Documentation:	**Performance analysis report for** JPBCBSDB **server**
Author:	**Sabyasachi Choudhury**
Version:	**V 1.0**
Date of documentation:	**18 June 2025**

Introduction

GeoPITS is currently managing 1000+ SQL databases remotely for our global customers. We are trusted by more than 100 clients across the globe. With 75+ database experts, our team is working on creating a minimal studio dedicated to building meaningful business processes and extensive support models with which we help companies build, manage, optimize and leverage data to empower their business. We are also a certified Microsoft Partner. Combined with an understanding of business goals and effective project management, we deliver efficient SQL server solutions.

About Author of the report

Sabyasachi Choudhury, Database Administrator who has 3+ years of experience in Microsoft data platform. Specialist in SQL Server administration and Performance tuning. He has certifications as mentioned below

- Microsoft Certified Azure Database Administrator

Analysis Report

Azure SQL Database analysis was performed by DB team to identify whether SQL instances are configured as per the industry standards and follow the SQL best practices. Analysis took place at Bangalore office. The output of this analysis report would give you complete insight into current SQL Server instance configuration. The report also provides recommendations to achieve better performance as per Microsoft standard and current workload.

Technical Environment

Environment	JPBCBSDB
SQL Server Collation	SQL_Latin1_General_CP1_CI_AS
SQL Server Edition	Enterprise Edition (64-bit)
SQL Server Machine Name	JPBCBSDB
SQL Server Clustered	No
SQL Server Version	15.0.4415.2
SQL Server Version Level	RTM
CPU count	72
RAM	224 GB
Configured Mem GB	176 GB
Availability Group	Yes
In-Memory Optimization	No

System Analysis

1. SQL Server Service Pack:

Microsoft distributes fixes in service packs. Service packs keep the product current. They include hotfixes and fixes to issues that are reported through the Microsoft SQL Server community. Service packs may also include issues that our support organization feels must be addressed. The most common metrics that are applied for triage of those items are customer call volume and customer satisfaction scores. These updates and components are conveniently bundled for easy downloading.

Each new service pack contains all the fixes that are in previous service packs, together with any new fixes. Latest Cumulative Updates need to be updated on the current SQL Instance.

Build	File version	KB / Description	Release Date
15.0.4430.1	2019.150.4430.1	5054833 Cumulative update 32 (CU32) for SQL Server 2019 Latest CU	45715
15.0.4420.2	2019.150.4420.2	5049296 Cumulative update 31 (CU31) for SQL Server 2019	45701

2. Trace Flags:

Trace flags are used to set specific server characteristics or to alter a particular behavior. Trace flags are frequently used to diagnose performance issues or to debug stored procedures or complex computer systems, but they may also be recommended by Microsoft Support to address behavior that is negatively impacting a specific workload.

Instance	Trace No	Description	Current Status	Reco Status
JPBCBSDB	TF 4199	Trace Flag that is used to enable the query optimizer hotfixes	Disabled	Enabled
	TF 3226	Avoids writing success messages into your error log	Disabled	Enabled
	TF 1118	All files of a file group grow at the same time	Disabled	Enabled
	TF 1117	Disable usage of mixed extents	Disabled	Enabled
	TF 1222	Capture deadlock details	Enabled	Enabled
	TF 2371	Change the threshold for statistics update	Disabled	Enabled
	TF9492	This trace flag enables DTC support for availability group configurations	Disabled	Enabled
	TF9495	Enables support for SQL Server's DTC for XA transactions in Always on Availability Groups.	Disabled	Enabled

Note: Above trace flags need to be enabled for better performance of SQL instances.

3. Fill Factor Reconfiguration

Currently, several indexes are configured with a fill factor of 0, which can lead to increased page splits and fragmentation, especially on tables with frequent insert and update operations. To optimize index performance and reduce fragmentation, it is recommended to reconfigure the fill factor to 95% for relevant indexes.

Setting the fill factor to 95% leaves space on each page for future growth, minimizing page splits and improving overall writing performance.

4. Cost Threshold for Parallelism Adjustment

The current SQL Server configuration sets the cost threshold for parallelism at 5, which is the default value. This low threshold can cause SQL Server to use parallel execution plans for relatively inexpensive queries, potentially leading to unnecessary CPU overhead and reduced overall system performance.

To address this, it is recommended to increase the cost threshold for parallelism to 50. Raising this value ensures that only more complex and resource-intensive queries are considered for parallel execution, while simpler queries continue to run serially.

This adjustment can help reduce CPU contention, improve query performance, and enhance system stability, especially in environments with high concurrency.

5. Force Parameterization:

You can override the default simple parameterization behavior of SQL Server by specifying that all SELECT, INSERT, UPDATE, and DELETE statements in a database be parameterized, subject to certain limitations. Forced parameterization is enabled by setting the PARAMETERIZATION option to FORCED in the ALTER DATABASE statement. Forced parameterization may improve the performance of certain databases by reducing the frequency of query compilations and recompilations. Databases that may benefit from forced parameterization are generally those that experience high volumes of concurrent queries from sources such as point-of-sale applications.

DB Name	Forced parameterization	Recommended value
BSGACCOUNTING	OFF	ON

6. Tempdb Analysis

The tempdb system database is a global resource that's available to all users connected to the instance of SQL Server or connected to Azure SQL Database. tempdb holds:

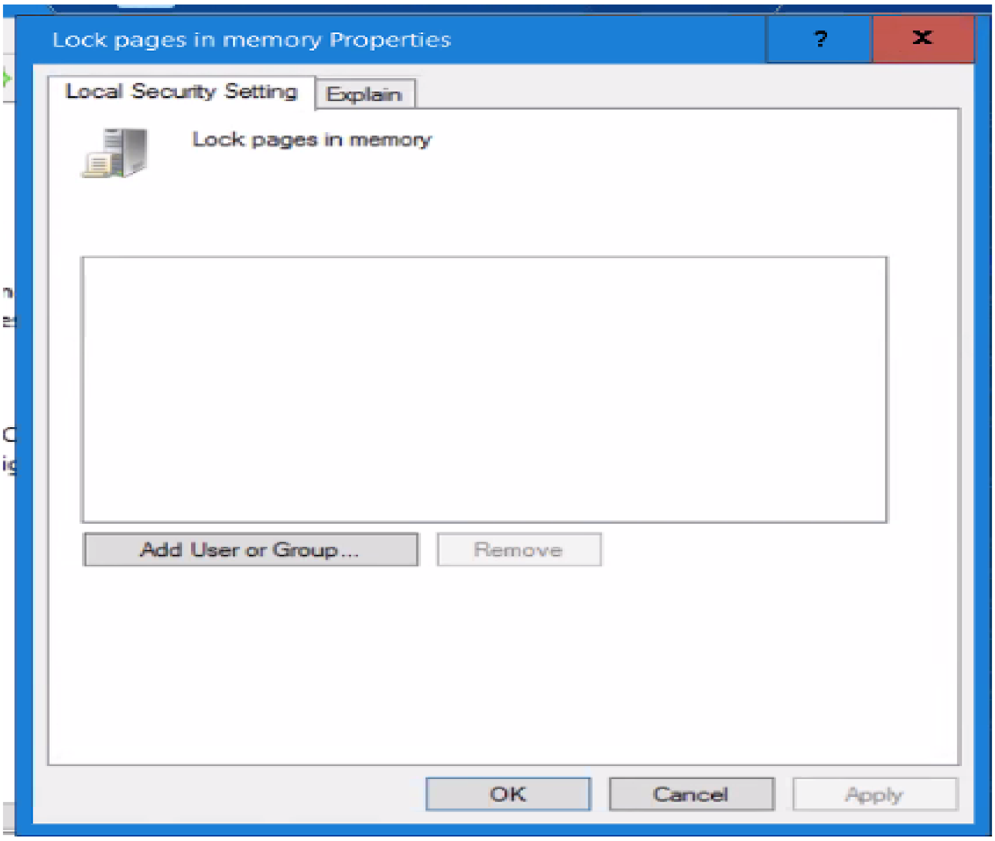
- **Temporary user objects** that are explicitly created. They include global or local temporary tables and indexes, temporary stored procedures, table variables, tables returned in table-valued functions, and cursors.
- **Version stores**, which are collections of data pages that hold the data rows that support features for row versioning. There are two types: a common version store and an online-index-build version store. The version stores contain:
    - Row versions that are generated by data modification transactions in a database that uses READ COMMITTED through row versioning isolation or snapshot isolation transactions.
    - Row versions that are generated by data modification transactions for features, such as online index operations, Multiple Active Result Sets (MARS), and AFTER triggers.
- **Internal objects** that the database engine creates. They include:
    - Worktables to store intermediate results for spools, cursors, sorts, and temporary large object (LOB) storage.
    - Work files for hash join or hash aggregate operations.
    - Intermediate sort results for operations such as creating or rebuilding indexes (if SORT_IN_TEMPDB is specified), or certain GROUP BY, ORDER BY, or UNION queries.

ServerName	Number of Database Files	Number of CPU	Required Files
JPBCBSDB\JPBCBSDB	4	72	8

7. Lock page in Memory

When the OS is running low on the memory, it starts clearing the cache of various applications which are installed on the server. During the process, Windows also clears the plan cache for SQL Server and releases the memory back to the OS. However, there are some cases where the memory is released back to OS even though there is no memory pressure. In such a scenario we should Enable Lock Pages in Memory (LPIM).

Add both the SQL Server and SQL Server Agent service accounts to the LPIM policy via Local Group Policy, then restart the services. This ensures SQL Server retains its memory in physical RAM, improving stability and performance.



8. Table rebuilds

Below tables are highly fragmented. This situation happens because of missing cluster index and whole table relay on heap. The solution is to rebuild the tables. This cause high forward record reference results in high usage of IO and degrades the overall performance of database.

