Jync
捷同 · 异构数据库同步

USER GUIDE

Jync 用户操作手册

从连接到同步，从监控到排障
一步一步用好 Jync 控制台

快速开始 同步任务配置 实时增量 故障恢复

v1.12.0

版本 v1.12.0 · 2026 年 11 月 · 适用于 Jync 1.12.x

目录

文档名称	Jync 用户操作手册	文档版本	v1.12.0
适用产品	Jync 捷同 1.12.x	更新日期	2026 年 11 月
读者对象	管理员、DBA、实施工程师	默认端口	8080

1 快速开始 6

1.1 登录与界面导览 6

1.2 七步完成第一次同步 8

1.3 账号、角色与修改密码 9

2 数据库连接管理 11

2.1 连接列表与分栏 11

2.2 新建连接 12

2.3 测试连接与常见错误 13

2.4 驱动与自定义类型 13

3 同步任务与同步配置 16

3.1 新建同步任务 16

3.2 选择同步对象 18

3.3 同步选项 20

3.4 捕获策略 21

3.5 调度设置 21

3.6 全量加载性能 23

3.7 列清洗 24

3.8 各表进度与游标列 25

4 运行与监控 27

4.1 看板 27
4.2 运行控制按钮 27
4.3 变更日志 28
4.4 审计日志 30
4.5 系统日志 30
4.6 性能页面与 Prometheus 指标 31
4.7 同步前检查 32
5 结构同步 34
5.1 结构同步流程 34
5.2 类型映射与建表 34
5.3 索引、视图与存储过程 35
5.4 转换审阅与 AI 辅助 36
6 全量同步 38
6.1 全量如何执行 38
6.2 分块、检查点与并行 38
6.3 观察吞吐与负载 39
6.4 中断、续传与重跑 39
7 增量同步与日志 CDC 41
7.1 增量轮询策略 41
7.2 选择与指定游标列 42
7.3 切换到日志 CDC 42
7.4 日志 CDC 的源库准备 42
7.5 CDC 状态面板 46
7.6 重建基线 47
8 ETL 任务与批次 49

8.1 新建 ETL 任务 49
8.2 写入模式与断点续跑 50
8.3 ETL 批次 51
9 数据校验与故障恢复 52
9.1 行数审计 52
9.2 手动核对行数 52
9.3 全表比对 53
9.4 停止、重置与重建 54
9.5 内容级校验 55
10 数据预览、克隆与列清洗 56
11 专属迁移 58
11.1 入口与概念 58
11.2 新建专属连接 58
11.3 新建专属任务 59
11.4 Elasticsearch / OpenSearch → Easysearch 59
11.5 MongoDB → MongoDB 59
11.6 MongoDB → KingbaseES 59
11.7 运行、取消与历史 60
11.8 权限与限制 60
12 系统管理 61
12.1 系统信息 61
12.2 AI 配置 62
12.3 外部认证 62
12.4 告警通知 64
12.5 版本检查与一键升级 66

12.6 手工升级（离线环境） 67

12.7 一键诊断包 67

13 典型场景操作 69

14 常见问题与已知限制 71

阅读约定

术语：**源库**指被读取的数据库，**目标库**指被写入的数据库；**同步任务**是一组源库、目标库、对象与策略的集合；**ETL 任务**是按取数 SQL 把结果写入目标表的任务，**ETL 批次**是共享一对连接与一条调度的 ETL 任务组；**游标**是轮询模式记录进度的列值；**位点**是 CDC 模式记录的日志位置；**基线**是一次完整的全量加载。

排版：等宽字体表示界面元素、配置键或 SQL；蓝色提示为补充说明，绿色为建议做法，黄色为需要确认的风险操作。

默认值：文中默认值来自发行包内的 `application.yml`，可被环境变量或同步任务级设置覆盖，以实际运行值为准。界面截图取自 1.x 系列，个别菜单名称可能随版本微调。

CHAPTER 01

快速开始

本章假设 Jync 已经启动并可通过浏览器访问。如果还没有部署，请先阅读《Jync 部署与运维手册》。整个首次同步大约需要十分钟，其中大部分时间花在等待全量数据写入上。

1.1 登录与界面导览

浏览器打开 `http://服务器IP:8080`。登录页会按当前认证方式分流：未启用外部认证时显示本地账号表单；启用后显示门户引导页或 OA 账号表单。首次启动会种入两个本地账号：

用户名	角色	初始密码	权限范围
`admin`	管理员	`123456`	全部操作：新建、修改、启停、删除、升级
`view`	访客	`123456`	只读：可查看页面、数据与 SQL，不能提交任何变更

首次登录必须改密

使用默认密码时，页面顶部会持续显示红色提醒。任何能访问该端口的人都可以用默认密码登录并修改连接配置，请在首次登录后立即通过右上角用户菜单修改密码。



图 1–1　登录页；中英文与昼夜主题可在登录后切换

登录后顶部导航分为三级：**看板**、**同步任务**、**配置管理**、**日志管理**、**系统信息**。其中「配置管理」包含数据库连接、AI 配置（AI 启用时）与外部认证（仅管理员），「日志管理」包含变更日志、审计日志（仅管理员）与系统日志。右侧依次是语言切换、主题切换、赞赏入口和账号菜单；窄屏下分组折叠为可展开的一行标题。

导航	用途
看板	全局统计、同步任务概览与最近活动
同步任务	创建同步任务，配置对象、策略与调度
配置管理 → 数据库连接	维护源库与目标库连接
配置管理 → AI 配置	配置用于存储过程转换的模型供应商（可选）

导航	用途
配置管理 → 外部认证	接入 OA 接口、网关头或票据单点登录（仅管理员）
日志管理 → 变更日志	查询每次结构变更与数据变更的明细
日志管理 → 审计日志	管理员操作留痕（仅管理员）
日志管理 → 系统日志	在页面内检索应用运行日志，所有登录用户可见
系统信息	运行参数、版本检查与一键升级

1.2 七步完成第一次同步

1. **确认网络**

   从 Jync 所在主机测试到源库与目标库的端口连通性。容器部署时注意容器内的 `localhost` 指容器自身。

2. **新建源连接**

   进入「数据库连接」，在源库分栏点「新增源连接」，填写信息并测试通过后保存。

3. **新建目标连接**

   同样方式新增目标连接。源与目标是两个独立连接，即使是同一个实例也要分别建。

4. **新建同步任务**

   进入「同步任务」→「新建同步任务」，填写名称，选择源连接与目标连接，保存。

5. **勾选对象与选项**

   在同步任务详情页勾选要同步的表、视图、过程；按需开启表结构、表数据、索引等开关。

6. **立即同步验证**

   点「立即同步」跑一轮，确认目标表已创建、数据行数正确。

7. **启动持续同步**

   点「启动同步」，同步任务按调度周期持续运行；在看板观察状态与变更量。

> **先小后大**
>
> 第一次建议只勾选一两张小表，确认账号权限、类型映射和目标库写入都正常后，再扩大对象范围。这样能把问题限定在最小范围。

1.3 账号、角色与修改密码

Jync 的权限模型只有一条硬规则：**所有 POST 请求都要求管理员**。访客账号可以浏览全部页面、查看数据预览与 SQL、导出日志，但界面上不会渲染写操作按钮；即使手工构造请求，也会被服务端拒绝。

修改密码

点击右上角用户名 →「修改密码」，输入旧密码与新密码。密码使用 BCrypt 存储，数据库中不保存明文。新密码建议至少 12 位，包含大小写、数字与符号。

忘记密码

系统不提供找回功能，也不发送邮件。本地账号的处理方式是：停止应用，删除元数据库中 `app_user` 表中对应用户的行，重启后应用会重新种入初始账号。此操作会重置该账号密码，请确认没有其他依赖。

通过外部认证登录的用户没有本地密码，也不写入 `app_user`，密码问题应回到企业 OA 或统一门户处理。如果关闭了本地登录、外部认证又暂时不可用，可以在服务器上以 `JYN_RESET_LOGIN=true` 启动一次应用，本地登录会自动恢复；修好外部认证配置后保存并去掉该变量再重启。

账号种入规则

首次启动时写入 `admin` 与 `view`；只要用户表非空就跳过种入，**不会覆盖任何已有密码**。因此升级不会把修改过的密码重置回默认值。

1.4 上线前检查清单

第一次在生产环境使用前，按下表逐项确认。清单覆盖的是最容易遗漏、又最难在事后补救的环节。

检查项	确认方式	期望结果
Java 版本	`java -version`	17 或更高
端口连通	`nc -vz 源库IP 端口`	succeeded
源库账号	用该账号执行一次 SELECT	可读目标表，无需写权限
目标库权限	建一张临时表后删除	具备 CREATE / INSERT / UPDATE / DELETE
元数据目录	查看 `data/` 所在磁盘	可写，剩余空间 ≥ 10GB
日志目录	查看 `logs/` 所在磁盘	可写，已配置滚动上限
时间同步	`timedatectl` / NTP	与源库、目标库偏差 < 5 秒
默认密码	用 admin 登录	已修改，页面无红色提醒
外部认证（如启用）	在「配置管理 → 外部认证」点测试	身份识别正确，来源可信；本地登录保留策略已确认

先演练再上线

在生产库执行前，用一套同版本、同类型的测试库完整走一遍「连接 → 结构 → 全量 → 增量」。Jync 的配置可以克隆，演练通过的配置能直接复用到生产，只改连接即可。

CHAPTER 02

数据库连接管理

连接是同步的起点。Jync 把连接分为「源」与「目标」两类，分别管理、分别分页。一个数据库实例既可以是某个同步任务的源，也可以是另一个同步任务的目标，但需要建立两条独立的连接记录。

2.1 连接列表与分栏

页面分为上下两个区域：源数据库连接与目标数据库连接。每个区域显示名称、类型、主机端口、数据库名/服务名、用户名和操作按钮。右上角的「新增源连接」「新增目标连接」分别创建对应用途的连接。

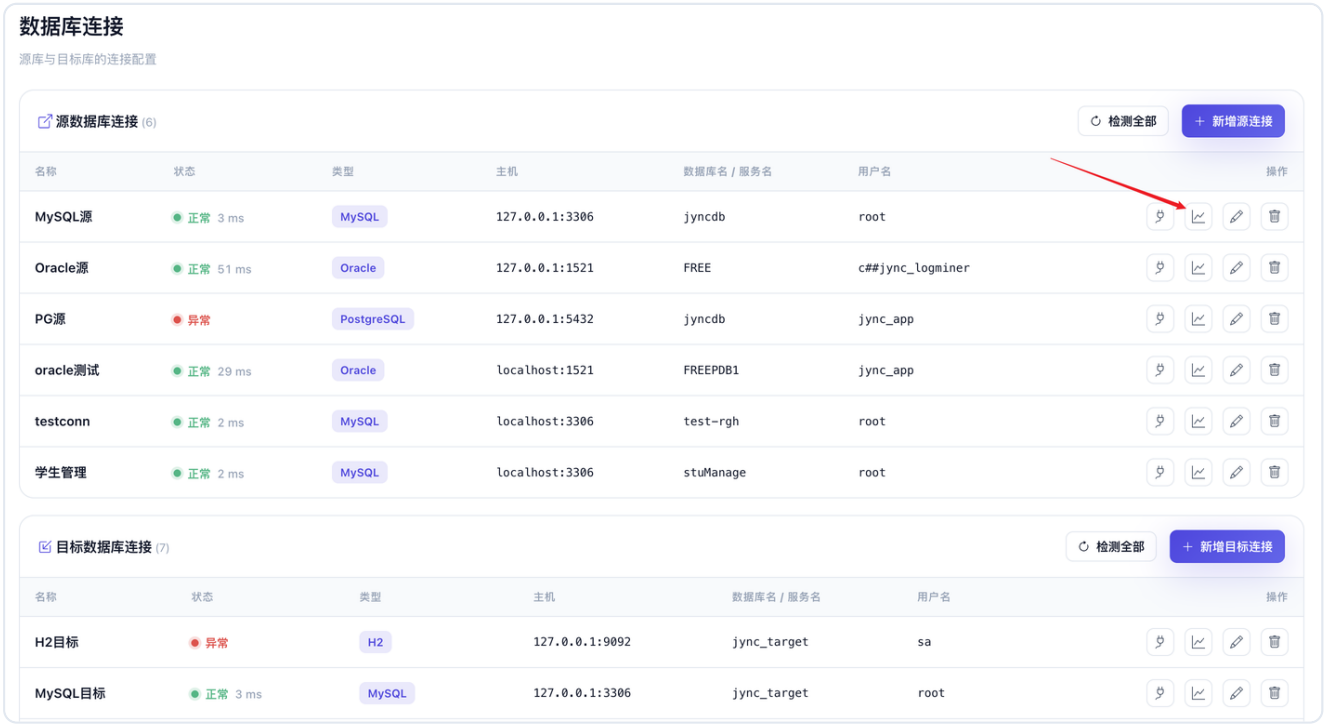


图 2–1　连接按源与目标分栏，操作列提供测试、编辑与删除

每一行的操作含义：

- **闪电图标**：立即测试连通性，不打开编辑页。
- **铅笔图标**：编辑连接信息。修改密码后需要重新测试。
- **垃圾桶图标**：删除连接。被同步任务引用的连接不能删除，需要先处理相关同步任务。

2.2 新建连接



图 2–2　新建连接：选择用途与数据库类型，其余字段按类型自动切换

字段	说明	示例
用途	源数据库（读取）或目标数据库（写入）	源数据库连接
名称	页面展示名，建议包含环境与库名	生产–订单库
类型	34 种内置类型或「自定义」	Oracle
主机 / 端口	数据库监听地址与端口	10.10.1.20 / 1521
数据库名 / 服务名	不同数据库含义不同：MySQL 为库名，Oracle 为服务名，SQL Server 为库名	ORCLPDB1
用户名 / 密码	建议使用专用的只读或读写账号	jync_reader

字段	说明	示例
附加 URL 参数	追加到 JDBC URL 末尾，用于编码、时区、SSL 等	useUnicode=true&characterEncoding=utf8

控制台会根据类型自动拼装 JDBC URL，不需要手工记忆模板。保存前建议先点「测试连接」；测试会真正建立连接并执行一次轻量查询，能发现地址、端口、账号、权限和驱动问题。

SQL Server 命名实例要留空端口

命名实例没有固定端口，由 SQL Server Browser 动态解析。连接时把端口留空，主机写 `主机名\实例名`，或在「附加 URL 参数」中填 `instanceName=实例名`；端口为空时 Jync 不会拼出 `:1433`。普通默认实例留空端口同样按 1433 连接。切换类型时端口框会自动清空，手动填过的端口不会被覆盖。

时区

MySQL 建议在附加参数中显式指定 `serverTimezone=Asia/Shanghai`，否则时间字段可能出现 8 小时偏差。Oracle 与 PostgreSQL 通常无需额外设置。

2.3 测试连接与常见错误

现象	常见原因	处理
连接超时	网络不通、防火墙拦截、端口写错	在服务器上用 telnet / nc 测试端口；容器部署检查是否用了容器内地址
SQL Server 命名实例超时（约 15 秒）	URL 中带了 `:1433`，而实例端口由 SQL Browser 动态分配	清空端口，主机写 `主机名\实例名` 或附加参数填 `instanceName`
拒绝连接	数据库未监听该地址，或只监听 localhost	检查数据库的 listen_addresses / bind-address
账号密码错误	密码含特殊字符被截断，或账号限定来源 IP	用同一账号在服务器本地登录验证；检查账号的 host 限制
数据库不存在	库名 / 服务名填错	Oracle 填服务名而非 SID；MySQL 填库名
驱动类找不到	自备驱动路径错误，或驱动类名与预设不一致	确认 jar 路径可读；在连接卡片中覆盖驱动类名
SSL 相关错误	数据库强制 TLS，或证书不受信	在附加参数中配置 SSL；测试环境可显式关闭

