Sl No	**Incident/Question-Concept**	**Solution/Answer/Discussion**

121	Missing Indexes	-- Extract missing Index data
		SELECT t.name AS 'table',
		(avg_total_user_cost * avg_user_impact) * (user_seeks + user_scans)
		AS 'potential_impact',
		'CREATE NONCLUSTERED INDEX ix_IndexName ON ' + SCHEMA_NAME(t.schema_id)
		+ '.' + t.name COLLATE DATABASE_DEFAULT + ' ('
		+ ISNULL(d.equality_columns, '')
		+ CASE WHEN d.inequality_columns IS NULL THEN ''
		ELSE CASE WHEN d.equality_columns IS NULL THEN ''
		ELSE ','
		END + d.inequality_columns
		END + ') ' + CASE WHEN d.included_columns IS NULL THEN ''
		ELSE 'INCLUDE (' + d.included_columns + ')'
		END + ';' AS 'create_index_statement'

		FROM sys.dm_db_missing_index_group_stats AS s INNER JOIN sys.dm_db_missing_index_groups AS g ON s.group_handle = g.index_group_handle INNER JOIN sys.dm_db_missing_index_details AS d ON g.index_handle = d.index_handle INNER JOIN sys.tables t WITH (NOLOCK) ON d.OBJECT_ID = t.OBJECT_ID WHERE d.database_id = DB_ID() AND s.group_handle IN (SELECT TOP 500 group_handle FROM sys.dm_db_missing_index_group_stats WITH (NOLOCK) ORDER BY (avg_total_user_cost * avg_user_impact) * (user_seeks + user_scans) DESC) AND t.name LIKE 'Person' ORDER BY (avg_total_user_cost * avg_user_impact) * (user_seeks + user_scans) DESC;
122	Gen rebuild/reorganize	------------------------------ ---- FOR REBUILD OF INDICES WITH FRAGMENTATION > 30 ; FF=90 FOR

```
CLUSTERED AND FF=95 FOR NON-CLUSTERED
SELECT i.name AS IndexName
       ,object_name(ips.object_id) AS
TableName
  ,ips.index_type_desc AS IndexType
 ,ips.avg_fragmentation_in_ percent
 ,i.fill_factor
 ,CASE WHEN type = 1 then 'ALTER INDEX
' + name + ' ON ' +
object_schema_name(ips.object_ id) +
'.' + OBJECT_NAME(ips.object_id) + '
REBUILD WITH (FILLFACTOR=90); GO' WHEN
type <> 1 then 'ALTER INDEX ' + name +
' ON ' + object_schema_name(ips.object_
id) + '.' + OBJECT_NAME(ips.object_id)
+ ' REBUILD WITH (FILLFACTOR=95); GO'
END AS COMMAND
    ,CASE WHEN type = 1 then '90
percent considering the row size and to
have space for the leaf data' WHEN type
<> 1 then '95 percent as the space is
only effective for the cluster key in
NCI' END AS REASON
FROM sys.dm_db_index_physical_stats
(DB_ID(),NULL, NULL, NULL ,NULL) AS ips
INNER JOIN sys.indexes AS i ON
ips.OBJECT_ID = i.OBJECT_ID
 AND ips.index_id = i.index_id AND
avg_fragmentation_in_percent > 30 and
type<>0  -- and page_count >1000
;
GO


----  FOR REORGANIZE OF INDICES WITH
AVG.FRAGMENTATION > 0 AND < 30

SELECT i.name AS IndexName
       ,object_name(ips.object_id) AS
TableName
  ,ips.index_type_desc AS IndexType
 ,ips.avg_fragmentation_in_ percent
 ,i.fill_factor
 ,'ALTER INDEX ' + name + ' ON ' +
object_schema_name(ips.object_ id) +
```

