# 3D 生成网站 · 数仓分层与 DWD / DWS（v1.0）

面向业务分析与运营看板，给出**可落地**的数据仓库分层（ODS → DIM/DWD → DWS → ADS），覆盖生成链路（Prompt→Job→Request→Asset/Model→Conversion→Share/Usage）与平台域（计费、限流、审核、通知）。以 **Hive/Presto/SparkSQL** 口径为主，字段命名统一小写下划线，日期分区 dt（YYYY-MM-DD）。

* 时区：**UTC 存储**，展示层按业务需要在语义层转换；若强制东八，则统一落表时换算。
* 粒度：DWD 保留**明细级**粒度；DWS 为**按日/周/月**的宽表或宽聚合；ADS 为主题域报表。
* 主键：明细层优先采用业务 ID（如 job\_id），维表采用**代理键**（sk\_\*）。

## 0. 分层与命名规范

* **ODS**（原始采集层）：与 OLTP 同构，最少清洗，仅做增量拉链/去重。
* **DIM**（维度层）：SCD2（拉链）维度，统一被业务事实引用。
* **DWD**（明细事实层）：围绕业务过程构建的事实明细，**一表一过程一粒度**。
* **DWS**（服务汇总层）：面向复用的公共汇总宽表（按日/周/月/模型/组织等维度）。
* **ADS**（应用主题层）：面向看板/报表的结果表，口径固定，轻计算。
* 表前缀建议：ods\_ / dim\_ / dwd\_ / dws\_ / ads\_；事实表可用 fct\_ 的别名视团队习惯。

## 1. 关键维度（DIM）

维表均采用 SCD2：effective\_from, effective\_to, is\_current；代理键以 sk\_ 开头。

### 1.1 dim\_date（日历维）

sk\_date（yyyyMMdd, PK）, date, year, quarter, month, week\_of\_year, day\_of\_week, is\_weekend, is\_holiday, tz

### 1.2 dim\_org（组织维）

sk\_org(PK), org\_id, org\_name, plan, status, SCD2 字段…

### 1.3 dim\_user（用户维）

sk\_user, user\_id, org\_id, email, role, status, created\_at, …

### 1.4 dim\_project（项目维）

sk\_project, project\_id, org\_id, project\_name, visibility, …

### 1.5 dim\_provider / dim\_provider\_model（供应商/模型维）

* dim\_provider：sk\_provider, provider\_id, name, status
* dim\_provider\_model：sk\_provider\_model, provider\_model\_id, provider\_id, model\_name, task\_type, status, limits\_json, pricing\_json

### 1.6 dim\_asset（资产维：3D/贴图/图像）

sk\_asset, asset\_id, org\_id, project\_id, kind, format, mime\_type, source\_type, license, visibility, size\_bytes, created\_at

### 1.7 其他常用维

* dim\_job\_status（标准化状态枚举）
* dim\_conversion\_operation（转换类型，如 glb\_to\_usdz, mesh\_simplify）
* dim\_region（worker/存储区域）

## 2. DWD 明细事实（业务常用）

分区：除特别说明外，均按 dt = from\_unixtime(event\_ts, 'yyyy-MM-dd') 落分区。

### 2.1 dwd\_prompt\_submit\_di（Prompt 提交明细，**一行 = 一次提交**）

主键：prompt\_id 字段：prompt\_id, org\_id, project\_id, user\_id, input\_type, text, input\_asset\_id, language, safety\_tags, parameters\_json, prompt\_hash, created\_at\_ts as event\_ts, dt 用途：提示词分析、主题/语言分布、去重命中率。

### 2.2 dwd\_job\_lifecycle\_di（生成任务生命周期明细，**一行 = 一个 job**）

主键：job\_id 字段（核心）： - 标识：job\_id, org\_id, project\_id, prompt\_id, provider\_id, provider\_model\_id - 状态：status\_final（最终态），attempt, retry\_of\_job\_id, started\_at\_ts, completed\_at\_ts - 时延：queue\_latency\_ms, run\_latency\_ms（=completed-started）, latency\_ms\_total - 成本：cost\_usd, billed\_units\_json - 归因：error\_code, error\_message - 环境：worker\_id, worker\_region - dt 用途：SLA/吞吐/成本/失败归因；与请求、资产、评估关联的**主事实表**。