2.4 驱动与自定义类型

34 种内置类型中，25 种的 JDBC 驱动随发行包提供。以下九种需要自备官方 jar：

数据库	驱动类名	说明
南大通用 GBase	按厂商提供	官方构件未发布到 Maven Central
神通 Oscar	按厂商提供	需填写 jar 路径
华为 GaussDB	`com.huawei.gaussdb.jdbc.Driver`	使用随库提供的 gaussdbjdbc.jar；不要按 openGauss 方式连接
SAP HANA	`com.sap.db.jdbc.Driver`	使用 SAP 提供的 ngdbc.jar
ClickHouse / Hive / Impala	按厂商提供	官方 JDBC 驱动未发布到 Maven Central
Elasticsearch	按厂商提供	使用 Elastic 提供的 JDBC 驱动
MongoDB（Atlas SQL）	按厂商提供	原生驱动已内置；Atlas SQL 走 JDBC 时需自备驱动

外部驱动通过独立的 `URLClassLoader` 加载，不污染主程序类路径。厂商 jar 中的驱动类名与预设不同时，可以在同一张连接卡片里覆盖驱动类，留空则使用预设值。

自定义类型

未预置的数据库选择「自定义」，填写 JDBC URL、驱动类名和驱动 jar 路径，并指定方言家族：

方言家族	适用情况
通用 ANSI	不确定时的保守选择，使用标准 SQL 与通用 upsert 模拟
Oracle	Oracle 协议兼容库，或需要 ROWID / MERGE 语义
MySQL	MySQL 协议兼容库
PostgreSQL	PostgreSQL 协议兼容库
SQL Server	SQL Server 协议兼容库
DB2	DB2 协议兼容库

> **协议兼容预设**
>
> 万里 GreatDB、银联 UPDRDB、星环 KunDB 兼容 MySQL 协议；平安 RASESQL、东方金信 SeaboxSQL 兼容 PostgreSQL 协议。这些预置复用内置驱动与对应方言族，不需要自备 jar。

驱动选择建议

场景	建议
使用预置类型	直接选择预置，驱动与方言随类型自动匹配，无需准备 jar

场景	建议
同一数据库多个版本	先用预置版本验证；确有兼容问题再上传对应版本驱动
信创协议兼容库	选兼容族预置（MySQL / PostgreSQL），不要按产品名另建类型
预置未覆盖的数据库	新建「自定义类型」，填写 JDBC URL、驱动类名与方言族
同库多版本驱动共存	避免同时上传多个版本，类加载顺序不确定会导致连接异常

驱动版本与数据库大版本对齐

驱动过旧可能不支持新版本的认证方式或数据类型；驱动过新也可能要求更高版本的 JDK。上传前先用「测试连接」确认，再保存连接。

CHAPTER 03

同步任务与同步配置

同步任务是 Jync 的配置单元：一个源连接、一个目标连接、一组同步对象和一套策略。同一对数据库之间可以建多个同步任务，用不同的对象范围或调度互不干扰地运行。

3.1 新建同步任务



图 3–1　新建同步任务只需名称、源连接与目标连接；详细配置在保存后进入详情页完成

1 **填写名称**

建议采用「源库→目标库–用途」的命名，例如「订单库→达梦–实时」。

2 **填写描述**

记录同步任务负责人、切换窗口或特殊约定，便于交接。

3 **选择源数据库**

只能选择源类型的连接。

4 **选择目标数据库**

只能选择目标类型的连接。

5 **保存**

保存后自动进入同步任务详情页；此时还没有勾选任何对象。

批量创建任务

表很多、表名没有规律时，不建议手工拆任务。任务管理页「新建任务」旁的「批量创建」（仅管理员）会读取一次源库目录统计，按估算行数把表均衡分进 N 个任务（1–50，默认 8），预览后一次建出多个「未启动」的任务。

1 **选择连接与任务数**

源 / 目标连接、任务数 1–50（默认 8）与名称前缀。名称重复时会在创建任何任务之前拒绝。

2 **读取统计**

优先读产品目录（不扫表）。需要更准时勾选「精确统计行数」，只对没有估算的表逐条执行 `COUNT(*)`，单表 15 秒、整页 120 秒预算，超预算停止并在页面注明。

3 **查看预览**

按估算行数从大到小，每张表放进当前总重最小的箱（LPT）；大小未知的表按已知大小的中位数计重。没有估算的表单独计数、不计入合计；可导出 CSV（UTF-8 BOM）。

4 **创建并逐个启动**

创建出的任务一律「未启动」，采集方式与选表随表单；确认每张表的清单后再逐个启动。规划只使用独立的两连接临时池，不占用同步连接池，也不影响正在运行的同步任务。

> **分箱是确定性的**
>
> 同样的源库统计与任务数，在预览与创建时得到同样的分箱；预览与创建之间源库若有变化，以创建时读取到的清单为准。



图 3–2　批量创建预览：每个任务的表清单、估算行数与统计来源；没有估算的表单独计数，可导出 CSV

3.2 选择同步对象

同步任务详情页左侧是同步对象区域，按「表」「视图」「存储过程与函数」分组。每组支持搜索、全选与清空选择。

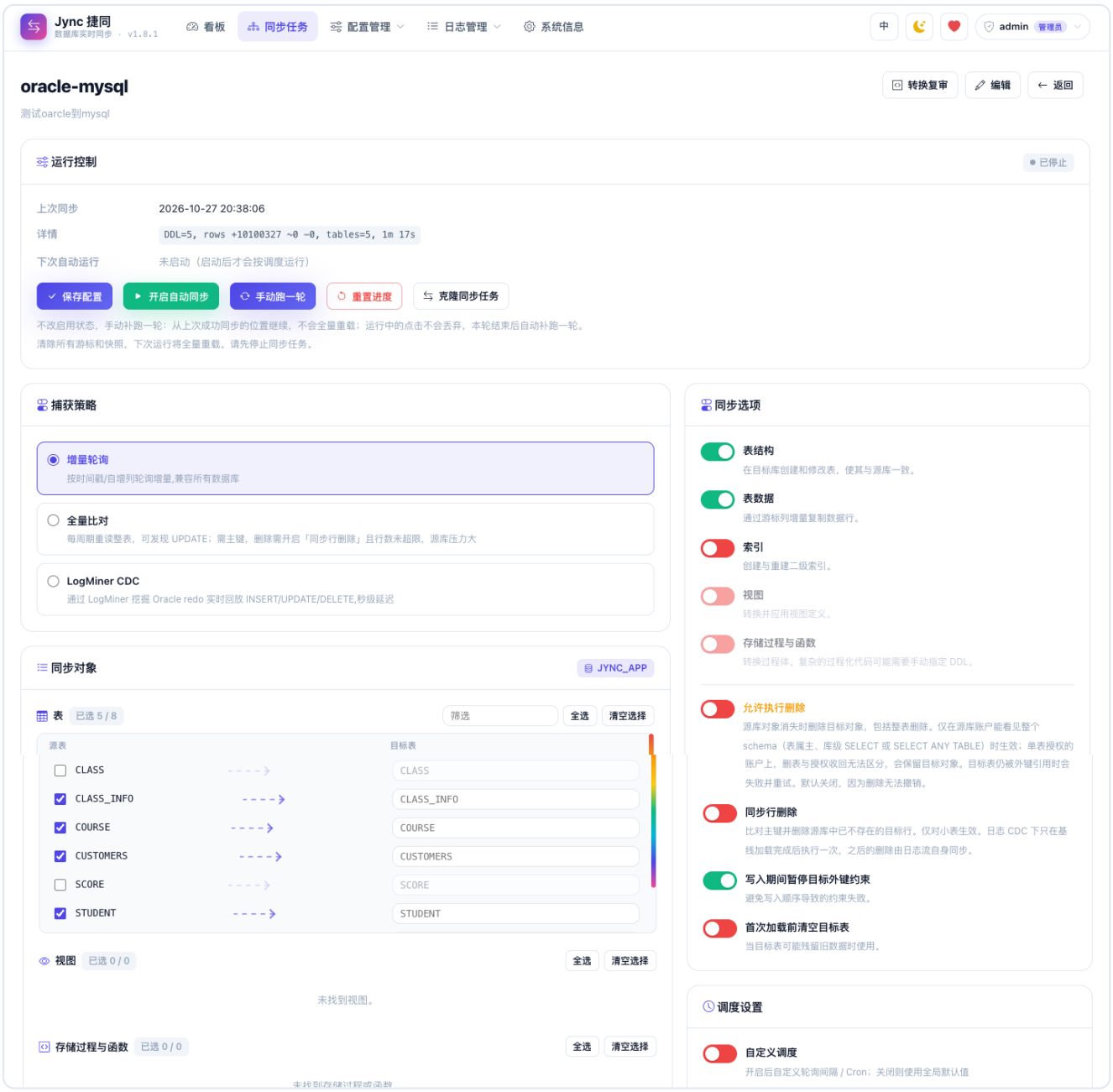


图 3–3　同步任务详情上半部分：运行控制、捕获策略、同步对象与同步选项

> **选择语义**
>
> 未勾选任何表时，表示同步源库的全部表（通配）。取消勾选某些表会收窄为白名单；把已勾选的表全部取消，也会回到「同步全部」。如果不希望同步全部，请至少保留一张表并明确勾选。

对象清单来自元数据快照。源库新增表后，需要刷新快照或下次结构同步后才会出现在列表中。视图与存储过程的转换依赖目标库方言，复杂对象建议先在测试库验证。

3.3 同步选项

选项	作用	建议
表结构	在目标库创建或修改表，使结构与源库一致	首次迁移开启；目标表已手工建好时可关闭
表数据	通过游标或 CDC 复制数据行	同步任务必开
索引	创建或重建二级索引	首次迁移开启；大表可配合延迟索引
视图	转换并应用视图定义	按需开启
存储过程与函数	转换过程体并创建到目标库	按需开启，配合转换审阅
允许执行删除	源对象消失时删除目标对象；源库账户需能看见整个 schema（见表下说明）	默认关闭；删除不可撤销，仅在确认后开启
同步行删除	比对主键并删除目标库中源库已不存在的行	仅对小表有效，需评估目标库写入压力
写入期间暂停目标外键约束	避免写入顺序导致的约束失败	目标库外键较多时开启
首次加载前清空目标表	全量前清空目标表残留数据	仅当目标表可清空时开启；同步任务克隆时强制关闭

清空目标表

「首次加载前清空目标表」会删除目标表现有数据，且不可撤销。生产环境开启前请确认目标表确实只由 Jync 写入，并已做好备份。同步任务克隆不会继承此开关。

删表与授权收回

源库表整表删除、且开启「允许执行删除」时，目标表会同步删除。但源库账户被收回某张表的 `SELECT` 后，这张表同样会从清单里消失——两者在源端无法区分。因此 Jync 只在源库账户能看见整个 schema（表属主、库级 `SELECT`、Oracle 的 `SELECT ANY TABLE`、SQL Server 的 `VIEW DEFINITION`，PostgreSQL 系默认可见）时，才把清单缺项当作删表；否则保留目标表，并在日志中告警一次。需要让删表联动生效时，请以表属主身份同步或授予库级 `SELECT`。该判断只在出现删表候选时才向源库查询，不影响正常轮次的开销。

运行中的同步任务需要先停止同步才能修改对象与选项，部分配置修改在下次启动时生效。页面会在运行状态下显示黄色提示条，说明哪些修改不会即时生效。

3.4 捕获策略

捕获策略决定数据如何被发现，三选一：

POLLING

增量轮询

按时间戳或自增列周期查询，兼容所有数据库，源库只需只读权限。默认每 2 秒一次。

COMPARE

全量比对

周期全表比对，能发现 UPDATE 与 DELETE，但压力大，仅适合小表。

CDC

日志 CDC

解析 MySQL 兼容家族 binlog、Oracle / 达梦 LogMiner、PostgreSQL 方言家族逻辑解码、SQL Server CDC 或崖山 YStream，秒级延迟。

NOTE

切换会重做基线

从轮询切到 CDC 会重新执行全量；切回轮询保留位点，但游标清零后先全量对齐。

v1.10.0 起，Doris、ClickHouse、Hive、Impala、Elasticsearch 与 MongoDB（Atlas SQL）也可以作为同步源出现在源库下拉中，但只提供「增量轮询」与「全量比对」两种策略，且不会出现在目标库下拉里。这些引擎一律按无主键读取：增量只识别新增，需要收敛更新与删除时请选全量比对。

3.5 调度设置

轮询与全量比对同步任务默认按全局轮询间隔运行（默认 2 秒）。开启「自定义调度」后，可以为当前同步任务单独覆盖调度方式；关闭开关时两个输入框折叠，同步任务回落到全局默认值。



图 3-4　调度设置：自定义调度开关、轮询间隔与 Quartz Cron 表达式

字段	说明	示例
轮询间隔	两次增量轮询之间的毫秒数，适用于持续追增量；不建议低于 1000ms	2000
Cron 表达式	Quartz 语法，6 或 7 段且秒在最前；填写后优先于轮询间隔，适合每天凌晨定时批量同步	`0 0 2 * * ?`

Cron　按服务器时区计算，保存后需要重新启动同步任务才生效。两个字段都留空时按全局轮询间隔运行；只填写轮询间隔时不使用 Cron。

CDC 同步任务

日志 CDC 使用长驻消费线程流式回放，不再走 Quartz 轮询，因此调度卡片会整体隐藏，轮询间隔与 Cron 表达式都不生效。CDC 的基线全量仍走同一套加载路径，所以「全量加载性能」卡片保持可见。

调度与性能分开

批次大小与并行度不属于调度设置，它们位于下一节的「全量加载性能」卡片中。这样即使关闭自定义调度、清空轮询间隔，性能参数也不会被一并清掉。

3.6 全量加载性能

「全量加载性能」卡片控制首次全量与 CDC 基线重建的读写节奏，与调度方式无关：无论同步任务采用轮询、全量比对还是日志 CDC，这张卡片都会显示。留空表示使用全局默认值，同步任务级设置优先于全局配置。



图 3–5 全量加载性能：批量大小与并行度，留空即使用全局默认值

字段	默认值	作用与边界
批量大小	2000	首次全量每个 JDBC 批次写入的行数，可填 1–20000，建议 1000–5000；含大字段（LOB）的表建议 500–2000。批次越大提交次数越少、加载越快，但单次事务更大，回滚更慢，目标库的 undo / redo 与锁持有时间也同步上升。
并行度	2	首次全量的分片 worker 数，可填 1–8，建议 1–4；1 表示顺序加载。实际值还受连接池上限约束。数据库较敏感时设为 1。

有主键的表都可以并行：平台对**整数前导主键**按区间算术切分，对**其他键形态**按采样边界切分，再由多个 worker 并行分页读取与写入。无主键表无法安全切分，只能顺序加载，不需要手工干预。

> **只影响全量**
>
> 批量大小与并行度只作用于首次全量和 CDC 基线重建。增量轮询与 CDC 回放继续使用全局的增量批次设置，不会因为这里调大而给在线流量增加压力。

先看数据库承受能力

并行度提高会近似线性增加源库与目标库的连接数、CPU 与 IO 压力。生产库较敏感时把并行度设为 1，批量大小回落到 1000–2000；调大后先观察一轮全量的吞吐日志与数据库监控，再决定是否继续提高。源库与目标库使用不同连接配置时，连接池需满足 `max-pool-size ≥ 并行度 + 1`；两者共用同一连接配置时需满足 `max-pool-size ≥ 2 × 并行度 + 2`。连接池如何限制并行度、如何阅读吞吐日志，见《部署与运维手册》第 5 章。

3.7 列清洗

列清洗在写入目标库之前、于内存中修改字符串值，源库数据始终不变。规则按「表 + 列」配置，同一张表的多条规则按列表从上到下依次执行。该卡片独立于主配置表单：添加、删除规则即时保存，不需要再点主表单的「保存配置」。



图 3–6　列清洗：按表与列选择规则，参数与替换内容随规则类型出现

规则	参数	行为
去首尾空白	无	去掉字符串开头与结尾的空白字符
转大写	无	按与区域无关的规则转换为大写
转小写	无	按与区域无关的规则转换为小写
空串转 NULL	无	空串或只含空白的字符串写成 NULL
去前缀	要移除的前缀	仅当值以该前缀开头时去掉一次
去后缀	要移除的后缀	仅当值以该后缀结尾时去掉一次
正则替换	正则表达式、替换为	替换所有匹配；替换内容可用 `$1` 引用捕获分组

