

SQL+ Cheatsheet

Francisco Pinto-Santos

March 15, 2022

Contents

1	Básicos	1
2	Creación de esquemas	2
3	Consultas sencillas	2
4	Unión de tablas	3
5	Agrupaciones	3
6	Subconsultas	3
7	Vistas, Índices, Sinónimos, Variables	4

1 Básicos

Comandos

- **LIST [número de línea]:** Muestra el mandato SQL. Si se especifica un número de línea, posiciona el cursor en dicha línea.
- **CHANGE /<palabra mal>/<palabra bien>/:** Tras posicionarse con **LIST** en una línea concreta, permite cambiar <palabra mal> por <palabra bien>.
- **RUN:** ejecuta el último mandato.
- **INPUT:** añade una línea debajo de la actual.
- **DEL:** Elimina una línea.
- **APPEND <texto>:** Añade <texto> al final de la línea.
- **CLEAR BUFFER:** Vacía el buffer de consultas.
- **CLEAN SCREEN:** Limpia la pantalla.
- **SAVE <ruta fichero>:** Guarda el contenido del buffer en <ruta fichero>.
- **GET <ruta fichero>:** Carga el contenido del fichero de <ruta fichero> en el buffer.
- **START <ruta fichero>:** Carga y ejecuta el contenido del fichero de <ruta fichero> en el buffer.
- **COMMIT:** Hace efectivos en el servidor los cambios realizados en la base de datos.
- **EXIT:** Termina la sesión.

Tipos de datos básicos

- **CHAR(N):** Cadena de texto de hasta N caracteres. Se pone entre comillas simples (').
- **LONG:** Cadena de texto de hasta 64K caracteres.
- **NUMBER:** Número entero de hasta 40 dígitos.
- **NUMBER(N):** Número entero de hasta N dígitos.
- **NUMBER(N, D):** Número entero de hasta N - D dígitos enteros y D decimales.
- **DATE:** Fecha. Se pone entre comillas simples ('). Para obtener la fecha actual se escribe sysdate.

Ejemplos

```
1 CHAR(12) -- cadena de hasta 12 caracteres
2 NUMBER(10) -- número de hasta 10 dígitos
3 NUMBER(10, 4) -- número de hasta 6 dígitos enteros y 4 decimales.
4 DATE -- fecha
```

Tipos de datos numéricos

- **ABS(columna):** valor absoluto.
- **GREATEST(columna1, columna2):** valor más grande.
- **LEAST(columna1, columna2):** valor más pequeño.
- **ROUND(columna, N):** Redondea a N dígitos.
- **TRUNC(columna, N):** Trunca a N dígitos.
- **TO_NUMBER(columnaChar):** Convierte en un número.

Tipos de datos fecha

Las operaciones entre dos fechas, se expresan en número de días.

- **GREATEST(columna1, columna2):** Valor más grande.
- **LEAST(columna1, columna2):** Valor más pequeño.
- **ADD_MONTHS(columna1, N):** Suma N meses.
- **MONTHS_BETWEEN(columna1, columna2):** Calcula el número de meses entre dos fechas.
- **TO_DATE(columnaChar, formato):** Convierte en un tipo fecha.

Valores nulos

La función NVL permite establecer un valor por defecto en caso de encontrar un valor NULL.

Ejemplos

```
1 NVL(<columna>, <valor por defecto>)
2
3 -- ejemplo: recuperar la media de edad de los
  usuarios, asumiendo que aquellos con valor NULL
  en la edad, toman una edad de 18 años.
4
5 SELECT AVG(NVL(edad, 18))
6 FROM Usuario;
```

Tipos de datos textuales

- **UPPER(columna):** convierte en mayúscula.
- **LOWER(columna):** convierte en minúscula.
- **INICAP(columna):** Capitaliza la columna (convierte en mayúscula la primera).
- **GREATEST(columna1, columna2):** valor más grande.
- **LEAST(columna1, columna2):** valor más pequeño.
- **SOUNDEX(columna):** Retorna la pronunciación.

Ejemplos

```
1 -- ejemplo: convertir el email a mayúscula,
  capitalizar el nickname y concatenarlos unidos
  por el carácter +. Solo se mostrarán los
  usuarios que empiecen por g.
2
3 SELECT UPPER(email) || '+' || INITCAP(nickname)
4 FROM Usuario
5 WHERE UPPER(nickname) LIKE 'G%';
```

2 Creación de esquemas

Creación de tablas

```
1 CREATE TABLE <nombreTabla>(  
2     column1 Tipo1 [restricciones],  
3     [columna2 Tipo2,]  
4     [restricciones]  
5 )  
6  
7 -- ejemplo: crear tabla usuarios  
8  
9 CREATE TABLE Usuario(  
10     id NUMBER,  
11     email CHAR(20),  
12     fechaAlta DATE,  
13     edad NUMBER  
14 );
```

