SQL Syntax Cheatsheet

Thomas Boxall
up2108121@myport.ac.uk

April 2023

1 PostegreSQL 'Slash' Commands

`\l` list all databases in the server
`\c databaseName` connect to database
`\d` list everything in the database
`\dt` list just the tables

2 `CREATE`

a database: `CREATE DATABASE databaseName;`
a table:

```
CREATE TABLE tableName(
    attribute dataType constraints,
    ...
);
```

a database owned by another user: `CREATE DATABASE userName OWNER userName;`

3 `DROP` & `DELETE`

a whole database: `DROP DATABASE databaseName;`
a single table: `DROP TABLE tableName;`
a single record: `DELETE FROM tableName WHERE attribute = value;`

4 `ALTER`

a table to add new column: `ALTER TABLE tableName ADD COLUMN colName dataType;`
add a unique constraint to an attribute:
`ALTER TABLE tableName ADD CONSTRAINT constraintName UNIQUE (attribute);`
add a length constraint to an attribute:
`ALTER TABLE tableName ADD CONSTRAINT constraintName CHECK(length(postcode)>5);`

5 `INSERT`

data to a table: `INSERT INTO tableName (attr1, attr2) VALUES(val1, val2);`

6 `UPDATE`

a record in a table: `UPDATE tableName SET attributeName = val WHERE idAttribute = val;`

7 `SELECT`

everything from a table: `SELECT * FROM tableName;`
only certain records from a table: `SELECT * FROM tableName WHERE attributeName = desiredValue;`
only certain attributes from a table: `SELECT attr1, attr2 FROM tableName;`
only distinct attributes from a table: `SELECT DISTINCT attributeName FROM tableName;`
only 5 records from a table: `SELECT * FROM tableName LIMIT 5;`
an attribute and give it an alias: `SELECT attributeName as "Alias" FROM tableName;`

8 Functions

number of values: `COUNT(toCount)`

8.1 Text Functions

convert character to ASCII: `ASCII('char')`
convert ASCII to character: `CHR(intVal)`
convert the first letter after a space to a capital: `INTCAP('string to convert')`
get the position of a substring in a string: `POSITION('substring' IN 'string to search')` (NB: string index starts at 1)
concatenate strings with a space between: `CONCAT(stringOne, ' ', stringThree)`
concatenate strings with the same delimiter between each: `CONCAT_WS(' ', stringOne, stringTwo)`
convert binary to text: `CONVERT_FROM(toConvert, 'utf-8)`
convert all characters to uppercase: `UPPER(toConvert)`

8.2 Date Functions

current date and time: `NOW()`
return the current time: `CURRENT_TIME`
extract part of a date: `DATE_PART('target_part', dateToExtractFrom)`
truncate date to specific level & fill the rest with `0`: `DATE_TRUNC('truncation-level', dateToTruncate)`
get the difference between now and a date as a formatted string: `AGE(dateToGetDifferenceOf)`
get the difference between two dates as a formatted string: `AGE(laterDate, earlierDate)`

8.3 Numerical Functions

maximum value: `MAX()`
minimum value: `MIN()`
average value: `AVG()`
generate a random number: `RANDOM()`

9 `JOIN`

two tables together using an inner join

```
SELECT tableName.attributeName, ... FROM tableName
JOIN tableToJoinName ON tableToJoinName.attribute = currentTableName.attribute;
```

two tables together not using inner join but achieve the same results:

```
SELECT attributeName, attributeName FROM tableOneName, tableTwoName
WHERE tableOneName.attributeName = tableTwoName.attributeName;
```

two tables together using a left join:

```
SELECT currentTableName.attributeName, ... FROM currentTableName
LEFT JOIN tableTwo ON tableToJoinName.attribute = currentTableName.attribute;
```

two tables together using an outer join:

```
SELECT currentTableName.attributeName, ... FROM currentTableName
RIGHT JOIN tableTwo ON tableToJoinName.attribute = currentTableName.attribute;
```

two tables together using a full outer join:

```
SELECT currentTableName.attributeName, ... FROM currentTableName
FULL OUTER JOIN tableTwo ON tableToJoinName.attribute = currentTableName.attribute;
```

10 `GROUP BY`

an attribute in the table:

```
SELECT attributeOne, attributeTwo FROM tableName
GROUP BY attributeOne;
```

an attribute in the table and only show certain records:

```
SELECT attributeOne, attributeTwo FROM tableName
GROUP BY attributeOne
HAVING attributeName = value;
```

11 `ORDER BY`

an attribute in ascending order:

```
SELECT attributeName FROM tableName ORDER BY attributeName;
```

an attribute in ascending order:

```
SELECT attributeName FROM tableName ORDER BY attributeName ASC;
```

an attribute in descending order:

```
SELECT attributeName FROM tableName ORDER BY attributeName DESC;
```

12 Wildcards

any number of characters: `%`

13 Views

create a view: `CREATE viewName AS queryString;`
execute view: `SELECT * FROM viewName;`

14 Roles

create a role: `CREATE ROLE username WITH LOGIN PASSWORD 'password';`
give role superuser permission: `ALTER ROLE username WITH SUPERUSER;`
give role read access to table: `GRANT select ON tableName TO username;`
give two roles write access to a table: `GRANS update ON tableName TO username, username;`
revoke role read access to a table: `REVOKE select ON tableName FROM username`
delete a role:

```
-- need to revoke all permissions first
DROP DATABASE username;
DROP ROLE username;
```

15 Encryption

Turn on encryption (off by default): `CREATE EXTENSION IF NOT EXISTS pgcrypto;`
Encrypt values (probably used as part of `INSERT`): `ENCRYPT('toEncrypt', 'key', 'aes')`
Decrypt values (provably used as part of `SELECT`): `DECRYPT(toDecrypt, 'key', 'aes')`

16 Dates

display data when between two dates:
`SELECT * FROM tableName WHERE dateAttr BETWEEN 'firstDate' AND 'lastDate';`
change date style: `SET DATESTYLE TO newDateStyle;`

17 JSON

return the raw value: `SELECT data -> 'key' FROM jsonTable;`
return the text value: `SELECT data ->> 'key' FROM jsonTable;`