SQL Data Science Quick Reference — THE DATA Hitchhiker

SQL Statement	Syntax
ALTER TABLE	ALTER TABLE table_name ADD|DROP column_name datatype
AS (alias)	SELECT column_name AS column_alias FROM table_name AS t1
BETWEEN	… WHERE column_name BETWEEN value1 AND value2
CASE	SELECT Case WHEN c1 = c2 then 1 Else 2 END …
CTE (Microsoft)	WITH cteName AS (select column from table)
CREATE VIEW	CREATE VIEW view_name AS SELECT column_name(s) …
EXISTS	IF EXISTS (SELECT * FROM table_name WHERE id = ?) BEGIN some-action END
GROUP BY	SELECT c1, aggregate_function(c2) FROM table_name GROUP BY column_name1
HAVING	GROUP BY c1 HAVING aggregate_function(c1) operator value
IN	… WHERE column_name IN (value1,value2,..)
LIKE	… WHERE column_name LIKE '%pattern%'
ORDER BY	… FROM t1 ORDER BY column_name [ASC|DESC]
SELECT	**SELECT c1, c2 FROM t1**
SELECT DISTINCT	SELECT DISTINCT column_name(s) FROM table_name
SELECT INTO (Copy Table)	SELECT c1, c2 INTO tnew [IN externaldatabase] FROM t1
SELECT TOP	SELECT TOP number|percent column_name(s) FROM t1
UNION (remove dup rows)	SELECT c1 from t1 UNION SELECT c2 from t2
UNION ALL (include dup rows)	SELECT c1 from t1 UNION ALL SELECT c2 from t2
WHERE AND / OR	… WHERE condition AND|OR condition

Join	Example	
INNER JOIN	SELECT c1, c2 FROM t1 INNER JOIN t2 ON t1.ID = t2.ID	
LEFT JOIN	SELECT c1, c2 FROM t1 LEFT JOIN t2 ON t1.ID = t2.ID	
RIGHT JOIN	SELECT c1, c2 FROM t1 RIGHT JOIN t2 ON t1.ID = t2.ID	
FULL OUTER JOIN	SELECT c1, c2 FROM t1 FULL OUTER JOIN t2 ON t1.ID = t2.ID	

Agg	Outcome
AVG	Mean
SUM	Sum
COUNT	Count Values
MIN	Lowest Value
MAX	Highest Value

Operator	Outcome
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulus
=	Equal
>	Great than
<	Less than
<=	Less than or equal
<>	Not Specific Value
>=	Greater than or equal

Logical	Outcome
IN	Match list
NOT	Negative
IS NULL	Null Value
IS NOT NULL	No Null Value
BETWEEN	Within Range

Function	Outcome
LEN	String Length
CONCAT	Combine String
TRIM	Remove Whitespace
CONVERT	Change datatype
UPPER LOWER	Convert String Case