UNIVERSIDAD AUTÓNOMA DE MADRID

ESTRUCTURAS DE DATOS
(2018 - 2019)

PRÁCTICA 1

Alba Ramos
Miguel Díaz
Grupo: 1211

Madrid, 19/10/2018

TABLA DE CONTENIDOS

INTRODUCCIÓN

En esta memoria vamos a exponer cómo hemos llevado a cabo la primera práctica. Vamos a hablar de cómo hemos diseñado e implementado una versión pequeña de una base de datos de Twitter y las consultas simples que hemos hecho para ver si funcionaba. A continuación explicaremos los problemas que hemos tenido al hacer una base de datos más grande y con mucha más información. También las consultas complejas que hemos hecho sacadas de las sugeridas en el enunciado de la práctica más otras dos que nos hemos inventado nosotros.

DESARROLLO

Diseño

Lo primero que hay que hacer antes de programar nada es realizar un esquema entidad-relación que explique la situación que queremos representar: en nuestro caso, una red social, Twitter, con usuarios que siguen a otros usuarios, escriben y retweetean. Distinguimos dos entidades: usuarios y tweets. Cada usuario consta de su id, su nombre y su fecha de registro. Cada tweet consta de su id, su texto, su fecha y su localización. Hemos puesto la localización como un atributo de los tweets ya que un tweet puede ser escrito desde diferentes lugares.

A continuación hemos visto cómo se relacionan los usuarios entre sí: cada usuario puede seguir a muchos otros, y un usuario puede tener muchos seguidores. Encontramos, así, una relación M:N llamada “sigue a”, de participación parcial en ambos lados, ya que no es necesario tener seguidores ni seguir a nadie.

Además, los usuarios se relacionan con los tweets, ya que un usuario puede escribir muchos tweets y cada tweet es escrito por un usuario. Encontramos una relación 1:N a la que llamaremos “tweeteado por”, de participación parcial por los usuarios (ya que no es necesario que hayan escrito nada) y total por los tweets (pues todo tweet tiene que haber sido escrito por alguien).

Además existe la relación “retweeteado por”, que es M:N ya que un usuario puede retweetear muchas cosas, y un tweet puede ser retweeteado por muchos usuarios. Es de participación parcial en ambos lados, pues no es necesario ni retweetear ni que te retweeteen.

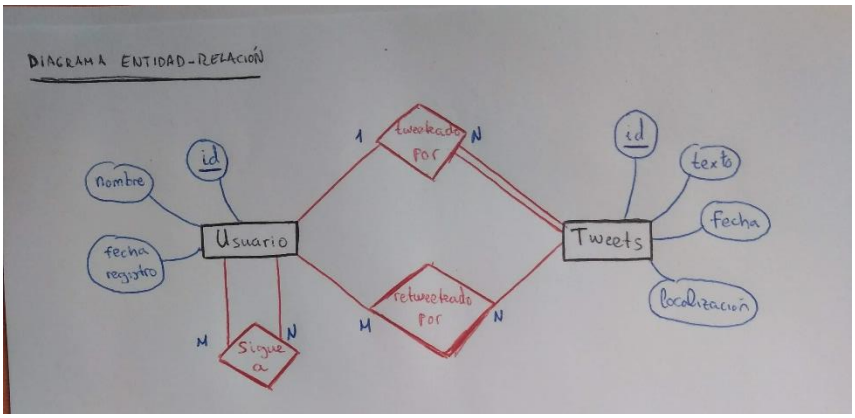
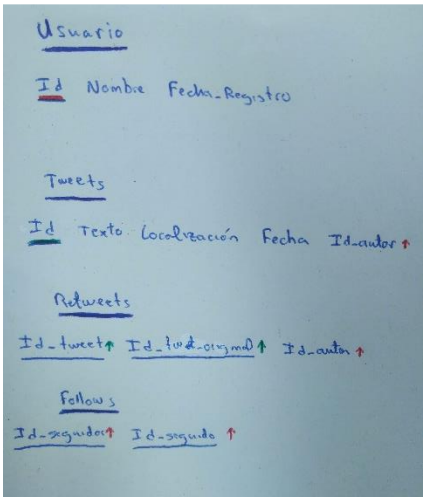


Diagrama Entidad-Relación

Una vez hecho esto, lo transformamos a esquema relacional. Para ello, de cada entidad normal sacamos una tabla con sus atributos y sus claves primarias: los ids.