### 2.3 dwd\_provider\_request\_di（底层 API 往返明细，**一行 = 一次调用**）

主键：request\_id 字段：request\_id, job\_id, request\_ts, response\_ts, http\_status, tokens\_input, tokens\_output, billed\_cost\_usd, payload\_req\_json, payload\_resp\_json, retry\_index, provider\_request\_id, dt 用途：与计费/超时/限流相关的根因排查与对账。

### 2.4 dwd\_model\_asset\_di（生成主模型明细，**一行 = 一个 3D 主文件**）

主键：model\_asset\_id 字段：model\_asset\_id, job\_id, asset\_id, format, vertex\_count, face\_count, material\_count, uv\_map\_count, triangulated, watertight, units, origin, bbox\_json, preview\_asset\_id, created\_at\_ts, dt 用途：质量/复杂度分布，后续转换输入。

### 2.5 dwd\_texture\_asset\_di（贴图明细）

id, model\_asset\_id, texture\_asset\_id, role, uv\_channel, size\_bytes, created\_at\_ts, dt

### 2.6 dwd\_conversion\_task\_di（转换/后处理任务明细）

conversion\_id (PK), org\_id, source\_asset\_id, operation, parameters\_json, status\_final, started\_at\_ts, completed\_at\_ts, run\_latency\_ms, output\_asset\_id, error\_code, dt

### 2.7 dwd\_share\_access\_di（外链/分享访问明细）

access\_id, org\_id, subject\_type, subject\_id, share\_token\_hash, user\_agent, ip\_region, action(view/download), http\_status, referred\_by, accessed\_at\_ts, dt

### 2.8 dwd\_api\_access\_di（API Key 调用审计明细）

access\_id, org\_id, key\_id, route, http\_method, http\_status, latency\_ms, bytes\_in, bytes\_out, throttled boolean, accessed\_at\_ts, dt

### 2.9 dwd\_evaluation\_di（评估与评分明细）

evaluation\_id, subject\_type(job/model\_asset), subject\_id, evaluator\_type, rubric\_id, metrics\_json, rating, created\_by, created\_at\_ts, dt

### 2.10 dwd\_moderation\_event\_di（审核事件明细）

event\_id, org\_id, subject\_type, subject\_id, rule, severity, decision, details\_json, created\_at\_ts, dt

### 2.11 dwd\_asset\_storage\_di（资产存储明细快照，**一行 = 一个资产/天**）

用于成本核算与清理策略。 asset\_id, org\_id, project\_id, kind, format, size\_bytes, storage\_region, lifecycle\_state, retention\_until, snapshot\_ts, dt

## 3. DWS 公共汇总宽表（业务常用）

典型按日聚合；如需周/月，沿用同口径二次汇总。

### 3.1 dws\_job\_day\_by\_org\_model（**组织×模型×日** 任务汇总）

粒度：org\_id, provider\_model\_id, dt 字段： - 量：job\_cnt, job\_cnt\_succ, job\_cnt\_fail, job\_cnt\_cancel - 成功率：succ\_rate = job\_cnt\_succ / nullif(job\_cnt\_succ + job\_cnt\_fail,0)（不含取消） - 时延：p50\_latency\_ms, p95\_latency\_ms, p99\_latency\_ms, avg\_latency\_ms - 成本：cost\_usd\_sum, cost\_usd\_avg\_per\_job - 重试：retry\_job\_cnt, avg\_attempt - 失败归因Top3：fail\_code\_top1, top1\_cnt, top2, top2\_cnt, top3, top3\_cnt

