

SQL

CHEATSHEET

By @AbzAaron

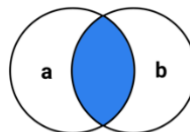
<https://twitter.com/AbzAaron>



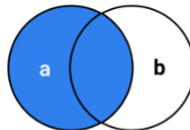
Common Commands

SELECT	Select data from database
AS	Rename column or table with alias
FROM	Specify table we're pulling from
WHERE	Filter query to match a condition
JOIN	Combine rows from 2 or more tables
AND	Combine conditions in a query. All must be met
OR	Combine conditions in a query. One must be met
LIKE	Search for patterns in a column
IN	Specify multiple values when using WHERE
IS NULL	Return only rows with a NULL value
LIMIT	Limit the number of rows returned
CASE	Return value on a specified condition
CREATE	Create TABLE, DATABASE, INDEX or VIEW
DROP	Delete TABLE, DATABASE, or INDEX
UPDATE	Update table data
DELETE	Delete rows from a table
ALTER TABLE	Add/Remove columns from table
GROUP BY	Group rows that have same values into summary rows
ORDER BY	Set order of result. Use DESC to reverse order
HAVING	Same as WHERE but used for aggregate functions
SUM	Return sum of column
AVG	Return average of column
MIN	Return min value of column
MAX	Return max value of column
COUNT	Count number of rows

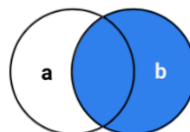
Joins



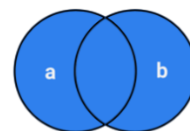
a INNER JOIN b



a LEFT JOIN b



a RIGHT JOIN b



a FULL OUTER JOIN b

Examples

Select all rows from table with filter applied

```
SELECT * FROM tbl WHERE col1 > 5;
```

Select first 10 rows for 2 columns

```
SELECT col1, col2 FROM tbl LIMIT 10;
```

Select all rows with multiple filters applied

```
SELECT * FROM tbl WHERE col1 > 5 AND col2 < 2;
```

Select all rows from col1 and col2 ordering by col1

```
SELECT col1, col2 FROM tbl ORDER BY 1;
```

Return count of rows in table

```
SELECT COUNT(*) FROM tbl;
```

Return sum of col1

```
SELECT SUM(col1) FROM tbl;
```

Return max value from col1

```
SELECT MAX(col1) FROM tbl;
```

Computer summary statistics by grouping col2

```
SELECT AVG(col1) FROM tbl GROUP BY col2;
```

Combine data from two tables using a left join

```
SELECT * FROM tbl1 AS t1  
LEFT JOIN tbl2 AS t2 ON t2.col1 = t1.col1;
```

Aggregate and filter results

```
SELECT  
col1,  
AVG(col2) * AVG(col3) AS total  
FROM tbl  
GROUP BY col1  
HAVING total > 2
```

Implementation of CASE statement

```
SELECT col1,  
CASE  
WHEN col1 > 10 THEN "more than 10"  
WHEN col1 < 10 THEN "less than 10"  
ELSE "10"  
END AS NewColumnName  
FROM tbl;
```

Create

```
CREATE DATABASE MyDatabase;
```

```
CREATE INDEX IndexName  
ON TableName(col1);
```

```
CREATE TABLE OurTable (  
id int,  
name varchar(12)  
);
```

Delete

```
DROP DATABASE OurDatabase;
```

```
DROP TABLE OurTable;
```

Update Table

```
UPDATE OurTable  
SET col1 = 56  
WHERE col2 = 'something';
```

Delete Records

```
DELETE FROM OurTable  
WHERE col1 = 'something';
```

Add/Remove Column

```
ALTER TABLE OurTable  
ADD col5 int;
```

```
ALTER TABLE OurTable  
DROP COLUMN col5;
```

Order of Execution

FROM
WHERE
GROUP BY
HAVING
SELECT
ORDER BY
LIMIT

Sources:

<https://www.dataquest.io/blog/sql-commands/#t-1613505022898>

<https://www.dataquest.io/blog/sql-joins-tutorial/>