data

## **1. Customer Master Data**

* **Identifiers**: customer\_id (internal), phone (E.164), email, social\_id, device\_id
* **Profile**: name, gender, birth year/age, city, country, preferred language & currency
* **Acquisition**: First Source / Last Source (UTM: source, medium, campaign, term, content)
* **Behavior**: registration date, last activity, sessions, devices used
* **Segments**: VIP, Active, Inactive, Cart Abandoner, COD High-Risk, New
* **Properties**: interests, AOV (average order value), LTV, risk score
* **Privacy**: communication consents (SMS, Email, Push, WhatsApp)

## **2. Product & Catalog Data**

* **Product Level**: product\_id, name, description, brand, category/subcategory
* **Variant Level**: variant\_id, SKU, color, size, material, images, videos
* **Pricing**: base price, discounted price, currency, validity dates
* **Inventory**: current stock, minimum stock, stock\_on\_hand, available\_to\_promise (from ledger)
* **Attributes**: SEO keywords, badges (new, best-seller), seasonality tags
* **Analytics**: product views, add-to-cart, conversion rate

## **3. Orders**

* **Header**: order\_id, customer\_id, created\_at, status (Created → Delivered/Returned/Cancelled)
* **Items**: order\_item\_id, variant\_id, price, qty, discount, tax
* **Payment**: method (COD, Online, Wallet), status (Authorized, Captured, Refunded)
* **Shipping**: carrier, tracking number, fee, SLA, status (Shipped, Out for Delivery, Delivered, RTO)
* **Marketing**: UTM source/medium/campaign at order level
* **Geo**: shipping address, city, region, zip code
* **Timestamps**: created, paid, shipped, delivered, cancelled

## **4. Payments**

* **Core**: payment\_id, order\_id, method, amount, currency
* **Status**: Authorized, Captured, Settled, Refunded, Failed
* **Attempts**: log of retries & errors
* **COD Settlement**: collected amount, courier, settlement date, bank transfer status
* **Risk Profile**: allowed\_state (Allowed/Temporarily Blocked), failed\_cod\_attempts, risk\_score

## **5. Shipments**

* **Identifiers**: shipment\_id, order\_id
* Carrier, service type (Standard, Express, Same-Day)
* Tracking number (AWB)
* Actual weight, chargeable weight
* Status: Created, Picked Up, In Transit, Out for Delivery, Delivered, RTO, Failed
* Delivery attempts (with reason: wrong address, refused, not available)
* Timestamps: created, picked, first attempt, delivered, RTO

## **6. Returns / RMA**

* **Core**: rma\_id, order\_id, order\_item\_id, customer\_id
* Type: Return, Replace, Refund
* Reason Codes: defective, wrong size, not satisfied…
* Resolution: Approved/Rejected/Pending
* Refund method: cash, wallet, voucher
* Timestamps: opened, received, inspected, closed

## **7. Wallet & Promotions**

* **Wallet**: wallet\_id, customer\_id, balance, credits, debits, holds
* **Sources**: RMA refunds, cashback, gift cards, promo codes
* **Coupons**: code, type (percentage/fixed), min order value, validity dates, usage count
* **Loyalty Program**: points balance, tier (silver/gold/vip)

## **8. Marketing & Analytics**

* **GA4/Firebase Events**: app\_open, view\_item, add\_to\_cart, begin\_checkout, purchase
* **Attribution**: first\_source, last\_source, medium, campaign, term, content
* **Cohorts**: weekly/monthly (first registration or first purchase)
* **Retention**: W1/W4, M1/M3 (activity & purchase retention)
* **Funnel**: Visit → Product View → Add to Cart → Checkout → Purchase
* **Cart Abandonment**: abandonment rate, recovery rate
* **Reactivation**: inactive customers reactivated after campaigns

## **9. Operations KPIs**

* **Inventory**: turnover ratio, days of supply, out-of-stock %
* **Shipping**: SLA adherence %, success rate, RTO %, avg delivery time
* **Payments**: COD fail rate, online payment success %, avg refund processing time
* **Customers**: CAC, LTV, repeat rate, reactivation rate
* **Marketing**: ROAS, CPA, first vs last touch impact

## **10. Advanced Data**

* **Logs**: app usage, errors, crashes
* **Monitoring**: latency, API calls, DB queries
* **Security**: audit logs (who changed what, when)
* **Governance**: data retention policies (how long to keep, archive, delete)

# **Summary**

A strong company must capture **full-spectrum data**:

1. **Customer Master Data** (identity, acquisition, segments).
2. **Product & Inventory** (catalog + ledger).
3. **Orders & Payments** (transactions + risk).
4. **Shipments & Returns** (logistics).
5. **Wallet & Promotions** (loyalty).
6. **Marketing & Analytics** (attribution, GA4, cohorts, retention).
7. **Operational KPIs** (CAC, LTV, ROAS, RTO, SLA).
8. **Advanced/Compliance** (logs, security, retention).

بنية قاعدة البيانات (MySQL 8 + phpMyAdmin)

# **بنية قاعدة البيانات (MySQL 8 + phpMyAdmin)**

**نطاق العمل:** العملاء، الكتالوج، المنتجات/النسخ، المخزون (Ledger)، السلال، الطلبات/المدفوعات/الشحن/المرتجعات، المحفظة والعروض، التحليلات، الحوكمة والأمان.

## **1) معمارية الطبقات (ERD عالي المستوى)**

* **هوية ومرجعية:** customers, addresses, devices, consents, segments, brands, categories (هرمية), attributes/media.
* **معاملات:** carts/cart\_items, orders/order\_items, payments/payment\_attempts/refunds, shipments/tracking, rmas, promotions/discounts, wallet.
* **المخزون:** inventory\_ledger (المرجع الوحيد للحركة) + stock\_snapshot اختياري للتسريع.
* **تحليلات:** events\_raw (اختياري), facts/dims (orders, order\_items, rma, inventory\_movements).

**مهم:** تفعيل Ledger للمخزون (لا تعديل مباشر للكميات) وربط كل تغيير بسبب واضح (purchase\_receipt, reservation, shipment\_captured, rto\_received, …).

## **2) قواعد تصميم إلزامية**

* **معرّفات لا تُخمن:** UUID (CHAR(36)) أو BINARY(16)؛ مفاتيح بديلة للأرقام المتسلسلة إن لزم.
* **Soft delete:** عمود archived\_at بدل الحذف الفعلي + فهارس تراعيه.
* **أعمدة تدقيق موحّدة:** created\_at, updated\_at, created\_by, updated\_by.
* **UTC دائماً** في الحفظ؛ التحويل للمنطقة الزمنية يتم بالتطبيق/التقارير.
* **عدم تكرار الحقيقة:** لقطات تُؤخذ وقت الحدث (مثلاً سعر الشراء داخل order\_items).

## **3) التسمية والأنواع**

* **أسلوب التسمية:** snake\_case, أسماء واضحة غير عامة.
* **النصوص:** VARCHAR بـ utf8mb4\_unicode\_ci.
* **الأموال:** DECIMAL(12,2) (عملة تُخزَّن في عمود مستقل).
* **التواريخ:** DATETIME(3) مع دقة ملي ثانية إن احتجت تتبّعاً أدق.
* **الحالات/الأكواد:** ENUM أو جداول مرجعية Domain (مثال: order\_status, payment\_state).

## **4) القيود والعلاقات (سلامة البيانات)**

* **FK إلزامية** لمنع السجلات اليتيمة (سياسات واضحة ON DELETE/UPDATE).
* **UNIQUE:** مثل sku على مستوى النسخة، وphone\_e164 على العملاء.
* **CHECK:** السعر ≥ 0، الكمية ≥ 0، صيغة الهاتف E.164، مجموعة عملات مسموحة.
* **Idempotency keys** لعمليات الدفع/التكرار.

## **5) الفهارس والأداء (ميزانيات زمنية)**

* **قوائم المنتجات:** فهرس مركّب مثل (category\_id, publish\_status, is\_visible, created\_at) + فهرس على priceللفرز.
* **البحث/الفلاتر:** brand\_id, (color, size), is\_on\_sale, is\_best\_seller.
* **السلال/الطلبات:** (customer\_id, last\_activity\_at) للسلال؛ (status, created\_at) و(customer\_id, created\_at) للطلبات؛ order\_no.
* **الشحن/المدفوعات/RMA:** tracking\_no، (order\_id)، حالات/مزود.
* **Partial/Covering indexes** للاستعلامات الثقيلة عند الحاجة.
* **أهداف أداء مبدئية:**
  + قائمة منتجات مفلترة ≤ **300ms** على **50K** variants
  + إنشاء طلب (confirm→packed) ≤ **2s**
  + حدث تتبّع شحنة جديد يظهر ≤ **1 دقيقة** من الـ webhook

## **6) الترحيلات (Migrations) والبيئات**

* **كل تغيير عبر Migration قابل للتراجع** (لا تعديل يدوي على الإنتاج).
* **مصادقة مراجعة (PR) + تفقد خطة التنفيذ (EXPLAIN)** قبل الدمج.
* **بيئات منفصلة:** dev/stage/prod وقواعد بيانات منفصلة + بيانات Seed للاختبار.
* **توسيم المخطط:** ملف /docs/db.md يشرح التطوير/الاختبار/الاستعادة.

## **7) الأمان والخصوصية وRBAC**

* **Least Privilege:** أدوار (Admin, Ops, CS, Finance, Marketing, DevReadOnly) مع Grants دقيقة.
* **PII Masking:** هواتف/عناوين تُعرض مقنّعة للأدوار غير المصرّح لها (عبر Views).
* **تشفير:** At-rest + In-transit (TLS)، والحساس جداً يمكن تشفيره عمودياً (AES) إن لزم.
* **Audit Logs:** عمليات حساسة (مدفوعات، ردّيات، تغيير حالات) سجلّ إضافي **append-only**.
* **إدارة الأسرار:** خارج الكود (بيئة/Secret Manager).

## **8) النسخ الاحتياطي والتعافي**

* **RPO ≤ 15 دقيقة** (WAL/Replication)، **RTO ≤ 60 دقيقة** (بيئة DR دافئة/باردة).
* **اختبار استعادة ربع سنوي** مع تقرير نجاح (checksums, counts) وتوثيق كامل.

## **9) المراقبة والتشغيل**

* **لوحات تشغيل:** زمن الاستعلامات، أخطاء القيود، أحجام الجداول، بطء/Locks.
* **تنبيهات:** ارتفاع 5xx/4xx، بطء ملحوظ، فشل نسخ احتياطي.
* **Runbooks:** تباطؤ، قفل جداول، تعارضات، فشل مزوّد دفع/شحن.

## **10) سياسة استخدام phpMyAdmin**

* **إنتاج:** قراءة فقط للأدوار غير الإدارية؛ **ممنوع** أي DDL/DML يدوي—الترحيلات فقط.
* **Stage/Dev:** مسموح للمهندسين مع تسجيل Audit للحركات.
* **حماية:** وصول خلف VPN/IP allowlist + 2FA لحسابات الإدارة.
* **نظافة البيانات:** العمليات الدُفعية/التنظيف تُنفّذ عبر Scripts مُراجعة وليس استعلامات يدوية.
* **الاستعلامات الثقيلة:** تُستقصى عبر EXPLAIN وخارج ساعات الذروة.

## **11) متطلبات وحدات أساسية (اختصار تنفيذي)**

* **Customers/Devices/Consents/Segments:** هاتف فريد E.164 + OTP، ربط الأجهزة وتوكن الإشعارات، موافقات قنوات دقيقة، عضوية شرائح قابلة للتدقيق.
* **Catalog/Variants/Media/Attributes:** قواميس موحّدة للألوان/المقاسات/الخامات قبل أي استيراد كبير.
* **Inventory (Ledger):** كل حركة بسبب/مرجع، Job لحساب stock\_on\_hand وavailable\_to\_promise لكل نسخة/مخزن.
* **Orders/Payments/Shipments/RMA/Wallet/Promotions:** لقطات القيم وقت الحدث، ربط ردّيات الدفع، سياسات RMA واضحة، خصومات مستوى البند/الطلب.
* **Attribution (First/Last):** حفظ أول وآخر مصدر على المستخدم والطلب لدقة التقارير خارج GA4.