### 3.2 dws\_request\_day\_by\_provider（**供应商×日** API 调用汇总）

粒度：provider\_id, dt 字段：request\_cnt, http\_2xx\_cnt, http\_4xx\_cnt, http\_5xx\_cnt, timeout\_cnt, avg\_req\_latency\_ms, billed\_cost\_usd\_sum

### 3.3 dws\_model\_quality\_day（**模型质量×日**）

粒度：provider\_model\_id, dt 字段：model\_cnt, avg\_vertex, avg\_face, watertight\_rate, triangulated\_rate, rating\_avg, fidelity\_avg\_from\_metrics

### 3.4 dws\_conversion\_day（**转换成功率×日**）

粒度：operation, dt 字段：conv\_cnt, conv\_succ\_cnt, succ\_rate, avg\_conv\_latency\_ms

### 3.5 dws\_storage\_day\_by\_org（**存储与资产×组织×日**）

粒度：org\_id, dt 字段：asset\_cnt\_new, size\_bytes\_new, asset\_cnt\_total, size\_bytes\_total, purge\_cnt, archive\_cnt

### 3.6 dws\_user\_activity\_day（**活跃与留存×日**）

粒度：org\_id, dt 字段：active\_users\_dau, new\_users, prompt\_submit\_users, job\_submit\_users, download\_users, dau\_wau\_ratio, dau\_mau\_ratio

### 3.7 dws\_funnel\_day（**漏斗×日**，全站）

字段：prompts, jobs\_created, jobs\_running, jobs\_succeeded, models\_created, shares\_created, shares\_downloaded, prompt2job\_rate, job2model\_rate, model2share\_rate

### 3.8 dws\_cost\_day\_by\_org（**成本×组织×日**）

org\_id, dt, infra\_cost\_storage\_usd, provider\_cost\_usd, egress\_cost\_usd, total\_cost\_usd > 说明：provider\_cost\_usd 来自 dwd\_provider\_request\_di 聚合；storage/egress 由对象存储账单或估算表对齐。

## 4. ADS 主题域（常用业务看板）

ADS 不做复杂逻辑，全部来源于 DWS/DIM 轻度再加工。

* **ads\_ops\_org\_kpi\_1d**：运营 KPI（成功率、P95、失败归因Top、重试率）
* **ads\_fin\_month\_invoice**：月度对账（组织×供应商×模型），含单价、折扣、税率
* **ads\_quality\_model\_leaderboard\_1d**：模型榜单（质量指标 + 用户评分）
* **ads\_growth\_activation\_cohort\_1w**：按注册周分群的激活/留存
* **ads\_reliability\_slo\_28d**：SLO 计算（成功率/延迟合规窗口）
* **ads\_storage\_purge\_candidates**：按策略筛选清理候选资产

## 5. 口径与计算规则（关键一致性）

* **成功率（任务）**：succ\_rate = succeeded / (succeeded + failed)；取消不计入分母。
* **时延**：优先使用 run\_latency\_ms；总时延可展示 queue + run。
* **成本**：
  + 供应商成本：聚合 dwd\_provider\_request\_di.billed\_cost\_usd；如为空，以 dwd\_job\_lifecycle\_di.cost\_usd 回填。
  + 存储成本：size\_bytes\_total × 当日存储单价（分层/区域价表）
  + 出网成本：bytes\_out × 出网单价（来自 dwd\_api\_access\_di/下载日志）
* **质量**：watertight\_rate = sum(watertight=true)/count(\*)；评分取 rating 均值并暴露样本量。
* **去重**：按 prompt\_hash 识别复用；同 prompt\_id 多次调用在 Job 层统计。

## 6. 参考建表样例（Hive/Presto）

-- DWD: job lifecycle  
create table if not exists dwd\_job\_lifecycle\_di (  
 job\_id string,  
 org\_id string,  
 project\_id string,  
 prompt\_id string,  
 provider\_id string,  
 provider\_model\_id string,  
 status\_final string,  
 attempt int,  
 retry\_of\_job\_id string,  
 started\_at\_ts bigint,  
 completed\_at\_ts bigint,  
 queue\_latency\_ms int,  
 run\_latency\_ms int,  
 latency\_ms\_total int,  
 cost\_usd double,  
 billed\_units\_json string,  
 error\_code string,  
 worker\_region string,  
 dt string  
)  
partitioned by (dt string)  
stored as parquet;  
  
