Chapter 13: Database Monitoring

একটি Production Database শুধু তৈরি করে রেখে দিলেই হয় না। Database সবসময় **Healthy, Available, Secure এবং Performant** অবস্থায় আছে কি না, তা নিয়মিত পর্যবেক্ষণ করতে হয়।

একটি Database ধীরে ধীরে সমস্যার দিকে যেতে পারে। যেমন—

- CPU Usage বেড়ে যাওয়া
- RAM কমে যাওয়া
- Storage Full হয়ে যাওয়া
- Blocking Session তৈরি হওয়া
- Deadlock হওয়া
- Long Running Query চলা
- Connection সংখ্যা বেড়ে যাওয়া
- Disk I/O বৃদ্ধি পাওয়া
- Backup ব্যর্থ হওয়া
- Replication/Standby Lag তৈরি হওয়া
- Database Service Down হয়ে যাওয়া

এই সমস্যাগুলো User-এর কাছে দৃশ্যমান হওয়ার আগেই শনাক্ত করার জন্য **Database Monitoring** অত্যন্ত গুরুত্বপূর্ণ।

সহজভাবে বলা যায়—

Database Monitoring হলো Database-এর Health, Performance, Availability, Resource Usage এবং বিভিন্ন সমস্যা নিয়মিত পর্যবেক্ষণ ও বিশ্লেষণ করার প্রক্রিয়া।

Database Monitoring-এর প্রধান উদ্দেশ্য

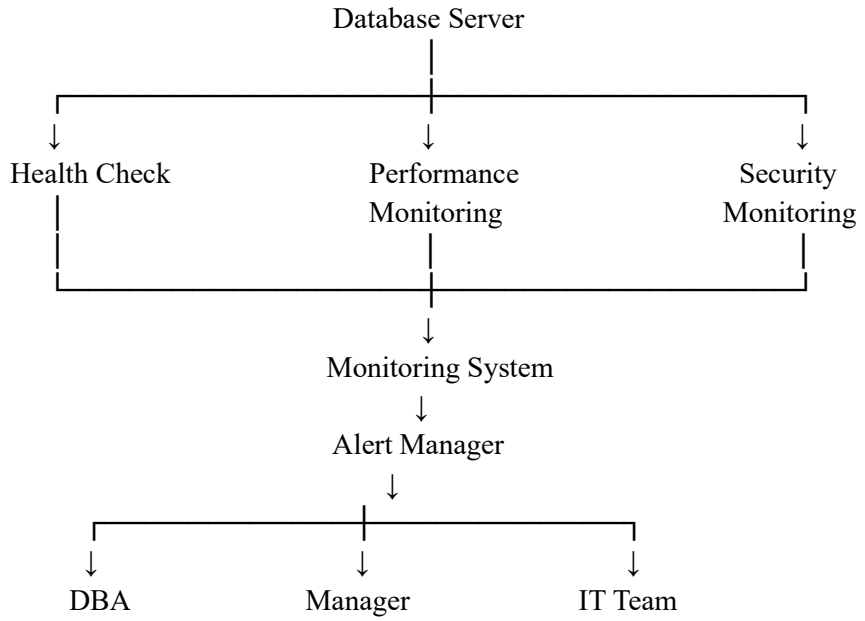
Database Monitoring-এর মূল উদ্দেশ্য হলো—

1. Database সবসময় Available রাখা
2. Performance সমস্যা দ্রুত শনাক্ত করা
3. Resource Usage পর্যবেক্ষণ করা
4. Blocking ও Deadlock শনাক্ত করা

5. Long Running Query খুঁজে বের করা
6. Capacity Planning করা
7. Backup ও Recovery Status Monitor করা
8. Security Incident শনাক্ত করা
9. Production Incident-এর Impact কমানো
10. ভবিষ্যতের সমস্যা আগে থেকে অনুমান করা

Database Monitoring Architecture

একটি Professional Database Monitoring System সাধারণভাবে এভাবে কাজ করতে পারে—



১. Database Health Check

Database Health Check কী?

Database Health Check হলো Database-এর সামগ্রিক অবস্থা পরীক্ষা করার একটি প্রক্রিয়া।

এখানে দেখা হয় Database স্বাভাবিকভাবে চলছে কি না এবং কোনো গুরুত্বপূর্ণ সমস্যা তৈরি হয়েছে কি না।

Health Check-এ কী কী দেখা হয়?