清洗只处理已经是字符串的值：数字、日期与二进制列原样通过，不会被隐式转换成字符串。列名按大小写不敏感匹配；如果清洗的是主键列，「同步行删除」在源端与目标端比对主键时也使用清洗后的值，避免把已清洗的行误判为源端已删除。

三条写入路径都生效

列清洗同时作用于首次全量、增量轮询与日志 CDC 回放。规则在保存时校验：正则表达式必须能编译，替换串中的 `$` 分组引用与反斜杠必须合法。历史上已损坏的规则在运行时会被跳过并记录警告，不会中断整轮同步。

典型用法

源库手机号带前导空格时先去首尾空白；历史数据用 `^1` 之类的前缀标记时去前缀；空字符串统一转 NULL，避免下游把空串当成有效值。规则越少越容易维护，优先在源端修正数据质量。

运行中不能改

同步任务运行期间列清洗规则不可修改，需要先「停止同步」。删除规则按列表位置执行，删除前确认当前顺序符合预期；清洗会改变写入值，启用前建议先在测试同步任务验证。

3.8 各表进度与游标列

页面底部的表格列出每张表的策略、游标列、最后同步值、累计同步行数、上次同步时间，以及每行的游标列设置入口。

列	含义
策略	自动识别或手动指定后的实际策略，例如「时间戳」「数字主键」「全量比对」
列名	用作游标的列；无可用列时为空
最后同步值	已成功写入目标库的最大游标值
累计同步行数	同步任务生命周期内写入的行数，含重复 upsert
上次同步	该表最近一次成功同步的时间
设置游标	手动指定游标列，优先级高于自动识别

策略旁出现黄色警告图标时，表示该表存在同步盲区，例如数字主键只能发现新增、无法发现更新。

选择游标列的三条原则

1. **更新时必须变化**：只有每次 UPDATE 都会写入新值的列，才能让更新时间之后的增量查询捕获到这一行。
2. **单调递增且类型稳定**：时间列或自增数字列均可；不要使用可能回退、重置或被业务改写的列。
3. **有索引最好**：增量查询会按游标列做范围过滤，没有索引时每次窗口都会扫描全表，表越大越慢。

```
-- MySQL：增加自动维护的更新时间列（推荐）
ALTER TABLE orders
  ADD COLUMN update_time DATETIME
  DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  ADD INDEX idx_orders_update_time (update_time);

-- Oracle：由应用或触发器维护更新时间
ALTER TABLE orders ADD (update_time TIMESTAMP DEFAULT SYSTIMESTAMP);
CREATE INDEX idx_orders_update_time ON orders(update_time);
```

如果业务表无法增加列，优先考虑日志 CDC；它直接读取数据库日志，不依赖任何业务列，也能捕获删除。只有在既不能改表、也不能开启日志时，才退回数字主键策略，并接受「更新不同步」的限制。

CHAPTER 04

运行与监控

4.1 看板

看板是进入系统后的默认页面，顶部四张卡分别统计同步任务总数、数据库连接数、同步表数与 24 小时变更量。下方左侧是同步任务概览，右侧是最近活动；v1.10.0 起还会并排展示实时同步与 ETL 两条趋势曲线，可在「变更数 / 行数」之间切换。

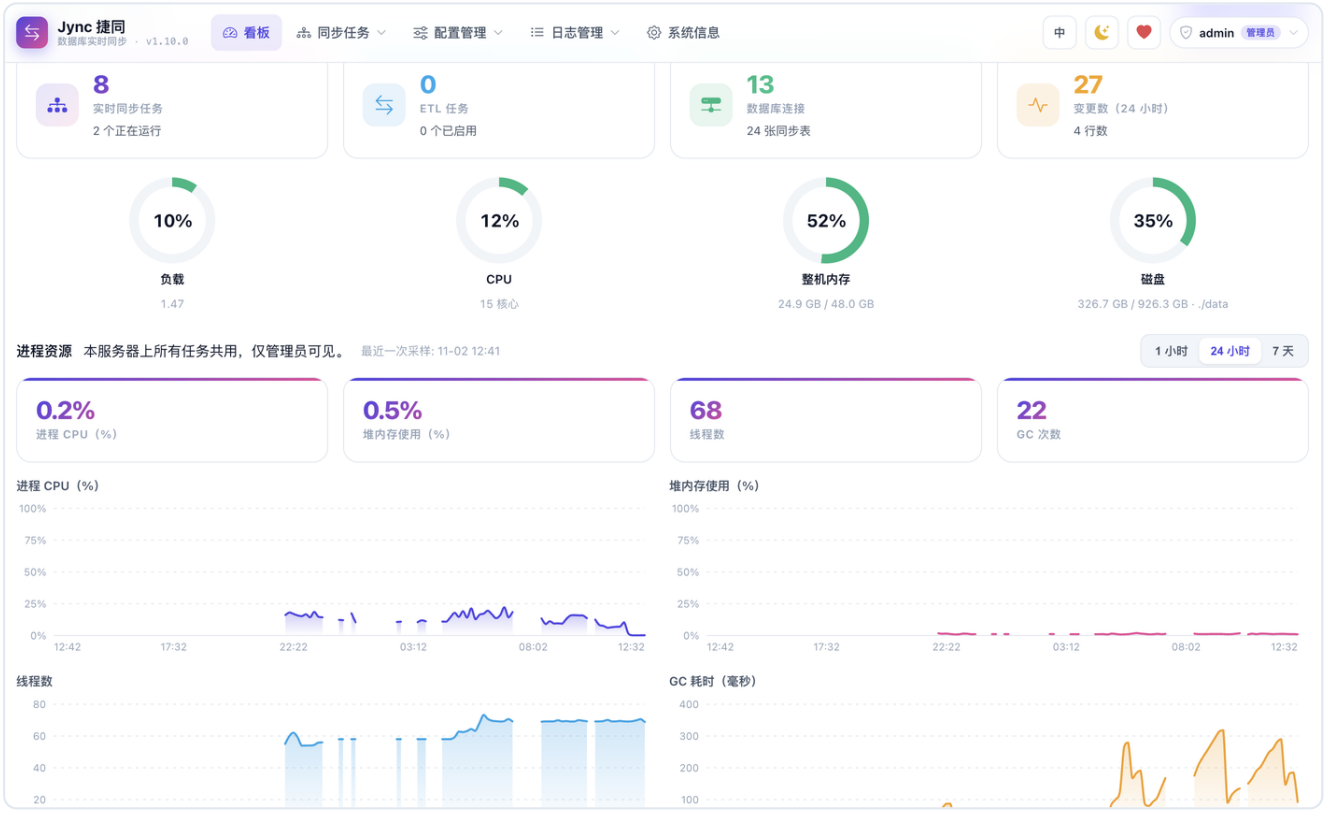


图 4–1　看板：同步任务状态、24 小时变更与实时同步 / ETL 双趋势曲线

- **同步任务状态**：运行中、已停止、异常、未开始。
- **已跟踪表**：同步任务当前纳入同步的表数量。
- **已同步行数**：同步任务累计写入行数。
- **上次同步**：最近一次完成的同步时间；异常同步任务会显示错误摘要。

4.2 运行控制按钮

按钮	行为	使用时机
保存配置	保存对象、选项、策略与调度	修改配置后；运行中部分配置需停止后保存

按钮	行为	使用时机
启动同步	把同步任务置为启用，Quartz 开始调度	首次配置完成，或故障处理后恢复
停止同步	停止调度，并通知正在执行的一轮在下一个已提交批次边界安全退出；CDC 同步任务会中断消费线程	修改配置、维护窗口、切换策略前
立即同步	立刻执行一轮；运行中点击会合并为一轮补跑	验证配置、手工补数据
重置进度	清除游标与快照，下次执行全量重载	数据不一致、游标失效、需要重建基线时
重建基线	CDC 位点失效后重新记录位点并全量	归档 / WAL / binlog 被清理导致位点失效
强制解锁	清除数据库中的同步锁	确认没有任务在跑，但锁未释放时

立即同步不会并发

同步任务正在运行时点击「立即同步」，不会启动第二个执行实例；当前轮次结束后会补跑一轮。这是三层锁机制在界面上的体现。

停止在批次边界生效

「停止同步」不会截断正在写入的批次：本轮在下一个已提交批次边界退出，已写入的数据与断点都保留，随后释放同步锁，配置可以立即修改。被停止的一轮记为「已停止」，不计入失败次数，也不会把任务推向错误停机。

「下次自动运行」时间已过、数据却不再增长

如果详情页的「下次自动运行」停在过去的时间，同时出现「上一轮仍在执行：最近一次进度在 N 分钟前」，说明上一轮还没有返回；调度会跳过它之后的所有触发。无进展超过 `sync.stuck-threshold-ms`（默认 30 分钟）后标红「疑似卡死」——等待不会恢复，需要重启本实例才会重新调度。

4.3 变更日志

变更日志记录每一次结构变更与数据变更，可按同步任务筛选、刷新、开启自动刷新或清空。每条记录包含发生时间、对象类型、对象名、变更类型、影响行数、耗时与详情。

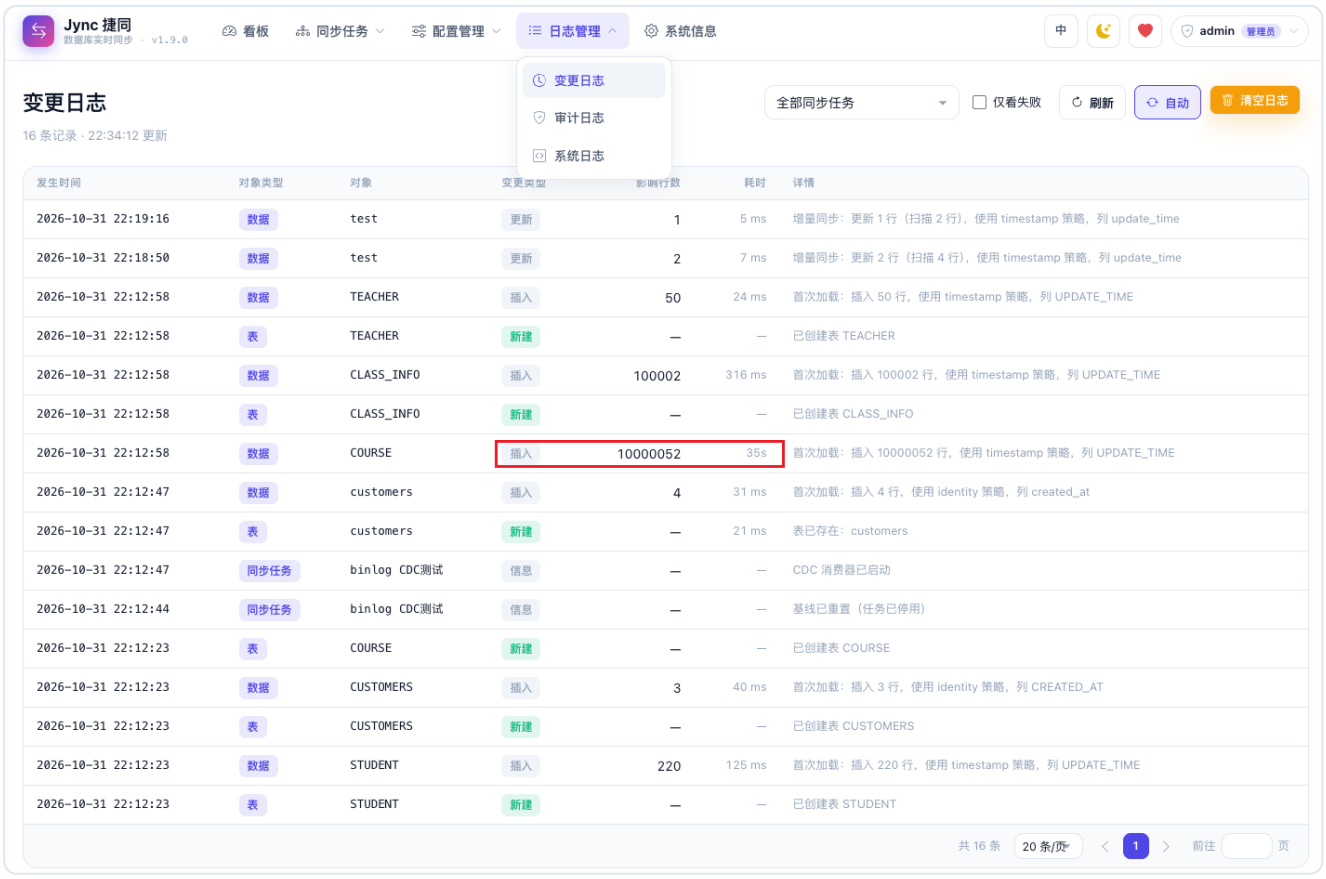


图 4–2　变更日志：结构变更与数据变更统一留痕

变更类型	典型详情
新建 / 修改	Created table orders with 2 index(es)
插入	Initial load: 104 row(s) inserted via timestamp strategy on update_time Initial load: 50 row(s) inserted (53 scanned, 3 already up to date) via timestamp strategy on update_time
更新	Incremental sync: 1 row(s) updated (2 scanned) via timestamp strategy
删除	CDC delete applied（CDC 模式）
存储过程	Created PROCEDURE InsertTestData

`scanned` 是本轮实际读取的行数。初始加载时，目标表已持有且值相同的行不会被 upsert 改动，因此只计入扫描数、不计入插入数，并单独标注为 `already up to date`；增量窗口与整表重扫的周期不显示该从句，避免每轮重复。

变更日志默认保留 30 天，超过保留期的记录由后台每小时清理一次。保留天数可在配置中调整，小于等于 0 表示永久保留，需要自行评估磁盘占用。

4.4 审计日志

管理员的所有写操作都会写入审计表，包括登录、修改密码、连接增删改、同步任务配置变更、启停、升级等。审计页位于「日志管理 → 审计日志」，支持按时间与操作类型筛选，用于回答「谁在什么时候改了什么」。外部认证的登录失败只记录用户名、来源与错误类别，密码、票据与外部响应内容都不会写进审计。

4.5 系统日志

系统日志页位于「日志管理 → 系统日志」，所有登录用户都可以访问，用于在不登录服务器的情况下查看应用自身的运行日志。页面只读取日志目录中由滚动策略生成的白名单文件，不能浏览任意路径。

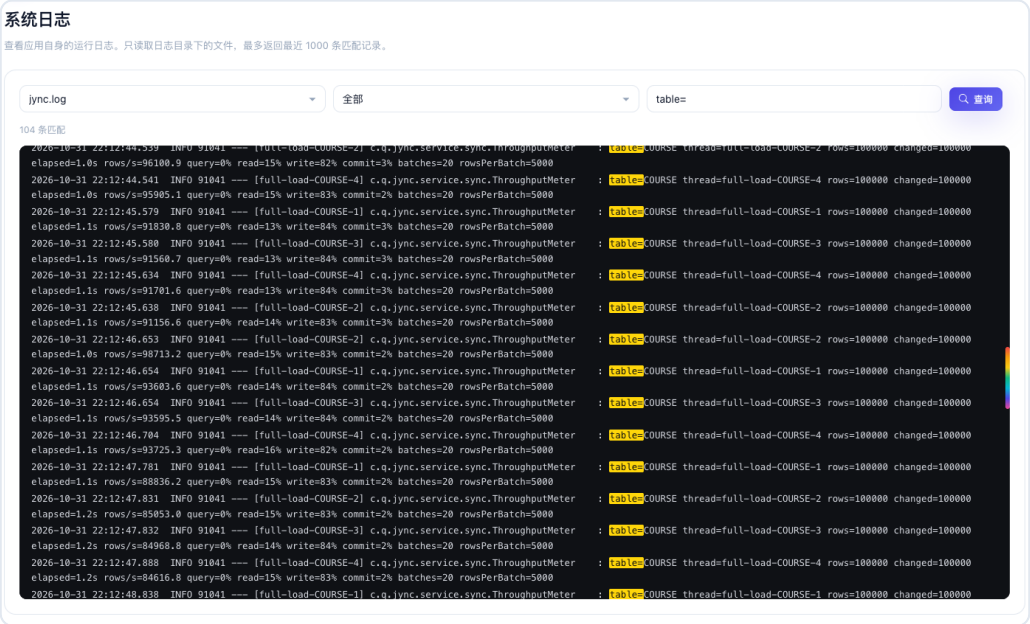


图 4–3　系统日志：按文件、级别与关键字检索，命中内容高亮显示

筛选项	说明
日志文件	选择当前日志或按日期滚动的历史文件；gzip 归档会在读取时自动解压
级别	DEBUG / INFO / WARN / ERROR，或「全部」
关键字	对日志内容做模糊匹配，不区分大小写，命中部分高亮

