BigQuery – معايير وتنفيذ

# **BigQuery – معايير وتنفيذ**

## **0) الهدف والنطاق**

* BigQuery هو **Data Warehouse مركزي** لتجميع بيانات التطبيق (أحداث GA4/Firebase) + بيانات الأعمال (الطلبات، العملاء، المخزون…).
* يُستخدم للتحليلات المتقدمة: Retention, Funnels, LTV, RFM, Attribution، ولوحات الأداء التشغيلية.

## **1) الهيكلة العامة (Projects → Datasets → Tables)**

* **المشاريع (GCP Projects):** zahraah-dev, zahraah-stg, zahraah-prod.
* **المجموعات (Datasets):**
  + raw\_analytics (بيانات خام من GA4/Firebase).
  + raw\_business (مصادر خارجية: Odoo/ERP/CSV).
  + staging (تنظيف وتوحيد).
  + mart\_product (جداول تحليل نهائية للاستهلاك BI).
* **البيئات:** فصل صارم dev/stg/prod (لا مشاركة جداول بين البيئات).

## **2) التسمية والمعايير (Naming & Types)**

* **حروف صغيرة + شرطات سفلية**: orders\_fact, events\_ga4\_raw.
* الجداول التحليلية: لاحقة واضحة (\_fact, \_dim, \_mart).
* الأعمدة الأساسية موحّدة:
  + event\_date (DATE, توقيت UTC)
  + event\_timestamp (TIMESTAMP)
  + user\_id (STRING) – مع تجنّب PII
  + session\_id (STRING/INT64)
  + device\_id (STRING)
  + order\_id (STRING)
  + sku, spu (STRING)
  + العملات: amount (NUMERIC), currency (STRING, ISO 4217)
* **المنطقة الزمنية:** تخزين **UTC** فقط. تحويل العرض يتم في طبقة BI.

## **3) التقسيم والتكتّل (Partitioning & Clustering)**

* **Partitioning** إلزامي حسب event\_date أو order\_date (TYPE: DATE).
* **Clustering** إلزامي لتحسين الاستعلام:
  + أحداث: (user\_id, event\_name)
  + طلبات: (user\_id, order\_status)
  + منتجات: (sku, category)
* ممنوع إنشاء جداول كبيرة بدون Partition/Cluster.

## **4) إدخال البيانات (Ingestion)**

* **GA4/Firebase Export**:
  + تفعيل التصدير اليومي/اللحظي إلى raw\_analytics.
  + جداول معيارية: events\_\* (اليومي)، events\_intraday\_\*.
* **مصادر الأعمال (Odoo/ERP/CSV)**:
  + تحميل إلى raw\_business عبر Cloud Storage + DataFlow/Cloud Run أو Scheduled Load.
  + صيغة CSV/Parquet مع **Schema موحّد**.
* **تتبّع النسخ**: أعمدة load\_at (TIMESTAMP) وsource\_file (STRING).

## **5) الطبقات التحويلية (ELT)**

* **Staging (staging)**:
  + تنظيف الحقول (nulls, types, trimming).
  + توحيد مفاتيح user\_id, device\_id, order\_id.
  + توحيد العملات إلى عملة مرجعية (مثلاً USD) + صرف يومي إن لزم.
* **Marts (mart\_product)**:
  + جداول **Fact/Dim**:
    - fact\_events, fact\_orders, fact\_sessions
    - dim\_users, dim\_products, dim\_categories
  + جداول تحليل جاهزة: mart\_retention\_daily, mart\_funnel, mart\_ltv\_monthly, mart\_rfm, mart\_attribution

## **6) الجودة والتحقق (Data Quality)**

* قواعد فحص **يومية** على staging وmart\_product:
  + لا توجد تواريخ مستقبلية.
  + نسبة null في مفاتيح الربط ≤ 0.5%.
  + توازن الأعداد بين raw وstaging (±5%).
  + فريدية المفاتيح: order\_id, sku, (user\_id, session\_id, event\_timestamp).
* كل فحص يكتب إلى جدول staging.dq\_checks\_log ويولّد تنبيه إذا فشل.

## **7) الأمن والصلاحيات (Security & IAM)**

* الوصول عبر **Service Accounts** فقط، لا مستخدمين بشريين على الجداول الحساسة.
* أدوار:
  + data.viewer للقراءة (BI)
  + data.developer للتحويلات داخل dev/stg
  + data.admin محدود للإدارة/الإنتاج
* **داتا حساسة/PII**: محظورة. لو لزم (بريد/هاتف) تُطبَّق **Tokenization/Hashing** وتُخزن في Dataset معزول بإذن خاص.
* تمكين **Column-level security** أو **Row-level security** للحقول الحساسة.

## **8) الأداء والتكلفة (Performance & Cost)**

* إلزام استخدام **Filters** على أعمدة Partition في جميع الاستعلامات.
* تقارير **Bytes Processed** تُراجع أسبوعيًا. أي استعلام > 2GB يُعاد تحسينه.
* استخدام **Materialized Views**/Tables للطلبات الثقيلة المتكررة.
* عدم استخدام SELECT \* في الإنتاج. تحديد الأعمدة فقط.
* تخزين النتائج الوسيطة كـ **Tables** بدلاً من Subqueries متداخلة جدًا.

## **9) الجداول الأساسية (حدّ أدنى)**

### **9.1 staging.fact\_events**

* مفاتيح: user\_id, session\_id, event\_timestamp, event\_name
* أعمدة رئيسية: event\_date, source, campaign, medium, device\_id, screen, event\_params (STRUCT/JSON)
* Partition: event\_date
* Cluster: (user\_id, event\_name)

### **9.2 staging.fact\_orders**

* مفاتيح: order\_id
* أعمدة: user\_id, order\_date, status, payment\_method (COD/…),  
  subtotal\_amount, discount\_amount, shipping\_amount, total\_amount, currency
* Partition: order\_date
* Cluster: (user\_id, status)

### **9.3 staging.dim\_users**

* مفاتيح: user\_id
* أعمدة: region, signup\_date, customer\_type (new/returning/VIP), cod\_risk\_band
* Partition: signup\_date
* Cluster: (region, customer\_type)

### **9.4 staging.dim\_products**

* مفاتيح: sku
* أعمدة: spu, title, brand, category, sub\_category, color, size, price, cost
* Cluster: (category, brand)

## **10) نماذج تحليلات إلزامية (Marts)**

### **10.1 الاحتفاظ (Retention)**

* جدول: mart\_product.mart\_retention\_daily
* منطق: Cohorts حسب signup\_date مع فواصل D1/D7/D30، وحسب قناة الاكتساب.
* مؤشرات: Retention%، Churn%.

### **10.2 مسار الشراء (Funnels)**

* جدول: mart\_product.mart\_funnel
* مراحل إلزامية: view\_item → add\_to\_cart → begin\_checkout → purchase
* تفصيل بالقناة/الحملة/المنطقة/الفئة.

### **10.3 قيمة العميل (LTV)**

* جدول: mart\_product.mart\_ltv\_monthly
* منطق: مجموع صافي الربح/الإيراد على مدى N أشهر منذ أول شراء.
* تقسيم شرائح حسب قناة الاكتساب وطريقة الدفع (COD/إلكتروني).

### **10.4 RFM**

* جدول: mart\_product.mart\_rfm
* الحقول: Recency, Frequency, Monetary + Score 1–5 + Segment Label.

### **10.5 Attribution**

* جدول: mart\_product.mart\_attribution
* منطق: (Last touch) افتراضيًا + خيار (Position-based 40/20/40) عند الحاجة.
* المصدر: أبعاد الحملة من GA4 + ربط order\_id.

## **11) الجدولة والأتمتة (Scheduling)**

* جميع التحويلات عبر **Scheduled Queries** أو **Dataform/dbt**.
* تكرار:
  + staging: كل 1–3 ساعات.
  + marts: يومي (05:00 UTC) + تحديثات سريعة كل 6 ساعات لضروريّات التشغيل.
* ترتيب التنفيذ Dependencies واضح (staging قبل marts).
* كتابة حالة التنفيذ في staging.etl\_runs\_log.

## **12) التوثيق والإصدارات**

* كل جدول/عرض له **README** موجز داخل وصف الجدول (Table Description).
* **Data Catalog** مفعّل مع Tags: owner, SLA, PII, refresh\_rate.
* التغييرات تُدار عبر Git (Dataform/dbt repo) مع PR Review + خطة ترحيل (Migration Plan).

## **13) المراقبة والتنبيهات**

* لوحات **Looker Studio** لفِرق: المنتج، التسويق، العمليات.
* تنبيهات Cloud Monitoring عند:
  + فشل Scheduled Queries.
  + انحراف حجم البيانات ±30%.
  + فشل قواعد DQ.
* Runbook مختصر للإصلاح السريع (إعادة تشغيل مهمة/إرجاع نسخة).

## **14) الحفظ والسياسات (Retention & Backup)**

* raw\_\*: احتفاظ 180 يومًا (أو حسب الحاجة التنظيمية).
* staging: 365 يومًا.
* marts: 730 يومًا.
* تمكين **Table Snapshots** للجداول الحرجة أسبوعيًا.

## **15) خصوصية البيانات (Privacy)**

* لا تُخزّن PII مباشرة (هاتف/إيميل) في جداول عامة.
* عند الحاجة: **Hash (SHA256 + Salt)** وخزن الـSalt في Secret Manager.
* الامتثال لسياسات الخصوصية الداخلية + أي متطلبات قانونية محليّة.

## **16) Definition of Done (لكل إضافة/تعديل)**

1. Partition/Cluster مطبّقة.
2. وصف الجدول/الأعمدة مكتمل.
3. اختبارات DQ ناجحة ومُسجّلة.
4. تكلفة الاستعلام < 2GB أو موثّقة بمبرّر.
5. لوحات/Views محدّثة (إن تأثرت).
6. PR مُراجع + خطة ترحيل مُنفّذة.

## **17) أمثلة استعلامات إلزامية (تُحفظ كـ Saved Queries)**

* **D7 Retention** لكل قناة.
* **Funnel Conversion** حسب المنطقة والقناة.
* **Top SKUs** حسب المبيعات والهوامش.
* **LTV بعد 90/180 يوم** حسب القناة وطريقة الدفع.
* **RFM Segments** لتغذية حملات التسويق (VIP/Dormant/Winback).

**ملاحظة تنفيذية:**يفضّل استخدام **Dataform أو dbt** لإدارة التحويلات (Versioning, Dependencies, Tests) مع تخزين الكود في GitHub، وربطه بـ CI لتنفيذ فحوصات نمطية (lint) وDQ قبل الجدولة على prod.

Data Warehouse & ELT/ETL on BigQuery + Dataform

# **Data Warehouse & ELT/ETL on BigQuery + Dataform**

# **1) Mechanism & Dependencies (Executive Summary)**

**Data Sources** (MySQL/API services, Firebase/GA4 Export, Payments & Shipping, Messaging, Ads)  
⟶ **RAW layer** in BigQuery (raw\_\*) via connectors/CDC or file drops (GCS)  
⟶ **Dataform** transformations (stg\_\* ⟶ ods\_\* ⟶ dim\_\*/fact\_\*/mart\_\*)  
⟶ **Dashboards & Reports** (Metabase/Looker, extracts, or data APIs).  
Surrounding controls: **IAM & Secrets**, **CMP/consents**, **freshness & quality alerts**, **cost quotas**, **separate Dev/Stg/Prod**.

# **2) Objectives**

* One **source of truth** for core business metrics (sales, conversion, retention, channels).
* **Multi-source consolidation** with **idempotent** loads and **late-arrivals** handling.
* **Transparency & repeatability** via named layers, contracts, and naming standards.
* **Security & privacy by design** (PII minimization, DSAR flows, consent).
* **Cost control & performance** (partitioning, clustering, materialization where needed).

# **3) Scope**

* BigQuery DWH for: customers, sessions/events, products, pricing/promotions, cart, orders, payments, shipping, returns, pricing, marketing/ads (GA4/Meta/Google Ads), communications (SMS/WA/Email/Push), support.
* Isolated environments: **zahraah-dwh-{dev|stg|prod}** with mirrored dataset names.

# 

# 

# 

# **4) Architecture (Layered)**

| **Layer** | **Dataset Prefix** | **Purpose** | **Examples** |
| --- | --- | --- | --- |
| RAW (Landing) | raw\_\* | As-is dumps, immutable | raw\_mysql\_orders, raw\_ga4\_events\_\*, raw\_ads\_google\_daily |
| STAGING | stg\_\* | Type hygiene, JSON flatten, dedup | stg\_orders, stg\_order\_items, stg\_ga4\_events |
| ODS (Core) | ods\_\* | Unified operational model (IDs, enums, time) | ods\_customers, ods\_products, ods\_payments |
| MARTS (BI) | dim\_\*, fact\_\*, mart\_\* | Star schema for analytics | dim\_customer, dim\_product, fact\_orders, mart\_daily\_sales |
| SANDBOX | sandbox\_\* | Analyst experiments (guard-railed) | sandbox\_marketing |

Default to **views + incremental tables**; materialize only where it pays off.

# **5) Data Sources & Contracts**

## **5.1 App DB (MySQL)**

* **Ingest:** Prefer **CDC** (Datastream/Airbyte) into GCS ⟶ load into raw\_mysql\_\*.  
  Fallback: time-boxed CSV/Parquet drops with **watermark** (updated\_at UTC).
* **Contract:** Every table has a **PK** and **updated\_at (UTC)**; stable data types.
* **Cadence:** near-real-time (CDC) or every 15–60 minutes.

## **5.2 GA4/Firebase**

* **Ingest:** **GA4 BigQuery Export** into raw\_ga4\_\* (same region as DWH).
* **Contract:** user\_pseudo\_id, event\_timestamp, event\_date, event keys; set **user\_id** upon auth (per your IAM policy).

## **5.3 Marketing & Ads**

* **Google Ads / Meta Ads** via connectors (Airbyte/official/Apps Script) into raw\_ads\_\* daily.
* **Contract:** cost, clicks, impressions, campaign/adgroup/ad ids, UTM.

## **5.4 Payments, Shipping, Messaging**

* Gateways, shippers, SMS/WA providers into raw\_payments\_\*, raw\_shipments\_\*, raw\_msgs\_\*.
* **Contract:** operation id, status, UTC timestamps, amounts/currencies.

Each source has a **Data Contract file**: columns, types, keys, cadence, owner, **failure policy**.

# **6) Data Modeling (Core & Marts)**

## **6.1 Core Dimensions**

* **dim\_date**, **dim\_customer** (SCD2 for key attributes), **dim\_product** (SCD2 for price/category), **dim\_channel**, **dim\_device**, **dim\_region**.

## **6.2 Facts**

* **fact\_orders** (order/customer/date/channel/payment),
* **fact\_order\_items** (SKU level),
* **fact\_sessions/fact\_events** (from GA4, link user\_id when present),
* **fact\_payments**, **fact\_refunds**, **fact\_shipments**, **fact\_messages**.

## **6.3 Golden Metrics (Canonical)**

* **GMV** = Σ(item\_price × qty) before discounts/shipping.
* **Net Sales** = GMV − discounts − returns − non-recoverable taxes.
* **AOV**, **CR**, **D7/D30 retention**, **FRM segments**, **CAC**, **ROAS**.

Define as **views/functions** to prevent metric drift.

# **7) Implementation Standards (BigQuery + Dataform)**

## **7.1 Naming & Conventions**

* **Datasets:** raw\_\*, stg\_\*, ods\_\*, bi\_\* (with dim\_, fact\_, mart\_).
* **Tables:** raw\_ga4\_events\_\*, stg\_orders, ods\_customers, fact\_orders.
* **Columns:** snake\_case, \*\_at as **UTC TIMESTAMP**.
* **Keys:** \*\_id (natural or surrogate); generate **surrogate keys** where needed.

## **7.2 Partitioning & Clustering**

* **Partition** by event\_date or DATE(created\_at).
* **Cluster** for common access paths: user\_id, order\_id, campaign\_id.
* **Rule:** no large tables without partitioning; marts at least daily partitions.

## **7.3 Loads & Transforms**

* **RAW:** append-only, immutable.
* **STAGING:** type fixes + **dedup** by ROW\_NUMBER() OVER (PK ORDER BY updated\_at DESC).
* **ODS:** **MERGE** (UPSERT) using updated\_at watermark.
* **MARTS:** **incremental** on last partition(s); monthly full rebuild as needed.

## **7.4 Idempotency & Late Data**

* Natural idempotency keys (order\_id, payment\_id, GA4 event\_id).
* **Late Arrivals:** **T+3 days** rolling recompute for recent partitions.
* All transforms **replayable** and **stateless**.

## **7.5 Data Quality (Dataform Assertions)**

* **EXPECT NOT NULL (PKs)**, **Unique PK**, **FK exists**, **enum values**, **value ranges** (amounts ≥ 0).
* **Threshold tests:** daily row count ≥ 80% of 7-day mean (alert).
* **Schema tests:** alert on source column add/remove vs contract.

## **7.6 SQL/SQLX Style (Dataform)**

* No SELECT \* in upper layers.
* Document exposed fields; use shared macros for **dates/timezones**, **currency handling**, **JSON flatten**, **SCD2**.
* Each model includes **description, owner, schedule, freshness SLA** in code.

## **7.7 Scheduling & Orchestration**

* Order: RAW ingests (outside Dataform) ⟶ STG (15–30m) ⟶ ODS (30–60m) ⟶ MARTS (hourly/daily).
* Explicit **dependencies** between models.
* Failure alerts to Slack/Email with run links.

## **7.8 Environments & CI/CD**

* Git branches: feature → develop (stg) → main (prod).
* **Code review mandatory** + **preview runs** on dev.
* Deploy via **Service Accounts** per environment with least privilege.

## **7.9 Documentation & Catalog**

* Dataform docs + column descriptions required.
* Publish a **Data Dictionary** for primary dims/facts to Metabase/Looker.

# **8) Security & Privacy (IAM / CLS / RLS / CMP)**

* Separate **GCP projects & SAs** for dev/stg/prod.
* BigQuery roles:
  + **dataViewer** on bi\_\* for analysts,
  + **dataEditor** on stg\_/ods\_ for data team,
  + **jobUser** to run queries,
  + deny direct access to raw\_\* except data owners.
* **Column-Level Security**: mask/obfuscate PII (phone/email) in marts; encrypt/hash where appropriate.
* **Row-Level Security** for geo/channel scoping if needed.
* **DSAR (delete/rectify/export):** procedure to obfuscate PII in ods\_/bi\_ while preserving aggregates.
* **Retention:** RAW 180–365d; STG/ODS 730d; MARTS ≥ 24 months; monthly GCS backups.

# **9) Observability & Quality**

* **Freshness** SLAs per source (alert on deviation).
* **Row counts, nulls, PK uniqueness** on core tables.
* Simple **anomaly detection** (spikes/drops in sales/sessions).
* **Logging** for all ingests/transforms (success/failure/duration/size).
* **Dashboards:** Pipeline Health, Data Freshness, Test Failures, Cost by Dataset/Query.

# **10) Cost & Performance**

* Single **region** (match GA4 Export).
* Always push **partition filters**; leverage **clustering**.
* **Materialized Views** for hot daily metrics.
* Load files as **Parquet/ORC**.
* Avoid wide joins on RAW; normalize in STG/ODS first.
* Consider **Slot reservations** if workload is steady; otherwise on-demand with alerts.
* Set **query caps/timeouts** in BI tools.

# **11) SLIs / SLOs**

| **Metric (SLI)** | **SLO** | **Measurement** |
| --- | --- | --- |
| fact\_orders freshness | ≤ 30 min lag | ODS vs RAW timestamps |
| Dataform daily success rate | ≥ 99% | Successful runs over 30 days |
| Critical quality test failures | 0 days/month | PK/Null/Uniqueness assertions |
| Daily query cost | Within budget | BigQuery Billing monitors |
| BI query p95 | ≤ 5s | Metabase/Looker stats |

If any two SLOs miss in consecutive periods, trigger a **remediation plan** and config review.

# **12) Runbooks**

## **RB-1: RAW source delay (e.g., GA4 export outage)**

1. Freshness alert ⟶ check source status.
2. If external outage: **pause dependent marts** and show status flag in reports.
3. On recovery: **backfill** only the missing window (T down to T−N).

## **RB-2: Critical quality test failure**

1. **Break the build** for affected ODS/MART.
2. Identify root cause (schema/nulls/duplicates), fix in STG.
3. Rerun chain; record an incident note.

## **RB-3: Unexpected cost spike**

1. Identify top costly queries (bytes/scans).
2. Enforce **required partition filters**, add **clustering** or an **MV**.
3. Revert recent changes and apply temporary BI tool limits.

## **RB-4: DSAR request**

1. Collect user\_id/identifiers mapping.
2. Run procedure to obfuscate/mask PII in ods\_/bi\_; tag as anonymized.
3. Validate aggregates remain stable.

## **RB-5: Source schema change**

1. Alert pinpoints table/column.
2. Update source contract + stg\_\* to accept/ignore new column.
3. Update tests; rerun & deploy.

# **13) “Owner / Cadence / Privacy” Sample Table**

| **Table** | **Owner** | **Cadence** | **PII** | **Retention** |
| --- | --- | --- | --- | --- |
| raw\_mysql\_orders | Backend | 15m (CDC) | No | 365d |
| stg\_orders | Data Team | 30m | No | 730d |
| ods\_customers | Data Team | 60m | Yes (masked) | 730d |
| fact\_orders | Data Team | 30m | No | Permanent |
| fact\_sessions | Data Team | Hourly/Daily | No | 24 mo |

# **14) Implementation Checklist**

**A) Foundation**

* Create separate GCP projects (dev/stg/prod) + SAs.
* Choose and lock region (align with GA4 Export).
* Set up RAW connectors (Datastream/Airbyte/jobs).
* Create datasets: raw\_\*, stg\_\*, ods\_\*, bi\_\*.

**B) Dataform**

* Initialize repo (Git + envs).
* Build model DAG with explicit dependencies.
* Enable **assertions** (critical tests) + alerts.
* Make heavy models **incremental** (partition + merge).

**C) Quality & Security**

* Source data contracts + schema tests.
* CLS/RLS for PII in marts.
* Document & test DSAR procedure.

**D) Performance & Cost**

* Partition/cluster all large tables.
* Materialize hot daily metrics.
* Set Billing alerts & quotas.

**E) Reporting**

* Connect Metabase/Looker to **bi\_\* only**.
* Publish Golden Metrics as shared views.
* Analyst guide (query scopes/limits).

# **15) Notes**

* Unstable connectivity ⇒ build **hourly marts** with **T−3** rolling backfill.
* Set **user\_id** in GA4/Firebase **after auth**, avoid sending raw PII.
* As scale grows, consider **Reservations (Slots)** for ODS/MART workloads.

علامة التبويب 38

**معايير تنفيذ** لـ **مستودع البيانات وعمليات ELT/ETL على BigQuery + Dataform**، بصيغة جاهزة للإدراج في ملفك الداخلي. رتّبتها على (ملخص اعتماديات → الأهداف → النطاق → المعمارية → عقود ومصادر البيانات → نمذجة → معايير تنفيذ تفصيلية → الأمان والخصوصية → الجودة والمراقبة → التكلفة والأداء → SLO/SLI → Runbooks) مع جداول وقوائم تنفيذ.

# **1) ملخص الآلية والاعتماديات**

**مصادر البيانات** (MySQL/خدمات الـAPI، Firebase/GA4 Export، مزوّدو الدفع والشحن، قنوات الرسائل/الإعلانات)  
⟶ **طبقة RAW (BigQuery datasets: raw\_\*)** عبر موصلات/تفريغ دوري (Datastream/Airbyte/Cloud Functions + Cloud Storage)  
⟶ **Dataform** (تحويلات SQLX مرتبة: stg\_\* ⟶ ods\_\* ⟶ dim\_\*/fact\_\*/mart\_\*)  
⟶ **لوحات وتقارير** (Metabase/Looker، ملفات مُستخرجة، API للبيانات).  
حول ذلك: **IAM وأسرار**، **CMP للخصوصية والموافقات**، **تنبيهات الجودة/التأخر**، **تكلفة/حصص BigQuery**، **Dev/Staging/Prod**منفصلة.

# **2) الأهداف**

* **مصدر حقيقة موحّد** لمقاييس العمل (المبيعات، التحويل، الاحتفاظ، القنوات).
* **تجميع متعدد المصادر** بآليات **Idempotent** والتعامل مع **وبيانات متأخرة**.
* **شفافية** عبر سلالم نمذجة ومعايير تسمية ثابتة، مع **جودة قابلة للقياس**.
* **أمن وامتثال** (PII، حق الحذف/التصحيح، موافقات القنوات).
* **تكلفة تحت السيطرة** وأداء مرتفع (تقسيم/تجميع، فهارس منطقية، ماديّات).

# **3) النطاق**

* مستودع BigQuery لمجالات: العملاء، الجلسات/الأحداث، المنتجات، الأسعار/العروض، السلة، الطلبات، المدفوعات، الشحن، المرتجعات، التسعير، التسويق والإعلانات (GA4/Meta/Google Ads)، الاتصالات (SMS/WA/Email/Push)، دعم العملاء.
* بيئات منفصلة: **zahraah-dwh-dev|stg|prod**، ودatasets متطابقة الأسماء عبر البيئات.

# **4) التصميم المعماري (طبقات)**

| **الطبقة** | **Dataset Prefix** | **الهدف** | **أمثلة** |
| --- | --- | --- | --- |
| RAW (Landing) | raw\_\* | تفريغ كما هو (As-Is)، بدون منطق | raw\_mysql\_orders, raw\_ga4\_events\_\*, raw\_ads\_google\_daily |
| STAGING | stg\_\* | تنظيف أولي، توحيد أنواع، فكّ أعمدة JSON، dedup | stg\_orders, stg\_order\_items, stg\_ga4\_events |
| ODS (Core) | ods\_\* | نموذج تشغيلي موحّد (IDs, enums, time) | ods\_customers, ods\_products, ods\_payments |
| MARTS (BI) | dim\_\*, fact\_\*, mart\_\* | نجميّة: أبعاد وحقائق + جداول تقارير | dim\_customer, dim\_product, fact\_orders, mart\_daily\_sales |
| SANDBOX | sandbox\_\* | تجارب محللين (مراقبة ومحدود) | sandbox\_marketing |

كل طبقة **مادية** (Materialized) عند الحاجة فقط؛ افتراضيًا **Views + Incremental Tables** محسّنة.

# **5) مصادر البيانات وعقودها (Data Contracts)**

## **5.1 قاعدة التطبيق (MySQL)**

* **آلية الجلب:**
  + مفضل: **Datastream** أو **Airbyte** (CDC) إلى **Cloud Storage** ⟶ استيراد إلى raw\_mysql\_\*.
  + بديل مؤقت: تفريغ ليلي CSV/Parquet مع **watermark** (updated\_at).
* **عقد بيانات:** كل جدول يملك **Primary Key** وعمود **updated\_at** (UTC)، وأنواع ثابتة.
* **تواتر:** شبه لحظي (CDC) أو كل 15–60 دقيقة.

## **5.2 GA4/Firebase**

* **الآلية:** **GA4 BigQuery Export** إلى dataset raw\_ga4\_\* (نفس المنطقة).
* **عقد:** user\_pseudo\_id, event\_timestamp, event\_date, مفاتيح الحدث، وربط **user\_id** عند المصادقة (حسب سياسة الهوية لديك).

## **5.3 التسويق والإعلانات**

* **Google Ads/Meta Ads**: سحب يومي عبر موصل (Airbyte/Apps Script/Partner) إلى raw\_ads\_\*.
* **عقد:** تكلفة، نقرات، ظهور، معرّفات الحملة/المجموعة/الإعلان، UTM.

## **5.4 الدفع والشحن والرسائل**

* **Gateways** (HyperPay/others)، **Shippers**، **SMS/WA**: سحب دوري إلى raw\_payments\_\*, raw\_shipments\_\*, raw\_msgs\_\*.
* **عقد:** معرّفات العملية، الحالة، الوقت (UTC)، المبالغ/العملات.

**كل مصدر يملك “ملف عقد”**: الأعمدة، الأنواع، المفاتيح، التواتر، مسؤول المالك (Owner)، **سياسة فشل**.

# **6) نمذجة البيانات (Core + Marts)**

## **6.1 أبعاد أساسية**

* **dim\_date** (تقويم)، **dim\_customer** (SCD2: تتبّع تغيّر السمات الأساسية)، **dim\_product** (SCD2: السعر/التصنيف)، **dim\_channel**, **dim\_device**, **dim\_region**.

## **6.2 جداول حقائق**

* **fact\_orders**: ربط order/customer/date/channel/payment.
* **fact\_order\_items**: على مستوى السطر (SKU).
* **fact\_sessions/fact\_events**: من GA4 (مع ربط user\_id عند توافره).
* **fact\_payments**, **fact\_refunds**, **fact\_shipments**, **fact\_messages**.

## **6.3 مقاييس ذهبية (Canonical Metrics)**

* **GMV** = sum(item\_price \* qty) قبل الخصومات/الشحن.
* **Net Sales** = GMV − خصومات − مرتجعات − ضرائب غير قابلة للاسترداد.
* **AOV** = Net Sales / عدد الطلبات.
* **CR** = الطلبات / الجلسات.
* **Retention (D7/D30)**، **FRM Segments**، **CAC**، **ROAS**.

تُعرَّف المقاييس كـ **Views/Functions** موحّدة لتجنب انحراف التعريف (Metric Drift).

# **7) معايير التنفيذ (BigQuery + Dataform)**

## **7.1 التسمية والبُنى**

* **Datasets:** raw\_\*, stg\_\*, ods\_\*, bi\_\* (تحوي dim\_, fact\_, mart\_).
* **الجداول:** raw\_ga4\_events\_\*, stg\_orders, ods\_customers, fact\_orders.
* **الأعمدة:** snake\_case، الطوابع الزمنية \*\_at **UTC TIMESTAMP**.
* **المفاتيح:** \*\_id نص قصير/عدد، مع **surrogate keys** عند اللزوم.

## **7.2 التقسيم والتجميع (Partition/Cluster)**

* **Partition** على event\_date أو created\_at::DATE.
* **Cluster** حسب **الاستخدام**: user\_id, order\_id, campaign\_id.
* **قاعدة:** لا جداول ضخمة بلا تقسيم. لكل Mart ثقيل، حد أدنى Partition يومي.

## **7.3 التحميلات والتحويلات**

* **RAW:** **Write-Once** (عدم التعديل)، إضافة فقط (Append).
* **STAGING:** تنظيف + **Dedup** بـ ROW\_NUMBER() OVER(PK, updated\_at DESC) ⟶ آخر سجل.
* **ODS:** **MERGE** دوري (UPSERT) باستخدام watermark updated\_at.
* **MARTS:** جداول **Incremental** على البارتيشن الأخير (أمس/اليوم)، مع إعادة بناء شهرية.

## **7.4 Idempotency & Late Data**

* **مفاتيح عدم التكرار:** لكل حدث/عملية مفتاح طبيعي (order\_id, payment\_id, ga4:event\_id).
* **Late Arrivals:** سياسة **T+3 أيام** لإعادة حساب البارتيشنات الأخيرة (Backfill micro).
* **إعادة التشغيل:** جميع التحويلات **قابلة للإعادة** (stateless) وتعتمد على مصادرها، لا افتراضات حالة.

## **7.5 جودة البيانات (Dataform Assertions)**

* **EXPECT NOT NULL (PKs)**, **Unique PK**, **FK exists**، **قيم ضمن قائمة** (enums)، **نطاقات** (مبالغ ≥ 0).
* **Threshold Tests:** عدّ السجلات اليومي لا يقل عن 80% من متوسط 7 أيام (تنبيه).
* **Schema Tests:** تنبيه عند إضافة/حذف أعمدة مصدر بدون عقد محدث.

## **7.6 أسلوب كتابة SQL/SQLX (Dataform)**

* لا SELECT \* في الطبقات العليا.
* الحقول المعروضة في Views **موثّقة ومسمّاة**.
* وحدات مشتركة (Macros) لـ: **التاريخ/المنطقة**، **التعامل مع العملات**، **فك JSON**، **SCD2**.
* كل جدول لديه **وثائق** (description, owner, schedule, freshness SLA) في الكود.

## **7.7 الجدولة والأوركسترة**

* **Dataform Schedules** بالترتيب: RAW imports (خارج Dataform) ⟶ STG (كل 15–30 دقيقة) ⟶ ODS (كل 30–60 دقيقة) ⟶ MARTS (ساعي/يومي).
* **اعتماديات صريحة** بين الجداول في Dataform.
* تنبيهات فشل **إلى Slack/Email** مع رابط الـ run.

## **7.8 البيئات وCI/CD**

* فروع Git: main → prod، develop → stg، feature branches → dev.
* **Code Review إلزامي** + **Preview Runs** على dev قبل الدمج.
* نشر Dataform بــ **Service Accounts** لكل بيئة مع أقل صلاحية.

## **7.9 التوثيق والكتالوج**

* توثيق Dataform + وصف الجداول/الأعمدة الإلزامي.
* Data Dictionary مختصر لأهم الـ dims/facts، يُصدر لـ Metabase/Looker.

# **8) الأمان والخصوصية (IAM/RLS/CLS/CMP)**

* **مشروع/حسابات خدمة منفصلة** لكل بيئة.
* أدوار BigQuery:
  + **dataViewer** للمحللين على bi\_\* فقط،
  + **dataEditor** لفريق البيانات على stg\_/ods\_,
  + **jobUser** لتشغيل الاستعلامات،
  + منع الوصول المباشر لـ raw\_\* إلا لمالكي البيانات.
* **Column-Level Security**: إخفاء/قناع الحقول الحساسة (الهاتف/الإيميل) في الـ marts، وتشفير/تجزئة عند اللزوم.
* **Row-Level Security** عند الحاجة (تجزئة جغرافية/قنوات).
* **حق الحذف/التصحيح:** مسار **DSAR**: تجمع user\_id/phones/emails ⟶ تشغّل **Procedure** لإخفاء/طمس الأعمدة الحساسة في ods\_ وbi\_، والإبقاء على قياسات مجمّعة.
* **سياسات الاحتفاظ:**
  + RAW: 180–365 يومًا (حسب المصدر).
  + STG/ODS: 730 يومًا.
  + Marts: حسب حاجة التحليلات (≥ 24 شهرًا).
  + لقطات نسخ احتياطي شهرية (GCS).

# **9) المراقبة والجودة (Observability)**

* **Freshness** لكل مصدر (انحراف > X دقائق ⟶ إنذار).
* **Row Count & Nulls/PK Uniqueness** على جداول أساسية.
* **Anomaly Detection** بسيط (زحف/هبوط مفاجئ للمبيعات/الجلسات).
* **Logging**: جميع عمليات الاستيراد/التحويل تُسجّل (نجاح/فشل/مدة/حجم بيانات).
* **لوحات**:
  + Pipeline Health، Data Freshness، Test Failures، Cost by Dataset/Query.

# **10) التكلفة والأداء**

* **منطقة واحدة ثابتة** (اختَر منطقة قريبة وتبقى موحّدة لكل المصادر، وGA4 Export في نفس المنطقة).
* **Partition Pruning** و**Clustering** فعّالان (شرط على عمود التقسيم دائمًا).
* **Materialized Views** للمقاييس الشائعة اليومية.
* **Storage**: استخدم **Parquet** / **ORC** عند تحميل ملفات.
* **الحد من الطول**: تجنّب ضمّ واسع على RAW؛ اعمل التحويل في STG/ODS أولًا.
* **حجز Slots** إذا الاستهلاك ثابت، وإلا **On-Demand** مع مراقبة.
* **قيود Query** على Metabase/Looker (timeouts/bytes caps).

