TABLE OF CONTENTS

DATA AND VALUES

DATA TYPES	
TEXT: Characters surrounded by quotes	`"String1"`
INTEGER: Whole number, positive or negative	`10`
REAL: Floating point (decimal) number	`-20.50`
BOOLEAN: True or False values	`TRUE / FALSE`
NULL: Value missing/not entered	`NULL`
DATABASE INFORMATION	
View all tables in a database	`.tables`
View all columns (and their datatype) in a table	`PRAGMA table_info(TableName);`

ACCESSING DATA

SELECT COMMAND	
Select all rows & columns from a table	`SELECT * FROM Table;`
Select all rows & a specific column from a table	`SELECT Column FROM Table;`
Select all rows, Col1 & Col2 from a table	`SELECT Col1, Col2 FROM Table;`
Using an alias	`SELECT Col AS Alias FROM Table;`
Select distinct	`SELECT DISTINCT Col FROM Table;`

WHERE Clause	

Select all rows matching the condition	`SELECT * FROM Table`
	`WHERE (condition);`

Conditional Operators:
> Greater than
< Less than
= Equal to
!= Not equal to
	`Example:`
	`WHERE Rating > 10`
	`WHERE Price < 20.00`
	`WHERE Name = "cstutor-sql"`
	`WHERE Location != "Toronto"`

Like: Matching the characters	`WHERE Column LIKE "chars"`

% Wildcard for 0 or more of any character
Example: Returns ‘San Francisco’, ‘Santa Ana'
	`%`
	`WHERE City like 'San%'`

_ Wildcard for a single character
Example: Returns 3-char string ending with ‘s’
	`_`
	`WHERE Name like '_ _s'`

Multiple conditions:
AND: Both conditions must be met
	`SELECT * FROM Table`
	`WHERE (condition1)`
	`AND (condition2);`

Multiple conditions:
Or: Either condition must be met
	`SELECT * FROM Table`
	`WHERE (condition1)`
	`OR (condition2);`

IN: Returns rows where the value in Column matches a value in the list of values	`SELECT * FROM Table`
	`WHERE Column`
	`IN (list_of_values);`

NOT IN: Returns rows where the value in Column does not match a value in the list of values	`SELECT * FROM Table`
	`WHERE Column`
	`NOT IN (list_of_values);`

Example: Return rows where the State column matches the values in the list	`SELECT * FROM Table`
	`WHERE State`
	`IN ('CA', 'NV', 'MN');`

BETWEEN: Values between x and y, upper and lower bound inclusive	`SELECT * FROM Table`
	`WHERE Column BETWEEN x AND y;`

IS NULL: Returns rows where the value in the column is NULL	`SELECT * FROM Table`
	`WHERE Column IS NULL;`

IS NOT NULL: Returns rows where the value in the column is NOT NULL	`SELECT * FROM Table`
	`WHERE Column IS NOT NULL;`

AGGREGATE FUNCTIONS

Syntax	Description	Example

SUM(col)	Adds up all the numeric values in col	SELECT SUM(TotalCost)
		FROM Orders;

AVG(col)	Returns the average value in col	SELECT AVG(HomePrice);
MAX(col)	Returns the maximum value in col	MAX(Name);
MIN(col)	Returns the minimum value in col	MIN(OrderCount);
COUNT(col)	Counts the number of rows in col	COUNT(*);

ORDER BY & LIMIT

Order by Syntax
Default: ASC
	SELECT *Column(s)*
	FROM *Table*
	ORDER BY *Column* [ASC | DESC];

Example: Order results by Price from low to high	SELECT * FROM Dish
	ORDER BY Price;

Order first by Col1, then Col2	SELECT *Column(s)*
	FROM *Table*
	ORDER BY Col1, Col2;

Limit syntax	SELECT *Column(s)*
	FROM *Table*
	LIMIT *#ofrows*;

Example: Limit to first 5 rows:	SELECT * FROM Dish
	LIMIT 5;

Example: Get most expensive dish	SELECT * FROM Dish
	ORDER BY Price DESC
	LIMIT 1;

INSERTING & DELETING

Insert data into Table	INSERT INTO *Table*(*Column1*, *Column2*, *…*)
	VALUES (*value1*, *value2*, *…*);

Insert values into all columns	INSERT INTO *Table*
	VALUES (*value1*, *value2*, *…*);

Example:	INSERT INTO Dish
	VALUES (6, "Falafel", "Dinner", TRUE, 14.99);

Delete all rows in table (Doesn’t delete table itself)	DELETE FROM *Table*;

Delete selected row(s) where condition is met	DELETE FROM *Table*
	WHERE *condition*;

Example: Deletes 1 dish with matching Id.	DELETE FROM Dish
	WHERE Id = 6;

Example: Deletes all dishes with name containing the word salad	DELETE FROM Dish
	WHERE Name LIKE '%salad%'

UPDATING

Syntax	UPDATE *Table*
	SET *col1* = *val1*, *col2* = *val2*, *…*
	[WHERE *condition*];

Update Column in every row in Table to *value*	UPDATE *Table*
	SET *Column* = *value*;

Example: Set all rows to NULL in the Vegetarian column	UPDATE Dish
	SET Vegetarian = NULL;

Update specific row in Table where condition is met	UPDATE *Table*
	SET *Column* = *value*
	WHERE *condition*;

Example: Updating two columns at once for a row with Id of 15	UPDATE Dish
	SET Name = 'Falafel', Vegetarian = TRUE
	WHERE Id = 15;