

# MDM\_DATA\_DICTIONARY

netxs

2025-12-27

## 人员主数据 (mdm\_identities)

为了解决多系统账号对齐问题，新增 mdm\_identities 作为“黄金记录”索引表。

建议采用 SCD Type 2（保留历史版本），以便追溯。

字段名	类型	约束	描述	示例
global_user_id	UUID	PK	全局唯一标识，集团级唯一身份 ID (OneID)	550e8400-e29b-41d4-a716...
employee_id	String	Unique	集团 HR 系统工号 (核心锚点)	GR10245
full_name	String	Not Null	法律姓名	张三
primary_email	String	Unique	集团官方办公邮箱	zhangsan@group.com
identity_map	JSONB	-	多系统账号映射关系 (Git/Jira/飞书等)。存储各子系统 ID 映射关系 (带 GIN 索引)。这是应对多源异构系统的“黄金解”	<pre>{"gitlab": 12, "jira": "J_01"}</pre> <p>在 PostgreSQL 中使用 JSONB 时，必须说明需建立 GIN 索引，否则千万级数据下的 -&gt;&gt; 键值查询会非常慢。</p>

<b>match_confidence</b>	Float	-	算法匹配置信度 (0.0-1.0)	<b>0.98</b>
<b>is_survivor</b>	Boolean	Default: T	是否为当前生效的“生存者”黄金记录	<b>true</b>
<b>is_active</b>	Boolean	-	账号状态 (在职/离职)	<b>true</b>
<b>updated_at</b>	Timestamp	-	最后更新时间 (用于 dbt 增量更新)。记录该用户最后一次在任意子系统活跃的时间。这对于判断账号是否应被回收 (权限治理) 至关重要	2025-12-21 09:00:00
<b>source_system</b>	String		标记该“生存者记录”的主来源系统 (如 HRMS), 定义数据的权威性来源。	HRMS
<b>sync_version</b>	BigInt/Int		乐观锁版本号。据。	在 CDC 频繁更新 identity 关系时, 防止旧数据覆盖新数

## 2. 组织架构主数据 (mdm\_organizations)

建立全集团的汇报线与成本中心映射，支持指标按部门层级汇总。

建议采用 **SCD Type 2**（保留历史版本），以便追溯“项目负责人变更”或“部门重组”前后的指标归属。

字段名	类型	约束	描述	示例
org_id	String	PK	组织唯一编码（全局 ID）	ORG_FIN_001
org_name	String	Not Null	组织/部门名称	财务部-结算组
parent_org_id	String	FK	父级组织 ID（用于递归树状结构）	ORG_FIN_000
org_level	Integer	-	组织层级（1-集团, 2-事业部, 3-部门）	3
manager_user_id	UUID	FK	部门负责人 ID（关联 mdm_identities）	550e8400-e29b...
cost_center	String	-	财务成本中心代码	CC_7788
is_deleted	Boolean	Default: F	逻辑删除标记	false

**OneID 穿透：**确保所有涉及“人”的字段（如 manager\_user\_id, pm\_user\_id）全部强制使用 mdm\_identities 表中的 global\_user\_id。

3 指标语义定义表 (mdm\_metric\_definitions)

这是“指标字典”的核心，确保全集团计算逻辑一致。

字段名	类型	约束	描述	示例
metric_code	String	PK	指标唯一编码（建议：域_类别_名称）	DORA_MTTR_PROD
metric_name	String	Not Null	指标中文名称	生产环境平均修复时间

<b>domain</b>	String	-	所属业务域（研发/财务/运营/人力）	DEVOPS
<b>metric_type</b>	Enum	-	指标类型（原子/派生/复合）	DERIVED
<b>calculation_logic</b>	Text	-	计算公式/dbt路径。描述核心计算逻辑	marts/pmo/fct_mttr.sql
<b>unit</b>	String	-	物理单位	Hours
<b>aggregate_type</b>	String	-	聚合方式（SUM/AVG/COUNT/MAX/MIN）	AVG
<b>source_model</b>	String	FK	来源模型名称（关联 dbt 模型或表）	marts.pmo.fct_incident
<b>dimension_scope</b>	JSONB	-	允许挂载的维度列表（防范非法下钻）	["dept", "app", "priority"]
<b>is_standard</b>	Boolean	Default: T	是否集团标准指标（涉及口径锁定）	true
<b>business_owner</b>	UUID	FK	业务口径负责人（关联 mdm_identities）	550e8400-e29b...
<b>time_grain</b>	String		支持的时间粒度	Daily, Weekly, Monthly
<b>update_cycle</b>	String		更新频率（实时/天/小时）	T+1
<b>status</b>	String		指标生命周期（发布/草稿/下线）	RELEASED
<b>is_active</b>	Boolean	Default: T	该指标是否在役（已弃用指标设为 F）	true
<b>updated_at</b>	Timestamp	-	最后一次口径变更时间	2025-12-21 23:00:00