# **11) SLI/SLO (مؤشرات وأهداف)**

| **المؤشر** | **SLO** | **القياس** |
| --- | --- | --- |
| حداثة fact\_orders | ≤ 30 دقيقة تأخير | فرق الوقت بين آخر سجل في ODS/RAW |
| نجاح تشغيل Dataform اليومي | ≥ 99% | نسبة نجاح runs خلال 30 يومًا |
| فشل اختبارات الجودة الحرجة | 0 يوم/شهر | Assertions حرجة (PK/Null/Uniqueness) |
| تكلفة الاستعلام/اليوم | ضمن الميزانية | مراقبة BigQuery Billing |
| زمن استعلام BI p95 | ≤ 5 ثوانٍ | Metabase/Looker Query Stats |

# **12) Runbooks (إجراءات جاهزة)**

## **RB-1: تأخر مصدر RAW (مثلاً توقّف GA4 Export)**

1. تنبيه Freshness ⟶ تحقّق من حالة المصدر.
2. إن كان انقطاعًا خارجيًا: علّق جداول تعتمد على المصدر (Skip) مع **علامة حالة** في mart.
3. عند عودة المصدر: نفّذ **Backfill** للفترة المتأخرة فقط (T إلى T−N).

## **RB-2: فشل اختبارات الجودة الحرجة**

1. أوقف ترقية ODS/MART المتأثرة (Break the build).
2. حدّد التغيير (Schema/Nulls/Duplicates)، أصلحه في STG.
3. أعد تشغيل سلسلة التحويل؛ وثّق السبب في سجل الحوادث.

## **RB-3: ارتفاع تكلفة غير متوقعة**

1. حدّد الاستعلامات الأعلى كلفة (BYTES/Query).
2. أضف **Partition Filter Required**، و**Cluster**، أو **MV**.
3. استرجاع تغييرات، وإعلام فريق الـ BI بقيود مؤقتة.

## **RB-4: طلب حذف بيانات (DSAR)**

1. تلقي user\_id/الهُويّات (هاتف/إيميل) ⟶ خريطة الربط.
2. تشغيل **Procedure**: طمس/تجزئة PII في ods\_/bi\_، تمييز السجلات “مجهّلة”.
3. التأكد أن التقارير المجمّعة سليمة (لا تغيّر الأرقام التاريخية).

## **RB-5: تغيّر مخطط مصدر (Column Added/Removed)**

1. التنبيه يحدّد الجدول/العمود.
2. تحديث عقد المصدر + **stg** لاستهلاك/تجاهل العمود الجديد.
3. تحديث الاختبارات، وإعادة ونشر.

# **13) جدول “مالكون + تواتر + خصوصية” (نموذج)**

| **الجدول** | **المالك** | **التواتر** | **PII** | **سياسة الاحتفاظ** |
| --- | --- | --- | --- | --- |
| raw\_mysql\_orders | Backend | 15 دقيقة (CDC) | لا | 365 يوم |
| stg\_orders | Data Team | 30 دقيقة | لا | 730 يوم |
| ods\_customers | Data Team | 60 دقيقة | نعم (مقنّع) | 730 يوم |
| fact\_orders | Data Team | 30 دقيقة | لا | دائم |
| fact\_sessions | Data Team | يومي/ساعي | لا | 24 شهر |

# **14) قائمة تنفيذ (مختصرة)**

**A) التأسيس**

* إنشاء مشاريع GCP منفصلة (dev/stg/prod) + Service Accounts.
* اختيار المنطقة وتوحيدها مع GA4 Export.
* إعداد موصلات RAW (Datastream/Airbyte/Jobs).
* إنشاء datasets: raw\_\*, stg\_\*, ods\_\*, bi\_\*.

**B) Dataform**

* تهيئة المستودع (Git + بيئات).
* بناء هرم الجداول والاعتماديات.
* تمكين **assertions** الحرجة + تنبيهات.
* جداول Incremental (Partition + Merge).

**C) الجودة والأمن**

* عقود بيانات للمصادر + اختبارات مخطط.
* CLS/RLS لأعمدة PII في marts.
* إجراءات DSAR موثّقة ومجربة.

**D) الأداء والتكلفة**

* Partition/Cluster لجميع الجداول الكبيرة.
* MVs للمقاييس اليومية الشائعة.
* مراقبة Billing وإنذارات التجاوز.

**E) التقارير**

* ربط Metabase/Looker بـ bi\_\* فقط.
* تعريف المقاييس الذهبية كـ Views موحّدة.
* دليل استخدام للمحللين (نطاق الاستعلامات/القيود).

## **ملاحظات خاصة بسياقكم**

* ربط **user\_id** من خدمة الهوية إلى GA4/Firebase **بعد التحقق** فقط، وتجنّب إرسال PII.
* أسواق ذات اتصال متذبذب ⇒ اجعل الـ marts حرجة تُبنى **ساعيًّا** مع **Backfill T−3** عند التأخر.
* عند توسع الحمل، فكّر في **Reservations** (Slots) للمجموعات الحرجة (ODS/MART).

لماذا BigQuery؟

# **لماذا BigQuery؟**

* **مستودع بيانات مركزي (DWH)** يجمع بيانات التطبيق وGA4 وعمليات الشحن/الدفع في مكان واحد.
* **تحليلات متقدمة وسريعة** (Retention، Cohorts، LTV، تجزئة العملاء، إسناد تسويقي).
* **تقارير موحّدة** للإدارة والتقنيين (Looker Studio).

# **ما الذي نُحمّله إلى BigQuery؟**

1. **أحداث GA4** (Export أصلي من GA4 → BigQuery يوميًا/لحظيًا).
2. **بيانات تشغيلية من قاعدة التطبيق**:
   * customers, orders, order\_items, payments, shipments, rma, inventory\_ledger, wallet.
3. **جداول مرجعية**: المدن/المناطق، القنوات التسويقية، شركات الشحن.
4. **تكاليف التسويق** (Meta/Google Ads) لتقييم CAC/ROAS.
5. **خرائط الإسناد**: first\_source/last\_source per user/order.

# **كيف نعمل عليها (خريطة تنفيذ سريعة)**

1. **إعداد المشروع والمساحات**
   * Project: zahraah-analytics
   * Datasets:
     + raw\_ga4 (تصدير GA4)
     + raw\_app (تفريغ جداول التطبيق)
     + stg (تجهيز/تنظيف)
     + mart (جداول نهائية للتقارير)
2. **الاستخراج والتحميل (EL / ELT)**
   * ربط **GA4 → BigQuery** من إعدادات GA4.
   * سحب جداول التطبيق يوميًا بواسطة **Airbyte / Fivetran** (أو Cron + bq load/COPY).
   * الحقول الحساسة (PII) تُجهّز مموّهة/مشفّرة قبل الوصول إلى marts.
3. **النمذجة (Modeling) — هيكل نجمي**
   * **Facts**: f\_orders, f\_payments, f\_shipments, f\_rma, f\_inventory\_moves, f\_events (من GA4).
   * **Dims**: d\_customers, d\_products, d\_date, d\_city, d\_channel.
   * حافظ على مفاتيح ربط موحّدة: customer\_id, order\_id, transaction\_id, user\_pseudo\_id↔customer\_id.
4. **التحويل (T) وأتمتة SQL**
   * استخدم **dbt** على BigQuery لبناء stg → mart (نسخ قابلة للتتبّع، اختبارات جودة).
   * أنشئ **Materialized Views** أو جداول مجدولة لأكثر التقارير استخدامًا (cohorts, LTV, funnel).
5. **التقارير والـ BI**
   * اربط mart مع **Looker Studio** لداشبود:
     + الإدارة: المبيعات، هامش الربح، ROAS.
     + التسويق: First/Last touch، CAC، Funnel.
     + العمليات: نجاح الشحن، RTO، زمن التسليم.
     + المنتج: أفضل الموديلات، دوران المخزون، Out-of-stock.

# **جداول “mart” المقترحة (مختصر)**

* mart.f\_cohorts\_activity: احتفاظ أسبوعي/شهري (W1/W4, M1/M3).
* mart.f\_ltv\_by\_month: LTV حسب شهر الانضمام/القناة.
* mart.f\_cart\_recovery: معدل استرجاع السلة بعد رسائل التذكير.
* mart.f\_cod\_score: خسائر/نِسب فشل COD ومؤشر المخاطرة.
* mart.f\_attribution: first\_vs\_last touch لكل طلب/عميلة.
* mart.f\_inventory\_turnover: دوران المخزون، أيام التغطية.

# **أفضل الممارسات (BigQuery)**

* **Partition** بالجداول الزمنية (على event\_date أو order\_date) لتقليل التكلفة.
* **Cluster** على مفاتيح الاستخدام (مثل customer\_id, source, city).
* استخدم **Scheduled Queries** لتحديث marts يوميًا/ساعيًا.
* فعّل **Cost Controls**: Quotas، تنبيهات إنفاق، الحذر من SELECT \*.
* **Schema Registry** بسيط: وثّق كل جدول/حقل وأصل بياناته.

# **مؤشرات أساسية تتابعها**

* **Revenue, AOV, Conversion Rate, CAC, ROAS**.
* **Retention (W1/W4, M1/M3)**، **Reactivation**، **Repeat Rate**.
* **LTV** حسب القناة والمدينة.
* **Cart Recovery Rate**، **Checkout Drop-off**.
* **COD Fail/RTO Rate**، وزمن التسليم.
* **Inventory Turnover**, **Days of Supply**.

# **تقدير تكلفة مبدئي (إرشادي)**

* التخزين: ~$0.02/GB/شهر.
* الاستعلام: **on-demand** ~$5 لكل 1 تيرابايت مفحوص (مع التقسيم/التجميع الجيد، التكلفة منخفضة).
* يمكنك تثبيت سقف إنفاق وتنبيهات يومية.

# **خطة بدء**

1. ربط **GA4→BigQuery** وتفعيل Export.
2. إعداد موصل **Airbyte** لجداول التطبيق (customers, orders, items, payments…).
3. إنشاء Datasets (raw, stg, mart) وقوالب dbt.
4. بناء نماذج stg + ثلاث جداول mart أولية: f\_orders, f\_cohorts\_activity, f\_attribution.
5. لوحة **Looker Studio** أولى: Revenue/AOV/CAC/Retention.

BigQuery

**BigQuery** هو مستودع بيانات سحابي (Cloud Data Warehouse) من Google يعمل **بدون إدارة خوادم** (Serverless) ومصمّم لتحليل كميات ضخمة جدًا من البيانات بسرعة باستخدام **SQL**.

## **الفكرة ببساطة**

* تتخيّله مكانًا مركزيًا تجمع فيه بياناتك من **GA4** و**Firebase** و**Odoo** والـAPI والسيرفر… ثم تسأل أسئلة تحليلية عليها بـSQL وتطلع لوحات تقارير.
* التخزين عنده **عمودي** وسريع، والتنفيذ **متوازي** على نطاق كبير—مناسب للتقارير والتحليلات، **وليس** لعمليات التطبيق اللحظية مثل إنشاء الطلب مباشرة.

## **لماذا يفيدك**

* **مصدر واحد للحقيقة (SSOT):** توحيد بيانات الحملات (Meta/Ads) مع الطلبات من Odoo وسلوك المستخدم من GA4/Firebase.
* **قرارات تسويق أدق:** حساب ROAS، LTV، Cohorts، نسبة تكرار الشراء، أثر نفاد المخزون على التحويل… إلخ.
* **سرعة ومرونة:** استعلامات على ملايين/مليارات السجلات خلال ثوانٍ–دقائق دون إدارة بنية تحتية.
* **تكلفة تحت السيطرة:** تدفع على قدر البيانات المخزّنة والمقروءة بالاستعلام—وتقلل الكلفة بـ **Partitioning & Clustering**.

## **كيف نستخدمه (خطوات عملية)**

1. **ربط GA4 → BigQuery** (تصدير تلقائي يومي/آني لأحداث التطبيق/الموقع).
2. **ربط Firebase** (Crash/Perf/RC وتجارب A/B إن احتجت).
3. **استيراد Odoo** (طلبات، عناصر طلب، مخزون، أسعار) عبر Job مجدول.
4. تنظيم البيانات في طبقات:
   * **raw** (خام كما هي)،
   * **staging** (تنظيف وتوحيد الحقول)،
   * **marts** (جداول تقارير نهائية: مبيعات يومية، قنوات، منتجات، محافظات…).
5. **لوحات وتقارير** على Looker Studio/لوكر: ROAS، Funnel، Cohorts، مخزون×تحويل.
6. **صلاحيات وأمن:** IAM، سياسات صف/عمود، وعدم تضمين أي PII غير لازمة.

## **أسئلة يجاوبها BigQuery فورًا**

* أي حملات/إعلانات جابت **طلبات مدفوعة** أعلى في صنعاء مقارنةً ببقيّة المحافظات؟
* **LTV** لعملاء اكتسبناهم من Meta مقابل Google Ads خلال 90 يومًا؟
* ما أثر **نفاد المخزون** لمنتجات معيّنة على هروب السلة ذلك الأسبوع؟
* أفضل 50 منتجًا **هامشًا** و**دورانًا** حسب الموسم/الفئة؟

## **نقاط تقنية مختصرة (مهمّة لفريقك)**

* وحدات البناء: **Dataset** (حاوية جداول)، **Table**، **View**، **Materialized View**.
* **Partitioning** غالبًا بالتاريخ (event\_date)، و**Clustering** حسب مفاتيح مثل (user\_id, product\_id) لتقليل البيانات المقروءة.
* **Scheduled Queries** لتحديث الجداول يوميًا، و**Streaming** لإدخال بيانات فورية لو لزم.
* يدعم **Federated queries** (قراءة من Google Sheets/Cloud Storage) عند الحاجة السريعة.

## **ماذا ليس BigQuery**

* ليس قاعدة معاملات تشغيلية (OLTP) ولا بديلًا لقاعدة بيانات التطبيق أو Odoo.
* ليس مناسبًا للاستعلامات اللحظية ضمن مسار الشراء؛ هو مخصّص **للتحليلات** والتقارير المتقدمة.

Airbyte + dbt

# **1) الهدف من الحزمة**

* **Airbyte (ELT):** سحب البيانات الخام من جميع المصادر إلى BigQuery كما هي (Raw).
* **dbt (Transform):** تحويل الخام إلى نماذج منظّمة (Facts/Dimensions)، مع اختبارات جودة ونسخ تاريخية.
* **Looker Studio (BI):** لوحات تحليليّة تنفيذية وتسويقية وتشغيلية تُقرأ من جداول dbt “المارت”.

# **2) مخطط التدفق (Flow)**

مصادر البيانات (Firebase/GA4, MMP, Meta/Google Ads, MySQL API/DB, sGTM logs, CRM)

└── Airbyte (Extract & Load) ──▶ BigQuery (raw\_\* datasets)

└── dbt (stg\_ → int\_ → fct\_/dim\_ في analytics)

└── Looker Studio Dashboards

# **3) Airbyte — ما الذي سنسحبه وكيف نُعدّه؟**

## **3.1 أهم المصادر**

* **تطبيق/سلوك:** Firebase/GA4 (Events, Users, App Info).
* **الإسناد الإعلاني (إن وُجد MMP):** AppsFlyer/Adjust (Installs, Re-attribution, Uninstalls).
* **القنوات الإعلانية:**
  + Meta Ads (Campaigns, Ad Sets, Ads, Spend, Impressions, Clicks, Conversions).
  + Google Ads (نفس المبدأ).
* **الخلفية/المعاملات:** قاعدة بيانات التطبيق/الـ API (Orders, Order Items, Users, Products, Inventory, Refunds, Coupons، COD status).
* **السيرفر/القياس:** سجلات sGTM/Cloud Run (اختياري)، و/أو Pub/Sub إن كنتم تدفعون الأحداث هناك.
* **CRM (إن تم استخدامه لاحقًا):** MoEngage/CleverTap/Braze (Campaigns, Sends, Opens, CTR).

## **3.2 إعداد Airbyte (أفضل الممارسات)**

* **وجهة واحدة:** BigQuery.
* **مخطط المستودع:** أنشئ Dataset لكل مصدر بصيغة raw\_<source> (مثال: raw\_ga4, raw\_meta, raw\_app).
* **التقسيم الزمني:** جداول **Partitioned** على عمود ingested\_at أو وقت الحدث (إن توفر)، مع **Clustering** على مفاتيح الاستعلام (مثل event\_name, campaign\_id, order\_id).
* **الجدولة:**
  + سلوك التطبيق & المعاملات: كل 15–30 دقيقة.
  + القنوات الإعلانية: كل 1–3 ساعات.
* **التطبيع (Normalization):** **عطّله في Airbyte** واعتمد التحويل في **dbt** (يوحّد المنطق والمصدر).
* **الوثوق والجودة:** فعّل **Incremental/CDC** حيثما أمكن، واستخدم **Primary Keys** الصحيحة (مثل transaction\_id, order\_id).
* **إدارة الأسرار:** عبر GCP Secret Manager (مفاتيح API والرموز).

# **4) BigQuery — هيكلة البيانات**

## **4.1 طبقات الـ Datasets المقترحة**

* raw\_\* : خام من Airbyte (كما هي).
* staging : نسخة منظّفة من الخام (أسماء أعمدة موحّدة، Types صحيحة، تفجير JSON).
* intermediate : نماذج وسيطة (ربط مصادر، قواعد الأعمال).
* analytics : جداول **المارت** (Facts/Dimensions) تُستهلك في Looker Studio.

## **4.2 جداول المارت الأساسية (analytics)**

**Facts (حقائق):**

* fct\_orders: أمر/طلب لكل صف.
  + أعمدة رئيسية: order\_id, user\_id, order\_datetime, order\_status, payment\_method, cod\_status, subtotal, discount\_amount, shipping\_fee, tax, total, currency, source\_medium\_campaign, first\_touch\_channel, last\_touch\_channel.
* fct\_order\_items: بند لكل صف.
  + order\_item\_id, order\_id, product\_id, sku, category, brand, unit\_price, qty, line\_total, discount, coupon\_code.
* fct\_events: أحداث GA4/التطبيق.
  + event\_id, event\_name, event\_datetime, user\_id, session\_id, device, os, app\_version, screen, item\_id, cart\_value, referrer, campaign, event\_params (flattened).
* fct\_marketing\_spend\_daily: إنفاق يومي لكل قناة/حملة.
  + date, platform, campaign\_id/name, adset, spend, impressions, clicks, installs, reporting\_timezone.
* fct\_attribution\_day (اختياري متقدم): ربط أوامر اليوم بقنوات (First/Last/Rules).

**Dimensions (أبعاد):**

* dim\_users (SCD2): user\_id, خصائص العميل (الجهاز المفضّل، المدينة إن وُجدت، تاريخ الإنشاء، شرائح VIP/عائدات، موافقات).
* dim\_products (SCD2): product\_id, sku, name, brand, categories (1..n), cost, price, attributes, is\_active.
* dim\_date، dim\_device، dim\_channel (موحّدة للاستخدام في التحليلات).

**SCD2** مهم لتتبّع تغييرات السعر/الفئات/سمات العميل عبر الزمن.

## **4.3 مفاتيح الزمن والمنطقة**

* **التخزين دائماً بـ UTC**، والعرض في Looker Studio بمنطقة **Asia/Aden (+03:00)**.
* أعمدة زمنية موحّدة: \*\_datetime (UTC) + date (مشتقة).

# **5) dbt — التحويلات والجودة**

## **5.1 هيكل مشروع dbt**

* **المجلدات:** models/staging, models/intermediate, models/marts (داخلها marts/marketing, marts/product, marts/finance).
* **التسمية:**
  + stg\_<source>\_\_<table> (نقّي/وحّد أعمدة الخام).
  + int\_<domain>\_\_<logic> (روابط/قواعد أعمال).
  + fct\_\* و dim\_\* في marts.
* **المادة (Materialization):**
  + incremental للجداول الكبيرة (events, orders, spend).
  + table للثابتة (dimensions الصغيرة).
  + استخدم unique\_key (مثل event\_id، order\_id+item\_id) مع منطق **upsert**.

## **5.2 اختبارات dbt (ضرورية)**

* **schema.yml**:
  + unique + not\_null للمفاتيح.
  + accepted\_values لحالات order\_status, cod\_status.
  + **relationships** للتأكد من أن fct\_order\_items.order\_id موجود في fct\_orders.
* **freshness** لمصادر raw\_\* لضمان وصول البيانات خلال نافذة زمنية مقبولة.
* **snapshots** لـ dim\_users, dim\_products (SCD2).

## **5.3 قواعد أعمال (نماذج مهمّة)**

* **إسناد القنوات (قواعد بسيطة عملية):**
  + last\_click\_7d و/أو first\_click\_30d باستخدام utm\_\* + (إن وُجد) معرّفات MMP.
  + تخزين النتيجة في fct\_attribution\_day وربطها بـ fct\_orders لحساب **ROAS/CAC**.
* **الـ Funnel:** من fct\_events (view\_item → add\_to\_cart → begin\_checkout → purchase) لكل قناة/فئة/جهاز.
* **الاحتفاظ (Retention):** تكوين Cohorts أسبوعية/شهرية في جداول محسوبة.
* **COD KPIs:** إنشاء حقول: cod\_delivery\_rate, cod\_return\_rate, cod\_fake\_order\_rate على مستوى اليوم/المدينة/القناة.

# **6) Looker Studio — لوحات وتقارير**

## **6.1 لوحات رئيسية مقترحة**

1. **لوحة المدير التنفيذي (CEO):**
   * GMV, Net Revenue, Orders, AOV, Conversion Rate, New vs Returning, Retention, Inventory Turns.
2. **لوحة التسويق:**
   * Spend, Impressions, Clicks, Installs, CRs, ROAS, CAC، تفصيل بالقناة/الحملة/القطعة الإبداعية.
3. **لوحة المنتج/الفئات:**
   * Top Sellers/Slow Movers، هوامش الربح، Elasticity (سعر ↔ طلب)، أداء الصور/الوصف إن توفر.
4. **لوحة القُمع (Funnel):**
   * view\_item → add\_to\_cart → begin\_checkout → purchase حسب القناة/الجهاز/المدينة.
5. **لوحة COD/التشغيل:**
   * lead time، نسبة تسليم COD، أسباب الإرجاع، مزوّد الشحن، المدن، فرق خدمة العملاء.

## **6.2 أفضل ممارسات الربط**

* اربط Looker Studio مباشرةً على **جداول analytics** (وليس raw\_\*).
* أضف **Parameters** لتغيير النطاق الزمني، الجهاز، القناة، المدينة.
* طبّق **Filters** على مستوى المصدر لتقليل الكلفة (اختيار الجداول المجمّعة عند الإمكان).

# **7) الحوكمة، الأمان، والتكاليف**

* **صلاحيات أدق:**
  + فريق التسويق: **قراءة** analytics فقط.
  + فريق البيانات: قراءة/كتابة على staging, intermediate.
  + Airbyte: كتابة على raw\_\* فقط.
* **المفاتيح/الأسرار:** Secret Manager.
* **ضبط الكلفة في BigQuery:**
  + التقسيم/العنقدة، جداول ملخّصة يومية، حظر SELECT \* في Looker Studio، و**Caching** للتقارير.
* **المراقبة:**
  + Airbyte job alerts، dbt run/test نتائج، تحذيرات freshness، وأخطاء BigQuery.
* **التوقيت:** خزّن UTC واعرض +03:00 في التقارير.

# **8) خطة تنفيذ (مبدئية)**

**-** إنشاء مشروع GCP، BigQuery datasets (raw\_\*, staging, intermediate, analytics) + أذونات.  
**-** تثبيت Airbyte، توصيل مصادر (GA4/Firebase, MySQL/API, Meta/Google Ads, MMP) → اختبارات سحب.  
**-** هيكلة dbt (Repo + بيئة) + نماذج stg\_\* + اختبارات schema الأساسية.  
**-** نماذج int\_\* (ربط spend مع الأحداث/الطلبات) + fct\_orders, fct\_order\_items, dim\_products.  
**-** نماذج التسويق (fct\_marketing\_spend\_daily, fct\_attribution\_day) + الـ Funnel + الاحتفاظ.  
**-** Snapshots (SCD2) + Freshness + تغطية الاختبارات + جدول ملخّص يومي.  
**-** إعداد Looker Studio (لوحات 1–3) + مراجعة الأداء والتكلفة + تسليم دليل التشغيل.

# **9) قائمة تحقق سريعة (قبل الإطلاق)**

* Airbyte يسحب كل المصادر إلى raw\_\* بدون أخطاء وبجدولة ثابتة.
* dbt يُشغّل بنجاح (stg → int → marts) مع **>95%** من الاختبارات خضراء.
* fct\_orders وfct\_marketing\_spend\_daily متّسقان زمنيًا (لا فجوات تواريخ).
* تعريفات مقاييس متّفق عليها (GMV, Net, AOV, ROAS, CAC, Retention).
* لوحات Looker Studio تعكس أرقام “مصدر الحقيقة” (تسويات مع النظام التشغيلي).
* أذونات BigQuery وGCP مضبوطة (مبدأ أقل صلاحية).
* تنبيهات أعطال Airbyte/dbt مفعّلة.

ربط Meta Pixel مع BigQuery

# **ربط Meta Pixel مع BigQuery (تجميع أحداث البيكسل Hit-level)**

تنبيه مهم: **Meta لا توفّر تصديرًا مباشرًا لأحداث البيكسل إلى BigQuery**. لديك طريقان عمليان:

1. **تجميعها بنفسك** عبر **sGTM/خادم وسيط** ثم إعادة إرسالها إلى Meta (CAPI) *و* إلى BigQuery.
2. سحب **بيانات الحملات/التكلفة من واجهة إعلانات Meta** (API/Connector) إلى BigQuery — *هذه بيانات تقارير وليس Hit-level للبيكسل*.

## **الخيار الموصى به (خادم وسيط عبر sGTM)**

**الهدف:** كل حدث (view\_content, add\_to\_cart, initiate\_checkout, purchase…) الذي يُرسل للبيكسل، نُرسله أيضًا لخادمنا (sGTM) ليقوم بأمرين:

1. يرسل الحدث إلى **Meta (CAPI)** مع **event\_id** نفسه (لـ Dedup بين client/server).
2. يُسجّل نسخة قياسية إلى **BigQuery** (Raw) عبر Pub/Sub أو إدراج مباشر.

### **التصميم باختصار**

Web/App ── Pixel/SDK ──▶ sGTM (sgtm.yourdomain)

├─▶ Meta CAPI (باستخدام event\_id نفسه + user\_data المُهشّ)

└─▶ BigQuery (raw\_meta\_events) عبر Pub/Sub/Streaming

### **ما الذي نخزّنه في BigQuery؟**

* **الحد الأدنى اللازم** (التزامًا بالسياسات والخصوصية):
  + event\_name, event\_id, event\_timestamp, source (web/server)
  + content\_ids / items[] (إن وُجدت)، value, currency
  + page\_url/referrer (للويب)، أو screen\_name (للتطبيق) عند الحاجة فقط
  + fbp/fbc إن متوفّرة (تحسين المطابقة لدى Meta)
  + **user\_data مُهشّ (SHA-256)** للبريد/الهاتف عند توفره + external\_id (user\_id)، **لا نخزن PII خام**
  + **consent flags** (analytics/ads) من الـ CMP
* **تقسيم الجداول**: dataset raw\_meta\_events، جدول partitioned يوميًا + clustering على event\_name, event\_id.

### **الجودة والخصوصية**

* **Dedup**: نستخدم event\_id نفسه في الإرسال المزدوج (client & server)، ونحتفظ بعلامة is\_deduped في التحويلات.
* **CMP/Consent**: لا نُسجل أي حقول تسويقية عند الرفض (أو نخزّنها بحالة “anonymized”)؛ الالتزام بـ Consent Mode.
* **سياسات Meta**: لا تجمع فئات حساسة/صحية/مالية… إلخ؛ احتفظ بفترة معقولة (مثلاً 6–12 شهر لأحداث التسويق التفصيلية).

### **كيفية الهبوط في مسار التحويل (Dataform)**

* **raw**: raw\_meta\_events.\* (كما وردت من sGTM).
* **staging**: تنظيف أسماء الحقول، تطبيع القيم، توحيد timezone (UTC).
* **intermediate**: ربط مع GA4/MMP عبر event\_id/transaction\_id/external\_id إن توفر.
* **marts**:
  + fct\_meta\_events: عدد الأحداث حسب اليوم/الاسم/القناة/الجهاز.
  + fct\_meta\_purchases: مشتريات المنشؤة عبر Meta مع قيمتها (حسب الربط).
  + **مؤشرات المطابقة**: Match Quality, Dedup Rate، فرق العدّ بين BigQuery وEvents Manager.
* **الفائدة**:
  + تحليل أدق لمسار المستخدم خارج حدود تقارير Ads Manager.
  + استرداد الحقيقة التحليلية داخليًا (hit-level)، وربطها بباقي القنوات/الأحداث/الطلبات.

## **بدائل مختصرة (بدون sGTM)**

* **Endpoint مخصص (Cloud Run/Function)**: تزرع سكربت JS يرسل الحدث إلى endpointك، وهو يبعث إلى Meta CAPI + BigQuery. نفس المبادئ (event\_id, hashing, consent).
* **Connectors (تقارير فقط)**: Airbyte/Fivetran/Supermetrics تسحب **تقارير الإعلانات** (حملات/إنفاق/نقرات/تحويلات مجمّعة)، وليست سجلات البيكسل التفصيلية. تفيد لحساب **ROAS/CAC** لكن لا تُغني عن Hit-level.

## **نضيف على المخطط !**

1. **Metabase ↔ BigQuery:**
   * ربط Metabase على **dataset analytics فقط**.
   * إنشاء **Dashboards**: CEO / Marketing / Product.
   * تفعيل الكاش والتنبيهات، وضبط التوقيت (+03:00).
2. **Meta Pixel ↔ BigQuery:**
   * تفعيل **sGTM** (أو endpoint بديل) للإرسال المزدوج: **CAPI + BigQuery**.
   * إنشاء **raw\_meta\_events**، ثم تحويلات في Dataform لإخراج fct\_meta\_events وربطها بجداول GA4/الطلبات.
   * مراقبة **Dedup** و**Match Quality** مقابل Meta Events Manager.

Google Ads ↔ BigQuery

**Google Ads ↔ BigQuery**

## **الملخص (آلية العمل + الارتباطات)**

* **لماذا؟** تجميع **تكاليف وأداء Google Ads** في BigQuery لحساب ROAS/CAC وربطها بمشتريات التطبيق، ثم **رفع تحويلات Offline/COD** لتحسين المزايدة.
* **يعتمد على:** BigQuery + **Data Transfer Service (DTS)** لجلب تقارير Ads يوميًا، و(**اختياري**) Google Ads API للتحديث داخل اليوم، و**Dataform** لبناء مارت التسويق. [Google Cloud+1](https://cloud.google.com/bigquery/docs/dts-introduction?utm_source=chatgpt.com)
* **المخرجات:** جداول خام raw\_google\_ads مجدولة يوميًا + مارت (fct\_marketing\_spend\_daily …) تُقرأ في Metabase/Looker. [Google Cloud+1](https://cloud.google.com/bigquery/docs/google-ads-transfer?utm_source=chatgpt.com)
* **التحويل العكسي (اختياري):** رفع تحويلات Offline/COD إلى Google Ads عبر API أو Data Manager (Enhanced Conversions for Leads). [Google for Developers](https://developers.google.com/google-ads/api/docs/conversions/upload-offline?utm_source=chatgpt.com)[دعم جوجل](https://support.google.com/google-ads/answer/15713840?hl=en&utm_source=chatgpt.com)
* **قيود مهمّة:** موصل DTS **يومي** مع نافذة تحديث يمكن ضبطها (حتى 30 يومًا لكل تشغيل). للتحديثات داخل اليوم استخدم API/Connector. [Google Cloud](https://cloud.google.com/bigquery/docs/google-ads-transfer?utm_source=chatgpt.com)

# **A) Google Ads → BigQuery (DTS)**

### **الهدف**

جلب تقارير Google Ads إلى BigQuery تلقائيًا على **جدولة يومية مُدارة** (بدون كود)، مع إمكانية **استرجاع أيام سابقة (Refresh window)** عند كل تشغيل. [Google Cloud+1](https://cloud.google.com/bigquery/docs/dts-introduction?utm_source=chatgpt.com)

### **ما الذي نحتاجه؟**

* مشروع GCP وDataset وجهة (مثال: raw\_google\_ads).
* صلاحية ربط حساب Google Ads الهدف.
* **حساب خدمة** للوصول (يفضّل بدل الحساب الشخصي). [Google Developer forums](https://discuss.google.dev/t/bigquery-data-transfer-service-for-google-ads/131492?utm_source=chatgpt.com)

### **الإعداد (خطوات عملية مختصرة)**

1. افتح BigQuery **Data transfers** وأنشئ **Transfer** جديدًا لمصدر **Google Ads**.
2. اختر **Customer ID**، اضبط **الوقت اليومي للتشغيل**، وحدّد **Refresh window** (مثل 7–30 يومًا) لتعبئة أي فروقات.
3. احفظ… وتابع أول تشغيل + راقب النتائج في dataset الوجهة. [Google Cloud](https://cloud.google.com/bigquery/docs/google-ads-transfer?utm_source=chatgpt.com)

### **المخرجات المتوقَّعة**

* جداول/تقارير Ads مقسّمة زمنيًا (حملات/Ad groups/Ads/Keywords …).
* استخدم دليل **تحويل تقارير Google Ads** من جوجل لخريطة الحقول وبناء الـ marts فوقها. [Google Cloud](https://cloud.google.com/bigquery/docs/google-ads-transformation?utm_source=chatgpt.com)

### **أفضل الممارسات (تكلفة/حوكمة)**

* اربط أدوات BI على **analytics** (مخرجات Dataform) بدل ضرب raw\_\*.
* فعِّل **Partitioning/Clustering** ووفِّر **جداول ملخّصة يومية** لاستهلاك اللوحات.
* قارن الإنفاق اليومي مع Ads UI، وفعّل تنبيهات فشل الـ DTS. [Google Cloud](https://cloud.google.com/bigquery/docs/google-ads-transfer?utm_source=chatgpt.com)

### **قيود/ملاحظات**

* **التكرار: يومي فقط**؛ لا يوجد تحديث “لحظي” عبر DTS. إن احتجت داخل اليوم، اسحب بالـ **Google Ads API** وأدرِج لـ BigQuery. [Google Cloud](https://cloud.google.com/bigquery/docs/google-ads-transfer?utm_source=chatgpt.com)

# **B) BigQuery → Google Ads (Offline Conversions / COD)**

### **الهدف**

رفع تحويلات **Offline/COD Delivered** إلى Google Ads لتحسين المزايدة ونمذجة التحويلات، أو الترقية إلى **Enhanced Conversions for Leads** للمطابقة الأقوى. [Google for Developers](https://developers.google.com/google-ads/api/docs/conversions/upload-offline?utm_source=chatgpt.com)[دعم جوجل](https://support.google.com/google-ads/answer/15713840?hl=en&utm_source=chatgpt.com)

### **ما الذي نحتاجه؟**

* **Conversion action** مهيّأ في Google Ads (Goals → Conversions → Import).
* **معرّفات مطابقة**:
  + **GCLID/GBRAID/WBRAID** عند النقرات (تأكّد من Auto-tagging). [Google for Developers](https://developers.google.com/google-ads/api/docs/conversions/legacy_oci_guide?utm_source=chatgpt.com)
  + أو **بيانات مستخدم مُهشّ (SHA-256)** للبريد/الهاتف في **Enhanced Conversions for Leads** عبر **Data Manager** أو API. [دعم جوجل](https://support.google.com/google-ads/answer/15713840?hl=en&utm_source=chatgpt.com)

### **الإعداد (مسار موصى به)**

* من Ads UI: إنشاء Conversion → **Import** → اختيار **Data Manager / API** كمصدر للـ offline/enhanced. [دعم جوجل](https://support.google.com/google-ads/answer/13763054?hl=en&utm_source=chatgpt.com)
* مهمة مجدولة (يوميًا) تقرأ من جدول BigQuery (تحويلات COD المؤكدة) وترفعها عبر **Google Ads API**. [Google for Developers](https://developers.google.com/google-ads/api/docs/conversions/upload-offline?utm_source=chatgpt.com)

### **نوافذ زمنية وقيود**

* استيراد التحويلات يجب أن يقع **ضمن نافذة 90 يومًا** من آخر نقرة (Enhanced Leads لبعض الحالات **63 يومًا**). [دعم جوجل](https://support.google.com/google-ads/answer/15081888?hl=en&utm_source=chatgpt.com)

### **جودة/خصوصية**

* أرسل **أقل قدر لازم** من البيانات؛ لا تحفظ PII خام في BigQuery—استخدم **تجزئة** عند اللزوم.
* طابق المنطقة الزمنية وتحقق أن **وقت التحويل بعد وقت النقر**. [Medium](https://medium.com/%40SNeefischer/troubleshooting-and-solutions-while-importing-offline-conversions-using-gclid-b312f07df06c?utm_source=chatgpt.com)

## **مخرجات Dataform (اقتراح سريع للمارت)**

* fct\_marketing\_spend\_daily (من DTS): تاريخ، Campaign/AdGroup/Ad، **Spend/Impr/Clicks/CTR/CPC**. [Google Cloud](https://cloud.google.com/bigquery/docs/google-ads-transformation?utm_source=chatgpt.com)
* ربطها مع fct\_purchases/fct\_attribution لحساب **ROAS/CAC** (Last/First/Rules).
* جدول fct\_offline\_conversions\_uploaded لتتبع ما تم رفعه وما تم قبوله من Ads.