Tables without Cluster Index:

DB Name: BSGACCOUNTING

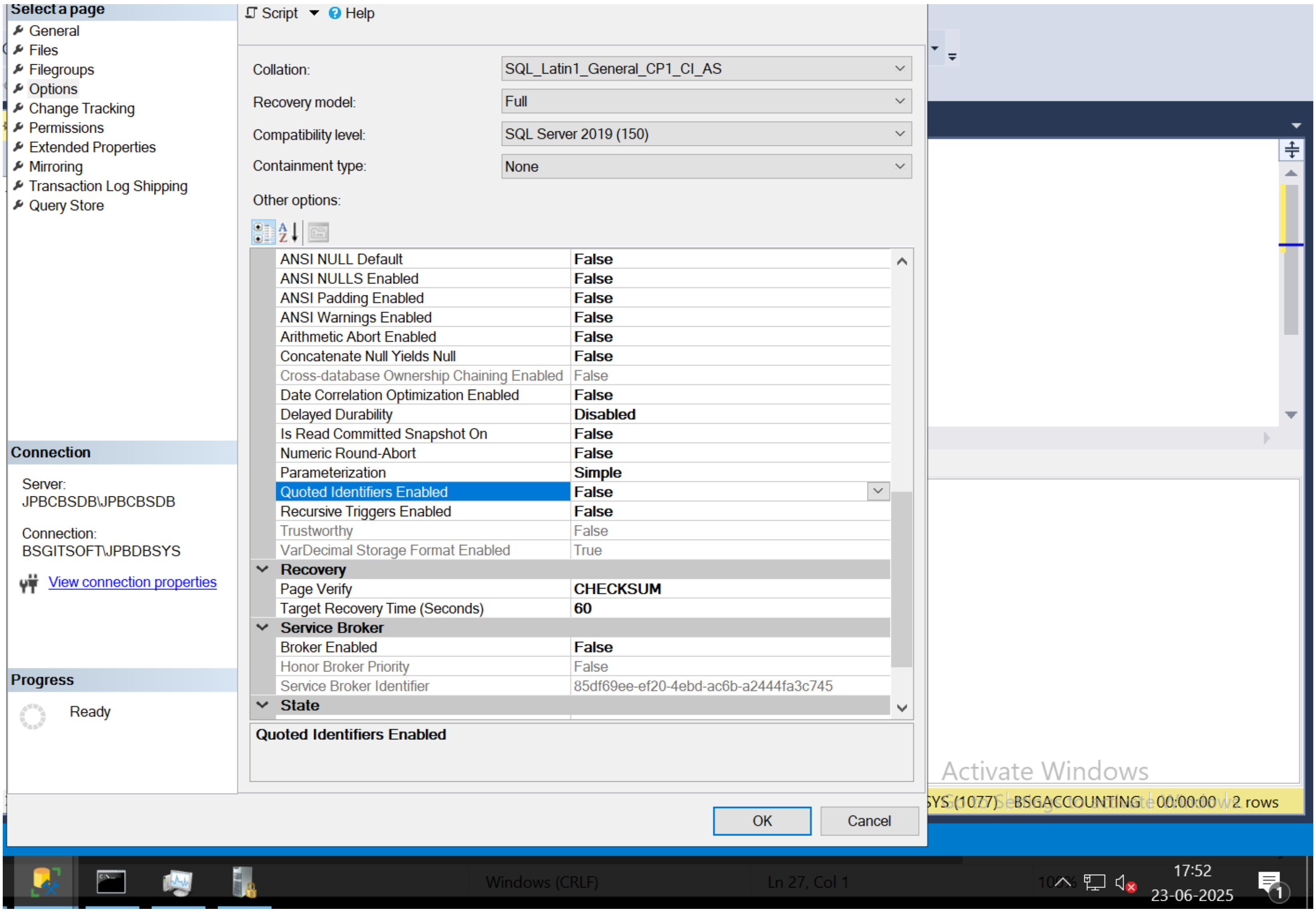
Name	Description	Table Type	Create Date
account_change_status	HEAP	USER_TABLE	22-12-2022 22:56
account_charges_legacy	HEAP	USER_TABLE	16-12-2020 23:40
account_last_txn	HEAP	USER_TABLE	16-12-2020 23:40
account_txn_history	HEAP	USER_TABLE	16-12-2020 23:40
activity_end_point	HEAP	USER_TABLE	16-12-2020 23:40
auto_transfer_products_mapping	HEAP	USER_TABLE	16-12-2020 23:40
bal_table	HEAP	USER_TABLE	16-12-2020 23:40

BULK_SMS_DETAILS	HEAP	USER_TABLE	16-12-2020 23:40
casa_interest_details	HEAP	USER_TABLE	16-12-2020 23:40
cash_token_data	HEAP	USER_TABLE	16-12-2020 23:40
cc_credit_interest_details	HEAP	USER_TABLE	07-02-2023 20:31
channel_request_Old_15052025	HEAP	USER_TABLE	16-12-2020 23:40
channel_request_transfer_15112024	HEAP	USER_TABLE	16-06-2023 11:14
charge_collection_history	HEAP	USER_TABLE	22-02-2022 18:14
charge_detail	HEAP	USER_TABLE	22-02-2022 18:14
charges_activity_master	HEAP	USER_TABLE	16-12-2020 23:40
crr	HEAP	USER_TABLE	16-12-2020 23:40
deaf_interest_slab	HEAP	USER_TABLE	16-12-2020 23:40
debitcard_charges_on_accounts	HEAP	USER_TABLE	16-12-2020 23:40
denomination_master	HEAP	USER_TABLE	16-12-2020 23:40
dispute_charges	HEAP	USER_TABLE	16-12-2020 23:40
ekyc_customer_creation	HEAP	USER_TABLE	14-01-2023 15:08
EPMHSG_Interest_Receivable	HEAP	USER_TABLE	16-12-2020 23:40
folio_charges_on_accounts	HEAP	USER_TABLE	16-12-2020 23:40
inoperative_charges_on_accounts	HEAP	USER_TABLE	16-12-2020 23:40
interest_accrual_pnr_pl_posting_1512022_loan_int	HEAP	USER_TABLE	15-12-2022 02:46
invalid_file_upload_txn	HEAP	USER_TABLE	16-12-2020 23:40
loan_actual_interest_ideal_interest_diff	HEAP	USER_TABLE	01-12-2022 12:43
loan_advance_payment_status	HEAP	USER_TABLE	04-08-2022 15:31
loan_bddr_txn_master	HEAP	USER_TABLE	16-12-2020 23:40
loan_holiday_master	HEAP	USER_TABLE	25-06-2022 11:48
loan_interest_details	HEAP	USER_TABLE	16-12-2020 23:40
loan_interest_reversal_details	HEAP	USER_TABLE	16-12-2020 23:40
loan_repayment_chart	HEAP	USER_TABLE	12-01-2023 15:07
loan_reschedule_master	HEAP	USER_TABLE	04-08-2022 11:18
loan_reschedule_product_details	HEAP	USER_TABLE	04-08-2022 11:18
loan_rescheduling_summary	HEAP	USER_TABLE	04-08-2022 11:18
maintenance_charges_on_accounts	HEAP	USER_TABLE	16-12-2020 23:40
min_bal_charges_on_accounts	HEAP	USER_TABLE	16-12-2020 23:40
partner_bank_sweep_out_txn_details_29052025	HEAP	USER_TABLE	29-05-2025 01:14
partner_sweepout_request_response_details_29052025	HEAP	USER_TABLE	29-05-2025 01:15
principle_turing	HEAP	USER_TABLE	16-12-2020 23:40
principle_turing_data	HEAP	USER_TABLE	16-12-2020 23:40
principle_turing_temp	HEAP	USER_TABLE	16-12-2020 23:40
Query	HEAP	USER_TABLE	16-12-2020 23:40
sweep_out_scanning_table	HEAP	USER_TABLE	01-03-2024 03:23
sweep_out_scanning_table_29052025	HEAP	USER_TABLE	29-05-2025 01:15
sweep_out_task_Details	HEAP	USER_TABLE	03-06-2023 19:43
sweepin_transaction_request	HEAP	USER_TABLE	01-03-2024 03:23
td_financial_year_bifurcation	HEAP	USER_TABLE	16-12-2020 23:40
td_interest_application	HEAP	USER_TABLE	16-12-2020 23:40
teller_exchange_transaction	HEAP	USER_TABLE	16-12-2020 23:40
transaction_code_balance_type	HEAP	USER_TABLE	16-12-2020 23:40
transaction_master_internal	HEAP	USER_TABLE	02-11-2023 00:26
transaction_monitor_rule	HEAP	USER_TABLE	16-12-2020 23:40

transaction_rule_account	HEAP	USER_TABLE	16-12-2020 23:40

9. Query Optimizer fixes

Overview: When applying any hot fixes, cumulative update or service pack, SQL server optimizer would utilize the changes immediately, in case you enabled the query optimizer fixes using trace flag 4199. The new patches can result in performance improvement.



Server:
JPBCBSDB\JPBCBSDB

Connection:
BSGITSOFT\JPBDBSYS

Collation: SQL_Latin1_General_CP1_CI_AS

Recovery model: Full

Compatibility level: SQL Server 2019 (150)

Containment type: None

Other options:

ANSI NULL Default	False
ANSI NULLS Enabled	False
ANSI Padding Enabled	False
ANSI Warnings Enabled	False
Arithmetic Abort Enabled	False
Concatenate Null Yields Null	False
Cross-database Ownership Chaining Enabled	False
Date Correlation Optimization Enabled	False
Delayed Durability	Disabled
Is Read Committed Snapshot On	False
Numeric Round-Abort	False
Parameterization	Simple
Quoted Identifiers Enabled	False
Recursive Triggers Enabled	False
Trustworthy	False
VarDecimal Storage Format Enabled	True
Recovery	
Page Verify	CHECKSUM
Target Recovery Time (Seconds)	60
Service Broker	
Broker Enabled	False
Honor Broker Priority	False
Service Broker Identifier	85df69ee-ef20-4ebd-ac6b-a2444fa3c745
State	

Quoted Identifiers Enabled

SYS (1077) BSGACCOUNTING 00:00:00 2 rows

Recommendation: Recommend Enabling Quoted Identifiers Enabled

10. Snapshot Isolation

The term "snapshot" reflects the fact that all queries in the transaction see the same version, or snapshot, of the database, based on the state of the database at the moment in time when the transaction begins. No locks are acquired on the underlying data rows or data pages in a snapshot transaction, which permits other transactions to execute without being blocked by a prior uncompleted transaction. Transactions that modify data do not block transactions that read data, and transactions that read data do not block transactions that write data, as they normally would under the default READ COMMITTED isolation level in SQL Server. This non-blocking behaviour also significantly reduces the likelihood of deadlocks for complex transactions.

Database name	Snapshot Isolation	Recommended Value
BSGACCOUNTING	OFF	ON

11. Log File Growth Limit

Currently there are no restrictions on log file growth, hence log file growth grows till 2TB. As per our recommendation log file settings must be reconfigured.

Database name	Recovery mode	Size MB	Max size(MB)	Recommended Value GB
BSGACCOUNTING	FULL	65243	Unlimited	500

Note: Please make sure Transactional Log backup is added in database backup policy.

12. Enable schema binding on Scalar User defined functions:

SCHEMA BINDING is commonly used with SQL Server objects like views and User Defined Functions (UDF). The main benefit of SCHEMA BINDING is to avoid any accidental drop or change of an object that is referenced by other objects. A User Defined Function (UDF) may or may not access any underlying database objects, but in this using SCHEMA BINDING with a UDF can improve performance even if there are no underlying objects.