#### 4. 实体/资产拓扑主数据 (mdm\_entities\_topology)

定义应用、微服务、代码库等技术资产。在研发和运维领域，这通常被称为 **CMDB（配置管理数据库）** 的高度抽象版。这是将 Commit/Log 数据挂载到业务线上的“钩子”。

是业务的“承载者”和“零部件”。(侧重逻辑归属)关注业务架构 (Architecture)，管理粒度，细粒度：是系统内部的组件（App、Repo、DB）。解决问题，业务是由哪些微服务构成的？代码在哪？

在 dbt 建模 中将此表设为 **SCD Type 2**，以记录资产归属权的变更历史。

字段名	类型	约束	描述	示例
entity_id	String	PK	全局资产唯一 ID（建议通过 system_code + internal_id 生成哈希）	APP_PAYMENT_SERVICE
entity_type	String	-	实体/资产类型（REPO/APP/ SERVER/DB）	MICRO_SERVICE
display_name	String	Not Null	实体/资产名称	支付网关服务
repo_url	String	Unique	关联的代码库地址	<a href="https://git.group.com">git.group.com</a>
parent_entity_id	String	FK	父级资产 ID（用于构建资产树，如：代码库属于某个应用）	ENT_APP_PAY_CENTER
owner_user_id	UUID	FK	资产主负责人。外键关联 mdm_identities	550e8400-e29b...
owner_org_id	String	FK	归属部门 ID（关联 mdm_organizations）	ORG_FIN_001
system_code	String	FK	所属物理系统编码。外键关联 mdm_systems_registry	GITLAB_CORP_PROD
internal_id	String	Unique Key	资产在源系统内的原生 ID	1024 (GitLab 项目 ID)
env_type	String	-	环境类型（PROD/STAGING/ DEV）	PROD
importance	String	-	业务重要性等级（Critical/ High/Medium/Low）	Critical
tags	JSONB	-	扩展属性（如语言、框架、 SLA等级）	{ “lang” : “Java” , “tier” : 1 }

is_active	Boolean	Default: T	该资产是否在役/未归档	true
last_sync_at	Timestamp	-	该资产最后一次从源系统同步的时间	2025-12-21 23:00:00

## 5. 项目/任务全生命周期主数据 (mdm\_projects\_tasks)

用于打通“需求 -> 开发 -> 测试 -> 发布”的交付链条。记录需求与研发项目的生命周期，用于计算交付效能和研发投入占比。

**作用：**它是 **SCD Type 2** 的典型应用场景，记录项目状态的变化。没有它，所有的 Commit 数据只是零散的代码改动，无法体现其业务价值（属于哪个迭代、解决了哪个 Bug）。

建议采用 **SCD Type 2**（保留历史版本），以便追溯。

字段名	类型	约束	描述	示例
project_id	String	PK	全局唯一项目 ID（建议：系统码+项目Key）	PRJ_JIRA_PAY_101
system_code	String	FK	来源系统编码。外键关联 mdm_systems_registry	JIRA_CORP_PROD
external_id	String	Unique	外部系统原生 ID（如 Jira 项目 Key 或 ID）	PAY-2025-001
project_name	String	Not Null	项目或迭代的正式名称	2025Q4 支付链路出海优化
project_type	Enum	-	项目类型（SPRINT/WATERFALL/RESEARCH）	SPRINT
status	String	-	项目当前状态（PLAN/DEV/TEST/RELEASED）	DEV
pm_user_id	UUID	FK	项目经理/负责人。外键关联 mdm_identities	550e8400-e29b...

