ICS 35.080
CCS L77

团 体 标 准

T/GDICA 004-2026

数据质量 监控与预警系统技术要求

（Technical Requirements for Data Quality Monitoring and Early Warning Systems）

2026 - 2 - 4 发布 2026-2 - 5 实施

广东省信息消费协会 发布

目　　次

前　言

本团体标准按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本标准由 广东省工信两化融合发展中心提出。

本文件由广东省信息消费协会归口管理。

本标准起草单位：广东省工信两化融合发展中心、长沙翼游数据科技有限公司、长沙羽之翼管理咨询有限公司、中通服建设有限公司、东莞广量控股集团有限公司、高颂数科（厦门）智能技术有限公司、武汉胜鹏信息技术有限公司、武汉九州羽翼管理咨询有限公司、东莞华南设计创新院、广州智联信息咨询有限公司、金鹏电子信息机器有限公司、广东中设智控科技股份有限公司。

本标准主要起草人：刘毅、袁宏伟、陈昭华、刘贤庆、麦可琪 、侯明军、陈孚、陈坤隽、唐小华、丁明亮、谭彬、周虹、许杰焜、邱国良、万里鹏、张际清、谷冬超、李秀芬、边荣国。

引　　言

随着数字经济成为全球经济增长的新引擎，数据作为关键生产要素，其质量直接关系到数据分析、决策支持、智能应用和价值创造的效能。当前，各行各业普遍面临数据质量不高、数据管理人才匮乏的挑战，严重制约了数字化转型的深度和广度。

本团体标准旨在指导组织建立和实施有效的数据质量监控与预警系统，提升数据质量管理水平，确保数据资产的价值和可靠性。随着数字化转型的深入，数据已成为组织的核心资产，数据质量直接影响业务决策、运营效率和风险控制。本标准的制定将有助于各行业、各领域组织统一数据质量监控与预警的理念和方法，推动数据质量管理工具和技术的标准化应用。

本标准强调了数据质量监控的实时性、自动化和预警机制的有效性，旨在帮助组织及时发现、定位和解决数据质量问题，降低因数据质量问题带来的潜在风险和损失。

数据质量 评估师能力要求与认证规范

1 范围

本标准规定了数据质量监控与预警系统的架构设计、功能要求、监控指标设定、预警机制建立、预警信息处理流程与应急响应机制、监控策略与方法、系统实现与部署以及系统运维与优化等方面的技术要求。

本标准适用于指导各类组织建立、实施和改进数据质量监控与预警系统，提升数据质量管理水平。

本标准阅读对象为数据质量管理人员、数据分析师、IT 技术人员以及任何对数据质量监控与预警感兴趣的相关人员。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25000.10《系统与软件工程 系统与软件质量要求与评价 (SQuaRE) 第 10 部分：系统与软件质量模型》

GB/T 36344—2018《信息技术 数据质量评价指标》

GB/T 37988《信息安全技术 数据安全能力成熟度模型》

ISO/TS 8000-1:2022《数据质量 — 第 1 部分：概述》

ISO 8000-2:2017《数据质量 — 第 2 部分：词汇》

ISO 8000-8:2015《数据质量 — 第 8 部分：信息和数据质量：概念和测量》

ISO 8000-61:2016《数据质量 — 第 61 部分：数据质量管理：过程参考模型》

ISO 8000-63:2019《数据质量 — 第 63 部分：数据质量管理：过程测量》

3 术语和定义

下列术语和定义适用于本文件。

a)数据质量在指定条件下使用时，数据的特性满足明确的和隐含的要求的程度。 [GB/T 36344—2018，定义 2.3]

b)数据质量监控通过持续地收集、分析和评估数据质量相关指标，以识别、定位和报告数据质量问题的过程。

c)数据质量预警基于预设的规则和阈值，在数据质量指标异常或偏离预期时，自动生成并发送通知，提示相关人员采取行动的机制。

d)数据源：数据产生的原始地点或数据存储的系统，例如数据库、文件系统、应用程序接口等。

e)监控指标：用于衡量数据质量某一特定维度的量化标准或度量值。

f)阈值：预设的、用于判断数据质量指标是否异常或触发预警的界限值。

g)实时监控：以最小的延迟对数据质量指标进行连续或近连续的观测和评估，以迅速发现和响应数据质量问题。

h)自动化工具：能够自动执行数据质量监控、分析、报告和部分预警处理任务的软件或系统。

i)应急响应机制：针对数据质量问题预警事件，组织预先制定的一系列旨在快速响应、止损、恢复和解决问题的程序和措施。

j)数据质量维度：衡量数据质量的各个方面的特征，例如规范性、完整性、准确性、一致性、时效性、可访问性等。 [GB/T 36344—2018，部分参考 5.1，5.2，5.3，5.4，5.5，5.6，5.7]

k)元数据：关于数据或数据元素的数据（可能包括其数据描述），以及关于数据拥有权、存取路径、访问权和数据易变性的数据。 [GB/T 36344—2018，定义 2.2]。

