**EXECUTION PLANS**

TO PROCESS OR RUN ANY QUERY, ONE EXECUTION PLAN IS AUTO GENERATED INTERNALLY.

THIS EXECUTION PLAN IS USED BY "QUERY OPTIMIZER" COMPONENT TO ENSURE THE BEST USAGE OF RESOURCES (IO, MEMORY & CPU).

TYPES OF EXECUTION PLANS (QUERY STATISTICS) :

1. **ESTIMATED EXECUTION PLAN** : AUTO GENERATED **BEFORE** THE QUERY EXECUTION

2. **ACTUAL EXECUTION PLAN** : AUTO GENERATED **AFTER** THE QUERY EXECUTION

3. **LIVE EXECUTION PLAN** : AUTO GENERATED **DURING** THE QUERY EXECUTION

ABOVE EXECUTION PLANS ALSO GIVE RECOMMENDATIONS ON MISSING INDEXES FOR A GIVEN QUERY.

**HOW TO VERIFY THE EXECUTION PLANS?**

LAUNCH SSMS TOOL > CONNECT TO SQL SERVER

NEW SESSION WINDOW > TEST A QUERY.

SELECT

PRODUCTS\_DATA.EnglishProductName,

SUM(SALES\_DATA.SalesAmount) AS TotalSales

FROM SALES\_DATA

INNER JOIN

PRODUCTS\_DATA

ON

SALES\_DATA.ProductKey = PRODUCTS\_DATA.ProductKey

GROUP BY PRODUCTS\_DATA.EnglishProductName

RIGHT CLICK ANYWHERE IN THE SESSION WINDOW > SELECT **ESTIMATED** EXECUTION PLAN.

THEN EXECUTE THE QUERY.

IN THE OUTPUT WINDOW > WE SEE THE PLAN.

**HOW TO UNDERSTAND THE EXECUTION PLAN?**

WE READ THE EXECUTION PLAN FROM RIGHT TO LEFT.

IDENTIFY THE TABLE THAT HAS MORE QUERY COST.

WE NEED TO OPTIMIZE IT.

**HOW DO YOU OPTIMIZE THE QUERY COST?**

RIGHT CLICK QUERY > **ANALYSE IN DTA**

GO TO TUNING OPTIONS > ADVANCED > SET INDEX SIZE (EX: 100 MB)

START ANALYSIS > GET THE RECOMMENDATIONS. RUN TSQL SCRIPTS MANUALLY.

**COMPONENTS IN THE EXECUTION PLAN:**

1. TABLE SCAN : TO READ ENTIRE TABLE DATA
2. INDEX SCAN : TO READ THE DATA FROM THE RELEVANT INDEX PAGES
3. INDEX SEEK : TO SEARCH THE DATA FROM THE RELEVANT INDEX PAGES

FROM EXECUTION PLAN, WE MAY GET THE RECOMMENDATIONS FOR MISSING INDEX.

IF ANY : CREATE SUCH MISSING INDEX.

HOVER ANY EXECUTION PLAN ITEM, WE SEE COST ANALYSIS:

**1. PHYSICAL SCAN:** REFERS TO DISK LEVEL READS & WRITES. INVOLVES MORE TIME, COST.

**2. LOGICAL SCAN:** REFERS TO MEMORY LEVEL READS & WRITES. FASTER, CHEAPER.

**3. ESTIMATED EXECUTION MODE:** REFERS TO THE STORAGE OF DATA FOR A GIVEN TABLE.

ROW STORE OR COLUMN STORE.

**4. QUERY COST TYPES: THRESHOLD : 0.05**

1. **IO COST**: REFERS TO THE DISK (DATA FILES) IO FOR READS & WRITES

FOR HIGHER IO COST, TRACK MISSING INDEXES BY USING DTA

2. **CPU COST**: REFERS TO THE THREAD MANAGEMENT FACTOR BASED ON "NUMA" NODES ON THE PROCESSOR. ADDING NEW PROCESSOR NODE=HOT CPU

SET PROPER THREAD COUNT.

SERVER > PROPERTIES > PROCESSOR > set Processor **Thread Count**

FOR SHARED SYSTEMS, BOOST SQL SERVER PRIORITY.

**3.SUB TREE COST**: REFERS TO COST INVOLVED IN ANALYSING COMPILE PLANS

**RECOMPILE**(REFRESH) THE DATABASE [LIKE SPs, VIEWS, ETC.].

**4. OPERATOR COST**: REFERS TO QUERY KEYWORDS, OPERATIONS WITHIN SQL QUERIES.

EX: IN OPERATOR IS FASTER THAN MULTIPLE EQUALITY OPERATORS.

FOR HIGHER OPERATOR COST, ENSURE **STATS UPDATES.**

AVOID SPOOLING ACTIVITY. FOR THIS, LIMIT THE USE OF TEMPDB.

**SPOOLING** : TO LOAD TABLE DATA FROM mdf OR ndf TO TEMP DB.

* IF THERE ARE ANY OPEN TRANSACTIONS, WE EXPERIENCE QUERY BLOCKING.
* TO IDENTIFY THE QUERY BLOCKING:

SELECT \* FROM SYSPROCESSES WHERE BLOCKED != 0

TO IDENTIFY THE DETAILS OF EACH PROCESS, WE NEED TO USE BELOW SP:

**EXEC SP\_WHO2**