Alteración de tablas

```
1 -- añadir columnas  
2  
3 ALTER TABLE <nombre tabla> ADD <nombre columna> <tipo>  
4     [<restricciones>];  
5  
6 -- ejemplo: añade la columna nickname  
7  
8 ALTER TABLE Usuario ADD nickname CHAR(10) UNIQUE NOT NULL;  
9  
10 -- modificar columnas  
11  
12 ALTER TABLE <nombre tabla> MODIFY <nombre columna> <tipo>  
13     [<restricciones>];  
14  
15 -- ejemplo: amplía el número de caracteres posibles en la  
16     columna email  
17  
18 ALTER TABLE Usuario MODIFY email CHAR(50) UNIQUE;
```

Eliminación de tablas

```
1 DROP TABLE <nombre tabla>;  
2  
3 -- ejemplo: eliminar ReferenciaUsuario  
4  
5 DROP TABLE ReferenciaUsuario;
```

Restricciones

- **NOT NULL:** prohíbe que el valor sea NULL.
- **UNIQUE:** Establece que el valor sea único. Si hay varias columnas implicadas en el UNIQUE (en el caso de que este se establezca en las restricciones al final de la tabla) la combinación de estas es la que tiene que ser única.
- **DEFAULT:** Establece un valor por defecto si no se establece valor para la columna.
- **CHECK:** Realiza una comprobación en la inserción y actualización sobre una o varias columnas en la tabla.
- **PRIMARY KEY:** establece un valor como clave primaria.
- **REFERENCES:** establece una clave foránea a otra tabla.

Ejemplo

```
1 CREATE TABLE Usuario(  
2     id NUMBER PRIMARY KEY, -- es la clave  
3     email CHAR(20) UNIQUE, -- cada email es  
4     fechaAlta DATE DEFAULT sysdate, -- pone por  
5     edad NUMBER NOT NULL -- no puede ser null  
6 );  
7  
8 CREATE TABLE ReferenciaUsuario(  
9     claveForaneaEjemplo NUMBER REFERENCES  
10     Ejemplo(id)  
11 );  
12  
13 -- Se pueden crear también así  
14  
15 CREATE TABLE Usuario(  
16     id NUMBER,  
17     email CHAR(20),  
18     fechaAlta DATE DEFAULT sysdate, -- pone por  
19     edad NUMBER NOT NULL, -- no puede ser null  
20     PRIMARY KEY(id), -- es la clave primaria  
21     UNIQUE(email) -- cada email es único  
22 );
```

3 Consultas sencillas

Consulta de datos básica

```
1 SELECT <columna1>, ..., <columnaN>  
2 FROM <tabla>  
3 [WHERE condiciones]  
4 [ORDER BY <columna1> [ASC/DESC], <columna2> [ASC/DESC]];  
5  
6 -- ejemplo: recuperar todos los datos de los usuarios cuya  
7     segunda letra es una a  
8  
9 SELECT *  
10 FROM Usuario  
11 WHERE nickname LIKE '_a%';  
12  
13 -- ejemplo: recuperar el email y nickname de todos los  
14     usuarios  
15  
16 SELECT email, nickname  
17 FROM Usuario;  
18  
19 -- ejemplo: recuperar todas las edades distintas ordenadas  
20     de forma descendente  
21  
22 SELECT DISTINCT edad  
23 FROM Usuario  
24 ORDER BY edad DESC;
```

Inserción de datos

```
1 INSERT INTO <nombre tabla>[(<columna1>, ..., <columnaN>)]  
2     VALUES(<valor1>, <valor2>, ..., <valorN>);  
3  
4 -- ejemplo: insertar un valor en Usuario  
5  
6 INSERT INTO Usuario(id, email, edad, nickname)  
7     VALUES(1, 'fpintosa@upsa.es', 23, 'gandalfran');  
8  
9 -- Nota: no se ha puesto la columna date, pero al tener la  
10     constraint 'DEFAULT sysdate' mete la fecha actual
```

Borrado de datos

```
1 DELETE FROM <nombre tabla>  
2     [WHERE condicion];  
3  
4 -- ejemplo: borrar al usuario con ID 1  
5  
6 DELETE FROM Usuario  
7     WHERE id=1;  
8  
9 -- ejemplo: borrar todo el contenido de la tabla Usuario  
10  
11 DELETE FROM Usuario;
```