Database Availability

Database Service
↓
Running?
↓
YES / NO
Database Service বা Instance Running আছে কি না পরীক্ষা করতে হবে।

Database Status

Database-এর বর্তমান Status পরীক্ষা করতে হবে।

Oracle Database-এর ক্ষেত্রে উদাহরণ:

```
SELECT STATUS

FROM V$INSTANCE;
```

Storage Usage

Database-এর Storage কতটুকু ব্যবহৃত হয়েছে তা Monitor করতে হবে।

Storage

80%

Storage 90% বা 95%-এর কাছাকাছি গেলে Alert তৈরি করা যেতে পারে।

Tablespace Usage

Oracle Database-এ Tablespace Usage Monitor করা গুরুত্বপূর্ণ।

বিশেষ করে—

- Total Space
- Used Space
- Free Space
- Usage Percentage

দেখতে হবে।

Datafile Status

Datafile Online আছে কি না এবং কোনো সমস্যা হয়েছে কি না পরীক্ষা করতে হবে।

Archive Log Usage

Oracle Database-এ Archive Log Destination-এর Space Full হয়ে গেলে Database-এর স্বাভাবিক Operation ব্যাহত হতে পারে।

তাই Archive Log Space Monitor করা অত্যন্ত গুরুত্বপূর্ণ।

২. Performance Monitoring

Performance Monitoring কী?

Database কত দ্রুত কাজ করছে এবং কোন Resource কতটুকু ব্যবহার করছে তা পর্যবেক্ষণ করাকে **Performance Monitoring** বলা হয়।

Performance Monitoring-এর গুরুত্বপূর্ণ Metric

CPU Usage

Database Server কতটুকু CPU ব্যবহার করছে।

CPU Usage = 85%

CPU দীর্ঘ সময় ধরে অত্যধিক ব্যবহার হলে Query এবং Application Performance প্রভাবিত হতে পারে।

Memory Usage

Database এবং Operating System কত Memory ব্যবহার করছে তা Monitor করতে হবে।

Disk I/O

Database-এর জন্য Disk Read/Write কত হচ্ছে তা দেখতে হবে।

High I/O-এর কারণ হতে পারে—

- Full Table Scan
- Poor Query Plan
- Heavy Reporting Query

- Large Sort
- Insufficient Memory
- Slow Storage

Database Connections

একই সময়ে কত User বা Application Connection Database-এর সঙ্গে যুক্ত আছে তা Monitor করতে হবে।

Connections

Current : 450

Maximum : 500

Connection Limit-এর কাছাকাছি চলে গেলে নতুন Application Connection ব্যর্থ হতে পারে।

Query Response Time

Query কত সময়ে Result দিচ্ছে সেটিও গুরুত্বপূর্ণ Metric।

Normal Query → 1 sec

Slow Query → 30 sec

Critical Query → 5 min

৩. Blocking Session

Blocking Session কী?

একটি Database Session যখন অন্য Session-এর Resource Access আটকে রাখে, তখন সেটিকে **Blocking Session** বলা হয়।

সহজ উদাহরণ—

```
Session A
  │
  └── Lock ধরে রেখেছে
      ↓
    Session B
      │
      └── অপেক্ষা করছে
```

ধরুন Session A একটি Row Update করেছে কিন্তু Transaction Commit করেনি।

অন্যদিকে Session B একই Row Update করতে চাইল।

Session B তখন Session A-এর জন্য অপেক্ষা করতে পারে।

Blocking কেন হয়?

সাধারণ কারণ—

- Long Transaction
- Uncommitted Transaction
- Excessive Lock
- Poor Application Design
- Long Running Update/Delete
- Incorrect Transaction Management

Blocking শনাক্ত করার পর কী করবেন?

প্রথমে দেখতে হবে—

- Blocking Session কে?
- কোন SQL চলছে?
- কতক্ষণ ধরে Lock আছে?
- Transaction কেন Commit করছে না?
- Blocking Session বন্ধ করা নিরাপদ কি না?

সরাসরি Session Kill করা উচিত নয়।

আগে Root Cause বুঝতে হবে।

৪. Deadlock

Deadlock কী?