Luego, de cada relación M:N sacamos otra tabla: tenemos así la tabla retweets, cuyos campos son una combinación del id del tweet original y del nuevo tweet (claves primarias en la tabla) y del usuario que retweetea, claves externas hacia el id del tweet y del usuario, respectivamente. También tenemos la tabla follows, cuyos campos son una combinación del id del usuario seguidor y su seguido (claves primarias de esta tabla y externas hacia el id de usuario).

Finalmente, la relación 1:N se incluye en la tabla tweets con una clave externa hacia el id de usuario, a la que llamamos id_autor. La incluimos en esta tabla y no en usuarios porque tiene participación total en la relación, y así evitamos valores NULL.



Esquema relacional

Implementación

A partir del esquema relacional hemos rellenado nuestra base de datos. Hemos creado las tablas solamente con claves primarias, las hemos rellenado y después hemos puesto las claves externas. Esto es así debido a que si metíamos claves externas antes de insertar los datos, al hacer inserciones se viola la integridad referencial (por ejemplo, al insertar cualquier tweet, el id del autor no existiría si teníamos la tabla de usuarios sin rellenar). Para evitar tener que rellenar las tablas en un orden específico, hemos decidido poner las claves externas al final.

Para los datos numéricos (ids) hemos elegido el tipo de datos integer. Para los nombres, varchar(50), ya que se trata de nombres cortos. Para el texto del tweet, varchar(250) porque son los máximos caracteres que permite twitter. Para las fechas hemos utilizado timestamp without timezone, para que permitiese guardar también la hora.

Consultas simples

En la primera consulta simple hemos buscado la lista de todos los ids de los usuarios que siguen al usuario que se llama bill mahler.

```
--busca los ids de la gente que sigue a bill mahler

select id_seguidor
from follows, usuarios
where id=id_seguido and nombre='bill_mahler';
```

Data Output | Explain | Mes

	id_seguidor integer
1	100
2	101
3	103

Código SQL y resultado de la consulta simple 1

En la siguiente consulta hemos buscado la lista de retweets que ha hecho el usuario llamado tora tora tora.

```
--busca los retweets de tora tora tora
select id_tweet
from usuarios, retweets
where id=id_usuario and nombre='tora_tora_tora';
```

Data Output | Explai

	id_tweet integer
1	202
2	203
3	208
4	212
5	215
6	219

Código SQL y resultado de la consulta simple 2

En la última consulta hemos buscado la lista de gente a la que sigue el usuario que se llama old fart.

```
-- busca la gente a la que sigue old fart
select id_seguido
from follows, usuarios
where id = id_seguidor and nombre = 'old_fart';
```

Data Output | Explain

	id_seguido integer
1	104
2	108
3	109

Código SQL y resultado de la consulta simple 3

Problemas

A lo largo de la implementación de la base de datos grande (p1_tweet) hemos tenido que cambiar varias cosas para que no hubiese errores a la hora de importar la información a las tablas. Tuvimos dos problemas.

El primer error que nos daba era que los ids de los usuarios eran demasiado grandes para ser almacenados en una variable int, por lo que lo cambiamos a bigint, que admite números mucho más grandes.

El otro error que se producía porque en la tabla de tweets teníamos las columnas en un orden diferente al que estaba organizado el fichero desde el que cargábamos los datos, por lo que tuvimos que cambiar de sitio la columna de la fecha en la que se realizaron los tweets.

