

3.1 data_vendor

Spalte	Typ	Hinweise
id	SMALLINT UNSIGNED oder INT UNSIGNED	Primärschlüssel; selten mehr als 64k Anbieter nötig.
code	VARCHAR(64)	UNIQUE , unveränderlicher Maschinencode (cockpit, partner_xyz, national_park_api).
name	VARCHAR(255)	Menschenlesbar.
legal_name	VARCHAR(255)	Nullable; für Verträge.
base_url	VARCHAR(512)	Nullable; API- oder CMS-Wurzel.
contact_email	VARCHAR(255)	Nullable.
is_active	TINYINT(1)	Standard 1.
created_at	DATETIME(3)	Server-Standard CURRENT_TIMESTAMP(3).

Indizes: UNIQUE (code).

3.2 data_import_batch (optional, dringend empfohlen)

Spalte	Typ
id	BIGINT UNSIGNED
vendor_id	SMALLINT UNSIGNED
batch_key	VARCHAR(128)
source_uri	VARCHAR(1024)
record_count	INT UNSIGNED
started_at	DATETIME(3)
finished_at	DATETIME(3)
status	ENUM (4 Werte)
error_summary	VARCHAR(2048)

Bedeutung

- **id** — Primärschlüssel, Auto-Increment.
- **vendor_id** — **FK** → **data_vendor(id)**.

- **batch_key** — Stabiler Batch-Schlüssel; **eindeutig** pro Anbieter oder global. Beispiele: Inhalts-Hash (sha256:...), oder zeitgestempelt Dump-Label (...Z_partner_dump).
- **source_uri** — Optional: Pfad, URL oder Dateiname der Importquelle.
- **record_count** — Optional: Zeilenanzahl aus der Datei.
- **started_at / finished_at** — Zeitfenster des Jobs.
- **status** — einer von: running, success, failed, partial.
- **error_summary** — Optional: kurzer Fehlertext.

Indizes

- (vendor_id, started_at)
- Entweder **UNIQUE (batch_key)** (global eindeutige Keys) oder **UNIQUE (vendor_id, batch_key)** (eindeutig pro Anbieter).

3.3 tour (kanonisch)

Spalte	Typ	Hinweise
tour_uid	BIGINT UNSIGNED	Primärschlüssel , AUTO_INCREMENT — kanonische ID (Alternative: BINARY(16) UUID v7, wenn undurchsichtige, nicht fortlaufende öffentliche IDs nötig sind).
created_at / updated_at	DATETIME(3)	updated_at in der Anwendung pflegen oder ON UPDATE.
primary_vendor_id	SMALLINT UNSIGNED	FK → data_vendor — „Hüter“ der Tour-Inhalte; nur bei Legacy nullable.
last_import_batch_id	BIGINT UNSIGNED	Nullable FK → data_import_batch — letzter erfolgreicher Import, der diese Tour geändert hat.
...	bestehende Spalten	name, length_m, country_code, center_point, JSON types, usw.

Regeln:

- **tour_uid wird von Anbietern beim Insert nie mitgeliefert** — nur von der DB (oder von der App mit UUID_TO_BIN, falls UUID als PK gewählt wird).
- **cockpit_mongo_id** (sowie weitere Legacy-Spalten nach Bedarf) — als nullable Spalten mit **UNIQUE**, sofern die Fachlogik Eindeutigkeit garantiert; es sind **Aliase**, nicht der kanonische

Schlüssel.

3.4 tour_external_alias

Sinnvoll, wenn mehrere Anbieter oder ID-Schemata dieselbe Tour meinen oder ein Anbieter mehrere Kennungen liefert.

Spalte	Typ	Hinweise
id	BIGINT UNSIGNED	PK, Auto-Increment.
tour_uid	BIGINT UNSIGNED	FK → tour(tour_uid) ON DELETE CASCADE.
vendor_id	SMALLINT UNSIGNED	FK → data_vendor — wer diese ID ausgegeben hat.
alias_type	VARCHAR(32)	z. B. cockpit_mongo, partner_slug, external_uuid.
alias_value	VARCHAR(191)	Normalisierte Stringform; Länge 191 für Kompatibilität mit älteren Indexlängengrenzen.
import_batch_id	BIGINT UNSIGNED	Nullable FK — welcher Batch diesen Alias zuerst oder zuletzt gesetzt hat.
valid_from / valid_to	DATETIME(3)	Optional; für abgelöste IDs ohne Historie zu löschen.
is_active	TINYINT(1)	Standard 1.

Indizes (wichtig für API-Performance):

- **UNIQUE (vendor_id, alias_type, alias_value)** — verhindert doppelte Zuordnung derselben externen Kennung.
- **INDEX (alias_type, alias_value)** — schnelle Auflösung WHERE alias_type = 'cockpit_mongo' AND alias_value = ?.
- **INDEX (tour_uid)** — alle Aliase einer Tour auflisten.