4 数据质量监控与预警系统概述

4.1 系统目标

数据质量监控与预警系统应实现以下目标：

a)实时洞察与早期预警： 及时发现并预警数据质量问题，避免问题扩大化对业务造成影响。

b)提升数据信任度： 通过持续监控和问题解决，增强用户对数据资产的信心。

c)优化数据管理效率： 自动化监控流程，减少人工干预，提高数据质量管理工作的效率。

d)支持决策制定： 提供准确、一致的数据，为业务分析和决策提供可靠基础。

e)促进数据治理： 发现数据质量根源问题，驱动数据治理策略的优化和落地。

f)量化数据质量： 提供可量化的数据质量指标，便于评估和改进数据质量水平。

4.2 系统原则

数据质量监控与预警系统应遵循以下原则：

a)全面性： 覆盖关键业务域、核心数据资产及数据生命周期中的重要环节。

b)及时性： 能够以实时或近实时的方式进行数据质量监控，确保预警信息的时效性。

c)准确性： 监控结果和预警信息应准确无误，避免误报和漏报。

d)可扩展性： 能够适应数据量、数据类型和监控需求的变化，支持新的监控规则和指标的扩展。

e)易用性： 系统界面直观友好，操作简便，便于不同角色的用户进行配置、查看和管理。

f)自动化： 尽可能利用自动化工具完成数据采集、处理、监控、分析和预警通知等任务。

g)可追溯性： 预警信息和问题解决过程应可追溯，便于问题分析和责任定位。

h)安全性： 确保监控过程中的数据安全和隐私保护，符合相关法律法规要求。

4.3 系统组成

数据质量监控与预警系统通常由以下核心组成部分构成：

a)数据采集与接入模块： 负责从各类数据源获取原始数据或元数据。

b)数据处理与转换模块： 对采集到的数据进行清洗、转换、标准化等操作，为质量评估做准备。

c)数据质量监控引擎： 根据预设的规则和指标，对数据进行实时或批量的质量检查。

d)监控指标管理模块： 提供监控指标的定义、配置、计算逻辑管理功能。

e)预警规则管理模块： 提供预警规则的定义、阈值设定、预警级别和通知方式配置功能。

f)预警与通知模块： 在触发预警条件时，生成并发送预警信息到指定接收人或系统。

g)报表与可视化模块： 以图表、仪表盘等形式展示数据质量现状、趋势和预警信息。

h)系统管理与配置模块： 提供用户权限、系统参数、日志管理等功能。

i)工作流程与问题处理模块： 支持预警事件的流转、分配、处理和闭环管理。

5 系统架构设计

5.1 总体架构

数据质量监控与预警系统应采用可扩展、高可用、模块化的架构设计，通常可参考以下分层架构：

a)数据源层： 包含组织内部的各种数据存储系统，如关系型数据库、非关系型数据库、数据湖、数据仓库、文件系统、API 接口等，以及外部数据源。

b)数据接入层： 负责与数据源进行连接，实现数据的采集、抽取和传输。可采用多种技术，如 ETL 工具、数据集成平台、消息队列等。

c)数据处理层： 对接入的数据进行预处理，包括数据清洗、格式转换、元数据解析、脱敏等操作，为数据质量评估提供标准化的输入。

d)数据质量引擎层：

1)监控规则管理： 定义和管理数据质量规则、监控指标和阈值。

2)质量评估引擎： 执行数据质量规则，计算各项监控指标，识别数据质量问题。

3)预警触发器： 根据指标评估结果和预设阈值，触发预警事件。

e)数据存储层： 存储元数据、监控规则、历史监控结果、预警日志、问题处理记录等信息。可使用关系型数据库、时序数据库或文档数据库等。

f)服务接口层： 对外提供数据质量查询、规则配置、预警管理等 API 接口，支持与其他系统集成。

g)应用展现层： 面向用户提供数据质量管理平台，包括数据质量仪表盘、监控报告、预警信息列表、规则配置界面、问题处理工作台等功能。

h)通知与集成层： 负责预警信息的发送（如邮件、短信、即时通讯工具）和与其他管理系统（如工单系统、告警系统）的集成。

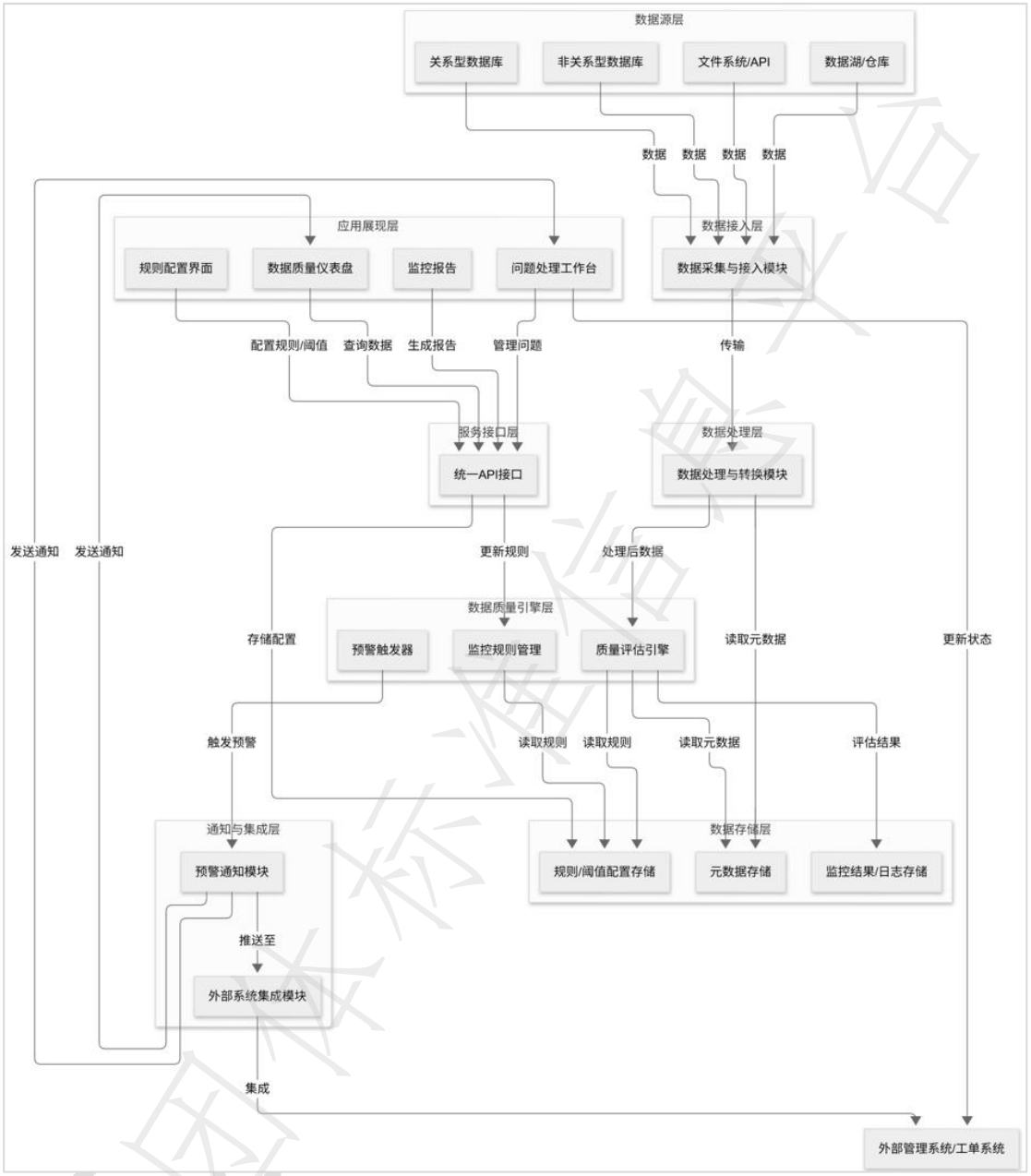

图 1 数据质量监控与预警系统架构图

5.2 核心模块设计

5.2.1 数据采集与接入模块

