

# What is Arel?

---

Arel is a domain-specific-language allowing to express your queries in relational algebra. In past versions of Rails it was rather common to have to resort to Arel in order to accomplish some rather frequently requested functionalities, though nowadays Rails 6's Active Record already covers most of these use cases.

When creating Arel queries one usually starts by getting the `Arel::Table` objects of the SQL tables we want to interact with:

```
people = People.arel_table
# => #<Arel::Table:0x007ffc5def7c50 @name="people" ...>

countries = Country.arel_table
# => #<Arel::Table:0x007ffc5dcb2328 @name="countries" ...>
```

The predicates `#eq`, `#not_eq`, `#lt`, `#gt`, `#lteq`, `#gteq` are equivalent to the operators `=`, `!=`, `<`, `>`, `<=`, `>=` and work like this:

```
countries_predicate = countries[:gdp].eq(10000)
# => #<Arel::Nodes::Equality:0x00007fd0141a29d0 ...>

people_lt_predicate = people[:birth_date].lt(Time.now - 10.years)
# => #<Arel::Nodes::LessThan:0x00007fd00baa0798 ...>

people_gt_predicate = people[:birth_date].gt(Time.now - 20.years)
# => #<Arel::Nodes::GreaterThan:0x007ffc5e1c47d0 ...>
```

The `#and` & `#or` can be used to create multitable predicates and work in a similar way, like this:

```
or_predicate =
countries[:gdp].gt(10000).or(people[:birth_date].gt(Time.now - 10.years))
# => #<Arel::Nodes::Grouping:0x00007fd0141d1ca8 ...>
and_predicate = countries[:gdp].gt(1000).and(countries[:gdp].lt(2000))
# => #<Arel::Nodes::And:0x00007fd0138b4648 ...>
```

We can later pass these predicates to `#where`, and as long as the required tables are joined the predicate will be successfully converted to SQL:

```
Person.joins(:country)
      .where(or_predicate)
# => SELECT "people".* FROM "people"
# INNER JOIN "countries"
# ON "countries"."id" = "people"."country_id"
```

```
# WHERE "countries"."gdp" > 10000  
# OR "people"."birth_date" > '2008-07-04'
```

Arel matches (SQL LIKE)

```
people_last_name = Person.arel_table[:last_name]  
  
Person.where(people_last_name.matches("A%"))  
# SELECT "people".* FROM "people"  
# WHERE "people"."last_name" LIKE 'A%'  
# => #<ActiveRecord::Relation [#<Person last_name: "Aufderhar" ... >, #  
<Person last_name: "Armstrong"...> ...]>
```

This returns people whose last\_name start with the character "A".