

Project Phase-2

Given a query at the fragmentation transparency level:

1. Parse the query
2. Perform query decomposition
3. Perform data localization

Assumptions:

- Queries are in CNF
- Queries are correct (both syntactically and semantically)
- No redundant/unnecessary conditions
- But in case an incorrect query is given, your system shouldn't crash. Should terminate properly/prompt user for correct input.

Task 1: Parsing

- Use any popular open-source SQL parser.
- Run the parser and read the output.
- Need to Demo the SQL parser and show the output.

Task 2: Decomposition

- Normalization, analysis, and elimination of redundancy are taken care of by assumptions!
- Rewriting: find the best relational algebra expression for the given query
 - The tree would be a nice structure to represent algebra expressions

Task 3: Rewriting

- Form the initial query tree
- Apply transformation rules and get "good" query trees
- Need selectivity factors for selections and joins
- For selections, assume selectivity to be uniform (0.5 or 0.3 etc)
- For joins, maintain a file with the selectivity factors
 - Consider them as inputs
- Use standard heuristics to reduce the time complexity

Task 4: Localization

- Localize the relational algebra expression generated in the previous step
- Should support

- Primary horizontal
- Derived horizontal
- Vertical
- All combinations above 3 (total 7 cases!)

Testing:

- You have to write a script that helps in populating your database with given csv files (populated with some pilot data) corresponding to each table in your application database. To understand this clearly, refer to the example below :
 - Consider that your database has 3 tables: table1, table2, table3. These will have some relationship between them. You will have 3 csv files that will correspond to these tables. A script should take input of these 3 csv's and then populate the database (based on the details in the system catalog, i.e the fragmentation and allocation schema) with the data taking care of the order to manage the relationships which the tables have in between them. During evals, the pilot data might be modified and then the script should run and insert the new data accordingly.
 - Given a .csv file of application database schema, with fragmentation and allocation schema, your utility should populate the system catalog. The format of the csv will be <Fragment Name, Fragmentation Type, Allocated Site, Fragmentation Condition, Columns(on that site)>.
 - Submit the pilot data also in your zip file submission.
 - NOTE: Any legal data for the relation should not give any errors when inserted in the tables. The code should not hard code the system to any particular database schema.

Deliverables:

- Upload the zip file with all code files, a readme file, csv files, and a report.
- The report should be in pdf format. In report:
 - Include the ER and Relational Model diagram.
 - Explain the data structures/algorithms in detail
 - For a few queries, in detail include the initial query tree, final optimized query tree, and output.
- Submission file name format - TeamName_phase2.zip
- Only one person from each team needs to submit on the portal.