a) 多源接入：支持主流数据库（如 Oracle, MySQL, SQL Server, PostgreSQL, MongoDB）、文件系统（如 HDFS, S3）、消息队列（如 Kafka, RabbitMQ）、Web API 等多种数据源类型。

b) 数据同步机制：支持全量同步和增量同步，可根据数据源特性和业务需求选择实时同步、定时批量同步等方式。

c) 元数据获取：能够自动或手动获取数据源的元数据，包括表结构、字段定义、数据类型、约束条件等，用于规则配置和问题定位。

d) 数据脱敏：在数据采集过程中，对于敏感数据应支持按需进行脱敏处理，确保数据安全。

5.2.2 数据处理与转换模块

a) 数据清洗： 支持缺失值填充、异常值处理、重复值剔除、格式统一等清洗操作。

b) 数据转换： 支持数据类型转换、编码转换、单位转换等。

c) 标准化： 根据预设的数据标准，对数据进行标准化处理，例如地址标准化、名称标准化等。

d) 数据溯源： 记录数据从源头到目标系统的处理路径和转换规则，支持数据血缘分析。

5.2.3 数据质量监控功能

a) 规则引擎： 提供灵活的规则定义语言或配置界面，支持基于 SQL、正则表达式、脚本等方式定义数据质量规则。

b) 多维度监控： 支持对数据规范性、完整性、准确性、一致性、时效性、可访问性等多个维度进行监控。

c) 实时与批量监控：具备实时数据流监控和周期性批量数据扫描的能力，满足不同场景下的监控需求。

d) 监控任务管理： 提供监控任务的创建、编辑、启停、调度和查看功能。

e) 异常数据识别： 能够识别不符合规则的异常数据，并记录其详细信息。

5.2.4 数据质量预警功能

a) 预警规则配置： 允许用户配置预警触发条件（例如，不合格率超过阈值、异常数据量超过上限等）。

b) 预警级别： 支持定义多级预警（如：一般、重要、紧急），不同级别可关联不同的处理流程和通知方式。

c) 预警通知： 支持邮件、短信、即时通讯工具、系统内部通知、API 回调等多种预警通知方式。

d) 预警收敛与抑制： 具备对短期内大量重复或相关预警进行收敛和抑制的能力，减少“告警风暴”。

5.2.5 报表与可视化功能

a) 数据质量仪表盘： 提供直观的数据质量总览，包括整体合格率、各维度得分、趋势变化、TOP N 问题等。

b) 详细报告： 生成详细的数据质量评估报告，包含具体的问题数据、不合格原因、建议修复方案等。

c) 趋势分析： 展示数据质量指标的历史趋势，支持时间段选择和对比分析。

d) 自定义报表： 提供灵活的报表定制功能，满足不同用户的个性化需求。

5.2.6 系统管理与配置功能

a) 用户与权限管理： 支持多用户、多角色管理，根据角色分配不同的操作权限和数据访问权限。

b) 日志管理： 记录系统运行日志、操作日志、预警日志等，便于审计和问题排查。

c) 参数配置： 提供系统级和模块级的参数配置功能，如通知服务配置、数据保留策略等。

d) 元数据管理： 提供对系统内部使用的元数据的管理和维护能力。

5.2.7 自动化与集成能力

a) API 接口： 提供标准化的 API 接口，支持与数据治理平台、数据开发平台、运维监控平台、工单系统等进行集成。

b) 工作流集成： 能够将数据质量问题发现后的处理流程与现有的工作流管理系统进行对接。

c) 脚本扩展： 支持用户通过自定义脚本扩展监控规则和处理逻辑。

5.3 部署要求

a) 部署方式： 支持单机部署、集群部署或云原生部署，可根据组织规模和业务需求进行选择。

b) 资源要求： 应明确对 CPU、内存、存储、网络等硬件资源的要求，并具备一定的可伸缩性。

c) 高可用性： 关键组件应支持高可用部署，避免单点故障。

d) 灾备能力： 具备数据备份和恢复机制，以及异地灾备能力。

e) 环境兼容性： 兼容主流操作系统和数据库环境。

6 功能要求

6.1 数据采集与接入功能

a)系统应支持从关系型数据库（如 Oracle、MySQL、SQL Server、PostgreSQL）、非关系型数据库（如 MongoDB、Redis）、数据仓库（如 Hive、ClickHouse）、数据湖（如 HDFS、S3）、文件系统（如 FTP、SFTP）、API 接口、消息队列（如 Kafka、RabbitMQ）等多种类型的数据源进行数据采集和接入。

