```
INSERT INTO locations (name, description, latitude, longitude, geom)
VALUES (
    ’Empire State Building’,
    ’Famous skyscraper in New York City’,
    40.7484,
    -73.9857,
    ST_SetSRID(ST_MakePoint(-73.9857, 40.7484), 4326)
);
```

```
// PostGIS Extension Types
type Point;
type Polygon;
type Geometry = {Point, Polygon};
type ST_MakePoint(Float, Float): Point;
type ST_SetSRID(Geometry, Integer): Point;

stmt get_users(
  created_between: Option<Tuple<Integer, Integer>>,
  limit: Option<Integer>
): Array<Row> {
  ‘SELECT * FROM users‘
  if created_between {
    ‘WHERE created_at BETWEEN {created_between[0]} AND {created_between[1]}‘
  }
  if limit {
    ‘LIMIT {limit}‘
  }
}

stmt bulk_add_users(usernames: Array<String>): Array<Row> {
  ‘INSERT INTO users (name) VALUES‘
  for un in usernames; ‘,‘ {
    ‘({un})‘
  }
  ‘RETURNING *‘
}
```

```
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
  [ { * | expression [ [ AS ] output_name ] } [, ...] ]
  [ FROM from_item [, ...] ]
  [ WHERE condition ]
  [ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
  [ HAVING condition ]
  [ WINDOW window_name AS ( window_definition ) [, ...] ]
  [ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
  [ ORDER BY expression [ ASC | DESC | USING operator ] [ NULLS { FIRST | LAST } ]
     [, ...] ]
  [ LIMIT { count | ALL } ]
  [ OFFSET start [ ROW | ROWS ] ]
  [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } { ONLY | WITH TIES } ]
  [ FOR { UPDATE | NO KEY UPDATE | SHARE | KEY SHARE } [ OF from_reference [, ...]
     ] [ NOWAIT | SKIP LOCKED ] [...] ]
```

```
INSERT INTO table_name [ AS alias ] [ ( column_name [, ...] ) ]
  [ OVERRIDING { SYSTEM | USER } VALUE ]
  { DEFAULT VALUES | VALUES ( { expression | DEFAULT } [, ...] ) [, ...] | query }
  [ ON CONFLICT [ conflict_target ] conflict_action ]
  [ RETURNING { * | output_expression [ [ AS ] output_name ] } [, ...] ]
```

```
// Should accept
SELECT *
FROM notes
WHERE created_at BETWEEN '2024-02-01 00:00:00' AND '2024-02-15 23:59:59'
  AND content = '';

// Should reject
SELECT *
FROM notes
WHERE created_at BETWEEN '2024-02-01 00:00:00'
  AND content = '';
```

A query has to be checked against:

- All referenced tables and their columns have to exist. Use a symbol table per query to check aliases.
- All referenced columns must be of the correct type. Compare the type of the column with the type of the desired expression.
- Order of the clauses matters. Not checked if parsers accepts yet, but based on its specification, it should.
- `WHERE` clause should allow most operators.
- Default values for insert

Questions:

- Datatype of timestamp vs text, how to define?
- How should types be defined for extensions/parsed?