Actualización de datos

```
1 UPDATE <nombre tabla>
2   SET <columna1>=<valor1>[, ..., <columnaN>=<valorN>]
3   [WHERE condicion];
4
5 -- ejemplo: actualizar el email de gandalfan
6
7 UPDATE Usuario
8   SET email='fpintosa+1@upsa.es'
9   WHERE nickname='gandalfan';
10
11 -- ejemplo: establecer a NULL la edad de todos los usuarios
12
13 UPDATE Usuario
14   SET edad=NULL;
```

4 Unión de tablas

Unión de tablas

```
1 SELECT <columna1>, ..., <columnaN>
2 FROM <tabla1> [<sinonimo 1>], <tabla2> [<sinonimo 2>]
3 WHERE <tabla1>.<columnaJoin> = <tabla2>.<columnaJoin> [AND
4   condiciones];
5
6 -- ejemplo: unir Usuario y RefUsuario
7
8 SELECT email
9 FROM Usuario U, ReferenciaUsuario R
10 WHERE U.id = R.claveForaneaEjemplo;
```

5 Agrupaciones

Agrupaciones

```
1 SELECT <columna1>, ..., <columnaN>
2 FROM <tabla>
3 [WHERE condiciones tuplas individuales]
4 [GROUP BY <columna1>, ..., <columnaN>]
5 [HAVING condiciones sobre grupos]
6 [ORDER BY <columna1> [ASC/DESC], <columna2> [ASC/DESC]];
7
8 -- ejemplo: recuperar el número de usuarios agrupados por
9 -- edad. Solo se mostrarán aquellos cuyo correo empiece por f
10 -- y cuya edad supere a la edad media de todos los usuarios.
11
12 SELECT edad, COUNT(*)
13 FROM Usuario
14 WHERE email LIKE 'f%'
15 GROUP BY edad
16 HAVING edad > AVG(edad);
17
18 -- nota: "cuyo correo empiece por f" es una condición que
19 -- se aplica sobre cada tupla individualmente, por tanto, se
20 -- anota en el WHERE. Y "cuya edad supere a la edad media de
21 -- todos los usuarios" es una condición de grupo y, por
22 -- tanto, se anota en el HAVING.
```

Operaciones sobre grupos de datos

- **AVG(columna):** retorna la media del grupo de valores.
- **COUNT(columna):** retorna el número de elementos distintos del grupo de valores.
- **MAX(columna):** retorna el valor más grande del grupo de valores.
- **MIN(columna):** retorna el valor más pequeño del grupo de valores.
- **SUM(columna):** retorna la suma del grupo de valores.

Ejemplos

```
1 -- ejemplo: seleccionar la edad media, máxima y
2 -- mínima de los usuarios. También selecciona la
3 -- suma de edad de todos.
4
5 SELECT MIN(edad) edad_minima, MAX(edad)
6        edad_maxima, AVG(edad) edad_media, SUM(edad)
7        suma_edad
8 FROM Usuario;
9
10 -- ejemplo: cuenta el número de usuarios con
11 -- cada una de las edades
12
13 SELECT edad, COUNT(id) numero_usuarios_por_edad
14 FROM Usuario
15 GROUP BY edad;
```

6 Subconsultas

Subconsultas

- Consisten en consultas dentro de una query principal.
- Se ejecutan y luego se ejecuta la query principal.
- Pueden tener varios niveles de anidamiento.

Ejemplos

```
1 -- ejemplo: recuperar el nickname de los
2 -- usuarios con más edad que la media
3
4 SELECT nickname
5 FROM Usuario
6 WHERE edad >= ( SELECT AVG(edad) FROM Usuario);
7
8 -- ejemplo: actualizar la edad TODOS los
9 -- usuarios cuyo nickname empieza por G. La edad
10 -- será la media de todos los usuarios.
11
12 UPDATE Usuario
13 SET edad = (SELECT AVG(edad) FROM Usuario)
14 WHERE UPPER(nickname) LIKE 'G%';
15
16 -- ejemplo: eliminar a los usuarios que tengan
17 -- más edad que ALGUNO de los usuarios cuyo
18 -- nickname empieza por G.
19
20 DELETE FROM Usuario
21 WHERE edad > ANY (SELECT edad FROM Usuario WHERE
22                   UPPER(nickname) LIKE 'G%');
```

Operadores ANY y ALL

Son dos operadores para tratar con subselect que devuelven varios datos en vez de uno. En el caso del ejemplo anterior, esta devolvía un único dato, pero para tratar con grupos de datos hay que establecer si se desea que la condición impuesta se cumpla sobre alguno o sobre todos.

ALL

Si se aplica este operador, la condición impuesta **tiene que cumplirse sobre todos** los datos que devuelva la subselect.

ANY

Si se aplica este operador, la condición impuesta **basta con que se cumpla sobre un** dato que devuelva la subselect.