b)系统应支持全量数据采集和增量数据采集两种模式，并可根据业务需求配置采集频率（如实时、分钟级、小时级、日级）。

c)系统应具备自动发现数据源元数据（如表结构、字段、数据类型、主键、外键、索引等）的能力，并支持手动导入和编辑元数据。

d)系统应提供数据源连接配置管理功能，包括连接信息加密存储、连接测试等。

e)对于敏感数据，系统应提供数据脱敏功能，支持多种脱敏算法（如替换、遮蔽、加密、哈希），并可在数据采集或处理阶段进行配置和应用。

6.2 数据处理与转换功能

6.2.1 系统数据清洗功能要求：

a) 缺失值检测与处理：识别空值、NULL 值，并支持自动填充（如默认值、均值、中位数）、删除或标记。

b) 异常值检测与处理：识别超出合理范围或不符合业务逻辑的数据，并支持标记、修正或删除。

c) 重复值检测与处理：识别完全重复或部分重复的记录，并支持去重或标记。

d) 格式规范化：统一数据格式，如日期格式、编码格式、大小写转换等。

6.2.2 系统数据转换功能要求：

a) 数据类型转换：例如字符串转数字、日期格式转换等。

b) 编码转换：支持不同字符集之间的转换。

c) 单位转换：支持不同计量单位之间的转换。

d)系统应支持数据标准化功能，可根据预定义的数据标准对特定字段进行标准化处理。

e)系统应提供数据溯源能力，记录数据的来源、流向、转换过程和处理规则，便于问题追溯和影响分析。

6.3 数据质量监控功能

a)系统应具备强大的规则引擎，支持用户通过图形化界面、SQL 语句或自定义脚本等方式灵活定义数据质量规则。

b)系统应支持对 GB/T 36344—2018 中定义的规范性、完整性、准确性、一致性、时效性、可访问性等数据质量维度进行监控。

6.3.1 系统应支持对以下类型的监控指标进行计算和评估：

a) 规范性指标： 数据符合数据标准、数据模型、业务规则的程度（例如，字段值是否在合法范围内、数据类型是否正确、数据格式是否一致）。

b) 完整性指标： 数据元素或记录被赋予数值的程度（例如，必填字段的填充率、记录的完整性）。

c) 准确性指标： 数据准确表示其所描述的真实实体的程度（例如，数据内容正确性、数据唯一性、脏

数据出现率）。

d) 一致性指标： 数据在不同位置、不同系统或不同应用中使用时保持一致的程度（例如，相同数据一致性、关联数据一致性）。

e) 时效性指标：数据在时间变化中的正确程度和及时性（例如，数据更新频率、数据延迟、时序关系）。

f) 可访问性指标： 数据在需要时可被获取和使用的程度（例如，数据可访问性、数据可用性）。

g)系统应支持配置监控任务的调度策略，包括定时执行、事件触发执行等。

h)系统应能记录每次监控任务的执行结果，包括监控开始时间、结束时间、扫描数据量、异常数据量、合格率等，并提供历史查询功能。

i)系统应支持异常数据的详细信息记录，包括异常数据行、异常字段、错误码、错误描述等，并支持导出。

6.4 数据质量预警功能

a)系统应提供灵活的预警规则配置功能，支持基于监控指标（如不合格率、异常数据量、波动幅度）和预警级别设定触发条件。

b)系统应支持多种预警级别定义（如信息、警告、错误、紧急），不同级别可关联不同的通知方式和处理流程。

c)系统应支持多种预警通知方式，包括邮件、短信、微信/钉钉等即时通讯工具、系统内部通知、API回调等。

d)系统应具备预警通知模板管理功能，支持自定义预警信息内容。

e)系统应提供预警收敛和抑制机制，例如在短时间内对同一类或相同的数据质量问题只发送一次预警，或根据重要性进行优先级排序。

f)系统应记录所有触发的预警事件，包括预警时间、预警内容、预警级别、相关数据源、监控规则、通知状态等，并提供查询和统计功能。

6.5 报表与可视化功能

6.5.1 系统应提供交互式的数据质量仪表盘，直观展示当前数据质量概况，包括：

a) 整体数据质量得分或合格率。

b) 各数据质量维度的得分或合格率。

c) 历史数据质量趋势图。

d) TOP N 数据质量问题（按类型、严重程度、发生频率等）。

e) 当前活跃的预警信息列表。

6.5.2 系统应支持生成可定制的数据质量报告，报告内容应包括但不限于：

a) 报告周期内的数据质量总体情况。

b) 各监控指标的详细评估结果。

c) 发现的异常数据列表及详情。

d) 预警事件统计与分析。

e) 改进建议与行动计划。

f)系统应提供数据质量趋势分析功能，支持用户选择不同的时间范围、数据源、监控指标进行对比分析。

g)系统应支持数据质量问题的可视化，如通过图表展示异常数据分布、问题严重程度等。

6.6 系统管理与配置功能

6.6.1 用户与权限管理：

a)支持创建、编辑、删除用户账户。

b)支持定义不同角色（如系统管理员、数据质量经理、数据分析师、开发人员），并为角色分配细粒度的操作权限和数据访问权限。

c)支持用户与角色的关联管理。

6.6.2 日志管理：

a)记录系统运行日志、用户操作日志、数据采集日志、监控执行日志和预警通知日志。

b)提供日志查询、筛选、导出功能。

c)支持日志级别配置和日志自动清理策略。

6.6.3 系统参数配置：

a)提供系统全局参数配置，如数据保留策略、并发任务数限制、邮件服务器配置、短信网关配置等。

b)支持配置数据质量规则模板和预警通知模板。

6.6.4 元数据管理：

a)提供对系统内存储的元数据进行查看、编辑、导入、导出等管理功能。

b)支持元数据版本管理和变更审计。

6.7 自动化与集成能力

系统应提供开放的 API 接口，支持外部系统调用数据质量监控任务、查询监控结果、获取预警信息等。

系统应支持与企业内部其他数据管理或治理平台集成，例如：