## **قائمة تحقق سريعة (قبل التشغيل)**

* تم إنشاء **DTS** لـ Google Ads مع **Refresh window** مناسب (7–30 يومًا). [Google Cloud](https://cloud.google.com/bigquery/docs/google-ads-transfer?utm_source=chatgpt.com)
* ظهرت جداول raw\_google\_ads وتُحدّث يوميًا بدون أخطاء.
* مارت التسويق جاهز ويغطي ROAS/CAC.
* تفعيل **Conversion action** و**Auto-tagging** + تجهيز مفاتيح المطابقة (GCLID/GBRAID/WBRAID أو بيانات مُهشّة). [Google for Developers](https://developers.google.com/google-ads/api/docs/conversions/legacy_oci_guide?utm_source=chatgpt.com)
* Job يومي يرفع **Offline/COD** ضمن النوافذ الزمنية المسموحة (حتى 90 يومًا، وبعض حالات Enhanced Leads حتى 63 يومًا).

ربط Metabase مع BigQuery (لوحات ذكية بأقل تكلفة)

# **ربط Metabase مع BigQuery**

## **الهدف**

تشغيل لوحات تنفيذية/تسويقية مباشرة من **جداول analytics** في BigQuery (التي نبنيها عبر Dataform) مع حوكمة وصلاحيات واضحة وتكاليف مضبوطة.

## **ما الذي نحتاجه؟**

* **Service Account** مخصص لـ Metabase (مثلاً: metabase-sa@…).
* **أذونات دقيقة**:
  + على مستوى المشروع/الـ dataset:
    - **BigQuery Data Viewer** (قراءة الجداول/Views).
    - **BigQuery Job User** (تشغيل الاستعلامات).
  + الأفضل قصر الوصول على **dataset analytics فقط** (وحجب raw\_\* وstaging).
* **إعدادات Metabase (Admin → Databases → Add Database → BigQuery)**:
  + Project ID + موقع البيانات (US/EU).
  + رفع JSON لمفاتيح الحساب الخدمي.
  + **Timezone للعرض:** اضبط Metabase على **Asia/Aden (+03:00)** (البيانات مخزنة UTC).
  + **Sync & Scan**:
    - مرّة يوميًا لمزامنة الميتاداتا.
    - تقليل “Fingerprinting” على الجداول الضخمة (يكفي على جداول المارت/الملخص).
  + **Caching**: فعّل الكاش (مثلاً 15–60 دقيقة) للوحات كثيرة الاستخدام.
* **حوكمة البيانات داخل Metabase**:
  + مجموعات مستخدمين (Exec/Marketing/Product) مع صلاحيات “Read” على جداول محددة.
  + إخفاء الجداول “الحساسة” أو الحقول (PII) من الواجهة.
  + إن احتجت **Row-Level Security**: استخدم **BigQuery Row Access Policies** أو **Authorized Views** وقدّمها لمتبايز كجداول للقراءة.

## **أفضل الممارسات لتقليل التكلفة والتعقيد**

* اربط Metabase فقط على **analytics** (جداول خلاصة) بدل ضرب جداول الأحداث الخام.
* وفّر **جداول ملخّصة يومية/أسبوعية** (counts, sums) للوحة الـ CEO والتسويق.
* تجنّب SELECT \* في الأسئلة؛ حدّد الأعمدة المطلوبة.
* فعّل تنبيهات (Pulses/Alerts) على مؤشرات رئيسية (انخفاض الشراء، ارتفاع المرتجعات…).

تأسيس مسار بيانات

تأسيس مسار بيانات بسيط يعتمد ربط GA4/Firebase مع BigQuery ثم تحويل البيانات عبر Dataform إلى جداول تحليلية جاهزة للوحات Looker Studio. المحتوى أدناه يشرح “ماذا نفعل ولماذا وكيف نتحقق” خطوة بخطوة، بدون أكواد. المبرمج سيُنفّذ التفاصيل التقنية لاحقًا.

## **1) ما الذي سنبنيه؟ (الصورة الكاملة)**

* المصدر: تطبيق Zahraah يرسل أحداث القياس إلى GA4/Firebase.
* الربط: تفعيل تصدير GA4/Firebase إلى BigQuery ضمن مشروع GCP الخاص بكم.
* التحويل: استخدام Dataform لتنظيف وتوحيد الجداول الخام وإنتاج جداول تحليلية (marts) واضحة.
* الاستهلاك: ربط Looker Studio مباشرةً بجداول marts لعرض تقارير تنفيذية وتسويقية وتشغيلية.

لماذا هذا المسار؟ لأنه الأقل تعقيدًا والأسرع للإطلاق، ويكفي لبدء التحليلات اليومية بحوكمة جيدة وتكلفة مضبوطة.

## **2) ربط GA4/Firebase بــ BigQuery**

1. في حساب Firebase/GA4: فعِّل BigQuery Linking على مشروع GCP نفسه الذي سيضم BigQuery.
2. فعِّل نوعي التصدير: اليومي (Daily) + اللحظي داخل اليوم (Streaming/Intraday) لضمان حداثة البيانات.
3. سمِّ مجلد البيانات (Dataset) باسم واضح مثل: ga4\_raw داخل مشروعكم (مثال: zahraah-analytics).
4. تأكد من الأذونات: حساب الخدمة الذي سيُشغّل Dataform يملك صلاحيات قراءة/كتابة مناسبة على BigQuery.
5. توقّع ظهور جداول يوميّة للأحداث، وأخرى لحظية لليوم الجاري. سيتم دمجها لاحقًا في مرحلة التحويل.

نصيحة تشغيلية: ابدأ بمراقبة عدد الأحداث اليومي في BigQuery مقابل واجهة GA4؛ فروق 5–10% خلال اليوم تعتبر طبيعية بسبب التدفّق.

## **3) تنظيم BigQuery (بدون تفاصيل تقنية دقيقة)**

* أنشئ Datasets طبقية حسب الغرض:
  + raw\_\* للاحتفاظ بالبيانات الخام القادمة من GA4 (مثل: ga4\_raw).
  + staging للتنظيف والتوحيد الأولي للأعمدة والقيم.
  + intermediate للمنطق الوسيط (جلسات، ربط مصادر، قواعد أعمال).
  + analytics لمخرجات التحليل النهائية (marts) التي ستُستهلك في التقارير.
* سياسة الزمن: التخزين في UTC دائمًا. العرض في التقارير يكون على المنطقة Asia/Aden (+03:00).
* الحفظ والتكلفة: اتّفقوا على فترة احتفاظ مناسبة بالخام (مثل 12–24 شهرًا) حسب التكلفة والاستخدام.

## **4) مشروع Dataform (ما الذي يفعله؟)**

* Dataform يدير تعريف الجداول التحويلية داخل BigQuery كتعريفات قابلة للتشغيل والجدولة.
* تقسيم منطقي للمجلدات (مبدأ):
  + staging: تجهيز جدول أحداث موحّد وحديث، تنظيف أسماء الأعمدة وأنواعها، إزالة التكرارات.
  + intermediate: حسابات وسيطة مثل الجلسات، توحيد مصادر الحركة، معرّفات المستخدم.
  + marts: جداول حقائق وأبعاد جاهزة للتقارير (التي يستخدمها التسويق والإدارة).
  + assertions: اختبارات جودة تُشغَّل تلقائيًا وتُوقف النشر عند وجود بيانات خاطئة.
* التسمية المقترحة: stg\_\* للتهيئة، int\_\* للوسيط، fct\_\* للحقائق، dim\_\* للأبعاد.

## **5) الجداول التحليلية الأساسية التي سننتجها (marts)**

1. fct\_purchases — “ملخّص عملية الشراء”
   * كل صف يمثل عملية شراء واحدة.
   * الحقول المنطقية: تاريخ ووقت الشراء (UTC + حقل عرض محلي)، معرّف المستخدم المجهول داخل GA4 (user\_pseudo\_id)، المبلغ الإجمالي، العملة، رقم العملية (transaction\_id) إن وُجد.
   * فائدة: مؤشّرات الإيراد وعدد المشتريات اليومية، والربط لاحقًا مع نظام الطلبات التشغيلي.
2. fct\_purchase\_items — “بنود الشراء”
   * كل صف يمثل بندًا داخل عملية شراء.
   * الحقول المنطقية: transaction\_id، معرّف المنتج (إن وُجد في items)، الاسم/الفئة/العلامة، السعر، الكمية، الخصم، الكوبون.
   * فائدة: تحليل سلة المشتريات، البائعين الأكثر مبيعًا، المنتجات الضعيفة.
3. fct\_events\_daily — “تجميع سلوكي يومي”
   * كل صف يمثل يومًا واسم حدث.
   * الحقول المنطقية: عدد الأحداث، عدد المستخدمين اليوميين (DAU)، عدادات لأحداث رئيسية (إضافة للسلة، البدء بالدفع، شراء).
   * فائدة: متابعة النشاط اليومي وتغيّر السلوك بسرعة.
4. fct\_user\_daily — “نشاط المستخدم اليومي”
   * كل صف يمثل مستخدمًا في يوم معيّن.
   * الحقول المنطقية: عدد الأحداث، عدد المشاهدات، إضافات للسلة، مشتريات.
   * فائدة: تحليل احتفاظ المستخدمين وشرائح النشاط.
5. fct\_funnel\_steps — “قُمع مبسّط”
   * كل صف يمثل يومًا مع أعداد المستخدمين الذين وصلوا لكل مرحلة من القمع: مشاهدة منتج → إضافة للسلة → بدء الدفع → شراء.
   * فائدة: اكتشاف أماكن التسرب بسرعة وتحسين تجربة الشراء.

## **6) ما الذي نحتاج تتبّعه في GA4/Firebase؟ (قائمة تشغيل)**

* أحداث المنتجات/السلة: view\_item، view\_item\_list، select\_item، add\_to\_cart، view\_cart، begin\_checkout، add\_shipping\_info، add\_payment\_info، purchase، refund.
* أحداث مساندة: login، sign\_up، search، apply\_coupon، remove\_from\_cart.
* عناصر المصفوفة items لكل حدث متعلق بمنتج: item\_id، item\_name، item\_brand، item\_category، price، quantity، discount، coupon.
* أمور حرجة للنجاح:
  + كل حدث شراء يجب أن يتضمن transaction\_id إن أمكن.
  + تمرير معلومات العملة والمبلغ بشكل متّسق.
  + استخدام user\_id عند تسجيل الدخول (متى توفر) إلى جانب user\_pseudo\_id للربط لاحقًا مع بيانات الطلبات.

## **7) الزمن والمناطق (لتوافق الأرقام)**

* نخزّن دائمًا بالـ UTC لضمان توحيد المصادر.
* نوفّر أعمدة عرض محلية (+03:00) للتقارير التنفيذية.
* نتفق مسبقًا على المنطقة الزمنية الافتراضية لكل لوحة في Looker Studio لتجنّب الالتباس.

## **8) اختبارات الجودة (Assertions) التي يجب تفعيلها**

* فحوص المفاتيح الأساسية: لا يوجد transaction\_id مكرّر في fct\_purchases.
* فحوص القيم: لا توجد مبالغ سالبة في ملخصات الشراء.
* فحوص الاكتمال: وصول حدّ أدنى متّفق عليه من الأحداث يوميًا (قيمة قابلة للتعديل حسب نمو الاستخدام).
* فحوص العلاقات: كل بند في fct\_purchase\_items يعود إلى عملية شراء موجودة في fct\_purchases.
* فحوص حداثة البيانات: جداول staging تُحدَّث خلال نافذة زمنية متفق عليها (مثل آخر 60 دقيقة).

## **9) الجدولة وترتيب التشغيل (بدون تفاصيل تقنية)**

* تهيئة staging: تُحدّث كل 30 دقيقة وتشمل نافذة الأيام الأخيرة (مثل 7 أيام) لمعالجة أي تأخير في وصول البيانات.
* تشغيل intermediate ثم marts: كل ساعة بعد اكتمال تحديث staging.
* تشغيل اختبارات الجودة مع كل تشغيل؛ أي فشل يرسل تنبيهًا للفريق قبل الإظهار في لوحات الإدارة.

## **10) الأذونات والحماية**

* مبدأ أقل صلاحية:
  + فريق البيانات: صلاحيات على staging/intermediate/analytics.
  + فريق التسويق: قراءة فقط على analytics.
  + حساب خدمة Dataform: تنفيذ التحويلات والقراءة من الخام والكتابة على طبقات التحليل.
* إدارة الأسرار (إن احتجناها لاحقًا): عبر Secret Manager في GCP.
* مراقبة الوصول: سجلات BigQuery + تنبيهات عند فشل الجداول أو تدهور التحديثات.

## **11) مراقبة الأداء والتكلفة**

* نتجنّب الاستعلامات واسعة النطاق على جداول الأحداث الخام.
* نبني جداول ملخّصة يومية/أسبوعية لتقارير Looker Studio لتقليل تكلفة الاستعلامات.
* نفعّل التصفية والمعلمات في لوحات Looker Studio (نطاق زمني، قناة، جهاز) بدل تجميع كل التاريخ في كل مرة.
* مراجعة الفواتير شهريًا ووضع حدود تنبيه.

## **12) ربط Looker Studio**

* لوحات مقترحة:
  + لوحة المدير التنفيذي: GMV تقديري من GA4، عدد الطلبات (تقريبي)، معدل التحويل، AOV، مستخدمون جدد/عائدون، احتفاظ.
  + لوحة التسويق: إنفاق (لاحقًا بعد ربط القنوات)، الظهور، النقرات، التحويلات، قُمع السلوك لكل قناة.
  + لوحة المنتج/الفئات: الأعلى مبيعًا، البطيء، الخصومات، الأداء حسب الفئة/السعر/العلامة.
* ملاحظات مهمة:
  + أرقام الإيراد النهائية يجب أن تأتي لاحقًا من نظام الطلبات التشغيلي (مصدر الحقيقة)، بينما GA4 يُستخدم كمؤشّر سريع.

## **13) تنفيذ**

* تفعيل ربط GA4/Firebase مع BigQuery + التحقق من وصول الجداول اليومية واللحظية.
* إعداد طبقة staging في Dataform (تنظيف وتوحيد وإزالة تكرارات) مع اتفاق على أسماء الحقول.
* إعداد طبقة intermediate وقواعد الأعمال الأولية (جلسات، مصادر حركة، حقول مشتقة أساسية).
* إنتاج جداول marts المذكورة + تفعيل اختبارات الجودة.
* إعداد لوحات Looker Studio الأساسية + فحص الأرقام مقابل واجهة GA4 + جلسة اعتماد.
* ربط نظام الطلبات التشغيلي كمصدر الحقيقة للإيراد والمرتجعات، ودمج الجداول مع fct\_purchases.
* إضافة قنوات الإعلان (Meta/Google Ads) لحساب ROAS/CAC.
* تعريف قاموس مقاييس رسمي ونشره على الفرق.
* تجهيز قُمع أكثر تفصيلاً (خطوات الدفع، شحن COD) بعد توفر المصادر.

## **14) مسؤوليات الفرق (RACI مبسّط)**

* المنتج/البيزنس: تحديد المقاييس التعريفية (ما هو GMV؟ ما هو AOV؟) وقبول اللوحات.
* الفريق التقني: تنفيذ الربط والتهيئة وضبط Dataform وتشغيل الجداول.
* فريق البيانات/التحليلات: تصميم النماذج التحليلية، كتابة اختبارات الجودة، مراقبة الصحة.
* التسويق: استخدام اللوحات يوميًا، إبلاغ الفريق بأي فجوات أو فروق غير منطقية.

## **15) شروط القبول قبل الإطلاق**

* ظهور البيانات في BigQuery بشكل ثابت خلال اليوم.
* توافق الأرقام الأساسية بين BigQuery وواجهة GA4 ضمن هامش منطقي.
* مرور جميع اختبارات الجودة بدون أخطاء.
* جاهزية ثلاث لوحات أساسية في Looker Studio وعرضها بالمنطقة الزمنية الصحيحة.

## **16) مخاطر شائعة وكيف نتفاداها**

* نسيان transaction\_id في الشراء: يعيق تحليل الإيراد على مستوى العملية — الحل: توثيق إلزامي لهذا الحقل في التتبع.
* فروق زمنية: خلط UTC مع +03:00 قد يسبب تشوّه يومي — الحل: قاعدة واضحة للتخزين والعرض.
* تضخيم التكلفة: الاستعلام على الخام مباشرةً في اللوحات — الحل: الاعتماد على جداول ملخّصة في marts.
* تأخر وصول البيانات اللحظية: طبيعي أحيانًا — الحل: تصميم اللوحات ليومي/أسبوعي مع مؤشرات “تحديث آخر”.

## 

أساسيات BigQuery

### **أساسيات BigQuery**

### **1) الإدخال (Ingestion)**

* **GA4/Firebase Export** إلى BigQuery (daily + intraday).
* **sGTM/CAPI**: إرسال أحداث التسويق (view\_content, add\_to\_cart, purchase…) إلى Meta CAPI **وبنفس الوقت** تسجيل نسخة خام في BigQuery (Raw Meta Events) مع event\_id لتجنّب الازدواج.
* **نظام الطلبات/المخزون** (قاعدة بيانات التطبيق أو API).
* **تكاليف الإعلانات**: Meta/Google Ads API (تقارير إنفاق/نتائج مجمّعة).
* **CRM/Push/WhatsApp** (إذا وُجد) + ملفات CSV لاستيراد سريع.
* **سياسة المعرّفات**: user\_id (عند تسجيل الدخول) + user\_pseudo\_id + transaction\_id + event\_id.

### **2) تصميم المستودع**

* Datasets: raw\_\* → staging → intermediate → analytics (المارت).
* **تقسيم زمني** على تاريخ الحدث + **Clustering** (مثلًا event\_name, campaign\_id).
* **منطقة زمنية**: التخزين UTC، العرض Asia/Aden (+03:00).
* **تسمية ثابتة**: stg\_, int\_, fct\_, dim\_.

### **3) الحوكمة والأمان**

* IAM بأقل صلاحية:
  + Marketing: قراءة **analytics** فقط.
  + Data/Eng: كتابة على staging/intermediate/analytics.
  + حسابات خدمة منفصلة لـ Dataform/Metabase.
* **Authorized Views / Row-Level Security** عند الحاجة.
* **عدم تخزين PII خام**؛ تخزين بريد/هاتف **مُهشّ** فقط، والالتزام بالموافقات (CMP/Consent).

### **4) الجودة والاختبارات**

* **Freshness**: تحقق وصول البيانات في نافذة زمنية متفق عليها.
* **Uniqueness/Not Null** للمفاتيح (transaction\_id, event\_id…).
* **Relationships** (كل بند شراء يعود لشراء).
* **Dedup** بين client/server باستخدام event\_id.
* **حد أدنى لحجم الأحداث اليومي** وتنبيهات عند الانحرافات.

### **5) التكلفة والأداء**

* اعتمد **Partitions + Clustering**، وجداول **ملخّصة يومية/أسبوعية** للوحات.
* تجنّب SELECT \* في BI، وفعّل **Cache**.
* **Materialized Views** عند التجميعات الشائعة.
* سياسات احتفاظ واضحة للجداول الخام والملخصة.

### **6) التحويل والجدولة**

* **Dataform** (أو dbt) لسير: staging → intermediate → marts.
* **Incremental** للجداول الكبيرة (الأحداث/الطلبات).
* جدولة: staging كل 30 دقيقة، marts كل ساعة، اختبارات الجودة مع كل تشغيل.
* تحكم بالإصدارات + CI خفيف لاختبارات المخطط.

### **7) المراقبة والتنبيه**

* BigQuery Job & Audit Logs + تنبيهات فشل تشغيل.
* مراقبة Airbyte/Connectors (إن استخدمت) وsGTM/Cloud Run.
* لوحة “صحة البيانات” (Freshness، أحجام، معدلات Dedup/Match).

### **8) الاستهلاك (BI)**

* **Metabase** و/أو **Looker Studio** يقرؤون من **analytics** فقط.
* إعداد **Timezone + Caching + صلاحيات المجموعات**.
* Dashboards جاهزة: CEO، التسويق (ROAS/CAC بعد ربط الإنفاق)، المنتج/القمع.

### **9) التسويق والقياس**

* **Meta Pixel ↔ BigQuery** عبر **sGTM/CAPI** مع event\_id موحّد، وتخزين نسخة خام لأحداث البيكسل.
* **تكاليف الإعلانات** إلى BigQuery لحساب ROAS/CAC وربطها بمشتريات GA4/الطلبات.
* **Consent/CMP** تُطبَّق على الإرسال والتخزين.

### **10) التوثيق وRACI**

* قاموس مقاييس (GMV, Net, AOV, ROAS, CAC, Retention…).
* RACI واضح (Product/Marketing/Data/Eng)، وخطوات Runbook للطوارئ.

- شرح وتلخيص لـ BigQuery

**- شرح وتلخيص لـ BigQuery**

BigQuery هو مستودع بيانات سحابي مُدار بالكامل من Google يُصبح “مصدر الحقيقة” لـ زهراء: كل ما تنتجه أنظمتك من أحداث وسجلات ومعاملات يتجمّع فيه، ثم يُنظَّم ويُحوَّل إلى طبقات واضحة تُغذّي التقارير والقرارات. آلية العمل تبدأ من المصادر: تطبيق الموبايل يرسل القياس إلى GA4/Firebase التي تُصدِّر تلقائيًا إلى BigQuery، وقنوات التسويق تُجلب عبر موصلات رسمية مثل Data Transfer Service لحساب Google Ads، ومعها يمكن تمرير أحداث التسويق عبر الخادم sGTM إلى Meta CAPI وفي الوقت نفسه حفظ نسخة خام في BigQuery، كما تُضاف بيانات النظام التشغيلي (الطلبات، البنود، العملاء، المخزون) من قاعدة بيانات التطبيق أو واجهاتها، وأي مصادر أخرى مثل CRM أو ملفات CSV. كل ذلك يهبط أولًا كبيانات خام، ثم تُنظَّف وتُوحَّد داخل BigQuery نفسه باستخدام أداة تحويل مثل Dataform لتكوين طبقات منطقية: “تهيئة” لتنظيف الحقول، “وسيطة” لتطبيق قواعد الأعمال والربط بين المصادر، و“تحليلية” نهائية (marts) جاهزة للاستهلاك. التخزين يكون دائمًا بتوقيت UTC لضمان الاتساق بين المصادر، بينما تُعرض المؤشرات في اللوحات على منطقتك الزمنية Asia/Aden، وتُقسَّم الجداول زمنيًا وتُعنقد على مفاتيح شائعة لتقليل التكلفة وتسريع الاستعلام. في الطرف الآخر تُستهلك المخرجات مباشرة عبر أدوات ذكاء الأعمال مثل Metabase أو Looker Studio من طبقة analytics فقط، ويمكن أيضًا تشغيل نماذج تنبؤية بسيطة عبر BigQuery ML أو إرجاع تحويلات “Offline/COD Delivered” إلى Google Ads لتحسين العطاءات، وكل ذلك تحت حوكمة صلاحيات دقيقة، وتجزيء الوصول، وتشفير أو تجزئة أي معرّفات شخصية مع احترام الموافقات القادمة من الـ CMP.

لإقامتها تحتاج مشروع GCP مُفعّل عليه الفوترة وخدمة BigQuery، وتحديد موقع البيانات (US/EU) ودِساتير واضحة (raw، staging، intermediate، analytics)، وحسابات خدمة مع صلاحيات أقل قدر ممكن لكل طرف: Dataform للتشغيل والتحويل، GA4/Firebase للربط، DTS لجلب تقارير الإعلانات، وأداة الـ BI للقراءة فقط. تحتاج كذلك إلى ربط GA4/Firebase بـ BigQuery من لوحة الإدارة، وتمكين موصل Google Ads عبر Data Transfer Service أو API عند الحاجة لتحديثات داخل اليوم، وضبط سياسات الاحتفاظ ومراقبة التكاليف، وجدولة تشغيل التحويلات مع اختبارات جودة تُراجع حداثة البيانات وتفرّد المفاتيح واتساق العلاقات. بهذه المنظومة تصبح BigQuery طبقة موحّدة وآمنة وقابلة للتوسّع تربط بين التسويق والمنتج والتشغيل، وتقدّم أرقامًا موثوقة تُتَّخذ على أساسها قرارات يومية وسريعة.

GTM

## **GTM للتطبيقات؟**

GTM هو طبقة إدارة وسوم/تتبّع بدون تعديل الكود كل مرة. في **التطبيقات** يعمل جنبًا إلى جنب مع **Firebase Analytics SDK**: أنت تُطلق أحداثك من التطبيق كالمعتاد، وGTM يستخدم هذه الأحداث كمدخلات ثم يقرر لأي “وجهات” يرسلها (GA4، Google Ads… وحتى وجهات سيرفرية). [مساعدة جوجل](https://support.google.com/tagmanager/answer/7003315?hl=en&utm_source=chatgpt.com)[Firebase](https://firebase.google.com/docs/analytics?utm_source=chatgpt.com)

## **لماذا قد يفيدنا؟**

* **تقليل إطلاق نسخ جديدة**: تغييرات التتبع الصغيرة تتم من لوحة GTM بدل تحديث التطبيق. [Google for Developers+1](https://developers.google.com/tag-platform/tag-manager/android/v5?utm_source=chatgpt.com)
* **المرونة**: تشغيل/إيقاف وسوم، إضافة معلمات، تجارب A/B خفيفة على مستوى التتبع. [Google for Developers](https://developers.google.com/tag-platform/tag-manager/android/v5?utm_source=chatgpt.com)
* **حماية وجودة البيانات (اختياري)** عبر **server-side tagging**: نفلتر/نُخفي بيانات قبل إرسالها لوجهات خارجية. مفيد للامتثال وتقليل تسريبات المعرّفات. [Google for Developers+1](https://developers.google.com/tag-platform/tag-manager/server-side/server-side-tagging-for-mobile-apps?utm_source=chatgpt.com)

## **كيف يندمج مع Firebase/GA4 (بسيط رسمياً):**

* **داخل التطبيق**: نستخدم **Firebase Analytics SDK** لإطلاق الأحداث والمعلمات.
* **في GTM (موبايل)**: نعرّف متغيّرات/مشغّلات مبنية على أحداث Firebase (مثل view\_item, add\_to\_cart).
* **وجهات**: نرسل الحدث إلى GA4 أو عبر **خادم GTM** إلى أدوات أخرى (Google Ads… أو وجهات مخصّصة). [مساعدة جوجل](https://support.google.com/tagmanager/answer/7003315?hl=en&utm_source=chatgpt.com)[Google for Developers](https://developers.google.com/tag-platform/tag-manager/server-side/server-side-tagging-for-mobile-apps?utm_source=chatgpt.com)

## **متى نستخدمه… ومتى نكتفي بـ Firebase/GA4 فقط؟**

**استخدم GTM إذا:**

* عندك فريق ماركتنج/تحليلات يحتاج يبدّل منطق التتبع كثيرًا بدون تحديث التطبيق.
* تريد **server-side tagging** للتحكم بالبيانات قبل إرسالها (إخفاء PII، توحيد المعرفات). [Google for Developers+1](https://developers.google.com/tag-platform/tag-manager/server-side/server-side-tagging-for-mobile-apps?utm_source=chatgpt.com)

**اكتفِ بـ Firebase/GA4 فقط إذا:**

* مخطط الأحداث ثابت ونادر التغيير.
* الفريق صغير والتركيز على البساطة وتقليل التعقيد.

## **سيناريو عملي لزهراء (مختصر خطوات)**

1. **ثبّت Firebase Analytics SDK** (موجود أساسًا ضمن خطة Firebase). أطلق أحداث التجارة الإلكترونية القياسية. [Firebase](https://firebase.google.com/docs/analytics?utm_source=chatgpt.com)
2. **أنشئ حاويتي GTM (Android وiOS)** من نوع “Mobile App” واربطهما بمشروعك. [Google for Developers+1](https://developers.google.com/tag-platform/tag-manager/android/v5?utm_source=chatgpt.com)
3. **عرّف في GTM**:
   * *Triggers* مبنية على أحداث Firebase (purchase, begin\_checkout…)،
   * *Variables* لقراءة المعلمات (القيمة/العملة/قائمة العناصر). [مساعدة جوجل](https://support.google.com/tagmanager/answer/7003315?hl=en&utm_source=chatgpt.com)
4. **أضف الوجهات**:
   * إرسال إلى **GA4** (للتقارير)،
   * (اختياري) إعداد **Server-Side GTM** لاستلام بيانات التطبيق ثم التوجيه الآمن إلى الوجهات (Google Ads، وجهات مخصّصة). [Google for Developers+1](https://developers.google.com/tag-platform/tag-manager/server-side/server-side-tagging-for-mobile-apps?utm_source=chatgpt.com)
5. **اختبر** كل شيء عبر وضع Debug (GTM Preview + GA4 DebugView) قبل الإطلاق. [مساعدة جوجل](https://support.google.com/tagmanager/answer/9442095?hl=en&utm_source=chatgpt.com)

## **أفضل ممارسات سريعة**

* أبقِ **مصدر الحقيقة للأحداث في التطبيق**: اسم الحدث، معاييره، وبناؤه ثابت. استخدم GTM فقط “للتحويل/التوجيه”. [Firebase](https://firebase.google.com/docs/analytics?utm_source=chatgpt.com)
* لو هتستخدم **server-side**: استضف حاوية GTM على خادومك/Cloud Run، وطبّق قواعد تنقية البيانات (إزالة أي PII، تعميم المعرّفات). [Google for Developers](https://developers.google.com/tag-platform/tag-manager/server-side/intro?utm_source=chatgpt.com)
* اجعل **التوثيق واحدًا**: قاموس أحداث موحّد (name، params، source)، ومنح الصلاحيات لفريق التسويق مع مراجعة تقنية قبل النشر.

## **محاذير**

* لا تحاول بناء “منطق أعمال” معقّد داخل GTM؛ اجعله لإدارة التتبع فقط.
* كثرة التحويلات داخل GTM قد تُصعّب الديبَغ إن لم توثّق جيدًا.
* بعض الميزات القديمة أو حِزم SDK تغيّر وضعها عبر السنين؛ راجع دومًا آخر إرشادات المطوّرين لـ GTM Mobile و**server-side for mobile**.

sGTM

# **1) sGTM؟**

**GTM Server-Side** = حاوية Google Tag Manager تعمل على خادمك (subdomain أول-طرف مثل sgtm.zahraah.app).  
**الفائدة:**

* رفع جودة القياس وتقليل الحجب (ITP/AdBlock) لأن الإرسال يتم من نطاقك الأول-الطرف.
* توحيد نقطة التجميع: تدخل الأحداث مرّة واحدة وتُوزّع تلقائيًا إلى GA4، Meta CAPI، TikTok، Google Ads… إلخ.
* **تقليل التسريبات**: ما نرسله لكل منصة “بالحد الأدنى اللازم” + تشفير/هَشّ الحسّاس.
* **تحكم بالخصوصية**: نطبّق “حالة الموافقة” القادمة من الـ CMP على مستوى الخادم (لا تنطلق أي تاغ بدون موافقة).

# **2) ما هي الـ CMP؟ ولماذا نحتاجها؟**

**CMP** (OneTrust أو Usercentrics) تدير موافقات المستخدمين (Analytics/Ads/Functional …)، وتولّد **Consent Signal** يحدد ما يُسمح بإرساله.  
**الفائدة:**

* التوافق مع المتاجر واللوائح (GDPR/EEA + Google Consent Mode v2).
* قاعدة واحدة للموافقة تُستَخدم في التطبيق والموقع وتمر عبر sGTM لجميع المنصات.

# **3) تدفق البيانات (نموذج عمل موحّد)**

تطبيق/موقع Zahraah ──(event + consent state)──▶ sgtm.zahraah.app (sGTM)

│ │

│ (يطبّق سياسات الخصوصية) ├─▶ GA4 (Measurement Protocol / Server Tag)

│ ├─▶ Meta CAPI (event\_id + user\_data hashed)

│ ├─▶ TikTok/Snap/Google Ads (حسب الحاجة)

└─◀── شاشة CMP: تجمع الموافقات وتحدّثها باستمرار ──────────┘

# **4) المتطلبات والبنية**

* **نطاق**: sgtm.zahraah.app (DNS + SSL).
* **استضافة**: App Engine أو Cloud Run (GCP) مع autoscaling.
* **حاويات**:
  + Web GTM (إن وُجد موقع/صفحات هبوط)
  + **Server GTM** (أساسي)
* **عملاء (Clients) داخل sGTM**:
  + GA4 Client (يستقبل أحداث GA4/MP).
  + HTTP API Client (لاستقبال Payload مخصّص من التطبيق إن رغبتُم).
* **أمان**: Cloud Armor (معدل الطلبات)، قبول مصادر محدّدة، سر/توقيع للطلبات، CORS مضبوط.

# **5) تصميم الأحداث الموحّد (للجميع)**

اعتمدوا **مخطط GA4** كمرجع، مع **event\_id** و**timestamp\_micros** و**user\_id** (عند تسجيل الدخول) و**app\_instance\_id**:

* **view\_item**, **add\_to\_cart**, **begin\_checkout**, **add\_shipping\_info**, **add\_payment\_info**, **purchase**, **refund**
* **search**, **view\_item\_list**, **select\_item**, **login**, **sign\_up**, **view\_cart**, **remove\_from\_cart**, **add\_to\_wishlist**, **apply\_coupon**
* عناصر الشراء: items[ ] وفيها: item\_id, item\_name, item\_brand, item\_category, price, currency, quantity, discount, coupon.

**مهم جدًا:**

* أرسِلوا نفس **event\_id** عند الإرسال المزدوج (client + server) لتمكين **الـ Deduplication** لدى Meta/TikTok.
* **purchase** يجب أن يحتوي transaction\_id (فريد)، وإلاّ سترتفع نسبة الأخطاء.

