BambiGO 資料庫設計規格 (Database Schema Specification)

版本：v4.1
設計原則：正規化核心實體、JSONB 保留擴充彈性、索引優化查詢效能

---

1. 設計哲學

```
核心實體 (高頻查詢) → 正規化表格 + 索引
擴充屬性 (低頻/動態) → JSONB 欄位
即時數據 (高頻更新) → Redis 快取
```

為什麼不全用 JSONB？

- `WHERE facilities_metadata->>'has_locker' = 'true'` 無法使用 B-tree 索引
- 正規化後：`WHERE has_locker = true` 可使用索引，查詢快 10-100 倍

為什麼不全部正規化？

- 未來新增屬性（如 `has_aed`）需要 ALTER TABLE
- JSONB 允許「無痛擴充」，適合快速迭代的 MVP

---

2. 核心表格

2.1 cities (城市/區域)

```
sql
```

```
create table cities (
  id text primary key,                  -- 'tokyo_taito', 'tokyo_chiyoda'
  name jsonb not null,                  -- {"ja": "台東区", "en": "Taito City"}
  timezone text not null default 'Asia/Tokyo',
  bounds geography(polygon, 4326),      -- 地理圍欄 (Geo-fence)

  -- City Adapter 設定 (Feature Flags)
  config jsonb not null default '{}',
  /*
    config 結構：
    {
      "has_subway": true,
      "has_shared_mobility": true,
      "has_bus": true,
      "odpt_operators": ["TokyoMetro", "Toei", "JR-East"],
      "gbfs_systems": ["docomo-cycle-tokyo", "hellocycling"],
      "default_language": "ja"
    }
  */

  enabled boolean default true,
  created_at timestamptz default now()
);

-- 初始數據
insert into cities (id, name, config) values
  ('tokyo_taito', '{"ja": "台東区", "en": "Taito City"}',
   '{"has_subway": true, "has_shared_mobility": true, "odpt_operators": ["TokyoMetro", "Toei", "JR-East"]}'),
  ('tokyo_chiyoda', '{"ja": "千代田区", "en": "Chiyoda City"}',
   '{"has_subway": true, "has_shared_mobility": true, "odpt_operators": ["TokyoMetro", "Toei", "JR-East"]}'),
  ('tokyo_chuo', '{"ja": "中央区", "en": "Chuo City"}',
   '{"has_subway": true, "has_shared_mobility": true, "odpt_operators": ["TokyoMetro", "Toei", "JR-East"]}');
```

2.2 nodes (節點主表)

sql

```
create table nodes (
  id text primary key,                    -- 'odpt:TokyoMetro.Ueno' 或 'osm:12345678'
  city_id text references cities(id),

  -- 基本資訊
  name jsonb not null,                    -- {"ja": "上野駅", "en": "Ueno Station"}
  type text not null,                     -- 'station', 'bus_stop', 'poi', 'bike_station'
  location geography(point, 4326) not null,
  geohash text not null,                  -- 用於快速鄰近查詢

  -- L1 核心屬性 (正規化，支援索引)
  vibe text,                              -- 'busy', 'quiet', 'historic', 'commercial'
  accessibility text default 'unknown',   -- 'full', 'partial', 'none', 'unknown'

  -- Hub/Spoke 繼承
  is_hub boolean default false,
  parent_hub_id text references nodes(id),
  persona_prompt text,                    -- Hub 專用，Spoke 為 null

  -- 路線關聯 (用於繼承演算法優化)
  line_ids text[],                        -- ['TokyoMetro.Ginza', 'JR-East.Yamanote']

  -- 數據來源
  source_dataset text not null,           -- 'odpt', 'osm', 'gbfs', 'manual'
  source_id text,                         -- 原始數據 ID

  -- 擴充屬性 (低頻查詢用 JSONB)
  metadata jsonb default '{}',
  /*
    metadata 結構 (依 type 不同)：

    type='station':
    {
      "operator": "TokyoMetro",
      "lines": ["Ginza", "Hibiya"],
      "exits": ["A1", "A2", "B1"],
      "connecting_stations": ["odpt:Toei.Ueno-Okachimachi"]
    }

    type='poi':
    {
      "category": "temple",
      "opening_hours": "6:00-17:00",
      "admission_fee": 0,
      "official_url": "https://..."
    }

    type='bike_station':
    {
      "system_id": "docomo-cycle-tokyo",
      "capacity": 20
    }
  */

  -- 外部連結
  external_links jsonb default '{}',
  /*
    {
      "barrier_free_map": "https://...",
      "official_site": "https://...",
      "google_maps": "https://..."
    }
  */

  created_at timestamptz default now(),
  updated_at timestamptz default now()
);
```

```
-- 索引
create index idx_nodes_city on nodes(city_id);
create index idx_nodes_type on nodes(type);
create index idx_nodes_geohash on nodes(geohash);
create index idx_nodes_vibe on nodes(vibe);
create index idx_nodes_hub on nodes(is_hub) where is_hub = true;
create index idx_nodes_parent on nodes(parent_hub_id);
create index idx_nodes_location on nodes using gist(location);
create index idx_nodes_lines on nodes using gin(line_ids);
```