a) 数据资产目录： 同步数据质量指标和评估结果到数据资产目录，丰富数据资产信息。

b) 数据开发平台： 在数据开发流程中嵌入数据质量检查环节。

c) 工单系统： 自动将数据质量预警转化为工单，驱动问题处理流程。

d) 运维监控系统： 将数据质量预警信息推送至统一的运维监控平台。

系统应支持自定义脚本或插件机制，允许用户扩展新的数据源连接、数据处理逻辑、监控规则和通知方式。

7 监控指标设定

7.1 数据质量维度

数据质量监控指标应覆盖数据质量的六个主要维度，参考 GB/T 36344—2018：

a) 规范性： 数据符合数据标准、数据模型、业务规则、元数据或权威参考数据的程度。

b) 完整性： 按照数据规则要求，数据元素被赋予数值的程度。

c) 准确性： 数据准确表示其所描述的真实实体（实际对象）真实值的程度。

d) 一致性： 数据与其他特定上下文中使用的数据无矛盾的程度。

e) 时效性： 数据在时间变化中的正确程度。

f) 可访问性： 数据能被访问的程度。

7.2 监控指标类型

针对上述数据质量维度，监控指标通常分为以下几类：

a)基于数量的指标： 如空值数量、重复记录数量、异常值数量等。

b)基于比率的指标： 如空值率、重复率、合格率、准确率等，通常以百分比表示。

c)基于时间间隔的指标： 如数据更新频率、数据延迟时间等。

d)基于规则符合度的指标： 如符合特定格式的记录数、符合业务逻辑的记录数等。

e) 基于统计分布的指标： 如字段值的均值、中位数、标准差、分布范围等，用于检测数据偏离或异常。

7.3 指标计算与评估方法

7.3.1 监控指标的计算应明确定义，包括计算公式、涉及的数据字段、统计范围和时间窗口。

7.3.2 指标计算应支持 SQL 查询、表达式计算、编程脚本等多种方式。

7.3.3 对于复杂指标，应提供详细的计算逻辑说明。

7.3.4 指标评估应基于预设的阈值进行，将计算结果与阈值进行比较，判断数据质量是否合格或是否触发预警。

7.4 阈值设定与动态调整

7.4.1 阈值设定：

a)应支持为每个监控指标设定一个或多个阈值。

b)阈值可根据业务重要性、数据源特性、历史数据质量情况等因素进行设定。

c)可设置单层阈值（如：合格率低于 90%）或多层阈值（如：合格率低于 90%为警告，低于 80%为紧急）。

d)阈值类型可包括绝对值、百分比、范围值等。

7.4.2 阈值动态调整：

a)系统应支持根据历史数据趋势、业务需求变化或通过机器学习算法自动推荐阈值。

b)应提供用户界面，允许数据质量管理人员对阈值进行手动调整和优化。

c) 阈值调整应有版本记录和审计功能。

8 预警机制建立

8.1 预警规则定义

预警规则应清晰、可量化，并与具体的监控指标关联。预警规则的定义应包括以下要素：

a) 规则名称： 描述预警规则的唯一标识。

b) 关联监控指标： 指定触发该预警的监控指标。

c) 触发条件： 定义当监控指标满足何种条件时触发预警（如：指标值 < 阈值，指标值 > 阈值，指标值在某个范围外）。

d) 预警级别： 指定该预警的严重程度。

e) 通知对象： 指定接收预警通知的人员或角色。

f) 通知方式： 指定预警通知的发送渠道。

g) 生效时间/周期： 定义预警规则的有效时间范围或触发周期。

h) 预警内容模板： 自定义预警通知的具体信息。

8.2 预警级别划分

系统应支持多级预警划分，并建议至少包含以下级别：

a)信息（Informational）： 表示轻微的数据质量偏差，通常不影响业务，但值得关注，可能预示潜在问题。例如，某个非关键字段的空值率略有上升。

b)警告（Warning）： 表示数据质量存在明显问题，可能对业务造成一定影响，需要关注和处理。例如，

关键字段的空值率超过了警告阈值。

c)错误（Error）： 表示严重的数据质量问题，已对业务造成较大影响，需要立即介入处理。例如，核心业务数据的准确率大幅下降。

d)d) 紧急（Critical）： 表示数据质量出现灾难性问题，可能导致业务中断或重大损失，需要最高优先级响应。例如，关键数据表无法访问或数据被大面积破坏。

8.3 预警触发条件

预警触发条件应支持基于以下逻辑进行配置：

a)指标阈值告警： 当某个监控指标的计算结果超过或低于预设的绝对阈值、百分比阈值时触发。

b)趋势告警： 当监控指标的趋势发生异常变化时触发，例如连续 N 次指标下降、指标波动幅度过大等。

c)基线偏离告警： 当监控指标与历史基线值或预期值存在显著偏离时触发。

d)复合条件告警： 支持多个指标或多个规则组合，通过逻辑运算（如 AND、OR）形成更复杂的触发条件。

e) 数据量告警： 当异常数据量达到一定数量或比例时触发。

8.4 预警通知方式

预警通知方式应多样化，确保预警信息能及时触达相关人员：

a)邮件通知： 通过邮件系统发送预警信息。

b)短信通知： 通过短信网关发送预警信息，适用于紧急预警。

c)即时通讯工具通知： 集成企业微信、钉钉、Slack 等即时通讯工具进行通知。

d)系统内部通知： 在数据质量监控系统内部提供消息通知或待办事项。

e)API 回调/Webhook： 将预警信息通过 API 接口推送给第三方系统（如工单系统、运维监控平台）。

f) 可视化界面告警： 在系统仪表盘上醒目展示活跃预警信息。

9 预警信息处理流程与应急响应机制

9.1 预警信息接收与确认

预警信息发送后，系统应确保通知的可靠送达。预警接收人应及时确认预警信息，避免遗漏。系统可提供确认功能，并记录确认时间、确认人等信息。对于紧急预警，应采用多渠道、多轮次的通知方式，确保关键人员及时响应。