读取是有界的：普通文件从尾部读取，归档文件从头部读取，单次最多扫描 16 MB，返回最近 1000 条匹配记录。超过上限时页面会提示「只显示最近的日志，更早的内容未加载」，需要更早的内容请下载日志文件或用 `grep` 检索。日志文件固定使用 UTF-8 编码，中文不会因为服务器默认字符集不同而乱码。

> **先定位再登录服务器**
>
> 用户反馈「同步没动」时，先在系统日志里按级别 ERROR 和关键字 `rows/s=` 看一轮，确认是源库读取慢、目标库写入慢还是锁与连接问题，再决定是否登录服务器取完整线程栈。

4.6 性能页面与 Prometheus 指标

v1.10.0 新增两个性能页面，入口分别在「数据库连接」与「同步任务」列表。采样由后台定时执行，重启后约 30 秒出现首个采样点；历史保留 7 天，超过后按天清理。

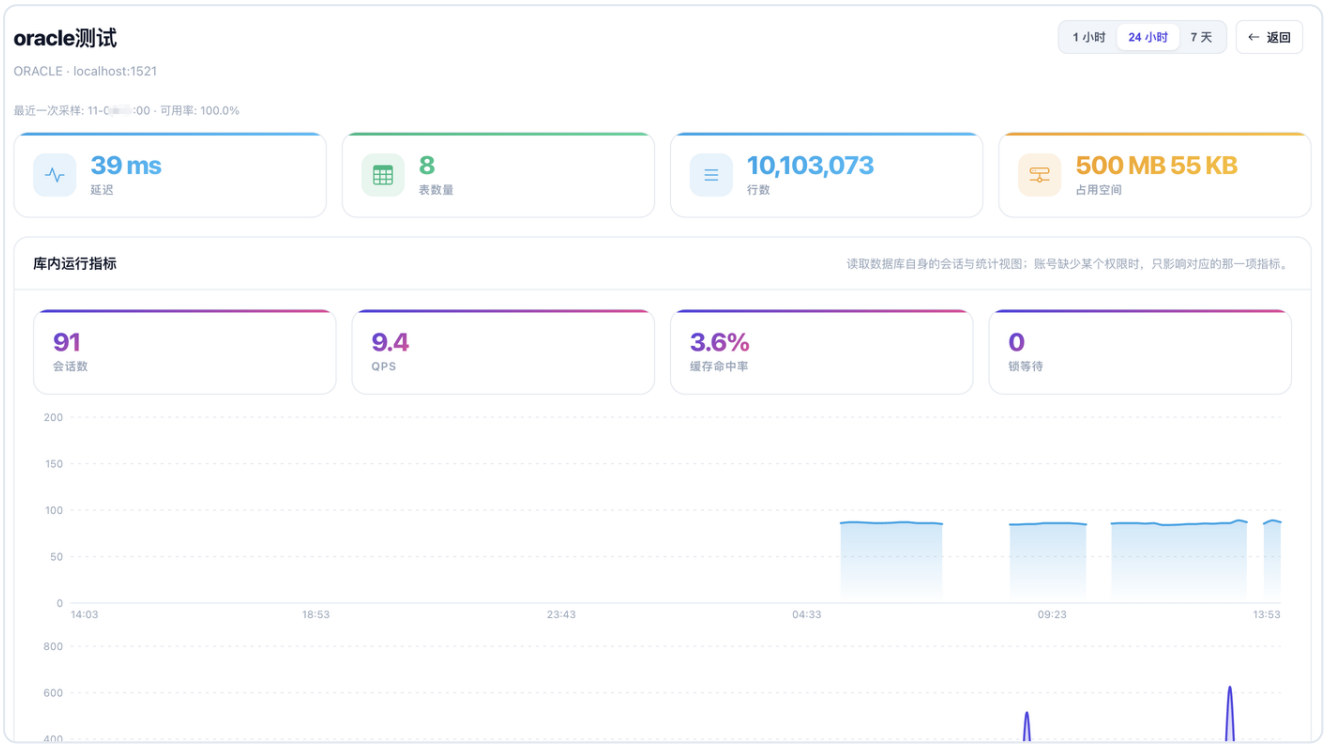


图 4–4　数据库性能：连接延迟与可用性、会话 / QPS / 缓存命中率 / 锁等待

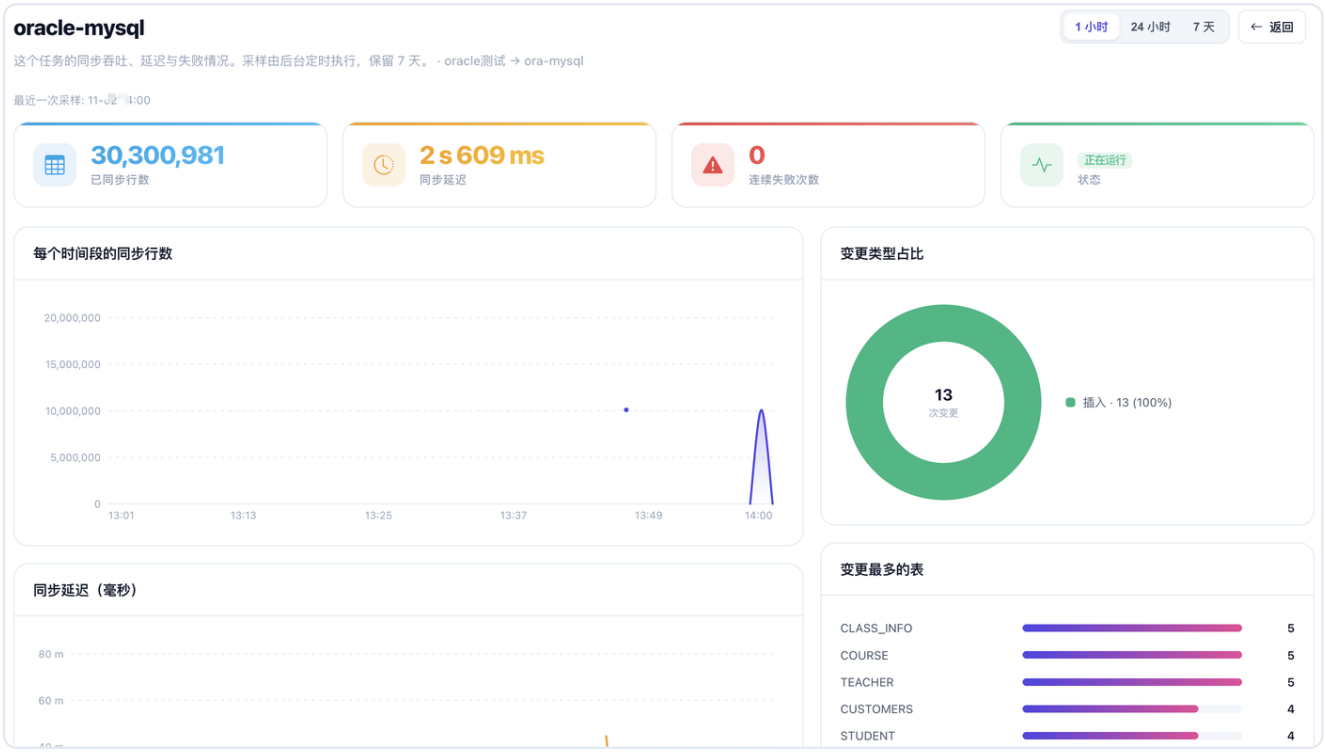


图 4–5　同步任务性能：同步行数、延迟、失败次数与变更构成

页面	看什么	怎么用
数据库性能	连接延迟与可用性、会话数、QPS、缓存命中率、锁等待、表数 / 行数 / 占用空间	在连接列表点「性能」进入；切换 1 小时 / 24 小时 / 7 天，对齐业务高峰判断容量
同步任务性能	同步行数、延迟、失败次数、变更构成与变更最多的表	在同步任务列表点「性能」进入；轮询任务按最近一次成功周期计算延迟

指标依赖账号权限

部分产品没有目录统计视图，容量列会显示「—」，其余指标不受影响。采样会在被监控库上打开会话，不需要时请管理员关闭 `metrics.enabled`。

接入 Prometheus

管理员配置 `metrics.prometheus-token`（或环境变量 `JYN_METRICS_TOKEN`）后，监控系统以 `Authorization: Bearer <token>` 访问 `GET /api/metrics/prometheus`，即可拿到进程、任务与连接三组最新样本；未配置 token 时端点返回 404，不暴露任何指标。

先看曲线再登录服务器

延迟持续上涨而位点仍在前进，通常是目标库写入跟不上；曲线平直而位点不动，则先查消费线程与租约。两者结合起来判断，比只看一个数字可靠。

4.7 同步前检查

同步任务详情页顶部（与「立即同步」同一排）有「同步前检查」按钮，仅管理员可见。它把下一轮要做的事先排练一遍：连一次源库与目标库、读一次源端元数据、与结构快照比对、检查目标表形状与轮询选项，然后把结论按严重级别列出来。

只读

预检不执行 DDL、不写结构快照、不推进游标、不写变更日志，也不触发 CDC 探测之外的任何动作；它可以反复运行，不会改变同步进度。

检查项	能提前发现什么
连通性	源库或目标库连不上、账号或网络策略变化
结构变更	下一轮将要执行的 CREATE / ALTER / DROP，以及表名映射冲突
目标表形状	目标表不存在、缺少列、类型不兼容；开启结构同步时这些会被自动创建或补齐，级别降为警告
轮询选项	没有可用的增量游标、游标列没有前导索引、表没有主键

检查项	能提前发现什么
CDC	日志 CDC 能力探测失败、基线尚未完成、位点没有记录列布局，以及源表列布局与记录位点时不一致（停机期间跑过 DDL）

结果分四组：**错误**表示下一轮会失败，需要先处理；**警告**表示能继续，但建议确认；**正常**与**跳过**分别是已通过和不适用于当前配置的检查。同一对象可能同时出现在结构变更与目标表形状两组里，这是正常的——一组说「会怎么改」，一组说「改完长什么样」。

binlog 布局漂移会直接报错

MySQL binlog CDC 依赖记录位点时的列顺序。若源表在 CDC 停机期间执行过加列 / 删列 / 改列，预检会报「源表列布局与记录 CDC 位点时不一致」。这时不要反复重试，应到 CDC 状态面板「重建基线」，让全量与位点一起重新开始。

预检不等于一次真实同步

预检比较的是源端当前结构与结构快照、目标表现状，不会真的读数据。数据量、锁竞争与目标库写入速度仍要在实际运行时观察；预检通过只说明「下一轮不会因为配置与形状问题失败」。

CHAPTER 05

结构同步

结构同步把源库的对象定义翻译成目标库方言。它是数据同步的前置条件：目标表不存在时，全量与增量都无处可写。

5.1 结构同步流程

1. **读取源库元数据**
   枚举勾选的表、列、主键、索引、视图与过程，生成元数据快照。
2. **对比目标库**
   判断对象是否存在、结构是否一致，生成差异清单。
3. **生成目标方言 DDL**
   做类型映射、默认值转换、标识符引用与索引语法转换。
4. **执行 DDL**
   按依赖顺序建表、建索引、建视图、建过程；失败的对象单独记录，不阻塞其他对象。
5. **记录结果**
   写入变更日志，展示成功与失败明细。

源端整表删除的联动删除见 3.3「允许执行删除」：Jync 只在源库账户能看见整个 schema 时，才把清单里消失的对象当作删表并删除目标对象；否则保留目标对象并告警一次，避免把授权收回误判成删表。

5.2 类型映射与建表

类型映射不是简单的名称替换，而是按目标库的能力收敛：源库允许的长度、精度、默认值，在目标库不支持时会被调整到合法范围。例如 Oracle 的 `VARCHAR2(4000)` 迁到 MySQL 时，会受字符集与行长度限制，必要时提升为 `TEXT` 或拆分。

源类型（示例）	目标：MySQL	目标：PostgreSQL	目标：达梦
NUMBER(10)	BIGINT / DECIMAL	NUMERIC(10)	NUMBER(10)
VARCHAR2(100)	VARCHAR(100)	VARCHAR(100)	VARCHAR2(100)
DATE	DATETIME	TIMESTAMP	TIMESTAMP

源类型（示例）	目标：MySQL	目标：PostgreSQL	目标：达梦
CLOB	LONGTEXT	TEXT	CLOB
BLOB	LONGBLOB	BYTEA	BLOB

上表为常见示例，实际映射由方言实现决定；以安装版本的行为为准。

建表时主键与唯一索引立即创建，因为幂等 upsert 依赖主键，唯一索引用于尽早发现源库重复数据。非唯一二级索引默认延迟到首次全量完成后再建，避免逐行维护索引拖慢加载。

5.3 索引、视图与存储过程

索引

普通索引、唯一索引与复合索引都会转换。索引名冲突时按目标库规则改名，并在变更日志中记录映射关系。大表首次全量建议保留「二级索引延后」选项。

视图

视图定义中的函数、运算符和分页语法会做方言替换。视图中引用未同步的表时，创建会失败并给出目标库的原始错误，处理方式是先补齐依赖对象。

存储过程与函数

过程体是转换中最复杂的部分。Jync 能可靠处理机械差异，例如参数写法、变量声明、异常块关键字、系统函数名；控制流、游标和动态 SQL 只能尽力转换。失败时不会写入半成品，而是进入「转换审阅」等待人工处理。