2.3 facilities (設施表 - L3 正規化)

```
sql
```

```
create table facilities (
  id uuid primary key default gen_random_uuid(),
  node_id text references nodes(id) on delete cascade,
  city_id text references cities(id),

  -- 基本資訊
  type text not null,                  -- 見下方 enum
  name jsonb,                          -- {"ja": "多目的トイレ"}

  -- 位置 (相對於 node)
  distance_meters int,
  direction text,                      -- '改札内北側', '東口出て右50m'
  floor text,                          -- 'B1', '1F', '2F'

  -- L3 供給標籤 (Supply Tags) - 正規化
  has_wheelchair_access boolean default false,
  has_baby_care boolean default false,    -- 尿布台/哺乳室
  is_free boolean default true,
  is_24h boolean default false,

  -- 即時狀態 (由 n8n 更新)
  current_status text default 'unknown',  -- 'available', 'busy', 'closed', 'unknown'
  status_updated_at timestamptz,

  -- 擴充屬性
  attributes jsonb default '{}',
  /*
    type='locker':
    {
      "size": "large",
      "price": 700,
      "payment": ["cash", "suica"],
      "count": 10,
      "provider": "ecbo"
    }

    type='toilet':
    {
      "has_ostomate": true,
      "has_changing_board": true
    }
  */

  -- 商業導流
  booking_url text,                -- ecbo cloak 預約連結

  -- 數據來源
  source_dataset text not null,     -- 'osm', 'scraper', 'manual'

  created_at timestamptz default now(),
  updated_at timestamptz default now()
);

-- 設施類型 (可查詢用)
comment on column facilities.type is '
  toilet      - 一般廁所
  toilet_accessible - 無障礙廁所
  locker_small   - 小型置物櫃
  locker_medium  - 中型置物櫃
  locker_large   - 大型置物櫃
  locker_service - 寄放服務 (ecbo cloak)
  bench       - 休息座椅
  charging     - 充電站/租借行動電源
  atm        - ATM
  tourist_info   - 觀光案內所
  elevator     - 電梯
  escalator     - 電扶梯
  wifi          - 免費 WiFi
';

-- 索引
create index idx_facilities_node on facilities(node_id);
create index idx_facilities_city on facilities(city_id);
create index idx_facilities_type on facilities(type);
create index idx_facilities_wheelchair on facilities(has_wheelchair_access) where has_wheelchair_access = true;
create index idx_facilities_status on facilities(current_status);
```