9.2 预警信息分析与定级

9.2.1 收到预警后，相关人员应立即对预警信息进行分析，包括：

a) 问题定位： 识别具体的数据源、表、字段以及受影响的数据范围。

b) 影响评估： 评估数据质量问题对业务运营、决策和下游系统的潜在影响。

c) 根因分析： 初步判断数据质量问题的可能原因（如数据源错误、ETL 过程问题、业务系统 BUG、数据录入错误等）。

根据问题影响程度和潜在风险，对预警事件进行再次定级，以指导后续处理的优先级。

9.3 应急响应流程

组织应建立标准化的数据质量问题应急响应流程，确保在发现数据质量问题时能够迅速响应并采取措施，该流程通常包括：

a)预警触发： 数据质量监控系统检测到异常并触发预警。

b)预警通知： 系统通过配置的通知方式将预警信息发送给相关人员。

c)初步分析与定级： 值班人员或数据质量管理人员接收预警并进行初步分析，确认问题性质和影响范围，并重新定级（如必要）。

d)事件派发： 将数据质量问题作为事件记录，并根据预警级别和问题类型，自动或手动派发给相应的数据所有者、IT 运维人员、数据开发团队或业务负责人。

e)问题诊断与修复：

1)根因查找： 负责处理团队深入分析问题，定位数据质量问题的根本原因。

2)数据修复： 针对受损数据，制定修复方案，并执行数据修正或回溯操作。修复过程应记录详细日志。

3)流程优化： 如果问题是由于数据生产流程、ETL 流程或业务系统缺陷引起的，则需对相关流程进行优化或系统升级。

f)验证与回归测试： 问题修复后，应重新运行相关监控规则或进行数据质量验证，确保问题已解决且未引入新的问题。

g)信息同步与沟通： 在整个应急响应过程中，应及时向相关利益方（包括受影响的业务部门、管理层）同步问题进展、影响范围和预计恢复时间。

h) 总结与复盘： 问题解决后，进行复盘总结，分析问题发生的原因、处理过程中的经验教训，并更新相关文档和流程。

9.4 问题解决与闭环管理

a)系统应支持对数据质量问题进行全生命周期管理，包括问题的创建、分配、处理、状态更新和关闭。

b)系统应提供工单管理功能，将数据质量预警自动转化为工单，并支持工单的流转、处理人指派、处理记录、附件上传等。

c)对于已修复的数据质量问题，应及时更新其状态为“已解决”，并在系统中进行记录。

d)应定期对已解决的数据质量问题进行效果评估，确保问题得到彻底解决，并防止再次发生。

e)预警事件应实现闭环管理，从发现到解决、验证、复盘，所有环节均应有记录和跟踪。

10 监控策略与方法

10.1 实时监控

a)适用场景： 适用于对时效性要求高、对业务影响大的核心数据，如交易数据、用户行为数据等。

b)技术要求： 系统应支持流式数据处理技术（如 Kafka Stream、Flink、Spark Streaming），能够对持续产生的数据流进行实时质量检查。

c)监控内容： 实时监控主要关注数据流的完整性（如数据丢包）、及时性（如数据延迟）、格式规范性等快速变化且影响业务连续性的指标。

d)预警响应： 实时监控发现的问题应立即触发高优先级预警，并启动快速响应机制。

10.2 定期审查与审计

a)适用场景： 适用于数据量大、业务复杂、不适合频繁实时监控的非核心数据，或作为实时监控的补充。

b)审查周期： 可根据数据重要性、变化频率和业务需求，设定每日、每周、每月或每季度等审查周期。

c)审查内容： 定期审查应包括全面的数据质量评估，覆盖所有数据质量维度，特别是准确性、一致性等需要交叉验证或复杂计算的指标。

10.2.1 审计要求：

a)应定期对数据质量管理流程、监控规则、预警机制的有效性进行审计。

b)审计应评估数据质量问题解决的效率和效果。

c)审计结果应形成报告，并作为数据质量改进的重要依据。

d)自动化工具： 应充分利用自动化工具执行定期的数据质量扫描和报告生成，提高审查效率。

10.3 监控报告与分析

a)报告内容： 监控报告应包含数据质量的整体概况、各维度指标表现、问题趋势、根因分析、改进建议以及行动计划。

b)报告周期： 可根据管理需求，生成日报、周报、月报、季度报告或年度报告。

c)数据可视化： 报告应采用图表、仪表盘等可视化方式展示数据质量状况，便于理解和分析。

d)深度分析： 除了常规报告，系统应支持对特定数据质量问题进行下钻分析，追溯问题根源，评估影响范围，并提供数据修复建议。

e)知识积累： 通过对监控报告和问题分析的总结，积累数据质量问题处理的知识库和最佳实践，持续优化数据质量管理策略。

11 系统实现与部署

11.1 技术选型

系统实现时应考虑以下技术选型：

a)编程语言： 建议采用成熟、生态系统丰富、性能优越的编程语言，如 Java、Python、Go 等。

b)数据存储：

1)关系型数据库： 用于存储元数据、规则配置、系统日志、用户信息等结构化数据（如 MySQL、PostgreSQL）。

2)时序数据库： 适用于存储大量的监控指标历史数据，便于趋势分析（如 InfluxDB、Prometheus）。

3)非关系型数据库： 适用于存储异常数据详情、复杂预警信息等半结构化或非结构化数据（如 MongoDB、Elasticsearch）。

c)数据处理框架：

1)批处理： 适用于定期审查和批量数据质量评估（如 Apache Spark、Apache Flink）。

2)流处理： 适用于实时监控和高时效性要求（如 Apache Flink、Apache Kafka Streams）。

d)消息队列： 用于解耦系统组件，实现异步通信和削峰填谷（如 Apache Kafka、RabbitMQ）。

e)前端技术： 采用现代前端框架和库（如 React、Vue.js、Angular）以提供良好的用户体验。

f) 可视化工具： 集成或使用专业的可视化库（如 ECharts、Grafana）来展示数据质量报告和仪表盘。

11.2 数据接口与集成要求

