



SQL

🕒 Created	@December 12, 2023 10:00 AM
🏷️ Tags	SQL

▼ Notes

- ▼ SQL is not case-sensitive
- ▼ Some programs require ; at the end of the code block
- ▼ Tab your shit in a neat fashion please
- ▼ INNER JOIN is equivalent to JOIN
- ▼ Use all lowercase because it is easier to read lol

Finding Things

SELECT column

- * for everything in that column
- **DISTINCT** for unique values

- **MIN()** to select smallest value in the column
- **MAX()** to select largest value in the column
- **COUNT(*)** returns number of records with specified value
- **AVG()** to calculate average of all values in the column
- **SUM()** to add total sum of all values in the column
- Column **AS** whatever your alias

FROM table

WHERE to find a specific string or numerical value

- **NOT**
- **AND**
- **OR**
- **IS NULL** means it's empty
- **IS NOT NULL** means it's not empty
- Column **LIKE 'a%'** finds records where value of column starts with letter 'a'
- Column **LIKE '%a'** finds records where value of column ends with letter 'a'
- Column **LIKE '%a%'** finds records where value of the column contains letter 'a'
- ^ You get the pattern, for things like value starts with letter 'a' and ends with letter 'b'
- Same rules ^ apply for column **NOT LIKE**
- Column **LIKE '_a%'** finds records where second letter in column is an 'a'
- ^ each underscore will count for a wildcard letter, adjust as needed
- Column **LIKE '[acs]%'** will find records in column where first letter is an 'a' or 'c' or 's'
- Column **LIKE '[a-f]%'** will find records in column where first letter starts with anything from 'a' to 'f'
- Column **LIKE '[!acf]%'** will find records in column where first letter is NOT 'a' or 'c' or 'f'

- Column **IN ('Norway', 'France')** finds records in column in either country
- Column **NOT IN** does the opposite find
- Column **BETWEEN** value **AND** value will get you results between those specific values alphabetically or numerically
- Column **NOT BETWEEN** value **AND** value will not get you results between those specific values alphabetically or numerically

GROUP BY

ORDER BY to sort results alphabetically

- **ASC** is optional to add
- **DESC** to sort backwards z-a
- **COUNT(CustomerID) DESC** most customers first

Creating a new record in the table:

INSERT INTO table (, , etc)

VALUES (, , etc)

Update a column in all records in a table

UPDATE table

SET column = ' ';

If you only want to update the ones with a specific value, add **WHERE**

Delete all records from a table where value is something

DELETE FROM table

WHERE

Insert missing parts in the JOIN clause to join two tables Order and Customers, using CustomerID field in both tables as the relationship between two tables

SELECT *

FROM Orders

LEFT JOIN Customers

ON Orders.CustomerID = Customers.CustomerID;

Choose the correct JOIN clause to select all records from the two tables where there is a match in both tables

SELECT *

FROM Orders

INNER JOIN Customers

ON Orders.CustomerID = Customers.CustomerID;

Choose the correct JOIN clause to select all the records from the Customers table plus all the matches

in the Orders table

```
SELECT *  
FROM Orders  
RIGHT JOIN Customers  
ON Orders.CustomerID = Customers.CustomerID;
```

Create a new database called testDB

```
Create database testDB
```

- Same applies for creating a new table

Delete a database named testDB

```
DROP DATABASE testDB
```

- Same applies for deleting table

Delete all data inside a table

```
TRUNCATE table
```

Add a column of type DATE called Birthday

```
ALTER TABLE Persons
```

```
ADD Birthday DATE;
```

Delete column Birthday from the Persons table

```
ALTER TABLE Persons
```

```
DROP COLUMN Birthday
```

