

PROYECTO 2.1

07/02/2022

PROYECTO BBDD EN SQL

Proyecto BBDD de una aplicación multiplataforma
en SQL Developer.

Realizado por Raúl Benítez Ordóñez.

1. EXPLICACIÓN PROYECTO.

Para empezar, vamos a explicar detalladamente en qué consiste la aplicación para la que estamos diseñando la base de datos.

Es necesario saber cómo va a trabajar nuestra aplicación para posteriormente conocer y entender mejor los tipos de datos o restricciones que le estamos dando a nuestra base de datos respecto de esta.

La aplicación que queremos desarrollar es un servicio de música y podcasts, que te da acceso a millones de canciones y podcasts de creadores de todo el mundo.

En la actualidad, disponemos de muchas aplicaciones de este estilo, por ejemplo, Spotify es una de las aplicaciones top en este ámbito, pero nuestra aplicación cuenta con muchas más cosas que esta.



Nuestro principal objetivo es darles a nuestros usuarios **LA MEJOR CALIDAD** en todos los sentidos:

Los mejores podcasts, las canciones más recientes, la mejor calidad de audio y los mejores precios.

El funcionamiento de la aplicación sería el siguiente:

Primero de todo necesitaríamos registrarnos para tener una cuenta en la plataforma, y ahí, elegir un tipo de suscripción.

El tipo de suscripción puede ser gratis o de pago. La suscripción de pago ofrece la posibilidad de escuchar canciones y podcasts días o semanas antes de que salgan públicamente en la aplicación, y, elimina completamente la publicidad entre canciones y podcasts.

Después, una vez creada la cuenta, ya tendríamos acceso a las siguientes características:

- Un buscador para explorar por las diferentes canciones y podcasts que podemos escuchar en la aplicación.
- Una biblioteca para almacenar playlists (listas de canciones y podcasts) personalizadas o incluso almacenar canciones y podcasts por separado.
- Un historial donde se mostrarán las últimas canciones y podcasts escuchados por el usuario en los últimos 7 días, por si quiere volver a escuchar algo que no haya guardado previamente.
- Un administrador de usuario donde podrá gestionar sus datos personales, compartir su perfil con otros usuarios e incluso cambiar de suscripción cuando desee.

2. CREACIÓN TABLAS.

En este apartado vamos a ver cómo creamos las tablas y los diferentes tipos de datos para nuestra base de datos.

Para empezar, vamos a declarar cuales serían las principales tablas que tendríamos que distinguir en nuestra base de datos.

Las tablas serían:

- Tabla “Usuario” donde vamos a registrar a todos los usuarios registrados en la aplicación.
- Tabla “Suscripción” donde vamos a registrar el tipo de suscripción de cada usuario y a su vez, reunir más datos del usuario.
- Tabla “Biblioteca” donde vamos a registrar todas las bibliotecas personales de todos los usuarios.
- Tabla “Historial” donde se vamos a registrar el cajón “historial” de cada usuario.
- Tabla “Playlist” donde vamos a registrar los distintos tipos de playlists de los usuarios con las canciones o podcasts que contienen.
- Tabla “Discos” donde vamos a almacenar todos los álbumes, ep’s o singles que podemos escuchar hasta la fecha en la aplicación.
- Tabla “Canciones” donde vamos a almacenar cada una de las canciones de todos esos discos.
- Tabla “Podcast” donde vamos a almacenar todos los podcasts disponibles para escuchar en la aplicación.

Una vez, distinguidas claramente las tablas con las que vamos a trabajar y su función dentro de la BBDD, ahora necesitamos declarar cada campo y tipo de dato que va a contener cada tabla.