-- DWS: jobs by org & model per day  
create table if not exists dws\_job\_day\_by\_org\_model as  
select org\_id,  
 provider\_model\_id,  
 dt,  
 count(\*) as job\_cnt,  
 count\_if(status\_final='succeeded') as job\_cnt\_succ,  
 count\_if(status\_final='failed') as job\_cnt\_fail,  
 approx\_percentile(run\_latency\_ms, 0.50) as p50\_latency\_ms,  
 approx\_percentile(run\_latency\_ms, 0.95) as p95\_latency\_ms,  
 approx\_percentile(run\_latency\_ms, 0.99) as p99\_latency\_ms,  
 avg(run\_latency\_ms) as avg\_latency\_ms,  
 sum(cost\_usd) as cost\_usd\_sum,  
 avg(cost\_usd) as cost\_usd\_avg\_per\_job  
from dwd\_job\_lifecycle\_di  
where dt = '${bizdate}'  
group by org\_id, provider\_model\_id, dt;

## 7. 调度与增量策略

* **抽取节奏**：
  + ODS：T+0 分批次导入；请求日志可 5~15 分钟一批。
  + DWD：T+0 小时级；跨天 Job 在完成时落入对应完成日 dt。
  + DWS/ADS：**日更**（01:00-03:00 窗口），周/月在周期末触发。
* **拉链/SCD**：维度按更新事件生成新版本；事实不回写、只追加。
* **迟到与纠偏**：对跨日完成、迟到日志设置 D+3 回补任务；DWS 采用**可重算**逻辑（先删后写 or merge）。

## 8. 数据质量与校验

* 完整性：DWD 与 ODS 行数/散列抽样比对；Job→Request→Asset 链路**外键可达性**检查。
* 一致性：成功率/成本在 DWS 与 ADS 互相校验；TopN 失败码分布日比对波动阈值告警。
* 唯一性：job\_id、request\_id、asset\_id 全局唯一；分区内**重复键**拒写或去重。
* 新鲜度：关键表设 **SLA**（如 DWD ≤ 30min；DWS ≤ 90min），超时报警。

## 9. 字段清单（精选 DWD/DWS）

便于落地开发，这里列出两张最常用表的字段口径细化。

### 9.1 dwd\_job\_lifecycle\_di 字段口径

* status\_final：取最后状态；若存在重试，保留本 job\_id 的实际终态。
* run\_latency\_ms：coalesce(completed\_at\_ts, now) - started\_at\_ts（毫秒）。
* cost\_usd：若空，按 provider\_request 聚合值回填；若都空则 0。
* error\_code：归一化为枚举：TIMEOUT/RATE\_LIMIT/PROVIDER\_4XX/PROVIDER\_5XX/PAYLOAD\_TOO\_LARGE/UNKNOWN。

### 9.2 dws\_job\_day\_by\_org\_model 字段口径

* succ\_rate：不含取消；当分母为 0 时置空。
* p50/95/99\_latency\_ms：run\_latency\_ms 的近似分位（Presto approx\_percentile）。
* cost\_usd\_sum：日内所有 job 成本之和；注意与请求级成本对齐差异 ≤1%。

## 10. 交付物与下一步

* 本最终版可直接进入建表与调度配置；如需，我可**生成所有表的 DDL** 与 **Airflow/Scheduler 任务清单**（含依赖图）。
* 若接入对象存储账单/下载日志，可补充 dwd\_storage\_bill\_di 与 dws\_cost\_day\_by\_org 自动对账链路。
* 若要搜索/推荐，建议叠加 embeddings 明细与召回特征层（可放 DWD/DWS-Feature 子域）。