LECTURE 4

USING HEURISTICS IN QUERY OPTIMIZATION (1/3)

- Apply heuristic rules to modify representation of a query which is in the form of a **query tree** or **query graph** data structure to improve performance
- Parser of a high-level query first generates initial internal representation, which is then optimized according to heuristic rules
- A query execution plan is then generated to execute groups of operations based on access paths available on the files involved in the query

USING HEURISTICS IN QUERY OPTIMIZATION (2/3)

- One of the main heuristic rules is to apply SELECT and PROJECT operations before applying the JOIN or other binary operations

- This is because the **size** of the file **resulting** from a binary operation, such as JOIN, is usually a multiplicative function of the sizes of input files

- **SELECT and PROJECT operations reduce the size of a file and hence should be applied before a JOIN or any other binary operation**

USING HEURISTICS IN QUERY OPTIMIZATION (3/3)

- Query tree is used to represent a relational algebra expression
- Query graph is used to represent a relational calculus expression

QUERY TREE (1/2)

- It is a tree data structure that corresponds to a relational algebra expression
- It represents **input relations** of the query as **leaf nodes** of the tree
- **Relational algebra operations as internal nodes**

QUERY TREE (2/2)

- Execution of a query tree consists of executing internal node operation whenever its operands are available and then replacing that internal node by the relation that results from executing the operation

- Execution terminates when the root node is executed and produces the result relation for the query

- It represents a **specific order of operations** for executing a query

QUERY GRAPH (1/2)

- **Relations** in the query are represented by relation nodes which are displayed as **single circles**
- **Constant values** from the query selection conditions are represented by constant nodes which are displayed as **double circles**
- **Selection and join** conditions are represented by the **graph edges**

QUERY GRAPH (2/2)

- **Attributes to be retrieved** from each relation are displayed in **square brackets** above each relation
- **It does not indicate an order** on which operations are to perform first
- There is only a **single graph** corresponding to **each query**
- In practice, **query tree is preferred**