Wenn nur ein einzeliges Modell auf tour vorgesehen ist, sind dennoch **primary_vendor_id** und optional **last_import_batch_id** erforderlich; die Alias-Tabelle wird nötig, sobald mehrere Anbieter existieren oder sich IDs ändern.

3.5 Provenienz auf Feldebene (optional)

Für strikte Revision („Wer hat ascent_m geliefert?“):

- **tour_field_provenance:** (tour_uid, field_name, vendor_id, import_batch_id, updated_at) mit PK (tour_uid, field_name) oder eine reine Anfüge-Historientabelle.

Für viele leseintensive APIs reicht **Provenienz auf Zeilenebene** (primary_vendor_id, last_import_batch_id).

4. MySQL-Designpraxis für diesen Anwendungsfall

4.1 Schlüssel und Joins

- **BIGINT UNSIGNED** als Surrogat-PK auf tour und FKs in Kindtabellen (tour_geo_city_drive, usw.) statt Joins über CHAR(24)-Mongo-Strings — **kleinere Indizes, bessere Nutzung des Buffer Pools**, klarere Zuständigkeit.
- Kindtabellen auf tour_uid-FK migrieren; tour_cockpit_mongo_id während der Übergangsphase indexiert lassen.

4.2 Räumliche Daten

- POINT-SRID **einheitlich** (z. B. SRID 4326); **SPATIAL INDEX** nur bei Abfragen nach Region/Bbox; Punktabfragen über Tour-ID brauchen keinen räumlichen Index auf center_point.

4.3 JSON-Spalten (types, beliebige Tags)

- MySQL 8 **funktionale Indizes** auf CAST(JSON_UNQUOTE(JSON_EXTRACT(...))), wo oft gefiltert wird; bestehende JSON_CONTAINS-Muster in Migrationsskripten dokumentieren und bei ausreichend niedriger Kardinalität generierte Spalten + B-Tree-Indizes ergänzen.

4.4 Zeichensatz und Kollation

- **utf8mb4** für alle nutzer- oder anbieterseitig sichtbaren Texte; utf8mb3 vermeiden.

4.5 Referentielle Integrität

- **ON DELETE RESTRICT** auf tour bezüglich Anbietern, wenn Touren keine „verwaisten“ Anbieterzeilen erzeugen dürfen; **ON DELETE SET NULL** auf last_import_batch_id, wenn Batches bereinigt werden.
- Für **tour_external_alias** ist **ON DELETE CASCADE** ausgehend von tour üblich.

4.6 Zeit

- **DATETIME(3)** in der Anwendungszeitzone **oder** UTC speichern und an der API-Kante umrechnen; konsequent bleiben. **TIMESTAMP** und **DATETIME** nicht ohne Dokumentation mischen.

5. API- und Anwendungsfolgerungen (kurz)

- (**vendor_code**, **alias_type**, **alias_value**) oder Legacy-?type=mongo|rid an der Controller-Grenze auf **tour_uid** auflösen; interne Dienste nutzen nur **tour_uid**.
- In JSON **tour_uid**, **primary_vendor_id** **oder** **vendor_code** sowie **last_import_batch_id** (oder ein öffentlicher **batch_key**) ausgeben — für nachgelagerte Systeme mit Audit-Bedarf.

6. Migrationspfad vom aktuellen Schema

1. **data_vendor**-Zeilen einfügen (mindestens einen für Cockpit / aktuelles CMS).
2. **tour_uid** (Auto-Increment) in **tour_v2** ergänzen, falls nicht vorhanden; backfill; PK wechseln oder Unique-Constraint setzen und schrittweise als FK-Ziel nutzen.
3. Nullable FKs **primary_vendor_id**, **last_import_batch_id** ergänzen.
4. **tour_external_alias** aus bestehenden Fremdkennungen befüllen (z. B. **cockpit_mongo_id** und weitere importierte Codes).
5. **tour_uid** in abhängigen Tabellen; per Join backfill; Anwendungsabfragen umstellen; redundante Join-Schlüssel entfernen, wenn unkritisch.

7. Kurzfassung

Bedarf	Antwort mit MySQL-Fokus
Stabile Identität	Surrogat- tour_uid (BIGINT AI oder BINARY(16) UUID v7), bei Neuanlage nie vom Anbieter geliefert.
Anbieter erkennen	data_vendor + primary_vendor_id auf tour; optional data_import_batch .
Externe Schlüssel	tour_external_alias mit UNIQUE (vendor_id, alias_type, alias_value) + Lookup-Indizes.
Performance	FK und Lookups auf BIGINT ; breite VARCHAR/CHAR-Indizes nur für Alias-Auflösung.
Audit	Batch-ID auf Zeilenebene; optional Provenienz-Tabelle auf Feldebene.

Damit bleibt MySQL die „Single Source of Truth“, interne Joins hängen weniger an anbieterspezifischen IDs, und **Datenanbieter** sowie **Import-Batch** werden für Compliance, Debugging und Partnerabrechnung zu expliziten Konzepten.