DB Name: BSGACCOUNTING

FunctionName	IsDeterministic	IsPrecise	IsSystemVerified	UserDataAccess	SystemDataAccess
[dbo].[fn_dayOfMonths_upi_request]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[fn_dayOfMonths_transactionMasterInternal]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[getConcessionMonth]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[getLastInstallmentDueDate]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[fn_dayOfMonths_transactionStatus]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[fn_dayOfMonths_transactionScroll]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[fn_calcInterestForLoanAgainstDeposit]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[fn_get_quarter_no]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[FullMonthsSeparation]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[IsLeapYear]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[fn_diagramobjects]	FALSE	FALSE	FALSE	TRUE	TRUE
[dbo].[fn_dayOfMonths_partitionTable]	FALSE	FALSE	FALSE	TRUE	TRUE

13. Configure Arithabort settings from application:

By default, in .Net it is OFF which leads to a lot of performance degradation. Use of arithabort ends a query when an overflow or divide-by-zero error occurs during query execution. We strongly recommend configuring connection string to use SET ARITHABOT ON

Current Status: Disabled

Recommended status: Enabled

14. Disk Performance:

Below table shows the disk performance. The Average Total Latency column represents the total latency about the database files, and we can use the following table as reference to evaluate the disk performance against latency.

Database Name	physical_name	avg_read_stall_ms	avg_write_stall_ms	avg_io_stall_ms

BSGACCOUNTING	G:\SQLDBLOGS\JPB\Logs\BSGACCOUNTING_1.ldf	3.7	2.8	2.8
BSGACCOUNTING	N:\Trnmasterint\trans_master_inter25.ndf	5.1	4.4	4.7
BSGACCOUNTING	N:\Trnmasterint\trans_master_inter27.ndf	5.7	4.6	5
BSGACCOUNTING	N:\Trnmasterint\trans_master_inter29.ndf	14.3	4.4	8.6
BSGACCOUNTING	N:\Trnmasterint\trans_master_inter31.ndf	15.3	4.4	9.1
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq1.ndf	4.4	4.7	4.6
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq11.ndf	6	4.4	4.6
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq13.ndf	9.6	4.6	5.6
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq15.ndf	5.5	4.7	4.8
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq17.ndf	5.7	4.6	4.7
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq19.ndf	4.5	4.4	4.4
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq21.ndf	4.3	4.4	4.4
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq23.ndf	4.5	4.3	4.3
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq25.ndf	4.2	4.3	4.3
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq27.ndf	4.2	4.5	4.4
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq29.ndf	3.8	4.3	4.3
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq3.ndf	4.7	4.8	4.8
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq31.ndf	3.8	4.9	4.8
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq5.ndf	4.1	4.4	4.4
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq7.ndf	4.2	4.5	4.5
BSGACCOUNTING	N:\UPImerchant\upimerchant_transreq9.ndf	4.3	4.4	4.4
BSGACCOUNTING	O:\Trnmasterint\trans_master_inter26.ndf	5.4	4.6	4.9
BSGACCOUNTING	O:\Trnmasterint\trans_master_inter28.ndf	4.7	4.4	4.5
BSGACCOUNTING	O:\Trnmasterint\trans_master_inter30.ndf	12.2	4.6	7.6
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq10.ndf	5.6	4.5	4.7
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq12.ndf	4.7	4.4	4.5

BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq14.ndf	7.4	4.7	5
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq16.ndf	4.5	5	4.9
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq18.ndf	9.3	4.6	5.6
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq2.ndf	4.6	4.5	4.5
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq20.ndf	5.3	4.4	4.6
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq22.ndf	4.6	4.4	4.4
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq24.ndf	4.4	4.5	4.5
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq26.ndf	4.8	4.4	4.5
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq28.ndf	4.3	4.4	4.4
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq30.ndf	5.3	4.5	4.5
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq4.ndf	4.7	4.5	4.5
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq6.ndf	4.3	4.7	4.6
BSGACCOUNTING	O:\UPImerchant\upimerchant_transreq8.ndf	4.9	4.6	4.6
BSGACCOUNTING	Q:\balance_queue\balance_queue1.ndf	5.8	5.1	5.3
BSGACCOUNTING	Q:\balance_queue\balance_queue11.ndf	3.8	4.9	4.6
BSGACCOUNTING	Q:\balance_queue\balance_queue13.ndf	4.4	4.8	4.7
BSGACCOUNTING	Q:\balance_queue\balance_queue15.ndf	4.2	4.7	4.6
BSGACCOUNTING	Q:\balance_queue\balance_queue17.ndf	3.3	4.5	4.3
BSGACCOUNTING	Q:\balance_queue\balance_queue19.ndf	3.2	5.2	4.6
BSGACCOUNTING	Q:\balance_queue\balance_queue21.ndf	4.6	5.4	5.2
BSGACCOUNTING	Q:\balance_queue\balance_queue23.ndf	3.3	5	4.5
BSGACCOUNTING	Q:\balance_queue\balance_queue25.ndf	3.9	4.9	4.6
BSGACCOUNTING	Q:\balance_queue\balance_queue27.ndf	3.5	4.9	4.6
BSGACCOUNTING	Q:\balance_queue\balance_queue29.ndf	4.8	4.9	4.8
BSGACCOUNTING	Q:\balance_queue\balance_queue3.ndf	3.7	5.3	4.9
BSGACCOUNTING	Q:\balance_queue\balance_queue31.ndf	4.7	4.8	4.7

BSGACCOUNTING	Q:\balance_queue\balance_queue5.ndf	3.3	4.9	4.6
BSGACCOUNTING	Q:\balance_queue\balance_queue7.ndf	3.4	5	4.7
BSGACCOUNTING	Q:\balance_queue\balance_queue9.ndf	3.7	4.8	4.5
BSGACCOUNTING	Q:\channel_request\channel_request1.ndf	4.5	4.8	4.8
BSGACCOUNTING	Q:\channel_request\channel_request11.ndf	4.9	4.4	4.5
BSGACCOUNTING	Q:\channel_request\channel_request13.ndf	4	4.6	4.5
BSGACCOUNTING	Q:\channel_request\channel_request15.ndf	4.6	4.8	4.8
BSGACCOUNTING	Q:\channel_request\channel_request17.ndf	4.8	4.6	4.6
BSGACCOUNTING	Q:\channel_request\channel_request19.ndf	4	5.1	4.9
BSGACCOUNTING	Q:\channel_request\channel_request21.ndf	5.3	5.3	5.3
BSGACCOUNTING	Q:\channel_request\channel_request23.ndf	4	4.6	4.6
BSGACCOUNTING	Q:\channel_request\channel_request25.ndf	5.2	5.2	5.2
BSGACCOUNTING	Q:\channel_request\channel_request27.ndf	4.4	4.8	4.8
BSGACCOUNTING	Q:\channel_request\channel_request29.ndf	5.1	5	5
BSGACCOUNTING	Q:\channel_request\channel_request3.ndf	4.6	5.2	5.2
BSGACCOUNTING	Q:\channel_request\channel_request31.ndf	4.8	4.7	4.7
BSGACCOUNTING	Q:\channel_request\channel_request5.ndf	4.8	4.7	4.7
BSGACCOUNTING	Q:\channel_request\channel_request7.ndf	5	4.6	4.7
BSGACCOUNTING	Q:\channel_request\channel_request9.ndf	4.1	4.2	4.2
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer1.ndf	6.2	5.1	5.4
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer11.ndf	4.9	4.8	4.9
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer13.ndf	5.6	4.8	4.9
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer15.ndf	4.8	4.6	4.7
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer17.ndf	5.8	4.5	4.6
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer19.ndf	4.5	5	4.9
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer21.ndf	5.5	5.3	5.4

BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer23.ndf	4.3	5	4.9
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer25.ndf	5	4.8	4.8
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer27.ndf	4.3	4.8	4.7
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer29.ndf	4.6	4.7	4.7
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer3.ndf	4.5	5.2	5.1
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer31.ndf	5.6	4.8	5
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer5.ndf	4.8	4.8	4.8
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer7.ndf	4.8	4.9	4.9
BSGACCOUNTING	Q:\channel_request_transfer\channel_request_transfer9.ndf	4.5	4.7	4.7
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs1.ndf	3	5.2	5.1
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs11.ndf	2.5	4.9	4.7
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs13.ndf	3.1	4.9	4.8
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs15.ndf	2.9	4.4	4.4
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs17.ndf	3	4.8	4.7
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs19.ndf	3.1	5.3	5.1
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs21.ndf	2.7	4.9	4.7
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs23.ndf	2.8	4.9	4.7
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs25.ndf	3.1	5.4	5.3
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs27.ndf	3.6	4.6	4.5
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs29.ndf	2.8	4.8	4.6
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs3.ndf	3.2	5.5	5.4
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs31.ndf	3.2	4.6	4.4
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs5.ndf	3.1	5.1	5
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs7.ndf	3.3	5.6	5.4
BSGACCOUNTING	Q:\neft_rtgs\neft_rtgs9.ndf	2.9	5.2	5
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction1.ndf	5.8	5.7	5.8

BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction11.ndf	4.3	4.9	4.6
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction13.ndf	4.5	4.9	4.7
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction15.ndf	3.3	4.7	4.3
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction17.ndf	3.3	4.7	4.3
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction19.ndf	4.1	6.1	5.4
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction21.ndf	4.3	5.6	5.1
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction23.ndf	3.8	5.1	4.6
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction25.ndf	3.7	5.2	4.6
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction27.ndf	4.1	5.6	4.9
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction29.ndf	4.3	5.5	5
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction3.ndf	5.3	6.3	5.8
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction31.ndf	4.8	5.8	5.3
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction5.ndf	7.5	4.7	6
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction7.ndf	4.1	4.7	4.4
BSGACCOUNTING	Q:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction9.ndf	3.8	5.5	4.9
BSGACCOUNTING	Q:\transaction_master\transaction_master1.ndf	5	5.5	5.5
BSGACCOUNTING	Q:\transaction_master\transaction_master11.ndf	4.7	5.3	5.3
BSGACCOUNTING	Q:\transaction_master\transaction_master13.ndf	6.3	5.2	5.3
BSGACCOUNTING	Q:\transaction_master\transaction_master15.ndf	6.3	5.1	5.1
BSGACCOUNTING	Q:\transaction_master\transaction_master17.ndf	4.9	4.8	4.8
BSGACCOUNTING	Q:\transaction_master\transaction_master19.ndf	5	5.6	5.6
BSGACCOUNTING	Q:\transaction_master\transaction_master21.ndf	5.4	5.6	5.6
BSGACCOUNTING	Q:\transaction_master\transaction_master23.ndf	4.6	5.2	5.2
BSGACCOUNTING	Q:\transaction_master\transaction_master25.ndf	4.6	5	5
BSGACCOUNTING	Q:\transaction_master\transaction_master27.ndf	4.9	5	5
BSGACCOUNTING	Q:\transaction_master\transaction_master29.ndf	4	4.9	4.8