org_id	String	FK	归属业务线/部门。外键关联 mdm_organizations	ORG_INTL_PAY
plan_start_date	Date	-	计划启动日期	2025-11-01
plan_end_date	Date	-	计划交付日期	2025-12-31
actual_start_at	Timestamp	-	实际开始时间（基于第一条研发日志）	2025-11-05 10:00:00
actual_end_at	Timestamp	-	实际完成时间	2025-12-20 18:00:00
budget_code	String	-	财务预算关联编码（用于 CAPEX/OPEX 审计）	BC_2025_TECH_01
budget_type	String	-	预算类型（CAPEX 资本化/OPEX 费用化）	
is_active	Boolean	Default: T	项目是否处于活跃/进行中	true
updated_at	Timestamp	-	最后一次状态变更时间	2025-12-21 23:00:00

**OneID 穿透：**确保所有涉及“人”的字段（如 manager\_user\_id, pm\_user\_id）全部强制使用 mdm\_identities 表中的 global\_user\_id。

6. 日历与时间维度主数据 (mdm\_calendar)

这是最容易被忽视但极其重要的基础表。**核心字段：**date\_day, year, quarter, month  
**作用：**所有的效能指标（如 MTTR、交付周期）必须扣除节假日，这是精确计算的前提。

字段名	类型	约束	描述	示例
date_day	Date	PK	日期主键	2025-12-21
year_number	Integer	-	公历年份	2025
month_number	Integer	-	月份 (1-12)	12
quarter_number	Integer	-	季度 (1-4)	4

day_of_week	Integer	-	星期几 (1-7, 1为周一)	7
is_workday	Boolean	-	是否为工作日 (含法定调休)	false
is_holiday	Boolean	-	是否为法定节假日	true
holiday_name	String	-	节日/假期名称	冬至/调休
fiscal_year	String	-	财年标识	FY2025
fiscal_quarter	String	-	财季标识	FY25-Q4
week_of_year	Integer	-	一年中的第几周	51
season_tag	String	-	业务季节/大促周期标签	WINTER_PROMO

## 7. 系统与接口元数据 (mdm\_systems\_registry)

**作用：**在 CDC（变更数据捕获）流程中，标记数据的来源系统。当某个源系统进行 DDL 变更时，可以通过此表评估对全集团数据链路的影响。

数据的“搬运工”和“出入口”。**(侧重连接)：**关注点 数据源头 (Source)。解决问题 数据从哪个实例来？怎么接？有没有权限？管理粒度 粗粒度：通常是一个独立部署的软件实例。

字段名	类型	约束	描述	示例
system_code	String	PK	系统唯一标识码	GITLAB_CORP_PROD
system_name	String	Not Null	系统中中文名称	集团生产 GitLab
system_type	Enum	-	系统类型	VCS (版本控制) / TICKET (工单)
base_url	String	-	系统根地址/入口	<a href="https://gitlab.group.com">https://gitlab.group.com</a>
api_version	String	-	当前对接的 API 版本（用于逻辑适配）	V4
auth_type	String	-	认证方式 (OAuth2/Token/ User_Pass)	OAuth2 / Personal_Token



<b>sync_method</b>	String	-	数据获取方式 数据获取方式 (CDC/ API Polling/ Webhook)	CDC / WebHook / API_Batch
<b>data_sensitivity</b>	String	-	数据敏感度级别	L3 (机密)
<b>sla_level</b>	String	-	系统重要性/可用性 等级	P1 (核心系统)
<b>technical_owner</b>	String	-	技术负责人 (关 联 mdm_identities)	550e8400-e29b..
<b>update_cycle</b>	String	-	预期同步频率 (Realtime/ Hourly/Daily)	Hourly
<b>is_active</b>	Boolean	Default: T	该同步源是否在役	true
<b>last_heartbeat</b>	Timestamp	-	该系统数据源最后 一次连接成功时间	2025-12-21 23:00:00
<b>remarks</b>	Text	-	备注 (如迁移计 划、特殊配置说 明)	

**OneID 穿透：**确保所有涉及“人”的字段（如 manager\_user\_id, pm\_user\_id）全部强制使用 mdm\_identities 表中的 global\_user\_id。