## **12) قائمة قبول (Definition of Done)**

1. **سلامة المخطط:** لا سجلات يتيمة؛ FK/UNIQUE/CHECK مُفعّلة؛ بيانات Seed تمرّ.
2. **3NF كحد أدنى**؛ أي إلغاء تطبيع مبرّر ومُوثّق.
3. **فهارس أساسية** موجودة وخطط تنفيذ مقبولة للاستعلامات الحرجة.
4. **أداء** يحقق الميزانيات المذكورة (قائمة/إنشاء طلب/تتبّع).
5. **أمان/خصوصية:** RBAC + Masking + Audit تعمل كما هو متوقّع.
6. **النسخ/الاستعادة:** تجربة ناجحة موثّقة مع RPO/RTO.
7. **الترحيلات:** كل تغيير عبر Migration مع Rollback مجرَّب.
8. **التوثيق:** /docs/db.md محدث ويُمكّن أي مهندس من التشغيل بلا أسئلة.

## **13) التسليمات المطلوبة**

* **ERD مفصّل** (صورة + ملف قابل للتعديل) على الطبقات أعلاه.
* **قاموس بيانات + Matrix قيود** لكل جدول/حقل.
* **خطة فهارس** مبنية على الاستعلامات الفعلية + أمثلة EXPLAIN.
* **سياسة النسخ/الاستعادة** بأرقام RPO/RTO وجدولة اختبارات.
* **RBAC/PII/Audit** (مصفوفة أذونات + Views مقنّعة).
* **تصميم Ledger المخزون** (جداول/أسباب/قواعد الحسبة).
* **سيناريوهات RMA مكتملة** وربطها ماليًا ومخزنيًا.
* **/docs/db.md** (تشغيل محلي، ترحيلات، اختبارات، استعادة، أفضل الممارسات).

مراجعات

## **مراجعات**

## **1) الإصدار والإعدادات الأساسية**

* **المحرك:** InnoDB فقط.
* **الإصدار الموصى به:** MySQL 8.x (أو MariaDB ≥ 10.6).
* **SQL Mode:** فعّل STRICT\_TRANS\_TABLES, وامنَع القيم الضمنية غير الصحيحة.
* **المنطقة الزمنية:** اضبط الخادم والتطبيق على UTC وخزّن الوقت بUTC، واعرض للمستخدم +03 عند الحاجة.
* **ملف لكل جدول:** innodb\_file\_per\_table=ON.
* **Binary Log:** فعّله لنسخ احتياطي تفاضلي واسترجاع النقطة الزمنية.

## **2) الترميز والترتيب (Collation)**

* **الترميز:** utf8mb4 للجداول وكافة الأعمدة النصية.
* **الترتيب:** MySQL 8: استخدم utf8mb4\_0900\_ai\_ci. (إن كان الإصدار أقدم، استخدم utf8mb4\_general\_ci).

## **3) أسلوب التسمية والبنية**

* **أسماء الجداول:** مفردة وsnake\_case (مثال: customer, order\_item).
* **عدم استخدام بادئات عشوائية** مثل tbl\_.
* **أعمدة قياسية:** id, created\_at, updated\_at, deleted\_at (للـsoft delete عند الحاجة).

## **4) المفاتيح والمعرّفات**

* **Primary Key:** BIGINT UNSIGNED AUTO\_INCREMENT أو **UUIDv7/ULID** مخزّن في BINARY(16) للأنظمة الموزعة.
* **العلاقات (FK):** استخدم قيودًا صريحة مع قواعد **ON UPDATE/DELETE** واضحة (غالبًا RESTRICT أو CASCADE عند الجداول الرابط).
* **Unique Constraints:** للبريد الإلكتروني، الكوبونات، الأكواد الفريدة.

## **5) الفهارس (Indexes)**

* **قاعدة عامة:** فهرس لكل مفتاح خارجي، ولكل عمود يُستعلم عنه في WHERE/ORDER BY بشكل متكرر.
* **المركّبة:** احرص على ترتيب الأعمدة حسب الانتقائية والاستخدام ((customer\_id, created\_at) مثلًا).
* **لا تفرط بالفهارس**؛ كل فهرس يُكلف كتابة/تحديثات أبطأ.

## **6) أنواع البيانات القياسية**

* **المال:** DECIMAL(12,2)، **لا تستخدم FLOAT**.
* **الحالات:** استخدم TINYINT مع جدول مرجعي أو ENUM محدودة جدًا.
* **JSON:** مسموح لـ meta الصغيرة، لكن لا تجعله يحل محل التصميم العلائقي.

## **7) الأمان والوصول**

* **حسابات منفصلة:** مستخدم للقراءة/الكتابة للتطبيق، وآخر للقراءة فقط للتحليلات—**لا تستخدم root**.
* **أقل صلاحيات ممكنة (Least Privilege)** عبر GRANT محدد على المخطط.
* **TLS** للاتصال بين التطبيق وقاعدة البيانات إن كانت عبر شبكة غير محمية.
* **تقييد IP** والوصول للوحة phpMyAdmin عبر VPN أو allowlist.
* **تدقيق**: فعّل سجل الاستعلامات البطيئة وراقب الأنماط.

## **8) الأداء والمراقبة**

* **Buffer Pool:** اضبط innodb\_buffer\_pool\_size (حتى 60–70% من RAM على خادم مخصص).
* **Slow Query Log:** فعّله وحسّن كل استعلام يتجاوز 200–500ms.
* **EXPLAIN/ANALYZE:** استخدمها قبل اعتماد أي استعلام ثقيل.
* **التقسيم (Partitioning):** فقط للجداول الضخمة (مثل أحداث التحليلات شهريًا).
* **الترقيم:** فضّل **Keyset Pagination** على OFFSET الكبير.

## **9) النسخ الاحتياطي والاسترجاع**

* **خطة عملية:**
  + نسخ **كامل يومي** + **تفاضلي عبر binlog كل 15–30 دقيقة**.
  + اختبارات استرجاع **شهرية** (لا تعتبر النسخ صحيحة حتى تُختبَر).
* **أدوات:** للمشاريع المتوسطة استخدم mysqldump (مع إغلاق كتابة أو قفل منطقي)، وللضخمة **Percona XtraBackup**.
* **تخزين خارجي** مُشفّر + احتفاظ بإصدارات 7/14/30 يومًا.

## **10) سياسة التغييرات (Migrations)**

* **ممنوع** تعديل المخطط يدويًا من phpMyAdmin في بيئة الإنتاج.
* **اعتمد أدوات ترحيل Schema** (Flyway/Liquibase أو سكربتات SQL مُرقّمة في مستودع الـAPI) مع مراجعة كود ونسخة احتياطية قبل التنفيذ.
* **بيئات منفصلة:** dev / staging / prod مع بيانات اختبار (لا بيانات عملاء حقيقية).

## **تشغيل يومي عبر phpMyAdmin (بدون مخاطرة)**

* **صلاحيات المستخدمين:** تأكد أن حساب phpMyAdmin اليومي **قراءة فقط** في الإنتاج، وأن التغييرات الهيكلية تتم عبر الـmigrations.
* **قوالب استعلامات آمنة:** أنشئ مجلد “Saved Queries” للاستعلامات التقارير الشائعة (قراءة فقط).
* **مراجعة الاستعلامات البطيئة أسبوعيًا** وتحويل التحسينات إلى تذاكر تقنية.

# **التحسين السريع**

**١) إصدار وإعدادات الخادم**

* استخدموا MySQL 8.x (InnoDB فقط) + utf8mb4 افتراضيًا.
* فعّلوا sql\_mode: STRICT\_TRANS\_TABLES, NO\_ZERO\_DATE, NO\_ZERO\_IN\_DATE, ERROR\_FOR\_DIVISION\_BY\_ZERO.
* Binlog: ROW + sync\_binlog=1 + innodb\_flush\_log\_at\_trx\_commit=1 لسلامة البيانات.
* سجّلوا الاستعلامات البطيئة slow\_query\_log=ON, long\_query\_time=0.2 لمراقبة الأداء.

**٢) الأمن والحوكمة**

* **phpMyAdmin خلف VPN/IP allowlist**، إيقاف الوصول العام، وتعطيل الدخول بـ root عن بعد.
* RBAC: حسابات **read-only** للإنتاج، وتنفيذ DDL/DML **فقط عبر migrations**.
* إجبار الاتصال بتشفير TLS، وتدوير كلمات المرور، وتفعيل تدقيق النشاط الحساس.

**٣) الترحيلات والعمليات**

* تبنّوا أداة ترحيلات (Flyway أو Liquibase أو ORM migrations) مع **Rollback** مجرّب.
* للتغييرات الكبيرة على الجداول استخدموا **gh-ost** أو **pt-online-schema-change** لتجنب التوقف.
* بيئات منفصلة (dev/stage/prod) + بيانات seed + تشغيل فحوصات ما بعد الترحيل.

**٤) التوافر والنسخ الاحتياطي**

* نسخ احتياطي يومي + **Point-in-Time Recovery** (binlog) مع **اختبار استعادة ربع سنوي**.
* نسخة قراءة (read-replica) للتقارير، أو خدمة مُدارة (AWS RDS/Aurora) إن أمكن.

**٥) الفهارس والأداء (مباشرة لسيناريوكم)**

* الكتالوج/القوائم: فهرس مركّب على (category\_id, publish\_status, is\_visible, created\_at) + فهرس للفرز بالسعر.

السلال/الطلبات:

* Carts: (customer\_id, last\_activity\_at)
* Orders: (status, created\_at) و (customer\_id, created\_at)، وفهرس فريد لـ order\_no.
* غطّوا الأعمدة المُستخدمة في الفلاتر الشائعة (brand/color/size/is\_on\_sale).
* راجعوا **EXPLAIN** لأثقل 10 استعلامات واعمَلوا Covering Indexes عند الحاجة.

**٦) التكامل التحليلي**

* أخرجوا التحليلات الثقيلة من الإنتاج عبر **CDC** (Debezium/Airbyte) → BigQuery.
* احفظوا **First/Last touch** داخل قاعدة البيانات (على مستوى العميل والطلب) لتقارير التسويق.

**٧) استخدام phpMyAdmin عمليًا**

* Prod: قراءة فقط (استعلامات تشخيصية خفيفة).
* Stage/Dev: مسموح مع سجل تدقيق.
* عمليات تنظيف/تعديل جماعي تُنفَّذ كسكربتات مُراجَعة، **ليس يدويًا** من الواجهة.

**٨) طبقات مساندة**

* **Redis** للجلسات والكاش والمعدلات (rate limits).
* طبقة Proxy (مثل **ProxySQL/HAProxy**) لإدارة الاتصالات وتوزيعها على الـ replicas.

بناء قاعدة البيانات MySQL

# **بناء قاعدة البيانات MySQL**

**الغرض**: مرجع تنفيذي موحّد للفريق التقني لضمان أن بنية قاعدة البيانات محكمة، آمنة، قابلة للتوسع، وسهلة التشغيل.

**النطاق**: MySQL 8.x + InnoDB + phpMyAdmin (كأداة وصول مُقيّدة)، مع تكاملات اختيارية للتحليلات (CDC → BigQuery).

**الجمهور**: مهندسو البنية الخلفية/البيانات، مسؤول قواعد البيانات، قيادة التقنية.

## **0) TL;DR**

* استخدموا **MySQL 8.x / InnoDB**، ترميز **utf8mb4**، توقيت **UTC**، و**migrations** لكل تغيير.
* مخزون بنمط **Inventory Ledger** (لا تعديل مباشر للكميات).
* **FK/UNIQUE/CHECK** إلزامية مع فهارس مدروسة.
* **RBAC** صارم + **PII Masking** عبر Views + تدقيق Audit.
* **PITR** عبر binlog + **اختبار استعادة** ربع سنوي.
* **phpMyAdmin** في الإنتاج قراءة فقط وخلف VPN.

## **1) الأهداف والنطاق**

* ضمان **سلامة البيانات** و**منع السجلات اليتيمة**.
* دعم **الاستعلامات الحرجة** (قوائم المنتجات، إنشاء طلب، تتبع الشحن) بزمن مقبول.
* تمكين **تحليلات دقيقة** (First/Last-touch، تقارير المبيعات، المرتجعات، المخزون).
* سهولة **التشغيل والتوسّع** (replicas، partitioning عند الحاجة).

