SQL Basic Cheat Sheet

DDL (Data Definition Language)

Create a new table t with three columns

```
CREATE TABLE t (
    id INT PRIMARY KEY,
    name VARCHAR NOT NULL,
    price FLOAT DEFAULT 0
);
```

Set c3 as a foreign key

```
CREATE TABLE t (
id INT PRIMARY KEY,
c3 INT,
FOREIGN KEY (c3) REFERENCES t2(id)
);
```

Delete the table t from the database

```
DROP table t;
```

Add a new column t to the table

```
ALTER table t ADD column;
```

Drop a column c from the table t

```
ALTER TABLE t DROP COLUMN c;
```

Add a constraint to the table t

```
ALTER TABLE t ADD constraint;
```

Drop a constraint

```
ALTER TABLE t DROP constraint;
```

Rename a table from t to t2

```
ALTER TABLE t RENAME to t2;
```

Rename a column from c to c2

```
ALTER TABLE t RENAME c to c2;
```

Remove all data in a table

```
TRUNCATE TABLE t;
```

```
CREATE VIEW v as
SELECT * FROM t
```

DML (Data Manipulation Language)

Insert a row into a table t

```
INSERT INTO t(column_list)
VALUES(value_list);
```

Insert multiple rows into a table t

```
INSERT INTO t(column_list)
VALUES(value_list),
      (value_list),...;
```

Insert rows from t2 into t

```
INSERT INTO t(column_list)
SELECT column_list
FROM t2;
```

Update the new value in the column name for all rows

```
UPDATE t
SET name = new_value;
```

Update values in the columns name and price that matches the condition

```
UPDATE t
SET name = new_value,
    price = new_value
WHERE condition;
```

Delete all data in a table

```
DELETE FROM t;
```

Delete a subset of rows in a table t

```
DELETE from t
WHERE condition;
```

LearnDataEngineering.com

Adapted from https://www.sqltutorial.org/sql-cheat-sheet/

SQL Basic Cheat Sheet

```
SELECT column_list,
<aggregate_function>
FROM t
GROUP BY column_list
```

Where <aggregate_function> can be:
COUNT() - the number of rows
AVG() - average value of a numeric column
SUM() - sum of a numeric column
MIN() - minimum of a specified column
MAX() - maximum of a specified column

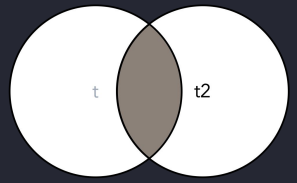
Comparison operator	Description
=	Equal
<>	Not equal
!=	Not equal
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal

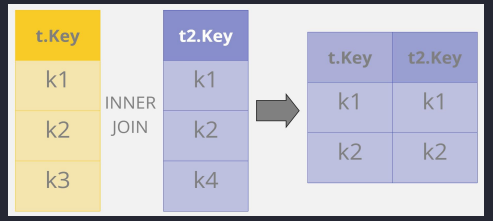
Comparison operator	Description
IN()	Matches values in a list
NOT	Negates a condition
BETWEEN	Within a range (inclusive)
IS NULL	NULL values
IS NOT NULL	Non-NULL values
LIKE	Pattern matching with %

JOIN Statements

INNER JOIN

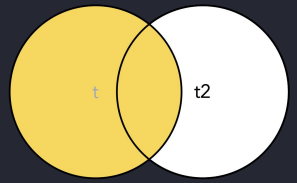
```
SELECT column_list
FROM t
[INNER] JOIN t2
ON t.Key = t2.Key
```

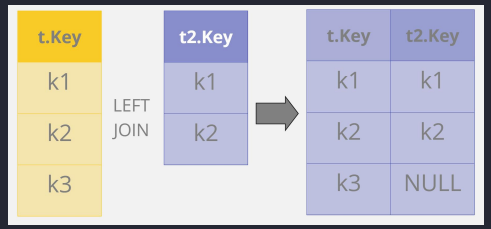




LEFT JOIN

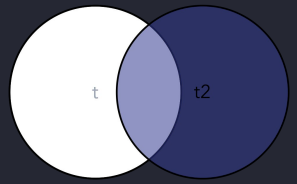
```
SELECT column_list
FROM t
LEFT [OUTER] JOIN t2
ON t.Key = t2.Key
```