BSGACCOUNTING	Q:\transaction_master\transaction_master3.ndf	5.1	5.6	5.6
BSGACCOUNTING	Q:\transaction_master\transaction_master31.ndf	4	5	4.9
BSGACCOUNTING	Q:\transaction_master\transaction_master5.ndf	4	5.1	5.1
BSGACCOUNTING	Q:\transaction_master\transaction_master7.ndf	6.7	5.3	5.3
BSGACCOUNTING	Q:\transaction_master\transaction_master9.ndf	6.5	5.2	5.3
BSGACCOUNTING	Q:\transaction_pending\transaction_pending1.ndf	6.8	5.6	5.8
BSGACCOUNTING	Q:\transaction_pending\transaction_pending11.ndf	5.5	5.3	5.4
BSGACCOUNTING	Q:\transaction_pending\transaction_pending13.ndf	6.9	5.3	5.5
BSGACCOUNTING	Q:\transaction_pending\transaction_pending15.ndf	6.2	5.1	5.3
BSGACCOUNTING	Q:\transaction_pending\transaction_pending17.ndf	8.5	5	5.4
BSGACCOUNTING	Q:\transaction_pending\transaction_pending19.ndf	7.9	5.7	6
BSGACCOUNTING	Q:\transaction_pending\transaction_pending21.ndf	7.1	5.8	6
BSGACCOUNTING	Q:\transaction_pending\transaction_pending23.ndf	7.8	5.4	5.8
BSGACCOUNTING	Q:\transaction_pending\transaction_pending25.ndf	5.4	5.2	5.2
BSGACCOUNTING	Q:\transaction_pending\transaction_pending27.ndf	6.3	5.3	5.4
BSGACCOUNTING	Q:\transaction_pending\transaction_pending29.ndf	6.9	5.1	5.4
BSGACCOUNTING	Q:\transaction_pending\transaction_pending3.ndf	7.7	5.8	6.1
BSGACCOUNTING	Q:\transaction_pending\transaction_pending31.ndf	5.4	5.1	5.2
BSGACCOUNTING	Q:\transaction_pending\transaction_pending5.ndf	6	5.3	5.4
BSGACCOUNTING	Q:\transaction_pending\transaction_pending7.ndf	6.8	5.5	5.6
BSGACCOUNTING	Q:\transaction_pending\transaction_pending9.ndf	8.7	5.3	5.8
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno1.ndf	6.3	5.5	5.7
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno11.ndf	4.6	5.2	5.1
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno13.ndf	4.3	5.2	5
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno15.ndf	4.5	5.2	5
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno17.ndf	5	5	5

BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno19.ndf	4.2	5.7	5.3
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno21.ndf	4.8	5.8	5.6
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno23.ndf	4.4	5.5	5.2
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno25.ndf	4.6	5.3	5.1
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno27.ndf	4	5.3	5
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno29.ndf	4.5	5.2	5
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno3.ndf	4.6	5.7	5.4
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno31.ndf	5.4	5.2	5.3
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno5.ndf	4.1	5.4	5.1
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno7.ndf	4.5	5.4	5.2
BSGACCOUNTING	Q:\transaction_scrollno\transaction_scrollno9.ndf	5	5.2	5.1
BSGACCOUNTING	Q:\transaction_status\transaction_status1.ndf	4.6	5.1	5
BSGACCOUNTING	Q:\transaction_status\transaction_status11.ndf	6	5.2	5.4
BSGACCOUNTING	Q:\transaction_status\transaction_status13.ndf	5.1	5	5.1
BSGACCOUNTING	Q:\transaction_status\transaction_status15.ndf	4.6	4.8	4.8
BSGACCOUNTING	Q:\transaction_status\transaction_status17.ndf	5.9	4.6	4.9
BSGACCOUNTING	Q:\transaction_status\transaction_status19.ndf	5.6	5.3	5.4
BSGACCOUNTING	Q:\transaction_status\transaction_status21.ndf	5.4	5.9	5.7
BSGACCOUNTING	Q:\transaction_status\transaction_status23.ndf	5.5	5.1	5.2
BSGACCOUNTING	Q:\transaction_status\transaction_status25.ndf	4.5	5	4.8
BSGACCOUNTING	Q:\transaction_status\transaction_status27.ndf	5.5	5	5.1
BSGACCOUNTING	Q:\transaction_status\transaction_status29.ndf	5.1	5	5
BSGACCOUNTING	Q:\transaction_status\transaction_status3.ndf	5.3	5.3	5.3
BSGACCOUNTING	Q:\transaction_status\transaction_status31.ndf	5.8	5.1	5.3
BSGACCOUNTING	Q:\transaction_status\transaction_status5.ndf	5.4	5	5.1
BSGACCOUNTING	Q:\transaction_status\transaction_status7.ndf	5.7	5	5.2

BSGACCOUNTING	Q:\transaction_status\transaction_status9.ndf	5.9	5	5.3
BSGACCOUNTING	Q:\upi_request\upi_request1.ndf	9.1	5.5	5.8
BSGACCOUNTING	Q:\upi_request\upi_request11.ndf	5.2	5.1	5.1
BSGACCOUNTING	Q:\upi_request\upi_request13.ndf	7.5	5	5.3
BSGACCOUNTING	Q:\upi_request\upi_request15.ndf	7.6	4.9	5.1
BSGACCOUNTING	Q:\upi_request\upi_request17.ndf	7	4.7	4.9
BSGACCOUNTING	Q:\upi_request\upi_request19.ndf	4.9	5.5	5.4
BSGACCOUNTING	Q:\upi_request\upi_request21.ndf	5.9	5.5	5.6
BSGACCOUNTING	Q:\upi_request\upi_request23.ndf	5.3	5.1	5.1
BSGACCOUNTING	Q:\upi_request\upi_request25.ndf	5.6	5.2	5.3
BSGACCOUNTING	Q:\upi_request\upi_request27.ndf	4.3	5.2	5.1
BSGACCOUNTING	Q:\upi_request\upi_request29.ndf	4.7	4.9	4.9
BSGACCOUNTING	Q:\upi_request\upi_request3.ndf	9.3	5.6	6
BSGACCOUNTING	Q:\upi_request\upi_request31.ndf	11.7	5	5.6
BSGACCOUNTING	Q:\upi_request\upi_request5.ndf	4.5	5.2	5.1
BSGACCOUNTING	Q:\upi_request\upi_request7.ndf	5.3	5.3	5.3
BSGACCOUNTING	Q:\upi_request\upi_request9.ndf	5.5	5.1	5.1
BSGACCOUNTING	R:\balance_queue\balance_queue10.ndf	4	4.8	4.6
BSGACCOUNTING	R:\balance_queue\balance_queue12.ndf	3	4.8	4.3
BSGACCOUNTING	R:\balance_queue\balance_queue14.ndf	3.5	4.9	4.5
BSGACCOUNTING	R:\balance_queue\balance_queue16.ndf	4	5	4.7
BSGACCOUNTING	R:\balance_queue\balance_queue18.ndf	3.1	5	4.4
BSGACCOUNTING	R:\balance_queue\balance_queue2.ndf	4	5.1	4.8
BSGACCOUNTING	R:\balance_queue\balance_queue20.ndf	3.2	5.1	4.6
BSGACCOUNTING	R:\balance_queue\balance_queue22.ndf	3	4.9	4.4
BSGACCOUNTING	R:\balance_queue\balance_queue24.ndf	3.6	5	4.6

BSGACCOUNTING	R:\balance_queue\balance_queue26.ndf	3.6	7.4	6.4
BSGACCOUNTING	R:\balance_queue\balance_queue28.ndf	4.9	4.9	4.9
BSGACCOUNTING	R:\balance_queue\balance_queue30.ndf	3.4	4.8	4.4
BSGACCOUNTING	R:\balance_queue\balance_queue4.ndf	3.5	4.9	4.5
BSGACCOUNTING	R:\balance_queue\balance_queue6.ndf	3	4.9	4.5
BSGACCOUNTING	R:\balance_queue\balance_queue8.ndf	3.8	4.9	4.7
BSGACCOUNTING	R:\channel_request\channel_request10.ndf	5.1	4.4	4.5
BSGACCOUNTING	R:\channel_request\channel_request12.ndf	5.2	4.3	4.4
BSGACCOUNTING	R:\channel_request\channel_request14.ndf	5.1	4.6	4.6
BSGACCOUNTING	R:\channel_request\channel_request16.ndf	4.8	4.7	4.7
BSGACCOUNTING	R:\channel_request\channel_request18.ndf	5.1	5.1	5.1
BSGACCOUNTING	R:\channel_request\channel_request2.ndf	5	5	5
BSGACCOUNTING	R:\channel_request\channel_request20.ndf	5	6.6	6.4
BSGACCOUNTING	R:\channel_request\channel_request22.ndf	4.3	4.9	4.9
BSGACCOUNTING	R:\channel_request\channel_request24.ndf	5.2	5.3	5.3
BSGACCOUNTING	R:\channel_request\channel_request26.ndf	5.1	4.8	4.8
BSGACCOUNTING	R:\channel_request\channel_request28.ndf	5.4	4.7	4.8
BSGACCOUNTING	R:\channel_request\channel_request30.ndf	5.5	4.7	4.8
BSGACCOUNTING	R:\channel_request\channel_request4.ndf	4.8	4.8	4.8
BSGACCOUNTING	R:\channel_request\channel_request6.ndf	4.8	4.6	4.6
BSGACCOUNTING	R:\channel_request\channel_request8.ndf	5.4	4.6	4.7
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer10.ndf	5.3	4.8	4.9
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer12.ndf	4.9	4.7	4.8
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer14.ndf	5.1	4.8	4.8
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer16.ndf	4.3	4.9	4.8
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer18.ndf	6.3	4.8	5.1

BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer2.ndf	5.8	5.1	5.3
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer20.ndf	5.8	5	5.2
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer22.ndf	4.9	4.8	4.8
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer24.ndf	4.8	4.8	4.8
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer26.ndf	4.6	4.8	4.8
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer28.ndf	4.5	4.8	4.7
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer30.ndf	5.4	4.8	5
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer4.ndf	4.1	4.9	4.7
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer6.ndf	4.5	4.8	4.7
BSGACCOUNTING	R:\channel_request_transfer\channel_request_transfer8.ndf	4.8	5	5
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs10.ndf	3.2	5.2	5
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs12.ndf	3	5	4.8
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs14.ndf	3.7	4.8	4.7
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs16.ndf	3.6	5.4	5.4
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs18.ndf	4.1	5.3	5.2
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs2.ndf	3.8	5.2	5.1
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs20.ndf	3.6	5.2	5.1
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs22.ndf	3.8	5.1	5
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs24.ndf	3.4	5	4.9
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs26.ndf	3.3	4.8	4.7
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs28.ndf	3.7	4.7	4.7
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs30.ndf	3.5	5	4.9
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs4.ndf	4.1	4.7	4.7
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs6.ndf	3.4	5.3	5.2
BSGACCOUNTING	R:\neft_rtgs\neft_rtgs8.ndf	3.6	4.8	4.7
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction10.ndf	5.4	4.9	5.2

BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction12.ndf	3.9	4.7	4.4
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction14.ndf	4.7	5.4	5
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction16.ndf	3.5	4.7	4.5
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction18.ndf	3.4	5.6	4.8
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction2.ndf	5	4.6	4.8
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction20.ndf	4	5.7	5
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction22.ndf	3.9	5	4.6
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction24.ndf	4.1	5.2	4.7
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction26.ndf	5.3	4.6	4.9
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction28.ndf	4.5	4.8	4.7
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction30.ndf	6.3	4.8	5.6
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction4.ndf	4	5	4.5
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction6.ndf	4.4	4.9	4.7
BSGACCOUNTING	R:\neft_rtgs_inward_transaction\neft_rtgs_inward_transaction8.ndf	3.2	4.9	4.2
BSGACCOUNTING	R:\transaction_master\transaction_master10.ndf	5.1	5.1	5.1
BSGACCOUNTING	R:\transaction_master\transaction_master12.ndf	4	5	4.9
BSGACCOUNTING	R:\transaction_master\transaction_master14.ndf	4.7	5.2	5.2
BSGACCOUNTING	R:\transaction_master\transaction_master16.ndf	4	5.6	5.6
BSGACCOUNTING	R:\transaction_master\transaction_master18.ndf	3.9	5.3	5.3
BSGACCOUNTING	R:\transaction_master\transaction_master2.ndf	4.3	5.4	5.3
BSGACCOUNTING	R:\transaction_master\transaction_master20.ndf	4.5	5.4	5.4
BSGACCOUNTING	R:\transaction_master\transaction_master22.ndf	4.6	5.1	5
BSGACCOUNTING	R:\transaction_master\transaction_master24.ndf	7	5	5.1
BSGACCOUNTING	R:\transaction_master\transaction_master26.ndf	4	5.1	5
BSGACCOUNTING	R:\transaction_master\transaction_master28.ndf	3.8	5	4.9
BSGACCOUNTING	R:\transaction_master\transaction_master30.ndf	3.2	4.8	4.7

BSGACCOUNTING	R:\transaction_master\transaction_master4.ndf	3.6	5.1	5
BSGACCOUNTING	R:\transaction_master\transaction_master6.ndf	3.6	5.2	5.1
BSGACCOUNTING	R:\transaction_master\transaction_master8.ndf	5.4	5.1	5.1
BSGACCOUNTING	R:\transaction_pending\transaction_pending10.ndf	8.9	5.3	5.9
BSGACCOUNTING	R:\transaction_pending\transaction_pending12.ndf	4.4	5.2	5
BSGACCOUNTING	R:\transaction_pending\transaction_pending14.ndf	9.5	5.3	5.9
BSGACCOUNTING	R:\transaction_pending\transaction_pending16.ndf	5.3	5.6	5.6
BSGACCOUNTING	R:\transaction_pending\transaction_pending18.ndf	5.9	5.4	5.5
BSGACCOUNTING	R:\transaction_pending\transaction_pending2.ndf	9.6	5.6	6.2
BSGACCOUNTING	R:\transaction_pending\transaction_pending20.ndf	8.1	5.5	5.9
BSGACCOUNTING	R:\transaction_pending\transaction_pending22.ndf	6.4	5.2	5.4
BSGACCOUNTING	R:\transaction_pending\transaction_pending24.ndf	7.6	5.2	5.6
BSGACCOUNTING	R:\transaction_pending\transaction_pending26.ndf	7.2	5.3	5.6
BSGACCOUNTING	R:\transaction_pending\transaction_pending28.ndf	8.5	5.1	5.6
BSGACCOUNTING	R:\transaction_pending\transaction_pending30.ndf	5.6	5	5.1
BSGACCOUNTING	R:\transaction_pending\transaction_pending4.ndf	6.5	5.3	5.5
BSGACCOUNTING	R:\transaction_pending\transaction_pending6.ndf	5.9	5.4	5.5
BSGACCOUNTING	R:\transaction_pending\transaction_pending8.ndf	6.9	5.3	5.6
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno10.ndf	4.5	5.2	5
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno12.ndf	4.2	5.2	4.9
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno14.ndf	4.9	5.2	5.1
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno16.ndf	4.1	5.4	5
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno18.ndf	4.7	5.4	5.2
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno2.ndf	5.2	5.5	5.4
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno20.ndf	3.7	5.6	5.2
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno22.ndf	4.2	5.3	5

BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno24.ndf	4.7	5.3	5.1
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno26.ndf	4.5	5.4	5.2
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno28.ndf	4.6	5.3	5.1
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno30.ndf	4.7	5.2	5.1
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno4.ndf	3.8	5.3	4.8
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno6.ndf	3.7	5.3	4.9
BSGACCOUNTING	R:\transaction_scrollno\transaction_scrollno8.ndf	4.4	5.4	5.1
BSGACCOUNTING	R:\transaction_status\transaction_status10.ndf	5.5	5.5	5.5
BSGACCOUNTING	R:\transaction_status\transaction_status12.ndf	4.5	5.1	4.9
BSGACCOUNTING	R:\transaction_status\transaction_status14.ndf	5.8	5	5.2
BSGACCOUNTING	R:\transaction_status\transaction_status16.ndf	5.2	5.2	5.2
BSGACCOUNTING	R:\transaction_status\transaction_status18.ndf	4.4	4.9	4.8
BSGACCOUNTING	R:\transaction_status\transaction_status2.ndf	4.7	5.2	5
BSGACCOUNTING	R:\transaction_status\transaction_status20.ndf	4.9	5.1	5.1
BSGACCOUNTING	R:\transaction_status\transaction_status22.ndf	5.6	5	5.1
BSGACCOUNTING	R:\transaction_status\transaction_status24.ndf	5.5	4.9	5.1
BSGACCOUNTING	R:\transaction_status\transaction_status26.ndf	5.5	5	5.2
BSGACCOUNTING	R:\transaction_status\transaction_status28.ndf	4.5	5	4.9
BSGACCOUNTING	R:\transaction_status\transaction_status30.ndf	5.2	4.9	5
BSGACCOUNTING	R:\transaction_status\transaction_status4.ndf	4.9	5	4.9
BSGACCOUNTING	R:\transaction_status\transaction_status6.ndf	4.5	5	4.8
BSGACCOUNTING	R:\transaction_status\transaction_status8.ndf	5.4	5.1	5.2
BSGACCOUNTING	R:\upi_request\upi_request10.ndf	6	5.1	5.2
BSGACCOUNTING	R:\upi_request\upi_request12.ndf	8.2	5.1	5.4
BSGACCOUNTING	R:\upi_request\upi_request14.ndf	8.3	5	5.3
BSGACCOUNTING	R:\upi_request\upi_request16.ndf	6.3	5	5.1

BSGACCOUNTING	R:\upi_request\upi_request18.ndf	6	5.3	5.4
BSGACCOUNTING	R:\upi_request\upi_request2.ndf	7.6	5.4	5.6
BSGACCOUNTING	R:\upi_request\upi_request20.ndf	5.6	5.5	5.5
BSGACCOUNTING	R:\upi_request\upi_request22.ndf	5.7	5.1	5.1
BSGACCOUNTING	R:\upi_request\upi_request24.ndf	6.1	5	5.1
BSGACCOUNTING	R:\upi_request\upi_request26.ndf	5.7	5.1	5.1
BSGACCOUNTING	R:\upi_request\upi_request28.ndf	5	5	5
BSGACCOUNTING	R:\upi_request\upi_request30.ndf	4.4	4.9	4.8
BSGACCOUNTING	R:\upi_request\upi_request4.ndf	6	5.1	5.2
BSGACCOUNTING	R:\upi_request\upi_request6.ndf	4.4	5.5	5.4
BSGACCOUNTING	R:\upi_request\upi_request8.ndf	5.2	5.2	5.2

15. Duplicate indexes:

Duplicate indexes refer to the situation where you have multiple indexes defined on a table that have the same key columns. These duplicate indexes can lead to increased storage requirements, longer data modification times, and potentially slower query performance. It's generally considered best practice to eliminate duplicate indexes to optimize database performance.

DB Name: BSGACCOUNTING

Table name	Index name	Column list	Index name	Column list
account_balance_internal	idx_account_id	account_id	account_id_txn_date	account_id, txn_date
account_balance_internal	idx_account_id_txn_date	account_id, txn_date	account_id_txn_date	account_id, txn_date
account_balance_internal	UNIQUE_Table	account_id, txn_date	account_id_txn_date	account_id, txn_date
account_balance_internal	account_id_txn_date	account_id, txn_date	idx_account_id_txn_date	account_id, txn_date
account_balance_internal	idx_account_id	account_id	idx_account_id_txn_date	account_id, txn_date
account_balance_internal	UNIQUE_Table	account_id, txn_date	idx_account_id_txn_date	account_id, txn_date
account_balance_internal	account_id_txn_date	account_id, txn_date	idx_fetch_records_based_acc_id_txn_date_isactive	account_id, txn_date, is_active
account_balance_internal	idx_account_id	account_id	idx_fetch_records_based_acc_id_txn_date_isactive	account_id, txn_date, is_active
account_balance_internal	idx_account_id_txn_date	account_id, txn_date	idx_fetch_records_based_acc_id_txn_date_isactive	account_id, txn_date, is_active
account_balance_internal	UNIQUE_Table	account_id, txn_date	idx_fetch_records_based_acc_id_txn_date_isactive	account_id, txn_date, is_active