2.4 facility_suitability (適用標籤 - L3 情境索引)

```
sql
```

```
-- 獨立表：讓 AI 能快速查詢「適合 X 情境」的設施
create table facility_suitability (
  id uuid primary key default gen_random_uuid(),
  facility_id uuid references facilities(id) on delete cascade,

  -- L3 適用標籤 (Suitability Tags)
  tag text not null,
  /*
    可用標籤：
    - good_for_waiting    適合久候 (有椅子、有遮蔽、有WiFi)
    - work_friendly       適合臨時工作 (有電源、有WiFi、安靜)
    - quiet_zone          安靜避難 (遠離人潮)
    - luggage_friendly    適合大行李 (有大型置物櫃、無障礙)
    - family_friendly     適合親子 (有尿布台、有哺乳室)
    - rain_shelter        可避雨
  */

  confidence float default 1.0,          -- 0-1，手動標註 = 1.0，AI 推測 < 1.0
  source text default 'manual',          -- 'manual', 'ai_inferred', 'user_feedback'

  created_at timestamptz default now()
);

-- 索引 (核心查詢：找出某 node 周邊「適合等待」的地方)
create index idx_suitability_tag on facility_suitability(tag);
create index idx_suitability_facility on facility_suitability(facility_id);

-- 複合索引 (加速常見查詢)
create index idx_suitability_tag_confidence on facility_suitability(tag, confidence desc);
```

2.5 shared_mobility (共享運具站點)

```
sql
```

```sql
-- 獨立表：GBFS 數據需要頻繁更新，獨立出來避免污染 nodes 表
create table shared_mobility_stations (
  id text primary key,                    -- 'docomo:12345'
  node_id text references nodes(id),      -- 關聯到 nodes 表 (type='bike_station')
  city_id text references cities(id),

  -- 系統資訊
  system_id text not null,                -- 'docomo-cycle-tokyo', 'hellocycling', 'luup'
  system_name text,                       -- 'ドコモ・バイクシェア'

  -- 靜態資訊
  name text not null,
  location geography(point, 4326) not null,
  capacity int,

  -- 車輛類型 (LUUP 有多種)
  vehicle_types text[] default array['bike'],  -- ['bike', 'ebike', 'scooter']

  -- 即時狀態 (由 n8n 每分鐘更新)
  bikes_available int default 0,
  docks_available int default 0,
  is_renting boolean default true,
  is_returning boolean default true,
  status_updated_at timestamptz,

  -- 商業導流
  app_deeplink text,                      -- 'https://luup.sc/...'

  created_at timestamptz default now()
);

-- 索引
create index idx_mobility_city on shared_mobility_stations(city_id);
create index idx_mobility_system on shared_mobility_stations(system_id);
create index idx_mobility_location on shared_mobility_stations using gist(location);
create index idx_mobility_available on shared_mobility_stations(bikes_available) where bikes_available > 0;
```

---

3. 用戶與行為表格

3.1 users (用戶)

```
sql
```

```
create table users (
  id uuid primary key references auth.users(id),

  -- 基本資訊
  display_name text,
  preferred_language text default 'ja',

  -- LINE 整合 (Trip Guard 推播用)
  line_user_id text unique,
  line_notify_token text,

  -- 偏好設定
  preferences jsonb default '{}',
  /*
    {
      "accessibility_needs": ["wheelchair", "elevator"],
      "preferred_transport": ["rail", "bike"],
      "avoid_crowds": true
    }
  */

  -- 狀態
  is_guest boolean default true,          -- 訪客模式

  created_at timestamptz default now(),
  updated_at timestamptz default now()
);
```

3.2 trip_subscriptions (Trip Guard 訂閱)

```
sql
```

```
create table trip_subscriptions (
  id uuid primary key default gen_random_uuid(),
  user_id uuid references users(id) on delete cascade,

  -- 監控目標
  route_ids text[] not null,          -- ['TokyoMetro.Ginza', 'JR-East.Yamanote']
  origin_node_id text references nodes(id),
  destination_node_id text references nodes(id),

  -- 時間條件 (可選)
  active_days int[] default array[0,1,2,3,4,5,6], -- 0=Sun, 1=Mon...
  active_start_time time,              -- 07:00
  active_end_time time,                -- 22:00

  -- 狀態追蹤
  last_known_status jsonb,
  last_notified_at timestamptz,
  notification_cooldown_minutes int default 15,

  is_active boolean default true,

  created_at timestamptz default now()
);

-- 索引
create index idx_trip_user on trip_subscriptions(user_id);
create index idx_trip_active on trip_subscriptions(is_active) where is_active = true;
create index idx_trip_routes on trip_subscriptions using gin(route_ids);
```