যখন দুই বা তার বেশি Transaction একে অপরের Resource-এর জন্য অপেক্ষা করতে থাকে এবং কেউই এগোতে পারে না, তখন তাকে **Deadlock** বলা হয়।

উদাহরণ:

```
Transaction A
    │
    ├── Lock Resource 1
    │
    └── Waiting for Resource 2
                  ↑
                  │
Transaction B     │
    │             │
    ├── Lock Resource 2
    │
    └── Waiting for Resource 1
```

অর্থাৎ—

A অপেক্ষা করছে B-এর জন্য এবং B অপেক্ষা করছে A-এর জন্য।

Deadlock-এর উদাহরণ

Transaction A:
Lock Table A
↓
Lock Table B
Transaction B:
Lock Table B
↓
Lock Table A

এক পর্যায়ে—

A → B-এর Resource-এর জন্য অপেক্ষা করছে

B → A-এর Resource-এর জন্য অপেক্ষা করছে

এটাই Deadlock।

Deadlock-এর কারণ

- Different Order-এ Resource Lock করা
- Long Transaction
- Poor Transaction Design
- Excessive Locking
- Application Concurrency Problem

Deadlock Prevention

একটি গুরুত্বপূর্ণ Strategy হলো সব Transaction-এ Resource একই Order-এ Access করা।

Transaction A:

Table A → Table B

Transaction B:

Table A → Table B

এতে Circular Waiting-এর সম্ভাবনা কমানো যায়।

৫. Long Running Query

Long Running Query কী?

যে Query স্বাভাবিক সময়ের তুলনায় অস্বাভাবিকভাবে বেশি সময় ধরে Execute হয় তাকে **Long Running Query** বলা হয়।

উদাহরণ:

Expected Time → 5 seconds

Actual Time → 15 minutes

এটি Database Performance-এর জন্য গুরুত্বপূর্ণ Warning Sign হতে পারে।

Long Running Query-এর কারণ

- Missing Index
- Poor Execution Plan
- Large Table Scan
- Incorrect Join
- Large Sort
- High I/O
- Blocking
- Resource Contention

- Poor SQL Design
- Large Data Processing

Long Running Query শনাক্ত করার Process

Slow Application
↓
Identify SQL
↓
Check Execution Plan
↓
Check Wait Event
↓
Check Index
↓
Check I/O / CPU
↓
Optimize

৬. Database Connection Monitoring

Database-এ কত Connection Open আছে তা নিয়মিত Monitor করা উচিত।

উদাহরণ:

Maximum Connection = 1000

Current Connection = 920

এটি একটি Warning Signal হতে পারে।

Connection বৃদ্ধির কারণ হতে পারে—

- Application Connection Leak
- Traffic Increase
- Connection Pool Misconfiguration
- Long Running Session
- Application Error

৭. Tablespace Monitoring

Oracle Database Environment-এ Tablespace Monitoring অত্যন্ত গুরুত্বপূর্ণ।

ধরুন—

TABLESPACE	USED
SYSTEM	45%
SYSAUX	60%
USERS	72%
APP_DATA	91%

APP_DATA 91% হলে DBA-কে দ্রুত ব্যবস্থা নিতে হবে।

কারণ Storage Full হয়ে গেলে নতুন Data Insert ব্যর্থ হতে পারে।

৮. Storage Monitoring

Database-এর Disk এবং Storage Health Monitor করতে হবে।

Monitor করা যেতে পারে—

- Disk Usage
- Free Space
- IOPS
- Read Latency
- Write Latency
- Throughput
- Disk Failure

বিশেষ করে Database Server-এ Storage Performance অত্যন্ত গুরুত্বপূর্ণ।

৯. Backup Monitoring

Backup সফল হয়েছে কি না সেটিও Database Monitoring-এর অংশ।

শুধু Backup Schedule তৈরি করলেই হবে না।

প্রতিদিন Check করতে হবে—

Backup Job
↓
SUCCESS / FAILED
↓
Backup Size
↓
Backup Duration
↓
Backup Location

Backup Monitoring Example

Full Backup → SUCCESS

Incremental → SUCCESS

Archive Log → SUCCESS

Backup Validation → SUCCESS

যদি Backup Failed হয়, তাহলে DBA Team-কে Alert দিতে হবে।

১০. Replication / Data Guard Monitoring

যদি Database Replication বা Oracle Data Guard ব্যবহার করা হয়, তাহলে Replication Health Monitor করতে হবে।

দেখতে হবে—

- Primary Status
- Standby Status
- Replication Status
- Transport Lag
- Apply Lag
- Archive Gap
- Redo Transport Error