		'.' + OBJECT_NAME(ips.object_id) + ' REORGANIZE; GO' FROM sys.dm_db_index_physical_stats (DB_ID(),NULL, NULL, NULL ,NULL) AS ips INNER JOIN sys.indexes AS i ON ips.OBJECT_ID = i.OBJECT_ID AND ips.index_id = i.index_id AND avg_fragmentation_in_percent > 0 and avg_fragmentation_in_percent < 30 and type<>0; GO
123	Gen Rebuild with FF based on index type	SELECT b.name,object_schema_name(a.object_id) AS 'schema', OBJECT_NAME(a.object_id) as 'Table',avg_fragmentation_in_percent,fill_factor,CASE WHEN type = 1 then 'ALTER INDEX ' + name + 'ON ' + object_schema_name(a.object_id) + ' REBUILD WITH (FILLFACTOR=90); GO ' WHEN TYPE <> 1 then 'ALTER INDEX ' + name + 'ON ' + object_schema_name(a.object_id) + ' REBUILD WITH (FILLFACTOR=95); GO ' END AS COMMAND FROM sys.dm_db_index_physical_stats (DB_ID(IDOfYourDB), NULL, NULL, NULL, NULL) AS a JOIN sys.indexes AS b ON a.object_id = b.object_id AND a.index_id = b.index_id where avg_fragmentation_in_percent >30 AND b.type <> 0 GO
124	Unused (since last restart of SQL server) Indexes	SELECT sum(Total_Scans) as Total_Scans,sum(Total_Seeks) as Total_Seeks,sum(Total_LookUps) as Total_LookUps,name,[schema],[Table],avg_fragmentation_in_percent

		FROM (SELECT u.user_scans as Total_Scans,u.user_seeks as Total_Seeks,u.user_lookups as Total_LookUps,b.name,object_schema_name (a.object_id) AS [schema], OBJECT_NAME(a.object_id) as [Table],avg_fragmentation_in_percent FROM sys.dm_db_index_physical_stats (12, NULL, NULL, NULL, NULL) AS a INNER JOIN sys.indexes AS b ON a.object_id = b.object_id AND a.index_id = b.index_id INNER JOIN sys.dm_db_index_usage_stats u on u.index_id = b.index_id and u.object_id = b.object_id and u.object_id = a.object_id INNER JOIN sys.objects o on a.object_id = o.object_id INNER JOIN sys.sysdatabases d on a.database_id = d.dbid where u.user_scans =0 and u.user_lookups=0 and u.user_seeks =0)TCC group by name,[schema],[Table],avg_fragmentation _in_percent GO
125	Total_Scans, Total Seeks,Total_Lookups,Last Scan, Last Seek, Last Lookup,schema,name,AvgFragPercent	USE DBNAME GO SELECT sum(Total_Scans) as Total_Scans,sum(Total_Seeks) as Total_Seeks,sum(Total_LookUps) as Total_LookUps,name,[schema],[Table],avg _fragmentation_in_percent FROM (SELECT u.user_scans as Total_Scans,u.user_seeks as

		Total_Seeks,u.user_lookups as Total_LookUps,b.name,object_schema_name (a.object_id) AS [schema], OBJECT_NAME(a.object_id) as [Table],avg_fragmentation_in_percent FROM sys.dm_db_index_physical_stats (12, NULL, NULL, NULL, NULL) AS a INNER JOIN sys.indexes AS b ON a.object_id = b.object_id AND a.index_id = b.index_id INNER JOIN sys.dm_db_index_usage_stats u on u.index_id = b.index_id and u.object_id = b.object_id and u.object_id = a.object_id INNER JOIN sys.objects o on a.object_id = o.object_id INNER JOIN sys.sysdatabases d on a.database_id = d.dbid)TCC group by name,[schema],[Table],avg_fragmentation_in_percent GO***********************
125	--------------------------------------- Duplicate / Overlapping Indexes ------------------------	;WITH DupIndex AS (SELECT Sch.[name] AS SchemaName, Obj.[name] AS TableName, Ind.[name] AS IndexName,

	----------------	INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 1) AS Col1, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 2) AS Col2, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 3) AS Col3, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 4) AS Col4, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 5) AS Col5, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 6) AS Col6, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 7) AS Col7, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 8) AS Col8, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 9) AS Col9, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 10) AS Col10, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 11) AS Col11, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 12) AS Col12, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 13) AS Col13, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 14) AS Col14, INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 15) AS Col15,

		INDEX_COL(Sch.[name] + '.' + Obj.[name], Ind.index_id, 16) AS Col16 FROM sys.indexes Ind INNER JOIN sys.objects Obj ON Ind.[object_id] = Obj.[object_id] INNER JOIN sys.schemas Sch ON Sch.[schema_id] = Obj.[schema_id] WHERE index_id > 0) SELECT DI1.SchemaName, DI1.TableName, DI1.IndexName, DI2.IndexName AS OverLappingIndex, DI1.Col1, DI1.Col2, DI1.Col3, DI1.Col4, DI1.Col5, DI1.Col6, DI1.Col7, DI1.Col8, DI1.Col9, DI1.Col10, DI1.Col11, DI1.Col12, DI1.Col13, DI1.Col14, DI1.Col15, DI1.Col16 FROM DupIndex DI1 INNER JOIN DupIndex DI2 ON DI1.tablename = DI2.tablename AND DI1.indexname <> DI2.indexname AND DI1.Col1 = DI2.Col1 AND (DI1.Col2 IS NULL OR DI2.Col2 IS NULL OR DI1.Col2 = DI2.Col2) AND (DI1.Col3 IS NULL OR DI2.Col3 IS NULL OR DI1.Col3 = DI2.Col3) AND (DI1.Col4 IS NULL OR DI2.Col4 IS NULL OR DI1.Col4 = DI2.Col4)