# **6) ربط التطبيق (Flutter) مع sGTM**

**خيار A (الأبسط/الأكثر شيوعًا للتطبيقات):**

* استخدموا **Firebase Analytics** للأحداث داخل التطبيق.
* أنشئوا **Cloud Function** (أو خدمة خفيفة) تلتقط أحداث Firebase (real-time stream أو Pub/Sub) وتُعيد إرسالها إلى **sGTM** كـ Measurement Protocol مع:  
  event\_name, event\_params, user\_id/app\_instance\_id, event\_id, engagement\_time\_msec, consent\_state.

**خيار B (أسرع وأقرب للزمن الحقيقي):**

* Service داخل التطبيق ترسل مباشرًا طلب HTTPS إلى https://sgtm.zahraah.app/mp بنفس بنية GA4 MP (مع Header سرّي).
* يلتقط **GA4 Client** داخل sGTM الحدث ويوزّعه للوجهات المصرّح بها حسب الموافقة.

**متى نستخدم MMP؟** (موصى به سابقًا: AppsFlyer/Adjust)  
للإسناد الإعلاني على iOS/Android (ما بعد ATT/SKAN)، ويمكن تفعيل **Forwarding** لما بعد التنصيب إلى BigQuery و/أو sGTM.

# **7) ربط sGTM بالوجهات (Tags)**

* **GA4 (Server Tag)**: يرسل إلى خاصية GA4 الرئيسية.
* **Meta CAPI**:
  + الماب: purchase, add\_to\_cart, view\_content, initiate\_checkout.
  + **Dedup**: استخدم event\_id نفسه من التطبيق/الويب.
  + **User data**: أرسل (إن توفّر) بريد/هاتف **مُهشّ بـ SHA-256** + external\_id (user\_id).
  + للفويب: مرّر fbp/fbc إن وُجدت لتعلية التطابق.
* **TikTok Events API / Snap CAPI / Google Ads EC**: نفس الفكرة (event\_id + user\_data hashed).
* **Webhook داخلي/BI** (اختياري): نسخة من الحدث إلى Pub/Sub → BigQuery (أرشفة/تدقيق).

# **8) CMP + Consent Mode (تطبيق عملي)**

## **8.1 على التطبيق (OneTrust/Usercentrics SDK)**

* اعرض **شاشة الموافقة** (مع تهيئة اللغات) عند التشغيل الأول + صفحة “الخصوصية” لتعديل الموافقات.
* **تصنيفات مقترحة**:
  + **Essential/Functional** (تشغيل التطبيق)
  + **Analytics** (GA4/Mixpanel)
  + **Marketing/Ads** (Meta/TikTok/Ads)
* خزّن الحالة محليًا، وأرفِقها مع كل حدث يُرسل إلى sGTM (ex: consent\_analytics=true/false, consent\_ads=true/false).
* على iOS: نسّق الـ **ATT** (AppTrackingTransparency) مع CMP:
  + Pre-prompt لشرح الفائدة → نافذة ATT → (حسب المنطقة) شاشة CMP.

## **8.2 على الويب (إن وُجد)**

* دمّج CMP مع **GTM Web** + فعّل **Google Consent Mode v2**:
  + إشارات Google: ad\_user\_data, ad\_personalization, analytics\_storage, ad\_storage…
  + عند الرفض: تُرسل Google **cookieless pings** (قياس مجمّع)، فيما sGTM يمنع أي تاغ تسويقي.
* **EEA/TCF 2.2**: فعّل “TC String” من الـ CMP، ومرّره تلقائيًا لعلامات Google عبر GTM/Consent Mode.

## **8.3 فرض/تطبيق الموافقة داخل sGTM**

* في كل Tag داخل sGTM: أضِف شرطًا (Trigger/Blocking) يعتمد على المتغيّرات consent\_analytics / consent\_ads.
* إن كانت الموافقة مرفوضة:
  + **لا ترسل** إلى Meta/TikTok.
  + GA4: إمّا لا ترسل، أو ترسل قياسًا مجمّعًا (حسب سياسة Consent Mode وخياراتكم).

# **9) الأمان والجودة**

* **توقيع الطلبات** (secret/JWT في Header) لمنع إساءة استخدام endpoint.
* **Rate limiting** عبر Cloud Armor.
* **تنقية البيانات** داخل sGTM: إزالة أي PII خام وإرسال هاش فقط.
* **مراقبة**: Stackdriver Logs في GCP + تنبيهات أخطاء/ارتفاع رفض التاجات.
* **تتبّع الجودة**: لوحة في Looker Studio:
  + عدد الأحداث/المرسَل لكل وجهة
  + نسبة الموافقات حسب القناة والمنطقة
  + معدلات التطابق (Match Quality) في Meta/TikTok
  + نسبة Dedup نجاحًا/فشلًا
  + “الأحداث المرفوضة بسبب الموافقة”.

# **10) خطة تنفيذ (مبدئية)**

-حجز sgtm.zahraah.app + تهيئة App Engine/Cloud Run + إنشاء حاوية sGTM.  
-إعداد **GA4 Client** + نقطة استقبال MP + سرّ التوقيع.  
-دمج **CMP SDK** في التطبيق + تخزين وتمرير حالة الموافقة.  
-إنشاء Tags في sGTM: GA4, Meta CAPI, TikTok.  
-تفعيل Dedup (event\_id موحّد) + اختبار عبر وضع Preview في sGTM و Events Manager.  
-سياسة الأمان (CORS/Rate limit) + سجلات وتنبيهات.  
-إطلاق تدريجي (10% → 50% → 100%) + مراجعة المقاييس.

# **11) قائمة تحقق مختصرة (قبل الإطلاق)**

* sgtm.zahraah.app يعمل عبر HTTPS + استجابة صح.
* كل حدث يحتوي: event\_name, event\_id, timestamp\_micros, user\_id/app\_instance\_id.
* **purchase** يحتوي transaction\_id + items[] كاملة.
* تمرير consent\_analytics/consent\_ads مع كل حدث.
* Meta/TikTok تظهر **Dedup = OK** و**Match Quality** جيد.
* لا توجد PII خام (Email/Phone) غير مُهشّ.
* لوحات مراقبة الجودة مفعّلة.

علامة التبويب 40

هذه **سياسة شاملة + آلية عمل + معايير تنفيذ مفصّلة** لـ **Google Tag Manager (GTM)** (ويب + Server-Side) جاهزة للإدراج في مستندك. رتّبتها: ملخّص الاعتماديات → الأهداف والنطاق → المعمارية → الحوكمة والصلاحيات → طبقة البيانات (Data Layer) → الخصوصية/الموافقات (Consent Mode v2) → إعداد العلامات والمحفّزات → GTM Server-Side → الأمن/CSP → ضمان الجودة والاختبار → الأداء → SLI/SLO → Runbooks → قوائم تنفيذ → ملاحق (نماذج كود).

# **1) ملخّص الاعتماديات (Executive Summary)**

**المتصفح/التطبيق** ⟶ **CMP** (الموافقة) ⟶ **GTM Web** (حاوية المتجر) ⟶ يطلق **GA4/إعلانات/بكسلات** بالاعتماد على **Data Layer** + الموافقات  
اختياري/مُستحسن: **GTM Server-Side (sGTM)** على نطاق فرعي أول-طرف gtm.yourdomain.com لاستقبال الضربات وإعادة توجيهها بشكل آمن وفعّال (CAPI/EC/…).

حول ذلك: **سياسة هوية (user\_id بعد التحقق فقط)**، **CMP Consent Mode v2**، **حظر PII**، **CSP/Headers أمنية**، **بيئات Dev/Stg/Prod**، **لوحات مراقبة**، **عمليات نشر بنظام الموافقات**.

# **2) الأهداف والنطاق**

* توحيد القياس والتحويل عبر القنوات مع **حماية الخصوصية**.
* طبقة بيانات **حتمية** وقابلة للاختبار، بلا **PII خام**.
* تقليل الحمل والأخطاء عبر **sGTM** وإطلاق مشروط بالموافقة.
* دعم **GA4**، **Google Ads (Conversions + EC)**، **Floodlight**، **Meta/TikTok/Snap**، وواجهات **CAPI**.

**النطاق:** موقع/ويب المتجر، صفحات الهبوط، لوحة التحكم (إن لزم)، sGTM على سحابة مُدارة، تكامل CMP.

# **3) المعمارية والبيئات**

* **حاويات GTM منفصلة**:
  + GTM-XXXX Dev (الديف) | GTM-YYYY Staging | GTM-ZZZZ Prod.
  + **Workspaces** لكل ميزة/حملة.
* **sGTM**: نشر على **App Engine/Cloud Run** مع **نطاق فرعي أول-طرف**: gtm.yourdomain.com + TLS.
* **DNS**: gtm. (Server) + www. (Web).
* **خط نشر**: Dev ⇢ Stg (مراجعة) ⇢ Prod (نشر بموافقة ثنائية).

# **4) الحوكمة والصلاحيات (Governance)**

* **أدوار GTM**:
  + **Viewer** (قراءة فقط)، **Editor** (تعديل بلا نشر)، **Approver** (مراجعة)، **Publisher** (نشر)، **Admin** (إدارة).
  + مبدأ **أقل صلاحية** + **Two-person rule** للنشر في Prod.
* **إدارة التغيير**: كل Workspace مربوط بتذكرة (Jira/Asana)، **تسمية الإصدارات**: YYYY-MM-DD\_[ticket]\_[scope].
* **قواعد المحتوى**:
  + **ممنوع** Tag من نوع **Custom HTML/JS** بلا مراجعة أمنية.
  + نستخدم **Community/Verified Templates** أولًا.
  + ملفات الطرف الثالث **مسموح بها** فقط من قائمة سماح (allowlist).
* **المراجعة/التدقيق**: تفعيل **Change History**، حفظ **Exports** شهريًا (نسخ احتياطي للحاوية).

# **5) طبقة البيانات (Data Layer) – المعايير**

## **5.1 مبادئ**

* كل حدث يُدفع إلى dataLayer.push({...}) بتسمية **snake\_case** ومعايير ثابتة.
* **لا PII خام** (لا بريد/هاتف/اسم) — استخدم **user\_id** الداخلي بعد Auth فقط.
* العملات/المبالغ: كود عملة ISO-4217 في currency, المبالغ أرقام عشرية.

## **5.2 مخطّط أساسي للأحداث (GA4 Enhanced Ecommerce)**

**page\_view** (على كل صفحة):

<script>

window.dataLayer = window.dataLayer || [];

dataLayer.push({

event: 'page\_view',

page: { url: location.href, path: location.pathname, title: document.title, locale: 'ar-YE' },

user: { user\_id: null, logged\_in: false }

});

</script>

**view\_item**:

dataLayer.push({

event: 'view\_item',

ecommerce: {

currency: 'SAR',

value: 199.00,

items: [{

item\_id: 'SKU123',

item\_name: 'Dress A',

item\_brand: 'BrandX',

item\_category: 'Dresses',

item\_variant: 'Red-S',

price: 199.00,

quantity: 1

}]

}

});

**add\_to\_cart** / **begin\_checkout** / **add\_payment\_info** (مع payment\_type) / **purchase**:

dataLayer.push({

event: 'purchase',

ecommerce: {

currency: 'SAR',

transaction\_id: 'ORD\_789',

value: 215.00, tax: 10.00, shipping: 6.00, coupon: 'EID10',

items: [ /\* نفس بنية view\_item، مع quantities \*/ ]

}

});

**قاعدة ذهبية**: transaction\_id **فريد** وواحد/طلب — يمنع تكرار الشراء.

## **5.3 خصائص مستخدم/جلسة (اختياري)**

dataLayer.push({

event: 'login',

user: { user\_id: 'u\_12345', tier: 'gold', region: 'SA' }

});

# **6) الخصوصية والموافقات (CMP + Consent Mode v2)**

## **6.1 الإشعال الافتراضي**

* قبل الموافقة: **رفض افتراضي** لكل التخزين باستثناء الأمني.
* بعد موافقة المستخدم عبر CMP: تحديث القيم.

**تهيئة (gtag consent, v2):**

<script>

window.dataLayer = window.dataLayer || [];

function gtag(){ dataLayer.push(arguments); }

gtag('consent', 'default', {

'ad\_storage': 'denied',

'analytics\_storage': 'denied',

'functionality\_storage': 'granted',

'security\_storage': 'granted',

'ad\_user\_data': 'denied',

'ad\_personalization': 'denied',

'wait\_for\_update': 500

});

</script>

**بعد موافقة CMP (مثال):**

gtag('consent', 'update', {

ad\_storage: 'granted',

analytics\_storage: 'granted',

ad\_user\_data: 'granted',

ad\_personalization: 'denied'

});

**قواعد**:  
• لا تُرسل **Enhanced Conversions**/CAPI أو أي معرفات إلا عند **ad\_user\_data = granted**.  
• **user\_id** لا يُضبط إلا بعد Auth ووفق السياسة.  
• احترم طلبات الحذف/التصدير (DSAR).

# **7) إعداد العلامات والمحفّزات والمتغيرات (Web GTM)**

## **7.1 التسمية**

* **Tag**: [GA4] Event – add\_to\_cart، [ADS] Conversion – Purchase
* **Trigger**: CUSTOM – event: purchase، PAGEVIEW – All pages
* **Variable**: DLV – ecommerce.value، JS – location.pathname

## **7.2 GA4**

* **Tag GA4 Configuration**:
  + Measurement ID (بيئة)، تفعيل **Send Page View** إن كان page\_view عبر gtag/auto؟ (نوصي **تعطيله** إذا تدفع page\_view يدويًا طبقة البيانات).
  + **User-ID** (من DLV user.user\_id) بعد تسجيل الدخول.
  + **Fields to Set**: allow\_ad\_personalization\_signals تبعًا للموافقة.
* **GA4 Events**: لكل حدث من Data Layer، **Tag** من نوع GA4 Event يقرأ من DLV.

## **7.3 Google Ads**

* **Conversion Tag**: تشغيل على purchase مع **transaction\_id** للـ dedup.
* **Enhanced Conversions (ملف شخصي)**:
  + **محجوبة افتراضيًا**؛ تُفعّل فقط عند ad\_user\_data='granted'.
  + البريد/الهاتف **مُجزأ SHA-256** عبر sGTM (مفضل) أو عميل عند الموافقة.

## **7.4 Floodlight/أخرى (Meta/TikTok/Snap)**

* استخدم **Templates الرسمية**.
* Meta/TikTok: إرسال عبر **sGTM (CAPI)** لتحسين التسليم وتقليل الحظر.
* **Dedup**: استخدم event\_id ثابت = transaction\_id أو حقل مركّب.

## **7.5 المحفّزات**

* **Custom Event** يطابق event من Data Layer (مفضل).
* Page View (DOM Ready/Window Loaded) **للحد الأدنى** فقط.
* Scroll/Click/Form/Timer **بحدود** (تجنب الضجيج).
* **استثناءات**: لا إطلاق بدون موافقة لازمة.

# **8) GTM Server-Side (sGTM)**

## **8.1 النشر**

* استضافة على **gtm.yourdomain.com** مع **TLS**، ربط عبر **Load Balancer** مُدار.
* **أوضاع**: Dev/Stg/Prod مستقلّة؛ سجلات وصول مهيكلة.

## **8.2 التوصيل (Routing)**

* **Clients**: GA4, Google Ads, FB CAPI, TikTok Events API.
* **Tags**: تحويل/تنقية/إثراء (IP Geo فقط إذا لزم ولا يُخزَّن)، إضافة **Server-Side Conversion Linker**.
* **1st-party endpoints**: يقلّل الحظر ويحسن الدقة.

## **8.3 الخصوصية والأمن**

* **PII**: لا تُخزَّن؛ إزالة الحقول الحساسة، Hash للمسموح عند الموافقة.
* **Access**: SA منفصل، جدار حماية على المنافذ، مراقبة RPS/Error.
* **تكلفة**: مراقبة CPU/GB-s، تجاوز الحمل بالحد الأدنى.

# **9) الأمن وCSP والرؤوس**

* **CSP (مقترح مبدئي)**:

Content-Security-Policy:

default-src 'self';

script-src 'self' 'unsafe-inline' 'unsafe-eval' https://www.googletagmanager.com https://gtm.yourdomain.com;

connect-src 'self' https://www.google-analytics.com https://www.googletagmanager.com https://gtm.yourdomain.com;

img-src 'self' data: https:;

style-src 'self' 'unsafe-inline' https:;

frame-src https://www.googletagmanager.com https://www.youtube.com;

شدّد السياسة تدريجيًا. مع GTM يصعب إزالة كل inline؛ استخدم **Nonce** إن أمكن للسكريبتات الحرجة.

* **Headers أخرى**: Referrer-Policy: strict-origin-when-cross-origin, Permissions-Policy (تعطيل ما لا يلزم)، X-Content-Type-Options: nosniff.
* **حظر Custom HTML**: لا يُسمح إلا بعد مراجعة أمنية + حقل **Source allowlist**.

# **10) ضمان الجودة والاختبار (QA)**

* **GTM Preview/Debug** + **Tag Assistant Companion** على Staging.
* **GA4 DebugView**: تحقق من وصول الأحداث ومعلماتها الصحيحة.
* **Network**: فحص الطلبات (status, payload, consent hints).
* **تطابق الشراء**: تقرير **Orders vs GA4 purchase** (انحراف ≤ 3%).
* **Dedup**: تأكد أن transaction\_id لا يُرسل مرتين.
* **CMP**: اختبارات مصفوفة (جميع تركيبات الموافقات) ⟶ التحقّق من السلوك.

# **11) الأداء**

* **ميزانية التاجات**: ≤ 1–2 Tags على Pageview (Config/Consent)، والباقي **على أحداث**.
* **Sequencing**: لا تُحمل بكسلات غير ضرورية عند الإقلاع.
* **إرجاء** (Defer) العلامات الثانوية حتى أول تفاعل/مشاهدة 50% من الصفحة (عند موافقة مناسبة).
* **sGTM** لتقليل ضغط الشبكة والمتتبعات العميلية.

# **12) SLI/SLO (قابلة للقياس)**

| **SLI** | **SLO** |
| --- | --- |
| نسبة صفحات بها page\_view صالح | ≥ 99.5% |
| توافر sGTM | ≥ 99.9% شهريًا |
| انحراف purchase vs الطلبات | ≤ 3% يوميًا |
| نسبة إطلاق بدون موافقة لازمة | 0 |
| زمن إطلاق GA4 p95 | ≤ 200ms بعد DOMReady |
| تكرار transaction\_id | 0 شهريًا |
| أعطال نشر Prod (Rollback) | 0 لكل ربع |

# **13) Runbooks (مختصرات جاهزة)**

**RB-1: توقفت الأحداث عن الوصول**

1. فحص نشر الحاوية الأخير (Diff + Preview).
2. التحقق من CMP وتحديثات Consent.
3. فحص أخطاء الشبكة/حظر المتصفحات/AdBlock؛ التحويل عبر sGTM endpoints.

**RB-2: تضاعف التحويلات/المشتريات**

1. فحص transaction\_id وتعدّد إطلاق Tag Ads/GA4.
2. عطّل أحد المسارات (onload/SPA) أو أضف **حارس sessionStorage**: لا تُرسل نفس transaction\_id مرتين.
3. معالجة بيانات المنصّة (exclude duplicates).

**RB-3: sGTM انقطاع**

1. تحويل مؤقت إلى وجهات Google المباشرة (Fail-open).
2. تحجيم الخدمة أو مراجعة الأخطاء/الحصص.
3. Postmortem + اختبارات ضغط.

**RB-4: مخالفة خصوصية (إطلاق دون موافقة)**

1. تعطيل التاج المخالف فورًا (Hotfix workspace).
2. مراجعة قواعد Trigger/Consent.
3. تقرير حادث + تدريب الفريق.

**RB-5: فشل تدقيق GA4/بيانات**

1. تحقق من خرائط Data Layer ↔️ Tags.
2. راجع تغييرات PIM/Checkout التي تؤثر على الأحداث.
3. أعد النشر بعد إصلاح/اختبار.

# **14) قوائم تنفيذ (Checklists)**

## **A) تأسيس**

* إنشاء حاويات Web + Server-Side (Dev/Stg/Prod).
* ربط **CMP** وإعداد **Consent Mode v2** (denied افتراضيًا).
* إعداد sGTM على gtm.yourdomain.com + TLS + Logs.
* تطبيق **CSP** مبدئي + رؤوس أمان.
* تصميم **Data Layer Spec** (الأحداث والمعلمات) واعتماده.

## **B) الإعداد القياسي**

* GA4 Config + Events (من Data Layer).
* Google Ads Conversions + Enhanced Conversions (مشروطة بالموافقة).
* Floodlight/Meta/TikTok عبر **sGTM** حيث أمكن (CAPI/Events API).
* Conversion Linker (Web + sGTM).
* متغيرات DLV لكل حقول ecommerce/user/page.

## **C) الحوكمة**

* أدوار GTM مقيّدة؛ منع Custom HTML إلا بمراجعة.
* Workspaces لكل ميزة؛ إصدارات مُسمّاة.
* موافقة ثنائية قبل نشر Prod؛ Export شهري.

## **D) الاختبار والنشر**

* خط اختبار: Preview + Tag Assistant + DebugView + Network.
* مصفوفة CMP (جميع التركيبات).
* تقرير **Orders vs GA4** يومي؛ تنبيه عند انحراف > 3%.
* نشر تدريجي: Stg ⇢ Prod (خارج الذروة).

## **E) التشغيل المستمر**

* مراجعة شهرية للعلامات غير المستخدمة.
* تفقد أداء التتبع (زمن إطلاق/عدد الطلبات).
* تحديث قوالب/قنوات جديدة عبر sGTM.

# **15) الملاحق (نماذج كود سريعة)**

## **A) إدراج حاوية GTM (Web)**

<!-- Head -->

<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':

new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],

j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=

'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);

})(window,document,'script','dataLayer','GTM-XXXX');</script>

<!-- Body -->

<noscript><iframe src="https://www.googletagmanager.com/ns.html?id=GTM-XXXX"

height="0" width="0" style="display:none;visibility:hidden"></iframe></noscript>

## **B) حماية من التكرار (purchase)**

if (!sessionStorage.getItem('purch\_'+transaction\_id)) {

dataLayer.push({ event: 'purchase', ecommerce: {/\*...\*/} });

sessionStorage.setItem('purch\_'+transaction\_id, '1');

}

## **C) ربط User-ID بعد التحقق**

dataLayer.push({ event: 'login', user: { user\_id: 'u\_12345', logged\_in: true } });

### **ملاحظات سياقية (المنطقة/الشبكات)**

* شبكات متذبذبة: اجعل إطلاق التتبع **بعد DOMReady** وقلّل عدد الطلبات الأولية؛ sGTM يساعد على تجميع الوجهات.
* الامتثال المحلي: لا تخزّن وسوم/تعريفات قد تُعد حسّاسة؛ احصر Enhanced Conversions بموافقة واضحة.

دليل مختصر (مهم)

ربط Google Analytics (GA4 عبر Firebase) وأين يتم تنفيذ كل جزء، مع **نِسَب توزيع العمل** بين التطبيق، الـAPI، والتحليلات/الإدارة.

# **1) المتطلبات الأساسية (مرة واحدة)**

* إنشاء **Firebase Project** وربطه بـ **GA4 Property** (قناتان منفصلتان: Android + iOS).
* إعداد **Data Streams** (Android/iOS) + تفعيل **BigQuery Export**.
* إنشاء **Measurement Protocol Secret** لاستخدامه من الخادم (لأحداث الخادم).
* تعريف **أدوار وصلاحيات**: Marketing (قراءة/تقارير)، Product (قراءة/إنشاء Events)، Dev (إعداد/Debug).
* **بيئات منفصلة**: dev / stage / prod (منع خلط بيانات التطوير بالإنتاج).
* سياسة **Consent/Privacy** داخل التطبيق (تمكين/تعطيل الجمع لكل مستخدم).
* توثيق **مخطط الأحداث (Event Taxonomy v1)** قبل البدء.

# **2) توزيع العمل**

| **المسار** | **أين يتم** | **النسبة التقريبية** | **المخرجات** |
| --- | --- | --- | --- |
| تتبّع التطبيق (SDK) | **التطبيق** | **65%** | ربط SDK، user\_id، الأحداث، الخصائص |
| أحداث التحقق النهائية | **الخادم / API** | **20%** | Measurement Protocol، توحيد order\_id، dedup |
| إدارة التحليلات/لوحات | **Analytics/Marketing** | **15%** | Conversions، Audiences، Funnels، تنبيهات |

النِّسب للدلالة على حجم العمل؛ قد تختلف قليلًا حسب جاهزية الكود.

# **3) ما الذي يتم داخل التطبيق (Flutter)**

* **الهوية**: تعيين user\_id بعد التسجيل/الدخول، واستخدام app\_instance\_id قبل الدخول.
* **UTM**: حفظ **أول وآخر مصدر** (first/last UTM) محليًا وربطهما كمعلومات **User Properties**.
* **الأحداث الأساسية (E-commerce GA4)**:  
  view\_item\_list, select\_item, view\_item, add\_to\_cart, view\_cart, begin\_checkout, add\_shipping\_info, add\_payment\_info, purchase, refund (عند اللزوم)، بالإضافة إلى login, sign\_up, search, share, screen\_view.
* **الخصائص (Event Params / User Properties)**: العملة، القيمة، الفئة، العلامة التجارية، المقاس/اللون، إصدار التطبيق، نظام التشغيل، قناة الاكتساب.
* **الجلسات**: احترام جلسات GA4 الافتراضية، وعدم إنشاء مؤقتات مخصّصة إلا للضرورة.
* **الخصوصية**: احترام موافقات المستخدم (تمكين/تعطيل الجمع)، وإخفاء أي PII.
* **الجودة**: DebugView في dev، ومراجعة Realtime بعد كل إصدار.

# **4) ما الذي يتم في الخادم / الـAPI**

* إرسال أحداث **حسّاسة/نهائية** عبر **Measurement Protocol** (MP):
  + purchase عند تأكيد الطلب (لـCOD: عند **تأكيد الاستلام** أو الحالة التي تعتمدونها كمؤشر إيراد).
  + refund عند الإرجاع.
* **توحيد الهوية**: تضمين user\_id وevent\_id وsession\_id (إن أمكن) في أحداث MP لتفعيل **الدمج وإزالة التكرار** (dedup) مع أحداث التطبيق.
* **المصدر/القناة**: تمرير آخر UTM مع المستخدم (إن وُجد) لتناسق التقارير.
* **الحَوْكمة**: عدم إرسال أحداث مزدوجة (نفس event\_id)؛ سياسة إعادة المحاولة مع طوابير عند انقطاع الشبكة.

# **5) ما الذي يتم في Analytics/Marketing (داخل GA4)**

* ضبط **Conversions**: purchase, begin\_checkout, add\_payment\_info, add\_shipping\_info, sign\_up.
* إنشاء **Audiences**: VIP، عملاء COD، حديثو الشراء، مهجورو السلة 7/14/30 يوم.
* بناء **Funnels/Explorations** لمسار الشراء + **Retention Cohorts**.
* **تنبيهات**: هبوط التحويل، ارتفاع أخطاء الدفع، تغيّر حاد في Add-to-Cart.
* ربط **BigQuery** و**لوحات Looker Studio** (CEO/Product/Marketing).
* تعريف **Channel Grouping** و**Campaign Naming** (اتفاق موحّد مع فريق الإعلانات).

# **6) مخطط الحد الأدنى للأحداث (جاهز للاعتماد)**

**شاشات/تصفح:** screen\_view, view\_item\_list, select\_item, view\_item, search  
**السلة/الدفع:** add\_to\_cart, view\_cart, begin\_checkout, add\_shipping\_info, add\_payment\_info, purchase, refund  
**الهوية/التفاعل:** login, sign\_up, share, tutorial\_complete (إن وُجد)

# **7) الهوية والـUTM (مهم)**

* **First/Last UTM**: خزّنها محليًا (secure storage) وارسِلها كـ **User Properties** (first\_source/medium/campaign & last\_source/medium/campaign).
* بعد تسجيل الدخول: **ارفع النسخ** للـAPI لربطها بملف العميل (للاستخدام التحليلي لاحقًا).
* حافظ على **الزمن** (timestamps) لكل من first/last.

# **8) البيئات والجودة**

* ثلاث بيئات منفصلة (dev/stage/prod) لكل من Firebase/GA4.
* حجب إرسال بيانات dev إلى prod، وتمكين **debug mode** فقط على dev.
* **فحوصات قبول (DoD)** قبل أي إصدار:
  1. ظهور جميع الأحداث في **DebugView**.
  2. مطابقة purchase بين GA4 وطلبات النظام خلال 24 ساعة (±2%).
  3. عمل الروابط العميقة وUTM وتعبئة الخصائص.

# **9) أمان وامتثال**

* عدم إرسال **PII** (أرقام هاتف/عناوين/بريد خام).
* حماية **MP Secret** على الخادم فقط (لا يوضع في التطبيق).
* نصوص الموافقات داخل الإعدادات (تمكين/تعطيل التحليلات والإشعارات).

# **10) مسؤوليات سريعة (RACI مبسّط)**

| **المهمة** | **التطبيق** | **الـAPI** | **التسويق/التحليلات** | **المالك النهائي** |
| --- | --- | --- | --- | --- |
| ربط SDK وتتبّع الأحداث | ✔️ |  |  | **التطبيق** |
| user\_id وUTM (first/last) | ✔️ | ◻️ رفع للملف |  | **التطبيق** |
| purchase/refund عبر MP |  | ✔️ |  | **الخادم** |
| Conversions/Audiences/Dashboards |  |  | ✔️ | **التحليلات** |
| BigQuery/Looker وربط البيئات |  | ◻️ حساب خدمة | ✔️ | **التحليلات** |
| سياسة الخصوصية/الموافقات | ✔️ (UI) | ◻️ | ✔️ (نصوص) | **المنتج** |

## **الخلاصة**

* **ما نحتاجه:** GA4+Firebase، BigQuery، MP Secret، مخطط أحداث موحّد، بيئات منفصلة، موافقات خصوصية.
* **أين العمل؟** 65% تطبيق (الأحداث/الهوية/UTM)، 20% خادم (أحداث نهائية/Dedup/MP)، 15% تحليلات (Conversions/Audiences/لوحات).
* بهذه القسمة، تكون البنية **صحيحة ومدروسة** وقابلة للقياس واتخاذ القرار.

علامة التبويب 5

**النقاط الرئيسية لـ Google Analytics (GA4)**

1. **هيكلة الـProperty & Streams**Property واحدة، وربط iOS وAndroid (وWeb إن وُجد) بمشروع Firebase نفسه.
2. **خطة القياس (Measurement Plan)**قائمة الأحداث والمعلمات والخصائص وتعريفات المؤشرات وصيغ الحساب.
3. **أحداث التجارة الإلكترونية القياسية**view\_item\_list → select\_item → view\_item → add\_to\_cart → begin\_checkout → add\_shipping\_info → add\_payment\_info → purchase مع مصفوفة items[].
4. **أحداث/معلمات مخصّصة لسوقك**payment\_type (COD/Prepaid)، delivery\_city/zone/partner، size، color، stock\_status، coupon/discount\_value.
5. **التحويلات (Conversions)**تمييز purchase أساسًا، ويمكن تمييز مراحل وسيطة مثل begin\_checkout/add\_payment\_info.
6. **خصائص المستخدم (User Properties)**مثل preferred\_payment، favorite\_size، favorite\_category، loyalty\_tier (اختَر 3–5 فقط).
7. **الاكتساب ونسب التحويل (Attribution & UTMs)**حوكمة UTM، تفعيل **Data-Driven Attribution**، وربط Google Ads واستيراد التحويلات والجماهير.
8. **التكاملات الأساسية**BigQuery Export، (اختياري) Search Console للويب، وربط CAPI/Conversions API للقنوات الإعلانية.
9. **أحداث السيرفر لما بعد الشراء (Measurement Protocol)**حدث order\_status\_update مع order\_id, order\_status, payment\_collected, cod\_attempts, delivery\_partner.
10. **الجماهير (Audiences) للتشغيل والتسويق**متروكو السلة، High AOV، مهتمو “فساتين السهرة”، ناجحو COD المتكرر.
11. **الاستكشافات (Explorations)**Funnels لمسار الشراء، Cohorts للاحتفاظ، تحليل البحث داخل التطبيق (CTR، Zero-Results).
12. **التقارير واللوحات**تنفيذية يومية، قنوات التسويق وROAS، مسار الشراء والبحث، عمليات COD/RTO حسب المدينة وشريك التوصيل.
13. **تكلفة القنوات وROAS مجمّع**استيراد تكلفة القنوات غير Google داخل GA4 أو احتسابها عبر BigQuery ولوحات Looker Studio.
14. **التنبيهات (Custom Insights)**هبوط مفاجئ في CR/ROAS، اختفاء حدث رئيسي، ارتفاع Zero-Results، أو Crash/Latency.
15. **الجودة والحوكمة**Naming موحّد (snake\_case)، إصدار مخطط schema\_version، أذونات الوصول، وسياسات الاحتفاظ بالبيانات.
16. **الخصوصية والامتثال**Consent Mode (إن لزم)، Google Signals بإعدادات مناسبة، تصفية الزيارات الداخلية وأجهزة الاختبار.
17. **ضمان الجودة (QA) وDebug**DebugView، سيناريوهات فحص كاملة، مطابقة الإيراد مع النظام الخلفي، فحوص قبل كل إصدار للتطبيق.
18. **الاختبارات والتحسين (Experimentation)**A/B عبر Firebase Remote Config، مع مؤشرات رئيسية وحواجز أداء (Crash/Latency).
19. **مصدر الحقيقة والتحليلات المتقدمة**BigQuery كمستودع خام، Views/Models موحّدة، ربط بيانات الطلب والشحن لتقارير COD/RTO المتقدمة.
20. **مخرجات ووثائق أساسية**وثيقة Measurement Plan، جدول الأحداث/المعلمات/الخصائص، مخطط تدفّق البيانات، قائمة التنبيهات، ودليل UTM.

علامة التبويب 1

# **هل الـ Analytics مستقل؟**

* **نعم “مسار مستقل” تشغيليًا** (له Backlog ومالِك وتسليمات خاصة)، لكنه **مُندمج** مع Firebase والتطبيق والباك-إند.
* التفكير الصحيح: **طبقة قياس/بيانات مشتركة** تخدم الـ Build/Run/AI/التسويق، وتحتاج تعاون التطبيق + الباك-إند + التسويق + البيانات.

# **بيانات/مصادر إضافية نحتاجها (غير ما جمعناه في Firebase)**

* **العمليات**: جدول الطلبات/المدفوعات/الشحن/المرتجعات (لتمييز *Placed* vs *Delivered* خصوصًا COD).
* **الكتالوج**: أبعاد المنتج (الفئة حتى 5 مستويات، العلامة، المقاس/اللون، السعر وقت الشراء).
* **التسويق**: **Ad Spend** يومي لكل حملة، قاموس **UTM** وcampaign\_id موحّد.
* **المراسلة**: معرف الحملة في FCM/Remote Config/تجارب A/B للربط بالأثر (campaign\_id, experiment\_id).
* **الهوية**: user\_id (auth.uid)، transaction\_id موحّد بين GA4 والباك-إند، device tier، نسخة التطبيق، المدينة/اللغة (بدون أي PII).
* **الخصوصية**: علم **Marketing Consent**، سياسة الاحتفاظ، المناطق الزمنية (UTC+3).