**النطاق الوظيفي:** هوية العملاء، الكتالوج، المتغيرات (SKU)، المخزون، السلال، الطلبات، المدفوعات/الرديات، الشحن/التتبّع، المرتجعات (RMA)، المحفظة/العروض/الكوبونات، التحليلات.

## **2) مواصفات الخادم والإعدادات**

* **الإصدار/المحرّك**: MySQL 8.x, InnoDB فقط.
* **الترميز والترتيب**: utf8mb4 وutf8mb4\_unicode\_ci افتراضياً.
* **sql\_mode**: STRICT\_TRANS\_TABLES,NO\_ZERO\_DATE,NO\_ZERO\_IN\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO.
* **binlog**: صيغة ROW + sync\_binlog=1 + innodb\_flush\_log\_at\_trx\_commit=1 لسلامة المعاملات.
* **الذاكرة**: innodb\_buffer\_pool\_size ≈ 70–80% من ذاكرة الخادم.
* **سجل البطيء**: slow\_query\_log=ON, long\_query\_time=0.2 لمراقبة الاستعلامات.

ملاحظة: استخدموا خدمة مُدارة (مثل RDS) إن أمكن لتبسيط التوفّر والنسخ.

## **3) معمارية وطبقات البيانات (ERD عالي المستوى)**

**المرجع/الهوية**

* customers, addresses, devices, consents, segments
* brands, categories (هرمية), attributes, media\_assets

**الكتالوج**

* products (SPU)، product\_variants (SKU)
* روابط السمات/الصور: variant\_attributes, product\_media

**المخزون**

* inventory\_ledger (المرجع الوحيد للحركة)
* stock\_snapshot (اختياري للتسريع)

**التجارة**

* carts, cart\_items
* orders, order\_items
* المدفوعات: payments, payment\_attempts, refunds
* الشحن: shipments, shipment\_events, carriers
* المرتجعات: rmas, rma\_items
* المحفظة/العروض: wallets, wallet\_tx, promotions, coupons, coupon\_redemptions

**التحليلات (اختياري)**

* events\_raw (تجميعي)، جداول حقائق/أبعاد: fct\_orders, fct\_order\_items, dim\_date, fct\_inventory\_moves

**قاعدة ذهبية**: قيّم في order\_items وغيره تُخزن كسَـنَـپشوت وقت الحدث (السعر، العملة، الخصم…)

## **4) معايير التصميم والأنواع**

* **التسمية**: snake\_case، أسماء واضحة غير عامة.
* **المعرّفات**: UUID (CHAR(36)) أو BINARY(16) حسب التفضيل؛ أرقام تلقائية داخلية مسموحة كمفاتيح بديلة.
* **التواريخ**: DATETIME(3) بدقة ملي ثانية؛ **UTC** دائماً.
* **الأموال**: DECIMAL(12,2) + عمود currency.
* **Soft delete**: عمود archived\_at مع فهارس تراعيه.
* **أعمدة تدقيق**: created\_at, updated\_at, created\_by, updated\_by.

## **5) القيود والعلاقات (سلامة البيانات)**

* **Foreign Keys** إلزامية مع سياسات ON DELETE/UPDATE واضحة.
* **UNIQUE**: مثل sku، وphone\_e164 للعملاء.
* **CHECK**: الأسعار ≥ 0، الكميات ≥ 0، مجموعة عملات مقبولة.
* **Idempotency keys**: لمحاولات الدفع أو العمليات الحساسة.

## **6) الفهارس والأداء**

**أهداف زمنية**

* قوائم/بحث المنتج ≤ **300ms** على ~50K متغير.
* تأكيد طلب → إنشاء السجلات الأساسية ≤ **2s**.
* وصول حدث تتبّع شحنة → انعكاسه ≤ **1m**.

**إرشادات عامة**

* فهارس مركّبة تُطابق شروط WHERE والفرز ORDER BY.
* **Covering Indexes** للاستعلامات الثقيلة المتكررة.
* مراجعة **EXPLAIN** لأثقل 10 استعلامات شهرياً.

**أمثلة عملية**

* القوائم: (category\_id, publish\_status, is\_visible, created\_at) + فهرس على price.
* البحث بالخصائص: brand\_id, (color, size), is\_on\_sale, is\_best\_seller.
* السلال: (customer\_id, last\_activity\_at).
* الطلبات: (status, created\_at) و(customer\_id, created\_at) + فريد order\_no.
* الشحن/المدفوعات: tracking\_no, (order\_id)، حالات المزود.

**Partitioning (عند الحاجة)**

* جداول ضخمة زمنية مثل events\_raw, inventory\_ledger عبر RANGE على created\_at.

## **7) نموذج المخزون بنمط Ledger**

* كل حركة في inventory\_ledger تحمل **سببًا** (reason) و**مرجعًا** (reference\_id/type):
  + purchase\_receipt, adjustment, reservation, shipment\_captured, rto\_received, rma\_returned …
* حساب stock\_on\_hand وavailable\_to\_promise يتم دوريًا (Job خلفي) أو View مادّي.
* يمنع التعديل المباشر على الرصيد؛ التغيير دائمًا عبر **إضافة حركة**.

## **8) الأمان والخصوصية وRBAC**

**تصنيف البيانات**

* **PII**: الهاتف، البريد، العنوان؛ **حسّاسة**: بيانات الدفع (Tokenized فقط)؛ **غير حساسة**: بيانات الكتالوج.

**أذونات الأدوار (مثال)**

* Admin/DBA (كامل)، Ops/Finance (قراءة + إجراءات محددة)، CustomerService (قراءة مقنّعة)، Marketing/Analyst (Views تحليلية)، DevReadOnly (قراءة فقط).

**سياسات**

* **Least Privilege** + مراجعة دورية للصلاحيات.
* **PII Masking** عبر Views (إخفاء جزء من الهاتف/العنوان للأدوار غير المصرّح لها).
* تشفير النقل (TLS) وتخزين الأسرار خارج الكود (Secret Manager).
* **Audit Log** لعمليات حساسة (رديات، تغييرات حالة، تسويات مخزون).

## **9) النسخ الاحتياطي والتعافي من الكوارث (DR)**

* **RPO ≤ 15m** عبر binlog + **RTO ≤ 60m** (بيئة DR دافئة/باردة).
* نسخ **يومي** + احتفاظ بالbinlogs لأسبوعين على الأقل.
* **اختبار استعادة ربع سنوي** مع تحقق checksums وأعداد الصفوف.
* توثيق مسار الاستعادة خطوة بخطوة.

## **10) الترحيلات (Migrations) وCI/CD**

* كل تغيّر Schema عبر **Migration** (Flyway/Liquibase/ORM) مع **Rollback**.
* فحوصات تلقائية: lint للـDDL، مراجعة EXPLAIN للتغييرات المؤثرة.
* تغييرات كبيرة بلا توقف عبر **gh-ost** أو **pt-online-schema-change**.
* باك-أب قبل الترحيل الإنتاجي + خطة رجوع.

## **11) البيئات والبيانات**

* بيئات منفصلة: dev / stage / prod ببيانات مستقلة.
* بيانات **Seed** للاختبار، مع قناع/إخفاء PII عند استخدام عينات من الإنتاج.
* توليد نسخ تحليلات من خلال **CDC** بدل تشغيل استعلامات ثقيلة على الإنتاج.

## **12) المراقبة والتشغيل**

* **لوحات**: زمن الاستعلامات، بطء/Locks، حجم الجداول، أخطاء القيود.
* **تنبيهات**: طفرات في 5xx/4xx، فشل نسخ احتياطي، ازدياد الاستعلامات البطيئة.
* **Runbooks**: تباطؤ مفاجئ، تعارضات قفل، فشل مزود دفع/شحن.

## **13) سياسة استخدام phpMyAdmin**

* **إنتاج**: قراءة فقط للأدوار غير الإدارية؛ لا **DDL/DML** يدوي—المسموح عبر **migrations** فقط.
* **Stage/Dev**: مسموح للمهندسين مع سجل تدقيق.
* وصول خلف **VPN/IP allowlist** + **2FA** لحسابات الإدارة.
* الاستعلامات الثقيلة تُنفذ خارج ساعات الذروة وبعد مراجعة.

## **14) تكامل التحليلات (اختياري موصى به)**

* **CDC** (Debezium/Airbyte) → Warehouse (BigQuery) لتقارير مرنة دون الضغط على الإنتاج.
* حفظ **First-/Last-touch** على مستوى المستخدم والطلب لقياس العائد الإعلاني بدقة.
* نمذجة حقائق/أبعاد: fct\_orders, fct\_order\_items, fct\_inventory\_moves, dim\_date, dim\_customer.

## **15) قوالب جاهزة (انسخ/املأ)**

### **15.1 قالب قاموس بيانات (Data Dictionary)**

* **الجدول**: …
* **الوصف**: …
* **الأعمدة**:
  + column\_name · النوع · السماحية (NULL/NOT NULL) · الافتراضي · القيود (PK/FK/UNIQUE/CHECK) · الوصف · مثال
* **الفهارس**: (الاسم، الأعمدة، الهدف من الاستعلام)
* **ملاحظات الأداء/الأمان**: …

### **15.2 قالب سياسة النسخ والاستعادة**

* **RPO/RTO**: …
* **نوع النسخ**: يومي تزايدي + binlogs
* **التحقق الدوري**: checksums، استعادة اختبارية
* **خطوات الاستعادة**: …

### **15.3 قالب مصفوفة الأذونات (RBAC)**

* **الدور** | **الجداول/Views** | **العمليات (SELECT/INSERT/UPDATE/DELETE)** | **قيود/ملاحظات**

### **15.4 قالب خطة الفهارس**

* **الاستعلام** | **التواتر** | **الجدول** | **اقتراح الفهرس** | **EXPLAIN قبل/بعد** | **التحسن**

### **15.5 قالب فحص جودة البيانات**

* **التحقق** | **القيد/القانون** | **نسبة التغطية** | **استثناءات** | **الإجراء التصحيحي**

## **16) قوائم تحقق (Checklists)**

### **16.1 أثناء التصميم**

### **16.2 قبل الإطلاق/الترحيل الإنتاجي**

### **16.3 التشغيل الدوري**

* **يومي**: مراقبة بطء الاستعلامات/الأخطاء/النسخ.
* **أسبوعي**: مراجعة الفهارس للأثقل استخدامًا.
* **شهري**: اختبار استعادة جزئي + تحسينات أداء.
* **ربع سنوي**: اختبار استعادة كامل + مراجعة RBAC.

## **17) ملحق (قاموس حالات موحّد كمثال)**

* **order\_status**: created → confirmed → packed → shipped → delivered → (rto/lost/cancelled)
* **payment\_state**: pending / authorized / captured / failed / refunded
* **shipment\_event**: picked\_up, in\_transit, out\_for\_delivery, delivered, rto\_initiated, rto\_received
* **rma\_status**: requested, approved, received, refunded/denied

الحماية والأساسيات التشغيلية

# **الحماية والأساسيات التشغيلية**

## **1. الحماية (Security)**

* تطبيق مبدأ **Least Privilege** مع أدوار واضحة (Admin, Ops, CS, Marketing, ReadOnly).
* تفعيل **MFA** لجميع حسابات الإدارة والقاعدة.
* استخدام **Secrets Manager** لتخزين المفاتيح والتوكنات وتدويرها بشكل دوري.
* تشفير البيانات **At-Rest** و **In-Transit (TLS 1.2+)**.
* تطبيق **PII Masking** للهاتف والعنوان في واجهات غير مخوّلة.
* تفعيل **Audit Log** لجميع العمليات الحساسة (تعديل بيانات، تغييرات مالية).
* الحماية من الهجمات: منع **SQL Injection/XSS**، واستخدام ORM آمن.
* تفعيل **Rate Limiting** على عمليات الدخول وOTP والدفع.

## **2. النسخ الاحتياطي والتعافي (Backup & Recovery)**

* تحديد **RPO ≤ 15 دقيقة** و **RTO ≤ 60 دقيقة**.
* نسخ احتياطي يومي/ساعي مشفّر مع مراقبة حالة النسخ.
* اختبار استعادة البيانات بشكل **ربع سنوي** وتوثيق النتائج.
* خطة **Disaster Recovery** ببيئة بديلة جاهزة (Warm/Cold).