5.4 转换审阅与 AI 辅助

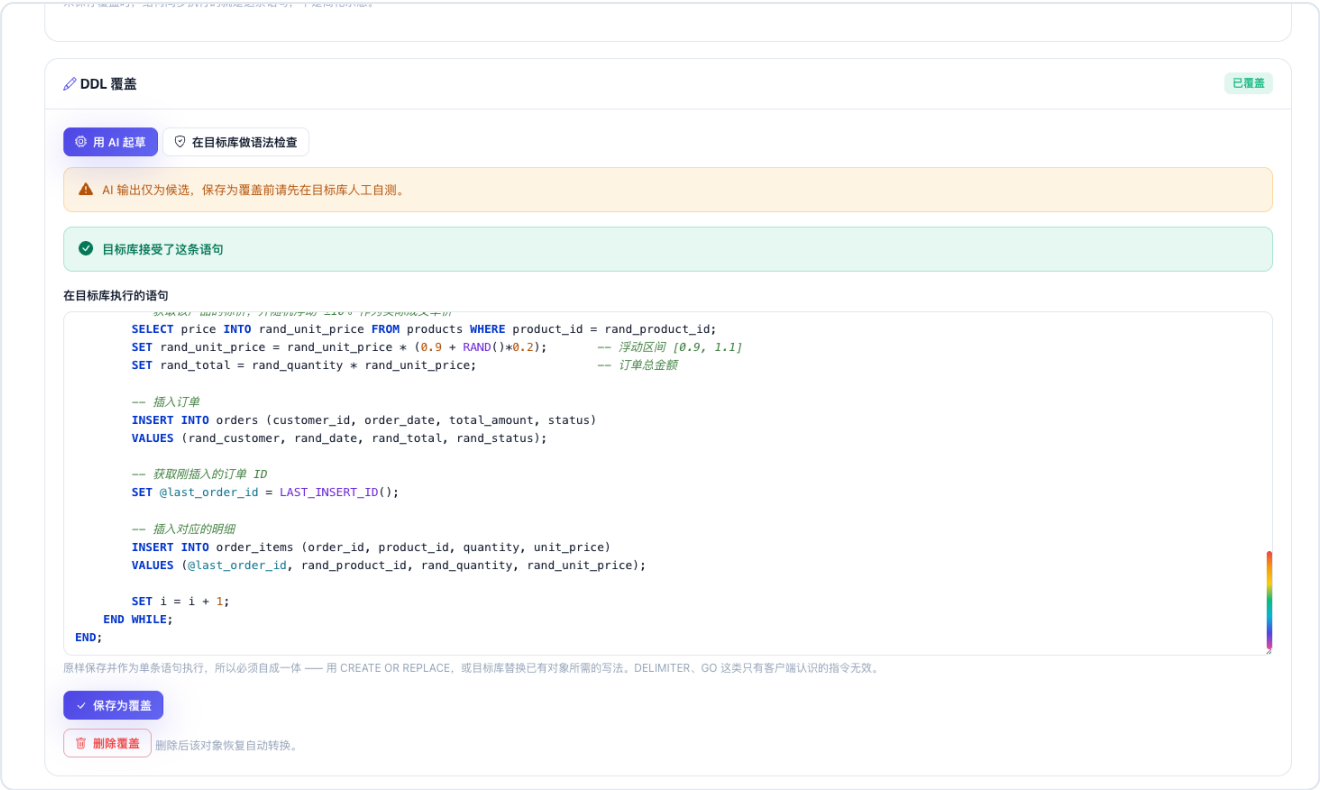


图 5–1　转换审阅：源语句、目标语句、语法检查与手工覆盖入口

转换审阅页提供三件事：查看源过程与转换结果、在目标库做语法检查、保存手工覆盖。语法检查会创建临时对象再删除；如果进程在检查过程中被强杀，可能残留 `JYNC_AI_CHECK_*` 临时对象，需要手工清理。

AI 起草

管理员可以在「AI 配置」中启用一个模型供应商。转换审阅页的「用 AI 起草」会把源过程发送给该端点，生成候选语句并做语法检查。候选**不会自动保存**，必须人工确认后点击「保存为覆盖」才会生效。

> **数据出网**
>
> 启用 AI 后，过程源码会发送到所配置的模型端点。内网或有合规要求的部署应使用本地推理端点，或保持 AI 关闭。未配置供应商时，Jync 不会产生任何外部请求。

审阅与覆盖的操作顺序

1. **看差异**
   对照源语句与目标语句，重点确认控制流、游标与动态 SQL 是否被正确改写。

2. **跑语法检查**
   在目标库创建临时对象验证语法。检查通过只代表能执行，不代表逻辑等价。

3 **人工修订**

在覆盖框中直接修改目标语句，保存后该对象以覆盖版本为准。

4 **重新执行结构同步**

覆盖只影响下一次执行，确认无误后再启动同步。

> **覆盖的边界**
>
> 覆盖按对象保存，不会自动合并源端后续的修改。源过程再次变更时，应重新审阅并更新覆盖，否则目标库会停留在旧版本。

CHAPTER 06

全量同步

全量同步把源表当前的全部数据写入目标表。它是增量同步的起点，也是「重建基线」的核心步骤。Jync 的全量不是一次性长事务，而是分块提交、可中断、可续跑的过程。

6.1 全量如何执行

确定范围	切分区间	批量读取	清洗与写入	提交检查点
读取主键边界与总行数	按主键区间或采样边界切分	按 fetch-size 拉取源数据	batch-size 一批 upsert	每块提交后记录进度

无主键表无法切分，只能单流顺序读取；这类表也不保证幂等，重放可能产生重复行，平台会在同步任务页告警。

6.2 分块、检查点与并行

参数	默认	作用	调整建议
`sync.chunk-size`	100000	每块行数，每块提交并写检查点；小于等于 0 关闭分块	大表保持默认；网络不稳可调小
`sync.full-load-batch-size`	2 000	每个 JDBC 批次的行数，1–20 000	建议 1000–5000，LOB 表 500–2000
`sync.full-load-fetch-size`	2 000	源库结果集每次拉取行数，1–20 000；MySQL 源库按行流式读取	与批次保持同一量级
`sync.full-load-parallelism`	2	单表并行 worker 数，1–8	有主键表都可切分；生产库可设 1
`sync.max-pool-size`	5	每个数据库连接的池上限	并行度受此约束，见下文

并行度不是想设多少就设多少。服务端会按连接池收敛：源与目标为不同连接时，上限为 `max-pool-size − 1`；源与目标是同一个连接配置时，上限为 `(max-pool-size − 2) / 2`。反过来，想稳定达到并行度 p，不同连接需满足 `max-pool-size ≥ p + 1`，共用连接需满足 `max-pool-size ≥ 2 × p + 2`。最终值至少为 1，并记录在启动日志中。

同步任务级覆盖

「全量加载性能」卡片可以为单个同步任务设置批次大小与并行度，优先级高于全局配置。CDC 同步任务的基线全量同样使用这组值。

6.3 观察吞吐与负载

每个数据拷贝阶段结束会输出一行日志，包含表名、线程、行数、实际变更行数、耗时、行/秒、读/写/提交占比、批次数与每批行数。通过它可以直接判断瓶颈在源库读取、目标库写入还是提交。只有单次扫描至少 1000 行且确有变更时才记 INFO；不足 1000 行的窗口/尾部与没有变更的轮次记为 DEBUG（默认 INFO 级别下不可见）。

```
table=ORDERS thread=full-load-ORDERS rows=100000 changed=100000 elapsed=41.2s rows/s=2427
read=18% write=74% commit=8% batches=50 rowsPerBatch=2000
```

观察	可能含义	处理
write 占比高	目标库写入是瓶颈	加大批次减少往返；检查目标库磁盘与索引
commit 占比高	提交本身慢：redo / 存储 / 备库同步	加大批次，用更少、更大的提交；检查存储延迟
read 占比高	源库读取慢或网络带宽不足	加大 fetch-size；检查源库负载与网络
rowsPerBatch 远小于设定	批次被提前拆散	检查驱动是否支持批量；查看目标库错误
并行后总吞吐未提升	目标库写入能力或连接池成为瓶颈	降低并行度，或提高连接池与目标库能力

不要压垮生产库

并行度与批次是双刃剑。与在线业务共享的数据库建议并行度 1、批次 1 000–2 000，并在低峰期执行首次全量。

6.4 中断、续传与重跑

全量中途停止或进程崩溃后，检查点保存在元数据库中。下次执行时，平台先读取上一次实际使用的区间布局，再判断能否原样续跑：

- **布局自洽**：区间从 0 开始连续编号、首尾相接、边界类型一致（全部按数值切分或全部按键元组切分），就按记录下来的区间继续，不重新计算切分。
- **源表在两轮之间增长**：新增行落在尾部——末尾区间还没完成时把它上界就地扩大到新的 MAX；末尾区间已完成时追加一个新区间。已完成的区间不会重读。
- **中间有区间没写入任何行**：下一轮用相邻两个区间之间的空隙补一个区间，只补缺口，不整表重来。
- **并行度变化**：区间布局是上一次记录下来的，并行度只决定同时跑几个区间，不改变切分，因此改并行度也能续跑。

- **布局无法信任**：记录前后矛盾、边界类型混用、编号不从 0 开始，或来自更早版本、没有区间信息的检查点，才丢弃旧检查点从头加载。
- 执行过「重置进度」或「首次加载前清空目标表」：检查点一并清除。
- 目标表丢失、由 Jync 自动重建：该表的游标、检查点与「已完成首次加载」标记一并重置，本轮按首次加载重灌——否则会在空表上按增量窗口跑，水位线以下的行永远补不回。

因为写入是幂等 upsert，重跑已完成的块不会产生重复数据，只会多消耗一些读时间。这也是 Jync 敢于在失败后直接重跑的原因。

> **续跑与水位线**
> 续跑时完成游标会回退到最早一次尝试的水位线，之后由增量轮询把区间之外的行重新送一遍，依靠幂等写入去重。因此「续跑」不会跳过失败前后那段时间的变更；代价是下一轮增量可能多读一些已同步过的行。

CHAPTER 07

增量同步与日志 CDC

全量给出一个基线，增量负责把基线之后的变更追上来。Jync 提供两种增量方式：零侵入的轮询，以及基于数据库日志的 CDC。两者可以在同一个同步任务上切换，但不能同时启用。

7.1 增量轮询策略

轮询只执行一条查询：`WHERE 游标列 > 上次游标 AND 游标列 <= 本次水位`。因此一行能否被同步，完全取决于游标列的值有没有增长。

策略	触发条件	新增	更新	删除
TIMESTAMP	存在「最后修改时间」命名约定的时间类型列	✓	✓（该列真被更新时）	—
ROW_SCN	Oracle 且建表带 ROWDEPENDENCIES	✓	✓	—
IDENTITY	存在「创建时间」约定列或单列数字主键	✓	—	—
FULL_COMPARE	无可用游标列、有主键且行数 ⩽ 上限（自动）；也可在任务级手动选择	✓	✓	✓（开启「同步行删除」且行数 ⩽ 上限时）
NONE	无可用游标列且表过大	—	—	—

命名约定按包含匹配、大小写不敏感：`UPDATE_TIME`、`UPDATED_AT`、`LAST_MODIFIED` 等判为 TIMESTAMP；`CREATE_TIME`、`CREATED_AT` 等判为 IDENTITY，但列类型必须确实是时间类型。

> **不能用 id 或 create_time 当更新游标**
>
> 这两类列在 UPDATE 时不会变大，用它做游标意味着更新永远同步不到。如果业务表只有创建时间，请增加一个真正会变化的更新时间列，或切换到日志 CDC。

时间戳水位的安全回退

数据库事务的提交顺序可能与时间戳顺序不一致。Jync 在读取水位时回退 `safety-lag-ms`（默认 1000ms），把晚提交的事务重新纳入窗口；重复读取由幂等 upsert 吸收。

7.2 选择与指定游标列

自动识别失败或识别错误时，可以在「各表同步进度」表格的「设置游标」中手动指定列。手动指定的优先级高于自动识别。

- 指定时间类型列：按 TIMESTAMP 处理，能发现新增与更新。
- 指定数字列：按 IDENTITY 处理，只能发现新增，页面会显示警告。
- 指定不存在的列：保存时拒绝。

> **推荐做法**
> MySQL 可以增加 `update_time DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP`；Oracle 可以用触发器或应用层统一维护更新时间列。一次改造，长期受益。

7.3 切换到日志 CDC

1. **停止同步任务**
   运行中的同步任务不能切换捕获策略。
2. **选择「日志 CDC」**
   页面会展示源库探测结果，逐项列出前提是否满足。
3. **处理不满足项**
   按提示开启日志、补充权限或调整参数；探测不通过时无法启用。
4. **保存并启动**
   启动后平台先记录位点，再执行全量基线，最后进入流式回放。
5. **观察状态面板**
   确认位点推进、落后秒数下降、吞吐稳定。

> **切换即重做基线**
> 从轮询切到 CDC 会重新执行一次全量。这是为了保证位点之前的数据完整；全量期间源库的变更已经记录在位点之后，不会丢。

7.4 日志 CDC 的源库准备

Jync 按源库的方言家族放行日志 CDC，保存时由页面逐项探测。同一家族里的产品并不等能力：MySQL 兼容产品（GBase、OceanBase、TiDB、GoldenDB、TDSQL、PolarDB 等）与 PostgreSQL 方言家族（Kingbase、HighGo、Vastbase、openGauss 等）都按实际能力放行，探测不过时会明确说明缺的是哪一项，而不是笼统拒绝。

MySQL 兼容家族（binlog）

要求 MySQL ⩾ 5.7（MariaDB 不支持），并满足以下条件：

- `log_bin=ON`、`binlog_format=ROW`、`binlog_row_image=FULL`。
- 账号具备 `REPLICATION SLAVE` 与 `REPLICATION CLIENT`。
- Jync 使用的假 replica `server_id` 不与源库或真实从库冲突；实际值为 `server-id-offset + 项目ID`。
- binlog 保留时间 ⩾ 24 小时，避免停机期间位点被清理。

```
-- 检查
SHOW VARIABLES LIKE 'log_bin';
SHOW VARIABLES LIKE 'binlog_format';
SHOW VARIABLES LIKE 'binlog_row_image';
SHOW MASTER STATUS;
-- 授权
GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'jync'@'%';
FLUSH PRIVILEGES;
```

Oracle LogMiner

要求 Oracle ⩾ 11g，且：归档模式开启、库级补充日志开启、相关表开启 ALL COLUMNS 补充日志、账号具备挖掘权限、表中无 LONG 列、归档保留 ⩾ 24 小时。CDB 环境必须以公共用户（`C##`）在 CDB$ROOT 中挖掘。

```
-- 归档模式（需停机）
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
-- 补充日志
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
ALTER TABLE APP.ORDERS ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
-- 授权
GRANT CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE,
      FLASHBACK ANY TABLE TO jync;
GRANT EXECUTE ON DBMS_LOGMNR TO jync;
-- 12c 及以上可授予 LOGMINING 角色
GRANT LOGMINING TO jync;
```

达梦 DM8（LogMiner）

要求 DM8 开启归档模式，且 `RLOG_APPEND_LOGIC=2`——只有 2 会在 redo 里写完整旧镜像，否则 UPDATE / DELETE 无法还原。达梦只能挖掘归档文件，CDC 延迟因此取决于归档切换的间隔；基线起点取最新归档的 `NEXT_CHANGE#`，没有任何归档时会明确提示先切换日志，而不是反复重试。表中同样需要 ALL COLUMNS 补充日志。

```
-- 归档模式（dm.ini：ARCH_INI=1 并配置归档）
SELECT ARCH_MODE FROM V$DATABASE;
-- 旧镜像写满（静态参数，修改后需重启实例）
SP_SET_PARA_VALUE(1, 'RLOG_APPEND_LOGIC', 2);
-- 授权
GRANT CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE TO jync;
GRANT EXECUTE ON SYS.DBMS_LOGMNR TO jync;
-- 切换日志，产生可挖掘的归档
ALTER SYSTEM SWITCH LOGFILE;
```

PostgreSQL 方言家族（逻辑解码）

要求 PostgreSQL ≥ 9.4，`wal_level=logical`，账号具备 REPLICATION 权限，且表的 `REPLICA IDENTITY` 不是 `NOTHING`。openGauss 等 PostgreSQL 方言产品按能力选择函数：有 `pg_current_wal_lsn()` 的按 WAL 位点，只有 `pg_current_xlog_location()` 的（如 openGauss 5.0）用旧函数名，探测时逐条真跑消费链路语句，建槽探测是唯一权威。

> **复制槽会滞留 WAL**
> 目标库停机或写入极慢时，槽不前进，源库 WAL 会一直堆积。PG 源的状态面板每 30 秒采样一次槽滞留量（基线期间也不例外），超过 `sync.cdc.pg.wal-retention-warn-bytes`（默认 4 GiB）标红并写 WARN 日志。openGauss 没有 `pg_lsn`，这一行不显示。

```
-- postgresql.conf 或 ALTER SYSTEM
ALTER SYSTEM SET wal_level = logical;
-- 重启后检查
SHOW wal_level;
-- 账号与权限
CREATE USER jync WITH REPLICATION LOGIN PASSWORD '******';
GRANT CONNECT ON DATABASE appdb TO jync;
GRANT USAGE ON SCHEMA public TO jync;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO jync;
-- 无主键表需要 FULL 身份
ALTER TABLE public.orders REPLICA IDENTITY FULL;
```

SQL Server CDC

要求 SQL Server ≥ 2012，数据库级 CDC 已启用，SQL Server Agent 正在运行，账号属于 `cdc_Admin` 或 `cdc_reader`，且每张表已建立 capture instance。

```
-- 数据库级启用
USE appdb;
EXEC sys.sp_cdc_enable_db;
-- 表级启用
EXEC sys.sp_cdc_enable_table
     @source_schema = N'dbo',
     @source_name   = N'orders',
     @role_name     = NULL;
-- 检查
SELECT name, is_tracked_by_cdc FROM sys.tables;
```