Consultas complejas

Consulta a. En esta consulta hemos recuperado el tweet más antiguo y los 20 primeros tweets más recientes, ya que en este segundo caso la consulta devolvía 212077 resultados. Para encontrar el tweet más antiguo hemos hecho una consulta que seleccione todos los datos de la tabla tweets donde el campo fecha se corresponda con la fecha más antigua de la tabla. Para los tweets más recientes, basta cambiar el min(fecha) por el max(fecha) para seleccionar la fecha más reciente.

```
--tweet más antiguo;
select *
from tweets
where fecha in (
    select min(fecha)
    from tweets
);
```

	id bigint	texto character varying (250)	localizacion character varying (200)	fecha timestamp without time z	id_autor bigint
1	092236312	Barcelona. Guardiola resta ...	[null]	2009-01-02 22:07:58	15095537

Código SQL y resultado de la consulta a.1): tweet más antiguo

	id bigint	texto character varying (250)	localizacion character varying (200)	fecha timestamp without time zone	id_autor bigint
1	25286913	We're ruled by "institutions...	London	2018-09-20 16:05:32	16033
2	69303040	RT @brianmlucey: http://t.c...	London	2018-09-20 16:05:32	16033
3	96046848	"O #Germany, pale mother...	London	2018-09-20 16:05:32	16033
4	45992197	RT @jdportes: The UK & th...	London	2018-09-20 16:05:32	16033
5	58237696	Esta foto es bestialmente b...	Sevilla, Espaa	2018-09-20 16:05:32	78913
6	66478594	Te lo vas a perder Como ve...	Sevilla, Espaa	2018-09-20 16:05:32	78913
7	29788672	En serio Habra que coger a...	Sevilla, Espaa	2018-09-20 16:05:32	78913
8	77425921	Odo a un vendedor de Vod...	Sevilla, Espaa	2018-09-20 16:05:32	78913
9	61374464	No crea que se pudiera ser...	Sevilla, Espaa	2018-09-20 16:05:32	78913
10	52764416	Airbus "A400M" sobre vola...	Sevilla, Espaa	2018-09-20 16:05:32	78913
11	12972544	RT si quieres q duerma con...	Sevilla, Espaa	2018-09-20 16:05:32	78913
12	98057728	Aire Acondicionado moder...	Sevilla, Espaa	2018-09-20 16:05:32	78913
13	41130496	Me parto en Atencin al Clie...	Sevilla, Espaa	2018-09-20 16:05:32	78913
14	04451840	Llegar a Sonido del #ECI #...	Sevilla, Espaa	2018-09-20 16:05:32	78913
15	11614977	Cuando veas el avatar de "...	Sevilla, Espaa	2018-09-20 16:05:32	78913
16	45900800	@rocioromerope y no cabe...	Sevilla, Espaa	2018-09-20 16:05:32	78913
17	30016512	Las doce y ya no veo... Visi...	Sevilla, Espaa	2018-09-20 16:05:32	78913
18	38677248	no sabes q hacer Pues esc...	Sevilla, Espaa	2018-09-20 16:05:32	78913
19	38856960	Acabo de inventar el "Vient...	Sevilla, Espaa	2018-09-20 16:05:32	78913
20	65435136	Queris una Tenis un minut...	Sevilla, Espaa	2018-09-20 16:05:32	78913

Código SQL y 20 primeros resultados de la consulta a.2): tweets más recientes

Consulta b. En esta consulta hemos calculado el promedio de seguidores por usuario. Para ello, hemos concatenado las tablas follows y usuarios para poder hacer la

media: hemos sumado todas las filas de la tabla follows y esto lo hemos dividido entre la suma de los distintos usuarios existentes.

```
select count(*)/count(distinct id) as promedio
from follows inner join usuarios
on id_seguidor = id;
```

	promedio bigint
1	67

Código SQL y resultado de la consulta b): densidad de la red

Consulta d. En esta consulta hemos seleccionado los seguidores comunes a los usuarios de id 783214 y 2142731. Para ello, hemos metido dentro de la consulta principal otras dos: una que mostrase los seguidores del usuario 783214 y otra que mostrase los del usuario 2142731. Después hemos hecho la intersección de estos dos conjuntos y, en la consulta principal, hemos seleccionado todos los seguidores que se encuentren en este nuevo conjunto. De los 26 resultados que produce la consulta, se muestran 20.

```
select distinct id_seguidor
from follows
where id_seguidor in
(
    (
        select id_seguidor
        from follows
        where id_seguido=783214
    )
    intersect
    (
        select id_seguidor
        from follows
        where id_seguido=2142731
    )
);
```

	id_seguidor bigint		
1	3144281	11	59163001
2	13201312	12	87499473
3	14761739	13	114298656
4	15900167	14	114870386
5	16212685	15	137299903
6	17772404	16	156770766
7	19923515	17	208155948
8	22401772	18	251864117
9	31430065	19	392473852
10	48614388	20	400086439