# **النقاط الرئيسية (مثل بقية الأقسام)**

1. **نموذج الأحداث GA4**: خريطة view\_item → add\_to\_cart → begin\_checkout → add\_payment\_info → purchase (+ refund) مع مصفوفة items[].
2. **التحويلات (Conversions)**: تعريف واضح لـ **Purchase** (Prepaid عند الدفع، COD عند التأكيد + “Delivered Purchase” للصافـي)، وإعلانها في GA4/استيرادها للإعلانات.
3. **الهوية والربط**: توحيد user\_id وtransaction\_id وربط حملات FCM/RC/Ads عبر campaign\_id.
4. **خصائص المستخدم والجماهير**: is\_cod\_user, customer\_tier, preferred\_size/color, city… وجماهير (New/ATC No Purchase/At-Risk/VIP…).
5. **BigQuery Export**: طبقة خام + **Curated** مع Partition/Cluster، فحوصات جودة يومية، وScheduled Queries.
6. **تكامل الإعلانات وUTM**: سياسة تسمية موحّدة، Ad Spend → BigQuery، Deferred Deep Linking، SKAN عند الحاجة.
7. **لوحات و-KPIs**: Funnel/Revenue/ROAS/Quality/Push/LTV… مع حواجز وتنبيهات.
8. **الحوكمة والجودة**: Data Catalog، فحوصات (ازدواج transaction\_id، فجوة GA4↔العمليات ≤ 3%)، ومراجعات دورية.
9. **الامتثال والتكلفة**: عدم تخزين PII، سياسات احتفاظ، مراقبة تكلفة الاستعلامات/التخزين.

# **تفاصيل مُوجّهة للمبرمجين (Checklist تسليم)**

**أ) تطبيق (iOS/Android):**

* إطلاق جميع أحداث التجارة مع items[] وبارامترات currency/value/transaction\_id/payment\_type/coupon.
* إرسال **User Properties**: is\_cod\_user, customer\_tier, city, app\_version, device\_tier…
* تمرير campaign\_id في الروابط العميقة من FCM/البنرات.
* احترام DebugView في dev/staging فقط.

**ب) الباك-إند/Functions:**

* توحيد **transaction\_id**، كتابة “لقطة” السعر/العناصر في الطلب، وإتاحة حالة **Delivered**.
* Webhooks الدفع (Idempotent) وتحديث حالات الطلب/الدفع، وتوليد أحداث **Refund** عند اللزوم (للتحليل فقط).
* تصدير جداول تشغيلية إلى BigQuery (Orders/Payments/Shipments/Products/Coupons).

**ج) البيانات/التحليلات:**

* تشغيل **GA4 → BigQuery Export** لكل البيئات، وبناء طبقة **Curated** (Facts/Dimensions).
* **Scheduled Queries** لملء Cohorts/LTV/Delivered-COD وجداول الجماهير اليومية.
* فحوصات جودة تلقائية: فجوات الأيام، ازدواج transaction\_id، نسبة purchase بلا items[].
* استيراد **Ad Spend** + ربط بالـ campaign\_id لإصدار **ROAS/CAC**.

**د) التسويق/النمو:**

* اعتماد **UTM & Naming Policy** + أداة Link Builder.
* استيراد التحويلات إلى Google Ads (purchase/begin\_checkout/add\_payment\_info) وتفعيل **DDA**.
* إنشاء جماهير أساسية في GA4 وربطها مع FCM/RC/Ads.

**هـ) اللوحات/التنبيهات:**

* Looker Studio (أو أداة BI) لـ Funnel/Revenue/ROAS/Quality/Push Impact.
* تنبيهات Guardrails: هبوط Purchase>3%، فجوة GA4↔العمليات>3%، ارتفاع Refund، تدهور p95.

**و) Definition of Done – Analytics:**

* أحداث GA4 كاملة وصحيحة في DebugView، و**purchase** يطابق التشغيل ضمن ≤ 3%.
* Conversions مفعّلة ومستوردة للإعلانات، وUTM+campaign\_id على جميع الروابط.
* BigQuery خام+Curated تعمل يوميًا، ولوحات KPIs حية.
* جماهير أساسية فعّالة ومتصلة بـ FCM/RC، مع قياس أثرها.
* فحوصات جودة وتنبيهات نشطة، وتقارير ROAS/CAC تفصل COD Delivered عن Prepaid.

علامة التبويب 2

**النقاط الرئيسية للـ Analytics** التي نبني عليها التحليل للتطبيق:

1. الأهداف وKPIs

* North Star: صافي الإيراد/الطلبات المكتملة.
* KPIs أساسية: GMV، صافي الإيراد، عدد الطلبات، AOV، CR (تحويل الجلسات لطلبات)، CAC، ROAS، Retention D1/D7/D30، Repeat Rate، Return/Refund Rate، COD Success Rate، RTO/NDR Rate، Crash Rate، App Load Time.

1. خطة القياس (Measurement Plan)

* ما الذي نقيسه ولماذا: خرائط للأحداث، المعلمات (parameters)، والخصائص (user properties).
* تعريفات دقيقة وصيغ حساب لكل مؤشر.

1. بنية التتبع والتقنيات

* GA4/Firebase Analytics (iOS/Android)، الربط مع BigQuery.
* بيئات Dev/Staging/Prod، وإصدار المخطط (Schema Versioning) وQA قبل الإطلاق.

1. الاكتساب ونَسَب التحويل (Attribution)

* تتبع UTM، قنوات الإعلانات (Meta/Google/ASA).
* iOS SKAdNetwork، أو MMP (Appsflyer/Adjust) إن لزم، وتتبع التثبيت/فك التثبيت/إعادة التثبيت.

1. سلوك التصفح والاكتشاف

* شاشات وتفاعلات: impressions، clicks، view\_item/list، scroll depth.
* تحليلات البحث داخل التطبيق: استعلامات شائعة، Zero-Results، CTR من نتائج البحث.

1. التجارة الإلكترونية ومسار الشراء

* أحداث قياسية: view\_item → add\_to\_cart → begin\_checkout → add\_shipping\_info → add\_payment\_info → purchase.
* تتبع الفلاتر/الفرز، توصيات، عرض المقاسات والتوافر وتأثيرها على الإضافة للسلة.

1. الدفع عند الاستلام واللوجستيات (حاسم لسوقكم)

* تتبّع حالات الطلب بعد الشراء: Confirmed, Packed, Shipped, Out for Delivery, Delivered, Paid (COD)، Returned، Canceled.
* مؤشرات: COD Success، RTO، NDR، محاولات التسليم، زمن التوصيل، أسباب الإلغاء/عدم التسليم.

1. العروض والتسعير

* تتبع القسائم، الخصومات، حملات الشحن المجاني، وElasticity للسعر (تغيّر التحويل مع تغيّر السعر).

1. الاحتفاظ والولاء

* Cohorts حسب القناة/الفئة/الولاية، تكرار الشراء، CLV/LTV، RFM Segmentation.
* تتبع الإشعارات: opt-in، الإرسال، الفتح، النقر، التأثير على الشراء (Notification → Purchase).

1. الجودة والأداء التقني

* Crashlytics، Performance Monitoring (وقت الإطلاق، زمن عرض القوائم/الصور، طلبات الـAPI الفاشلة).
* تأثير الأداء على التحويل (مثلاً: كل +1s في التحميل مقابل −CR).

1. خدمة العملاء وتجربة ما بعد الشراء

* تتبع تواصل الدعم، سياسات الإرجاع/الاستبدال، زمن الحل، ورضا العميل (CSAT/NPS داخل التطبيق إن وُجد).

1. البيانات الموحّدة والتكاملات

* BigQuery كـ Single Source of Truth.
* ربط بيانات الطلب والتسليم من النظام الخلفي (POS/OMS) عبر Measurement Protocol/ETL.
* ربط CRM/CDP، وقنوات الإعلانات عبر CAPI/Conversions API.

1. لوحات المعلومات والتقارير

* تنفيذي (يومي/أسبوعي): الإيرادات، الطلبات، CR، CAC/ROAS، COD Success، RTO.
* تسويق: قنوات/حملات، تكلفة لكل تثبيت/طلب، تكرار الشراء حسب القناة.
* منتج: مسارات، Drop-offs، بحث، أداء الفئات والمقاسات.
* عمليات: SLA الشحن، NDR/RTO حسب المنطقة وشركة التوصيل.

1. الاختبارات والتحسين (Experimentation)

* A/B عبر Firebase Experiments/Remote Config.
* أهداف التجربة، مدة، حجم العيّنة، مؤشرات رئيسية وحواجز (Guardrails) مثل Crash/Latency.

1. الحوكمة والخصوصية

* Naming Conventions، توثيق، مراجعات دورية للبيانات، أذونات الوصول.
* موافقات الخصوصية (Consent) وتخزين/أمان البيانات بما يتوافق مع المتطلبات المحلية.

1. التنبيهات والمراقبة

* تنبيهات تلقائية للانحرافات: هبوط مفاجئ في CR/ROAS، ارتفاع RTO، زيادة Crash Rate، تعطّل التتبّع.

1. جدول مخرجات واضح (Deliverables)

* وثيقة خطة القياس.
* جدول الأحداث والمعلمات والخصائص.
* مخطط البيانات والتدفقات (SDK → GA4 → BigQuery → Dashboards).
* قائمة اللوحات والتقارير، وقائمة التنبيهات.
* خطة QA والاختبارات قبل كل إصدار.

علامة التبويب 4

**النقاط الرئيسية لـ Google Analytics (GA4)** بنسخة مختصرة:

1. هيكلة الـ Property & Data Streams

* Property واحدة.
* App Streams: iOS + Android (مرتبطة بمشروع Firebase).
* (اختياري) Web Stream إذا كان عندك موقع/لاندنغ.

1. خطة القياس (Measurement Plan)

* قائمة الأحداث القياسية + المخصّصة.
* المعلمات (parameters) والخصائص (user properties) وتعريفات كل مؤشر.

1. الأحداث القياسية للتجارة الإلكترونية

* من view\_item\_list → view\_item → add\_to\_cart → begin\_checkout → add\_shipping\_info → add\_payment\_info → purchase.
* مصفوفة items[] بمعايير GA4 (item\_id, item\_name, price, quantity, item\_category…).

1. أحداث/معلمات مخصّصة لسوقك (COD واللوجستيات)

* معلمات على مراحل الدفع/الشراء: payment\_type (COD/Prepaid)، delivery\_city, delivery\_partner, size, color.
* إرسال **حالة الطلب بعد الشراء** (Confirmed/Delivered/Returned/Cancelled) عبر **Measurement Protocol** للسيرفر.

1. التحويلات (Conversions)

* تمييز الأحداث: purchase، و(اختياريًا) begin\_checkout, add\_payment\_info, add\_shipping\_info.
* تعريف نوافذ التحويل ومطابقة القنوات.

1. الخصائص (User Properties)

* أمثلة: المحافظة/المدينة، تفضيل الدفع (COD/Prepaid)، شريحة الولاء، اهتمامات الفئات، مقاس مفضل.

1. الاكتساب ونَسَب التحويل (Attribution)

* تفعيل **Data-driven attribution**.
* حوكمة UTM، وربط Google Ads (استيراد التحويلات والجماهير).

1. الربط والتكاملات

* Link مع: Google Ads، BigQuery Export، (اختياري) Search Console للويب.
* استيراد تكلفة القنوات الخارجية (Meta/ASA) إلى GA4 أو إلى لوحات Looker Studio عبر موصلات.

1. الخصوصية والامتثال

* **Consent Mode v2** (إن لزم)، Google Signals (مع ضبط الاحتفاظ بالبيانات).
* تصفية الزيارات الداخلية/الاختبارية وتفعيل DebugView للحِسبة.

1. الجماهير (Audiences) والريمَركتِنغ

* جماهير تشغيلية: “Added to cart ولم يشترِ”، “High AOV”، “مهتم بفئة فساتين السهرة”، “مستخدمو COD الناجحون”.
* تفعيل Audience Triggers (إطلاق حدث عند دخول الجمهور).

1. الاستكشافات (Explorations)

* Funnels لمسار الشراء، مسارات الأحداث، Cohorts للاحتفاظ، تحليل البحث داخل التطبيق (Zero-results، CTR).

1. لوحات وتقارير

* تنفيذية يومية: الإيراد/الطلبات/CR/AOV/CAC/ROAS.
* قنوات: أداء الحملات والقنوات.
* منتج: أداء الفئات/المقاسات/البحث.
* عمليات: COD Success, RTO/NDR حسب المدينة وشركة التوصيل (يبنى من BigQuery).

1. BigQuery كمنبع البيانات

* Export يومي و(اختياري) Streaming.
* نمذجة جداول الطلبات/الشحن (من النظام الخلفي) + ربطها ببيانات GA4.

1. التنبيهات (Custom Insights)

* تنبيهات لهبوط CR أو ROAS، ارتفاع RTO، اختفاء حدث رئيسي، ارتفاع Crash/Latency (يُستكمل بتنبيهات Firebase).

1. ضمان الجودة (QA) والحَوْكمة

* Naming Conventions، وثائق الجدول الزمني للأحداث، إصدارات المخطط (Schema v1/v2)، قائمة فحوص قبل الإصدار.
* صلاحيات الوصول وتقادم البيانات/الاحتفاظ.

1. الاختبارات والتحسين

* تصميم A/B عبر Firebase + قراءة النتائج في GA4/BigQuery.
* Guardrails (لا تتجاوز Crash/Latency) ومدة وحجم عينة واضحين.

العمل بين firebase & analytics

## **الفكرة الأساسية**

* **Firebase Analytics** (اسمه داخل فيربيز: *Google Analytics for Firebase*) هو **طبقة الجمع والتطبيق** الموجهة للموبايل (SDKs + أدوات المطوّر).
* **Google Analytics (GA4)** هو **طبقة التقارير والتسويق** عبر الويب والتطبيق معًا في واجهة GA4.

كلاهما يكتبان إلى **نفس الـ GA4 Property** إذا ربطت مشروع Firebase بالـ GA4؛ ما في “ازدواجية” بيانات.

## **مقارنة سريعة**

* **التركيز**
  + Firebase: تطبيقات الموبايل (iOS/Android)، مع أدوات مطوّر (Crashlytics, Performance, Remote Config, A/B Testing, In-App Messaging).
  + GA4: تقارير تسويقية وتنفيذيّة عبر **App + Web**، قنوات الاكتساب، الاستكشافات المتقدمة، ونسب التحويل عبر القنوات.
* **التجميع (SDKs)**
  + Firebase SDK للموبايل هو **الموصى به**. للويب تستخدم gtag/GA4.
* **التقارير**
  + Firebase Console: تقارير مختصرة للموبايل، DebugView، تتبع الأحداث الجاهزة مثل first\_open، جلسات، إلخ.
  + GA4: Explorations، Attribution، مسارات التحويل متعددة القنوات، الجمهور (Audiences) والتصدير لـ Ads.
* **الاكتساب/الإسناد**
  + Firebase: عرض أساسي لاكتساب المستخدمين.
  + GA4: **نِسب تحويل متقدمة** (data-driven attribution)، مسارات، تكامل أعمق مع الحملات.
* **البيانات الخام**
  + كِلاهما يدعمان ربط **BigQuery** لنفس الـ Property. (التخزين والاستعلامات على BigQuery قد تكون لها تكاليف حسب الاستخدام).
* **الخصوصية والحَوْكمة**
  + الإعدادات (الاحتفاظ بالبيانات، consent، الأذونات) تُدار على مستوى الـ GA4 Property وتشمل التطبيقين.
* **التجارب والتشغيل**
  + Firebase يوفّر Remote Config + A/B Testing على مستوى التطبيق (تبديل واجهة/ميزات حيّة)، وهو غير موجود في GA4.

## **ماذا تختار لتطبيق؟**

* **اِجمع** بيانات التطبيق عبر **Firebase SDK** (Analytics + Crashlytics + Performance).
* **اربط** مشروع Firebase بــ **GA4 Property واحدة**:
  + تشاهد تقارير المطور في Firebase، وتقارير التسويق والاكتساب في GA4—لنفس البيانات.
* **اعمل تحليلات وتسويق** في GA4 (القنوات، ROAS، المسارات)،  
  و**شغّل التجارب والتنبيهات التقنية** من Firebase (A/B، Remote Config، Performance).
* **خُذ البيانات الخام** إلى **BigQuery** لبناء لوحات تنفيذية مخصّصة (COD/RTO، المقاسات، سلوك البحث…).

### **أسئلة سريعة**

* **هل نحتاج الاثنين؟** عمليًا نعم—تفعّل الربط فيصبحا وجهين لنفس الـ Property.
* **هل تختلف الأحداث/الأسماء؟** نفس *نموذج الأحداث* في الاثنين؛ المهم توحيد التسمية والـparameters.
* **أين أتابع COD/RTO؟** عبر Events مخصّصة + تصدير إلى BigQuery ولوحات مخصّصة (GA4 لا يملك تقارير COD جاهزة).

\*معايير قاعدة البيانات ومنع الأخطاء

# **Zahraah – معايير قاعدة البيانات ومنع الأخطاء (النسخة التنفيذية)**

**هدف الملف**: أن يكون مرجعًا تنفيذيًا واحدًا يمنع الوقوع في الأخطاء عبر كامل دورة حياة قاعدة البيانات (تصميم → بناء → تشغيل → تحليلات → أمان → نسخ/تعافٍ). **النطاق**: MySQL 8.x + InnoDB، مع Redis (Cache)، OpenSearch (بحث)، BigQuery/Metabase (تحليل)، وخدمات دفع/شحن خارجية.

## **0) ملخّص آلية العمل والاعتماديات (كيف تتكامل المنظومة)**

* **قاعدة البيانات التشغيلية (OLTP)** هي *مصدر الحقيقة* لكل معاملات المتجر (عملاء، كتالوج، عربة، طلبات، دفع، شحن…)، وتُستهلك عبر **API/خدمات خلفية**.
* **Cache /Redis**: لتسريع القراءة والجلسات وRate-Limiting، مع سياسة إبطال دقيقة (Cache-aside) – الحقيقة تبقى في DB.
* **Search (OpenSearch/Elasticsearch**: لفهرسة المنتجات والبحث النصّي/المرادفات، مع **مزامنة تدريجية** من DB.
* **Object Storage**: صور وملفات خارج DB (يُخزَّن في DB المسار والميتا فقط).
* **تحليلات/BI**: أحداث تطبيق إلى GA4/Firebase؛ و**CDC/ETL** من MySQL → BigQuery؛ وMetabase/Looker للقراءة من BigQuery/Replica فقط.
* **أمن/حماية**: طبقة أذونات دقيقة، أسرار عبر Secret Manager، اتصال TLS، Logging/Audit، نسخ احتياطي + تعافٍ.

أي خلل في طبقة (سكيما/استعلام/كاش/بحث/تحليل/أمان) ينعكس على بقية الطبقات. هذا الملف يضع معايير وقوائم تدقيق لمنع الخطأ قبل ظهوره.

## **1) معايير تصميم السكيما (Schema Standards)**

**الترميز والتوقيت**

* utf8mb4 + Collation ملائم للعربية (مثل: utf8mb4\_0900\_ai\_ci).
* كل الأوقات بـ**UTC** في DB؛ التحويل للمناطق الزمنية يتم في الواجهة.

**الأنواع (Types)**

* الأموال: DECIMAL(12,2) (أو حسب الحاجة) – **ممنوع** FLOAT/DOUBLE للأموال.
* الهاتف/الإيميل/المعرفات الخارجية: VARCHAR مع UNIQUE حيث يلزم.
* JSON للاحتياجات *الاختيارية/المرنة* فقط، مع حقول أساسية منظّمة في أعمدة طبيعية.
* **ممنوع** تخزين الصور/الملفات داخل DB (BLOB)؛ خزّن **المسارات** وميتا البيانات فقط.

**المفاتيح الأساسية والأولية**

* جداول OLTP الحرجة: BIGINT UNSIGNED AUTO\_INCREMENT كمفتاح أساسي.
* ULID/UUIDv7 عند الحاجة لتكامل بين أنظمة (تجنّب UUID v4 كمفتاح أساسي في InnoDB).

**القيود والمرجعيات**

* **FK إلزامية** مع أفعال ON UPDATE/DELETE مناسبة.
* **Lookup tables بدل ENUM** للقيم المتغيّرة (حالات الطلب/الشحن/الدفع…).
* UNIQUE حيث تتطلب الأعمال (الإيميل، باركود داخلي، رقم الطلب…).

**معايير الحقول**

* أعمدة نظامية لكل جدول: id, status, created\_at, updated\_at, created\_by, updated\_by, deleted\_at NULL.
* التسمية: snake\_case؛ المفاتيح الخارجية تنتهي بـ\_id؛ تواريخ بـ\*\_at.
* **Soft-delete** عبر deleted\_at + فهارس/Views تراعي الحذف.
* **Partitioning** للجداول الضخمة الزمنية (logs/notifications) مع سياسة دوران/أرشفة.

**فصل OLTP عن OLAP**

* السكيما التشغيلية للمعاملات فقط؛ التقارير الثقيلة والـBI في BigQuery/Replica.

## **2) نموذج كيانات التجارة الإلكترونية (مختصر تنفيذي)**

**الهوية/العملاء**: users, user\_devices, addresses, consents, segments\_membership.

**الكتالوج والمخزون**: categories, brands, products (SPU), attributes/values, skus (SKU), inventory, media, price\_lists, sku\_prices.

**العربة/الطلبات/الدفع/الشحن**: carts, cart\_items, orders, order\_items, payments, refunds, returns/return\_items, shipments.

**الخصومات والكوبونات**: promotions, promotion\_rules, promotion\_actions, coupons, coupon\_redemptions.

**التفاعل/الولاء**: wishlists, wishlist\_items, reviews, loyalty\_ledger, notifications\_log.

**الإسناد التسويقي**: attributions أو جداول utm\_first\_touch وutm\_last\_touch.

جميع الجداول أعلاه تلتزم بمعايير النوع/المفاتيح/القيود المذكورة في §1.

## **3) الفهارس والاستعلامات (Indexing & Query Patterns)**

* فهرس لكل FK، وفهارس مركّبة لأكثر WHERE/ORDER BY شيوعًا.
* ترتيب أعمدة الفهرس: من الأكثر انتقائية للأقل، ومطابقة ترتيب الفرز إن أمكن.
* **Covering Index** للاستعلامات الحرجة القراءة.
* **ممنوع** SELECT \*؛ حدّد الأعمدة المطلوبة.
* **ممنوع** الدوال على الأعمدة المُفلترة (DATE(col)=…)، استخدم نطاقات (col BETWEEN …).
* **LIKE '%term'** يمنع الفهرس؛ استخدم محرك البحث للنصوص.
* Pagination كبيرة: **Keyset pagination** بدل OFFSET/LIMIT.
* معالجة **N+1** عبر JOIN/IN/Batch.
* راجع كل استعلام حرِج بـEXPLAIN قبل الإنتاج.

**أنماط فهارس مقترحة**

* منتجات حسب القسم/الظهور: (category\_id, is\_active, sort\_order).
* طلبات حسب العميل/التاريخ: (user\_id, created\_at DESC) + فهرس على (created\_at) للتقارير.
* عناصر الطلب: (order\_id) و(sku\_id).
* مخزون: (sku\_id), وفهارس لمخزون المستودعات إن تعددت.
* كوبونات: code UNIQUE, واستعلامات coupon\_redemptions(user\_id, coupon\_id).

## **4) المعاملات والقفل (Transactions & Locking)**

* معاملات قصيرة، هدف واحد لكل معاملة.
* ترتيب ثابت للعمليات الحسّاسة (حجز مخزون → حساب السعر/الخصم → الدفع → تأكيد الطلب).
* استخدم SELECT … FOR UPDATE عند الحجز/التناقص لمنع Oversell.
* **Idempotency** لكل مسارات الدفع/الطلب عبر idempotency\_key.
* انتبه لقراءات الـReplica (Replication Lag)؛ لمسارات “اقرأ بعد كتابة” استخدم الأساس/آليات توكيد.

## **5) الطبقات المساندة (Cache/Search/Storage/Async)**

* **Cache-aside** مع مفاتيح محدّدة وTTL وسياسة إبطال عند الكتابة/التغيير.
* **Search** بمزامنة تدريجية (Change Data Capture/Events) لا إعادة فهرسة كاملة إلا لضرورة.
* **Object Storage** للصور/الملفات؛ في DB: المسار/الأبعاد/النوع/الترتيب فقط.
* **مهام ثقيلة** (فواتير/PDF/تجميع تقارير) تُنقَل لخدمات غير متزامنة.

## **6) الأمن والامتثال (Security & Compliance)**

* **أقلّ صلاحية**: مستخدمون منفصلون (admin محدود، app\_rw، app\_ro، analyst\_ro).
* اتصال **TLS**، وولوج الإنتاج عبر VPN/Bastion فقط.
* الأسرار عبر **Secret Manager**؛ ممنوع تخزينها في الكود/الريبو.
* **Prepared Statements** دائمًا لمنع حقن SQL.
* تصنيف **PII** (هاتف/إيميل/عنوان) وتقليل الوصول؛ Masking في أدوات التحليل.
* **Audit** لتغييرات حسّاسة (الأسعار/الخصومات/النقاط/الموافقات).
* حماية أي أدوات إدارة (مثل phpMyAdmin) خلف VPN و2FA وقيود IP، وتعطيل العمليات الخطرة للمستخدمين غير الإداريين.

## **7) النسخ الاحتياطي والتعافي (Backup & DR)**

* **نسخ كامل يومي** + **Binlogs** لـ7–14 يومًا (PITR).
* اختبار **استعادة ربع سنوي** موثّق + تمارين Failover نصف سنوية.
* تخزين النسخ في موقع منفصل/سحابة مختلفة (Offsite).
* تعريف واقعي لـ**RPO/RTO** ومراجعة سنوية.

## **8) المراقبة والتشغيل (Observability & Ops)**

* فعّل **Slow Query Log**, Performance Schema.
* لوحات مراقبة: QPS, p95/p99 latency, error rate, buffer pool hit ratio, deadlocks, replication lag, connections, table growth.
* تنبيهات على العتبات، ومسار Incident واضح (On-call/Runbook).
* جدولة ANALYZE TABLE الدوري للجداول الكبيرة لتحديث الإحصاءات.

## **9) الترقيات والترحيلات (Migrations)**

* كل تغييرات الـSchema/البيانات عبر **Flyway/Liquibase** فقط.
* **Zero-downtime pattern**:
  1. إضافة حقول جديدة اختيارية.
  2. تعبئة خلفية على دفعات.
  3. تحديث التطبيق للقراءة/الكتابة الجديدة.
  4. إزالة القديم لاحقًا.
* **Online DDL** للجداول الكبيرة + نافذة صيانة مخططة عند الحاجة.
* سياسة "أربع عيون" لمراجعة أي ALTER/DELETE/UPDATE واسع.

## **10) الجودة والاختبارات (Quality & Testing)**

* اختبارات وحدة لطبقة الوصول للبيانات، وتكامل على Staging ببيانات قريبة من الإنتاج (مموّهة).
* اختبارات **dbt/SQL**: NOT NULL, UNIQUE, FK, نطاقات القيم.
* **Test Cases حرِجة**:
  + أموال (تقريب/ضرائب/خصومات)،
  + قفل المخزون ومنع Oversell،
  + Idempotency للدفع،
  + ترميز عربي/رموز/إيموجي،
  + مناطق زمنية،
  + تكرار الإيميل/الهاتف،
  + سلوك N+1،
  + قراءة بعد كتابة مع Replica.

## **11) الاحتفاظ/الأرشفة/الخصوصية (Retention & Archiving)**

* سياسات احتفاظ لكل كيان (مثال: إشعارات 6–12 شهرًا، شحن 24 شهرًا…).
* عمليات **Purging/Archiving** مجدولة إلى جداول أرشيف/مخزن بارد.
* مسارات **حذف/إخفاء** بيانات العميل عند الطلب (حق النسيان/الامتثال).

## **12) التكامل التحليلي (MySQL → BigQuery/GA4/Metabase)**

* **CDC** (Debezium/Maxwell أو ETL مُدار) للجداول التشغيلية إلى BigQuery.
* نموذج نجمي في BI: fact\_orders, fact\_order\_items, dim\_customers, dim\_products, dim\_sku, dim\_date, dim\_channel…
* إخفاء/تجزئة PII قبل التحليل؛ عدم قراءة أدوات التحليل من إنتاج مباشرة.
* أحداث GA4/Firebase تُربَط بـuser\_id وUTM (أول/آخر) داخل BigQuery.

## **13) مكافحة الأخطاء الخاصة بالتجارة الإلكترونية**

* **Oversell**: الحجز/التناقص داخل معاملة مقفلة + ترتيب ثابت + FOR UPDATE.
* **الكوبونات/الخصومات**:
  + تثبيت (snapshot) قيمة الخصم عند إنشاء الطلب؛
  + UNIQUE على code؛
  + سجل coupon\_redemptions بمنع التكرار حسب القواعد.
* **السعر النهائي**: تخزين snapshot للسعر والضرائب والخصومات في order\_items.
* **COD Risk**: جداول سلوك COD (محاولات فاشلة/عناوين مكررة/حدود)، مع سياسات حظر ذكية.
* **الإسناد التسويقي**: حفظ UTM أول/آخر على مستوى الطلب/المستخدم وقت الحدث.
* **العربية والبحث**: استخدام محرك بحث مع محلّل عربي ومرادفات؛ لا تعتمد على LIKE فقط.

## **14) إدارة الأداء والكلفة**

* مراجعة فهارس ربع سنوية، وتنظيف الفهارس غير المستخدمة.
* مراجعة الاستعلامات الأكثر بطئًا شهريًا وتحسينها.
* ضبط Connection Pool من التطبيق، وmax\_allowed\_packet حسب الأحجام.
* تجنّب OPTIMIZE TABLE العشوائي؛ استخدمه فقط عند وجود تجزئة حقيقية وبعد ساعات الذروة.
* مراقبة نمو الجداول ووضع حدود/تنبيهات مبكرة.

## **15) قوائم تدقيق (Checklists جاهزة)**

**A) إنشاء قاعدة بيانات إنتاج جديدة**

1. MySQL 8/InnoDB + utf8mb4 + utf8mb4\_0900\_ai\_ci + UTC.
2. شبكات: Private Subnet + TLS + قيود IP + VPN/Bastion.
3. مستخدمون/أدوار: admin (مقفل)، app\_rw, app\_ro, analyst\_ro.
4. Slow Query Log + Performance Schema مفعّلان.
5. نسخ كامل يومي + Binlogs (7–14 يومًا) + اختبار استعادة.
6. إعداد Migrations (Flyway/Liquibase) + سياسة مراجعة.
7. Replica للقراءة + خطة Failover موثّقة.
8. مراقبة/لوحات/تنبيهات + Runbooks.
9. أسرار عبر Secret Manager؛ ممنوع في الكود.
10. حماية أدوات الإدارة (phpMyAdmin إن وُجد) خلف VPN/2FA.

**B) قبل أي إطلاق/ترقية**

1. Migration مُراجَع + خطة رجوع.
2. EXPLAIN للاستعلامات الحرجة + فهارس مطابقة.
3. معاملات قصيرة + اختبارات سباق/قفل.
4. بيانات Staging قريبة من الإنتاج (مموّهة) + اختبارات قبول.
5. مراقبة لصيقة بعد الإطلاق (p95/p99, deadlocks, lag).

**C) التعامل مع حادث أداء**

1. فحص الموارد/الاتصالات/النسخ/التأخير.
2. استخراج Top slow queries + EXPLAIN.
3. قرارات: فهرس/إعادة كتابة/Cache/Replica/تقسيم.
4. توثيق السبب الجذري + متابعة بعد 7 أيام.

**D) تغييرات سكيما دون توقف**

1. أضف الجديد كاختياري.
2. تعبئة خلفية على دفعات.
3. شغّل القراءة/الكتابة مزدوجين.
4. أزل القديم لاحقًا.
5. Online DDL للجداول الكبيرة.

**E) أمن الوصول**

1. أقلّ صلاحية + فصل حسابات DDL عن RW.
2. 2FA + تدوير أسرار دوري.
3. تدقيق (Audit) على العمليات الحساسة.
4. اختبارات اختراق نصف سنوية على طبقة DB.

**F) نسخ/تعافٍ (DR Drill)**

1. استعادة نسخة إلى بيئة معزولة كل ربع.
2. قياس زمن الاستعادة ومقارنته بـRTO.
3. تحقق من سلامة البيانات والفهارس.
4. توثيق مُحدَّث لخطوات الاستعادة.

**G) جودة بيانات شهرية**

1. dbt tests (NOT NULL/UNIQUE/FK).
2. تقارير مفاتيح مكسورة/قيم خارج النطاق.
3. مراجعة نمو الجداول ومعدلات التكرار.
4. خطة تنظيف/أرشفة.

## **16) ملحقات سريعة (Cheat Sheets)**

**تسمية**: جداول جمع snake\_case (مثل order\_items)، أعمدة snake\_case، FK تنتهي بـ\_id، تواريخ \*\_at، حالة status.

**أعمدة قياسية**: id, status, created\_at, updated\_at, created\_by, updated\_by, deleted\_at, وبالجداول المعروضة slug, is\_active, sort\_order.

**متى أستخدم JSON؟** عندما تكون البيانات اختيارية/متغيرة جدًا ولا تؤثر على استعلامات حرجة. وإلا: نمذجة طبيعية + فهارس.

**متى أجزّئ (Partition)؟** جداول زمنية ضخمة (logs/notifications) + حاجة لأرشفة/تنظيف سريع.

**متى أستخدم Replica؟** تقارير/قراءات كثيفة غير حرجة للاتساق الفوري؛ تجنّب “اقرأ بعد كتابة” عليها لمسارات حسّاسة.

## **17) قائمة Anti‑Patterns (تجنّبها دائمًا)**

* SELECT \*، وLIKE مع بادئة %، وORDER BY RAND() على جداول كبيرة.
* JSON/EAV بدل نمذجة سليمة لجداول أساسية.
* ENUM لقيم متغيرة، وأعمدة نصية ضخمة بلا حاجة.
* تغييرات DDL مباشرة على الإنتاج، أو بدون خطة رجوع.
* صور/ملفات داخل DB، أو تشغيل تقارير ثقيلة على إنتاج.
* حساب root للجميع، وأسرار داخل الكود، وغياب TLS/VPN.
* نسخ بلا اختبارات استعادة، وغياب PITR.
* تجاهل p95/p99, deadlocks, replication lag في المراقبة.

### **ختام عملي**

* التزم بهذه المعايير والقوائم وستُغطي >95% من فئات الأخطاء الشائعة.
* أي استثناء يجب أن يُوثَّق بمبررات ومخاطر مخفَّفة.

هيكلة الـProperty & Streams (GA4)

## **#1) هيكلة الـProperty & Streams) GA4)**

**الاستفادة (Why it matters)**

* توحيد بيانات iOS/Android (وويب إن وُجد) داخل مصدر واحد دقيق.
* تقارير متّسقة للشراء والقنوات بدون تلوّث أو ازدواجية.
* جاهزية للتكاملات (Firebase، BigQuery، Google Ads، Metabase، Pixel).
* احترام المنطقة الزمنية والعملة لتطابق الإيرادات.
* فصل واضح بين بيانات الإنتاج والاختبار.
* أساس ثابت لأي تحسينات لاحقة (Audiences، Attribution، لوحات).

**الهدف (What we want to achieve)**

* إنشاء Property واحدة مضبوطة على Asia/Aden وعملة YER.
* تفعيل App Data Streams لـ iOS وAndroid (وWeb إن وُجد) والربط بمشروع Firebase نفسه.
* تعريف التحويلات الأساسية (خصوصًا purchase).
* تفعيل الفلاتر للزيارات الداخلية/أجهزة التطوير.
* توحيد هوية المستخدم (user\_id عند تسجيل الدخول).