崖山 YashanDB（YStream）

崖山走自带的逻辑日志 YStream（`DBMS_YSTREAM_ADM`），不建触发器、不改表结构。要求：崖山实例、归档模式开启且为主库、账号具备 `YSTREAM_CAPTURE` 与 `SELECT ANY DICTIONARY`（后者用于读取 `V$YSTREAM_SERVER` 管理视图），每张同步表开启 ALL COLUMNS 补充日志并有主键。实测崖山 23.4.7.100 没有 `DBMS_LOGMNR`，崖山源请选 YStream，不要选 LogMiner。

```
-- 归档模式与主库角色
SELECT LOG_MODE, DATABASE_ROLE FROM V$DATABASE;
-- YStream 管理视图可读（探测以此判断能力）
SELECT COUNT(*) FROM V$YSTREAM_SERVER;
-- 授权
GRANT SELECT ANY DICTIONARY TO jync;
GRANT YSTREAM_CAPTURE TO jync;
-- 逐表补充日志，缺了 DML 会静默丢失
ALTER TABLE APP.ORDERS ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

> **源侧服务由 Jync 托管**
>
> YStream 的源侧服务在创建同步任务、追加表与启动时由 Jync 创建，位点是提交 SCN、实例、组 LSN、组偏移与行号的五元组；删除同步任务或切换策略时尽力停止并删除服务。停止前请确认该服务只被当前任务使用。

7.5 CDC 状态面板

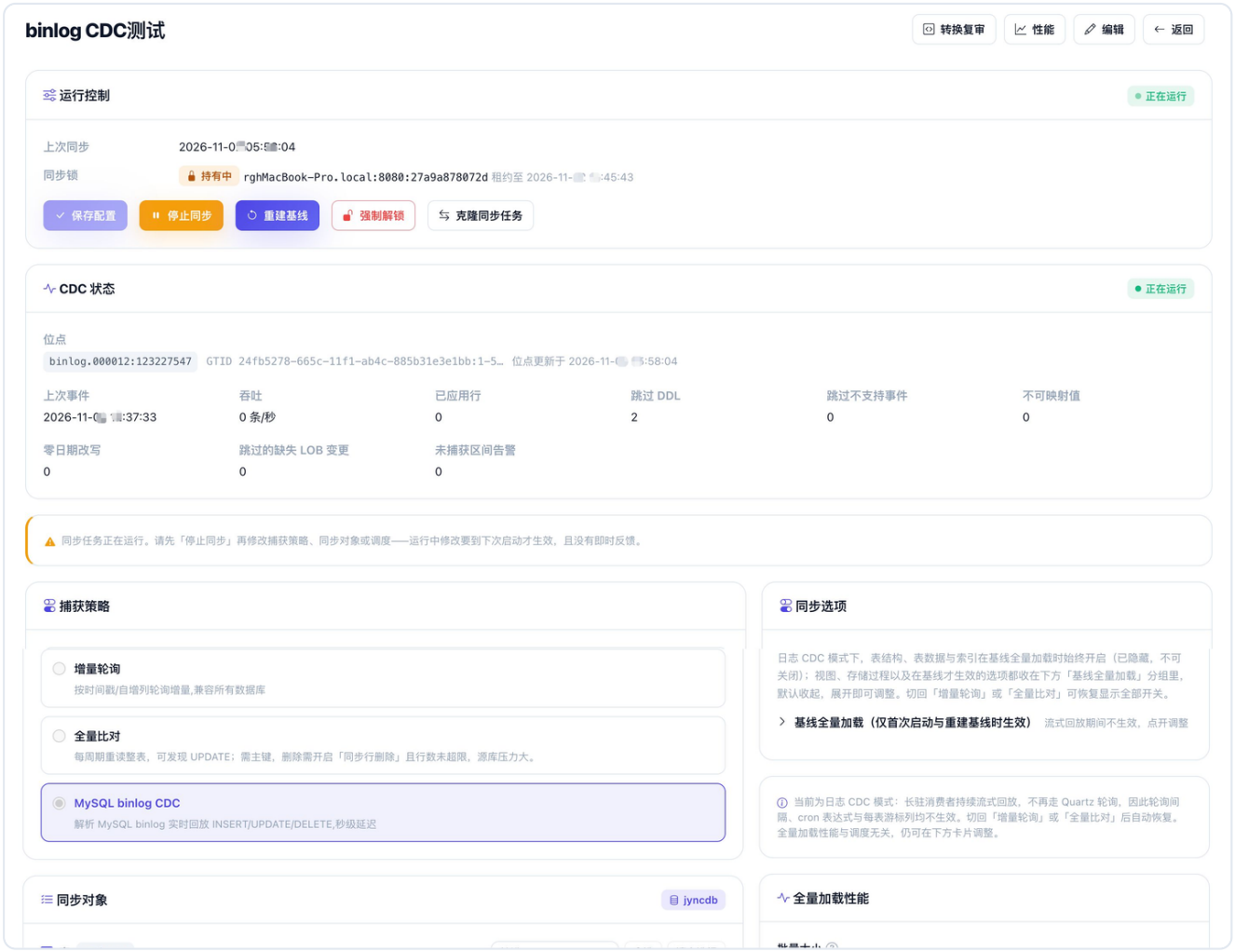


图 7–1　CDC 状态面板：位点、落后秒数、吞吐、跳过计数与捕获策略

指标	含义	关注点
位点	当前已应用的日志位置（binlog / SCN / LSN）	持续前进即正常
落后	源库当前位点与已应用位点的秒数差	稳态应接近 0；持续增长需排查
吞吐	每秒应用的变更条数	与源库写入量对比
已应用行	累计应用的 upsert / delete 数	重启后重新计数
跳过 DDL	未捕获的 DDL 数量	大于 0 时应检查结构差异
跳过不支持事件	引擎返回但无法映射的事件	少量可接受，持续增长需排查
不可映射值 / 零日期	类型转换失败或零日期被改写	核对目标表数据

变更日志在 CDC 期间按表聚合，每张表每 10 秒最多一条「N upserts / M deletes（截至位点）」，停止同步时补刷最后一个窗口。这样避免高频写入把日志表撑爆，同时保留可读的进度记录。

v1.10.0 起位点同时保存每张表的列序指纹。消费者停机期间源表执行过 DDL 时，重启会停在「列布局漂移」上并提示重建基线，面板的「跳过 DDL」与错误摘要会直接指出受影响的表。

一个源事务在目标端按一笔提交，事务内的变更先在内存里折叠：同一主键只保留最后一次镜像，删除与写入互相抵消。折叠按主键索引完成，代价随行数线性增长，因此批量 `REPLACE` / `MERGE` 或数据修复不会因为单个事务很大而拖住整条链路。

7.6 重建基线

以下情况会导致位点失效，同步任务被置为 ERROR：

- MySQL binlog 被 PURGE，位点文件已不存在。
- Oracle 归档日志被删除或备份后清理。
- PostgreSQL WAL 被回收，复制槽失效。
- SQL Server CDC 的 capture instance 被重建。

处理方式是点击「重建基线」：平台清除旧位点，重新记录当前位点并执行全量，随后从新位点继续回放。重建期间目标库会有一次全量写入压力，建议在低峰期执行。

> **不要直接重置进度**
> CDC 同步任务应使用「重建基线」而不是「重置进度」。前者会重新记录位点，后者只清游标，可能留下不匹配的旧位点。

7.7 多节点与租约

CDC 消费线程是长驻的，不经过 Quartz 的每次调度，因此它使用一套独立的短租约来保证同一同步任务只有一个节点在消费。持锁节点定期续租；节点宕机后，其他节点在租约到期后的巡检周期内接管。

配置键	默认	作用
`sync.cdc.lock-ttl-ms`	30 000	CDC 消费者租约时长
`sync.cdc.supervise-interval-ms`	10 000	各节点巡检并抢占过期租约的周期
`sync.cdc.reconnect-backoff-ms`	1 000	断线重连初始退避，逐次翻倍
`sync.cdc.reconnect-backoff-max-ms`	60 000	退避上限
`sync.cdc.change-window-ms`	10 000	变更日志按表聚合窗口，0 关闭聚合

多节点部署时务必保证节点间时钟同步。租约以数据库时间为准，但如果节点本地时间偏差过大，日志时间与「落后秒数」的展示会失真。接管期间可能出现最多一个租约周期（约 30–40 秒）的暂停，位点不会丢失，恢复后继续回放。

CDC 链路自检

检查项	正常表现	异常时的处理
位点推进	「已应用位点」持续前进	先确认源库日志仍保留，再检查账号读取权限
消费租约	同一同步任务只有一个节点显示「消费中」	多节点同时消费时检查时钟同步与元数据库连通
回放延迟	落后秒数在秒级波动	持续上涨说明目标写入跟不上，降低批次或错峰
错误计数	面板无新增错误	出现 ERROR 先看日志中的 SQL 与主键，再决定重试或重建
变更聚合	变更按窗口成批出现	单表变更过密时调大 `change-window-ms`

租约与调度锁不同

Quartz 调度锁只覆盖一次执行；CDC 租约覆盖整个消费线程的生命周期。停止同步任务时两种锁都会释放，但强制解锁只释放数据库锁行，不会终止已经运行的消费线程。

先看位点再看延迟

位点不前进时，延迟数字没有意义；先确认消费是否被阻塞，再判断是否只是回放偏慢。

CHAPTER 08

ETL 任务与批次

同步任务保证「两边持续一致」，ETL 任务负责「按你写的 SQL 取数，再写入目标表」。两者在「同步任务」页面分区管理：上面是 ETL 批次与 ETL 任务，下面是普通同步任务。ETL 任务不改源表结构、不比对两侧数据，只把查询结果复制过去。

8.1 新建 ETL 任务



图 8–1　新建 ETL 任务：取数 SQL、目标表与写入方式，可载入 .sql 文件

1 **选择连接**

源连接可以是任意 SQL 数据源，也可以选 MongoDB 原生连接；原始 MongoDB 永远只能作为 ETL 源，不能作为目标。

2 **填写取数 SQL**

在「取数 SQL」框直接编辑，或点「载入 .sql 文件」从本地导入。多语句脚本用分号分隔、按顺序执行，最后一条必须是返回数据的 `SELECT`，它的列名就是目标表的列名，表达式请起别名。

3 **指定目标表**

可写表名或 `schema.表名`，也可以从目标连接的表清单中挑选。

4 **选择写入方式**

覆盖、追加或范围覆盖，见 8.2。

5 **预检并保存**

保存前逐项检查取数、目标表、主键与 SQL 本身，问题在保存前暴露，而不是等运行失败。

配置项	作用
名称 / 描述	页面展示用，建议写清取数口径与目标表
源连接 / 目标连接	源只读；目标需要可写，且支持 ETL
取数 SQL	页面编辑或载入 `.sql`；只接受 SQL 源
目标表	不存在时按结果列建一次，之后不再自动改结构
写入方式	覆盖 / 追加 / 范围覆盖
断点续跑 key	指定唯一且单调的列后，追加与范围覆盖可续跑
预执行 / 后置 SQL	`pre-sql` 取数前执行，`post-sql` 装载后执行
查询超时	单次取数与写入的超时上限

> **目标表结构由你负责**
>
> Jync 只在目标表缺失时按查询结果建一次，之后不再修改；也不保证目标表与源表的数据一致。需要「两边持续一致」的场景请用同步任务，ETL 只负责按 SQL 复制结果行。

8.2 写入模式与断点续跑

写入方式	行为	适用场景
覆盖	先清空目标表，再写入本次查询结果	全量重取，结果集就是目标表的完整内容
追加	不清除已有数据，直接写入	日志型数据，或查询本身已按条件只取新增部分
范围覆盖	按 `range condition` 只删除指定范围，再写入	按天 / 按月重跑，其余历史数据保留

- **断点续跑**：指定唯一且单调的 `resume key` 后，追加与范围覆盖在中途失败时从上次位置继续，不重复写；覆盖模式每轮从空表开始，不需要续跑键。
- **前后置 SQL**：`pre-sql` 适合清表、建临时表；`post-sql` 适合汇总、改状态、清理临时对象。

- **多语句脚本**：分号分隔、按顺序执行，前面语句的报错会使本次任务失败，不会只跑一半还报成功。

8.3 ETL 批次

同源同目标的多张表建议组成一个批次：共享一对连接与一条 cron，组内按顺序执行，单个成员失败不中断后续，批次页汇总每次运行的成败与行数。



图 8–2　新建 ETL 批次：一个批次共享一对源 / 目标连接与一条 cron

1. **新建批次**

   填写名称、描述、源连接、目标连接、Cron 表达式与查询超时，然后勾选要纳入的数据表。

2. **生成成员**

   勾选的表按「`select *` + 覆盖 + 目标表 = 表名」生成默认成员，之后逐个改成自己的取数 SQL 与写入方式。

3. **启用与运行**

   批次默认停用；启用后按 cron 顺序执行，也可以随时点「运行批次」。

4. **维护成员**

   在批次页「添加数据表」追加成员，或把成员移出批次；移出会清空检查点，同一批次内两张成员不得写同一张目标表。

5. **删除批次**

   连带删除成员任务与检查点。

- **连接创建后锁定**批次创建后源 / 目标不可更改，防止成员被改挂到别的库；需要换库请新建批次。
- **成员不再单独调度**批次内的任务不占用自己的 cron，成员页只保留「立即运行」。

> **顺序执行会拉长总时长**
>
> 一批几十张大表顺序跑，单批可能很久；需要并行的表请拆成多个批次。

CHAPTER 09

数据校验与故障恢复

9.1 行数审计

平台按 `row-count-audit-interval-ms`（默认 60 秒）对已同步的表做一次双边行数核对：源库游标覆盖的行数与目标库实际行数比较。发现差异时写入告警日志，并在同步任务页提示。

行数审计能发现「同步漏行」与「目标库被外部删除」两类问题，但无法发现同数量的内容差异；内容差异需要用全表比对或业务校验。

需要立即确认、或想看清每张表差多少行时，用 9.2 的手动核对；它执行精确 `COUNT(*)`，不等待下一个自动审计周期。

9.2 手动核对行数



图 9–1　核对行数：逐表比较源 / 目标 COUNT(*)，差值与状态一目了然

「各表同步进度」卡片右上角提供**「核对行数」**。与 9.1 按周期在后台运行的自动审计不同，它由管理员手动触发，逐表串行执行，结果只保留在当前弹窗里。

1. **打开并确认**点「核对行数」后先出现确认框，说明将对多少张表在源 / 目标各执行一次 `COUNT(*)`。点「开始核对」后才发起查询；页面加载时不会自动查询。

2. **查看结果**弹窗列出对象、源行数、目标行数、差值、累计同步与状态。差值为源行数减目标行数；累计同步是该表历史写入总量，与当前行数不是一回事。
3. **单行重查**某张表刚补完数据、想立刻再看一次时，点该行的「重查」即可，不必整批重跑。
4. **随时停止**「停止」会结束当前队列；已完成的表保留结果，未执行的表不再发起查询。

状态	含义
一致	两侧行数相同。只说明总量一致，不代表每一行内容相同。
不一致	两侧行数不同，差值列给出差额；先查变更日志与目标库的外部写入。
缺失	该表在源库或目标库不存在（常见于目标表尚未自动创建）。
超时	单侧查询超过 15 秒。`COUNT(*)` 在大表上会扫描整表，建议在业务低峰执行。
异常	连接失败、权限不足或其他数据库错误；同一张表的另一侧结果仍会保留。

大表会占用数据库 IO

核对是精确 `COUNT(*)`，不是估算。千万行级表可能触发全表 / 全索引扫描；单侧超过 15 秒会标记为「超时」，不会无限等待。业务高峰请避开，或先只核对小表。

串行执行

核对任务逐表串行，不会并发压库；同一时刻只会有一个核对在跑。访客账号看不到入口，接口也仅管理员可调用。

9.3 全表比对

有主键的小表可以启用 FULL_COMPARE 策略：每个周期读取源表全部行，按主键 upsert 到目标库，因此能看到 UPDATE。开启「同步行删除」且表行数不超过 `full-compare-max-rows`（默认 20 000）时，才会比对主键并删除目标库中源库已不存在的行。自动选择只在行数不超过该上限时启用；在同步任务级手动选择「全量比对」不受上限约束，但每个轮询周期都会整表扫描。详情页与启动日志会列出超限、或按当前轮询间隔负担过重的表，并给出建议间隔（按约 1,000 行/秒的整表读写速度估算，实际以数据库性能为准）；当前间隔已经长于估算所需时，只提示超限，不再给出更短的建议。无主键表只做一次种子加载，之后跳过。

能力	增量轮询	全量比对	日志 CDC
发现新增	✓	✓	✓
发现更新	依赖游标列	✓	✓
发现删除	—	仅开启「同步行删除」且行数 ≤ 上限	✓
源库压力	低	高	低（读日志）
延迟	轮询间隔	轮询间隔	秒级

9.4 停止、重置与重建

操作	清除游标	清除检查点	清除 CDC 位点	下次行为
停止同步	否	否	否	从当前进度继续
重置进度	是	是	否	全量重载后继续轮询
重建基线	是	是	是（重新记录）	记录新位点 → 全量 → 回放
强制解锁	否	否	否	仅释放锁，不改变进度

强制解锁的前提

只有在确认没有任何节点在执行该同步任务时才使用。如果另一个节点仍在写目标库，强制解锁会导致同一同步任务被并发执行。先检查各节点日志与数据库锁行。

灰色「上一轮残留（已过期）」不是锁

已过期、且本实例没有轮次在跑的租约只是上一轮留下的行，下一次取锁会直接接管，页面不提供强制解锁，也不需要处理。只有过期但本地仍有轮次持有时，才显示红色与强制解锁——那种情况说明本轮丢了租约，会在下一个安全点自行退出。

恢复操作顺序

1. **确认差异范围**

   先看行数审计与变更日志，判断是漏行、错行，还是目标库被外部修改。

2. **停止同步**

   停止同步任务并等待当前批次结束，避免恢复过程中仍有写入。

3. **选择恢复动作**

   位点失效用「重建基线」；仅游标异常用「重置进度」；怀疑锁残留才用「强制解锁」。

4. **小范围验证**

   恢复后先核对一张小表的行数与关键字段（可用「各表同步进度 → 核对行数」，见 9.2），再观察一个审计周期。

5. **记录处置**

   在运维记录中写明原因、动作与验证结果，便于复盘。

先取证再修复

恢复动作会改变游标与位点，执行前先导出相关日志片段；否则问题复现时已经没有现场。

9.5 内容级校验

行数审计只能发现「目标比源少」，发现不了行数相同但内容不同——更新丢失、字段截断、字符集或类型转换错误都会表现为行数一致。内容级校验在每张表增量同步成功后，按主键抽样，把源库与目标库的行值逐列比对，补上这块盲区。

何时自动运行

默认每 15 分钟最多一次、每表抽 50 行、单表源行数不超过 200 万、每个项目每周期最多校验 1 张表、单次查询 60 秒超时。抽样不是取前 N 行，而是按主键范围分桶后每桶取一个键，避免只看到某一时间段的数据。配置项见 `sync.content-audit-*`，把 `sync.content-audit-interval-ms` 设为 `0` 可完全关闭自动校验。

哪些表会被跳过

情况	原因
首次全量尚未完成	目标端还没有完整的可比数据
表没有主键，或游标是伪列	无法在目标端定位同一行
主键列配置了清洗规则	键值已被改写，无法按原值查目标行
游标列没有索引且表较大	抽样会退化为全表扫描，受 `max-unindexed-scan-rows` 保护
源行数估计超过上限	默认 200 万行，避免给大表增加额外负担

LOB、JSON、XML、几何等无法可靠比较的列会自动排除，并在结果中给出排除列数；数字、时间、布尔会先规范化再比较，时间按实际时刻对齐，减少时区与精度带来的误报。

手工校验单表

管理员可在「各表同步进度」中对应表行点「校验内容」。手工触发不受 15 分钟间隔与每周期表数限制，但上面的资格条件仍然生效；不满足时直接返回「已跳过」并给出原因，不会创建进度行，也不会替该表补一次同步。

怎么看结果

结果分 **一致**、**内容不一致**、**已跳过**、**失败**，并给出抽样行数、一致行数、不一致行数、目标缺失行数与存在差异的列名。界面只显示计数与列名，不显示行值，避免把业务数据带到页面上。

> **只报告，不修复**
>
> 发现差异时，Jync 写一条变更记录并发出 `DATA_DRIFT` 告警，不会自动改写目标行。目标端若有外部触发器、手工修订或转换规则，也会表现为差异；请先确认来源，再决定重跑、重置进度还是修正目标数据。同一张表、同一组差异列在短时间内只告警一次，避免刷屏。

CHAPTER 10

数据预览、克隆与列清洗

10.1 只读数据预览

同步任务详情页提供源表与目标表的只读预览，同时列出列结构与候选游标列。预览只执行 SELECT，不会写入任何数据；访客账号看不到入口，管理员可以查看前若干行用于核对。