La tabla “Usuario” tendría estos campos:

```
create table USUARIO
(  IdUsuario NUMBER(5,0) CONSTRAINT pkidusuario primary key,
  NombreUsuario VARCHAR2(15) CONSTRAINT unnomusuario unique,
  FechaNacimiento DATE CONSTRAINT nnfnacusuario not null,
  PaisOrigen VARCHAR2(30) CONSTRAINT nnpaisorigusuario not null,
  NumTlf VARCHAR2(15) CONSTRAINT utelfusuario unique,
  FotoUsuario BLOB
);
```

La tabla “Suscripción” tendría estos campos:

```
create table SUSCRIPCION
(  IdSuscripcion NUMBER(5,0) CONSTRAINT pkidsuscripcion primary key,
  CorreoElectronico VARCHAR2(100) CONSTRAINT ucorreousuario unique,
  TipoSuscripcion VARCHAR2(10) CONSTRAINT nntiposuscripcion not null,
  EntidadBancaria VARCHAR2(30) CONSTRAINT nnentidadbancaria not null,
  CuentaBancaria CHAR(24) CONSTRAINT nncuentabancaria not null,
  PrecioSuscripcion NUMBER(4,2) CONSTRAINT nnpreciosuscrip not null,
  MesesSuscrito NUMBER(4,0) CONSTRAINT nmesessuscrip not null
);
```

La tabla “Biblioteca” tendría estos campos:

```
create table BIBLIOTECA
( IdBiblioteca NUMBER(5,0) CONSTRAINT pkidbiblioteca primary key,
  CONSTRAINT fkbibliotecausuar foreign key (IdBiblioteca) REFERENCES USUARIO(IdUsuario),
  CreadorBiblioteca VARCHAR2(15) CONSTRAINT nncreadorbiblioteca not null,
  CONSTRAINT fkbibliotecausuario foreign key (CreadorBiblioteca) REFERENCES USUARIO(NombreUsuario) );
```

La tabla “Historial” tendría estos campos:

```
create table HISTORIAL
( IdHistorial NUMBER(5,0) CONSTRAINT pkidhistorial primary key,
  CONSTRAINT fkidhistorial foreign key (IdHistorial) REFERENCES USUARIO(IdUsuario),
  CreadorHistorial VARCHAR2(15) CONSTRAINT nncreadorhistorial not null,
  CONSTRAINT fkhistorialusuar foreign key (CreadorHistorial) REFERENCES USUARIO(NombreUsuario) );
```

La tabla “Playlist” tendría estos campos:

```
create table PLAYLIST
( IdPlaylist NUMBER(10,0) CONSTRAINT pkidplaylist primary key,
  CONSTRAINT fkplaylistbiblioteca foreign key (IdPlaylist) REFERENCES BIBLIOTECA(IdBiblioteca),
  NombrePlaylist VARCHAR2(50) CONSTRAINT nnombplaylist not null,
  CreadorPlaylist VARCHAR2(15) CONSTRAINT nncreadorplaylist not null,
  CONSTRAINT fkplaylistusuar foreign key (CreadorPlaylist) REFERENCES USUARIO(NombreUsuario),
  Disco1 NUMBER(20,0),
  Disco2 NUMBER(20,0),
  Disco3 NUMBER(20,0),
  Disco4 NUMBER(20,0),
  Disco5 NUMBER(20,0),
  CONSTRAINT fkdisco1playlist foreign key (Disco1) REFERENCES DISCOS(IdDisco),
  CONSTRAINT fkdisco2playlist foreign key (Disco2) REFERENCES DISCOS(IdDisco),
  CONSTRAINT fkdisco3playlist foreign key (Disco3) REFERENCES DISCOS(IdDisco),
  CONSTRAINT fkdisco4playlist foreign key (Disco4) REFERENCES DISCOS(IdDisco),
  CONSTRAINT fkdisco5playlist foreign key (Disco5) REFERENCES DISCOS(IdDisco) );
```

La tabla “Discos” tendría estos campos:

```
create table DISCOS
( IdDisco NUMBER(20,0) CONSTRAINT pkiddisco primary key,
  NombreDisco VARCHAR2(50) CONSTRAINT nnnombdisco not null,
  GeneroDisco VARCHAR2(50) CONSTRAINT nngendisco not null,
  Artista VARCHAR2(30) CONSTRAINT nnartistadisco not null,
  FechaLanzamiento DATE CONSTRAINT nnflanzdisco not null,
  TipoDisco VARCHAR2(10) CONSTRAINT nntipodisco not null );
```

La tabla “Canciones” tendría estos campos:

```
create table CANCIONES
( IdCancion NUMBER(38,0) CONSTRAINT nnidcancion not null,
  CONSTRAINT fkcanciondisco foreign key (IdCancion) REFERENCES DISCOS(IdDisco),
  NombreCancion VARCHAR2(50) CONSTRAINT nnnombcancion not null,
  GeneroCancion VARCHAR2(50) CONSTRAINT nngencancion not null,
  Artista VARCHAR2(25) CONSTRAINT nnartistacancion not null );
```

