To write SQL queries, the users need to know the exact schema of the database. The query language Schema-free SQL proposed, enables the user to write queries using whatever partial schema they know. This is obtained by:

- Schema Relaxation: Users can use partial names for tables and attributes
- Join Path Relaxation: users need not specify the join path, including any intermediate relations that are involved

This project aims at understanding the limitations of the suggested approach, testing and improving the existing implementation.

### Proposed Approach

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Output query: SELECT Person1.name from Actor;</td>
</tr>
<tr>
<td>Expected result: Select Actor.person_id from Actor;</td>
</tr>
</tbody>
</table>

#### Query Parsing

Parse the input query to get schema elements from FROM/SELECT and WHERE clauses. Tool used: ANTLR.

#### Relation Tree Mapper

Generate relation trees from schema elements and map the merged rTrees to schema tables, by computing similarity evaluation using WUP similarity algorithm $\text{Sim}(\text{rTree}, R) = \text{Sim}(n(\text{rTree}), R) \cap \text{Sim}(\text{att}, R)$

#### Network Builder: Extended View Graph

Join all tables based on foreign key references, user specified join paths, the relation trees that are mapped to tables. The weights for each join path are calculated and optimal k are considered.

### SQL Composer

The vague schema elements are replaced with the mapped relations, Where clause is formed by choosing optimal join paths, from clause has all tables. These are combined to get final query.

### Testing

SQL grammar was modified to consider schema-elements in FROM Clause. The system was tested for all the input flexibilities mentioned in the paper, and analyzed the join networks formed, to identify the limitations in the methodology.