3.3 nudge_logs (意圖日誌 - 核心商業數據)

```
sql
```

```
create table nudge_logs (
  id uuid primary key default gen_random_uuid(),
  city_id text references cities(id),

  -- Session 追蹤 (訪客也有)
  session_id text not null,
  user_id uuid references users(id),      -- 可為 null (訪客)

  -- 觸發情境
  trigger_type text not null,            -- 'map_tap', 'search', 'gps', 'qr'
  trigger_node_id text references nodes(id),
  trigger_location geography(point, 4326),

  -- 用戶意圖 (Intent Data 核心)
  query_type text not null,              -- 'route', 'facility', 'info', 'emergency'
  intended_destination_id text,          -- 想去的地方
  intended_destination_text text,        -- 自由輸入的文字
  query_raw text,                        -- 原始查詢內容

  -- AI 回應
  response_type text,                    -- 'action_cards', 'facility_list', 'chat'
  response_summary text,                 -- AI 建議摘要

  -- L4 Action Cards 詳情
  action_cards jsonb,
  /*
    [
      {
        "type": "transit",
        "title": "搭銀座線",
        "provider": "TokyoMetro",
        "eta_minutes": 3,
        "selected": false
      },
      {
        "type": "taxi",
        "title": "搭 GO Taxi",
        "provider": "go_taxi",
        "price_estimate": 1200,
        "deeplink": "https://...",
        "selected": true,
        "clicked_at": "2025-12-14T10:30:00Z"
      },
      {
        "type": "bike",
        "title": "騎 LUUP",
        "provider": "luup",
        "deeplink": "https://...",
        "selected": false
      }
    ]
  */

  -- 用戶行為追蹤
  card_selected int,                  -- 選了第幾張卡 (0-based)
  deeplink_clicked boolean default false,
  clicked_provider text,               -- 'go_taxi', 'luup', 'ecbo'

  -- 商業價值追蹤
  potential_revenue_type text,         -- 'taxi_cpa', 'locker_commission', 'none'

  created_at timestamptz default now()
);

-- 索引
create index idx_nudge_city on nudge_logs(city_id);
create index idx_nudge_session on nudge_logs(session_id);
create index idx_nudge_query_type on nudge_logs(query_type);
create index idx_nudge_created on nudge_logs(created_at);
create index idx_nudge_clicked on nudge_logs(deeplink_clicked) where deeplink_clicked = true;
create index idx_nudge_provider on nudge_logs(clicked_provider) where clicked_provider is not null;

-- 商業分析用複合索引
create index idx_nudge_revenue on nudge_logs(city_id, potential_revenue_type, created_at);
```

---

4. 快取與即時數據 (Redis)

```
# L2 即時狀態 (TTL: 60秒)
l2:status:{node_id} = {
  "transit_status": "normal" | "delayed" | "suspended",
  "delay_minutes": 0,
  "crowding_level": 75,
  "crowding_trend": "increasing",
  "updated_at": "2025-12-14T10:30:00Z"
}

# 路線異常 (TTL: 300秒)
l2:disruption:{route_id} = {
  "status": "delayed",
  "cause": "混雜",
  "resume_estimate": "2025-12-14T11:00:00Z",
  "affected_sections": ["Ueno", "Asakusa"],
  "updated_at": "2025-12-14T10:25:00Z"
}

# 共享運具即時 (TTL: 60秒)
gbfs:status:{station_id} = {
  "bikes": 5,
  "docks": 10,
  "updated_at": "2025-12-14T10:30:00Z"
}

# 過度旅遊警示 (TTL: 300秒)
overtourism:{node_id} = {
  "level": "warning",
  "crowding_index": 1.35,
  "suggested_alternatives": ["node_id_1", "node_id_2"],
  "updated_at": "2025-12-14T10:30:00Z"
}

# Hub Persona 快取 (TTL: 3600秒)
persona:{hub_id} = "完整的 Persona Prompt 文字..."
```