La tabla “Podcast” tendría estos campos:

```
create table PODCAST
( IdPodcast NUMBER(10,0) CONSTRAINT pkidpodcast primary key,
  NombrePodcast VARCHAR2(50) CONSTRAINT nnnombpodcast not null,
  Podcaster VARCHAR2(50) CONSTRAINT npodcaster not null,
  FechaEstreno DATE CONSTRAINT nnfecestreno not null,
  Tematica VARCHAR2(50) CONSTRAINT nntematica not null );
```

3. MODIFICACIONES.

Entonces, una vez definidos los campos con sus tipos de datos y restricciones, vamos a ver cómo podemos aplicar algún tipo de modificación para que todos los números de teléfono o los correos electrónicos sigan una estructura.

Vamos a aplicar modificaciones para ciertos campos en las tablas Usuario y Suscripción.

Para estas restricciones vamos a utilizar la sentencia **ALTER TABLE** para modificar la estructura de una tabla que ya existe.

Para añadir una nueva restricción lo vamos a realizar siempre con esta estructura:

ALTER TABLE tab1 ADD CONSTRAINT c1 UNIQUE (col3)

Añadimos la siguiente restricción para el nombre de usuario:

```
alter table USUARIO add CONSTRAINT cknombusuario  
CHECK (REGEXP_LIKE(NombreUsuario, '^[[:upper:]]{1}[:,graph:]+$');
```

Con esta restricción estamos indicando que el nombre de Usuario tiene que empezar sí o sí con una letra mayúscula, y después especificamos que puede seguir hasta el final de la cadena, con el operador +\$, con la clase de caracteres [:graph:] que incluye letras mayúsculas, minúsculas, números y hasta signos de puntuación, como guiones, barras bajas, etc.

Entonces, aplicamos la restricción en nuestra base de datos y vamos a comprobar que esta funciona correctamente.

The screenshot shows the SQL Developer interface with the 'USUARIO' table. The table has columns: IDUSUARIO, NOMBREUSUARIO, FECHANACIMIENTO, PAISORIGEN, NUMTLF, and FOTOUSUARIO. The first row has values: 1, 'rrbzn', 12/03/22, España, 651652838, and (BLOB). Below the table, the 'Mensajes - Log' window shows the following messages:

```
INSERT INTO "PROYECTOSQL"."USUARIO" (IDUSUARIO, NOMBREUSUARIO, FECHANACIMIENTO, PAISORIGEN, NUMTLF, FOTOUSUARIO) VALUES ('1', 'rrbzn', TO_DATE('2022-03-12 12:56:54', 'YYYY-MM-DD HH24:MI:SS'),  
ORA-02290: check constraint (PROYECTOSQL.CKNOMBUSUARIO) violated  
ORA-06512: at line 1
```

Se ha producido un error al guardar los cambios en la tabla "PROYECTOSQL"."USUARIO":
Fila 1: ORA-02290: check constraint (PROYECTOSQL.CKNOMBUSUARIO) violated
ORA-06512: at line 1

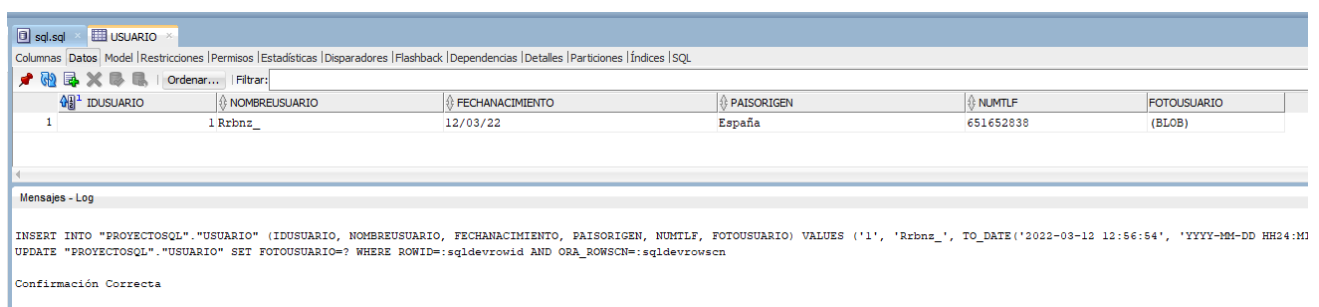
The second screenshot shows the same table with the first row having values: 1, '1Rrbzn', 12/03/22, España, 651652838, and (BLOB). The 'Mensajes - Log' window shows the following messages:

```
INSERT INTO "PROYECTOSQL"."USUARIO" (IDUSUARIO, NOMBREUSUARIO, FECHANACIMIENTO, PAISORIGEN, NUMTLF, FOTOUSUARIO) VALUES ('1', '1Rrbzn', TO_DATE('2022-03-12 12:56:54', 'YYYY-MM-DD HH24:MI:SS'),  
ORA-02290: check constraint (PROYECTOSQL.CKNOMBUSUARIO) violated  
ORA-06512: at line 1
```