## **3. المراقبة والموثوقية (Monitoring & Reliability)**

* مؤشرات أداء مستهدفة (SLOs):
  + استعلام قائمة المنتجات ≤ 300ms (50K variants).
  + إنشاء طلب ≤ 2s.
  + Webhook مزوّد دفع ≤ 1s.
* تفعيل **Logs، Metrics، Traces** بشكل منظم.
* تنبيهات على ارتفاع الأخطاء (5xx/4xx)، محاولات الدخول الفاشلة، وفشل النسخ الاحتياطي.
* اختبارات **Load/Stress** دورية للتأكد من تحمل النظام.

## **4. الخصوصية والامتثال (Privacy & Compliance)**

* تصنيف البيانات (Public / Internal / Confidential / Sensitive).
* تطبيق سياسة **الاحتفاظ بالبيانات (Data Retention)** مع جداول زمنية للحذف أو الأرشفة.
* تقليل تخزين البيانات الحساسة والاكتفاء بالضروري فقط (**Data Minimization**).
* حقوق العملاء: تصحيح/حذف بياناتهم عند الطلب مع وجود إجراءات واضحة.
* اتفاقيات حماية بيانات (DPA) مع المزوّدين الخارجيين (الدفع، الشحن، الرسائل).

## **5. إدارة التغيير والحوادث (Change & Incident Management)**

* سجل تغييرات (Change Log) قبل أي نشر.
* **Runbooks** جاهزة للأحداث الشائعة (فشل الدفع، مشاكل الشحن، توقف خدمة).
* تطبيق **Post-Mortem** إلزامي لأي حادثة حرجة (Sev-1/Sev-2) مع خطة منع التكرار.

NOTE

# **الهدف**

* قاعدة بيانات **سليمة، متّسقة، قابلة للتوسّع**، سهلة التكامل والاختبار، وتحمي الخصوصية.

# **النطاق**

* العملاء، الكتالوج، المنتجات، الطلبات، السلة، المفضلة، المدفوعات، الشحن، المرتجعات والاستبدال، العروض والخصومات، المشتريات والتوريد، التحليلات، المحفظة

# **ما يجب توفره (Outputs)**

1. **ERD نهائي** يوضّح الجداول والعلاقات واتجاهاتها.
2. **معايير تسمية وحوكمة** (أسماء، snake\_case، الحقول القياسية مثل created\_at/updated\_at، سياسة الحذف المنطقي).
3. **قاموس بيانات** يحدد النوع والقيود والسماحية لكل حقل.
4. **مصفوفة قيود**: مفاتيح خارجية/فريدة/تحقق (FK/UNIQUE/CHECK).
5. **خطة فهارس** مبنية على الاستعلامات الأكثر استخدامًا.
6. **خطة ترحيلات (Migrations)** لكل تغيير على المخطط مع إمكانية التراجع.
7. **سياسة خصوصية وأمان**: تصنيف PII، تشفير/إخفاء، أدوار وصلاحيات وصول، تدقيق (Audit).
8. **سياسة نسخ احتياطي واستعادة** مع اختبار استعادة مُوثّق.
9. **خطة اختبار**: سلامة/قواعد/تزامن/أداء + بيانات Seed.
10. **توثيق موجز** في /docs/db.md يشرح كيف نطوّر/نختبر/نستعيد.

# **معايير القبول (يجب تحققها قبل الاعتماد)**

* **سلامة البيانات:** لا سجلات يتيمة؛ جميع العلاقات محمية بـFK؛ قيود CHECK لكل الحدود (سعر ≥ 0، كمية ≥ 0…).
* **اتساق التصميم:** تطبيع لا يقل عن 3NF؛ أي إلغاء تطبيع مبرَّر للأداء.
* **اتساق الأسماء والأنواع:** لا أعمدة عامة/غامضة؛ وصف واضح لكل حقل.
* **الفهارس:** موجودة على أعمدة الفرز/التصفية الشائعة؛ لا فهارس مكررة.
* **الأمن والخصوصية:** أقل صلاحيات للوصول (Roles/Grants)، تشفير/إخفاء للـPII، سجل تدقيق للعمليات الحساسة.
* **الأداء:** ميزانيات زمنية واضحة للاستعلامات الرئيسية (قائمة، تفاصيل، إنشاء طلب) ضمن حدود متفق عليها.
* **الترحيلات:** كل تغيير يمر عبر Migration قابل للتراجع، ولا تعديل يدوي مباشر على الإنتاج.
* **الاختبارات:** تغطية CRUD، حالات سلبية وحدّية، تعارضات التزامن، واختبار أداء بسيط—all ناجحة على بيئة تجريبية.
* **التوثيق:** محدث وكافٍ لتشغيل وتطوير القاعدة بدون أسئلة إضافية.

# **مبادئ تصميم إلزامية**

* **IDs غير قابلة للتخمين** (UUID)، **Soft delete** عبر archived\_at بدل الحذف الفعلي.
* **أعمدة تدقيق موحّدة:** created\_at/updated\_at/by، ومصدر آخر تعديل عند الحاجة.
* **فصل الحسّاس/PII** في جداول أو أعمدة مع تشفير/Masking.
* **Ledger لحركة المخزون** بدل تعديل كميات مباشرة (traceability).
* **عدم تكرار الحقيقة**: كل حقيقة تُخزَّن مرّة واحدة؛ اللقطات تُلتقط عند الحاجة (مثل snapshot لسعر الشراء في order\_items).

# **الاختبار والجودة (بدون أدوات محددة)**

* **سلامة المخطط:** إدخالات غير صالحة تُرفَض بالقيود، والعلاقات تمنع اليُتم.
* **قواعد العمل:** جميع الحالات السلبية تُرجع أخطاء مفهومة (وليس صمتًا).
* **التزامن:** تحديثان متوازيان لنفس السجل لا يسببان فقدان بيانات (تفادي السباق).
* **الأداء:** الاستعلامات الحرجة تحت الميزانيات المحددة وبخطط تنفيذ (Explain) مقبولة.
* **الاسترجاع:** سيناريو استعادة من نسخة احتياطية يثبت إمكانية العودة لوضع سليم.

# **الأمان والخصوصية**

* **Least Privilege:** حسابات التطبيق تمتلك أقل صلاحية لازمة فقط.
* **فصل الأسرار:** لا أسرار داخل الكود؛ إدارة أسرار خارجية.
* **التدقيق والمراقبة:** تسجيل من غيّر ماذا ومتى؛ مؤشرات لأخطاء القيود وزمن الاستعلامات.

# **المراقبة والتشغيل**

* **لوحات مراقبة** لزمن الاستعلامات والأخطاء وأحجام الجداول.
* **تنبيهات** عند تجاوز عتبات الأداء أو ارتفاع أخطاء القيود.
* **Runbook مختصر**: ماذا نفعل عند بطء/قفل/خطأ قيود.

أهم الفجوات

# **أهم الفجوات**

## **1) ERD تفصيلي**

**المطلوب الآن**: إخراج ERD على 3 طبقات: الهوية/المرجعية، المعاملات، التحليلات. (يُسلم بصيغة صورة + mermaid/plantuml اختيارياً) — يشمل:

* **العملاء**: customers, addresses, devices, consents, segments (M2M), cod\_profile, comms\_log.
* **الكتالوج**: brands, products, product\_variants, categories (self-parent), product\_categories (Lk), attributes/attribute\_values/product\_attributes/variant\_attributes, media, collections (+ rules).
* **المخزون**: warehouses, **inventory\_ledger** (event\_time, variant\_id, warehouse\_id, qty\_change, reason, ref\_type/ref\_id), stock\_snapshot (اختياري للتسريع). (المبدأ مذكور ويحتاج تفصيل)
* **الطلبات**: orders, order\_items, order\_status\_history, shipments, shipment\_items, tracking\_events, payments, payment\_attempts, refunds, cod\_remittance, promotions, order\_discounts, address\_snapshot.
* **السلة**: carts, cart\_items, cart\_coupons.
* **المفضلة**: wishlists, wishlist\_items.
* **RMA**: rmas, rma\_items, rma\_logistics, rma\_resolutions.
* **التحليلات**: events\_raw (للاستيعاب إن لزم)، tables للـ RFM/LTV مجمّعة (اختياري لاحقًا).

## **2) قاموس بيانات + مصفوفة قيود**

* **أنواع/قيود إلزامية**:
  + UUID/VARCHAR للمعرّفات غير القابلة للتخمين.
  + ENUM/Domain للجداول المرجعية (order\_status، payment\_state، shipment\_state…).
  + **CHECK**: price ≥ 0، qty ≥ 0، phone بصيغة E.164، currency in {YER,SAR,…}.
  + **UNIQUE**: sku على مستوى variant، (product\_id, color, size)، phone على customers.
  + **FK** مع ON DELETE/UPDATE سياسات مدروسة لمنع اليُتم.
* **Soft-delete**: archived\_at مع فهارس تراعيه.

## **3) فهارس مبنية على الاستعلامات الفعلية**

ضع خطة مبدئية (قابلة للتعديل بعد قياس الإنتاج):

* **قوائم المنتجات**: idx on (category\_id, is\_visible, publish\_status, created\_at), و(price) للفرز.
* **البحث/الفلاتر**: فهارس على (brand\_id), (color, size), (is\_on\_sale), (is\_best\_seller).
* **السلة/الطلب**: (customer\_id, last\_activity\_at) للسلة؛ (order\_no)، (status, created\_at), (customer\_id, created\_at) للطلبات.
* **الشحن**: (tracking\_no), (courier\_id, mapped\_status), (order\_id).
* **المدفوعات**: (order\_id), (state), (provider\_ref).
* **RMA**: (order\_id), (state), (reason\_code).
* **تحسين لاحق**: partial indexes (publish\_status='published'), وcovering indexes للاستعلامات الثقيلة. (الخطة مطلوبة كمتطلب عام)

## **4) النسخ الاحتياطي/الاستعادة + RTO/RPO**

* وثّق: **RPO** ≤ 15 دقيقة (WAL/streaming)، **RTO** ≤ 1 ساعة (بيئة DR باردة/دافئة).
* **اختبار استعادة فصلي** مع تقرير نجاح وإثبات صحة (checksums، counts). (السياسة مطلوبة بالملف وتحتاج أرقامًا وتنفيذ اختبار دوري)

## **5) أمن/خصوصية وRBAC مفصّل**

* **تصنيف PII** (phone, address, comms\_log) + **Masking** لأدوار غير مخوّلة.
* **RBAC**: أدوار (Admin, Ops, CS, Finance, Marketing, DevReadOnly) وGrants لكل جدول/عمود.
* **Audit** غير قابل للتعديل لعمليات الدفع/الاسترداد/تغيير الحالات. (مذكور كمتطلب ويحتاج مصفوفة أذونات مفصلة)

## **6) مشتريات/توريد + تكاليف**

* أضِف جداول: suppliers, purchase\_orders, po\_items, grn (استلام), supplier\_invoices, landed\_cost\_allocations لاحتساب تكلفة القطعة الواحدة بدقّة وتغذية التسعير/الهامش. (النطاق ذكر “المشتريات والتوريد” دون تفصيل)

## **7) Ledger المخزون (تفصيل عملي)**

* **inventory\_ledger** يسجل كل تغيير مع السبب: (purchase\_receipt, reservation, shipment\_captured, rto\_received, rma\_returned, adjustment, stocktake).
* **قواعد**: لا تعديل مباشر للمخزون؛ فقط عبر أحداث ledger + job تَحسِب **stock\_on\_hand** و **available\_to\_promise** لكل variant/warehouse. (المبدأ مذكور ويحتاج تصميم جدول واضح)

## **8) إكمال RMA**

* أضف حالات واضحة: opened → awaiting\_pickup → in\_quality\_check → approved\_refund/approved\_exchange → completed; مسارات جانبية: rejected / cancelled.
* **سياسات مالية**: ربط Refund بالدفع الأصلي (أونلاين) أو **store\_credit** للمحفظة عند COD. (الإطار موجود ويحتاج إكمال حالات/سياسات)

## **9) التحليلات**

* توحيد user\_id مع GA4/Firebase وربط UTM على مستوى الطلب (مذكور).
* طبقة **business facts**: fact\_orders, fact\_order\_items, fact\_rma, fact\_inventory\_movements + dimensions (date, product, variant, category, city, courier, source/medium).
* KPIs المذكورة للمنتج/المحتوى تُحسب تلقائيًا في لوحات.

