t		-1	12130	668	1	1663	{=c/postgres,postgr
2	template0		10	6	Spanish Costa Rica.1252	Spanish Costa Rica.1252	t	f		-1	12130	668	1	1663	{=c/postgres,postgr
3	postgres		10	6	Spanish Costa Rica.1252	Spanish Costa Rica.1252	f	t		-1	12130	668	1	1663	
4	banco		10	6	Spanish Costa Rica.1252	Spanish Costa Rica.1252	f	t		-1	12130	668	1	1663	
5	electronics labs		10	6	Spanish Costa Rica.1252	Spanish Costa Rica.1252	f	t		-1	12130	668	1	1663	

7. Para saber que tablas tengo en la base de datos actual:

```
SELECT * FROM pg_class;
```



previous queries Delete Delete All

```
SELECT * FROM pg_class;
```

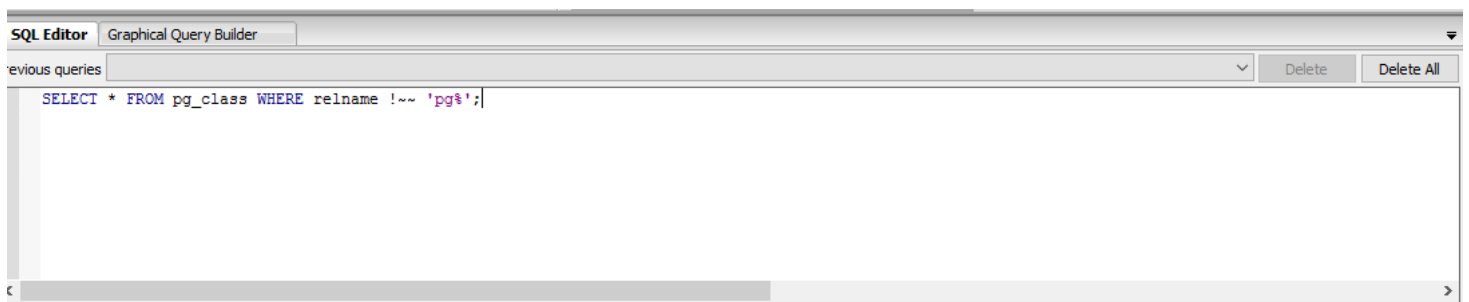
Output pane

Data Output Explain Messages History

	relname name	relnamespace oid	reltype oid	reloftype oid	relowner oid	relam oid	relfilenode oid	reltablespace oid	relpages integer	reltuples real	relallvisible integer	reltoastrelid oid	relhasindex boolean	relisshared boolean	relpersistenc "char"	relkind "char"	relnatt smallin
1	pg_statistic	11	10912	0	10	0	11867	0	15	399	15	2840	t	f	p	r	2
2	pg_type	11	71	0	10	0	0	0	8	339	8	0	t	f	p	r	3
3	pg_toast 16419	99	16423	0	10	0	16422	0	0	0	0	0	t	f	p	t	
4	pg_toast 16419 index	99	0	0	10	403	16424	0	1	0	0	0	f	f	p	i	
5	pg_toast 2619	99	11146	0	10	0	11869	0	2	10	2	0	t	f	p	t	
6	pg_toast 2619 index	99	0	0	10	403	11871	0	2	10	0	0	f	f	p	i	
7	pg_authid rolname index	11	0	0	10	403	0	1664	2	1	0	0	f	t	p	i	
8	pg_authid oid index	11	0	0	10	403	0	1664	2	1	0	0	f	t	p	i	
9	pg_attribute relid attnam index	11	0	0	10	403	0	0	17	2739	0	0	f	f	p	i	
10	pg_attribute relid attnum index	11	0	0	10	403	0	0	12	2739	0	0	f	f	p	i	
11	pg_toast 1255	99	11143	0	10	0	0	0	0	0	0	0	t	f	p	t	
12	pg_toast 1255 index	99	0	0	10	403	0	0	1	0	0	0	f	f	p	i	
13	pg_toast 2604	99	11140	0	10	0	11903	0	0	0	0	0	t	f	p	t	
14	pg_toast 2604 index	99	0	0	10	403	11905	0	1	0	0	0	f	f	p	i	
15	pg_attrdef adrelid adnum index	11	0	0	10	403	11906	0	1	0	0	0	f	f	p	i	