Data Guard Example

Primary Database
│
│ Redo
↓
Standby Database
│
↓
Apply
যদি Standby অনেক পিছিয়ে যায়—

Primary
↓
Standby Lag = 2 Hours
তাহলে এটি একটি Critical Warning হতে পারে।

১১. Wait Event Monitoring

Database Performance Analysis-এর জন্য **Wait Event** Monitor করা অত্যন্ত গুরুত্বপূর্ণ।

Database কোন Resource-এর জন্য অপেক্ষা করছে তা জানা গেলে Root Cause Analysis সহজ হয়।

উদাহরণ—

High CPU

High I/O

Lock Wait

Network Wait

Memory-related Wait

এটি Performance Tuning-এর সঙ্গে সরাসরি সম্পর্কিত।

১২. Top SQL Monitoring

Database-এ সবচেয়ে বেশি Resource ব্যবহার করা SQL Query শনাক্ত করা উচিত।

দেখা যেতে পারে—

- CPU Usage
- Execution Count
- Elapsed Time

- Logical Reads
- Physical Reads
- I/O
- Average Execution Time

উদাহরণ:

SQL ID CPU Time

--- ---

SQL001 85%

SQL002 10%

SQL003 5%

এখানে SQL001 প্রথমে বিশ্লেষণ করা যেতে পারে।

১৩. Database Error Monitoring

Database Error Log নিয়মিত Monitor করতে হবে।

খেয়াল রাখতে হবে—

- Database Error
- Connection Error
- Storage Error
- Authentication Failure
- Deadlock
- Crash
- Data Corruption
- Backup Error

১৪. Alert এবং Notification

Database Monitoring-এর সবচেয়ে গুরুত্বপূর্ণ অংশ হলো **Alerting**।

শুধু Problem Detect করলে হবে না—Problem হলে Responsible Team-কে জানাতে হবে।

Database
↓
Monitoring
↓
Threshold Crossed
↓
Alert
↓
DBA / IT Team
↓
Investigation

উদাহরণ

CPU > 90%
↓
Critical Alert
↓
DBA Team
অথবা—
Tablespace > 85%
↓
Warning Alert

১৫. Monitoring Threshold

সব Metric-এর জন্য Threshold নির্ধারণ করা উচিত।

উদাহরণ:

Metric	Warning	Critical
CPU	> 75%	> 90%
Memory	> 80%	> 90%
Disk	> 80%	> 90%
Tablespace	> 80%	> 90%
Connection	> 80%	> 95%
Replication Lag	> 5 min	> 30 min

নোট: এগুলো উদাহরণ মাত্র। Production Environment-এ Business Workload, Hardware এবং Application Pattern অনুযায়ী Threshold নির্ধারণ করতে হবে।

১৬. Database Monitoring Tools

Database Monitoring-এর জন্য Database-specific এবং Enterprise Monitoring—দুই ধরনের Tool ব্যবহার করা যায়।

Oracle Database

Oracle Environment-এ ব্যবহার করা যেতে পারে—

Oracle Enterprise Manager

Database-এর—

- Performance
- Sessions
- SQL
- Storage
- Wait Events
- Availability

ইত্যাদি Monitor করতে সাহায্য করে।

AWR

Automatic Workload Repository ব্যবহার করে Historical Performance Analysis করা যায়।

ASH

Active Session History ব্যবহার করে Active Session ও Wait Analysis করা যায়।

ADDM

Automatic Database Diagnostic Monitor Oracle Database-এর Performance Problem বিশ্লেষণ করতে সাহায্য করে।

Microsoft SQL Server

MSSQL Environment-এ ব্যবহার করা যেতে পারে—

- SQL Server Management Studio (SSMS)
- SQL Server Profiler
- Extended Events
- Performance Monitor

- Query Store
- Dynamic Management Views (DMVs)

বিশেষ করে **Query Store** Query Performance Analysis-এর জন্য অত্যন্ত কার্যকর।

PostgreSQL

PostgreSQL-এর ক্ষেত্রে ব্যবহার করা যেতে পারে—

- pg_stat_activity
- pg_stat_database
- pg_stat_statements
- PostgreSQL Logs
- pgAdmin

MySQL

MySQL-এর ক্ষেত্রে—

- Performance Schema
- sys Schema
- Slow Query Log
- MySQL Enterprise Monitor
- MySQL Workbench