# **اقتراح هيكل العلاقات (High-level ERD نصّي)**

* **customers(1)** —< **addresses** —< **orders** —< **order\_items**
* **orders(1)** —< **shipments** —< **tracking\_events**
* **orders(1)** —< **payments** —< **payment\_attempts**; **payments(1)** —< **refunds**
* **products(1)** —< **product\_variants** —< **inventory\_ledger**
* **products( M )**—< **product\_categories** >—(1) **categories**
* **products(1)** —< **media** ; **products(M)**—< **collections\_link** >—(1) **collections**
* **customers(1)** —< **carts** —< **cart\_items**
* **customers(1)** —< **wishlists** —< **wishlist\_items**
* **orders(1)** —< **rmas** —< **rma\_items** (ترتبط أيضًا بـ shipments/logistics)
* **promotions** ترتبط بـ orders و/أو order\_items (خصم سطر/طلب كامل)

# **خطة اختبارات قبول (مبنية على مستندك + إضافات عملية)**

1. **سلامة المخطط**: محاولات إدخال مخالفات (سعر سالب، SKU مكرر، هاتف غير E.164) تُرفض.
2. **CRUD أساسي** لكل كيان (منتج، نسخة، سلة، طلب، شحنة، دفع، RMA).
3. **تزامن/تعارض**: تحديثان متوازيان لنفس السلة/الطلب لا يفقدان البيانات.
4. **أداء**:
   * قائمة منتجات مصفّاة: ≤ 300ms على 50K variant مع فهرسة مناسبة.
   * إنشاء طلب (confirm→packed): ≤ 2s مع حجز مخزون.
   * تتبع شحنة “out\_for\_delivery” يظهر خلال ≤ 1 دقيقة من webhook.
5. **استعادة نسخة احتياطية**: محاكاة فشل واسترجاع ناجح موثّق.

# **تسليمات عمليّة (What we’ll ship)**

1. **ERD** (صورة + ملف قابل للتعديل) يغطي كل الجداول والعلاقات.
2. **قاموس بيانات**: جدول لكل حقل (النوع/الطول/NULL/DEFAULT/UNIQUE/CHECK/FK).
3. **Matrix القيود** (FK/UNIQUE/CHECK) مجمّعة.
4. **خطة فهارس** مع لائحة الاستعلامات النموذجية وخطط التنفيذ (EXPLAIN).
5. **سياسة نسخ/استعادة** برقم RPO/RTO وجدولة اختبارات.
6. **RBAC**: مصفوفة أدوار × جداول/أعمدة + سياسة Masking للـPII.
7. **تفصيل Ledger المخزون** (جداول، أسباب، روابط).
8. **RMA مكتمل**: حالات، قواعد، تكامل مالي/مخزني ورسائل.
9. **/docs/db.md**: تشغيل محلي، ترحيلات، اختبارات، الاستعادة، والممارسات.

# **أولويات التنفيذ**

1. ERD + قاموس بيانات + Matrix قيود.
2. Ledger المخزون + مشتريات/توريد (مصدر الحقيقة للمخزون والتكلفة).
3. فهارس + ميزانيات أداء + مراقبة (لوحات زمن الاستعلامات/أحجام الجداول).
4. RBAC/PII/Audit + سياسة نسخ/استعادة بأرقام RPO/RTO واختبار دوري.
5. إكمال RMA + ربطه بالمدفوعات/الشحن والمخزون.
6. توحيد القواميس (ألوان/مقاسات/أقمشة) قبل أي استيراد كبير — لمنع الفوضى النصية في الكتالوج.
7. طبقة التحليلات (facts/dims) للـ KPIs المذكورة.

ملاحظات

# **ملاحظات**

1. **تصميم ERD تفصيلي**
   * رسم علاقات الجداول الرئيسية (Customers, Orders, Payments, Shipments, RMA, Products, Variants, Inventory Ledger, Wallet).
   * إضافة قيود **Primary / Foreign Keys** بشكل واضح.
   * إعداد **قاموس بيانات (Data Dictionary)** يشرح كل جدول وحقل.
2. **Inventory Ledger (دفتر المخزون)**
   * تصميم جدول ledger يحوي كل حركات المخزون (شراء، استلام، حجز، شحن، إرجاع، جرد).
   * حساب الكميات (stock\_on\_hand و available\_to\_promise) بشكل ديناميكي من الـ ledger فقط.
   * منع التعديل المباشر على المخزون.
3. **RBAC / PII / Audit**
   * مصفوفة صلاحيات واضحة تحدد من يقرأ أو يعدل أي بيانات.
   * **Masking** للهاتف والعنوان عند عرض البيانات لموظفين غير مخوّلين.
   * إنشاء جداول **سجل تدقيق (Audit Log)** append-only لتوثيق العمليات الحساسة.
4. **خطة النسخ الاحتياطي والاستعادة**
   * تحديد أرقام RPO (≤15 دقيقة) وRTO (≤1 ساعة).
   * وضع سياسة اختبار استعادة ربع سنوية والتوثيق الكامل لنتائجها.
5. **خطة الفهارس والأداء**
   * فهارس على الحقول الأكثر استخدامًا (sku, phone, order\_no, brand, color, size, customer\_id, status).
   * استخدام partial وcovering indexes حيث يلزم.
   * إضافة حدود زمنية للأداء (مثال: إنشاء طلب ≤2 ثانية، قائمة منتجات ≤300ms).
6. **توحيد القواميس والمعايير**
   * إنشاء جداول مرجعية للألوان، المقاسات، المدن، الخامات، وربطها مع المنتجات.
   * وضع Mapping مع خيارات البحث والفلاتر.
7. **الاختبارات وقبول التسليمات**
   * تعريف **Definition of Done** لكل كيان (CRUD + قيود + سيناريوهات فشل).
   * تضمين حالات edge cases (مثل SKU مكرر، رصيد سلبي، هاتف غير صحيح).
8. **التزامن والمعالجة المتزامنة**
   * استخدام أعمدة version أو updated\_at للتعامل مع التعارض (Optimistic Locking).
   * تطبيق مفاتيح Idempotency في عمليات الدفع والطلبات لتجنب التكرار.
9. **التحليلات والتقارير**
   * إنشاء fact tables (Orders, Payments, Inventory Movements, Traffic).
   * إضافة Materialized Views للـ Funnel، السلات المهجورة، العملاء المتكررين.
   * ربط بيانات UTM مع الطلبات لتتبع الحملات.
10. **المشتريات والتوريد**

* إضافة **Vendor KPIs** (التأخير، نسب الأخطاء، جودة الشحن).
* وضع قواعد إعادة الطلب (Reorder Rules) وفق المخزون والدوران.

علامة التبويب 2

## **فجوات للوصول إلى 95–100% (عملي ومختصر)**

1. **خصوصية وأمن**: تشفير الهاتف/الحقول الحساسة، إدارة مفاتيح، **RBAC** (صلاحيات حسب الدور)، **Audit Logs** للوصول والتعديلات.
2. **امتثال وقيود بيانات**: سياسة احتفاظ وحذف، مسارات **DSR** (تصدير/حذف بيانات العميل)، توثيق دليل الموافقة مع طابع زمني، تكامل قوائم **Suppression** مع مزودي الرسائل.
3. **حوكمة وجودة البيانات**: قاموس بيانات رسمي، قواعد تحقق وإغلاق التكرارات آليًا، فحوص تكامل مرجعي، اختبارات بيانات دورية.
4. **اعتمادية وتشغيل**: نسخ احتياطي واستعادة مُختبرة، أهداف **RPO/RTO**، نهج للهجرات والترحيلات مع تراجع (rollback) واضح.
5. **قابلية المراقبة**: لوحات مراقبة زمن/أخطاء واجهات CRUD، تنبيهات، **SLO** لزمن الاستجابة والأخطاء.
6. **الأداء والتوسّع**: فهارس/تقسيم جداول عالية الحجم (contact\_log)، خطة أرشفة، اختبارات ضغط (P95).
7. **واجهات واندماجات**: Pagination/Search موحَّدة، **Rate limiting** و**Idempotency** للواجهات، مخطط أحداث (webhooks) قياسي.
8. **تحليلات متقدمة**: ETL ثابت إلى Warehouse (مثلاً BigQuery)، تعريف موحَّد للأحداث والقُمع، Cohorts/Retention قياسية.
9. **تشغيل COD**: Runbook لضبط عتبات الحظر/الفكّ، تقارير فشل حسب المدينة/المندوب مع إجراءات تصحيح.
10. **أمن تطبيقي**: مراجعة أمنية/اختبار اختراق أساسي، إدارة أسرار مركزية.

\*معايير قاعدة البيانات ومنع الأخطاء

# **Zahraah – معايير قاعدة البيانات ومنع الأخطاء**

**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)**

**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.

العملاء (Customers)

# **العملاء (Customers)**

# **1) الهدف**

* توحيد هوية العميل (هاتف/إيميل/اسم) وربطها بعناوينه وأجهزته وسلوكه.
* تمكين رسائل CRM ذكية (ترحيب، سلة مهجورة، استرجاع، VIP).
* تحسين تجربة التوصيل وملف الدفع عند الاستلام (COD) وتقليل الرفض.
* تجهيز بيانات واضحة للتحليلات والقرار التسويقي (LTV، RFM، الاحتفاظ).

## **2) النطاق (ما المطلوب بناؤه)**

* **العملاء**: سجل موحّد لكل عميل + مؤشرات تشغيلية أساسية (عدد الطلبات، آخر طلب، القيمة العمرية…).
* **العناوين**: شحن/فوترة بدقة محلية (مدينة/حي/معلم قريب) مع عنوان افتراضي لكل عميل.
* **الأجهزة**: ربط أجهزة التطبيق بالعميل + حفظ توكن الإشعارات وإصدار التطبيق.
* **الموافقات**: إدارة موافقات القنوات (Push/SMS/WhatsApp/Email) واحترامها في أي إرسال.
* **الشرائح**: تعريف شرائح (VIP، غير نشط، مهجور سلة…) وعضوية العملاء فيها.
* **ملف COD**: حقول بسيطة لمخاطر COD (مسموح/محظور مؤقتًا/عدد محاولات فاشلة/درجة مخاطرة).
* **سجل التواصل**: أرشفة أي رسالة مرسلة للعميل (القناة، الحالة، الوقت) للمتابعة والامتثال.
* **التحويلات التسويقية**: حفظ أول وآخر مصدر تسويقي (source/medium/campaign) للفهم والقياس.
* **دمج المكررات**: آلية لتجميع حسابات مكررة لنفس العميل عند الحاجة.
* **(اختياري)** رصيد متجر/بطاقات هدية + تذاكر دعم مرتبطة بالعميل.

## **3) العلاقات الرئيسية**

* عميل ↔ عناوين: واحد إلى متعدد.
* عميل ↔ أجهزة: واحد إلى متعدد.
* عميل ↔ شرائح: متعدد إلى متعدد.
* عميل ↔ سجل تواصل/تذاكر دعم: واحد إلى متعدد.

## **4) قواعد العمل الأساسية**

* **الهاتف فريد** وبصيغة دولية E.164، والتحقق عبر OTP.
* عند توثيق الهاتف: **ربط** أي سلة/جهاز سابق بهذا العميل.
* **عنوان الطلب لقطة ثابتة** داخل الطلب (لا يُحدّث بأثر رجعي إذا غيّر العميل عنوانه لاحقًا).
* **COD**: زيادة درجة المخاطرة عند عدم الرد/رفض الاستلام؛ إيقاف COD مؤقتًا عند تجاوز عتبة محددة.
* احترام **الموافقات**: العميل الرافض لقناة معيّنة لا يُستهدف بها.
* تحديث **توكن الإشعارات** والإصدار عند كل فتح للتطبيق.

## **5) تدفّقات تشغيل مختصرة**

* **تسجيل/تعرّف عميل**: يبدأ زائرًا (جهاز فقط) → يوثّق الهاتف → يصبح عميلًا معروفًا، وتُنقل له السلة والجهاز.
* **تحديث العنوان الافتراضي**: عنوان واحد افتراضي دائمًا لكل عميل.
* **سلة مهجورة**: اكتشاف سلال بلا شراء خلال 7 أيام → استهداف بإشعار/عرض مخصص.
* **إدارة COD**: رفع أو خفض مخاطر العميل بناءً على نتائج الشحن الفعلية.