account_balance_internal	idx_txn_date	txn_date	IX_account_balance_internal	txn_date, account_id, is_active, id
account_balance_internal	account_id_txn_date	account_id, txn_date	UNIQUE_Table	account_id, txn_date
account_balance_internal	idx_account_id	account_id	UNIQUE_Table	account_id, txn_date
account_balance_internal	idx_account_id_txn_date	account_id, txn_date	UNIQUE_Table	account_id, txn_date
account_balance_td	PK_account_balance_td_ordered	id	NonClusteredIndex-20180902-062146	id, td_account_id, txn_date, is_active
account_balance_version	UQ__account___46A222CCC493D33D	account_id	idx_account_id_version	account_id, version
cc_credit_interest_details	cc_credit_interest_details_account_id	account_id	cc_credit_interest_details_account_id1	account_id, is_active
channel_request_transfer	id	id, DayOfMonth	channel_request_transfer_DayOfMonth	id, DayOfMonth
channel_request_transfer	channel_request_transfer_DayOfMonth	id, DayOfMonth	id	id, DayOfMonth
clearing_outward_transaction	txn_ref_no	txn_ref_no	account_id-txn_ref_no	txn_ref_no, account_id
clearing_outward_transaction	txn_ref_no	txn_ref_no	account_no-txn_ref_no	txn_ref_no, account_no
clearing_outward_transaction	txn_ref_no	txn_ref_no	combined	txn_ref_no, txn_amount, is_active, account_no
interest_pnr_pl_mapping	NonClusteredIndex-20221204-143540	pnr_account_id	pnr_account_id	pnr_account_id, is_active
product_balance	idx_product_id	product_id	idx_fetch_records_based_prodb_id_txn_date_isactive	product_id, txn_date, is_active
product_balance	idx_product_id_txn_date	product_id, txn_date	idx_fetch_records_based_prodb_id_txn_date_isactive	product_id, txn_date, is_active
product_balance	idx_product_id	product_id	idx_product_id_txn_date	product_id, txn_date
product_balance_internal	idx_internal_product_id	internal_product_id	idx_fetch_records_based_prod_id_txn_date_isactive	internal_product_id, txn_date, is_active
product_balance_internal	IX_product_balance_internal	internal_product_id, txn_date	idx_fetch_records_based_prod_id_txn_date_isactive	internal_product_id, txn_date, is_active
product_balance_internal	idx_internal_product_id	internal_product_id	IX_product_balance_internal	internal_product_id, txn_date
td_interest_application	NonClusteredIndex-20190701-164224	td_account_id	NonClusteredIndex-20190209-111131	td_account_id, txn_date, is_applied
transaction_master_15112024	NonClusteredIndex-20170829-043657	txn_date, account_id	activity_account	txn_date, account_id, activity_id, is_active
transaction_master_20231101	NonClusteredIndex-20170829-043657	txn_date, account_id	activity_account	txn_date, account_id, activity_id, is_active
transaction_pending_15112024	txn_ref_no	txn_ref_no	scroll	txn_ref_no, txn_branch, txn_date, batch_code, scroll_no
transaction_pending_20231101	txn_ref_no	txn_ref_no	scroll	txn_ref_no, txn_branch, txn_date, batch_code, scroll_no
transaction_reversal	txn_ref_no	reversal_ref_no, is_active	reversal_ref_no	reversal_ref_no, is_active

transaction_reversal	reversal_ref_no	reversal_ref_no, is_active	txn_ref_no	reversal_ref_no, is_active
transaction_status_20231101	txn_type	txn_type	type_state_br	txn_type, txn_branch_code, txn_state, txn_date, is_active
transaction_status_22092024	txn_type	txn_type	type_state_br	txn_type, txn_branch_code, txn_state, txn_date, is_active

16. Unused Indexes:

Unused indexes in SQL Server refer to indexes that are not being utilized by queries. These indexes consume storage space and can impact the performance of data modification operations (such as INSERT, UPDATE, and DELETE) as well as the overall database performance.

DB Name: BSGACCOUNTING

Table_name	Index_name	User seeks	User scans	User Updates
transaction_scrollno	branch-batch-code	0	0	247603888
transaction_scrollno	IX_transaction_scrollno_created_date	0	0	247602615
transaction_master_internal	idx_id_principal_id	0	0	100582747
transaction_reversal	reversal_ref_no	0	0	2362596
BULK_SMS_DETAILS	isActive	0	0	977241
BULK_SMS_DETAILS	NULL	0	0	977241
customer_balance	wallet_no	0	0	652388
neft_rtgs	remitter_princ_id_r_p_t__is_active	0	0	174716
principle_turing_temp	idx_principle_key	0	0	47463
tq_batch_processing_status	idx_for_last_processed_id	0	0	2434
account_last_transaction_date	account_id	0	0	96
txn_ref_token_mapping	token_id	0	0	92
txn_ref_token_mapping	txn_ref_token_id_is_Active	0	0	92
transaction_denomination	txn_ref_no	0	0	92
transaction_denomination	token_id	0	0	92
teller_denomination	txn_date_principal_type_txn_branch	0	0	92
teller_denomination	principal_type_principal_id	0	0	92
teller_denomination	principal_type	0	0	92
teller_denomination	principal_id	0	0	92
clearing_inward_transaction	txn_ref_no	0	0	92
clearing_inward_transaction	short_acc_txn_code	0	0	92
clearing_inward_transaction	account_id	0	0	92
token_status	token_id-no-state-nature-is_Active	0	0	92
token_status	token_no_is_issued	0	0	92
token_status	token_no_txn_date	0	0	92
token_status	txn_date_nature_den_add_act_bran_used	0	0	92
token_status	token_id_is_active	0	0	92
ecs_inward_transaction	txn_ref_no	0	0	92
ecs_inward_transaction	post_date_is_active	0	0	92
ecs_inward_transaction	file_id_is_active	0	0	92
ecs_inward_transaction	account_no_ref_no_amount_txn_nature_is_active	0	0	92

clearing_outward_transaction	account_id-txn_ref_no	0	0	92
clearing_outward_transaction	account_no-txn_ref_no	0	0	92
clearing_outward_transaction	combined	0	0	92
clearing_outward_transaction	settlement_product_id_idx	0	0	92
clearing_outward_transaction	txn_ref_no	0	0	92
interest_pnr_pl_mapping	NonClusteredIndex-20221204-143540	0	0	2
interest_pnr_pl_mapping	pl_account_id	0	0	2
bc_transaction_accounts	idx_for_account_id_bc_transaction_accounts	0	0	1

17. Missing Indexes:

The audit noted that the all database could benefit from additional indexes to improve query performance. While not critical, the absence of these indexes may lead to slower query execution. Implementing the suggested indexes can help streamline operations and enhance overall efficiency.

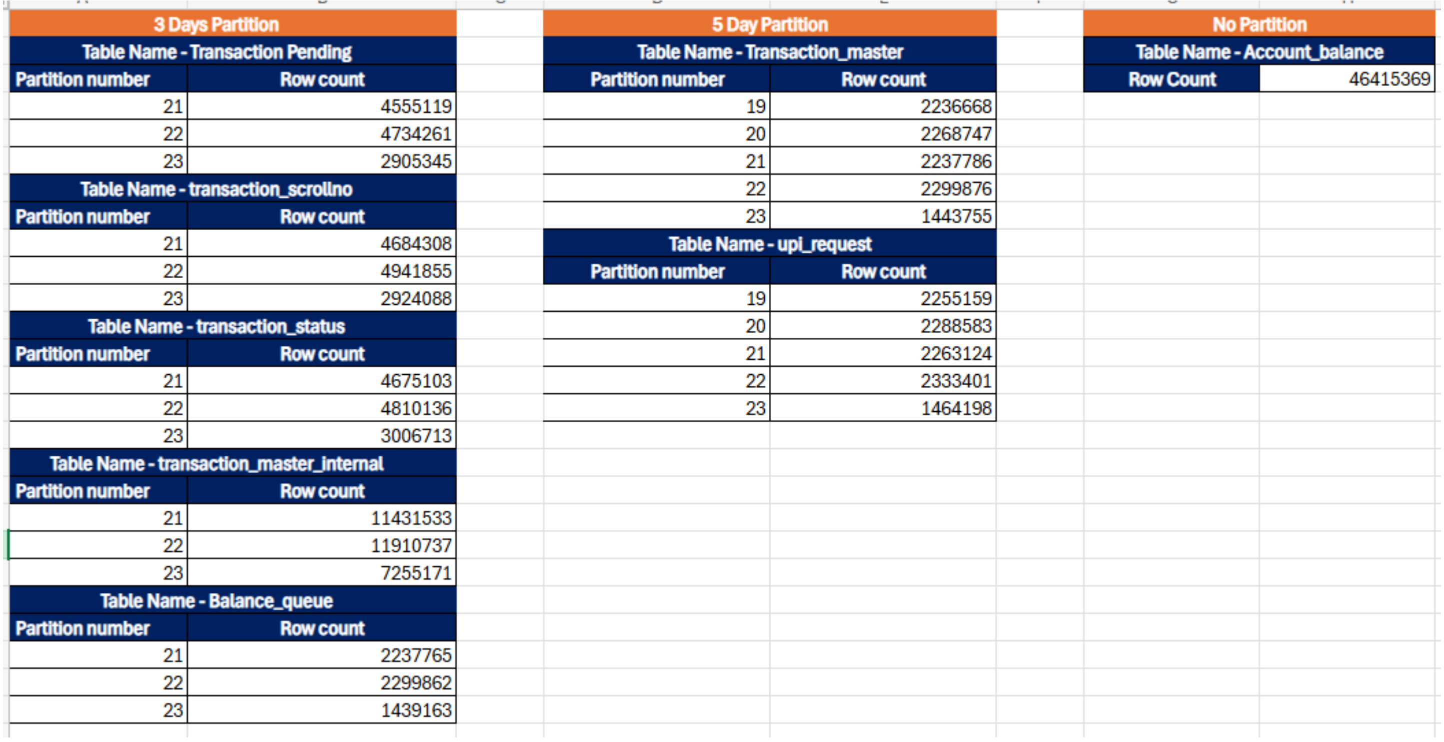
Create index statement	User seeks	Last user seek	Avg user impact
CREATE INDEX missing_index_25916_25915 ON [BSGCRM].[dbo].[customer_ind_info] ([pan_no], [is_active]) INCLUDE ([customer_id])	658739	2025-06-23 17:58:27.830	98.78
CREATE INDEX missing_index_25918_25917 ON [BSGCRM].[dbo].[customer_ind_info] ([aadhar_no], [is_active]) INCLUDE ([customer_id])	360954	2025-06-23 17:58:28.727	90.84
CREATE INDEX missing_index_146_145 ON [BSGCRM].[dbo].[customer_unification] ([principle_id], [is_valid])	9489246	2025-06-23 15:02:02.967	87.5
CREATE INDEX missing_index_29871_29870 ON [BSGCRM].[dbo].[customer_ind_info] ([pan_no]) INCLUDE ([customer_id])	122364	2025-06-23 17:57:58.667	98.7
CREATE INDEX missing_index_29873_29872 ON [BSGCRM].[dbo].[customer_ind_info] ([aadhar_no]) INCLUDE ([customer_id])	122321	2025-06-23 17:57:58.950	91.53
CREATE INDEX missing_index_2785_2784 ON [BSGDEEPFREEZE].[dbo].[transaction_master_internal_df] ([txn_posting_date], [principal_id], [is_active]) INCLUDE ([txn_ref_no], [txn_sub_ref_no], [txn_branch], [home_branch], [txn_date], [txn_time], [txn_type], [txn_value_date], [principal_type], [batch_code], [scroll_no], [set_no], [txn_amount], [txn_nature], [instr_no], [instr_date], [activity_id], [activity_sub_type_id], [narration], [sprinkle], [created_by], [created_date], [last_modified_by], [last_modified_date], [is_reconciled])	7182	2025-06-09 13:00:01.637	80.56
CREATE INDEX missing_index_31471_31470 ON [BSGCRM].[dbo].[customer_kyc_details] ([document_no])	21785	2025-06-23 17:56:16.447	99.84