ব্যবহার করা যেতে পারে।

১৭. Enterprise Monitoring Tools

একটি বড় Enterprise Environment-এ Database-এর পাশাপাশি Server, Network এবং Application-ও Monitor করা দরকার।

ব্যবহার করা যেতে পারে—

- Prometheus
- Grafana
- Zabbix

- Nagios
- Datadog
- New Relic
- Elastic Stack

একটি Enterprise Monitoring Architecture হতে পারে—

```
Database
Server
Network
Application
   |
   ↓
Monitoring Agents
   |
   ↓
Monitoring Platform
   |
   ↓
Dashboard
   |
   ↓
Alert
   |
   ↓
IT / DBA Team
```

১৮. Database Dashboard

একটি Professional Database Monitoring Dashboard-এ গুরুত্বপূর্ণ তথ্য এক জায়গায় দেখা উচিত।

উদাহরণ:

```
┌──────────────────────────────────────────┐
│      DATABASE MONITORING                 │
├──────────────────────────────────────────┤
│ Database Status     : UP                 │
│ CPU Usage           : 62%                │
│ Memory Usage        : 71%                │
│ Storage Usage       : 78%                │
│ Connections         : 420                │
│ Blocking Sessions   : 2                  │
│ Deadlocks           : 0                  │
│ Long Queries        : 3                  │
│ Backup Status       : SUCCESS            │
│ Replication Lag     : 2 sec              │
└──────────────────────────────────────────┘
```

এতে DBA দ্রুত Database-এর Overall Condition বুঝতে পারেন।

১৯. Automated Database Health Check

Professional Environment-এ প্রতিদিন Manual Health Check করার পরিবর্তে Automation ব্যবহার করা ভালো।

একটি Automated Health Check Script—

```
Database
  ↓
Health Check Script
   ├── Database Status
   ├── Tablespace
   ├── CPU
   ├── Memory
   ├── Sessions
   ├── Blocking
   ├── Long Query
   ├── Backup
   └── Replication
        ↓
     Report
        ↓
   Email / Notification
```

২০. Daily Database Monitoring Checklist

প্রতিদিন DBA-এর একটি Checklist থাকা উচিত।

Database

- Database Up/Down Status
- Instance Status
- Listener Status
- Connection Status

Performance

- CPU Usage
- Memory Usage
- Disk I/O
- Wait Events

- Top SQL
- Long Running Query

Transaction

- Blocking Session
- Deadlock
- Long Transaction

Storage

- Disk Space
- Tablespace Usage
- Archive Log Space
- Datafile Status

Backup

- Full Backup
- Incremental Backup
- Archive Log Backup
- Backup Job Status

DR / Replication

- Standby Status
- Replication Status
- Transport Lag
- Apply Lag
- Archive Gap

Security

- Failed Login
- Privilege Changes
- Suspicious Activity
- Audit Log

২১. Database Monitoring Incident Workflow

ধরুন Monitoring System একটি Alert দিল—

CPU Usage = 95%

DBA কীভাবে কাজ করবে?

Alert
↓
Verify Problem
↓
Check CPU-consuming Process
↓
Identify Top SQL
↓
Check Execution Plan
↓
Check Wait Events
↓
Check Blocking
↓
Identify Root Cause
↓
Apply Fix
↓
Monitor Again
↓
Close Incident

অর্থাৎ Monitoring-এর উদ্দেশ্য শুধু Alert তৈরি করা নয়; **Alert থেকে Root Cause পর্যন্ত যাওয়া।**

২২. Monitoring বনাম Performance Tuning

দুটো বিষয় একে অপরের সঙ্গে সম্পর্কিত হলেও একই নয়।

Monitoring	**Performance Tuning**
Database-এর অবস্থা পর্যবেক্ষণ	Performance সমস্যা সমাধান
Problem Detect করে	Root Cause Fix করে
Continuous Process	Problem-driven Process
Dashboard/Alert ব্যবহার করে	Query/Index/Memory ইত্যাদি Optimize করে

সহজভাবে—

Monitoring বলে সমস্যা কোথায়।
Performance Tuning চেষ্টা করে সমস্যাটি কেন হচ্ছে এবং কীভাবে ঠিক করা যায়।

২৩. Proactive Database Monitoring

Professional DBA শুধু Problem হওয়ার পরে কাজ করবেন না।

লক্ষ্য হওয়া উচিত—

Problem হওয়ার আগেই Problem শনাক্ত করা।

উদাহরণ:

আজ Tablespace Usage:

70%

গত সপ্তাহে:

60%

প্রতি সপ্তাহে 10% করে বাড়ছে।

তাহলে DBA অনুমান করতে পারেন যে ভবিষ্যতে Storage Full হতে পারে।

Historical Data
↓
Growth Analysis
↓
Capacity Planning
↓
Problem Beforehand
এটাই **Proactive Monitoring**।

২৪. Predictive Monitoring

Advanced Environment-এ Historical Monitoring Data ব্যবহার করে ভবিষ্যতের সমস্যা অনুমান করা যায়।

উদাহরণ—

Storage Growth
↓
Historical Trend
↓
Forecast
↓

Expected Full Date
↓
Capacity Planning
একইভাবে—

- CPU Growth
- Memory Growth
- Connection Growth
- Database Size Growth
- Transaction Growth

Analyze করা যায়।

২৫. Database Monitoring-এর Professional Architecture

একটি Enterprise Database Environment-এর জন্য একটি পূর্ণাঙ্গ Monitoring Architecture হতে পারে—

```
┌──────────────────────────┐
│        DATABASE          │
└──────────────────────────┘
            │
┌───────────┼───────────┐
↓           ↓           ↓
Health      Performance Security
Monitor     Monitor     Monitor
│           │           │
└───────────┼───────────┘
            ↓
      Monitoring Platform
            │
┌───────────┼───────────┐
↓           ↓           ↓
Dashboard   Alerts      Reports
            │
            ↓
        DBA / IT Team
            │
            ↓
        Investigation
            │
            ↓
        Root Cause
            │
```

↓
Action

অধ্যায়ের সারসংক্ষেপ

Database Monitoring একটি Database-এর Health, Performance, Availability, Security এবং Capacity নিয়মিত পর্যবেক্ষণ করার প্রক্রিয়া।

এই অধ্যায়ে আমরা দেখলাম—

- **Database Health Check** Database-এর Overall Health যাচাই করে।
- **Performance Monitoring** CPU, Memory, I/O, Connection এবং Query Performance পর্যবেক্ষণ করে।
- **Blocking Session** একটি Session-এর কারণে অন্য Session আটকে গেলে তা শনাক্ত করতে সাহায্য করে।
- **Deadlock** একাধিক Transaction-এর মধ্যে Circular Waiting-এর সমস্যা।
- **Long Running Query** অস্বাভাবিকভাবে বেশি সময় নেওয়া Query শনাক্ত করে।
- **Connection Monitoring** Database Connection-এর চাপ বোঝাতে সাহায্য করে।
- **Storage ও Tablespace Monitoring** Space-related সমস্যা প্রতিরোধ করতে সাহায্য করে।
- **Backup Monitoring** Backup সফল হয়েছে কি না নিশ্চিত করতে সাহায্য করে।
- **Replication/Data Guard Monitoring** Primary ও Standby-এর মধ্যে Synchronization পর্যবেক্ষণ করে।
- **Wait Event Monitoring** Database কোথায় সময় ব্যয় করছে তা বুঝতে সাহায্য করে।
- **Top SQL Monitoring** সবচেয়ে বেশি Resource ব্যবহারকারী Query শনাক্ত করতে সাহায্য করে।
- **Alerting** Problem হলে DBA/IT Team-কে দ্রুত জানায়।
- **Dashboard** গুরুত্বপূর্ণ Database Metrics এক জায়গায় দেখায়।
- **Automated Health Check** Manual Monitoring-এর কাজ কমায়।
- **Proactive Monitoring** Problem হওয়ার আগেই সম্ভাব্য সমস্যা শনাক্ত করতে সাহায্য করে।
- **Predictive Monitoring** Historical Data ব্যবহার করে ভবিষ্যতের Capacity বা Performance সমস্যা অনুমান করতে সাহায্য করে।

সবশেষে মনে রাখতে হবে—

একজন Professional DBA শুধু Database চালু আছে কি না তা দেখেন না; তিনি Database কতটা Healthy, কতটা Fast, কতটা Stable, কতটা Secure এবং ভবিষ্যতে কী ধরনের সমস্যা হতে পারে—সেটিও Monitor করেন।

Database Monitoring → Early Detection → Root Cause Analysis → Corrective Action → Stable Database

এটাই একটি Professional Database Monitoring Strategy-এর মূল লক্ষ্য।