8. Lo mismo, pero sólo las definidas por el usuario, excluyendo las del sistema:

```
SELECT * FROM pg_class WHERE relname !~~ 'pg%';
```



SQL Editor Graphical Query Builder

previous queries Delete Delete All

```
SELECT * FROM pg_class WHERE relname !~~ 'pg%';
```

Output pane

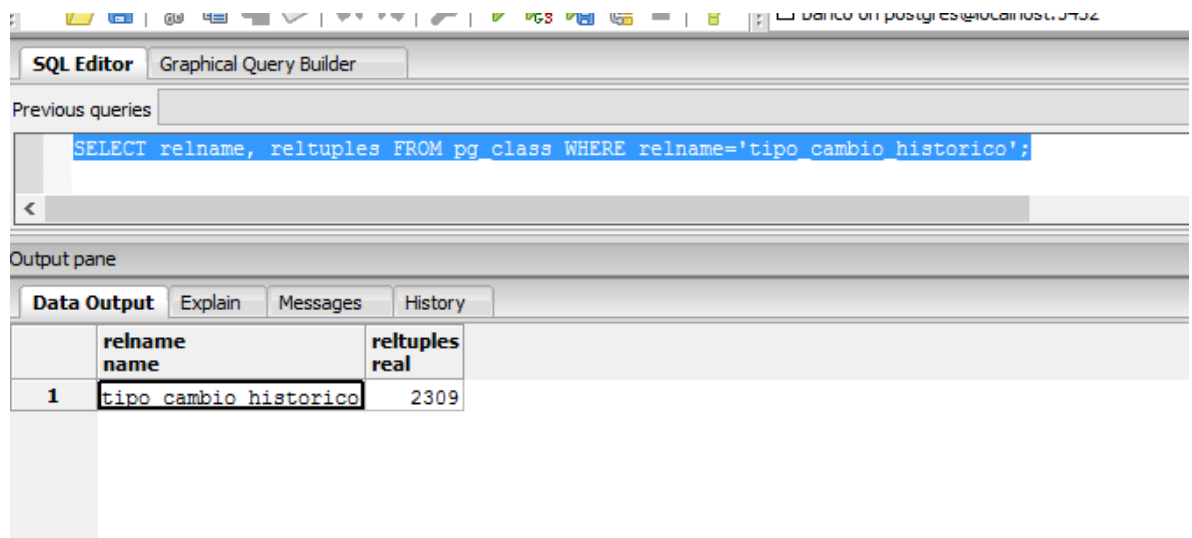
Data Output Explain Messages History

	relname name	relnamespace oid	reltype oid	reloftype oid	relowner oid	relam oid	relfilenode oid	reltablespace oid	relpages integer	reltuples real	relallvisible integer	reltoastrelid oid	relhasindex boolean	relisshared boolean	relpersistenc "char"	relkind "char"	relnatt smallin
1	tipo cambio id seq	2200	16398	0	10	0	16397	0	1	1	0	0	f	f	f	f	
2	pk exchange rate	2200	0	0	10	403	16403	0	1	0	0	0	f	f	f	f	
3	tipo cambio historico id seq	2200	16406	0	10	0	16405	0	1	1	0	0	f	f	f	f	
4	pk tipo cambio historico	2200	0	0	10	403	16411	0	9	2309	0	0	f	f	f	f	
5	tipo cambio	2200	16401	0	10	0	16399	0	0	0	0	0	t	f	f	f	
6	tipo cambio historico	2200	16409	0	10	0	16407	0	15	2309	0	0	t	f	f	f	
7	information schema catalog name	11585	11603	0	10	0	11602	0	0	0	0	0	f	f	f	f	
8	applicable roles	11585	11609	0	10	0	11608	0	0	0	0	0	f	f	f	f	
9	administrable role authorizations	11585	11613	0	10	0	11612	0	0	0	0	0	f	f	f	f	
10	attributes	11585	11616	0	10	0	11615	0	0	0	0	0	f	f	f	f	
11	character sets	11585	11620	0	10	0	11619	0	0	0	0	0	f	f	f	f	
12	check constraint routine usage	11585	11624	0	10	0	11623	0	0	0	0	0	f	f	f	f	
13	check constraints	11585	11628	0	10	0	11627	0	0	0	0	0	f	f	f	f	
14	collations	11585	11632	0	10	0	11631	0	0	0	0	0	f	f	f	f	
15	collation character set applicability	11585	11635	0	10	0	11634	0	0	0	0	0	f	f	f	f	