CREATE INDEX missing_index_6_5 ON [BSGADMIN].[dbo].[employee_master] ([employee_id], [is_active])	117755114	2025-06-23 17:58:28.093	31.34
CREATE INDEX missing_index_746_745 ON [BSGCORE].[dbo].[change_account_product] ([id])	218572	2025-06-23 02:56:00.830	99.68
CREATE INDEX missing_index_153_152 ON [BSGCRM].[dbo].[customer_unification] ([principle_id], [old_customer_id], [is_valid])	1541537	2025-06-23 15:02:02.797	99.08
CREATE INDEX missing_index_106681_106680 ON [BSGDEEPFREEZE].[dbo].[transaction_master_internal_df] ([principal_id], [txn_amount], [narration])	17	2025-06-02 20:25:08.770	94.04
CREATE INDEX missing_index_29643_29642 ON [BSGCRM].[dbo].[customer_unification] ([principle_id], [new_customer_id], [is_valid])	706068	2025-06-23 15:02:02.510	99.11
CREATE INDEX missing_index_106326_106325 ON [BSGDEEPFREEZE].[dbo].[transaction_pending_df] ([txn_ref_no], [txn_nature],[account_id]) INCLUDE ([txn_branch], [home_branch], [txn_amount], [narration])	31	2025-06-03 11:16:07.310	99.63
CREATE INDEX missing_index_1749_1748 ON [BSGDEEPFREEZE].[dbo].[transaction_master_internal_df] ([txn_ref_no]) INCLUDE ([txn_sub_ref_no], [txn_branch], [home_branch], [txn_date], [txn_time], [txn_type], [txn_posting_date], [txn_value_date], [principal_id], [principal_type], [batch_code], [scroll_no], [set_no], [txn_amount], [txn_nature], [instr_no], [instr_date], [activity_id], [activity_sub_type_id], [narration], [sprinkle], [is_active], [created_by], [created_date], [last_modified_by], [last_modified_date], [is_reconciled], [source_principal_type], [source_principal_id], [fcy_currency], [lcy_amount], [fcy_exchange_rate])	27	2025-06-03 15:33:57.073	100
CREATE INDEX missing_index_22854_22853 ON [BSGCORE].[dbo].[change_account_product] ([account_id], [transfer_date], [is_active])	75495	2025-06-23 02:52:47.303	99.79
CREATE INDEX missing_index_27401_27400 ON [BSGDEEPFREEZE].[dbo].[transaction_master_internal_df] ([txn_ref_no]) INCLUDE ([txn_posting_date])	8	2025-06-04 13:14:25.113	100
CREATE INDEX missing_index_83_82 ON [BSGCRM].[dbo].[customer_poi_poa_failure_details] ([customer_id], [poi_updated], [updated_status], [is_active])	36400	2025-06-23 17:57:29.740	99.76
CREATE INDEX missing_index_111459_111458 ON [BSGDEEPFREEZE].[dbo].[transaction_pending_df] ([txn_ref_no], [txn_date]) INCLUDE ([txn_branch], [home_branch], [txn_time], [txn_type], [value_date], [account_id], [batch_code], [scroll_no], [set_no], [txn_amount], [txn_nature], [instr_no], [instr_date], [narration], [activity_id], [activity_sub_type_id], [sprinkle], [is_active], [created_by], [created_date], [last_modified_by], [last_modified_date], [txn_type_base], [internal_product_id], [fcy_currency], [lcy_amount], [fcy_exchange_rate])	12	2025-06-04 22:18:43.010	100

18. Table Size:

Analysis of the top 20 tables reveals that several large, high-activity tables—such as transaction_master_internal, transaction_pending, and transaction_status—are partitioned and consume significant storage, with individual tables using up to 91 GB. The partitioned tables listed in the attached summary are experiencing slowness during peak evening insert loads, likely due to high row counts per partition and potential partition or index hot spots.

Table	Used mb	Allocated mb
dbo.transaction_master_internal_20231101	91519.16	91549.69
dbo.account_balance_backup	77878.24	79138.39
dbo.transaction_pending_20231101	37508.79	37515.05
dbo.transaction_status_20231101	27280.63	27284.8
dbo.account_balance	16312.55	16611.14
dbo.transaction_master_internal	14477.98	14538.1
dbo.transaction_master_20231101	11982.39	11985.82
dbo.upi_merchant_transactions_request_24022024	10264.16	10278.03
dbo.principle_turing_temp	9289.81	9295.94
dbo.transaction_scrollno_20231101	7082.36	7083.55
dbo.upi_request	7025.3	7176.41
dbo.transaction_pending	5938.37	5972.3
dbo.transaction_master	5124.05	5172.13
dbo.transaction_status	5119.91	5168.33
dbo.transaction_status_22092024	4323.55	4377.88
dbo.transaction_pending_15112024	4299.17	4363.87
dbo.transaction_scrollno	3592.25	3645.55
dbo.account_wise_interest_accrual_temp	3162.89	3164.18
dbo.upi_request_03092024	3032.86	3039.63



3 Days Partition	
Table Name - Transaction Pending	
Partition number	Row count
21	4555119
22	4734261
23	2905345
Table Name - transaction_scrollno	
Partition number	Row count
21	4684308
22	4941855
23	2924088
Table Name - transaction_status	
Partition number	Row count
21	4675103
22	4810136
23	3006713
Table Name - transaction_master_internal	
Partition number	Row count
21	11431533
22	11910737
23	7255171
Table Name - Balance_queue	
Partition number	Row count
21	2237765
22	2299862
23	1439163

5 Day Partition	
Table Name - Transaction_master	
Partition number	Row count
19	2236668
20	2268747
21	2237786
22	2299876
23	1443755
Table Name - upi_request	
Partition number	Row count
19	2255159
20	2288583
21	2263124
22	2333401
23	1464198

No Partition	
Table Name - Account_balance	
Row Count	46415369

Recommendation:

It is advisable to move historical or archival tables—such as those with date-specific suffixes (e.g., _20231101, _15112024)—to a separate archival database. This approach will reduce the size and index maintenance overhead on your primary database, improve insert performance for active partitions, and simplify backup and recovery operations.

19. Top 15 Wait Type:

The wait statistics indicate that SOS_WORK_DISPATCHER accounts for the majority of wait time (71.93%), reflecting idle CPU capacity and no performance bottleneck. Significant waits for CXCONSUMER (8.42%) and CXPACKET (0.97%) highlight heavy parallel query activity, which may require tuning to avoid parallelism inefficiencies. Transaction log-related waits such as LOGMGR_QUEUE (5.12%) and WRITELOG (1.57%) suggest potential log write bottlenecks or disk latency. SOS_SCHEDULER_YIELD (1.56%) points to some CPU pressure during peak workloads. Moderate waits like LCK_M_U (1.10%) and PAGELATCH_EX (0.49%) indicate locking and in-memory page contention, respectively. Other waits related to diagnostics and background tasks are minimal and generally not concerning. Overall, the system shows good CPU availability but experiences contention in parallelism, log writes, and locking, which should be addressed to improve performance during peak insert periods.

Wait type	Wait time seconds	Waiting tasks count	Wait percentage
SOS_WORK_DISPATCHER	493557969.706000	21979790954	71.93
CXCONSUMER	57838329.996000	1629273126	8.42
LOGMGR_QUEUE	35155770.943000	3266888105	5.12
WRITELOG	10810783.012000	2660473103	1.57
SOS_SCHEDULER_YIELD	10738155.639000	17333578427	1.56
SP_SERVER_DIAGNOSTICS_SLEEP	8868380.079000	9008439	1.29
QDS_PERSIST_TASK_MAIN_LOOP_SLEEP	8867531.939000	147781	1.29

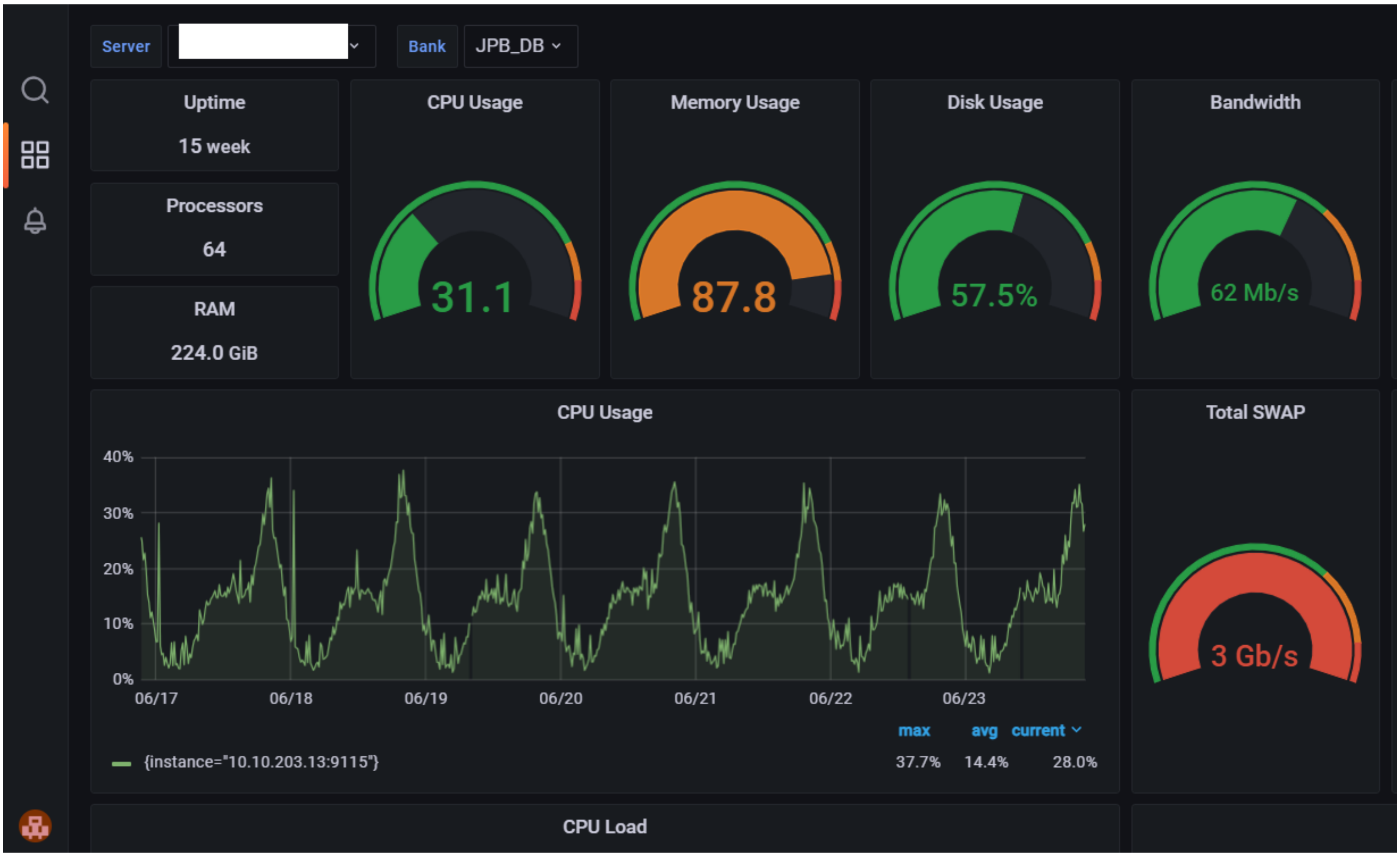
PREEMPTIVE_XE_DISPATCHER	8846668.143000	25	1.28
DIRTY_PAGE_POLL	8842316.209000	83280215	1.28
CHECKPOINT_QUEUE	8785796.225000	7964092	1.28
LCK_M_U	7612689.502000	4067677	1.10
CXPACKET	6663809.182000	2009905047	0.97
PAGELATCH_EX	3365663.520000	1238141056	0.49

