SQL SERVER Performance Tuning Notes

Don't ever do SELECT *. It's really a bad idea. Why?

Well if you do select * in your query then you will end up with many problems; some of the are given below-

1. **Bad Query Plan / Proper Indexes may be ignored / Less Index seeks - Too much IO & memory**
2. **Missing indexes in DMVs**
3. **What if we add more fields to the existing table**

Let's explore each of the above point in detail.

**Bad Query Plan / Proper Indexes may be ignored / Less Index seeks - Too much IO & memory**

Now let's check out below queries written in the box below. Both the below queries are identical with the only difference of selecting columns. In one we are selecting all columns using * and in another query we are picking the fields I need.

```
USE ADVENTUREWORKS2012
GO

SELECT * FROM [Sales].[SalesOrderDetail]
WHERE ProductID = 776

SELECT SalesOrderID,SalesOrderDetailID FROM [Sales].[SalesOrderDetail]
WHERE ProductID = 776
```

Now put these queries in a single query window and include actual execution plan. Now execute both the queries and check out the execution plan. Now this is the excellent feature of SQL Server management studio where we can execute multiple queries in a single query window and compare their execution plans, we need to check which one has done bulk of the work (processing time and cost).

```
SELECT * FROM [Sales].[SalesOrderDetail]
WHERE ProductID = 776

SELECT SalesOrderID,SalesOrderDetailID FROM [Sales].[SalesOrderDetail]
WHERE ProductID = 776
```
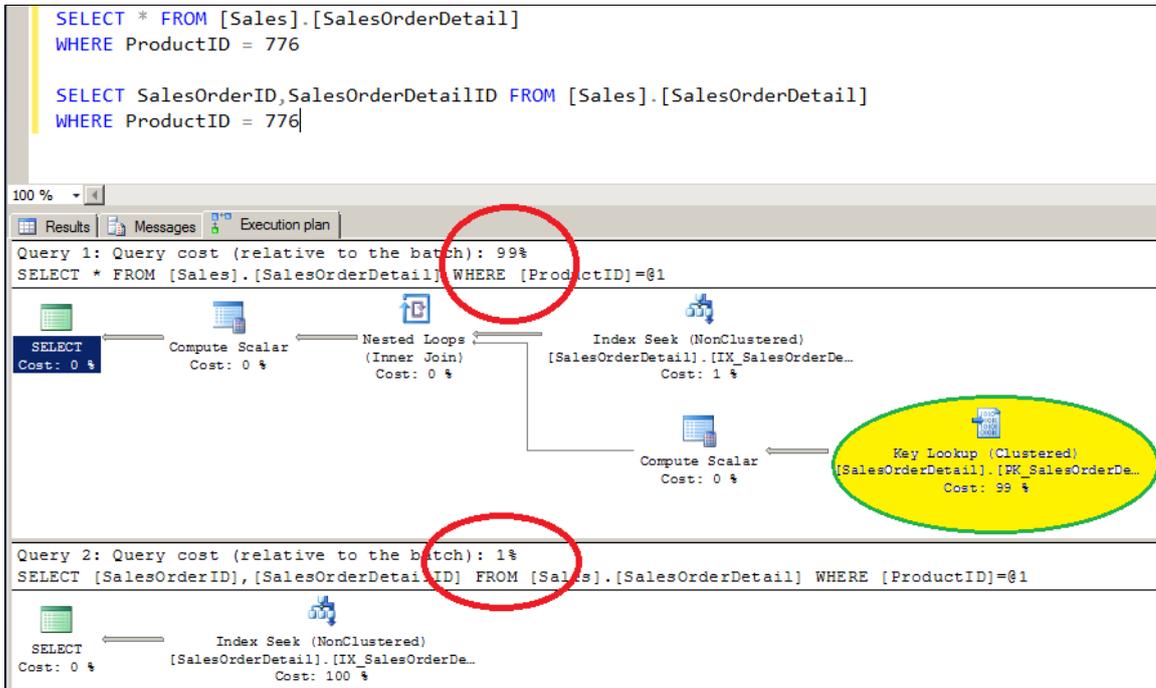
In second query we are only doing Non clustered Index seek and in the first query we are doing nonClustered index seek with a clustered key lookup to the main table to fetch remaining columns. In the second query we are consuming more IO and memory.

Top 1 query is 99% more expensive then the second query. 99% faster means 99% more scalable application. That's why we should always use columns we require in SELECT clause.

Missing indexes in DMVs

These kinds of queries will also hoses up the missing index DMVs. DBA use DMV's to find out missing indexes. DMV's (Dynamic Management Views) used to tell us which all missing indexes we should add to tune up the queries.

But the problem is when we select *, Missing Index DMVs will give bad results out of that because they can't predict what fields we really need. If you just named 2, 3 or 4 fields you needed missing DMVs can directly say that hey there users are constantly querying for these fields; let's add these indexes to tune up the queries.

Now if you use select * then DMVs will inform us a ridiculously wide index which DBA's don't agree upon. They say I don't going to create this as it is too intensive on my database.

That's why we should always use columns we require in SELECT clause.

What if we add more fields to the existing table?

Now let's say the development is done & you think everything is done. Now the DBA or any other developer adds 5 more fields to the table. E.g. a VARCHAR (4000) column is added.

Now as per your query you don't need them because when you created that SQL these fields were not there. So here comes the additional cost we have to pay. If we would have named our columns then we would not be in any sort of trouble.

That's why we should always use columns we require in SELECT clause.