**التفاصيل (How to do it — مختصر)**

* إعداد الـProperty: Google Signals، Data Retention 14 شهر، Demographics إن سمحت اللوائح.
* إنشاء Streams: iOS (Bundle ID)، Android (Package + SHA-1)، (اختياري) Web Stream.
* ربط Firebase: تشغيل Crashlytics/Performance/Remote Config على نفس الـProperty.
* بيئات العمل: Prod منفصلة عن Staging (أو وسم environment=staging + فلتر).
* الهوية: إرسال user\_id عند تسجيل الدخول، والاعتماد على app\_instance\_id لغير المسجّلين.
* التحويلات: وسم purchase (وأحداث وسيطة إن احتجت).
* الفلاتر: Internal/Developer traffic مفعّلة؛ Bot filtering افتراضي.
* التعاريف المخصّصة: تسجيل أهم الـparameters (مثل payment\_type, delivery\_city, size, color) كـ Custom Dimensions/metrics ضمن حدود GA4.

## **#2) خطة القياس (Measurement Plan)**

**الاستفادة (Why it matters)**

* يضمن أن “ما نرسله” يجيب بدقّة عن أسئلة العمل (تحويل، AOV، COD/RTO…)، بدون فجوات أو ضوضاء.
* يوفّر لغة مشتركة بين المنتج/المطور/التسويق ويُسهّل الـQA.
* يقلّل أخطاء التسمية والازدواجية، ويُسرّع بناء اللوحات والاختبارات.

**الهدف (What we want to achieve)**

* توثيق “ماذا ولماذا ومتى وكيف” لكل حدث/معامل/خاصية.
* تحديد معايير الحساب لكل مؤشر (CR, AOV, ROAS…) وتعريف مصدر الحقيقة.
* تحديد أحداث التطبيق القياسية للتجارة الإلكترونية + أحداث مخصّصة لسوقك (COD واللوجستيات).
* ضبط التحويلات، والـUser Properties المسموح بها، وقاموس UTM.

**التفاصيل (How to do it — مختصر)**

* **قائمة الأحداث القياسية (GA4 E-commerce):**view\_item\_list → select\_item → view\_item → add\_to\_cart → begin\_checkout → add\_shipping\_info → add\_payment\_info → purchase  
  مع مصفوفة items[] (لازم: item\_id, item\_name, price, quantity, item\_category…).
* **أحداث/معلمات مخصّصة لسوقك:**payment\_type (COD/Prepaid)، delivery\_city/zone/partner، size, color, stock\_status, coupon, discount\_value.  
  (احفظ الأسماء بصيغة snake\_case وبالإنجليزية؛ القيم يمكن أن تكون عربية).
* **User Properties (اختَر 3–5):**preferred\_payment, favorite\_size, favorite\_category, loyalty\_tier.
* **Conversions:**وسم purchase أساسًا (+ begin\_checkout/add\_payment\_info/add\_shipping\_info إن أردت تحليلًا أدق).
* **Measurement Protocol (Server):**حدث order\_status\_update لِما بعد الشراء: order\_id, order\_status (Confirmed/Delivered/Returned/…)، payment\_collected, cod\_attempts, delivery\_partner.
* **صيغ الحساب/التعريفات:**
  + CR = الطلبات ÷ الجلسات (عرّف بدقة أي “جلسات”).
  + AOV = الإيراد ÷ عدد الطلبات (استبعد الملغى/المرتجع حسب سياسة التقرير).
  + COD Success = Delivered & Paid COD ÷ جميع طلبات COD.
  + RTO Rate = Returned to Origin ÷ شحنات COD.  
    (دوّن الصيغ في نفس وثيقة الخطة).
* **حوكمة الأسماء والـUTM:**دليل تسمية للأحداث والمعاملات + جدول قيم مسموحة لـ utm\_source/medium/campaign/....
* **QA لكل حدث:**Trigger واضح، معلمات إلزامية/اختيارية، مثال Payload، وخطوات اختبار في DebugView.

أحداث التجارة الإلكترونية القياسية (GA4)

## **#3) أحداث التجارة الإلكترونية القياسية (GA4)**

**الاستفادة (Why it matters)**

* ترسم مسار الشراء من أول ظهور المنتج حتى الدفع، فتعرّف أين يحصل الهدر (Drop-offs) وتُحسّن الـCR و الـAOV.
* تجعل الإيراد والطلبات قابلة للقياس بدقّة، وتغذّي التحويلات للإعلانات والجماهير للريمَركتِنغ.
* توحِّد تحليل أداء القوائم والفئات والمقاسات والألوان.

**الهدف (What we want to achieve)**

* تطبيق سلسلة أحداث GA4 القياسية مع مصفوفة items[] الصحيحة.
* توحيد المعلمات الإلزامية والمستحسنة لضمان تقارير سليمة (سعر/عملة/كمية/فئة…).
* مطابقة قيمة الطلب مع مجموع العناصر + الشحن/الضريبة بدون ازدواجية.

**التفاصيل (How to do it — مختصر ومنظّم)**

القاعدة الذهبية لكل حدث:

* أرسل currency (مثال: YER) و**سعّر** بدقّة.
* املأ items[] لكل حدث له علاقة بالمنتجات.
* ثبّت أسماء المعلمات بصيغة snake\_case.

### **الحدث 1) view\_item\_list (عرض قائمة/شبكة منتجات)**

* **متى؟** عند ظهور قائمة (مثلاً “فساتين السهرة”).
* **Params مهمّة:**
  + item\_list\_id, item\_list\_name
  + items[] (اختياري لكنه مفيد): لكل عنصر أرسل على الأقل: item\_id, item\_name, item\_category, index (ترتيبه في القائمة)
* **نصائح:** احسب *الانطباعات* بدقّة (لا تُكرر مع كل Scroll إلا عند ظهور دفعة جديدة).

### **الحدث 2) select\_item (نقر عنصر من القائمة)**

* **متى؟** عند ضغط المستخدم على منتج داخل القائمة.
* **Params:**
  + item\_list\_id, item\_list\_name, items[0] للعنصر المختار فقط (+ index إن أمكن)

### **الحدث 3) view\_item (عرض صفحة المنتج)**

* **متى؟** عند فتح صفحة تفاصيل المنتج.
* **Params (ضمن items[0]):**
  + إلزامي: item\_id, item\_name, item\_category, price
  + مستحسن: item\_brand (إن لزم)، item\_variant، stock\_status (= in\_stock/out\_of\_stock)
  + إضافي مفيد: size\_options\_count
* **معلومة:** السعر هنا هو **سعر الوحدة الظاهر للعميل** بعد الخصم إن كان دائمًا ظاهرًا.

### **الحدث 4) add\_to\_cart**

* **متى؟** عند الإضافة للسلة (أو تحديث كمية).
* **Params:**
  + مستوى الحدث: currency, value *(اختياري)*
  + items[0]: item\_id, item\_name, price, quantity,  
    size, color, stock\_status,  
    (اختياري) discount (خصم على مستوى المنتج), coupon
* **قاعدة السعر:** price = **سعر الوحدة النهائي**، وvalue = price \* quantity (إن أرسلتَه).

### **الحدث 5) remove\_from\_cart *(اختياري لكنه مفيد)***

* **متى؟** عند إزالة عنصر من السلة.
* **Params:** items[0] بنفس هيكلة add\_to\_cart + quantity المُزالة.

### **الحدث 6) view\_cart *(اختياري)***

* **متى؟** عند فتح شاشة السلة.
* **Params:** currency, value (إجمالي السلة)، وitems[] كاملة.

### **الحدث 7) begin\_checkout**

* **متى؟** عند بدء إتمام الطلب.
* **Params:**
  + مستوى الحدث: currency, value, coupon (إن وُجد)
  + items[] (القائمة الحالية في السلة)
  + إضافي: shipping\_tier (اقتصادي/سريع)

### **الحدث 8) add\_shipping\_info**

* **متى؟** عند إدخال/اختيار معلومات الشحن.
* **Params:**
  + currency, value, shipping\_tier
  + delivery\_city, delivery\_zone
  + items[]

### **الحدث 9) add\_payment\_info**

* **متى؟** عند اختيار وسيلة الدفع.
* **Params:**
  + currency, value, payment\_type = COD أو Prepaid
  + items[]

### **الحدث 10) purchase**

* **متى؟** عند إنشاء الطلب بنجاح (تأكيد نهائي).
* **Params (مستوى الحدث):**
  + transaction\_id (فريد)، value, currency
  + coupon, shipping, tax, (اختياري) affiliation
  + (اختياري) payment\_type لإعادة توثيق نوع الدفع
* **Params (items[]):**
  + لكل عنصر: item\_id, item\_name, price, quantity,  
    (اختياري) discount, coupon, item\_category, item\_variant, item\_brand
* **تعريف القيمة:**
  + **value = صافي إجمالي الطلب** بعد الخصومات **+ الشحن + الضريبة** (وفق سياسة تقاريرك).
  + تأكّد أن: value ≈ sum(items.price\*qty) + shipping + tax − discounts.

### **عناصر إضافية مُهمّة**

* **الاسترجاع/الإلغاء:**
  + لمدفوعات *مسبقة* يمكن استخدام refund (جزئي/كامل).
  + في COD غالبًا لا تُسجِّل refund (لا يوجد تحصيل فعلي)، وبدلًا من ذلك نرسل حالة ما بعد الشراء عبر **Measurement Protocol** (تم تفصيله في نقطة لاحقة).
* **الدقة في القيم:**
  + لا تُرسل أسعارًا مختلفة بين الأحداث لنفس الـSKU خلال نفس الجلسة إلا لو تغيّر السعر أمام المستخدم.
  + أرسل currency دائمًا مع أحداث السلة/الدفع/الشراء.
* **المخزون والمقاس:**
  + أرسل stock\_status في view\_item وadd\_to\_cart لقياس أثر نفاد المخزون.
  + أرسل size/color في add\_to\_cart وما بعده.
* **ترتيب العنصر في القائمة:**
  + استخدم index في view\_item\_list/select\_item لفهم أثر الموضع على CTR.

### **معايير قبول (QA سريعة)**

* تظهر كل الأحداث في **DebugView** بترتيب منطقي دون ازدواجية.
* purchase يظهر بقيمة صحيحة ومصفوفة items[] مكتملة (لا نقص في item\_id أو price).
* الفروقات بين إيراد GA4 والنظام الخلفي ضمن **±5–10%** ومفسّرة.
* لا يوجد تكرار لـ purchase عند إعادة فتح شاشة النجاح/إعادة محاولة الدفع.
* جميع أحداث السلة/الدفع تحمل currency و(إن أمكن) value.

الأحداث/المعلمات المخصّصة لسوقك (COD واللوجستيات)

## **#4) الأحداث/المعلمات المخصّصة لسوقك (COD واللوجستيات)**

**الاستفادة (Why it matters)**

* تمنحك رؤية تشغيلية حقيقية بعد الشراء: **نجاح COD، أسباب RTO/NDR، الالتزام بالـSLA**، أداء شركاء التوصيل والمدن.
* تغلق فجوة لا يوفّرها GA4 افتراضيًا، وتغذّي لوحات **COD/RTO** واتخاذ القرار (إيقاف مدينة/شركة، تعديل سياسة الدفع…).

**الهدف (What we want to achieve)**

* توحيد **أحداث سيرفر** عبر *Measurement Protocol* لكل تغيّر حالة للطلب ومحاولات التسليم وتحصيل الـCOD.
* إرسال **معلمات قياسية** عن المدينة/الشريك/السبب/نوع الدفع… قابلة للتقارير في GA4 وBigQuery.
* تمكين حسابات مثل **COD Success، RTO/NDR، On-time Delivery** لكل مدينة/شريك/فئة.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) أحداث السيرفر الموصى بها (Measurement Protocol)**

تُرسل من السيرفر عند حدوث التغيير (idempotent: لا تُكرّر نفس التحديث لنفس order\_id وstatus).

1. **order\_status\_update** — تغيّر حالة الطلب

* **متى؟** عند Confirmed / Packed / Shipped / OutForDelivery / Delivered / Canceled / Returned\_to\_origin.
* **معلمات (أساسية):**
  + order\_id *(سلسلة فريدة)*
  + order\_status ∈ {confirmed,packed,shipped,out\_for\_delivery,delivered,canceled,rto}
  + payment\_type ∈ {COD,Prepaid}
  + delivery\_partner *(اسم شركة التوصيل)*
  + delivery\_city, delivery\_zone
  + promised\_sla\_days *(عدد الأيام الموعودة)*
  + shipping\_cost *(رقم)*
  + *(اختياري)*: cancel\_reason، rto\_reason
* **ملاحظات:** لا تُرسل PII (أسماء/هواتف). اعتمد user\_id إن متاح أو app\_instance\_id للربط.

1. **delivery\_attempt** — محاولة تسليم (ناجحة/فاشلة)

* **متى؟** عند كل محاولة زيارة من المندوب.
* **معلمات:**
  + order\_id, attempt\_number *(1,2,…)*
  + attempt\_outcome ∈ {success,failed}
  + إذا فشل: failure\_reason ∈ {customer\_unavailable,wrong\_number,address\_incomplete,refused\_cod,security,weather,other}
  + delivery\_partner, delivery\_city, delivery\_zone

1. **cod\_collection** — تحصيل مبلغ الـCOD

* **متى؟** لحظة استلام العميل ودفعه نقدًا.
* **معلمات:**
  + order\_id
  + cod\_collected ∈ {true,false}
  + collected\_amount *(رقم؛ إن true)*
  + delivery\_partner

**نتيجة الاشتقاق:**

* **COD Success** = عدد cod\_collection=true ÷ عدد طلبات payment\_type=COD.
* **RTO Rate** = عدد حالات order\_status=rto ÷ الشحنات المرسلة.
* **NDR Rate** = عدد محاولات delivery\_attempt الفاشلة ÷ إجمالي المحاولات.
* **On-time Delivery** = Delivered ضمن promised\_sla\_days.

### **B) معلمات نضيفها أيضًا في أحداث الشراء والمسار**

* على purchase ومرحلة الدفع/الشحن:
  + payment\_type (COD/Prepaid)
  + delivery\_city, delivery\_zone, delivery\_partner
  + *(اختياري)* cod\_fee, discount\_value
* على add\_to\_cart وview\_item:
  + size, color, stock\_status (= in\_stock/out\_of\_stock)

### **C) تعريف القيم (Enumerations موحّدة)**

* **الحالات:** confirmed,packed,shipped,out\_for\_delivery,delivered,canceled,rto.
* **أسباب الإلغاء:** customer\_request,out\_of\_stock,address\_issue,pricing\_issue,other.
* **أسباب RTO/الفشل:**customer\_unavailable,refused\_cod,address\_incomplete,wrong\_number,security,weather,other.
* **نوع الدفع:** COD,Prepaid.

حافظ على **ثبات القيم** حرفيًا (lowercase + snake\_case) لتجنّب تضارب الأبعاد.

### **D) جودة البيانات وقيود مهمّة**

* **Idempotency:** أرسل مفتاحًا مركّبًا (مثل order\_id + order\_status + event\_date) وتجاهل التكرارات.
* **حدود GA4 للتعاريف المخصّصة:** سجّل فقط الأهم (مثل order\_status, delivery\_partner, payment\_type, failure\_reason, rto\_reason, delivery\_city).
* **عدم إرسال PII** (الهاتف/العنوان التفصيلي) — استخدم معرّفات داخلية فقط.
* **الاتساق الزمني:** وقت الحدث = وقت وقوع الحالة، لا وقت المعالجة.

### **E) مؤشرات تحسب لاحقًا (BigQuery)**

* **COD Success Rate**, **RTO Rate**, **NDR Rate**, **On-time %**.
* **Avg Delivery Days** = تاريخ delivered − تاريخ shipped.
* **Attempts per Order** = MAX(attempt\_number) لكل order\_id.
* **Partner/City Scorecards**: مقارنة الشركاء والمدن بهذه المؤشرات.

### **F) QA سريعة لهذه النقطة**

* يظهر order\_status\_update لكل حالة رئيسية مرة واحدة لكل طلب.
* delivery\_attempt يزيد رقم المحاولة تدريجيًا نفس order\_id.
* cod\_collection=true فقط للطلبات التي نوع دفعها COD.
* عدم وجود قيم فارغة لأبعاد رئيسية (payment\_type, delivery\_partner, order\_status).
* التطابق بين **قيمة تحصيل COD** ومبالغ الطلب ضمن سياسة التقارير.

التحويلات (Conversions / Key Events)

## **#5) التحويلات (Conversions / Key Events)**

**الاستفادة (Why it matters)**

* تجعل “نتيجة العمل” قابلة للقياس: كم طلبًا اكتمل؟ بقيمة كم؟ ومن أي قناة؟
* تُغذّي حملات الإعلانات بالتحويلات والقيم لاحتساب **ROAS** بدقّة.
* تمنحك مؤشرات مساندة لمسار الشراء (micro-conversions) لاكتشاف مكان الهدر مبكرًا.

**الهدف (What we want to achieve)**

* اعتماد **purchase** كتحويل الإيراد الوحيد المستخدم للحملات والقيمة.
* تعريف تحويلات مساندة اختيارية لمسار الشراء (ليس للـROAS): begin\_checkout, add\_shipping\_info, add\_payment\_info.
* معالجة خصوصية **COD**: إبقاء purchase عند إنشاء الطلب، واستخدام أحداث ما بعد الشراء (سيرفر) للتشغيل والتحليل—not للإعلانات—لتجنّب ازدواج الإيراد.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **1) ما الذي نعلّمه كتحويل؟**

* **أساسي (للإعلانات والقيمة):**
  + purchase فقط.
* **اختياري (صحة المسار/تحسين المنتج—not Ads ROAS):**
  + begin\_checkout, add\_shipping\_info, add\_payment\_info.
  + يمكن تعليمها “Key Events” داخل GA4 لاستخدامها في التقارير و**Explorations**، لكن **لا نستوردها كـ Primary conversions** في منصّات الإعلانات.

إن احتجت “تتبّع تثبيت التطبيق” تسويقيًا، يمكنك تعليم first\_open كـ Key Event **للتقارير الداخلية فقط**، بينما الاعتماد على Firebase/ASA/SKAN أو MMP لقياس اكتساب iOS بدقة.

### **2) الإعداد داخل GA4**

* من **Configure → Events**: علِّم purchase (والمساندة إن رغبت) كـ **Key events**.
* **Counting method**:
  + اترك purchase على **Once per event** (الافتراضي).
  + للمساندة، يمكن **Once per session** لتجنّب تضخيم العدّ إن كرّر المستخدم الخطوة.
* تأكّد أن purchase يرسل: transaction\_id (فريد)، value, currency, items[].
* **القيمة** المعتمدة في التقارير وROAS هي Param value مع currency. تجنّب إرسال value=0.

### **3) خصوصية الدفع عند الاستلام (COD)**

* نحسب **التحويل** لحظة إنشاء الطلب (purchase).
* لتحصيل النقد لاحقًا، نستخدم سيرفر:
  + order\_status\_update و/أو cod\_collection (غير مستوردة كتحويلات Ads).
* في منصّات الإعلانات:
  + **استورد purchase فقط كـ Primary conversion بالقيمة.**
  + أبقِ أحداث COD اللاحقة للتشغيل/التحليل في GA4/BigQuery حتى لا تضاعف الإيراد.

### **4) الاستيراد إلى منصّات الإعلانات**

* **Google Ads**: اربط GA4 واستورد purchase فقط (Primary).
* **Meta/أخرى**: أرسل التحويل عبر CAPI/SDK مع نفس transaction\_id وvalue لتسهيل المطابقة—و**تجنّب الازدواج** (لا تجمع بين طريقتين لنفس المنصّة بلا ضبط).
* في Google Ads يمكنك تمييز التحويلات الثانوية (Observation) إن رغبت بمؤشرات مساندة.

### **5) النوافذ ونَسَب التحويل (Attribution)**

* فعِّل **Data-Driven Attribution** في GA4.
* اضبط **Lookback windows** بما يلائم دورة قرار الشراء لديك (غالبًا 30 يومًا تكفي للتطبيقات الاستهلاكية).
* التزم بحوكمة **UTM** الصارمة لضمان دقّة إسناد القنوات.

### **6) منع الازدواج والأخطاء الشائعة**

* لا تجعل purchase يُطلق عند **إعادة فتح** شاشة النجاح أو عند **استعادة الجلسة** بعد عطل؛ استخدم منطق **idempotent** (التحقّق من transaction\_id).
* لا ترسل purchase من **التطبيق والسيرفر** معًا لنفس الطلب دون إزالة أحدهما أو توحيد المفتاح.
* استبعد الأجهزة/الحسابات الداخلية حتى لا تُحتسب تحويلاتها.

### **7) QA سريعة**

* رحلة اختبار: منتج → سلة → دفع → purchase بقيمة صحيحة وعملة صحيحة.
* تحقق من **Realtime/DebugView** أن purchase يظهر مرة واحدة مع items[] مكتمل.
* قارن إيراد GA4 مع النظام الخلفي (عينة يومية) ضمن **±5–10%** مع تفسير الفروقات (إلغاءات/فروقات توقيت).

خصائص المستخدم (User Properties)

## **#6) خصائص المستخدم (User Properties)**

**الاستفادة (Why it matters)**

* تفهم “من يشتري وماذا يفضّل”: مقاس/فئة مفضلة، تفضيل الدفع، شريحة الولاء…
* تُمكِّن جماهير دقيقة للريمَركتِنغ والتخصيص (Remote Config/Notifications).
* تقلّل الهدر: عروض مناسبة لكل شريحة بدل رسائل عامة.

**الهدف (What we want to achieve)**

* اعتماد **3–5 خصائص أساسية** فقط (لتجنّب تجاوز حدود GA4 ولضمان جودة).
* أسماء ثابتة بصيغة snake\_case وقيم منظّمة (قوائم ثابتة/تصنيفات).
* استخدام الخصائص في **Audiences** و**التخصيص داخل التطبيق**، مع احترام الخصوصية (بدون أي PII).

**التفاصيل (How to do it — مختصر ومنظّم)**

### **أ) قائمة الخصائص المقترَحة لتطبيق “زهراء”**

اختر 3–5 من التالي بحسب الأولوية، ولا ترسل PII.

1. preferred\_payment

* **القيم**: cod, prepaid, unknown
* **المصدر**: آخر اختيار دفع ناجح أو متكرر.
* **الاستخدام**: خصومات/رسائل مخصّصة حسب تفضيل الدفع.

1. favorite\_size

* **القيم**: مثل s,m,l,xl … أو unknown
* **المصدر**: أكثر مقاس أُضيف للسلة/أُشتري.
* **الاستخدام**: ترتيب المقاسات، تنبيه بتوافر مقاسه.

1. favorite\_category

* **القيم**: evening\_dresses, shoes, bags, …
* **المصدر**: أعلى مشاهدات/مشتريات.
* **الاستخدام**: توصيات وفئات مبدئية في الصفحة الرئيسية.

1. loyalty\_tier

* **القيم**: none,silver,gold,platinum
* **المصدر**: من نظام الولاء/إجمالي الإنفاق.
* **الاستخدام**: قسائم/شحن مجاني مخصص.

1. (اختياري) city\_bucket

* **القيم**: أسماء مدن رئيسية أو other
* **الاستخدام**: عروض/سياسات COD/شحن حسب المدينة.

ملاحظات:

* لا تُضمّن البريد/الهاتف/العنوان.
* إن احتجت خاصية تقنية لإدارة الإصدارات، استخدم schema\_version كـ **Event param** وليس User Property.

### **ب) قواعد التسمية والقيم**

* الاسم: أحرف صغيرة + شرطة سفلية snake\_case.
* القيم: **محدودة ومُقنَّنة** (Enumerations)، وتجنّب نصوص حرّة.
* لا تغيّر معنى القيم مع الزمن (ثبات الدلالات).

### **ج) الضبط في GA4/Firebase**

* من التطبيق: استخدم setUserProperty("preferred\_payment", "cod") (iOS/Android عبر Firebase SDK).
* في GA4: سجّل كل خاصية ضمن **Configure → Custom definitions** بنطاق **User** لتظهر في التقارير والجماهير.
* حدود GA4 تقريبية: عدد محدود لخصائص المستخدم والأبعاد المخصّصة—فاحفظها لأهم ما يفيد القرارات.

### **د) استخدامات عملية**

* **Audiences**:
  + preferred\_payment=cod AND cart\_abandoner → رسالة تشجّع إتمام COD.
  + favorite\_category=evening\_dresses → ريمَركتِنغ لفساتين السهرة.
  + loyalty\_tier=gold → عروض حصرية.
* **التخصيص داخل التطبيق (Remote Config/In-App Messages)**:
  + إبراز فئة مفضّلة في الهوم.
  + تنبيه “عاد مقاسك” لمَن يملك favorite\_size.
* **التحليلات**: تقسيم CR/AOV حسب preferred\_payment أو city\_bucket لفهم فروقات الأداء.

### **هـ) خصوصية وجودة**

* **بدون PII** إطلاقًا في الخصائص.
* حدّث الخاصية عند تغيّر السلوك، ولا تُرسل قيمًا متناقضة لنفس المستخدم.
* راقب نسب unknown—إن زادت، أصلح منطق الاستنتاج.

### **و) QA سريعة**

* تظهر الخصائص في **User properties** وتقارير **Audiences** بعد تعريفها.
* تحقق بعينة مستخدم: تغيّر preferred\_payment فعليًا بعد شراء Prepaid بدل COD.
* لا توجد قيم خارج القائمة المسموح بها.

الاكتساب ونسب التحويل (Attribution & UTMs)

## **#7) الاكتساب ونسب التحويل (Attribution & UTMs)**

**الاستفادة (Why it matters)**

* تعرف القناة/الحملة التي تجلب **طلبات وقيمة حقيقية** (ROAS)، بدل ضوضاء “زيارات” فقط.
* تمنع “الترافيك المظلم” (روابط بدون UTM) وتقلّل تضارب القنوات.
* أساسي لقرارات الميزانية: أين نزيد الإنفاق وأين نقلّص؟

**الهدف (What we want to achieve)**

* حوكمة صارمة لـ **UTM** على كل رابط خارجي (إعلانات/مؤثرين/رسائل).
* تفعيل **Data-Driven Attribution** داخل GA4، وربط Metabase,Pixel,Google Ads لاستيراد التحويلات/الجماهير.
* إعداد تدفّق صحيح للتثبيت والزيارات عبر **Deep Links** (مع تمرير UTM) + مراعاة iOS (SKAdNetwork) وAndroid (Play Install Referrer).
* منع الازدواج بين **SDK وواجهات الإعلانات** (CAPI/SDK) باستخدام مفاتيح dedup.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) حوكمة UTM (قاعدة ذهبية)**

* استخدم دائمًا: utm\_source, utm\_medium, utm\_campaign, (اختياري: utm\_content, utm\_term).
* **تنسيق موحّد (snake\_case)** وثابت عبر الفريق.
* أمثلة خريطة قياسية:
  + Google Ads: source=google, medium=cpc, campaign=2025q3\_generic\_dresses\_ar
  + Meta (FB/IG): source=meta, medium=cpc, campaign=2025q3\_dp\_feed\_ar
  + TikTok: source=tiktok, medium=cpc
  + Snapchat: source=snapchat, medium=cpc
  + Influencer: source=instagram, medium=influencer, content=@creator\_handle
  + Email/SMS: source=crm, medium=email أو sms
  + Organic Social: اتركها **بدون UTM** أو استخدم medium=organic\_social بوضوح
* **ممنوع**: قيم عامة مثل utm\_medium=ad أو أحرف عربية في المفاتيح (القيم يجوز تعريبها).
* أنشئ **دليل داخلي** + مولّد روابط (Sheet بسيط) لتفادي الأخطاء.

### **B) الروابط والتعمّق (Deep Linking)**

* استخدم **Firebase Dynamic Links** (أو Universal/App Links) مع تمرير UTM تلقائيًا إلى التطبيق.
* فعّل **Deferred deep link** للتثبيت: إذا المستخدم يثبت أولًا ثم يفتح، تُحفظ الـUTM وتُنسب الزيارة/الشراء للحملة الصحيحة.
* اختبر: رابط منتج من إعلان → يفتح التطبيق مباشرةً على صفحة المنتج مع نفس الـUTM.

### **C) iOS وAndroid (الاكتساب الفني)**

* **iOS**: التزم **ATT**؛ عند عدم الموافقة، يعتمد القياس على **SKAdNetwork** (تجميع/تأخير).
  + صمّم **Conversion Values** للتركيز على purchase وقيمته (خريطة مبسّطة أولًا).
* **Android**: فعّل **Play Install Referrer** لضمان نقل الـUTM بدقة من المتجر للتطبيق.
* إن كان الحجم الإعلاني كبيرًا ومتعدد المنصات، يمكن النظر في **MMP** (Appsflyer/Adjust) لاحقًا—اختياري.

### **D) الربط مع Google Ads وباقي المنصّات**

* **Google Ads**: اربط الحساب بـGA4، **استورد purchase فقط** كتحويل أساسي بالقيمة.
* **Meta/TikTok/Snap**: أرسل الحدث عبر **CAPI/SDK** بنفس transaction\_id وvalue (لـdedup والتطابق).
  + في Meta استخدم **event\_id** = transaction\_id لتجنّب الازدواج بين Web/App/CAPI.

### **E) النوافذ ونسبة الإسناد (Attribution)**

* فعّل **Data-Driven Attribution** في GA4.
* ثبّت نوافذ ملائمة لدورة قرار الشراء لديك (مثلاً 7–30 يوم نقرة / 1 يوم مشاهدة حسب السياسة التسويقية).
* لا تخلط تعريفات مختلفة بين المنصات دون توثيق—وحافظ على جدول يوضح الفروقات.

### **F) تكلفة القنوات (Cost) وROAS مجمّع**

* **Google Ads** تُجلب التكلفة تلقائيًا بعد الربط.
* للقنوات الأخرى: إمّا **Data Import** في GA4 (قوالب CSV بانتظام) أو احتساب **ROAS** في BigQuery ولوحات Looker Studio.
* الحقول الموصى بها للتكلفة: date, source, medium, campaign, adset/adgroup, spend, impressions, clicks.

### **G) مشاركة/روابط العملاء (Dark Social)**

* أزرار “مشاركة المنتج” تُضيف UTM واضحة:
  + مثال واتساب: utm\_source=whatsapp&utm\_medium=share&utm\_campaign=product\_share
* بذلك لا تضيع زيارة/شراء جاء عبر صديقة شاركت الرابط.

### **H) منع الازدواج (Dedup)**

* لا ترسل نفس التحويل من قناتين بدون **مفتاح تطابق** موحّد:
  + استخدم transaction\_id كمرجع في كل مكان.
  + في Meta: (event\_name + event\_time + event\_id) → اجعل event\_id=transaction\_id.

### **I) QA سريعة**

* اختبر روابط UTM حقيقية لكل قناة وشاهدها في **Realtime/Traffic Acquisition**.
* ثبّت تسمية الحملات وفق الدليل، ولاحظ عدم ظهور مصادر مكررة/غامضة.
* اختبر تثبيت iOS/Android من إعلان → تأكد من وصول الـUTM عبر Deferred deep link.
* راقب **(not set)** في تقارير GA4—إن ارتفع، هناك تسريب UTM.

التكاملات الأساسية (BigQuery، Ads، CAPI…)

## **#8) التكاملات الأساسية (BigQuery، Ads، CAPI…)**

**الاستفادة (Why it matters)**

* تجعل GA4 جزءًا من منظومة بيانات متكاملة: تقارير تنفيذية دقيقة، ROAS موحّد، وجماهير تُشغِّل التسويق تلقائيًا.
* تمكّنك من تحليل عميق (COD/RTO/بحث/مقاسات) لا يتوفّر جاهزًا داخل GA4 وحده.
* تقلّل الاعتماد اليدوي وتمنحك **مصدر حقيقة واحد** يمكن الوثوق به.

**الهدف (What we want to achieve)**

* ربط GA4 بـ **BigQuery** كمستودع خام وتحليلات متقدّمة.
* ربط **Google Ads** لاستيراد التحويلات والجماهير تلقائيًا.
* تفعيل **Conversions API** للقنوات (Meta/TikTok/Snap) مع منع الازدواج.
* (اختياري) ربط **Search Console** لـ Web Stream، وربط أدوات الـCRM/التراسل.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) BigQuery (مصدر الحقيقة)**

* **الربط**: Admin → BigQuery Links → اختر المشروع/المنطقة (يفضل أقرب منطقة).
* **Dataset**: سمِّه مثلاً ga4\_zahraah\_prod. فعّل **Daily export** (و**Streaming** لاحقًا لو أردت بيانات لحظية).
* **تقليل التكلفة/تسريع الاستعلام**:
  + الجداول **مقسّمة** على event\_date.
  + **عنقد** (Cluster) على: event\_name, platform, traffic\_source.source, user\_pseudo\_id.
* **Views مُقترَحة** (تُبنى مرة ويُعاد استخدامها في اللوحات):
  + fct\_purchases (كل purchase مع تفكيك items[]).
  + fct\_funnel\_steps (خطوات المسار لكل جلسة/مستخدم).
  + fct\_search (الاستعلامات + zero\_results).
  + fct\_order\_status (يستلِم من أحداث السيرفر — نقطة #4).
* **الصلاحيات (IAM)**:
  + محلل البيانات: BigQuery Data Viewer على الـdataset.
  + من ينشئ اللوحات/الاستعلامات: BigQuery Job User.
  + الحساب الخدمي للسيرفر (Measurement Protocol/ETL): BigQuery Data Editor على جداولك الوسيطة فقط.
* **لوكر ستوديو (Looker Studio)**: اربطه بـBigQuery مباشرةً لبناء لوحات: تنفيذية، مسار الشراء، COD/RTO.

### **B) Google Ads (Link & Import)**

* **الربط**: Admin → Product Links → Google Ads → اربط الحساب الصحيح.
* **استيراد التحويلات**: استورد **purchase فقط** كتحويل أساسي بالقيمة/العملة.
* **الجماهير**: فعّل مشاركة **Audiences** من GA4 إلى Google Ads (Cart abandoners، High AOV…).
* **Auto-tagging**: مفعّل (gclid) لضمان الإسناد الدقيق.

### **C) Conversions API (Meta/غيرها) + منع الازدواج**

* **Meta CAPI** (لإعلانات فيسبوك/إنستغرام):
  + أرسل حدث **Purchase** من السيرفر **بنفس** transaction\_id وvalue وcurrency المرسلة من التطبيق (SDK) لاستخدام **dedup**.
  + ضَع event\_id = transaction\_id.
  + content\_ids = قائمة item\_id، content\_type=product, ويمكن تمرير contents (id, quantity, item\_price).
  + لا تُرسل PII؛ وإن اضطررت (لبعض مزايا المطابقة) يجب **تجزئته/تشفيره** حسب متطلبات المنصّة.
* **TikTok/Snap**: نظير CAPI موجود—نفس مبدأ الـdedup بواسطة event\_id/transaction\_id.
* **قاعدة ذهبية**: لا تجمع إرسال نفس التحويل من SDK وCAPI **بدون** مفتاح تطابق موحّد، كي لا يتضاعف التحويل.

### **D) Data Import للتكلفة (Cost)**

* للقنوات خارج Google (Meta/TikTok/Influencer…):
  + إمّا **Data Import** داخل GA4 (ملف يومي يحتوي: date, source, medium, campaign, adset/adgroup, spend, clicks, impressions).
  + أو احتساب **ROAS** داخل **BigQuery** ثم عرض النتائج في Looker Studio (أكثر مرونة، خاصةً عند اختلاف تسمية الحملات).
* **توحيد التسمية** مع دليلك في #7 مهم كي تتطابق التكلفة مع الزيارات/التحويلات.