Código SQL y resultado de la consulta d): 20 primeros seguidores comunes a 2 usuarios elegidos al azar

Consulta e. En esta consulta hemos buscado los usuarios comunes seguidos por los usuarios con los ids 13139 y 2929271. Para ello hemos hecho la intersección de dos tablas en las que se muestran los ids de los usuarios a los que siguen cada uno de los usuarios.

```sql
--e)usuarios comunes seguidos por dos usuarios dados: 13139 y 2929271

select distinct id_seguido
from follows
where id_seguido in
(

        (
                select id_seguido
                from follows
                where id_seguidor=13139
        )
        intersect
        (
                select id_seguido
                from follows
                where id_seguidor=2929271
        )
);
```

Data Output | Explain

	id_seguido bigint
1	19017608
2	37883400
3	760839
4	6931722
5	362565198
6	15234407
7	18944456
8	15453259
9	62237307

Código SQL y resultado de la consulta e): seguidos comunes a 2 usuarios elegidos al azar

Consulta g. En esta consulta hemos buscado cuál es el usuario con más seguidores. Para ello, hemos hecho una tabla con el id y el número de seguidores de cada usuario, y de ahí hemos seleccionado el id que tenía el valor máximo del recuento de seguidores.

```sql
--g) Usuario mas seguido
select nombre
from usuarios
where id =
(
        select id_seguido
        from
        (
                select id_seguido, count(*)
                from follows
                group by id_seguido
        ) as num_seguidos
        where count = (select max(count) from
                (
                        select id_seguido, count(*)
                        from follows
                        group by id_seguido
                ) as num_seguidos
        )
)
```

Data Output | Explain | Messages

	nombre character varying(200)
1	Atleti

Código SQL y resultado de la consulta g): El usuario más seguido

Consulta h. En esta consulta hemos buscado el usuario que más tweets ha escrito. Para ello, hemos hecho una tabla con el id del autor y el número de tweets que tiene cada uno. De ahí hemos seleccionado el que tenía un mayor número de tweets escritos.

```sql
--h) usuario que más tweets ha escrito;

select nombre
from usuarios
where id =
(
	select id_autor
	from
	(
		select id_autor, count(*)
		from tweets
		group by id_autor
	)as id
	where count =
	(
		select max(count)
		from
		(
			select id_autor, count(*)
			from tweets
			group by id_autor
		) as autor_max
	)
)
```

Data Output | Explain | Messages

	nombre character varying(200)
1	Atleti

Código SQL y resultado de la consulta h): Usuario que más tweets ha escrito

Consultas nuevas

Consulta 1. En esta consulta hemos recuperado la información de todos los tweets en orden cronológico del usuario más antiguo del sistema. Para ello, hemos concatenado las tablas tweets y usuarios, hemos seleccionado el usuario con fecha de registro más antigua y hemos ordenado sus tweets cronológicamente de más antiguos a más recientes. De los 162 resultados, se muestran 20.

```sql
select *
from usuarios, tweets
where usuarios.id = id_autor
    and fecha_registro in (
        select min(fecha_registro)
        from usuarios
    )
    order by (fecha) asc
```

