Query Management

When an SQL query is expressed via an interface to some DBMS

- An application program
- An end-user

the query goes through a number of stages:

1. The query is checked for ***syntax*** errors.

   This phase is often referred to as the PARSING phase.

- Is the statement valid
- are the clauses in the correct order
- are there sufficient arguments supplied etc.

2. The query is then validated against the data stored in the data dictionary.

- Does the table(s) exist in the database
- Do the column names specified exist in the named table(s)
- Does the user (application) have the required privileges (eg. are they allowed to DELETE)

3. Next, the query is ***optimised***.

   The DBMS contains a piece of software, the ***query optimiser***, which translates the query into an execution plan.

   There may be several ways to execute a query. The query optimiser must determine the ***best*** or ***optimal*** execution plan for the query.

4. The query is then executed.

5. The results of query are returned to the user/application.

Query Optimisation

A ***query optimiser*** is a piece of software that is responsible for determining the ***optimal*** execution plan of a query.

A query might be executed in more than one way. Which is the best (most optimal) ?

Example

Suppose we have the following tables:

Student(**StudID**, Surname, Forename Address, Phone, Course)

Subjects(**SubjectId**, Title)

StudSubjects(**StudId**, **SubjectId**, GradeAvg)

Suppose we wish to know:

Which students are taking the subject ***Relational Database Systems (subject Id = 2)***

The application or the user might issue the following query:

```
SELECT Surname, Forename
FROM Student St, StudSubjects Su
WHERE St.StudNo = Su.StudNo AND
      Su.SubjectId = 2;
```

There are two ways in which this query can be implemented:

METHOD 1

Execution Plan:

1. Join the Student table and the StudSubjects table.

2. Extract those records with a subject Id = 2

Method 2

Execution Plan:

- Extract from the table ***StudSubjects*** all the records with SubjectId = 2

2. Join this result with the table ***Student*** to get the corresponding student name details.

Assume that:

- the Student Table contains 100 records
- the StudSubjects table contains 1000 records
- Only 50 students are enrolled in ***Relational Database Systems***

The Query optimiser uses a performance measure to determine the most optimal execution plan.

The number of ***Inputs*** and ***Outputs*** (ie. Reads & Writes) is a commonly used performance measure.

Calculate the number of I/O's for each method.

Compare the I/O's for each method.

Select the best (lowest).

Method 1

- Join the tables ***Student*** and ***StudSubjects***
  Read 100 records from student table
  Compare each record with 1000 records in StudSubjects (100 X 1000 = 100,000 reads **(Is)**)
- Write the joined (matching) records back to disk (1000 writes **(Os)**)
- Read 1,000 records and write 50 back to disk (1,050 I/O's)

TOTAL: 102,150 I/O's

Method 2

- Read 1000 records in the table ***StudSubjects*** (1000 reads **(Is)**). Check the subjectId of each.
- Write 50 records to disk (50 Writes **(Os)**)
- Read 50 records to match with each of 100 student records (50 X 100 = 5000 reads **(Is)**)
- Write 50 records (50 writes **(Os)**)

TOTAL = 1000 + 50 + 5,000 + 50 = 6,100 I/O's

Compare

Method 1	102,150 I/O's
Method 2	6,100 I/O's

Method 2 is the most optimal