a)系统应提供基于 RESTful API 或 gRPC 等标准协议的接口，实现与外部系统（如数据治理平台、数据开发平台、业务系统）的无缝集成。

b)接口设计应遵循安全性原则，采用认证、授权机制（如 OAuth2.0、API Key）保护接口访问。

c)接口文档应清晰完整，包含接口地址、请求方法、参数说明、返回示例、错误码等信息。

d)对于数据量较大的数据交换，可考虑采用文件传输或消息队列进行异步集成。

11.3 安全性要求

11.3.1 数据安全：

a)数据存储应加密，特别是敏感数据。

b)数据传输应采用加密协议（如 HTTPS、SSL/TLS）。

c)应对访问数据源的凭证进行加密存储和安全管理。

11.3.2 系统访问安全：

a)提供基于角色的访问控制（RBAC），严格控制用户对系统功能和数据的访问权限。

b)支持用户认证，可集成企业统一身份认证系统。

c)应记录所有用户操作日志，便于审计和追踪。

11.3.3 网络安全：

a)系统部署应遵循网络安全最佳实践，采取防火墙、入侵检测、DDoS 防护等措施。

b)隔离敏感数据处理环境。

11.3.4 漏洞管理：

定期进行安全漏洞扫描和渗透测试，及时修复发现的漏洞。

11.4 性能要求

a)响应时间：

1)数据质量监控任务的执行时间应满足业务需求的时效性。

2)系统界面响应时间、报表生成时间应在可接受范围内。

b)并发能力： 系统应能够支持一定数量的并发监控任务和用户访问，且性能不受显著影响。

c)吞吐量： 系统应具备处理大量数据、高速数据流的能力，能够满足组织数据增长的需求。

d)可伸缩性： 系统架构应支持水平扩展和垂直扩展，以便根据业务量和数据量的增长动态调整资源。

e)资源利用率： 系统应高效利用计算、存储和网络资源，避免资源浪费。

12 系统运维与优化

12.1 系统维护

a)监控与告警： 对系统自身的运行状态（如 CPU 利用率、内存使用、磁盘空间、网络 I/O、服务可用性、进程状态）进行实时监控，并配置相应的告警机制。

b)日志管理： 定期审查系统日志，及时发现并解决潜在问题。对历史日志进行归档和清理。

c)数据备份与恢复： 定期对系统的配置数据、元数据、核心监控结果等进行备份，并定期测试数据恢复流程，确保数据安全和业务连续性。

d)版本升级与补丁管理： 关注系统所依赖的第三方组件和底层技术的安全漏洞和性能改进，及时进行版本升级和补丁安装。

e)故障管理： 建立完善的故障管理流程，包括故障上报、诊断、修复、验证和归档。

12.2 性能优化

a)优化数据采集： 优化数据采集策略，如利用数据源的增量同步机制、并行采集、数据压缩等。

b)优化数据处理： 采用高效的算法和数据结构，对数据清洗、转换和质量评估逻辑进行优化，利用分布式计算框架提升处理效率。

c)索引优化： 对数据库中的核心表建立合适的索引，提高查询效率。 12.2.4 资源配置调整： 根据系统运行情况，动态调整计算、存储资源的配置，如增加节点、扩充内存、优化磁盘 I/O 等。

d)缓存机制： 对频繁访问的数据或计算结果引入缓存机制，减少重复计算和数据库访问。

12.3 持续改进

a)定期复盘： 定期组织数据质量管理团队、业务部门和 IT 团队进行复盘会议，总结数据质量问题和预警处理经验，分析不足。

b)规则和指标优化： 根据业务需求变化、数据质量趋势和复盘结果，持续优化监控规则和指标的定义、阈值设定。

c)系统功能迭代： 收集用户反馈，定期评估系统功能，持续进行功能迭代和新特性开发。

d)技术更新： 关注数据质量监控与预警领域的新技术、新方法，适时引入以提升系统能力。

e)知识沉淀： 建立数据质量知识库，沉淀常见问题解决方案、最佳实践和操作手册，促进知识共享和团队能力提升。

附录 A（资料性附录）数据质量维度示例

本附录提供数据质量维度及其子维度示例，供组织在设定监控指标时参考。具体的维度和子维度可根据组织的业务特点和数据特性进行调整和扩展。

A.1 规范性（Validity/Conformity）

A.1.1 格式规范性：数据值是否符合预期的格式要求（如日期格式 YYYY-MM-DD，手机号 11 位数字）。
A.1.2 类型规范性： 数据值的数据类型是否与元数据定义一致（如数字型字段是否包含非数字字符）。
A.1.3 值域规范性： 数据值是否在合法的值域范围内（如性别字段只允许“男”、“女”）。
A.1.4 业务规则规范性： 数据值是否符合业务逻辑规则（如订单金额不能为负数）。
A.1.5 引用完整性： 外键值是否在主键表中存在。

A.2 完整性（Completeness）

A.2.1 字段完整性： 必填字段的填充率，即非空值的比例。
A.2.2 记录完整性： 记录中关键字段的填充率或所有必填字段都已填充的记录比例。
A.2.3 范围完整性： 在给定时间段内，是否存在应有的数据但未被记录的情况。

A.3 准确性（Accuracy）

A.3.1 数据内容准确性： 数据值是否真实、正确地反映了客观实体或事件。
A.3.2 数据唯一性： 关键字段或主键是否存在重复记录。
A.3.3 脏数据出现率： 存在明显错误、无法使用或干扰分析的数据比例。
A.3.4 一致性准确性： 数据与权威参考数据源或标准数据的符合程度。

A.4 一致性（Consistency）

A.4.1 相同数据一致性： 同一实体在不同数据源、不同系统或不同时间点上的相同属性值是否保持一致。
A.4.2 关联数据一致性： 相互关联的数据之间是否符合逻辑关系（如订单总金额等于商品金额之和）。
A.4.3 跨系统一致性： 不同业务系统或应用中相同业务含义的数据是否一致。

A.5 时效性（Timeliness）

A.5.1 数据更新频率： 数据更新是否符合业务要求，是否有及时更新。
A.5.2 数据延迟： 数据从产生到可用所需的时间是否满足时效性要求。
A.5.3 时序关系： 序列数据之间的时间顺序是否正确，是否存在乱序或跳序。