### **E) Search Console (اختياري – Web Stream)**

* إن كان لديك Web Stream/موقع: اربط **Search Console** لرؤية Queries/Pages العضوية داخل GA4.
* ينفع لفهم **النيّة العضوية** وتحسين صفحات الهبوط.

### **F) CRM/التراسل (اختياري لكنه مفيد)**

* اربط منصّة الرسائل (Email/SMS/Push) عبر **Webhook/ETL** لتغذية جماهير GA4/BigQuery:
  + أمثلة جماهير تشغيلية: cart\_abandoners, dress\_enthusiasts, cod\_success\_repeaters.
  + استخدم **Remote Config/In-App Messaging** لتخصيص التجربة داخل التطبيق بناءً على جماهير GA4.

### **G) قنوات المشاركة (Dark Social) وروابط عميقة**

* **Firebase Dynamic Links** مع تمرير UTM و**Deferred Deep Link** للتثبيت.
* أزرار المشاركة تضيف UTM قياسية (source=whatsapp|telegram, medium=share).

### **H) QA سريعة**

* BigQuery يتلقّى الجداول يوميًا، واستعلامات العيّنة تعمل بسرعة (بسبب Partition/Cluster).
* Google Ads يرى purchase بالقيمة الصحيحة ويعرض الجماهير المستوردة.
* Meta/غيرها تُظهر **تطابقًا عاليًا** (Match/Dedup) لتفادي تضخيم التحويل.
* ملفات **Cost Import** تطابق مخطط الحقول، وتظهر الأرقام في تقارير الاكتساب أو في لوحات BigQuery.

أحداث السيرفر لما بعد الشراء ( Protocol)

## **#9) أحداث السيرفر لما بعد الشراء (Measurement Protocol)**

**الاستفادة (Why it matters)**

* تغلق الفجوة بين “طلب أُنشئ” و“طلب اكتمل فعلًا”: نجاح **COD**، محاولات التسليم، أسباب الفشل/RTO.
* تمنح رؤية تشغيلية دقيقة حسب **المدينة/شركة التوصيل/نوع الدفع** وتغذّي قرارات التوزيع وسياسات COD.
* تحافظ على نقاء تحويلات الحملات (ROAS) عبر ترك **purchase** للإعلانات، واستخدام أحداث السيرفر للتشغيل والتحليل.

**الهدف (What we want to achieve)**

* إرسال تغيّرات حالة الطلب ومحاولات التسليم وتحصيل الـCOD إلى **نفس GA4 Property** من السيرفر بصيغة موحّدة.
* ربط هذه الأحداث بالمستخدم/الجهاز قدر الإمكان (user\_id أو app\_instance\_id) لضمان تتبّع صحيح عبر القنوات.
* تخزين كل شيء في **BigQuery** لاشتقاق مؤشرات: COD Success، RTO/NDR، زمن التوصيل، الالتزام بالـSLA… إلخ.

**التفاصيل (How to do it — مختصر ومنظَّم)**

### **A) ما هي الأحداث التي نرسلها من السيرفر؟**

1. **order\_status\_update** — تغيّر حالة الطلب
   * order\_id (فريد داخليًا)
   * order\_status ∈ {confirmed,packed,shipped,out\_for\_delivery,delivered,canceled,rto}
   * payment\_type ∈ {COD,Prepaid}
   * delivery\_partner, delivery\_city, delivery\_zone
   * promised\_sla\_days, shipping\_cost
   * (اختياري) cancel\_reason, rto\_reason
2. **delivery\_attempt** — محاولة تسليم
   * order\_id, attempt\_number (1,2,…)
   * attempt\_outcome ∈ {success,failed}
   * إذا **failed**: failure\_reason ∈ {customer\_unavailable,refused\_cod,address\_incomplete,wrong\_number,security,weather,other}
   * delivery\_partner, delivery\_city, delivery\_zone
3. **cod\_collection** — تحصيل نقدي عند التسليم
   * order\_id
   * cod\_collected ∈ {true,false}
   * إذا **true**: collected\_amount
   * delivery\_partner

**ملحوظة**: لا نعلّم هذه الأحداث كتحويلات Ads. تبقى لتحليل العمليات واتخاذ القرار.

### **B) ربط الحدث بالمستخدم/الجهاز (الهوية)**

* أرسل مع كل نداء واحدًا مما يلي (المتاح لديك):
  + user\_id إن كان المستخدم مسجّل الدخول.
  + أو app\_instance\_id (خزّنه عند إنشاء الطلب من التطبيق لترسله من السيرفر لاحقًا).
* أرسل **timestamp** وقت وقوع الحدث الحقيقي (وليس وقت المعالجة) للحفاظ على تسلسل زمني صحيح.
* إن كانت لديك موافقات خصوصية مقيِّدة، أرسل الحدث كـ **غير مُخصّص للإعلانات** (non-personalized) عند اللزوم.

### **C) الاتساق ومنع التكرار (Idempotency)**

* أنشئ مفتاحًا مركّبًا تمنع به الازدواج، مثل:
  + dedup\_key = order\_id + ":" + order\_status لأحداث الحالة،
  + وorder\_id + ":" + attempt\_number لمحاولات التسليم.
* تجاهل أي نداء مُعاد بنفس المفتاح. سجّل الاستثناءات في لوج واضح.

### **D) مثال حمولة (شكل توضيحي مبسّط)**

{

"app\_identity": {

"firebase\_app\_id": "<your\_app\_id>",

"user\_id": "U\_984321",

"app\_instance\_id": "a1b2c3d4e5"

},

"events": [

{

"name": "order\_status\_update",

"timestamp\_micros": 1734470400000000,

"params": {

"order\_id": "ZHR-2025-000123",

"order\_status": "out\_for\_delivery",

"payment\_type": "COD",

"delivery\_partner": "FastX",

"delivery\_city": "Sanaa",

"delivery\_zone": "Hadda",

"promised\_sla\_days": 2,

"shipping\_cost": 1200

}

},

{

"name": "delivery\_attempt",

"timestamp\_micros": 1734553200000000,

"params": {

"order\_id": "ZHR-2025-000123",

"attempt\_number": 1,

"attempt\_outcome": "failed",

"failure\_reason": "customer\_unavailable",

"delivery\_partner": "FastX",

"delivery\_city": "Sanaa",

"delivery\_zone": "Hadda"

}

}

],

"user\_properties": {

"preferred\_payment": { "value": "cod" },

"city\_bucket": { "value": "sanaa" }

}

}

الغرض توضيحي للحقول والمنطق—not مواصفات API حرفيًا. الأسماء والقيم ثابتة وفق قاموسك.

### **E) أين تُسجِّل الأبعاد المخصّصة؟**

* في **GA4 → Configure → Custom definitions**:
  + سجّل كأبعاد حدثية: order\_status, delivery\_partner, delivery\_city, delivery\_zone, attempt\_outcome, failure\_reason, payment\_type.
* لا تُفرط في التسجيل—GA4 لديه حدود. اختر الأهم لأسئلة عملك ولوحاتك.

### **F) ماذا نستنتج في BigQuery؟ (أمثلة عملية)**

* **COD Success** = عدد cod\_collection=true ÷ عدد طلبات payment\_type=COD.
* **RTO Rate** = عدد order\_status=rto ÷ الشحنات المرسلة.
* **NDR Rate** = محاولات delivery\_attempt الفاشلة ÷ كل المحاولات.
* **On-time %** = delivered خلال promised\_sla\_days.
* **Avg Delivery Days** = تاريخ delivered − تاريخ shipped.
* **Partner/City Scorecards** للمقارنة وإيقاف/تحسين المدن أو الشركاء ذوي الأداء الضعيف.

### **G) جودة البيانات (Best Practices مختصرة)**

* **ثبات القيم** (lowercase + snake\_case) وتجنّب النصوص الحرّة.
* **لا PII** داخل الأحداث أو الـparams.
* الترتيب الزمني مهم: استخدم الطابع الزمني الحقيقي.
* اختبر على طلبات تجريبية كاملة المسار وتحقق من ظهور الأحداث في **Realtime/DebugView** ثم في **BigQuery**.

### **H) أخطاء شائعة يجب تجنّبها**

* إرسال purchase مرّة أخرى من السيرفر لنفس الطلب (يسبّب ازدواج الإيراد).
* نسيان ربط الحدث بـ user\_id/app\_instance\_id → فقدان القدرة على التحليل لكل مستخدم/قناة.
* قيم متعدّدة لنفس البُعد (مثل شركاء توصيل بأسماء غير موحّدة).
* تجاهل حالات فشل التسليم → RTO/NDR غير قابلة للقياس.

الجماهير (Audiences)

## **#10) الجماهير (Audiences)**

**الاستفادة (Why it matters)**

* توصل الرسالة الصحيحة للشخص الصحيح: **تحويل أعلى وتكلفة أقل**.
* تشغيل تلقائي للتسويق: ريماركتنغ، إشعارات داخل/خارج التطبيق، وتجارب مخصّصة.
* قياس سلوك الشرائح (COD، مدينة، فئة) بدل المتوسط العام المضلِّل.

**الهدف (What we want to achieve)**

* بناء **مجموعة جماهير أساسية** ثابتة تخدم: (1) التسويق المدفوع، (2) الرسائل/الإشعارات، (3) التحليلات.
* استخدام **شروط دقيقة + نوافذ زمنية** (عضوية/استرجاع) مع استثناءات تمنع التداخل والازدواج.
* ربط الجماهير بـ **Google Ads**، واستهلاكها داخل التطبيق عبر **Firebase (Remote Config / In-App Messaging)**، وللقنوات الأخرى عبر BigQuery/CDP.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) مبادئ تصميم الجمهور**

* **بسيط وقابل للتنفيذ**: شروط قليلة واضحة، اسمٌ وصفي.
* **نافذة زمنية** مناسبة للسلوك (ساعات/أيام لترك السلة، 30–90 يومًا للاحتفاظ).
* **استثناءات** صريحة (Exclude purchasers) لتجنّب الإنفاق على من أتمّ الشراء.
* استخدم **User properties** لتخصيص أدق (مثلاً preferred\_payment=cod).

### **B) جماهير تشغيلية (Operational) — أمثلة جاهزة**

1. **Cart Abandoners (24–72h)**
   * دخل begin\_checkout أو add\_to\_cart **ولم** يطلق purchase خلال النافذة.
   * *Exclude*: من اشترى بعد ذلك.
   * استخدام: إشعارات/إيميل تذكير + إعلان ديناميكي.
2. **COD Successful Repeaters**
   * preferred\_payment=cod + ≥2 purchase بدون rto.
   * استخدام: عروض تكرار/باقة COD، اختبار رفع AOV.
3. **Churn Risk (30d)**
   * مستخدم نشِط تاريخيًا (≥2 زيارات أو 1 شراء) **ولم** يزر/يشترِ خلال 30 يومًا.
   * استخدام: رسالة رجوع + كوبون محدود.
4. **Dress Enthusiasts**
   * ≥3 view\_item أو ≥1 purchase ضمن فئة evening\_dresses في آخر N يوم.
   * استخدام: توصيات/حملات فساتين فقط.
5. **Search Zero-Results**
   * ≥2 search مع results\_count=0 خلال 7 أيام.
   * استخدام: تنبيه داخلي لتحسين الفهرس + رسالة بدائل.
6. **High AOV**
   * متوسط سلة ≥ حد معيّن في آخر 90 يومًا.
   * استخدام: عروض حصرية/شحن مجاني.
7. **City/Partner Watchlist**
   * city\_bucket∈{Sanaa, Aden,…} أو delivery\_partner=….
   * استخدام: سياسة COD/الشحن وقياس RTO/On-time.

### **C) جماهير تسويقية (Remarketing / Acquisition)**

* **View → No Add to Cart** (إقناع أولي).
* **Add to Cart → No Purchase** (استرجاع قوي).
* **Repeat Buyers (90d)** (قيمة عمرية أعلى).
* **New Installers (7d)** (التفعيل المبكر: إبراز فئة/عرض ترحيبي).

دائمًا **Exclude Purchasers (recent)** عندما لا تريد إنفاقًا على من اشترى للتو.

### **D) جماهير تسلسلية (Sequence) ونوافذ**

* مثال تسلسل:
  + خطوة 1: view\_item لفئة فساتين
  + ثم خلال **48h**: **لم** يحدث add\_to\_cart
  + → جمهور **“لم يتحرّك بعد المشاهدة”**.
* اضبط **Membership duration** بما يلائم دورة القرار لديك.

### **E) Audience Triggers (مفيد داخل التطبيق)**

* عند دخول المستخدم جمهورًا معيّنًا، أطلق حدثًا يطفّر رسالة/تجربة:
  + مثال: دخول **Cart Abandoner** → In-App Message بقسيمة صغيرة.

### **F) الربط والاستهلاك**

* **Google Ads**: مشاركة الجماهير مباشرة من GA4 (Cart abandoners، High AOV…).
* **داخل التطبيق**: Firebase **Remote Config / In-App Messaging** لاستبدال بنرات ومحتوى حسب الجمهور.
* **قنوات غير Google**: استخرج الجمهور من **BigQuery** إلى CDP/أتمتة الرسائل (Email/SMS/Push).

### **G) التداخل ومنع الازدواج**

* أنشئ **Hierarchies**:
  + إذا كان المستخدم في **Purchasers (7d)** → استبعده من **Cart Abandoners**.
  + أعطِ الأولوية لجماهير القيمة الأعلى لتقليل تضارب الحملات.

### **H) خصوصية وجودة**

* لا تُدرج أي **PII** في شروط الجمهور.
* راقب حجم الجمهور وتغيّره؛ إن كان صغيرًا جدًا، وسّع النافذة أو خفّف الشروط.
* تأكّد أن خصائص المستخدم (preferred\_payment, favorite\_category) يتم تحديثها باستمرار.

### **I) QA سريعة**

* اختبر دخول/خروج جهاز تجريبي من الجمهور (باستخدام Debug + شروط واضحة).
* تحقّق من توافر الجمهور في Google Ads واستخدامه في مجموعة إعلانية.
* راقب الأداء: CTR/CPA/ROAS لكل جمهور، وعدّل النوافذ/الاستثناءات وفق النتائج.

الاستكشافات (Explorations في GA4)

## **#11) الاستكشافات (Explorations في GA4)**

**الاستفادة (Why it matters)**

* تحوّل البيانات الخام إلى إجابات عملية: أين يتسرّب مسار الشراء؟ أي قناة/مدينة/مقاس يضغط التحويل؟
* تساعدك على **تشخيص المشكلة بسرعة** بدل انتظار لوحات عامة قد تُخفي التفاصيل.
* تمكّنك من اختبار فرضياتك (قبل/بعد تجربة، قبل/بعد تعديل سعر/شحن) بدقائق.

**الهدف (What we want to achieve)**

* بناء مجموعة استكشافات “قوالب” جاهزة: **Funnels، Cohorts، Path، Segment Overlap، Free-form**.
* تفكيك النتائج حسب أبعاد مفيدة لعملك: **القناة، المدينة، شركة التوصيل، نوع الدفع، الفئة، المقاس**.
* اعتماد تسميات/حفظ Explorations كقوالب ثابتة للفريق (قابلة للتكرار أسبوعيًا).

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) Funnel Exploration (مسار الشراء)**

* **الخطوات المقترحة:**view\_item\_list → select\_item → view\_item → add\_to\_cart → begin\_checkout → add\_shipping\_info → add\_payment\_info → purchase
* **إعدادات مهمّة:**
  + **Closed vs Open Funnel:** استخدم Closed لقياس الالتزام، وOpen لاكتشاف دخول متأخر للخط.
  + **Exclusion Step:** استبعد خطوات “خارج المسار” لو لزم.
  + **Breakdown:** source/medium, city, delivery\_partner, payment\_type, item\_category.
* **ماذا تبحث عنه؟**
  + Drop-off كبير بين add\_payment\_info → purchase مع **payment\_type=COD**؟ راقب سياسة COD/رسالة الشحن.
  + فرق مدينة/شريك توصيل عند add\_shipping\_info يدلّ على عناوين/رسوم مربكة.
  + فئة بعينها تتسرّب بشدة؟ راجع صور/مقاسات/السعر.

### **B) Path Exploration (تحليل المسارات)**

* **وضعان مفيدان:**
  + **Forward from view\_item**: إلى أين يذهب المستخدم؟ (size\_guide\_view؟ add\_to\_cart؟ مشاركة؟)
  + **Reverse from purchase**: ما المسار الأكثر شيوعًا قبل الشراء؟ هل هناك صفحات “تُميت” المسار؟
* **استخدم فلاتر:** جهاز (iOS/Android)، قناة، مدينة.
* **مؤشرات سريعة:**
  + زيارة **size\_guide\_view** قبل add\_to\_cart قد ترفع التحويل—أبرز دليل المقاس.
  + قفزات إلى صفحات سياسة/شحن قبل الشراء تعني غموضًا بالرسوم أو زمن التوصيل.

### **C) Cohort Exploration (الاحتفاظ والتكرار)**

* **تجميعات مقترحة:**
  + على أساس **أول شراء** (Retention شراء D30/90).
  + على أساس **أول جلسة** (تفعيل المستخدمين الجدد).
* **تفكيك حسب:** قناة الاكتساب، payment\_type, فئة أول شراء (فساتين، أحذية…).
* **ماذا تبحث عنه؟**
  + قنوات تجلب **Retention** أعلى (حتى لو CAC أعلى).
  + مستخدمو COD: هل يعودون أسرع/أبطأ من Prepaid؟
  + فئة “فساتين السهرة”: هل تدفع لتكرار الشراء أم شراء موسمي فقط؟

### **D) Segment Overlap (تداخل الشرائح)**

* **أمثلة عملية:**
  + **Cart Abandoners ∩ preferred\_payment=cod ∩ city=sanaa** → حملة استرجاع مخصصة COD صنعاء.
  + **High AOV ∩ evening\_dresses** → عروض VIP/مجموعات منسّقة.
* **الفائدة:** تقليل الهدر عبر فهم التطابقات الحقيقية بين الاهتمامات والسلوك.

### **E) Free-form / Pivot (جداول مرنة سريعة)**

* **لوحات جاهزة كاستكشافات:**
  + **Search Performance:** search\_term × results\_count × add\_to\_cart\_rate → اكتشف *Zero-Results*.
  + **List Performance:** item\_list\_name × CTR(select\_item/view\_item\_list) × add\_to\_cart\_rate.
  + **Size Insights:** size × add\_to\_cart\_rate × out\_of\_stock\_rate → قرارات شراء مخزون.
  + **City/Partner Ops:** delivery\_city/delivery\_partner × RTO/On-time% (من BigQuery أو أبعاد الحالة إن سجّلتها).

### **F) أفضل الممارسات والحوكمة**

* **تسمية الاستكشافات:** EXP\_Funnel\_Zahraah\_Checkout\_v1, EXP\_Cohort\_FirstPurchase\_Q3…
* **حفظ Segments/Audiences** الناتجة لإعادة استخدامها في الحملات/التخصيص.
* **قلّل الأبعاد عالية التعدّد** داخل Exploration لتجنّب (other)/(not set).
* عند الحاجة للتفصيل الثقيل (مقاسات/مدن/شركاء كثيرين)، **انقل التحليل إلى BigQuery** ثم اعرض النتائج بلوحة Looker.

### **G) قراءات سريعة تُحفّز القرار**

* **Drop-off مرتفع عند add\_payment\_info لـ COD** → اختبر ترتيب طرق الدفع/شرح COD/رسوم عند الاستلام.
* **Zero-Results عالية لبحث “عبايات”** → أضِف فئة/كلمات مرادفة/تحسين فهرس.
* **مسارات تُظهر قفزة لسياسة الإرجاع قبل الشراء** → إبراز سياسة مختصرة في صفحة المنتج/الدفع.
* **Cohort قناة مؤثرين** أفضل D30 → زِد الميزانية وكرّر المحتوى الرابح.

### **H) QA سريعة**

* تأكّد من ظهور الأبعاد المخصّصة (city/partner/payment\_type) في الاستكشافات بعد تعريفها.
* اختبر عينات صغيرة أولًا (فلتر تاريخ/قناة) ثم وسّع.
* راقب إشارات **thresholding** أو **sampling**؛ عند ظهورها بكثرة انتقل لـBigQuery لنفس الأسئلة.

التقارير واللوحات (Dashboards & Reports)

## **#12) التقارير واللوحات (Dashboards & Reports)**

**الاستفادة (Why it matters)**

* تحوّل القياس إلى قرارات يومية: أين نضخ الميزانية؟ ماذا نُحسّن في المنتج؟ مَن الشريك أو المدينة التي ترفع RTO؟
* تقلّل “العمى التشغيلي” عبر لوحات مركّزة لكل فريق: الإدارة، التسويق، المنتج، العمليات، والتقنية.
* توحّد الحقيقة عبر BigQuery/GA4 بدل تقارير متعارضة.

**الهدف (What we want to achieve)**

* حزمة لوحات أساسية ثابتة، مع مؤشرات وحدود مراقبة (Guardrails) وتنبيهات.
* تقسيم واضح: ماذا يرى المدير التنفيذي يوميًا، وماذا يحتاج التسويق/المنتج/العمليات أسبوعيًا.
* خرائط مصدر بيانات لكل لوحة (GA4 مباشرةً أو BigQuery Views).

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) لوحة تنفيذية يومية (Executive Daily)**

**مؤشرات أساسية:** Revenue, Orders, AOV, CR, Active users, New users, Purchasers, Blended ROAS (7D), Channel mix (%), Top cities, **COD Success**, **RTO%, NDR%**, On-time Delivery%, Out-of-stock Rate, Crash Rate.  
**تفكيكات:** اليوم/7 أيام/28 يوم (Sparklines)، قناة، مدينة.  
**تنبيهات (Guardrails):**

* CR ↓ أكثر من 20% يوم-بيوم.
* COD Success ↓ دون 75%.
* RTO% ↑ فوق 18%.
* Crash Rate ↑ فوق 1.5%.  
  **المصدر:** GA4 (+ BigQuery لـ COD/RTO).

### **B) التسويق وROAS (Marketing & Acquisition)**

**مؤشرات:** Spend, Clicks, Installs, Sessions, Purchases, Revenue, **CPA/CPP**, **ROAS** (قناة/حملة/مجموعة).  
**تفكيكات:** source/medium/campaign/adset/ad, بلد/مدينة, جهاز.  
**تنبيهات:**

* ROAS قناة < 1.2 (7D) مع إنفاق > عتبة.
* قناة ترتفع فيها (not set) > 5%.  
  **المصدر:** GA4 (Google Ads linked) + **Cost Import/BigQuery** لغير Google.

### **C) المنتج والتجاري (Product & Merchandising)**

**مؤشرات:**

* قائمة/شبكة: Impressions, CTR(select\_item/view\_item\_list)
* PDP: ATC Rate, Size guide views
* سلة/شراء: CVR إلى purchase
* **Price/Discount Impact**, **Size Availability**, **Recommendation CTR**
* **Search**: Top queries, CTR من النتائج، **Zero-Results Rate  
  تفكيكات:** فئة/قائمة، مقاس/لون، حالة مخزون، سعر/خصم.  
  **تنبيهات:** Zero-Results Rate > 8%، ATC لفئة ينخفض > 25% أسبوعيًا.  
  **المصدر:** GA4 + BigQuery Views (fct\_search, fct\_items).

### **D) مسار الشراء والدفع (Funnel & Checkout)**

**مؤشرات:** نسب الانتقال خطوة-بخطوة، Drop-offs، Errors/Validation في الدفع، **Payment Type Split (COD/Prepaid)**.  
**تفكيكات:** جهاز، قناة، مدينة، **delivery\_partner**, **payment\_type**.  
**تنبيهات:** Drop-off عند add\_payment\_info→purchase ↑ فوق 35%، أو فرق iOS/Android > 10 نقاط مئوية.  
**المصدر:** GA4 Explorations + BigQuery (fct\_funnel\_steps).

### **E) العمليات اللوجستية (Operations: COD/RTO/NDR)**

**مؤشرات:** Shipped, Delivered, **cod\_collection=true**, **COD Success%**, **RTO%**, **NDR%**, Attempts/Order, Avg Delivery Days, SLA%.  
**تفكيكات:** مدينة، **Partner**، منطقة، نوع الدفع.  
**تنبيهات:**

* RTO% لمدينة/Partner > 22%.
* On-time Delivery% < 80%.
* Attempts/Order > 1.6.  
  **المصدر:** BigQuery (fct\_order\_status, delivery\_attempts) من أحداث السيرفر.

### **F) التقنية والأداء (Tech Performance)**

**مؤشرات:** Crash Rate, ANR, **App start time**, API error rate, Image load time، أثر الأداء على CR.  
**تفكيكات:** إصدار التطبيق، جهاز/نظام، شاشة.  
**تنبيهات:** Crash > 1.5% أو App start > 2.5s.  
**المصدر:** Firebase Crashlytics/Performance + ربط مع GA4.

### **G) الاحتفاظ والقيمة العمرية (Retention & LTV)**

**مؤشرات:** D1/D7/D30 Retention، **Repeat Rate**, **LTV** حسب قناة/فئة أول شراء/نوع الدفع.  
**تنبيهات:** هبوط Retention D7 لقناة رئيسية > 5 نقاط.  
**المصدر:** GA4 Cohorts + BigQuery (حساب LTV).

### **H) الرسائل والإشعارات (CRM/Push)**

**مؤشرات:** Sends, Deliveries, Open Rate, CTR, **Purchase Uplift** بعد الرسالة (Holdout بسيط إن أمكن).  
**تفكيكات:** جمهور، قناة (Push/Email/SMS)، توقيت.  
**تنبيهات:** CTR < 1% لحملة كبرى، أو ارتفاع Unsubscribe.  
**المصدر:** منصة الرسائل + ربط لـ GA4/BigQuery.

### **I) التجارب (Experiments)**

**مؤشرات:** اسم التجربة، Variant A/B، **المؤشر الأساسي** (CR/AOV)، **حجم العيّنة/المدة**، **Lift**، Guardrails (Crash/Latency).  
**تنبيهات:** توقف جمع البيانات، أو تجاوز Guardrail.  
**المصدر:** Firebase A/B + قراءة الأثر في GA4/BigQuery.

### **J) خرائط مصدر البيانات (مختصر)**

* **GA4 مباشرةً:** اكتساب، قنوات، مسارات أساسية.
* **BigQuery:** COD/RTO/NDR، بحث متقدم، مقاسات/ألوان، ROAS مجمّع (مع تكاليف متعددة المنصّات)، LTV.
* **Firebase:** الأعطال والأداء + ربطها بالتأثير على التحويل.

### **K) حوكمة اللوحات**

* تسمية قياسية: DB\_Exec\_Daily, DB\_Mkt\_ROAS, DB\_Product\_Search, DB\_Ops\_COD\_RTO, …
* وصف لكل لوحة: الغرض، المالك، تكرار المراجعة، قائمة المؤشرات، مصادر البيانات.
* **مقارنات زمنية ثابتة**: Day/7D/28D، ومرشّحات قناة/مدينة/Partner/Device.

### **L) QA سريعة للّوحات**

* اتساق القيم بين GA4 ولوحات BigQuery (±5–10%).
* لا يوجد (not set) أو (other) مرتفع في أبعاد رئيسية (>5%).
* التواريخ/المنطقة الزمنية = **Asia/Aden**، والعملة موحّدة في العرض.
* تنبيهات تعمل وتصل للمالكين (بريد/Slack…).

تكلفة القنوات وROAS المجمّع

## **#13) تكلفة القنوات وROAS المجمّع**

**الاستفادة (Why it matters)**

* “مصدر حقيقة واحد” للإنفاق والعائد عبر كل القنوات (Google/Meta/TikTok/Snap/Influencers/Email…).
* توزيع ميزانية أدق: تعرف أين تضع كل ريال ليعطي أعلى **عائد على الإنفاق الإعلاني (ROAS)**.
* تقليل تضارب تقارير المنصّات (platform vs GA4) عبر تعريفات موحّدة ونوافذ إسناد واضحة.

**الهدف (What we want to achieve)**

* توحيد **تكاليف الإعلانات** وإقرانها بزيارات/تحويلات GA4 على نفس أبعاد التسمية (source/medium/campaign/adset/ad).
* حساب نوعين من ROAS يناسبان سوقك:
  + **Booked ROAS** (عند purchase).
  + **Realized ROAS** خاص بـCOD (العائد المحقَّق بعد cod\_collection=true أو بعد تسليم الطلب).
* اعتماد **عملة موحّدة** ومنطقة زمنية واحدة (Asia/Aden) مع تحويل عملات تلقائي.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) أين نخزّن التكلفة وكيف نأتي بها؟**

* **Google Ads**: تُسحب تلقائيًا بعد الربط مع GA4.
* **باقي القنوات** (Meta/TikTok/Snap/Influencers/Networks/Email):
  + إمّا **Data Import** إلى GA4 بملف يومي.
  + أو (موصى به) **BigQuery**: تخزّن كل التكاليف في جدول موحّد ثم تُحتسب ROAS في View/لوحة.

**مخطط جدول التكلفة الموحّد (BigQuery)**

date, platform, source, medium, campaign, adset, ad, country, currency, spend, impressions, clicks

* حافظ على نفس **قاموس UTM** (#7) لضمان التطابق.

### **B) التطابق (Matching) بين التكلفة والتحويل**

* **مفاتيح الربط**: date, source, medium, campaign (+ adset, ad إن توافرت في GA4/UTM).
* استخدم **Data-driven Attribution** داخل GA4 لقيمة الإيراد المنسوبة، أو احتسب توزيعات الإسناد في BigQuery إن رغبت.
* إذا اختلفت تسمية الحملات بين المنصّات وUTM، أنشئ **جدول خرائط** (Mapping) لتوحيد الأسماء.

### **C) عملات متعددة وتحويل العملة**

* خزّن currency وspend الأصلية.

حوِّل إلى **عملة العرض** (مثل YER) عبر **FX Table** يومي داخل BigQuery:  
spend\_converted = spend \* fx\_rate(currency -> YER on date)

* افعل الشيء نفسه لقيمة الإيراد إن كانت تُرسل بعملات متنوّعة.

### **D) تعريفات ROAS الموحّدة**

**Booked ROAS** (افتراضي للتسويق):  
booked\_roas = revenue\_attributed\_to\_channel / spend\_converted

* **Realized ROAS (COD)** (تحليلي/تشغيلي):
  + إمّا استخدام إيراد الطلبات **الـDelivered & cod\_collection=true** فقط،
  + أو ضرب إيراد COD في **نسبة التحصيل المتحقّقة** (per city/partner) كعامل تصحيحي:

realized\_revenue = revenue\_cod \* cod\_success\_rate

realized\_roas = (realized\_revenue + revenue\_prepay) / spend\_converted

### **E) نوافذ ونِسب الإسناد (Attribution Windows)**

* ثبّت نافذة موحّدة (مثلاً 7–30 يوم نقرة)، و**استخدم نفس النافذة في كل القراءة**.
* وضّح الفرق بين:
  + **GA4** (multi-touch, data-driven)
  + **منصة الإعلانات** (غالبًا last-click/last-touch خاص بها)
* اعتمد GA4/BigQuery كمرجع عند اتخاذ قرارات الميزانية، مع مراقبة فروق المنصّات لأغراض التوافق.

### **F) مؤشرات إضافية بجانب ROAS**

* **CAC/CPP**: تكلفة لكل مشتري/طلب.
* **Spend Mix %**: توزيع الإنفاق بين القنوات.
* **Incrementality** (اختياري لاحقًا): قنوات ذات رفع حقيقي (Lift) عبر holdout/Geo-test.
* **Cost per Add-to-Cart / per Begin Checkout** لتحسين أعلى القمع.

### **G) مؤشرات جودة وQA**

* % من التكلفة **غير المطابقة** لأي حملة في GA4 < 3%.
* نسبة (not set) في تقارير الاكتساب < 5% (وإلا هناك تسريب UTM).
* لا توجد أيام **spend=0** لقناة فعّالة (إن ظهرت → فحص خط السحب).
* فروق العملة/المنطقة الزمنية مصحّحة (Asia/Aden + عملة واحدة للعرض).

### **H) مخرجات عملية**

**View** في BigQuery: vw\_channel\_perf يعرض يوميًا:  
date, source, medium, campaign, adset, ad,

spend\_converted, clicks, sessions, purchases, booked\_revenue, booked\_roas,

realized\_revenue (COD-adjusted), realized\_roas

* **لوحة ROAS موحّدة** (Looker Studio):
  + تبويبات: إجمالي/قناة/حملة/مدينة، تبديل بين Booked vs Realized.
  + فلاتر: الجهاز، نوع الدفع، المدينة، الشريك.

### **I) أخطاء شائعة وكيف نتفاداها**

* **ازدواج التحويل** بين SDK وCAPI → استخدم transaction\_id/event\_id للـdedup.
* تغيّر أسماء الحملات بين المنصّات وUTM → طبّق **Mapping** ثابت.
* عدم شمول المؤثّرين/الكوبونات في التكلفة → سجّلهم كـ source=instagram, medium=influencer, content=@handle.
* احتساب ROAS لـCOD على قيم “محجوزة” فقط دون تصحيح التحصيل → اعرض أيضًا **Realized ROAS**.

التنبيهات (Custom Insights & Alerts)

## **#14) التنبيهات (Custom Insights & Alerts)**

**الاستفادة (Why it matters)**

* تكتشف المشاكل فورًا بدل اكتشافها متأخرًا في التقارير (هبوط CR، تعطل حدث، ارتفاع RTO/Crash).
* توجّه الفريق مباشرةً إلى *سبب المشكلة* وتُقلّل الهدر (ميزانية/وقت) وتمنع فقدان مبيعات.

**الهدف (What we want to achieve)**

* مجموعة **تنبيهات قياسية** تغطي: التحويل/الإيراد، القنوات والإنفاق، مسار الشراء، COD/RTO، الأداء التقني، وصحة جمع البيانات.
* قنوات إشعار واضحة (Email/Slack) مع **مالك لكل تنبيه** و**Runbook** مختصر: ماذا نفعل خلال 60 دقيقة.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) تنبيهات التحويل والإيراد (GA4 Custom Insights)**

* **CR Drop**: *Conversion Rate* ↓ أكثر من **20%** مقارنة بمتوسط آخر 7 أيام، خلال نافذة 2–4 ساعات.
* **Revenue Drop**: *Revenue* ↓ أكثر من **25%** يوم-بيوم (حتى مع ثبات الجلسات).
* **Purchase Count Anomaly**: انخفاض غير طبيعي في purchase (استخدم Insight من نوع Anomaly/Condition).
* **AOV Spike/Drop**: تغير **±15%** قد يشير لخصم/خلل تسعير.

**التفعيل (مختصر):** من لوحة التقارير في GA4 → View all insights → Create → اختر الحدث/المؤشر والشرط والفترة وقناة الإرسال.

### **B) تنبيهات القنوات والإنفاق (GA4 + BigQuery)**

* **ROAS < حد**: ROAS قناة رئيسية < **1.2** (7D) مع إنفاق > **$200**/يوم.
* **Spend Anomaly**: إنفاق قناة ↑ **+50%** مقابل متوسط 7 أيام دون زيادة مقابلة في purchase.
* **(not set) Surge**: نسبة الزيارات بدون UTM > **5%**.

**التفعيل:**

* في GA4 (للـROAS Google Ads) + **تنبيه BigQuery/تقرير مجدول** لباقي القنوات بعد استيراد التكلفة.

### **C) مسار الشراء (Funnels)**

* **Drop-off Spike** عند add\_payment\_info → purchase > **35%** (ساعة/يوم).
* **Zero-Results Rate** للبحث > **8%** خلال يوم.

