

FunSQL:  
A library for compositional construction  
of SQL queries

*Find all patients born at or after 1950.*

```
function find_patients(conn)
    sql = """
    SELECT person_id
    FROM patient
    WHERE year_of_birth >= 1950
    """
    DBInterface.execute(conn, sql)
end
```

- What is SQL? Data is often stored in relational databases, and to retrieve it, we write queries in SQL.
- Popular databases with SQL interface include MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Redshift, BigQuery, and many others.
- Julia already has a number of libraries that let you interact with SQL databases.
- Why another tool?

*Find all patients born between*

1950

*and*

```
function find_patients(conn; start_year = nothing,  
                        end_year = nothing)  
  
    sql = ""  
    SELECT person_id  
    FROM patient  
    ""  
  
    conditions = String[]  
    if start_year != nothing  
        push!(conditions, "year_of_birth >= $start_year")  
    end  
    if end_year != nothing  
        push!(conditions, "year_of_birth <= $end_year")  
    end  
    if !isempty(conditions)  
        sql *= "\nWHERE " * join(conditions, " AND ")  
    end  
    DBInterface.execute(conn, sql)  
end
```

*Find all patients born between*

*and*

*living in*

*with conditions*

  

*who visited a doctor*

"location"

"condition\_occurrence"

"visit\_occurrence"

```
function find_patients(conn; start_year = nothing,  
                          end_year = nothing,  
                          state = nothing,  
                          conditions = []  
                          last_visit = nothing)
```

```
    sql = ""
```

```
    SELECT person_id
```

```
    FROM patient
```

```
    ""
```

```
    ???
```

```
    DBInterface.execute(conn, sql)
```

```
end
```

- Clearly, a more systematic approach is necessary.