预览的用途是快速回答三个问题：目标表是否已建、数据是否写进去、时间字段是否正确。它不替代专业客户端，复杂查询仍需在数据库侧执行。

只看「两边行数是否一致」时，用第 8 章的**「核对行数」**更快：无需逐页翻数据，直接给出源 / 目标 `COUNT(*)` 与差值。

10.2 同步任务克隆

从一个已有同步任务复制全部同步设置创建新同步任务，目标连接可以另选。适合「同一源库同步到多套目标」或「生产配置复制到测试环境」。

- 复制内容：对象选择、同步选项、捕获策略、调度、列清洗规则、性能参数。
- 不复制：运行状态、游标、检查点、变更日志。
- 强制关闭：「首次加载前清空目标表」，避免克隆配置误清目标数据。

10.3 列清洗规则

清洗在写入目标库之前生效，按表按列配置，只作用于字符串列；数值、日期与二进制列原样写入。

规则	效果	示例
TRIM	去除首尾空格	`" abc "` → `"abc"`
UPPER / LOWER	统一大小写	`"Abc"` → `"ABC"`
空串转 NULL	把空字符串写成 NULL	`""` → `NULL`
去前缀 / 去后缀	删除指定前后缀	`"CN-001"` → `"001"`
正则替换	按 Java 正则替换	`\s+` → 单个空格

正则替换的替换串在保存时即按 Java 语义校验，支持 `$n` 分组引用、`${name}` 命名分组与反斜杠转义；非法表达式直接拒绝，不会等到同步周期才失败。

修改时机

运行中的同步任务需要先停止同步才能修改清洗规则。规则变更不会回溯已经写入的数据，只影响之后的写入。

CHAPTER 11

专属迁移

专属迁移用于把一整套对象一次性复制到另一套系统：Elasticsearch / OpenSearch → Easysearch、MongoDB → MongoDB、MongoDB → KingbaseES。它使用独立的专属连接与专属任务，不占用数据库连接、同步任务或 ETL 的位置。

11.1 入口与概念

左侧菜单「专属迁移」进入任务首页，三张卡片对应三种可用类型；「专属连接」管理迁移专用的连接。每种任务只接受对应类型的源 / 目标连接，其他类型的连接不会出现在下拉中。

概念	说明
专属连接	按产品类型保存端点与凭据，密码 / API Key 加密存储，编辑时不回显；与「数据库连接」互不共用
专属任务	一对源 / 目标连接 + 对象范围 + 选项 + 调度；可立即运行，也可按 cron 定时
运行记录	每次运行的状态、开始 / 结束时间、对象计数与错误摘要；每个任务保留最近 50 次

11.2 新建专属连接

类型	端点	认证	额外字段
Elasticsearch / OpenSearch / Easysearch	http(s) URL	无 / Basic / API Key	—
MongoDB	mongodb:// 或 mongodb+srv://	无 / Basic（SCRAM）	数据库名（也可写在 URI 路径，两者需一致）、额外参数
KingbaseES	jdbc:kingbase8://host:port/db	仅 Basic	数据库名、schema 名、额外参数

- **密码不写进连接串**：MongoDB URI 与 Kingbase URL / 额外参数中禁止出现密码；Kingbase 还禁止 user / password 参数。凭据填在密码框，保存后加密。
- **编辑不回显**：再次编辑时密码留空表示保留原值；TLS 按各产品方式配置（MongoDB 走 URI / 额外参数，Kingbase 走驱动参数）。
- **测试连接**：保存前后都可测试；测试结果只包含产品、版本与库 / schema 等非敏感信息。
- **产品类型锁定**：连接仍被任一专属任务引用时，不能修改产品类型，避免已保存任务失效。

11.3 新建专属任务

1. 在类型首页点「新建」，选择源连接与目标连接；同一任务要求源与目标不是同一条连接。
2. 填写对象范围（索引或集合的模式，支持逗号 / 换行分隔），点「发现对象」确认匹配结果。
3. 选择目标命名方式（同名 / 前缀 / 后缀）与写入方式；不同任务类型的选项见后续小节。
4. 保存后先运行预检，确认源端可达、对象匹配、目标冲突与命名合法；预检通过再运行。
5. 需要定时执行时启用调度并填写 cron；启用 / 停用立即生效。

11.4 Elasticsearch / OpenSearch → Easysearch

- **对象发现**：按索引 glob 匹配，跳过点号开头的系统索引；没有匹配到任何索引时任务按失败处理，避免「什么都没复制却显示成功」。
- **settings / mappings / aliases**：复制前清理只读与实例相关 settings，再按目标重建；别名一并复制。
- **文档写入**：scroll 读取 + _bulk 分页写入，保留 _id 与 _routing；写入后刷新并做计数与抽样校验。
- **预检**：源端可达、_source 关闭与非兼容 mapping 类型、FAIL_IF_EXISTS 冲突、目标临时索引兼容性探测。

11.5 MongoDB → MongoDB

- **集合与命名**：按模式发现普通集合，默认跳过 system.*；目标名可选同名 / 前缀 / 后缀，默认后缀 _migrated。目标名不能为空、含 $ 或 \0、以 system. 开头或超过 120 字节。
- **复制内容**：可分别开关集合选项、索引与文档。集合选项保留 capped / size / max / validator / collation 等，索引跳过 _id_ 并去掉只读字段。
- **写入方式**：OVERWRITE 先删目标集合再按源选项重建；APPEND 保留目标已有文档，_id 冲突跳过并告警；FAIL_IF_EXISTS 遇到同名集合直接判失败且不改动目标。capped 集合按有序方式插入。
- **批量与校验**：按条数与 UTF-8 字节双重分批（默认 1 000 条 / 8 MB）；运行结束可做行数校验与按 _id 抽样比对。
- **注意**：跨库复制需要建两条连接记录（同 host、不同库名）；源与目标解析为同一个数据库时任务拒绝运行。

11.6 MongoDB → KingbaseES

- **类型推断**：按样本（默认 200 条，可调）推断列类型：ObjectId → VARCHAR(24)、字符串 → TEXT、整数 / 长整数 → INTEGER / BIGINT、Decimal128 → NUMERIC(38,10)、布尔 → SMALLINT、日期 → TIMESTAMP、二进制 → BYTEA。
- **主键**：_id 默认映射为主键且 NOT NULL；createPrimaryKey 关闭时 _id 可为空。样本中没有 _id 会在预检报错。

- **写入方式**：OVERWRITE 先删表再建表写入；APPEND 表不存在则建、存在则追加；FAIL_IF_EXISTS 遇到同名表判失败且不改动。upsertOnId 打开后使用 ON CONFLICT DO UPDATE，要求同时保留主键。
- **嵌套数据**：文档与数组写成 TEXT 的 relaxed Extended JSON，不展开为列；需要展开时请先用 ETL。
- **命名与 schema**：表名可选同名 / 前缀 / 后缀，默认同名；schema 留空使用连接当前 schema，表名与 schema 均按 63 字节上限校验。

11.7 运行、取消与历史

- **立即运行**：详情页点「运行」；同一任务在同一实例内不会并发执行，定时触发时若上一轮仍在运行，本轮直接跳过。
- **取消**：运行中可取消，取消在批间生效；已写入的数据保留。
- **定时**：启用后按 cron 触发，禁用立即停止后续触发，不影响正在运行的一轮。
- **重启恢复**：实例重启时，遗留的 RUNNING 记录标记为 INTERRUPTED；需要重新运行一次，专属迁移没有断点续传。
- **历史**：详情页展示最近 50 次运行；失败原因包含明确的对象名与错误。

11.8 权限与限制

- 查看任务与运行历史需要登录；专属连接、任务表单、对象发现与运行 / 取消等写操作仅管理员可用，全部走 POST 与 CSRF 保护。
- 专属迁移只复制运行那一刻的数据，不持续跟随源端变化；需要持续同步请使用实时同步或 ETL。
- view 与 timeseries 集合不复制；分片 key、用户 / 角色 / 认证信息不迁移；MongoDB 普通索引不会翻译成 SQL 索引。
- KingbaseES 已在 V9R1C1B25 的 PostgreSQL 与 Oracle 兼容模式下验证；V8R3 缺少 PG_CATALOG 兼容视图，与内置驱动不兼容，建议升级。

CHAPTER 12

系统管理

12.1 系统信息



图 12–1　系统信息：运行参数、项目与作者、并发与故障恢复、版本信息

页面展示运行实例、Java　版本、已调度同步任务数、快照目录，以及同步默认值（轮询间隔、批次大小、读取批量、失败阈值、全表比对上限、变更日志保留天数）。这些值来自配置文件，页面只读；修改需要编辑 `application.yml` 并重启。

12.2 AI 配置



图 12–2　AI 配置：同一时刻只启用一个供应商

支持多种协议（如 Anthropic、OpenAI 兼容、Ollama 原生），填写 Base URL、API Key 与模型名后可以测试连通性。启用后，转换审阅页出现「用 AI 起草」按钮。

- 同一时刻只能启用一个供应商，切换会停用其他供应商。
- API Key 加密存储，页面不回显明文。
- 设置 `sync.ai.enabled=false` 会整体移除菜单与接口，适合纯内网合规场景。

12.3 外部认证

「配置管理 → 外部认证」把控制台登录接入企业已有的身份体系。三种方式**同一时刻只启用一种**，共用一张配置卡：启用开关、认证方式、管理员用户名列表（逗号分隔、区分大小写）、门户登录入口与退出后跳转地址。



图 12–3　外部认证：选择认证方式并填写对应的接口或请求头配置

认证方式	配置要点	登录流程
账号密码（OA 接口校验）	登录接口地址、表单或 JSON 请求格式、请求与响应字段名、成功标识、管理员标识、超时（500–30000 ms）	用户在 Jync 登录页输入账号密码，Jync 在服务端调用 OA 接口校验，不落库、不写日志
网关头（统一认证）	身份头名称、管理员标识头与标识值、可选共享密钥头；依赖可信代理列表	网关完成登录后注入身份头，Jync 只采信直接对端在回环或可信代理内的请求
票据校验（统一认证）	票据校验地址、GET 或 POST、票据参数名（默认 `ticket`）、响应字段映射	门户签发一次性票据后跳回 `/sso/entry?ticket=...`，Jync 在服务端换取账号后建立会话

外部用户只保留会话：不写入 `app_user`，没有本地密码，也没有修改密码入口。管理员判定有三种来源：配置中的管理员用户名列表命中；账号密码或票据响应中的管理员标识字段命中；网关头模式下的管理员标识头命中。其余外部用户为只读角色。

保留或关闭本地登录

启用外部认证后，默认仍保留本地账号登录，访问 `/login` 会按当前认证方式分流，本地表单固定在 `/login/local`。只有先完成外部认证配置，才允许关闭本地登录；关闭时页面会二次确认，关闭后 `POST /login` 也会被拒绝。

> **关闭本地登录后的应急恢复**
> 如果外部认证随后不可用，将无法从页面登录。此时在服务器上设置 `JYN_RESET_LOGIN=true` 并重启一次，本地登录会自动恢复；进入系统修好外部认证配置并保存后，去掉该变量再重启。`JYN_RESET_LOGIN` 平时不生效，不是长期开关。

连通性与安全

- **先测试再启用**网关头与票据模式提供测试按钮，返回当前请求会被识别成谁、来源是否可信，以及一次真实票据校验的结果；测试不会创建会话。
- **只填可信内网地址**OA 登录地址与票据校验地址由管理员填写，属于服务端外呼（SSRF 面），不要填公网或不可信地址。只接受 http(s)，明文 HTTP 必须显式开启「允许明文 HTTP」。
- **网关头必须部署在网关之后**Jync 只信任直接 TCP 对端位于回环或 `app.security.trusted-proxies` 内的身份头；网关必须剥离客户端伪造的同名头，Jync 不能直接暴露到客户端。
- **失败锁定**本地账号与 OA 账号共用「用户名 + 客户端 IP」失败计数，5 次失败锁定 15 分钟；票据入口按客户端 IP 单独限流，20 次校验失败后暂停 15 分钟。
- **退出跳转**可以把退出登录后的落点指向门户退出页，支持同站路径或 http(s) 地址；留空则回到登录页。

12.4 告警通知

「配置管理 → 告警通知」用 Webhook 把运行异常推送到钉钉、企业微信、飞书或自建接口。告警只读取 Jync 已有的运行状态，不改变同步、CDC 与 ETL 任务的行为；关闭开关后不评估任何规则，也不发送消息。



图 12–4　告警通知：Webhook 地址、消息格式、分类与冷却时间

填写 Webhook 地址、选择消息格式与要推送的分类，设置重复提醒间隔后，可先点「发送测试消息」确认接收端配置。地址加密保存，保存后页面只回显尾号；未配置地址时不产生任何外呼，适合默认不出网的内网环境。

告警分类	触发条件
同步失败	同步任务一轮执行失败并达到失败阈值
轮询卡死	一轮长时间没有进展（阈值见 `sync.stuck-threshold-ms`）
项目未运行	应有任务在跑，但调度处于停止状态
CDC 停摆	日志 CDC 位点长时间不推进
数据漂移	行数审计发现源 / 目标行数差异
连接健康	连接测试失败或健康检查异常
PG slot 积压	PostgreSQL 复制槽 WAL 滞留超过阈值

- **四种消息格式**：通用 JSON / 钉钉 / 企业微信 / 飞书，按接收端选择。
- **冷却与恢复**：同一事件在重复提醒间隔内不再发送，间隔可取 5–1440 分钟；故障恢复时总发送一次恢复通知。
- **只读评估**：告警不写任务、不改位点、不触发重试；关闭后系统行为与未配置时完全一致。

12.5 版本检查与一键升级

系统信息页在后台检查新版本，顺序为 GitHub 优先、Gitee 兜底；单次请求超时 5 秒。成功结果缓存 10 分钟，失败缓存 1 分钟并自动重试。

版本号一致时，Jync 还会比对发布制品：GitHub 使用附件的 SHA-256，Gitee 没有摘要也没有大小时退化为读取下载地址的 `Content-Length`。发布方在同一 tag 下重新上传安装包时，版本卡显示「同一版本号的安装包已重新发布」，一键升级同样可用；证明不了差异（例如下载地址 HEAD 失败）时按已是最新处理，不会误报。管理员可以点「忽略此提示」，发布方再上传新文件后会自动重新提醒。设置 `app.update-check.match-artifact=false` 可关闭比对，只比较版本号。

两个平台的匿名 API 都按客户端 IP 限流。超出配额时，升级弹窗会说明「Gitee 拒绝了匿名请求（HTTP 403，通常是匿名调用限流）。请切换下载源，或稍后重试」——换一个下载源通常即可继续。下载源始终是你选定的那个，Jync 不会悄悄改源；设置页版本卡的后台检查不受影响，仍是 GitHub 优先、失败自动回退 Gitee。

一键升级步骤

1. **检查版本**
   发现新版本后，版本卡出现升级入口。
2. **选择下载源**
   默认 Gitee，也可以选择 GitHub。
3. **下载与校验**
   下载新版 `jync.jar`，校验文件大小（接口提供时）与 zip 完整性。
4. **替换文件**
   校验通过后替换当前运行文件，旧文件保留为 `jync.jar.bak`。
5. **重启**
   选择「立即重启」由辅助脚本等旧进程退出后拉起新进程；也可以稍后手动重启。
6. **自动回滚**
   新进程异常退出或端口未就绪时，脚本自动用 `.bak` 回滚。

> **升级前备份**
> 一键升级只替换程序文件，不备份元数据库。升级前请复制 `data` 目录；数据库结构由应用自动演进，但备份是最后的退路。

纯内网无法访问 GitHub / Gitee 时，设置 `app.update-check.enabled=false`，升级入口与黄色提示会一并消失。

12.6 手工升级（离线环境）

无法访问 GitHub / Gitee，或一键升级被安全策略禁止时，按以下步骤手工替换。整个过程不改变元数据库结构，升级后首次启动会自动演进表结构。

1. **停止服务**
   确认没有同步任务在运行，再停止应用。

2. **备份**
   复制 `data/` 与当前 `jync.jar`，保留可回退版本。

3. **替换文件**
   把新包命名为 `jync.jar`，覆盖旧文件；注意文件属主与执行权限。

4. **启动并验证**
   启动后访问登录页，检查系统信息中的版本号；查看日志确认无 ERROR。

5. **抽查同步**
   运行一次「立即同步」，确认变更日志正常写入。