Recommendation:

- Increase Cost Threshold parallelism to 50.
- MAXDOP=4 is appropriate for OLTP/mixed workloads, if specific queries cause high CXPACKET, use OPTION (MAXDOP N) hints only for those queries.
- Re-write the maintenance plan , we observed huge difference between rows & sampled rows

20. Performance Metrics & Connections:

Performance metrics show that while CPU usage averages 14.4% with peaks up to 37.7%, memory usage is consistently high at nearly 88%, and disk usage stands at 57.5%1. User connection analysis over the past week reveals a clear pattern of increased activity during evening hours, with connection counts regularly spiking above 1,150 compared to lower levels in the mornings. This indicates that system load and resource consumption are significantly higher in the evenings, which aligns with reported periods of performance degradation and should be a focus area for further tuning and capacity planning.



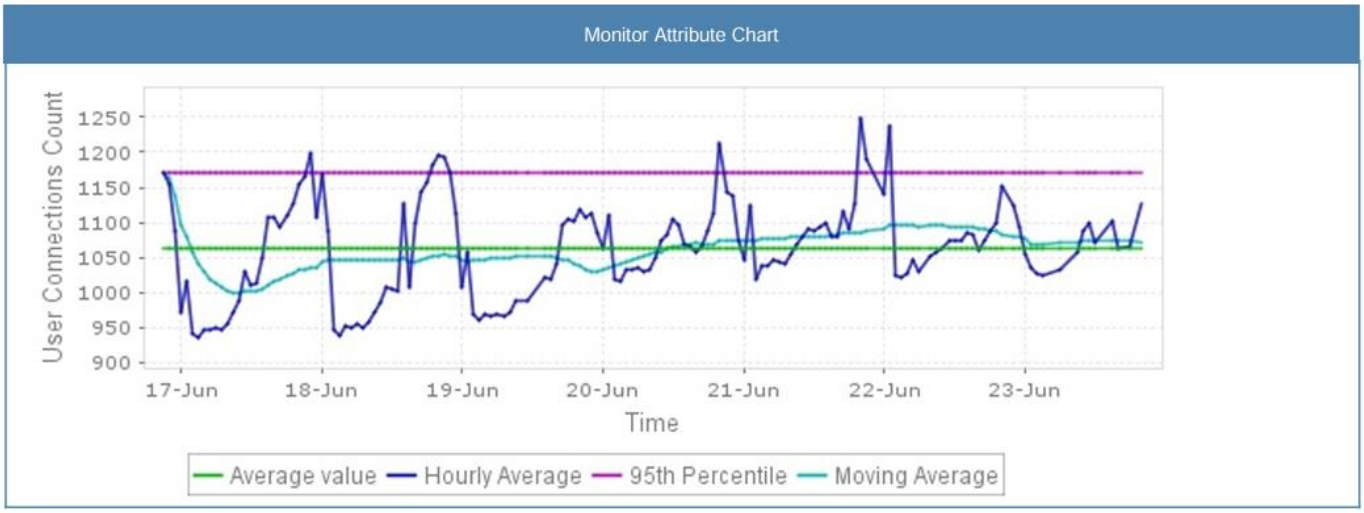
Server
Bank JPB_DB

Uptime
15 week

Processors
64

RAM
224.0 GiB

Last 7 Days User Connections - Attribute Report for JPB_CBS_DB



21. Index Fragmentation & Recommendation:

The index fragmentation analysis shows that while the majority of indexes (2,752) have low fragmentation (0-10%), a significant number of indexes—over 300—exhibit moderate to high fragmentation levels (above 50%), which can degrade query performance and increase I/O.

Fragmentation In Percentage	Index Count
90-100%	157
80-90%	38
70-80%	22
60-70%	40
50-60%	37
40-50%	13
30-40%	23
20-30%	59
10-20%	31
0-10%	2752

Recommendation: To address this, it is recommended to implement a robust index maintenance strategy using Ola Hallengren's widely recognized scripts, which provide best practices for rebuilding and reorganizing indexes efficiently. For partitioned tables, index maintenance should be performed at the partition level where possible to minimize resource usage and downtime. Additionally, updating statistics daily is essential to ensure the query optimizer has accurate data distribution information, improving execution plan quality and overall system performance.

22. Table & Row Compression:

In reviewing the database structure and performance, it is evident that several large and partitioned tables consume substantial storage and experience heavy insert activity. Implementing table and index compression is recommended to optimize storage utilization and enhance I/O efficiency.

Starting with row-level compression on active partitions can reduce the storage footprint with minimal CPU overhead, while page-level compression may be suitable for historical or less frequently updated partitions to maximize space savings.

23. Memory Utilization and Configuration:

The server has a total physical memory of 224 GB, with approximately 176 GB currently committed. To ensure optimal SQL Server performance and prevent excessive memory pressure on the operating system, it is recommended to configure the SQL Server **max server memory** setting to about 90% of the total physical memory, which equates to roughly 200 GB. This configuration reserves sufficient memory for the OS and other processes, helping maintain system stability while allowing SQL Server to efficiently utilize available memory for caching and query processing.

24. NUMA Reconfiguration

The current SQL Server environment is configured with 9 NUMA nodes, each with 8 schedulers, which is typical for servers with a large number of cores. While this configuration appears balanced, we are recommending to a 4-node NUMA configuration to potentially improve performance and simplify resource management.

25. Cumulative Recommendations for Performance Improvement:

1. **Update SQL Server to Latest Patch**
   Apply the latest cumulative update for SQL Server 2019 to ensure all recent fixes and improvements are in place. As of June 2025, the latest available is CU32 (Build 15.0.4430.1) released on February 27, 2025.

2. **Enable Recommended Trace Flags**
   Enable the following trace flags globally at SQL Server startup to improve query optimizer behavior and file growth consistency:

   - TF 4199 (enables query optimizer fixes)

   - TF 1117 (ensures uniform file growth for all files in a filegroup)

   - TF 1118 (allocates full extents to tempdb to reduce contention)
     Use the -T startup option or DBCC TRACEON with global scope and restart SQL Server for these to take effect.

3. **Increase Cost Threshold for Parallelism**
   Increase cost threshold for parallelism from 5 to 50 to reduce unnecessary parallel query plans and improve overall CPU utilization.

4. **Change to Fill Factor to 95%:**
   Set fill factor to 95% on indexes to leave free space for inserts and reduce page splits during index rebuilds.

5. **Enable Forced Parameterization:**
   Turn ON forced parameterization on the BSGACCOUNTING database to reduce query plan compilations for repetitive insert statements, improving insert performance.

6. **Optimize Tempdb Configuration:**
   Increase the number of tempdb data files to at least 8 (one per CPU up to 8) to reduce allocation contention and improve concurrency.

7. **Enable Lock Pages in Memory (LPIM):**
   Configure the SQL Server service account to have the LPIM privilege to prevent the OS from paging out SQL Server memory, stabilizing performance under load.

8. **Rebuild Tables with Clustered Indexes:**
   Convert heap tables (tables without clustered indexes) to clustered indexes to reduce fragmentation, improve IO efficiency, and speed up inserts and queries.

9. **Enable Snapshot Isolation:**
   Enable snapshot isolation to reduce locking and blocking during heavy concurrent insert operations, improving concurrency and reducing wait times.

10. **Manage Transaction Log Growth:**
    Set a maximum size limit on the transaction log file (e.g., 500 GB) to prevent uncontrolled growth and ensure regular transaction log backups are scheduled to maintain log size and performance.

11. **Enable Schema Binding on Scalar UDFs:**
    Review scalar user-defined functions and enable schema binding where possible to improve query plan stability and performance.

12. **Enable SET ARITHABORT ON in Application:**
    Modify application connection strings to enable SET ARITHABORT ON to avoid query plan regressions and improve query performance.

13. **Monitor and Tune Disk Performance:**
    Continuously monitor disk latency and IO throughput on drives hosting database files, especially those containing partitioned filegroups, to identify and mitigate IO bottlenecks.

14. **Index Optimization:**

    - Drop duplicate indexes that cause unnecessary overhead.

    - Drop unused indexes to reduce maintenance and IO costs.

    - Create missing indexes identified by query analysis to improve query performance.

15. **Review and Optimize Partitioning Strategy:**
    Since only 3 days & 5 days of data are stored in the partitioned tables. Consider archiving large

historical data tables to a separate archive database to reduce active database size and improve performance.

16. **Re-write Maintenance Scripts Using Ola Hallengren's Solution:**
Replace existing maintenance scripts with Ola Hallengren's well-known index and statistics maintenance scripts. Prioritize rebuilding or reorganizing indexes with fragmentation greater than 90%, especially those on critical partitioned tables, to improve IO efficiency and query performance.

17. **Table Compression:**
Row-level compression on active partitions can reduce the storage footprint with minimal CPU overhead, while page-level compression may be suitable for historical or less frequently updated partitions to maximize space savings.

18. **Offloading Read-Only Workloads:**
To improve performance and reduce load on the primary server, we recommend adding two synchronous secondary nodes and configuring round-robin read-only routing across all three nodes. This will balance read workloads, enhance scalability, and maintain data consistency.

19. **Memory Utilization and Configuration Recommendation:**
Configure SQL Server max memory to approximately 90% of total physical memory (around 200 GB) to ensure balanced memory allocation between SQL Server and the operating system.

20. **NUMA Reconfiguration:**
The current SQL Server environment is configured with 9 NUMA nodes, each with 8 schedulers, which is typical for servers with a large number of cores. While this configuration appears balanced, we are recommending to a 4-node NUMA configuration to potentially improve performance and simplify resource management.