A.6 可访问性（Accessibility）

A.6.1 数据可访问性： 数据在需要时是否能够被授权用户获取，且获取过程是否便捷。
A.6.2 数据可用性： 数据是否在设定的有效生存周期内可供使用，且系统稳定可靠，无故障。

附录 B（资料性附录）监控指标示例

本附录提供具体的监控指标示例，供组织在制定监控策略时参考。这些指标可以根据实际情况进行选择、组合或扩展。

B.1 规范性指标示例

B.1.1 格式不规范率：(不符合格式要求的数据行数 / 总数据行数) * 100%。例如，手机号码字段非 11 位数字的比例。

B.1.2 类型不匹配率：(数据类型与元数据定义不符的字段数 / 总字段数) * 100%。

B.1.3 值域超范围率：(数据值超出预定义合法范围的数据行数 / 总数据行数) * 100%。

B.1.4 业务规则不符率：(不符合特定业务逻辑规则的数据行数 / 总数据行数) * 100%。例如，订单金额为负数的订单比例。

B.1.5 引用不完整率：(外键值在主键表中不存在的记录数 / 总记录数) * 100%。

B.2 完整性指标示例

B.2.1 必填字段空值率：(必填字段为空的记录数 / 总记录数) * 100%。

B.2.2 字段填充率：(某字段非空值的记录数 / 总记录数) * 100%。

B.2.3 记录完整率：(所有必填字段均有值的记录数 / 总记录数) * 100%。

B.3 准确性指标示例

B.3.1 数据唯一性不合格率：(非唯一的主键或关键字段值数量 / 总记录数) * 100%。

B.3.2 脏数据率：(被识别为脏数据（如乱码、明显错误值）的记录数 / 总记录数) * 100%。

B.3.3 数据正确率：(与权威参考数据源一致的数据行数 / 总数据行数) * 100%。

B.3.4 数据误差率：(与真实值或参考值相比，误差超过允许范围的数据行数 / 总数据行数) * 100%。

B.4 一致性指标示例

B.4.1 相同数据不一致率：(同一实体在不同数据源或字段中值不一致的记录数 / 总记录数) * 100%。

B.4.2 关联数据不一致率：(相互关联字段之间逻辑关系不符的记录数 / 总记录数) * 100%。例如，用户表中的总积分与积分明细表计算所得总积分不一致的比例。

B.4.3 跨系统数据不一致率：(同一业务对象在不同系统中数据值不一致的比例)。

B.5 时效性指标示例

B.5.1 数据更新延迟：(数据从源系统更新到目标系统所需的时间)。单位可以是秒、分钟、小时等。

B.5.2 数据更新频率：(数据在特定时间段内更新的次数或频率)。

B.5.3 数据新鲜度：(数据上次更新时间距当前时间的间隔)。

B.5.4 数据老化率：(数据超过预设新鲜度要求的数据记录数 / 总记录数) * 100%。

B.6 可访问性指标示例

B.6.1 数据可访问性：(授权用户成功访问数据的请求次数 / 总访问请求次数) * 100%。

B.6.2 数据可用性：(数据存储或服务在指定时间段内正常运行的时间 / 总监测时间) * 100%。

附录 C（资料性附录） 预警规则示例

本附录提供一些预警规则的示例，供组织在建立预警机制时参考。实际应用中应根据具体业务场景和数据特性进行细化和定制。

C.1 规范性预警规则示例

C.1.1 手机号码格式异常： 当“用户表.手机号码”字段的格式不符合正则表达式^1[3-9]\d{9}$的记录数超过 10 条，或格式不符率超过 0.1%时，触发警告，通知数据管理员。

C.1.2 订单状态值域非法： 当“订单表.订单状态”字段出现非“待支付”、“已支付”、“已发货”、

“已完成”、“已取消”的值时，触发错误，通知业务负责人和数据开发团队。

C.1.3 数据类型不匹配： 当“产品表.产品价格”字段检测到非数值类型数据时，触发错误，通知 IT 运维和数据开发团队。

C.2 完整性预警规则示例

C.2.1 客户姓名空值率超标： 当“客户表.客户姓名”字段的空值率超过 5%时，触发警告，通知销售部门和数据管理员。

C.2.2 订单关键信息缺失： 当“订单表”中“收货地址”或“联系电话”字段的空值率超过 2%时，触发错误，通知物流和客服部门。

C.3 准确性预警规则示例

C.3.1 用户 ID 重复： 当“用户表.用户 ID”字段检测到重复值时，触发错误，通知数据管理员和开发团队。

C.3.2 商品库存异常：当“商品表.库存数量”字段出现负值时，触发紧急，通知供应链和 IT 运维团队。

C.3.3 地区编码不正确： 当“地址表.地区编码”与国家标准地区编码不符的记录数超过阈值时，触发警告，通知数据管理团队。

C.4 一致性预警规则示例

C.4.1 账户余额不一致： 当“用户账户表.余额”与“交易明细表”汇总计算的账户余额存在差异且差异金额超过 100 元或差异率超过 1%时，触发紧急，通知财务和开发团队。

C.4.2 跨系统客户信息不一致： 当 CRM 系统中的客户名称与 ERP 系统中的客户名称存在明显不一致的记录数超过 N 条时，触发警告，通知业务部门和数据治理团队。

C.5 时效性预警规则示例

C.5.1 数据更新延迟： 当“数据仓库.每日销售表”的更新时间超过每天 02:00 时，触发警告，通知数据开发和运维团队。

C.5.2 实时数据流中断： 当某个实时数据流（如 IOT 设备数据）在过去 10 分钟内没有接收到任何新数据时，触发紧急，通知 IT 运维团队。

C.6 可访问性预警规则示例

C.6.1 核心数据库连接失败： 当数据质量监控系统无法连接到核心业务数据库持续 5 分钟时，触发紧急，通知 IT 运维团队。

C.6.2 数据报告生成失败： 当关键数据质量报告未能按时生成时，触发警告，通知数据管理团队。