```
# systemd 托管时的典型操作
sudo systemctl stop jync
sudo cp -a /opt/jync/data /opt/jync/data.bak-$(date +%F)
sudo cp jync.jar /opt/jync/jync.jar
sudo systemctl start jync
sudo systemctl status jync --no-pager
```

> **回退**
> 新版本异常时，停止服务，把备份的 jar 与 data 目录换回，再启动。元数据库结构由 Hibernate 自动演进，旧版本通常可以打开新结构；如遇字段不兼容，使用升级前的 data 备份。

12.7 一键诊断包

设置页的「诊断包」卡片仅管理员可见，点「下载诊断包」会在浏览器中下载一个 zip，用于提交工单或发给技术支持。导出动作本身会写入审计日志（`DIAGNOSTICS_EXPORT`）。

包内条目	内容
README.txt	包说明、脱敏边界与大小上限
manifest.json	每个条目的文件名、大小与 SHA-256，便于校验传输中未被改动
summary.json	版本、生成时间、项目数与连接数、日志行数

包内条目	内容
config/	同步与应用的运行参数（只含参数值，不含原始配置文件）
connections.json	连接的产品类型、角色、主机、端口、库 / schema 与用户名
projects/	每个项目的进度、CDC 位点摘要
changelog/ 与 audit/	各最近 200 条变更记录与管理员操作审计
logs/jync.log.tail.txt	应用日志尾部，最多 2000 行 / 2 MB

不含什么

密码与 `enc:` 密文、API Key、Webhook 地址、带账号信息的连接串、自定义驱动路径、游标原值、CDC 布局签名、`jync.key`、`data/` 下的文件与原始 `application.yml` 都不入包。日志、变更详情与审计详情还会经过一层正则脱敏，屏蔽赋值形式的密码 / token、Bearer、URL 账号信息、敏感查询参数与私钥块。脱敏是尽力而为，包内仍可能含业务标识与日志正文，请按敏感文件处理，对外提供前先过一遍。

大小上限

单个 JSON 条目 1 MB、日志尾部 2 MB、整个包压缩前 10 MB。超过上限时页面会提示「诊断包超过 10 MB 上限」，此时可先缩小日志或减少最近记录，再重新导出；Jync 不会静默截断出一个不完整的包。

CHAPTER 13

典型场景操作

13.1 Oracle 到达梦：不停机迁移

1. 建立 Oracle 源连接与达梦目标连接，分别测试通过。
2. 新建同步任务，勾选全部业务表、索引、视图与过程；先不勾选「允许执行删除」。
3. 执行结构同步，在转换审阅页逐条处理失败的存储过程。
4. 低峰期执行「立即同步」完成全量基线；观察吞吐日志与达梦的 CPU、IO。
5. 开启日志 CDC（需要 Oracle 归档与补充日志），观察落后秒数稳定在秒级。
6. 业务切换前做一次行数与关键表内容核对；切换后保留 Oracle 写入一段时间作为回退。

13.2 MySQL 到 PostgreSQL：报表库同步

1. 源库使用只读账号；目标库使用具备建表与写入权限的账号。
2. 只勾选报表需要的表，视图按需同步。
3. 为字符串列配置 TRIM 与空串转 NULL，统一两边的空值语义。
4. 轮询间隔可以放宽到 10—30 秒，降低源库压力。
5. 在目标库建立物化视图或汇总表，由报表工具直接读取。

13.3 生产到测试：按需抽取

1. 克隆生产同步任务，目标连接改为测试库。
2. 取消勾选不需要的大表，只保留测试所需的配置表与样本表。
3. 确认「首次加载前清空目标表」处于关闭状态。
4. 执行一次「立即同步」，随后停止同步任务，避免持续写入测试库。

敏感数据

Jync 的列清洗只做格式规整，不做脱敏。把生产数据复制到测试环境前，请在源端或目标端另行处理手机号、身份证号等敏感字段。

13.4 多源汇聚到分析库

为每个源库建立一个同步任务，目标都指向同一个分析库，用不同的 schema 或表名前缀隔离。每个同步任务独立调度、独立游标、独立锁，互不影响。分析库的写入能力是共同瓶颈，建议错开各同步任务的全量时间，并统一在低峰期执行。

13.5 场景与策略速查

需求	推荐捕获策略	关键设置
一次性迁移，之后不再同步	增量轮询（全量后停止）	全量后停止同步任务，保留配置备查
持续同步，源库不能改	增量轮询	确认存在更新时间列并建索引
需要同步删除	日志 CDC	开启对应日志并保留 ⩾ 24h
秒级延迟、变更量大	日志 CDC	观察落后秒数与吞吐，必要时扩容目标库
小表、有主键且无游标列	全量比对	行数低于 `full-compare-max-rows`；这样才会自动选用
生产库负载敏感	增量轮询 + 低并行	并行度 1，批次 1000–2000，低峰全量

不要频繁切换策略

每次从轮询切到 CDC 都会重做基线；切回轮询会清零游标并重新全量。策略切换应视为一次小型迁移，安排在维护窗口内。

CHAPTER 14

常见问题与已知限制

14.1 常见问题

问题	原因与处理
登录后立刻掉登录态	HTTPS 部署未设置 `JYN_COOKIE_SECURE=true`，或代理未传 `X-Forwarded-Proto`；iframe 嵌入时 SameSite 需设为 `None` 并同时开启 Secure
登录页只有门户入口，没有账号密码框	已启用外部认证且关闭了本地登录；从企业门户进入，或在服务器上以 `JYN_RESET_LOGIN=true` 启动一次恢复本地登录
OA 登录一直提示账号或密码错误	核对 OA 接口地址、请求 / 响应字段映射与成功标识；连续 5 次失败会按「用户名 + IP」锁定 15 分钟
外部用户找不到修改密码	外部用户没有本地密码，改密应回到 OA 或统一门户；服务端会拒绝 `/account/password`
系统日志页没有可选文件或提示不可用	应用尚未写出日志，或 `logging.file.name` 指向了白名单之外的位置；确认日志目录与文件权限
页面时间差 8 小时	容器缺少 `TZ=Asia/Shanghai`，或数据库连接时区不一致
连接测试全部超时	容器内 `localhost` 指容器自身；改用宿主机地址或 `host.docker.internal`
表列表为空	元数据快照未刷新；执行一次结构同步或刷新同步任务
更新同步不到	游标列是 id / create_time；改用 update_time 或 CDC
删除不同步	轮询模式不感知删除；切换到 CDC 或对小表启用全量比对
「下次自动运行」时间已过去、目标行数不再增长	上一轮没有返回，Quartz 已跳过后续触发；详情页会显示「上一轮仍在执行」或「疑似卡死」，等待不会恢复，需重启本实例
同步任务一直显示运行中	可能卡在长事务或锁等待；查看日志与数据库会话，必要时停止后强制解锁
CDC 落后持续增长	目标库写入慢或源库变更量超过消费能力；查看吞吐与目标库负载
CDC 同步任务变 ERROR	位点失效；使用「重建基线」
崖山源选 LogMiner 探测不通过	崖山没有 `DBMS_LOGMNR`；把捕获策略改为 YStream
手动同步出现黄色提示「连接池被其他任务占满」	同一连接配置上的任务争抢连接池；本轮已让出、下个调度周期自动重试。调大 `sync.max-pool-size` 或错开各任务的调度时间
存储过程转换失败	控制流无法自动翻译；在转换审阅页手工覆盖
页面出现 ??key??	语言包缺失；升级到完整发行包并清理浏览器缓存
磁盘被日志占满	v1.7.7 已限制 10MB × 14 天 × 500MB；旧版本请升级或手工清理

14.2 已知限制

- **存储过程语义**：机械差异可自动转换，控制流无法可靠自动翻译；语法可执行不等于业务语义等价，关键过程必须实测。
- **无主键表**：无法保证幂等，重放可能重复；CDC 流式阶段跳过无主键表，仅做基线播种。
- **目标库写入方**：假定目标库只由 Jync 写入；外部同时写入可能被覆盖或产生冲突。
- **删除检测**：轮询模式不感知删除；CDC 能处理删除；FULL_COMPARE 只有在开启「同步行删除」且表有主键、行数不超过上限时才处理。
- **DDL 捕获**：CDC 不捕获 DDL，只记录告警并使结构缓存失效；结构变化大时重建基线。
- **CDC 列布局漂移**：消费者停机期间源表执行过 DDL，重启会停在「列布局漂移」并要求重建基线；这是有意的停车，不会猜测列映射把值写进相邻列。
- **分析型引擎作同步源**：Doris、ClickHouse、Hive、Impala、Elasticsearch 与 MongoDB（Atlas SQL）只作源、不作目标，按无主键读取，增量只识别新增；Hive、Impala、Elasticsearch 与 MongoDB（Atlas SQL）不能按 OFFSET 分页，大表只能单次流式读取；Elasticsearch 还受服务端 `index.max_result_window` 限制。
- **轮询延迟**：下限即轮询间隔；需要秒级延迟请使用 CDC。
- **临时对象残留**：语法检查过程中进程被强杀可能留下 `JYNC_AI_CHECK_*` 对象，需要手工清理。
- **外部认证不管理账号生命周期**：Jync 只校验登录并保留会话，不创建本地账号；账号的创建、停用与改密都在企业 OA 或统一门户完成。
- **外部认证地址由管理员填写**：属于服务端外呼，只能填可信内网地址；网关头模式要求 Jync 部署在网关之后，不能直接暴露给客户端。
- **系统日志是有界检索**：单次最多扫描 16 MB、返回最近 1000 条匹配记录；更早的内容需要下载日志文件或到服务器上检索。
- **专属迁移没有断点续传**：只复制运行那一刻的数据，取消只在批间生效；失败重跑按写入方式决定是否先清理目标对象，OVERWRITE 可收敛，APPEND 可能重复。
- **专属迁移的对象边界**：view 与 timeseries 集合不复制，分片 key、用户 / 角色 / 认证信息不迁移；MongoDB 的 APPEND 保留目标已有文档，与 ES 的同名写入方式语义不同。

> **遇到手册未覆盖的问题**
>
> 先查看 `logs/jync.log` 中同一时间段的 ERROR 与 WARN，再到同步任务页确认锁、位点与游标状态。多数问题能在这两处定位。仍无法解决时，附上版本号、数据库类型与相关日志片段联系支持。

Jync
捷同 · 异构数据库同步

把复杂留给平台
把简单留给使用者

Jync 捷同 · 异构数据库同步平台

用户操作手册 · v1.12.0

jync.qqmu.com · v1.12.0 · 2026 年 11 月

技术交流 QQ 群 426669837 · 邮箱 vfaner@88.com

© Jync 捷同 · 沐沐开源工坊