9. Si sólo queremos saber cuántos registros tiene una tabla, basta con preguntar:

```
SELECT relname, reltuples FROM pg_class WHERE relname='tipo_cambio_historico';
```

Solución=2309



10. A continuación, presentamos una forma de extraer información de las tablas de la base de datos. `pg_database` tiene las bases de datos, `pg_class` tiene las tablas, y `pg_attribute` tiene los campos. Si lo que queremos saber es el número de registros en una tabla determinada, basta con preguntar lo siguiente, conociendo la tabla, adicionalmente se obtienen los campos. Para, además, saber los tipos de los campos, habría que hacer un query a `pg_type` con los oids de los campos. ¿Puede construir la consulta para hacerlo?

```
SELECT relname, reltuples, attname, attnum  
FROM pg_class, pg_attribute  
WHERE pg_class.relname='tipo_cambio_historico'  
AND pg_attribute.attrelid=pg_class.oid  
AND attnum > 0  
ORDER BY attnum;
```

previous queries				
<pre> SELECT relname,reltuples,attname,attnum FROM pg_class,pg_attribute WHERE pg_class.relname='tipo_cambio_historico' AND pg_attribute.attrelid=pg_class.oid AND attnum > 0 ORDER BY attnum; </pre>				
Output pane				
<div> <div>Data Output</div> <div> <div>Explain</div> <div>Messages</div> <div>History</div> </div> </div>				
	relname name	reltuples real	attname name	attnum smallint
1	tipo cambio historico	2309	id	1
2	tipo cambio historico	2309	id tipo cambio	2
3	tipo cambio historico	2309	fecha	3
4	tipo cambio historico	2309	monto	4

11. Si lo que queremos es saber todo de todas las tablas de la base que no sean del sistema, sino del usuario, por supuesto ordenadas por oid, se hace lo siguiente, pero con el problema de que incluye los índices:

```

SELECT pg_class.oid, relname, reltuples, attname
  FROM pg_class, pg_attribute
 WHERE pg_class.relname !~~ '%pg%'
       AND pg_attribute.attrelid=pg_class.oid
       AND attnum > 0
 ORDER BY pg_class.oid;

```

SQL EditorGraphical Query Builder

previous queries

▼DeleteDelete

```
SELECT pg_class.oid,relname,reltuples,attname
FROM pg_class,pg_attribute
WHERE pg_class.relname !~~ '%$pg%'
AND pg_attribute.attrelid=pg_class.oid
AND attnum > 0
ORDER BY pg_class.oid;
```

Output pane

Data OutputExplainMessagesHistory

	oid oid	relname name	reltuples real	attname name
643	16399	tipo cambio	0	tipo moneda
644	16399	tipo cambio	0	url
645	16399	tipo cambio	0	id
646	16403	pk exchange rate	0	id
647	16405	tipo cambio historico id seq	1	sequence name
648	16405	tipo cambio historico id seq	1	last value
649	16405	tipo cambio historico id seq	1	start value
650	16405	tipo cambio historico id seq	1	increment by
651	16405	tipo cambio historico id seq	1	max value
652	16405	tipo cambio historico id seq	1	min value
653	16405	tipo cambio historico id seq	1	cache value
654	16405	tipo cambio historico id seq	1	log cnt
655	16405	tipo cambio historico id seq	1	is cycled

Práctica

1. Obtener todos los registros reales de la tabla tipo_cambio.
2. Obtener todos los usuarios y los roles de la base de datos.
3. Obtener todos los usuarios que son super usuarios o **rolsuper**.
4. Obtener todas las tablas del usuario.
5. Obtener todos los index creados por el usuario.
6. Obtener las columnas de la tabla tipo_cambio.
7. Obtener las columnas y tipos de la tabla tipo_cambio.
8. Obtener las tablas que al menos tienen un index.
9. Obtener todos los atributos de la base de datos.
10. Obtener todos los tipos de datos de la base de datos.