Se ha producido un error al guardar los cambios en la tabla "PROYECTOSQL"."USUARIO":
Fila 1: ORA-02290: check constraint (PROYECTOSQL.CKNOMBUSUARIO) violated
ORA-06512: at line 1

Como veis, si al añadir una fila en la BBDD empezamos el usuario con una letra minúscula o con un número, nos salta un mensaje diciendo que estamos violando la restricción que acabamos de añadir a la tabla.

Sin embargo, si empezamos con una letra mayúscula y después añadimos lo que queramos, nos deja perfectamente insertar la fila sin ningún problema.



The screenshot shows the SQL Developer interface with a table named 'USUARIO' in the 'PROYECTOSQL' schema. The table has columns: IDUSUARIO, NOMBREUSUARIO, FECHANACIMIENTO, PAISORIGEN, NUMTLF, and FOTOUSUARIO. A single row is displayed with values: 1, Rrbns_, 12/03/22, España, 651652838, and (BLOB). Below the table, the 'Mensajes - Log' window shows the following SQL statement and its execution result:

```
INSERT INTO "PROYECTOSQL"."USUARIO" (IDUSUARIO, NOMBREUSUARIO, FECHANACIMIENTO, PAISORIGEN, NUMTLF, FOTOUSUARIO) VALUES ('1', 'Rrbns_', TO_DATE('2022-03-12 12:56:54', 'YYYY-MM-DD HH24:MI'))  
UPDATE "PROYECTOSQL"."USUARIO" SET FOTOUSUARIO=? WHERE ROWID=:sqldevrowid AND ORA_ROWSCN=:sqldevrowscn
```

Confirmación Correcta

Ahora, para el número de teléfono vamos a utilizar la siguiente restricción:

```
alter table USUARIO add CONSTRAINT cknumtlfusuario  
CHECK (REGEXP_LIKE(NumTlf, '^[:digit:]{1,3}[:space:]{1}[:digit:]+$'));
```

Aquí estamos indicando que el número de tlf va a contener el prefijo del país del que proviene separado con un espacio obligatorio del número del tlf como tal.

Como el prefijo puede variar de uno a tres números, ponemos el rango de {1,3} para poner por ejemplo el prefijo 7 de Rusia o el prefijo 34 de España, el espacio obligatorio que se indica con la clase [:space:] y, por último el número de tlf como tal hasta el final de la cadena.

Entonces, comprobamos la modificación que hemos hecho:

sql.sql x USUARIO

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

IDUSU...	NOMBREU...	FECHANA...	PAISORIG...	NUMTLF	FOTOUSUA...
+1	Rrbnz	12/03/22	España	34651652838	(BLOB)

Mensajes - Log

```

INSERT INTO "PROYECTOSQL"."USUARIO" (IDUSUARIO, NOMBREUSUARIO, FECHANACIMIENTO, PAISORIGEN, NUMTLF, FOTOUSUARIO) VALUES ('1', 'Rrbnz', '34651652838', empty_blob())
ORA-02290: check constraint (PROYECTOSQL.CKNUMTLFUSUARIO) violated
ORA-06512: at line 1

Se ha producido un error al guardar los cambios en la tabla "PROYECTOSQL"."USUARIO":
Fila 1: ORA-02290: check constraint (PROYECTOSQL.CKNUMTLFUSUARIO) violated
ORA-06512: at line 1

