**Definitions used throughout this cheat sheet**

**Primary key:** A primary key is a field that uniquely identifies each record in a table in a relational database. Primary keys are used as keys to join tables on.

**Foreign key:** A foreign key is a field in a table that references the primary key of another table in a relational database. One way to join two tables is by constructing a foreign key from one table to the primary key of another.

**One-to-one relationship:** In a one-to-one relationship, a record in one table is uniquely related to a single record in the other table.

**Many-to-many relationship:** In a many-to-many relationship, records in a given table A can be related to one or more records in another table B, and records in table B can be related to many records in table A.

**Sample Data**

### INNER JOIN

An inner join between two tables will return only records where a joining field, such as a key, finds a match in both tables.

**INNER JOIN on one field**

<table>
<thead>
<tr>
<th>artist_id</th>
<th>name</th>
<th>album_id</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>1</td>
<td>Song1</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>2</td>
<td>Song2</td>
</tr>
<tr>
<td>3</td>
<td>John</td>
<td>3</td>
<td>Song3</td>
</tr>
</tbody>
</table>

**INNER JOIN on multiple fields**

<table>
<thead>
<tr>
<th>artist_id</th>
<th>name</th>
<th>album_id</th>
<th>title</th>
<th>artist_i</th>
<th>album_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>1</td>
<td>Song1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>2</td>
<td>Song2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>John</td>
<td>3</td>
<td>Song3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

**Self Join**

Self joins are used to compare values in one table to other values of the same table by joining different parts of the same table together.

**Full Join**

A full join combines a left join and right join. A full join will return all records from both tables. In cases of identical values, full join returns all records from the joining field in the first table. Records from the second table are also returned if they match.

**Cross Join**

A cross join returns all possible combinations of two tables. Cross join does not require a condition to join on.

**Union All**

The UNION ALL operator returns only those rows from the left table that do not present in the right table.

**Except**

The EXCEPT operator returns only those rows from the left table that are not present in the right table.

**Semi Join**

A semi join returns records in the left table where a condition is met in the second table. A semi join makes use of a WHERE clause to use the second table as a filter for the first.

**Right Join**

A right join keeps all of the original records from the right table and returns missing values for any column from the left table where there is no match.

**Left Join**

A left join keeps all of the original records from the left table and returns missing values for any column from the right table where there is no match.

**Set Theory Operators in SQL**

**UNION**

The UNION operator returns all the values in the first table and returns only values that are not present in the second table.

**INTERSECT**

The INTERSECT operator returns only those rows from both tables.

**EXCEPT**

The EXCEPT operator returns only those rows from the left table that are not present in the right table.

**Anti Join**

An anti join returns records in the left table where a condition is not met in the second table. It makes use of a WHERE clause to use explicit values from the second table.