		AND (DI1.Col5 IS NULL OR DI2.Col5 IS NULL OR DI1.Col5 = DI2.Col5) AND (DI1.Col6 IS NULL OR DI2.Col6 IS NULL OR DI1.Col6 = DI2.Col6) AND (DI1.Col7 IS NULL OR DI2.Col7 IS NULL OR DI1.Col7 = DI2.Col7) AND (DI1.Col8 IS NULL OR DI2.Col8 IS NULL OR DI1.Col8 = DI2.Col8) AND (DI1.Col9 IS NULL OR DI2.Col9 IS NULL OR DI1.Col9 = DI2.Col9) AND (DI1.Col10 IS NULL OR DI2.Col10 IS NULL OR DI1.Col10 = DI2.Col10) AND (DI1.Col11 IS NULL OR DI2.Col11 IS NULL OR DI1.Col11 = DI2.Col11) AND (DI1.Col12 IS NULL OR DI2.Col12 IS NULL OR DI1.Col12 = DI2.Col12) AND (DI1.Col13 IS NULL OR DI2.Col13 IS NULL OR DI1.Col13 = DI2.Col13) AND (DI1.Col14 IS NULL OR DI2.Col14 IS NULL OR DI1.Col14 = DI2.Col14) AND (DI1.Col15 IS NULL OR DI2.Col15 IS NULL OR DI1.Col15 = DI2.Col15) AND (DI1.Col16 IS NULL OR DI2.Col16 IS NULL OR DI1.Col16 = DI2.Col16) ORDER BY DI1.SchemaName,DI1.TableName,DI1.IndexName Q2.

		--- Total_Scans,Total Seeks,Total_Lookups,Last Scan, Last Seek, Last Lookup,schema,name,AvgFragPercent USE DBNAME GO SELECT top 1000 sum(Total_Scans) as Total_Scans,sum(Total_Seeks) as Total_Seeks,sum(Total_Lookups) as Total_LookUps,last_user_scan,last_user_seek,last_user_lookup,name,[schema],[Table],avg_fragmentation_in_percent FROM (SELECT u.user_scans as Total_Scans,u.user_seeks as Total_Seeks,u.user_lookups as Total_Lookups,u.last_user_scan,u.last_user_seek,u.last_user_lookup,b.name,obje

		ct_schema_name(a.object_id) AS [schema], OBJECT_NAME(a.object_id) as [Table],avg_fragmentation_in_percent FROM sys.dm_db_index_physical_stats (12, NULL, NULL, NULL, NULL) AS a INNER JOIN sys.indexes AS b ON a.object_id = b.object_id AND a.index_id = b.index_id INNER JOIN sys.dm_db_index_usage_stats u on u.index_id = b.index_id and u.object_id = b.object_id and u.object_id = a.object_id INNER JOIN sys.objects o on a.object_id = o.object_id INNER JOIN sys.sysdatabases d on a.database_id = d.dbid TCC group by last_user_scan,last_user_seek,last_user_lookup,name,[schema],[Table],avg_fragmentation_in_percent

		GO
126	Name some metrics to be considered for disk bottlenecks issues.	Avg Writes / Second Avg Second / Write Avg Reads / Second Avg Second / Read Max Writes / Second Max Second / Write Max Reads / Second Max Second / Read Example, Avg being too close to max indicates issues.Has to be looked into the pending io flag (1-pending). Min Writes / Second Min Second / Write Min Reads / Second Min Second / Read
127	In an active-active cluster setup, is it	Having different set for each instance makes sense, as to avoid a single point of failure. If one instance goes down, the other should not have interruption.

	better to have one set of shared disks for both the instances or two different sets?	
128	Infrastructure requirements for Always ON.	VNN (Virtual Network Name akin to Virtual cluster name) DNS names for the AV group. Virtual IPs Port numbers Non-shared disk groups for each node (unlike Cluster setup)
129	List out order wise , the resources for performance starting with the most important.	1.RAM (so that the applications have the sufficient memory to sit in in the first place, then comes CPU) 2.CPU (leverage upon the RAM considerations. Processor Core architecture like core to duo , dual core etc to be considered carefully) 3.Storage