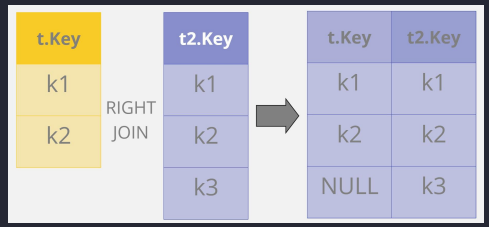




RIGHT JOIN

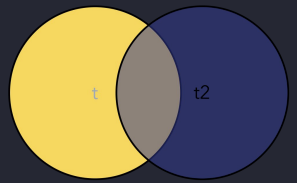
```
SELECT column_list
FROM t
RIGHT [OUTER] JOIN t2
ON t.Key = t2.Key
```

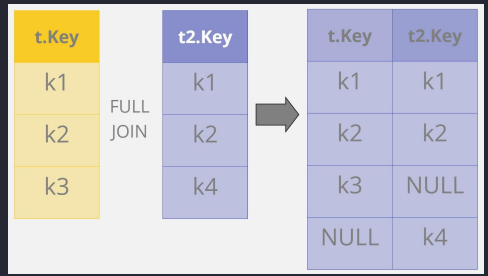




FULL JOIN

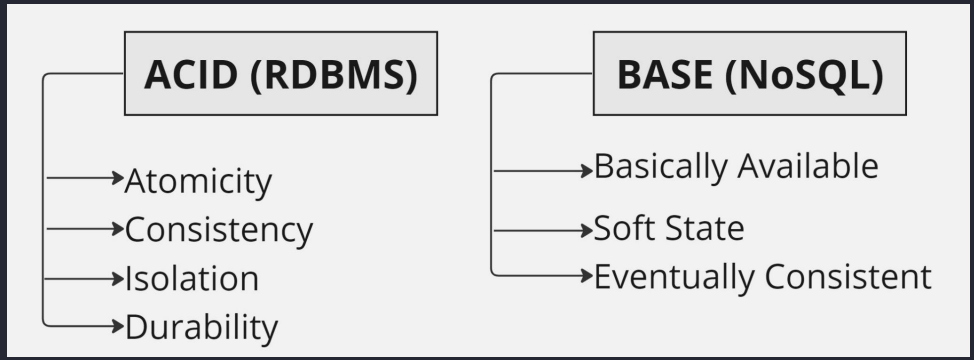
```
SELECT column_list
FROM t
FULL [OUTER] JOIN t2
ON t.Key = t2.Key
```





Adapted from
https://www.ml4devs.com/articles/sql-joins-tutorial-inner-join-left-join-right-join-full-join-cross-join/

SQL Cheat Sheet



`COMMIT` -saves all the transactions made on a database

```
DELETE FROM t
WHERE c = 'value';
COMMIT;
```

`ROLLBACK` -undoes transactions which are not yet been saved

```
DELETE FROM t
WHERE c = 'value';
ROLLBACK;
```

`ROLLBACK` -rolls transaction back to a certain point without having to roll back the entirety of the transaction

```
SAVEPOINT SAVED;
DELETE FROM t
WHERE c = 'value';
ROLLBACK TO SAVED;
```

SQL Cheat Sheet

Subquery/nested query

```
SELECT column_list
FROM t
INNER JOIN
(SELECT column_list FROM t1
WHERE <condition>) t1
ON t.Key = t1.Key
```

main query

subquery (returns data that is used in main query)

Common Table Expression/CTE

```
WITH cte as
(SELECT column_list
FROM t)
SELECT column_list
FROM t1
INNER JOIN cte ON t1.Key = cte.Key
```

CTE body/definition

CTE usage

Adapted from https://learnsql.com/blog/what-is-cte/

Window Functions Cheat Sheet

```
SELECT column_list
<window_function()> OVER (
PARTITION BY <...>
ORDER BY <...>
<window_frame>)
<window_column_alias>
FROM t;
```

Where <window_function()> can be:
AVG(expr) - average value for rows within the window frame
COUNT(expr) - count of values for rows within the window frame
MAX(expr) - maximum value within the window frame
MIN(expr) - minimum value within the window frame
SUM(expr) - sum of values within the window frame

FIRST_VALUE(expr) - the value for the first row within the window frame
LAST_VALUE(expr) - the value for the last row within the window frame

ROW_NUMBER() - unique number for each row within partition, with different numbers for tied values
RANK() - ranking within partition, with gaps and same ranking for tied values
DENSE_RANK() - ranking within partition, with no gaps and same ranking for tied values

LEAD(expr, offset, default) - the value for the row offset rows after the current; offset and default are optional; default values: offset = 1, default = NULL
LAG(expr, offset, default) - the value for the row offset rows before the current; offset and default are optional; default values: offset = 1, default = NULL