## **6) مهام مجدولة (Jobs)**

* **بناء الشرائح** يوميًا (VIP، غير نشط، مهجور 7 أيام…).
* **تحديث مخاطر COD** عند تغيّر حالات الشحن + مراجعة يومية.
* **فحص مكررات العملاء** (نفس الهاتف/نفس الجهاز) ووضعها في طابور مراجعة.

## **7) تكاملات لازمة**

* ربط التطبيق بـ **Firebase/GA4** وتعيين user\_id = معرف العميل.
* إرسال أحداث أساسية: عرض منتج، إضافة للسلة، إكمال شراء، فتح جلسة…
* استخدام **FCM** للإشعارات مع روابط عميقة (Deep Links) لفتح السلة/المنتج مباشرة.

## **8) التقارير والمؤشرات (الحد الأدنى)**

* **RFM** (حداثة آخر شراء، التكرار، الإنفاق).
* **LTV** والقيمة المتوسطة للطلب.
* **معدلات الاحتفاظ** (أسبوعي/شهري).
* **معدلات فشل COD** حسب المدينة/شركة الشحن.
* **حجم السلال المهجورة** وقيمتها.

## **9) معايير قبول (يعتبر العمل مكتملًا إذا…)**

* إنشاء وتحديث عميل يعمل مع تحقق الهاتف وتوحيد الهوية.
* عنوان افتراضي واحد فقط لكل عميل ويُستخدم تلقائيًا في الطلب.
* الأجهزة مربوطة بالعميل وتُحدَّث معلوماتها وتوكِن الإشعارات باستمرار.
* الشرائح تُبنى دوريًا ويمكن استهدافها في الحملات.
* الموافقات محترمة تلقائيًا في أي إرسال جماعي.
* ملف COD يتأثر تلقائيًا بنتائج الشحن ويوقف COD عند العتبة.
* تتوفر تقارير أساسية (RFM/LTV/Retention/COD) للاطلاع.

## **10) متطلبات جودة البيانات**

* توحيد أسماء المدن/الأحياء بقوائم مقيدة (لتحسين التوصيل والتقارير).
* طوابع زمنية للإنشاء والتحديث بكل السجلات.
* خصوصية: عدم إظهار بيانات حساسة لفِرق لا تحتاجها، وتقييد الوصول حسب الدور.

**ملاحظات قصيرة;**

* ركّز على بساطة الجداول وقابلية التوسّع، ولا تربط التنفيذ بأداة محددة.
* وفّر نقاط تكامل لاحقة مع الطلبات والسلال بدون كسر التصميم.
* اجعل الهاتف هو المعرف العملي الأول، مع دعم الدمج عند اكتشاف تكرار.

# 

علامة التبويب 22

# **توضيح لنقطة التحويلات ومعرفة من اين جاء العميل**

# **1) الفكرة باختصار**

* **First touch**: أول قناة وصل منها المستخدم (نحفظها مرة واحدة ولا نبدّلها).
* **Last touch**: آخر قناة “مؤهّلة” قبل التحويل (نحدّثها كل ما ظهرّت حملة/UTM جديدة).
* **قاعدة مهمّة**: لا نسمح لمصدر **direct / (none)** أنه يكتب فوق مصدر سابق غير مباشر.

# **2) الحقول القياسية التي سنجمعها**

* utm\_source, utm\_medium, utm\_campaign (+ اختياري: utm\_term, utm\_content)
* معرفات الشبكات: gclid (Google Ads), fbclid (Meta), ttclid (TikTok) … إلخ
* referrer\_domain (مصدر الإحالة إن لم توجد UTM)
* طابع زمني: first\_touch\_ts, last\_touch\_ts
* **قناة موحّدة** (Mapping): channel\_group (Organic Social, Paid Social, Paid Search, Referral, Direct…).

# **3) قواعد الإسناد (Recommended)**

* **First touch**: أوّل زيارة/تنصيب مع UTM/CLID/Referrer ≠ direct → تُحفظ 2 سنة.
* **Last touch**: آخر زيارة فيها **مصدر مؤهّل** خلال نافذة 90 يوم.
* **Direct** لا يكتب فوق آخر مصدر غير مباشر خلال 90 يوم.
* **Qualifying touch** = وجود أي UTM/CLID أو referrer domain خارجي.
* **Cross-device**: عند تسجيل الدخول، نربط first/last بالمستخدم (user\_id) ونمنع الازدواجية.

# **4) التنفيذ على الويب (JavaScript)**

### **أ) الالتقاط + التخزين المحلي**

* نحفظ **First** في Cookies طويلة (730 يوم).
* نحفظ **Last** في Cookies 90 يوم، ونحدّثها فقط عند لمسة مؤهّلة.
* عند عدم وجود UTM لكن في document.referrer دومين خارجي → source=referrer\_domain, medium=referral, campaign=(referral).

**مثال كود مختصر (مفهومياً):**

<script>

function getParams() {

const u = new URL(location.href);

const qp = (k) => u.searchParams.get(k);

const params = {

utm\_source: qp('utm\_source'),

utm\_medium: qp('utm\_medium'),

utm\_campaign: qp('utm\_campaign'),

gclid: qp('gclid'),

fbclid: qp('fbclid'),

};

// Referrer fallback

const ref = document.referrer;

const refDomain = ref ? (new URL(ref)).hostname : '';

const isDirect = !ref || refDomain === location.hostname;

if (!params.utm\_source && !params.gclid && !params.fbclid && !isDirect) {

params.utm\_source = refDomain;

params.utm\_medium = 'referral';

params.utm\_campaign = '(referral)';

}

return { ...params, refDomain };

}

function setCookie(n, v, days) {

const d = new Date(); d.setTime(d.getTime() + (days\*24\*60\*60\*1000));

document.cookie = `${n}=${encodeURIComponent(v||'')};path=/;expires=${d.toUTCString()};SameSite=Lax`;

}

function getCookie(n) {

return document.cookie.split('; ').find(r => r.startsWith(n+'='))?.split('=')[1];

}

function hasQualifying(p){ return p.utm\_source || p.gclid || p.fbclid; }

(function main(){

const p = getParams(); if (!p) return;

// FIRST

if (!getCookie('ft\_source') && hasQualifying(p)) {

setCookie('ft\_source', p.utm\_source, 730);

setCookie('ft\_medium', p.utm\_medium||'(none)', 730);

setCookie('ft\_campaign', p.utm\_campaign||'(none)', 730);

setCookie('first\_touch\_ts', Date.now(), 730);

}

// LAST (ignore "direct")

if (hasQualifying(p)) {

setCookie('lt\_source', p.utm\_source, 90);

setCookie('lt\_medium', p.utm\_medium||'(none)', 90);

setCookie('lt\_campaign', p.utm\_campaign||'(none)', 90);

setCookie('last\_touch\_ts', Date.now(), 90);

}

})();

</script>

### **ب) الإرسال إلى الخادم وGA4**

* عند **Sign-Up** و**Checkout** نرسل ft\_\* وlt\_\* مع الطلب.
* في GA4 نضبط **user\_properties**: first\_source/medium/campaign (مرة واحدة) وlast\_source/medium/campaign(يُحدّث عند كل جلسة مؤهّلة).
* نُمرّر نفس القيم في حدث purchase.

# **5) التنفيذ على التطبيق (Flutter + Firebase)**

### **أ) الالتقاط (Deferred Deep Links / Dynamic Links)**

* استخدم **Firebase Dynamic Links** لالتقاط UTM عند التنصيب والفتح.
* Android: إن توفر Play Install Referrer (اختياري) لالتقاط مصدر التنصيب.
* خزّن **First** في SharedPreferences عند أول فتح مؤهّل، و**Last** تُحدّث عند كل فتح برابط فيه UTM/CLID.

**مثال Flutter (مختصر):**

// pubspec.yaml: firebase\_dynamic\_links, shared\_preferences, firebase\_analytics

Future<Map<String,String>> \_extractParams(PendingDynamicLinkData? data) async {

final uri = data?.link;

if (uri == null) return {};

String q(String k)=> uri.queryParameters[k] ?? '';

return {

'utm\_source': q('utm\_source'),

'utm\_medium': q('utm\_medium'),

'utm\_campaign': q('utm\_campaign'),

'gclid': q('gclid'),

'fbclid': q('fbclid'),

};

}

Future<void> initAttribution() async {

final prefs = await SharedPreferences.getInstance();

// Initial link (first open)

final initData = await FirebaseDynamicLinks.instance.getInitialLink();

var p = await \_extractParams(initData);

// First touch

bool firstExists = prefs.containsKey('ft\_source');

bool qualifying = p.values.any((v) => v.isNotEmpty);

if (!firstExists && qualifying) {

await prefs.setString('ft\_source', p['utm\_source'] ?? '');

await prefs.setString('ft\_medium', p['utm\_medium'] ?? '(none)');

await prefs.setString('ft\_campaign', p['utm\_campaign'] ?? '(none)');

await prefs.setInt('first\_touch\_ts', DateTime.now().millisecondsSinceEpoch);

}

// Last touch

if (qualifying) {

await prefs.setString('lt\_source', p['utm\_source'] ?? '');

await prefs.setString('lt\_medium', p['utm\_medium'] ?? '(none)');

await prefs.setString('lt\_campaign', p['utm\_campaign'] ?? '(none)');

await prefs.setInt('last\_touch\_ts', DateTime.now().millisecondsSinceEpoch);

}

// Listen for future links (app in foreground)

FirebaseDynamicLinks.instance.onLink.listen((d) async {

final p2 = await \_extractParams(d);

final q2 = p2.values.any((v)=>v.isNotEmpty);

if (q2) {

await prefs.setString('lt\_source', p2['utm\_source'] ?? '');

await prefs.setString('lt\_medium', p2['utm\_medium'] ?? '(none)');

await prefs.setString('lt\_campaign', p2['utm\_campaign'] ?? '(none)');

await prefs.setInt('last\_touch\_ts', DateTime.now().millisecondsSinceEpoch);

}

});

}

### **ب) إرسالها لــ Firebase Analytics + الخادم**

final analytics = FirebaseAnalytics.instance;

Future<void> setUserPropsFromPrefs() async {

final p = await SharedPreferences.getInstance();

// set once

if (p.containsKey('ft\_source')) {

analytics.setUserProperty(name: 'first\_source', value: p.getString('ft\_source'));

analytics.setUserProperty(name: 'first\_medium', value: p.getString('ft\_medium'));

analytics.setUserProperty(name: 'first\_campaign', value: p.getString('ft\_campaign'));

}

// update-able

if (p.containsKey('lt\_source')) {

analytics.setUserProperty(name: 'last\_source', value: p.getString('lt\_source'));

analytics.setUserProperty(name: 'last\_medium', value: p.getString('lt\_medium'));

analytics.setUserProperty(name: 'last\_campaign', value: p.getString('lt\_campaign'));

}

}

وعند الشراء:

await analytics.logEvent(

name: 'purchase',

parameters: {

'value': orderTotal,

'currency': 'YER',

'first\_source': p.getString('ft\_source'),

'first\_medium': p.getString('ft\_medium'),

'first\_campaign': p.getString('ft\_campaign'),

'last\_source': p.getString('lt\_source'),

'last\_medium': p.getString('lt\_medium'),

'last\_campaign': p.getString('lt\_campaign'),

},

);

// وأرسل نفس الحقول إلى API إنشاء الطلب لتخزينها في الـDB

# **6) الخادم وقاعدة البيانات (مهم جدًا)**

## **أ) في جدول المستخدمين users**

* first\_source, first\_medium, first\_campaign, first\_touch\_ts
* last\_source, last\_medium, last\_campaign, last\_touch\_ts
* channel\_group\_first, channel\_group\_last

نحدّث حقول **first** لأول مرة فقط.  
نحدّث **last** عند كل جلسة مؤهّلة أو قبل إنشاء الطلب.

## **ب) في جدول الطلبات orders**

* order\_id, user\_id, order\_ts, value
* **حقول الإسناد المجلوبة وقت الشراء**:
  + first\_source/medium/campaign
  + last\_source/medium/campaign
  + session\_source/medium/campaign (اللمسة الحالية إن وُجدت)
  + channel\_group\_first, channel\_group\_last, channel\_group\_session

بهذه الطريقة تقدر تعمل تقارير **عبر الزمن** بدون الاعتماد فقط على GA4.