```

sql.sql x USUARIO

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

IDUSU...	NOMBREU...	FECHANA...	PAISORIG...	NUMTLF	FOTOUSUA...
+1	Rrbnz	12/03/22	España	3444 651652838	(BLOB)

Mensajes - Log

```

INSERT INTO "PROYECTOSQL"."USUARIO" (IDUSUARIO, NOMBREUSUARIO, FECHANACIMIENTO, PAISORIGEN, NUMTLF, FOTOUSUARIO) VALUES ('1', 'Rrbnz', '3444 651652838', empty_blob())
ORA-02290: check constraint (PROYECTOSQL.CKNUMTLFUSUARIO) violated
ORA-06512: at line 1

Se ha producido un error al guardar los cambios en la tabla "PROYECTOSQL"."USUARIO":
Fila 1: ORA-02290: check constraint (PROYECTOSQL.CKNUMTLFUSUARIO) violated
ORA-06512: at line 1

```

sql.sql x USUARIO

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

IDUSU...	NOMBREU...	FECHANA...	PAISORIG...	NUMTLF	FOTOUSUA...
1	1 Rrbnz	12/03/22	España	34 651652838	(BLOB)

Mensajes - Log

```

INSERT INTO "PROYECTOSQL"."USUARIO" (IDUSUARIO, NOMBREUSUARIO, FECHANACIMIENTO, PAISORIGEN, NUMTLF, FOTOUSUARIO) VALUES ('1', 'Rrbnz', TO_DATE('2022-03-12 13:19:58', 'YYYY-MM-DD HH24:MI:SS'), 'España', '34 651652838', empty_blob())
UPDATE "PROYECTOSQL"."USUARIO" SET FOTOUSUARIO=? WHERE ROWID=:sqldevrowid AND ORA_ROWSCN=:sqldevrowscn
Confirmación Correcta

```

Como veis, si no pongo el espacio o incluso pongo un prefijo de 4 dígitos, me salta la restricción para el número de tlf, pero si lo pongo correctamente, me deja guardar los cambios sin ningún problema.

Por último, vamos a ver dos restricciones que podemos añadir a la tabla Suscripción.

Vamos a modificar los siguientes campos para que el correo electrónico y la cuenta bancaria sigan un patrón.

El patrón que va a seguir el correo electrónico es el siguiente:

```
alter table SUSCRIPCION add CONSTRAINT ckcorreoelectronico  
CHECK (REGEXP_LIKE(CorreoElectronico, '^[[:alpha:]]{1}[[:graph:]]{4,25}@[[:alpha:]]{5,19}\.[[:alpha:]]{2,3}');
```

Va a empezar obligatoriamente con un carácter alfabético (ya sea mayúscula o minúscula), después podrá tener desde 4 a 25 caracteres de la clase de caracteres que explicamos antes y después de eso, va a contener obligatoriamente un @.

Esto no lo habíamos visto antes, pero podemos indicar específicamente que una cadena empieza por un dígito o carácter específico o incluso que lo contenga a mitad de la cadena.

Después del @, va a contener la organización del correo electrónico, como puede ser gmail, yahoo, outlook, etc. y, por último, un punto obligatorio y el tipo de correo electrónico, por ejemplo .com o .es.

En la comprobación, si por ejemplo empezamos por un signo de puntuación o incumplimos el máximo o mínimo de caracteres, nos va a saltar el error, pero si nos ajustamos perfectamente a la cadena, nos deja guardar los datos sin ningún problema.

sql.sql x SUSCRIPCION x

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

	IDSUSCRI...	CORREOELECTRONICO	TIPOSUSCRIPCION	ENTIDAD...	CUENTABANCARIA	PRECIOS...	MESESSU...
+1	1	lrrbnz@hotmail.com	Normal	BBVA	ES22222222222222222222	4,99	12

Mensajes - Log

```

INSERT INTO "PROYECTOSQL"."SUSCRIPCION" (IDSUSCRIPCION, CORREOELECTRONICO, TIPOSUSCRIPCION, ENTIDADBANCARIA, CUENTABANCARIA, PRECIOSUSCRIPCION, MESESSUSCRITO)
ORA-02290: check constraint (PROYECTOSQL.CKCORREOELECTRONICO) violated
ORA-06512: at line 1

Se ha producido un error al guardar los cambios en la tabla "PROYECTOSQL"."SUSCRIPCION":
Fila 1: ORA-02290: check constraint (PROYECTOSQL.CKCORREOELECTRONICO) violated
ORA-06512: at line 1