---

5. 查詢範例

5.1 找出某節點周邊「適合等待」的設施

```
sql
```

```
-- 高效查詢：使用正規化欄位 + 索引
select
  f.id,
  f.type,
  f.name,
  f.distance_meters,
  f.direction
from facilities f
join facility_suitability s on s.facility_id = f.id
where f.node_id = 'odpt:TokyoMetro.Ueno'
  and s.tag = 'good_for_waiting'
  and s.confidence >= 0.8
order by f.distance_meters asc
limit 5;
```

5.2 找出有空位的大型置物櫃

sql

```
select
  f.id,
  f.name,
  f.distance_meters,
  f.attributes->>'price' as price,
  f.booking_url
from facilities f
where f.node_id = 'odpt:JR-East.Tokyo'
  and f.type = 'locker_large'
  and f.current_status = 'available'
order by f.distance_meters asc;
```

5.3 取得節點的繼承 Persona

```
sql

-- 如果是 Hub，直接取；如果是 Spoke，取 parent_hub 的
select
  coalesce(
    n.persona_prompt,
    (select persona_prompt from nodes where id = n.parent_hub_id)
  ) as effective_persona
from nodes n
where n.id = 'odpt:TokyoMetro.Iriya';
```

5.4 商業轉換率分析

```
sql

-- 計算各 Provider 的點擊率
select
  clicked_provider,
  count(*) as clicks,
  count(*) * 100.0 / sum(count(*)) over () as click_rate_pct
from nudge_logs
where deeplink_clicked = true
  and created_at >= now() - interval '7 days'
group by clicked_provider
order by clicks desc;
```

5.5 找出同路線的 Hub (繼承演算法優化)

```
sql
```

```
-- 用於 resolveNodePersona：優先找同路線的 Hub
select h.*
from nodes h
where h.is_hub = true
  and h.line_ids && (
    select line_ids from nodes where id = 'odpt:TokyoMetro.Iriya'
  )
order by
  -- 同路線數量越多越優先
  array_length(
    array(select unnest(h.line_ids) intersect select unnest(
      (select line_ids from nodes where id = 'odpt:TokyoMetro.Iriya')
    )), 1
  ) desc nulls last
limit 1;
```

---

6. 資料遷移與版本控制

6.1 Supabase Migration 檔案結構

```
supabase/migrations/
├── 20251214000001_create_cities.sql
├── 20251214000002_create_nodes.sql
├── 20251214000003_create_facilities.sql
├── 20251214000004_create_suitability.sql
├── 20251214000005_create_shared_mobility.sql
├── 20251214000006_create_users.sql
├── 20251214000007_create_trip_subscriptions.sql
├── 20251214000008_create_nudge_logs.sql
└── 20251214000009_create_indexes.sql
```

6.2 未來擴充：新增設施類型

sql

```sql
-- 假設要新增 AED (自動體外除顫器)
-- 不需要 ALTER TABLE，直接插入新數據

insert into facilities (node_id, city_id, type, name, distance_meters, direction, source_dataset)
values (
  'odpt:TokyoMetro.Ueno',
  'tokyo_taito',
  'aed',  -- 新類型
  '{"ja": "AED (上野駅改札内)"}',
  0,
  '改札内、駅務室横',
  'manual'
);

-- 若需要新的 Suitability Tag，也是直接插入
insert into facility_suitability (facility_id, tag, source)
values (
  (select id from facilities where type = 'aed' and node_id = 'odpt:TokyoMetro.Ueno'),
  'emergency_ready',
  'manual'
);
```

---

7. 效能考量

查詢類型	預期 QPS	優化策略
節點查詢 (by id)	100	Primary Key
鄰近節點 (by location)	50	GiST Index + Geohash
設施查詢 (by node + type)	200	Composite Index
適用標籤查詢	100	Tag Index
商業分析	1 (批次)	時間分區 + 複合索引

瓶頸預防：

- `nudge_logs` 會快速成長，建議 3 個月後啟用 Table Partitioning (by created_at)
- 共享運具即時狀態走 Redis，不直接查 Supabase