# **7) GA4 & BigQuery (تقارير)**

* في GA4: أنشئ **Dimensions مخصّصة** من user\_properties و event parameters (first\_*, last\_*, session\_\*).
* في BigQuery: ابنِ جدول أبعاد للمستخدم dim\_users من أحدث user\_properties، وجدول حقائق fact\_orders من الـDB أو من purchase events.

**استعلام BQ مثال (استخراج user\_props من GA4 export):**

WITH users AS (

SELECT

user\_pseudo\_id,

MAX(IF(ukey = 'first\_source', uval, NULL)) AS first\_source,

MAX(IF(ukey = 'first\_medium', uval, NULL)) AS first\_medium,

MAX(IF(ukey = 'first\_campaign', uval, NULL)) AS first\_campaign,

MAX(IF(ukey = 'last\_source', uval, NULL)) AS last\_source,

MAX(IF(ukey = 'last\_medium', uval, NULL)) AS last\_medium,

MAX(IF(ukey = 'last\_campaign', uval, NULL))AS last\_campaign

FROM (

SELECT

user\_pseudo\_id,

up.key AS ukey,

COALESCE(up.value.string\_value, CAST(up.value.int\_value AS STRING)) AS uval

FROM `project.dataset.events\_\*`,

UNNEST(user\_properties) up

)

GROUP BY user\_pseudo\_id

)

SELECT \* FROM users;

# **8) توحيد القنوات (Channel Grouping)**

ابنِ دالة Mapping بسيطة حسب source/medium/clid:

* إذا gclid → Paid Search (Google Ads)
* إذا fbclid أو utm\_source in (facebook, instagram, meta) → Paid Social
* إذا utm\_medium=organic\_social → Organic Social
* referral → Referral
* بدون أي شيء ومع referrer فارغ → Direct

خزّن channel\_group\_first/last/session في نفس الوقت.

# **9) حالات خاصة يجب تغطيتها**

* **WhatsApp / Telegram**: غالبًا referrer مفقود → استخدم روابط UTM للمنشورات والبايو والبنرات داخل التطبيق.
* **المكالمات/الطلبات الهاتفية (COD)**: أعطِ الفريق حقلاً attribution\_code يربط الإحالة بالطلب.
* **Internal traffic**: استبعد دوميناتك من الإحالات.
* **Cross-device**: عند تسجيل الدخول اربط السجلات بـ user\_id وقم بمزامنة first/last إلى حساب المستخدم مرّة واحدة.

# **10) اختبار وضمان الجودة (QA)**

* سيناريوهات:
  1. دخول عبر Google Ads → تسجيل → شراء: يجب أن تكون first=google/cpc/… وlast=google/cpc/…
  2. دخول Organic Search ثم بعد أسبوع دخول Instagram Ads ثم شراء: first=organic، last=paid social.
  3. دخول via campaign ثم عودة Direct فقط: لا يُسمح لـ Direct أن يكتب فوق last خلال 90 يوم.
* راقب نسب Direct بعد الإطلاق (يجب أن تنخفض إن كان الالتقاط صحيحًا).
* قارن الحساب بين **orders DB** و**GA4** يوميًا.

# **11) قائمة تنفيذ سريعة (Checklist)**

1. **ويب**: كود التقاط UTM+Referrer + Cookies (First/Last) + إرسالها للـAPI وGA4.
2. **تطبيق**: Dynamic Links + حفظ First/Last في SharedPreferences + user\_properties + تمريرها في purchase.
3. **خادم**: حقول في users وorders + منطق عدم الكتابة فوق First وعدم السماح لـ Direct بالكتابة فوق Last.
4. **Channel mapping**: دالة موحّدة.
5. **GA4**: إنشاء أبعاد مخصّصة من user\_properties وEvent params.
6. **BigQuery**: بناء جداول dim/fact والاستعلامات.
7. **QA**: سيناريوهات، لوحات مراقبة، مقارنة يومية.

علامة التبويب 23

**توضيح ل ربط كل جهاز بعميل معيّن** وتخزين **توكن الإشعارات (FCM token)** و**إصدار التطبيق** مع بيانات الجهاز بشكل منظّم وآمن، بحيث:

* نقدر نرسل إشعارات دقيقة (طلبات، عروض، استرجاع سلة…).
* نراقب الإصدارات ونفرض التحديثات الحرجة عند الحاجة.
* نتعامل مع تغيّر التوكن تلقائياً وننظّف التوكنات المعطوبة.

# **1) نموذج البيانات في قاعدة البيانات**

## **جدول devices**

* device\_id (UUID, PK)
* user\_id (FK nullable) — يرتبط عند تسجيل الدخول
* fid (Firebase Installation ID, unique nullable)
* fcm\_token (string, unique index)
* platform (enum: android/ios)
* device\_model (e.g., “iPhone15,2” أو “SM-A546E”)
* os\_version (e.g., “iOS 17.5”, “Android 14”)
* app\_version\_name (e.g., “2.7.3”)
* app\_build\_number (int, لفرض الحد الأدنى للإصدار)
* lang (ISO, مثل ar, en)
* tz\_offset\_minutes (int) فرق التوقيت
* push\_permission (bool) هل المستخدم أعطى إذن الإشعارات
* marketing\_opt\_in (bool) موافقة تسويقية (اختياري)
* is\_primary (bool) جهاز المستخدم الرئيسي (اختياري)
* token\_status (enum: active/invalid/expired)
* first\_seen\_ts (datetime)
* last\_seen\_ts (datetime)
* last\_push\_sent\_ts (datetime, اختياري)
* last\_push\_result (string, اختياري)

فهارس مهمّة:  
(user\_id), (fcm\_token UNIQUE), (fid UNIQUE), (last\_seen\_ts)

## **جدول أحداث اختياري device\_events**

* device\_id, event (token\_refresh, login, logout, push\_failed\_unregistered, app\_open, update)
* ts, meta (JSON)

# **2) تدفّق العمل (Lifecycle)**

1. **التثبيت/الفتح الأول**: التطبيق يجمع fid, fcm\_token, بيانات الجهاز/الإصدار ويرسل **Register**.
2. **تسجيل الدخول**: استدعاء **Associate** لربط device\_id مع user\_id.
3. **تحديث التوكن**: عند تغيّر FCM token يستدعي **Token Refresh** (Patch).
4. **نبضة حياة (Heartbeat)**: عند كل فتح (وأحياناً دوريًا) يحدّث last\_seen وapp\_version وpush\_permission.
5. **تسجيل الخروج**: **Dissociate** (نفصل user\_id عن الجهاز لكن نحتفظ بالتوكن لو أردنا إشعارات عامة).
6. **إرسال الإشعار**: السيرفر يرسل للتوكنات الفعّالة؛ لو رجع FCM بخطأ NotRegistered نعلّم التوكن invalid ونوقّفه.

# **3) واجهات الـAPI (مقترحة)**

جميعها خلف **Auth** مناسبة (JWT أو Firebase Auth) + معدل الطلبات (Rate limit).

### **POST /v1/devices/register**

**Request**

{

"fid": "abc123...",

"fcm\_token": "fcm\_token\_here",

"platform": "android",

"device\_model": "SM-A546E",

"os\_version": "Android 14",

"app\_version\_name": "2.7.3",

"app\_build\_number": 20703,

"lang": "ar",

"tz\_offset\_minutes": 180,

"push\_permission": true,

"marketing\_opt\_in": false

}

**Response**

{ "device\_id": "b2f0b3f6-...", "token\_status": "active" }

**Upsert** بالاعتماد على fid أو fcm\_token. حدّث السجلات بدل تكرارها.

### **POST /v1/devices/associate**

يربط الجهاز بالمستخدم (يُستدعى بعد تسجيل الدخول).

{ "device\_id": "b2f0b3f6-...", "user\_id": "U12345" }

### **POST /v1/devices/dissociate**

عند تسجيل الخروج:

{ "device\_id": "b2f0b3f6-..." }

### **PATCH /v1/devices/token**

عند تحديث التوكن:

{ "device\_id": "b2f0b3f6-...", "fcm\_token": "new\_token\_here" }

### **POST /v1/devices/heartbeat**

تحديث دوري/عند الفتح:

{

"device\_id": "b2f0b3f6-...",

"app\_version\_name": "2.8.0",

"app\_build\_number": 20800,

"push\_permission": true

}

# **4) تطبيق Flutter (عملي)**

## **الحِزم**

* firebase\_core, firebase\_messaging
* package\_info\_plus (الإصدار/الـbuild)
* device\_info\_plus (موديل/OS)
* (اختياري) firebase\_auth لو كنت تعتمد توثيق Firebase

## **التهيئة + التسجيل الأول**

import 'package:firebase\_core/firebase\_core.dart';

import 'package:firebase\_messaging/firebase\_messaging.dart';

import 'package:package\_info\_plus/package\_info\_plus.dart';

import 'package:device\_info\_plus/device\_info\_plus.dart';

import 'dart:io';

Future<void> initPushAndDevice() async {

await Firebase.initializeApp();

// iOS: طلب الإذن – Android 13+ أيضاً يحتاج POST\_NOTIFICATIONS

final settings = await FirebaseMessaging.instance.requestPermission(

alert: true, badge: true, sound: true,

);

final pushPermission = settings.authorizationStatus == AuthorizationStatus.authorized ||

settings.authorizationStatus == AuthorizationStatus.provisional;

final fcm = FirebaseMessaging.instance;

final token = await fcm.getToken(); // قد تكون null لو لم يُمنح الإذن

final fid = await fcm.getToken(vapidKey: null); // بديل غير دقيق؛ الأفضل استخدام FIS عبر native، لكن نكتفي بالتوكن كمُعرّف

final pkg = await PackageInfo.fromPlatform();

final deviceInfo = DeviceInfoPlugin();

String model, osVer, platform;

if (Platform.isAndroid) {

final a = await deviceInfo.androidInfo;

model = "${a.manufacturer} ${a.model}";

osVer = "Android ${a.version.release}";

platform = "android";

} else {

final i = await deviceInfo.iosInfo;

model = i.utsname.machine ?? "iPhone";

osVer = "iOS ${i.systemVersion}";

platform = "ios";

}

// نادى API register

await apiPost('/v1/devices/register', {

"fid": null, // إن توفر لديك طريقة للحصول على FID أصلي أرسله، وإلا اتركه null

"fcm\_token": token,

"platform": platform,

"device\_model": model,

"os\_version": osVer,

"app\_version\_name": pkg.version,

"app\_build\_number": int.tryParse(pkg.buildNumber) ?? 0,

"lang": Platform.localeName.split('\_').first,

"tz\_offset\_minutes": DateTime.now().timeZoneOffset.inMinutes,

"push\_permission": pushPermission,

"marketing\_opt\_in": false

});

// استمع لتحديثات التوكن

FirebaseMessaging.instance.onTokenRefresh.listen((newToken) async {

await apiPatch('/v1/devices/token', {

"device\_id": await currentDeviceId(), // خزّنه بعد register

"fcm\_token": newToken

});

});

}

## **عند تسجيل الدخول/الخروج**

Future<void> onLogin(String userId) async {

await apiPost('/v1/devices/associate', {

"device\_id": await currentDeviceId(),

"user\_id": userId

});

}

Future<void> onLogout() async {

await apiPost('/v1/devices/dissociate', {

"device\_id": await currentDeviceId()

});

}

## **نبضة حياة عند الفتح**

Future<void> heartbeat() async {

final pkg = await PackageInfo.fromPlatform();

await apiPost('/v1/devices/heartbeat', {

"device\_id": await currentDeviceId(),

"app\_version\_name": pkg.version,

"app\_build\_number": int.tryParse(pkg.buildNumber) ?? 0,

"push\_permission": await hasPushPermission()

});

}

ملاحظات:

* Android 13+ يتطلب إذن الإشعارات Runtime — تأكد من إضافته في Manifest وطلبه.
* توكِن FCM **قد يتغيّر** في أي وقت: الاستماع لـ onTokenRefresh ضروري.
* لو رفض الإذن، خزِّن push\_permission=false لتجنب إرسال إشعار فاشل.