	id bigint	nombre character varying (200)	fecha_registro timestamp without time zone	id bigint	texto character varying (250)	localizacion character varying (200)	fecha timestamp without time zone	id_autor bigint
1	13139	furilo	2006-11-19 23:38:11	12290560	La historia de Fitbit hasta e...	Madrid, Spain	2015-06-18 19:41:56	13139
2	13139	furilo	2006-11-19 23:38:11	08856064	#SwissLeaks by @icij gana...	Madrid, Spain	2015-06-19 10:41:30	13139
3	13139	furilo	2006-11-19 23:38:11	78453249	@xavijam se ha ido de cam...	Madrid, Spain	2015-06-19 15:38:13	13139
4	13139	furilo	2006-11-19 23:38:11	49392384	Detalles judiciales sobre la ...	Madrid, Spain	2015-06-20 11:51:37	13139
5	13139	furilo	2006-11-19 23:38:11	56541184	Paso Stelvio, from above ht...	Madrid, Spain	2015-06-20 12:07:54	13139
6	13139	furilo	2006-11-19 23:38:11	96026112	@ikerarmentia @rubencar...	Madrid, Spain	2015-06-22 08:13:59	13139
7	13139	furilo	2006-11-19 23:38:11	40449024	@bomberstudios tampoco ...	Madrid, Spain	2015-06-23 13:20:55	13139
8	13139	furilo	2006-11-19 23:38:11	10708996	Muy buena pinta esta app ...	Madrid, Spain	2015-06-23 18:17:53	13139
9	13139	furilo	2006-11-19 23:38:11	16044800	Los mejores juegos de mes...	Madrid, Spain	2015-06-24 00:12:35	13139
10	13139	furilo	2006-11-19 23:38:11	08057856	@agustinjv yo en madrid y...	Madrid, Spain	2015-06-24 15:08:59	13139
11	13139	furilo	2006-11-19 23:38:11	60531456	@juanmilleiro me siento or...	Madrid, Spain	2015-06-24 17:01:25	13139
12	13139	furilo	2006-11-19 23:38:11	33229056	@aldamiz jajaja. por otro la...	Madrid, Spain	2015-06-24 19:29:34	13139
13	13139	furilo	2006-11-19 23:38:11	89588992	@angeljimenez (y si te enc...	Madrid, Spain	2015-06-24 19:59:02	13139
14	13139	furilo	2006-11-19 23:38:11	03725824	@angeljimenez yayayaya e...	Madrid, Spain	2015-06-24 21:12:04	13139
15	13139	furilo	2006-11-19 23:38:11	65953537	@Guillermolo echan a grecia	Madrid, Spain	2015-06-28 23:41:15	13139
16	13139	furilo	2006-11-19 23:38:11	67996672	Diplomacia Chat https://t.c...	Madrid, Spain	2015-06-29 11:40:33	13139
17	13139	furilo	2006-11-19 23:38:11	25148416	@carlosalonso ojo que no ...	Madrid, Spain	2015-06-29 11:59:18	13139
18	13139	furilo	2006-11-19 23:38:11	18985728	@carlosalonso "proteger u...	Madrid, Spain	2015-06-29 12:02:16	13139
19	13139	furilo	2006-11-19 23:38:11	63307008	@carlosalonso (las consecu...	Madrid, Spain	2015-06-29 12:36:44	13139
20	13139	furilo	2006-11-19 23:38:11	17177344	@carlosalonso al contrario,...	Madrid, Spain	2015-06-29 13:03:58	13139

Código SQL y 20 primeros resultados de la consulta inventada 1

Consulta 2. En esta consulta hemos hecho una búsqueda de cuál es el usuario que más veces ha retweeteado. Para ello hemos juntado la columna de ids de usuarios con la de ids de tweets de la tabla retweets y hemos contado cuántas veces se repetían los autores. Luego buscamos el nombre del autor que se haya repetido más veces.

```
--Inventado: usuario que más retweets ha hecho



select nombre
from usuarios
where id =
(
	select id_autor
	from
	(
		select id_autor, count(*)
			from
			(
				select tweets.id_autor, retweets.id_tweet
				from tweets
				join retweets on tweets.id = retweets.id_tweet
			)as usuariosr
			group by id_autor
	)as usuariosr
	where count =
	(
		select max(count) from
		(
			select id_autor, count(*)
			from
			(
				select tweets.id_autor, retweets.id_tweet
				from tweets
				join retweets on tweets.id = retweets.id_tweet
			)as usuariosr
			group by id_autor
		)as rec

	)
)
```

Data Output | Explain | Messages

	nombre character varying(200)
1	OmarMedhouni475

Código SQL y resultado de la consulta inventada 2

CONCLUSIONES

Al realizar esta práctica, hemos creado una base de datos robusta que almacena una gran cantidad de datos de la red social twitter. Hemos realizado consultas para aprender el manejo de grandes cantidades de datos y familiarizarnos con el lenguaje SQL. En definitiva, hemos mejorado nuestras habilidades de diseño e implementación y hemos conseguido desarrollar una base de datos que nos será de utilidad para futuras prácticas.