**التفعيل:** Custom Insight على نسب الانتقال/معدل zero\_results (يُحتسب من حدث search أو View في BigQuery).

### **D) COD/RTO (أحداث السيرفر)**

* **COD Success** ↓ دون **75%** يوميًا.
* **RTO%** ↑ فوق **22%** لأي مدينة/Partner.
* **Attempts/Order** > **1.6** لمدينة/Partner (علامة NDR).
* **On-time Delivery%** < **80%**.

**التفعيل:** عبر **Scheduled Query** في BigQuery + إشعار Email/Slack (أو لوكر ستوديو Alerts إن استخدمت موصل BigQuery).

### **E) الأداء التقني (Firebase + GA4)**

* **Crash Rate** > **1.5%** لإصدار جديد.
* **App Start Time** > **2.5s** (متوسط آخر 2 ساعة).
* **API Error Rate** > **3%** في مسار الدفع.
* **Image Load Time** ↑ يلامس 3s في PDP.

**التفعيل:** Firebase Crashlytics/Performance Alerts + لوحة Performance، وربطها بإشعار Slack/Email.

### **F) صحّة جمع البيانات (Instrumentation Health)**

* **اختفاء حدث رئيسي** (مثل add\_to\_cart أو purchase) لمدة **≥30 دقيقة**.
* **Mismatch**: فرق بين إيراد GA4 وإيراد النظام الخلفي يتجاوز **10%** في أي يوم.
* **Custom Dimension Exhaustion**: الاقتراب من حدود GA4 للأبعاد (تحذير مبكّر).

**التفعيل:**

* GA4 Custom Insight لغياب الحدث.
* BigQuery مقارنة يومية GA4 vs Backend.
* مراجعة شهرية لتعريفات GA4.

### **G) قنوات الإشعار والملكية (Who/Where)**

* **التسويق**: ROAS/Spend/UTM → قناة Slack #mkt-alerts + Email لمدير التسويق.
* **المنتج/التقنية**: CR/Drop-offs/Instrumentation → #product-ops.
* **العمليات**: COD/RTO/On-time → #ops-logistics.
* **الـSRE/التطبيق**: Crash/Latency/API Errors → #app-health.

لكل تنبيه **مالك** + **Runbook** (3–5 خطوات سريعة): التحقق، السبب المحتمل، إجراءات فورية، متى نتصعيد.

### **H) تخفيف الضوضاء (Noise Control)**

* **جداول زمنية**: قلّل تنبيهات منتصف الليل غير الحرجة.
* **عتبات ديناميكية**: استخدم متوسط 7/14 يوم بدل أرقام ثابتة عند اللزوم.
* **Dedup**: اجمع تنبيهات نفس الفئة في رسالة واحدة بالساعة.

### **I) سريعة QA**

* اختبارات وهمية: خفّض purchase على بيئة Staging للتحقق من وصول التنبيه.
* راقب **معدل الدقة**: هل 80% من التنبيهات مفيدة؟ إن كان أقل → عدّل العتبات.
* وثّق كل تنبيه: الهدف، الصيغة، المالك، قناة الإرسال، خطوات المعالجة.

الجودة والحوكمة (Governance & Data Quality)

## **#15) الجودة والحوكمة (Governance & Data Quality)**

**الاستفادة (Why it matters)**

* بيانات موثوقة “من أول مرة” بدون ضوضاء أو تكرار أو اختلاف تعريفات.
* تسليم أسرع للقرارات (لوحات ثابتة، تنبيهات دقيقة)، وتقليل أعطال القياس وكلفته.
* التزام بالخصوصية والأمان وتتبّع قابل للمراجعة والتدقيق.

**الهدف (What we want to achieve)**

* معايير تسمية وإصدارات واضحة للأحداث والمعلمات، مع دليل استخدام واحد للفِرق.
* أذونات وصول “أقل امتياز” على GA4/BigQuery، وتتبّع تغييرات مُوثَّق.
* سياسات خصوصية واحتفاظ بيانات متوافقة محليًا، وفلترة صارمة للزيارات الداخلية.
* إجراءات QA قبل/بعد كل إطلاق، وتنبيهات لصحة القياس.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) معايير التسمية (Naming Conventions)**

* **الأحداث**: أفعال بصيغة إنجليزية وsnake\_case (مثال: add\_to\_cart, order\_status\_update).
* **المعلمات**: snake\_case، بدون مسافات/حروف خاصة؛ قوائم قيَم ثابتة (enumerations) بالأحرف الصغيرة (cod, prepaid, …).
* **الخصائص**: 3–5 فقط (مثال: preferred\_payment, favorite\_size).
* **UTM**: قاموس ثابت لـ source/medium/campaign/... (يرتبط بما عرّفناه في #7).

### **B) إدارة الإصدارات (Schema Versioning & Change Management)**

* أضِف schema\_version كـ **event param** عالمي؛ ابدأ بـ 1.
* **Changelog** مركزي: ماذا تغيّر؟ لماذا؟ لمن التأثير؟ كيف نختبر؟
* سياسة إزالة/استبدال: أعلن **Deprecated** أولًا، ثم أزِل بعد دورة تقرير متفق عليها.

### **C) الوصول والأمان (Access & Security)**

* **GA4**: أدوار **Viewer/Analyst** للأغلبية؛ **Editor** لعدد محدود؛ **Admin** لشخصين كحد أقصى.
* **BigQuery**: مبدأ أقل امتياز (Viewer، Job User، Editor لجداول وسيطة فقط).
* حسابات خدمية منفصلة لـ Measurement Protocol/ETL؛ تدوير مفاتيح كل 90 يومًا.
* أسرار/مفاتيح في خزنة آمنة (Secret Manager).
* **عدم إرسال PII** داخل GA4/MP (لا هاتف، لا عنوان نصي).

### **D) الخصوصية والامتثال (Privacy & Retention)**

* **Retention** في GA4: 14 شهرًا.
* **Timezone/عملة**: Asia/Aden + YER (أرسل currency بكل حدث شراء).
* **Consent Mode/ATT** حسب المتطلبات المحلية؛ زر “الخصوصية” واضح للمستخدم.
* **حق الحذف**: عملية موثّقة لحذف بيانات مستخدم عند الطلب (User Deletion API/أدواتك).

### **E) فلاتر الصحة والاختبار**

* **Internal/Developer traffic** مفعّل + قائمة أجهزة/حسابات اختبار.
* **Bot filtering** قيد التشغيل.
* وسم بيئات **staging** (param environment=staging) مع استبعادها من تقارير الإنتاج.

### **F) توثيق مركزي (Single Source of Truth)**

* **Measurement Plan**: جدول أحداث/معلمات/خصائص/تحويلات + أمثلة Payload + أسباب القياس.
* **Data Dictionary**: تعريف كل بُعد/مقياس وصيغ الحساب (CR, AOV, COD Success…).
* **Runbooks**: ماذا نفعل عند هبوط CR/ROAS أو اختفاء حدث/ارتفاع RTO؟ (3–5 خطوات).
* **Mapping Tables**: توحيد أسماء الحملات/القنوات عند الحاجة.

### **G) QA قبل/بعد الإطلاق (Release Checklist)**

* **قبل**: DebugView لرحلة شراء كاملة، تحقق من items[]/currency/value، ومطابقة عينة إيراد مع النظام الخلفي (±5–10%).
* **بعد** (24–48 ساعة): مراجعة فروقات الإيراد، التحقق من عدم تكرار purchase، عدم ارتفاع (not set) و(other).

### **H) مراقبة جودة القياس (Monitoring)**

* **Custom Insights**: اختفاء add\_to\_cart/purchase ≥ 30 دقيقة؛ قفزة (not set) > 5%.
* **BigQuery**: فحوص يومية للفجوات/التكرار + مقارنة GA4 مقابل Backend.
* **Cardinality Control**: راقب انفجار تنوّع القيم في الأبعاد (تجنّب كلمات حرّة).

### **I) حوكمة التكلفة والأداء (Cost & Performance)**

* **BigQuery**: تقسيم الجداول على event\_date وعنقدة على event\_name/platform/source؛ انتهاء صلاحية جداول مؤقتة؛ حدود ميزانية للاستعلامات.
* جداول **Cost Import** مجدولة وموحّدة الحقول؛ تنبيهات عند فشل التحديث.

### **J) التدقيق والتغييرات (Auditability)**

* استخدم **Change History** في GA4 دوريًا.
* أي تعديل (تحويل/تعريف مخصّص/فِلتر) يوثّق في Changelog ويُراجع من شخص ثانٍ (4-eyes principle).

### **K) معايير قبول (QA سريعة لهذه النقطة)**

* وجود مستندات: Measurement Plan + Data Dictionary + UTM Guide + Changelog + Runbooks.
* تفعيل الفلاتر/الأذونات/الاحتفاظ كما هو موثّق.
* آخر إطلاق اجتاز قائمة QA مع سجّل اختبار محفوظ.
* تنبيهات الصحة تصل للمالكين الصحيحين وتُتبع بإجراءات.

الاختبارات والتحسين (Experimentation

## **#16) الاختبارات والتحسين (Experimentation عبر Firebase A/B + Remote Config)**

**الاستفادة (Why it matters)**

* تثبت “سببية” الأثر قبل التعميم: نطلق الرابح فقط → **تحويل أعلى وهدر أقل**.
* سرعة تعلُّم منظّمة: كل تجربة تُصبح معرفة مُوثّقة للفريق.
* تخفّض المخاطر بوجود **حواجز/حماية** (Crash/Latency/أخطاء دفع).

**الهدف (What we want to achieve)**

* تشغيل إطار اختبارات خفيف باستخدام **Firebase A/B + Remote Config** مع قراءة النتائج في **GA4/BigQuery**.
* لكل تجربة: فرضية واضحة، جمهور مستهدف، مقاييس رئيسية، **Guardrails**، وحُكم إطلاق مبني على بيانات.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) الأدوات والإعداد**

* **Remote Config**: مفاتيح تتحكم بالواجهة/السلوك (ترتيب طرق الدفع، حد الشحن المجاني، تصميم PDP…).
* **Firebase A/B Testing**: يوزّع الترافيك A/B ويحاسب النتائج تلقائيًا.
* **Sticky Assignment**: تثبيت المستخدم على نفس المتغير طوال التجربة.
* **تسمية موحّدة**: EXP\_<area>\_<hypothesis>\_v1.

### **B) تصميم التجربة (Hypothesis → Variants → Targeting)**

* **صياغة الفرضية**:  
  “إذا وضعنا COD أول خيار دفع لمستخدمي المدن ذات RTO المنخفض، فسنرفع CR بنسبة +10% لأن…”
* **المتغيرات**: A=تحكم، B=تجربة (يمكن C إن لزم، ابدأ بسيطًا).
* **الاستهداف**: منصة (iOS/Android)، مدينة/شريحة (preferred\_payment=cod)، مستخدم جديد/عائد.
* **الرَفع التدريجي (Ramp)**: 1% → 10% → 50% → 100% بعد مرور فحوص الصحة.

### **C) المقاييس**

* **رئيسية (واحدة فقط غالبًا)**:
  + إما **CR to purchase**، أو **Revenue per user (RPU)** عندما تؤثر التغييرات على السلة/السعر.
* **مسانِدة**: begin\_checkout, add\_payment\_info, **AOV**, **Add-to-cart rate**.
* **Guardrails (حواجز حماية)**:
  + **Crash Rate** < 1.5%، **App start time** < 2.5s، **API error rate (checkout)** < 3%.
  + (اختياري) **زيادة RTO** لا تتجاوز +2 نقاط مئوية لمدن التجربة (يُقرأ لاحقًا من أحداث السيرفر).

### **D) الحجم والمدة (Rule-of-thumb)**

* لا توقف قبل **دورة أسبوع كاملة** (تأثير أيام الأسبوع).
* تقدير سريع لحجم العينة (اختبار نسبتين، α=5%، قوة≈80%):  
  nلكل ذراع≈16⋅p(1−p)Δ2nلكل ذراع​≈Δ216⋅p(1−p)​  
  مثال: **CR=2%** ونستهدف **+10% نسبي** (Δ=0.002) ⇒ 16×0.02×0.98/0.000004≈78,40016×0.02×0.98/0.000004≈78,400 مستخدم/ذراع.  
  مثال: **CR=3%** و\*\*+10% نسبي\*\* (Δ=0.003) ⇒ ~**51,700**/ذراع.
* إن كانت العينة أصغر، استخدم **مقاييس أقرب للقُمع** (مثل add\_to\_cart أو begin\_checkout) كإشارات مبكرة، وواصل جمع الشراء حتى تكفي العينة.

### **E) التحليل والحُكم**

* **Lift** = (Metric\_B / Metric\_A − 1).
* استخدم **GA4 Explorations** أو **BigQuery** لحساب الفروق وفواصل الثقة.
* **اطلق** فقط إذا:
  + الحد الأدنى لفاصل الثقة لـ**المقياس الرئيسي** > 0،
  + **كل الحواجز** ضمن الحدود،
  + لا وجود **SRM** (اختلال نسبة العيّنة): تحقق أن توزيع A/B قريب من المخطط.

### **F) أمثلة تجارب مناسبة لتطبيقك**

1. **ترتيب طرق الدفع**: COD أولًا مقابل ثانيًا.
   * رئيسي: CR، مساند: add\_payment\_info, AOV.
2. **حد الشحن المجاني**: 0 مقابل 10,000 YER.
   * رئيسي: RPU/AOV، مساند: CR، Guardrail: Margin.
3. **PDP**: إبراز **جدول المقاسات** كبانر أعلى الصور.
   * رئيسي: Add-to-cart، مساند: CR، Guardrail: Latency للصور.
4. **نتائج البحث**: ترتيب “الأكثر توافرًا بالمقاسات” بدل “الأكثر مشاهدة”.
   * رئيسي: Add-to-cart من البحث، مساند: Zero-results rate.

### **G) تشغيل التنبيهات أثناء التجربة**

* Custom Insights: هبوط CR > 20%، ارتفاع Crash > 1.5%، زيادة زمن البدء.
* تنبيه BigQuery (أسبوعي) إن ارتفع **RTO** في مدن التجربة > +2 نقاط مئوية.

### **H) QA قبل/أثناء/بعد**

* **قبل**: تأكد من قراءات DebugView للحدث/المعلمات، وثبات التخصيص لكل مستخدم.
* **أثناء**: راقب SRM، الحواجز، وانخفاضات غير طبيعية.
* **بعد**: وثّق النتيجة، حدّث **Runbook التصميم**، و**طبّق الفائز** عبر Remote Config للجميع.

### **I) قالب موجز تجربة (Experiment Brief)**

* الاسم: EXP\_checkout\_payment\_priority\_v1
* الفرضية: …
* الجمهور: …
* المقاييس: رئيسي …، مساندة …، Guardrails …
* خطة الرَفع: 1% → 10% → 50% → 100%
* المدة المستهدفة: أسبوع/أسبوعان (حتى بلوغ العينة)
* معايير الإطلاق/الإلغاء: …
* المالك/المراجِع: … / …
* ملاحظات التنفيذ: مفاتيح Remote Config، Screens المتأثرة.

ضمان الجودة وDebug (QA & Debug)

## **#17) ضمان الجودة وDebug (QA & Debug)**

**الاستفادة (Why it matters)**

* تضمن أن الأرقام التي تراها **صحيحة من أول يوم** (بدون ازدواج/نقص).
* تكتشف الأعطال فورًا: حدث مكسور، قيمة ناقصة، عملة خاطئة، أو تضارب purchase.
* تقلّل وقت التتبع اليدوي وتمنح ثقة للإدارة في القرارات.

**الهدف (What we want to achieve)**

* إطار **QA ثابت** قبل/بعد كل إطلاق، مع سيناريوهات اختبار تغطي مسار الشراء كاملًا وCOD بعد الشراء.
* أدوات Debug واضحة للمطورين (Realtime, DebugView, سجلات الجهاز) + مصالحة يومية مع النظام الخلفي.
* تنبيهات لصحة القياس وتحرّكات تلقائية عند الانحراف.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) سيناريوهات QA الأساسية (قبل كل إطلاق)**

1. **رحلة شراء كاملة** (iOS وAndroid):  
   view\_item\_list → select\_item → view\_item → add\_to\_cart → begin\_checkout → add\_shipping\_info → add\_payment\_info → purchase
   * تحقق من: currency, value, items[] (item\_id, price, quantity, item\_category).
2. **التسعير/العملة**:
   * خصم/كوبون، شحن/ضريبة، عملات مختلفة إن وُجدت.
   * **قاعدة**: value ≈ sum(items) + shipping + tax − discounts.
3. **COD vs Prepaid**:
   * payment\_type يصل صحيحًا في add\_payment\_info وpurchase.
   * لا يحدث **refund** للـCOD (يُدار كسيرفر بعد الشراء).
4. **عدم ازدواج purchase**:
   * إعادة فتح شاشة النجاح/ضعف الشبكة/إعادة المحاولة لا تُرسل الحدث ثانية.
   * تحقّق من **idempotency** على السيرفر (عدم إرسال Purchase من جهتين لنفس الطلب).
5. **UTM & Deep Links**:
   * رابط إعلان يفتح **صفحة المنتج في التطبيق** ويحمل نفس UTM (Deferred deep link بعد التثبيت).
   * ظهور المصدر في **Realtime/Traffic Acquisition**.
6. **أحداث السيرفر بعد الشراء**:
   * order\_status\_update (shipped→delivered) وcod\_collection=true لمحاكاة التسليم.
   * ظهور الأبعاد: delivery\_partner, delivery\_city, order\_status.
7. **فلاتر البيئة/الداخلية**:
   * أجهزة الاختبار **مستبعدة** من تقارير الإنتاج (تظهر فقط في Debug/Realtime).
8. **الأخطاء الشائعة**:
   * item\_id أو price مفقودان، currency غير مرسلة، قيم عربية في أسماء **المفاتيح** (القيم فقط يمكن تعريبها).

### **B) أدوات وأوضاع Debug الموصى بها**

* **GA4 Realtime + DebugView**: راقب تسلسل الأحداث ومعلماتها لحظيًا.
* **Firebase Debug Logging**:
  + **Android**:
    - تفعيل وضع تتبع فيربيز:  
      adb shell setprop debug.firebase.analytics.app com.zahraah.app
    - سجلات تفصيلية (اختياري):  
      adb shell setprop log.tag.FA VERBOSE و adb shell setprop log.tag.FA-SVC VERBOSE
  + **iOS (Xcode)**: تشغيل التطبيق مع وسائط **-FIRAnalyticsDebugEnabled** أو **-FIRDebugEnabled** لرؤية سجلات التتبّع.
* **Event Inspector** في Firebase/GA4: تأكد من وصول المعلمات المخصّصة وظهورها في **Custom definitions**.
* **Looker Studio (لوحة QA)**: جداول يومية ترصد اختفاء حدث/قفزة (not set)/فروقات الإيراد.

### **C) مصالحة البيانات (GA4 ↔ النظام الخلفي)**

* **تسامح قياسي**: فرق **±5–10%** يوميًا (توقيات/إلغاءات/فلترة داخلية).
* **تقارير المصالحة** (BigQuery Views):
  1. daily\_backend\_revenue مقابل daily\_ga4\_revenue
  2. orders\_count مقابل purchase\_count
  3. تفكيك حسب payment\_type، قناة، منصة.
* **إن تجاوز الفرق العتبة**: شغّل **Runbook**:
  1. هل اختفى حدث (Custom Insight)؟
  2. هل تغيّر Schema/Param؟
  3. هل هناك ازدواج (SDK + Server)؟
  4. هل UTM تسربت إلى (not set)؟

### **D) قائمة فحص ثابتة (Checklist)**

* purchase يظهر **مرة واحدة** فقط لكل transaction\_id.
* كل أحداث السلة/الدفع تحمل currency و(إن أمكن) value.
* items[] مكتملة (لا نقص في item\_id, price, quantity).
* payment\_type, delivery\_city, delivery\_partner موجودة حيث لزم.
* **UTM** سليمة وتظهر في الاكتساب، و\*\*(not set)\*\* < 5%.
* الأجهزة/الحسابات الداخلية **مستبعدة**.
* حدود **Custom definitions** غير متخمة (راقب الاستخدام).
* **BigQuery Export** فعّال والجداول محدّثة.
* تنبيهات الصحة (اختفاء حدث/هبوط CR) مفعّلة وتصل للمالكين.

### **E) اختبارات خاصة بالبحث والمخزون**

* **Search**: search\_term, results\_count، وقاعدة حساب **Zero-Results Rate**.
* **المخزون/المقاسات**: stock\_status في view\_item/add\_to\_cart، وsize/color حيث يلزم—تحليل أثر النفاد على التحويل.

### **F) مكافحة الأخطاء الشائعة**

* **مفاتيح بالعربية**: ممنوع (استخدم الإنجليزية وsnake\_case).
* **انفجار تنوّع القيم** (Cardinality): لا تستخدم نصوصًا حرّة كقيم أبعاد رئيسية.
* **PII**: لا ترسل هاتف/عنوان/بريد في GA4/MP.
* **الطابع الزمني**: استخدم وقت الحدث الحقيقي للأحداث السيرفرية، لا وقت المعالجة.

### **G) معايير قبول لهذه النقطة**

* وثيقة **QA Runbook** قصيرة (صفحتان) + Checklist موقّع قبل كل إصدار.
* لوحة **QA** تعرض يوميًا: وجود الأحداث، (not set), فروق الإيراد، واستخدام الأبعاد المخصّصة.
* سجل اختبارات DebugView (صور/IDs) محفوظ في مستودع الفريق.

مصدر الحقيقة والتحليلات المتقدمة (& Analytics)

## **#18) مصدر الحقيقة والتحليلات المتقدمة (BigQuery Modeling & Analytics)**

**الاستفادة (Why it matters)**

* يجعل **BigQuery** مرجعًا موحّدًا لكل شيء (App + سيرفر + تكلفة + لوجستيات) بدل تقارير متفرّقة.
* يتيح تحليلات لا يوفّرها GA4 جاهزًا: **COD/RTO/NDR، بحث، مقاسات، LTV، ROAS مجمّع**.
* سرعة قرارات: لوحات مستقرة، تنبيهات دقيقة، ونماذج تنبؤية خفيفة (RTO/Churn).

**الهدف (What we want to achieve)**

* نمذجة بيانات على شكل **نجمي (Star Schema)**: جداول وقائع (Facts) وأبعاد (Dims) واضحة.
* طبقة **Views** جاهزة للوحات (ROAS، المسار، البحث، المقاسات، اللوجستيات).
* تشغيل ETL/ELT مجدول، قليل التكلفة، مع ضمان جودة واختبارات تلقائية.

**التفاصيل (How to do it — مختصر ومنظّم)**

### **A) نمذجة الجداول (اقتراح عملي)**

**وقائع (Facts):**

1. fct\_events\_ga4 (من Export) — أساس التحليلات العامة.
2. fct\_purchases — تفكيك items[] بحيث كل سطر = عنصر داخل الطلب.
   * مفاتيح: transaction\_id, item\_id, price, quantity, currency, payment\_type.
3. fct\_sessions — تلخيص جلسات (زيارة، قناة، جهاز).
4. fct\_order\_status — من أحداث السيرفر (#9): حالة الطلب عبر الزمن.
5. fct\_delivery\_attempts — محاولات التسليم (رقم/نتيجة/سبب).
6. fct\_cost — إنفاق القنوات (Google + غيره) بعد التوحيد.
7. fct\_search — مصطلحات البحث، عدد النتائج، CTR، zero-results.
8. fct\_notifications — إرسال/فتح/نقر رسائل (إن رُبطت منصة الرسائل).
9. (اختياري) fct\_app\_perf — مقاييس الأداء (زمن البدء، أخطاء API…).

**أبعاد (Dims):**

* dim\_date، dim\_item (فئة/ماركة/موسم)، dim\_city, dim\_delivery\_partner, dim\_campaign (من UTM)، dim\_user*(بدون PII)*.

**مفاتيح ربط موحّدة:** transaction\_id, order\_id, user\_id (إن وُجد)، app\_instance\_id, ومعرّفات الحملة وفق قاموس UTM (#7).

### **B) مسار البيانات (ETL/ELT)**

* **المصدر**: جداول GA4 المصدّرة يوميًا + Payload السيرفر (MP) + ملفات التكلفة + (اختياري) CRM.
* **خطوات أساسية:**
  1. **Standardize** العملة → عمود value\_yer عبر جدول **FX** يومي.
  2. **Flatten** مصفوفة items[] في الشراء إلى fct\_purchases.
  3. **Dedup/Idempotency** على transaction\_id وorder\_id+status.
  4. **Map** القنوات/الحملات لجدول dim\_campaign (حل اختلاف الأسماء).
  5. **Join** حالات الطلب ومحاولات التسليم لتحضير مؤشرات COD/RTO/NDR.
  6. إنشاء **Views مادية (Materialized Views)** لقراءات متكررة (قنوات/ROAS، مسار، بحث…).
* **الجدولة**: استعلامات يومية + تحديثات كل ساعة لضروريات التسويق (اختياري).

### **C) طبقة الـViews (جاهزة للوحات)**

1. vw\_channel\_perf — إنفاق، جلسات، مشتريات، **Booked ROAS** و**Realized ROAS (COD)**.
2. vw\_funnel — نسب الانتقال خطوة بخطوة + Drop-offs حسب قناة/جهاز/مدينة.
3. vw\_search — أفضل/أسوأ استعلامات، **Zero-Results Rate**، CTR، مساهمة في ATC/Purchase.
4. vw\_size\_curve — توزيع المقاسات مبيعًا/إضافة للسلة + **Out-of-stock impact**.
5. vw\_cod\_ops — **COD Success%، RTO%، NDR%، Attempts/Order، On-time%** حسب مدينة/Partner.
6. vw\_retention\_cohort — احتفاظ شراء D30/90 حسب القناة/الفئة/نوع الدفع.
7. vw\_ltv — قيمة عمرية تراكمية (نفقات – عوائد) مع تقسيم قنوات.

### **D) معايير وحسابات أساسية**

* **Booked ROAS** = الإيراد المنسوب ÷ الإنفاق المحوّل.
* **Realized ROAS (COD)** = (إيراد COD × **COD Success%**) + إيراد Prepaid ÷ الإنفاق.
* **Zero-Results Rate** = عدد search بنتيجة 0 ÷ إجمالي search.
* **On-time %** = Delivered ضمن promised\_sla\_days.
* **Size OOS impact** = فرق CR عند نفاد مقاس مقابل توافره لنفس SKU/القناة.

### **E) الأداء والتكلفة (Cost Control)**

* **Partition** كل الجداول على event\_date أو order\_date.
* **Cluster** على event\_name, platform, source, user\_pseudo\_id.
* استخدم **Materialized Views** و**Tables مجدولة** للأسئلة الثقيلة المكررة.
* حدّد **Budgets/Quotas** وراقب الاستهلاك؛ احذف/أرشِف البيانات الوسيطة القديمة.

### **F) الأمان والامتثال**

* فصل Dataset **مقيّد** لأي بيانات تشغيلية حسّاسة (إن وُجدت خارج GA4).
* **أقل امتياز** (IAM): Viewer/Job User/Editor بحسب الدور.
* لا تخزّن PII من التطبيق في جداول GA4 المشتقّة.
* **Data Catalog**: أوصاف للأعمدة والمؤشرات وصيغها.

### **G) نماذج تنبؤية خفيفة (اختياري لاحقًا)**

* **RTO Propensity**: مدخلات (مدينة، Partner، COD تاريخي، Attempts، قيمة السلة، فئة).
* **Churn Propensity**: مدخلات (قناة اكتساب، تكرار الزيارة/الشراء، فئة مفضّلة، إشعارات).
* المخرجات تُستخدم كـ **Scoring** داخل vw\_audiences\_ml لتفعيل عروض/سياسات.

### **H) ضمان الجودة (Data Quality)**

* اختبارات: **Uniqueness** (transaction\_id)، **Not Null** (price/currency)، **Referential Integrity** (وجود order\_id في جداول الحالة).
* **مصالحة يومية**: GA4 مقابل النظام الخلفي (إيراد/عدد الطلبات ±5–10%).
* تنبيهات عند فجوات/قفزات غير طبيعية أو فشل Job مجدول.

### **I) معايير قبول لهذه النقطة**

* Dataset مُنشأ وبهيكل **Facts/Dims/Views** أعلاه.
* لوحتان تعملان على Views (vw\_channel\_perf, vw\_cod\_ops) بأداء جيد.
* فحوص جودة تلقائية ناجحة + تقرير مصالحة يومي.
* كتيّب مختصر يشرح الحقول الرئيسية ومكان استخدامها.

المخرجات والوثائق الأساسية (Deliverables & Docs)

## **#19) المخرجات والوثائق الأساسية (Deliverables & Docs)**

**الاستفادة (Why it matters)**

* تحوِّل المنظومة إلى عمل يومي منضبط: كل فريق يعرف *ماذا يفعل* و*كيف يقيس* و*متى يتصرف*.
* تقلّل الأخطاء وتسرّع القرار لأن كل التعريفات والقواعد **موثّقة ومشتركة**.

**الهدف (What we want to achieve)**

* حزمة وثائق موحَّدة، بأسماء واضحة، مالك لكل وثيقة، ومكان حفظ واحد.
* قوالب جاهزة (Templates) لتسليم سريع للمبرمجين، التسويق، والعمليات.

### **A) قائمة المخرجات (ماذا نُنتج بالضبط)**

1. **Measurement Plan**جدول الأحداث/المعلمات/الخصائص/التحويلات + أمثلة Payload + شروط الـQA لكل حدث.
2. **Data Dictionary**تعريف كل مقياس وبُعد وصيغ الحساب (CR, AOV, ROAS, COD Success, RTO…)، ونوافذ الإسناد.
3. **GA4 Property & Streams Config**المنطقة الزمنية/العملة/Signals/Retention/Conversions/Custom Definitions/Filters/Links.
4. **UTM Governance Guide**القيم المسموحة (source/medium/campaign/…)، أسلوب التسمية، ومُولّد روابط بسيط.
5. **Server Events (Measurement Protocol) Spec**مخطط أحداث ما بعد الشراء، قواعد idempotency، الطوابع الزمنية، والخصوصية.
6. **Audiences Catalog**تعريف كل جمهور (شرط/مدة/استثناء/قناة استهلاك: Ads/Remote Config/CRM).
7. **Alerts & Runbooks**كتالوج التنبيهات (الشرط/العتبة/المالك/قناة الإرسال) + خطوات “خلال 60 دقيقة”.
8. **Dashboards Catalog**ما اللوحات المتاحة؟ لمن؟ المؤشرات، مصادر البيانات، وتواتر المراجعة.
9. **BigQuery Modeling Spec**Star schema (Facts/Dimensions)، Partition/Cluster، Views مادية، جداول FX/Cost.
10. **Cost Ingestion SOP**مصادر التكلفة، مخطط الملفات، جداول Mapping، تحويل العملة، فحوص الجودة.
11. **Experimentation Playbook**فرضية/جمهور/مقاييس/Guardrails/حجم عيّنة/SRM/خطوات الإطلاق.
12. **QA Checklist (Pre/ Post Release)**سيناريوهات الاختبار، أجهزة الاختبار، فلاتر الداخلية، مطابقة الإيراد.
13. **Privacy & Compliance Addendum**Consent Mode/ATT، سياسة الاحتفاظ، عدم إرسال PII، حق الحذف.
14. **Change Log & Versioning**سجل تغييرات المخطط/الأحداث والتحويلات مع تواريخ واعتمادات.
15. **Owner Matrix (RACI)**من يملك كل قطعة (Marketing/Analytics/Backend/App/OPS).
16. **Training One-Pagers**صفحات تعريف سريعة لقراءة كل لوحة رئيسية.
17. **Roadmap & Backlog**تحسينات قادمة (Incrementality, ML scores, Streaming…).

### **B) هيكل مجلدات مقترَح (مكان واحد لكل شيء)**

/Analytics

/01\_Strategy

- Analytics\_Framework.pdf

- Roadmap\_Backlog.md

/02\_GA4\_Config

- GA4\_Property\_Streams\_Config.md

- UTM\_Guide.xlsx

/03\_Measurement

- Measurement\_Plan.xlsx

- Data\_Dictionary.xlsx

- Change\_Log.md

/04\_Server\_Events

- MP\_Server\_Spec.md

- Idempotency\_Guide.md

/05\_BigQuery

- Modeling\_Spec.md

- Views\_Definitions.sql

- Cost\_SOP.md

/06\_Audiences\_Alerts

- Audiences\_Catalog.xlsx

- Alerts\_Catalog\_Runbooks.md

/07\_Dashboards

- Dashboards\_Catalog.md

- Training\_OnePagers/

/08\_QA\_Privacy

- QA\_Checklist.md

- Privacy\_Compliance\_Addendum.md

/09\_Experiments

- Experimentation\_Playbook.md

- Templates/

### **C) تسميات الملفات والمعايير**

* لغة الأسماء: إنجليزية بحروف صغيرة وsnake\_case، مع إصدار: v1, v2.
* تواريخ ISO: 2025-08-18.
* كل ملف يبدأ بملخّص 5 أسطر: الغرض/المالك/آخر تحديث/النطاق/الاعتمادات.

### **D) معايير قبول لهذه النقطة**

* كل عنصر في (A) موجود بحده الأدنى ومؤشّر **مالك** و**تاريخ آخر تحديث**.
* الوصول مُنظّم (Viewer/Editor) ومفعّل للجميع.
* لا توجد تعارضات في التعريفات بين الوثائق (CR/AOV/ROAS موحّدة).
* روابط متبادلة بين الوثائق (مثلاً Dashboard يشير لتعريف المؤشر في Data Dictionary).

### **E) قوالب سريعة (جاهزة للنسخ)**

**1) Template — Measurement Plan (صف واحد مثال):**

event\_name: add\_to\_cart

trigger: tap on "Add to Cart" on PDP

required\_params: currency, items[0].{item\_id,item\_name,price,quantity,size,color,stock\_status}

optional\_params: value, coupon, discount

user\_properties\_used: preferred\_payment, favorite\_size

qa\_steps: DebugView → verify params + value = price\*qty; appears once per tap; currency present

owner: Mobile Team / Reviewer: Analytics

**2) Template — UTM Row:**

source=meta | medium=cpc | campaign=2025q3\_dp\_feed\_ar | adset=lookalike\_2p | ad=carousel\_v2

naming\_rules: lower\_snake\_case, no spaces, no Arabic keys

**3) Template — Alert + Runbook:**

**alert: CR\_drop\_20pct\_2h**

**condition: conversion\_rate ↓ >20% vs last 7d, window=2h**

**channel: #product-ops (Slack) + email ops@...**

**owner: Product Analyst**

**runbook (<=60m):**

**1) Check GA4 Realtime for purchase & add\_to\_cart presence**

**2) Compare by platform (iOS/Android) and payment\_type**

**3) Check release notes / feature flags**

**4) If SDK issue: rollback Remote Config / notify Eng**

**5) Open incident ticket, post status in #exec**

**4) Template — Experiment Brief:**

**name: EXP\_checkout\_payment\_priority\_v1**

**hypothesis: prioritizing COD for low-RTO cities increases CR by +10%**

**primary\_metric: purchase\_CVR**

**guardrails: crash\_rate<1.5%, app\_start<2.5s**

**sample\_plan: 1%→10%→50% ramp; min 1 week; SRM check**

**decision\_rule: launch if 95% CI lower bound > 0 and guardrails pass**

**alert: CR\_drop\_20pct\_2h**

**condition: conversion\_rate ↓ >20% vs last 7d, window=2h**

**channel: #product-ops (Slack) + email ops@...**

**owner: Product Analyst**

**runbook (<=60m):**

**1) Check GA4 Realtime for purchase & add\_to\_cart presence**

**2) Compare by platform (iOS/Android) and payment\_type**

**3) Check release notes / feature flags**

**4) If SDK issue: rollback Remote Config / notify Eng**

**5) Open incident ticket, post status in #exec**