```

sql.sql x SUSCRIPCION x

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

	IDSUSCRI...	CORREOELECTRONICO	TIPOSUSCRIPCION	ENTIDAD...	CUENTABANCARIA	PRECIOS...	MESESSU...
+1	1	rbz@hotmail.com	Normal	BBVA	ES22222222222222222222	4,99	12

Mensajes - Log

```

INSERT INTO "PROYECTOSQL"."SUSCRIPCION" (IDSUSCRIPCION, CORREOELECTRONICO, TIPOSUSCRIPCION, ENTIDADBANCARIA, CUENTABANCARIA, PRECIOSUSCRIPCION, MESESSUSCRITO) VALUES ('1', 'rbz
ORA-02290: check constraint (PROYECTOSQL.CKCORREOELECTRONICO) violated
ORA-06512: at line 1

Se ha producido un error al guardar los cambios en la tabla "PROYECTOSQL"."SUSCRIPCION":
Fila 1: ORA-02290: check constraint (PROYECTOSQL.CKCORREOELECTRONICO) violated
ORA-06512: at line 1

```

sql.sql x SUSCRIPCION x

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

	IDSUSCRI...	CORREOELECTRONICO	TIPOSUSCRIPCION	ENTIDAD...	CUENTABANCARIA	PRECIOS...	MESESSU...
1	1	lrrbnz@hotmail.com	Normal	BBVA	ES22222222222222222222	4,99	12

Mensajes - Log

```

INSERT INTO "PROYECTOSQL"."SUSCRIPCION" (IDSUSCRIPCION, CORREOELECTRONICO, TIPOSUSCRIPCION, ENTIDADBANCARIA, CUENTABANCARIA, PRECIO
Confirmación Correcta

```

Y para el patrón de la cuenta bancaria, simplemente vamos a decirle que los dos primeros caracteres sean alfabéticos para identificar el código del país y los siguientes 22 dígitos sean únicamente números como en todas las cuentas bancarias.

Entonces, lo hacemos con la siguiente sentencia:

```
alter table SUSCRIPCION add CONSTRAINT ckcuentabancaria  
CHECK (REGEXP_LIKE(CuentaBancaria, '^[:upper:]{2}[:digit:]{22}'));
```

Y, para la comprobación, pues como siempre, pues intentamos introducir datos que no se ajusten a la cadena:

sql.sql x SUSCRIPCION x

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

	IDSUSCRI...	CORREOE...	TIPOSUS...	ENTIDAD...	CUENTABANCARIA	PRECIOSUSC...	MESESSU...
+1	1	rrbnz@g...	Normal	BBVA	ESS22222222222222222222	4,99	12

Mensajes - Log x

```
INSERT INTO "PROYECTOSQL"."SUSCRIPCION" (IDSUSCRIPCION, CORREOELECTRONICO, TIPOSUSCRIPCION, ENTIDADBANCARIA, CUENTABANCARIA,
ORA-02290: check constraint (PROYECTOSQL.CKCuentabancaria) violated
ORA-06512: at line 1

Se ha producido un error al guardar los cambios en la tabla "PROYECTOSQL"."SUSCRIPCION":
Fila 1: ORA-02290: check constraint (PROYECTOSQL.CKCuentabancaria) violated
ORA-06512: at line 1
```

sql.sql x SUSCRIPCION x

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Ordenar... | Filtrar:

	IDSUSCRI...	CORREOE...	TIPOSUS...	ENTIDAD...	CUENTABANCARIA	PRECIOSUSC...	MESESSU...
1	1	rrbnz@g...	Normal	BBVA	ESS22222222222222222222	4,99	12

Mensajes - Log x

```
INSERT INTO "PROYECTOSQL"."SUSCRIPCION" (IDSUSCRIPCION, CORREOELECTRONICO, TIPOSUSCRIPCION, ENTIDADBANCARIA, CUENTABANCARIA,
Confirmación Correcta
```