Ejemplos

```
1 -- ejemplo: recuperar los usuarios que tengan
  más edad que TODOS los usuarios cuyo nickname
  empieza por G.
2 SELECT nickname
3 FROM Usuario
4 WHERE edad > ALL (SELECT edad FROM Usuario WHERE
  UPPER(nickname) LIKE 'G\%');
5
6 -- ejemplo: recuperar los usuarios que tengan
  mas edad que ALGUNO de los usuarios cuyo
  nickname empieza por G.
7 SELECT nickname
8 FROM Usuario
9 WHERE edad > ANY (SELECT edad FROM Usuario WHERE
  UPPER(nickname) LIKE 'G\%');
```

7 Vistas, Índices, Sinónimos, Variables

Variables

Al ejecutarse la query, se solicita el valor de estas. Se definen igual que una constante, estableciendo el carácter delante.

Ejemplos

```
1 -- ejemplo: establecer una variable para
  nickname y otra para fechaAlta
2
3 SELECT *
4 FROM Usuario
5 WHERE nickname='&varNickname' AND
  fechaAlta='&varFechaAlta';
```

Vistas

Consisten en tablas virtuales

Ejemplos

```
1 CREATE VIEW <nombre vista> AS
2   <Consulta select>;
3
4 DROP VIEW <nombre vista>;
5
6 -- También se puede crear una tabla con la
  información de otras tablas, esta NO es virtual
7
8 CREATE TABLE <nombre vista> AS
9   <Consulta select>;
10
11 -- ejemplo: crear una vista con los usuarios
  cuyo nickname empieza por G.
12 CREATE VIEW GStartUsers AS
13   SELECT nickname
14   FROM Usuario
15   WHERE UPPER(nickname) LIKE 'G\%';
16
17 -- ejemplo: eliminar la vista
18
19 DROP VIEW GStartUsers;
```

Índices

Permiten reducir el tiempo de ejecución de las consultas. Se deben crear si se cumplen estas condiciones:

- Si la tabla tiene más de 200 tuplas.
- Contar con las columnas más utilizadas en la cláusula WHERE.
- Columnas de las tablas que se emplean en los JOIN.
- No crear más de 2/3 índices sobre una tabla o se ralentizará el rendimiento.

Ejemplos

```
1 CREATE [UNIQUE] INDEX <nombre indice> ON <nombre
  tabla>(
2   <columna1 [ASC/DESC]>
3   [, <columna2 [ASC/DESC]>]
4 ) [CLUSTER];
5
6 DROP INDEX <nombre indice>;
7
8 -- ejemplo: crear un indice descendiente sobre
  nickname
9
10 CREATE INDEX UsersNicknameIndex On Usuario(
11   nickname DESC
12 );
13
14 -- ejemplo: eliminar el índice
15
16 DROP INDEX UsersNicknameIndex;
```

Concesión de permisos

```
1 GRANT <ALTER/SELECT/INSERT/DELETE/UPDATE/INDEX> -- en vez
  de todas estas operaciones, se puede poner ALL para
  especificar todas a la vez
2 ON <tabla>
3 TO <usuario> -- si en vez de usuario se pone PUBLIC se
  concede a todos los usuarios
4 [WITH GRANT OPTION]; -- si se establece esta opción, el
  usuario puede conceder los permisos que se le han dado a
  otros usuarios
5
6 -- ejemplo: conceder permisos de alteración de tabla y
  selección de datos en Usuario a todos los usuarios
7
8 GRANT ALTER, SELECT
9 ON Usuario
10 TO PUBLIC;
11
12 -- ejemplo: conceder permisos de inserción a Fran con la
  posibilidad de que Fran conceda permisos posteriormente
13
14 GRANT INSERT
15 ON Usuario
16 TO Fran
17 WITH GRANT OPTION;
```

Revocación de permisos

```
1 REVOKE <ALTER/SELECT/INSERT/DELETE/UPDATE/INDEX> -- en vez
  de todas estas operaciones, se puede poner ALL para
  especificar todas a la vez
2 ON <tabla>
3 TO <usuario>; -- si en vez de usuario se pone PUBLIC se
  concede a todos los usuarios
4
5 -- ejemplo: quitar permisos de inserción a Fran en
  Usuario. Nota: Al quitárselos a Fran también se les
  quitarán a los usuarios a los que se los concedió Fran.
6
7 REVOKE INSERT
8 ON Usuario
9 TO Fran;
```

Sinónimos

```
1 CREATE SYNONYM <nombre sinónimo> FOR <tabla>;
2
3 DROP SYNONYM <nombre sinónimo>;
4
5 -- ejemplo: crear sinónimo para la tabla Usuarios en
  inglés "Users"
6
7 CREATE SYNONYM Users FOR Usuario;
8
9 -- ejemplo: eliminar sinónimo
10
11 DROP SYNONYM Users;
```