# **5) الإرسال من السيرفر + تنظيف التوكنات**

(مثال TypeScript مع Firebase Admin)

import { getMessaging } from 'firebase-admin/messaging';

async function sendToUser(userId: string, payload: any) {

const tokens = await db.tokens.findActiveByUser(userId); // رجّع قائمة FCM tokens

if (!tokens.length) return;

const res = await getMessaging().sendEachForMulticast({

tokens,

notification: { title: payload.title, body: payload.body },

data: payload.data ?? {}

});

// نظّف التوكنات غير الصالحة

res.responses.forEach((r, idx) => {

if (!r.success) {

const err = r.error?.code;

if (err === 'messaging/registration-token-not-registered' ||

err === 'messaging/invalid-registration-token') {

db.tokens.markInvalid(tokens[idx]); // token\_status=invalid

}

}

});

}

* استخدم سياسة إعادة المحاولة (Exponential backoff) عند quotaExceeded أو internal.
* حدّس last\_push\_sent\_ts وlast\_push\_result.

# **6) السياسات والقواعد**

* **تعدّد الأجهزة**: اسمح بعدة أجهزة لكل مستخدم. أرسل إشعار الطلب لكل الأجهزة الفعّالة.
* **الخصوصية**: لا تحفظ معرّفات إعلانية حسّاسة (IDFA) بدون موافقة. fid وfcm\_token كافيان.
* **الأمن**: اربط كل استدعاء Device بــ JWT للمستخدم إن وُجد، أو توكن ضيف (Guest) قصير العمر.
* **الحد الأدنى للإصدار**: احتفِظ بإعداد مركزي min\_supported\_build. في كل **Heartbeat** أو عند أي طلب API، إن كان app\_build\_number < min\_supported\_build أرجِع 426 (Upgrade Required) مع رابط المتجر.
* **الاختيارات التسويقية**: احترم marketing\_opt\_in، واستخدم **Topics** اختيارية (مثل promos, new\_arrivals) للمشتركين فقط.

# **7) اختبار (QA) سريع**

* تثبيت جديد → Register يتم إنجاحه والتوكن محفوظ.
* رفض الإذن ثم منحه لاحقاً → push\_permission يتحدث وonTokenRefresh يشتغل.
* دخول من جهازين لنفس الحساب → كلاهما يستلم إشعار حالة الطلب.
* حذف التطبيق/إعادة تثبيت → إرسال إشعار يجب أن يرجع NotRegistered للتوكن القديم ويتم تنظيفه.
* تحديث إصدار التطبيق → app\_version يتحدث في heartbeat.
* تسجيل خروج → الجهاز ينفصل عن user\_id لكن يبقى صالح لإشعارات عامّة (إن رغبت).

علامة التبويب 24

**إدارة موافقات القنوات (Push / SMS / WhatsApp / Email)** حسب **الغرض** (Transactional / Marketing / …) واحترامها قبل أي إرسال. دعم المواضيع (Topics) اختياريًا، وسجل تدقيقي كامل، والتكامل مع جدول الأجهزة (push\_permission).

# **1) التعاريف (Definitions)**

* **Channel**: قناة الإرسال (push, sms, whatsapp, email).
* **Purpose**: الغرض من الرسالة (transactional, marketing, survey, security).
* **Topic** (اختياري): تصنيف تسويقي دقيق (promos, new\_arrivals, …).
* **Status**: حالة الموافقة (granted, denied, revoked).
* **Evidence**: دليل جمع الموافقة (IP, device\_id, النص الظاهر للموافقة، timestamp).

# **2) مخطط البيانات (DDL)**

**ملاحظة**: غيّر أسماء الجداول/المخططات حسب اصطلاحاتك، وثبّت ترميز utf8mb4.

-- 2.1 بيانات تواصل المستخدم (للتحقق قبل الإرسال)

CREATE TABLE user\_contacts (

user\_id BIGINT NOT NULL,

email VARCHAR(255),

email\_verified TINYINT(1) NOT NULL DEFAULT 0,

email\_verified\_at DATETIME NULL,

phone\_e164 VARCHAR(32), -- +9677xxxxxx

phone\_verified TINYINT(1) NOT NULL DEFAULT 0,

phone\_verified\_at DATETIME NULL,

wa\_phone\_e164 VARCHAR(32) NULL, -- إن كان رقم واتساب مختلفاً

created\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

updated\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (user\_id),

UNIQUE KEY uq\_contacts\_email (email),

UNIQUE KEY uq\_contacts\_phone (phone\_e164),

UNIQUE KEY uq\_contacts\_wa\_phone (wa\_phone\_e164)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- 2.2 الموافقات الحالية (Effective)

CREATE TABLE user\_consents (

user\_id BIGINT NOT NULL,

channel ENUM('push','sms','whatsapp','email') NOT NULL,

purpose ENUM('transactional','marketing','survey','security') NOT NULL,

topic VARCHAR(64) NULL, -- NULL = موافقة عامة على الغرض في القناة

status ENUM('granted','denied','revoked') NOT NULL,

source ENUM('app\_checkbox','profile\_center','checkout\_optin','wa\_click\_to\_chat',

'sms\_keyword','email\_unsubscribe\_link','cs\_agent','system') NOT NULL,

evidence JSON NULL, -- {ip, device\_id, user\_agent, text, ts}

language CHAR(2) NULL, -- ar / en

jurisdiction VARCHAR(32) NULL,

created\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

updated\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (user\_id, channel, purpose, topic),

KEY idx\_consents\_user (user\_id),

KEY idx\_consents\_channel (channel),

KEY idx\_consents\_updated (updated\_at)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- 2.3 سجل تغيّر الموافقات (Audit)

CREATE TABLE user\_consent\_history (

id BIGINT NOT NULL AUTO\_INCREMENT,

user\_id BIGINT NOT NULL,

channel ENUM('push','sms','whatsapp','email') NOT NULL,

purpose ENUM('transactional','marketing','survey','security') NOT NULL,

topic VARCHAR(64) NULL,

prev\_status ENUM('granted','denied','revoked') NULL,

new\_status ENUM('granted','denied','revoked') NOT NULL,

source ENUM('app\_checkbox','profile\_center','checkout\_optin','wa\_click\_to\_chat',

'sms\_keyword','email\_unsubscribe\_link','cs\_agent','system') NOT NULL,

evidence JSON NULL,

changed\_by ENUM('user','system','agent') NOT NULL,

changed\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (id),

KEY idx\_hist\_user (user\_id),

KEY idx\_hist\_changed (changed\_at)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- 2.4 تكامل مع الأجهزة (من مواصفة الأجهزة لديك)

-- نضمن وجود push\_permission وإصدار التطبيق لإحترام دفع الإشعارات

-- مثال مبسّط للحقلين الأساسيين (جداولك قد تكون أوسع):

-- ALTER TABLE devices ADD COLUMN push\_permission TINYINT(1) NOT NULL DEFAULT 0;

-- ALTER TABLE devices ADD COLUMN app\_build\_number INT NOT NULL DEFAULT 0;

-- فهارسك موجودة مسبقًا (user\_id, fcm\_token, fid, last\_seen\_ts ...)

**ملاحظات معايير:**

* **المفتاح المركّب** في user\_consents يسمح بموافقة عامة (topic=NULL) وأخرى موضوعية (topic='promos').
* استخدم JSON في evidence لتخزين تفاصيل قابلة للتوسّع دون تغيير المخطط.
* جميع الجداول تحمل طوابع زمنية، مع فهارس على updated\_at و/أو changed\_at لسهولة التقارير.

# **3) سياسات افتراضية (Defaults)**

* افتراضيًا:
  1. **Transactional** عبر كل قناة = granted (ما لم يلغِ المستخدم قناة بعينها).
  2. **Marketing** لكل قناة = denied حتى opt-in صريح.
* **Push** يتطلب شرطين معًا للإرسال:
  1. موافقة تسويقية/خدمية حسب الرسالة في user\_consents.
  2. إذن نظام/جهاز: devices.push\_permission = 1 + توكن صالح.

# **4) عمليات الكتابة (Upsert) وأمثلة SQL**

### **4.1 قراءة مصفوفة الموافقات للمستخدم**

SELECT channel, purpose, topic, status, source, language, updated\_at

FROM user\_consents

WHERE user\_id = ?;

### **4.2 Upsert موافقة (عام/Topic)**

تطبيقياً: سجّل تغييرًا في user\_consent\_history **قبل** أو **بعد** الـUpsert (اختر نهجًا واحدًا وثبّته).

-- مثال: منح Email Marketing عام (بدون topic)

INSERT INTO user\_consents (user\_id, channel, purpose, topic, status, source, evidence, language)

VALUES (?, 'email', 'marketing', NULL, 'granted', 'profile\_center',

JSON\_OBJECT('ip', ?, 'text', 'I agree to receive marketing emails', 'ts', NOW()), 'ar')

ON DUPLICATE KEY UPDATE

status = VALUES(status),

source = VALUES(source),

evidence = VALUES(evidence),

language = VALUES(language),

updated\_at = CURRENT\_TIMESTAMP;

-- سجل تاريخ:

INSERT INTO user\_consent\_history (user\_id, channel, purpose, topic, prev\_status, new\_status, source, evidence, changed\_by)

VALUES (?, 'email', 'marketing', NULL, ?, 'granted', 'profile\_center',

JSON\_OBJECT('ip', ?, 'screen', 'preference\_center'), 'user');

### **4.3 إلغاء اشتراك (Unsubscribe)**

-- إلغاء Email Marketing:

INSERT INTO user\_consents (user\_id, channel, purpose, topic, status, source, evidence)

VALUES (?, 'email', 'marketing', NULL, 'revoked', 'email\_unsubscribe\_link',

JSON\_OBJECT('reason', 'unsubscribe\_link', 'ts', NOW()))

ON DUPLICATE KEY UPDATE

status = 'revoked',

source = VALUES(source),

evidence = VALUES(evidence),

updated\_at = CURRENT\_TIMESTAMP;

INSERT INTO user\_consent\_history (...)

VALUES (..., /\* prev\_status \*/ ?, 'revoked', 'email\_unsubscribe\_link', JSON\_OBJECT('message\_id', ?), 'user');

# **5) خوارزمية قرار الإرسال (Eligibility)**

**المدخلات**: user\_id, channel, message\_type (transactional|marketing), topic (اختياري).  
**المخرجات**: allowed (bool), reason (نص قصير).

### **5.1 قواعد عامة**

* **Transactional**:
  + مطلوب موافقة القناة نفسها ≠ denied/revoked + وسيلة الاتصال **مُتحقَّقة** (email\_verified/phone\_verified).
  + Push: يشترط devices.push\_permission=1 وتوكن صالح.
* **Marketing**:
  + يشترط status='granted' لنفس القناة والغرض.
  + إن وُجد Topic محدّد: تحترم موافقة الـTopic إن كانت موجودة؛ وإلا تسقط على الموافقة العامة للقناة/الغرض (topic=NULL).
  + لا “Fallback” لقناة أخرى إذا قناة معيّنة مرفوضة (احترام نية المستخدم).
* **قيود تشغيلية**: ساعات هادئة، وحدّ التكرار (Frequency Caps)، وتهيئة حد أدنى للإصدار (للـPush إن رغبت).

### **5.2 Pseudo-code (على الخادم)**

function can\_send(user\_id, channel, msg\_type, topic):

consents = SELECT \* FROM user\_consents WHERE user\_id=? AND channel=?;

contacts = SELECT \* FROM user\_contacts WHERE user\_id=?;

if msg\_type == 'transactional':

if channel in ('sms','whatsapp'):

if channel == 'sms' and contacts.phone\_verified == 0: return (false, 'phone\_not\_verified')

if channel == 'whatsapp' and (contacts.wa\_phone\_e164 is null or contacts.phone\_verified == 0):

return (false, 'wa\_not\_verified')

if channel == 'email' and contacts.email\_verified == 0: return (false, 'email\_not\_verified')

if channel == 'push':

if NOT exists valid device with push\_permission=1 and valid token: return (false, 'push\_permission\_denied')

st = effective\_status(consents, channel, 'transactional', topic)

if st in ('denied', 'revoked'): return (false, 'consent\_denied')

return (true, 'ok')

else if msg\_type == 'marketing':

st = effective\_status(consents, channel, 'marketing', topic)

if st != 'granted': return (false, 'marketing\_not\_opted\_in')

-- اختياري: تحقق ساعات هادئة وحدود التكرار هنا

if violates\_frequency\_caps(user\_id, channel): return (false, 'frequency\_cap')

if channel == 'push' and NOT exists valid push device: return (false, 'push\_permission\_denied')

if channel == 'email' and contacts.email\_verified == 0: return (false, 'email\_not\_verified')

if channel in ('sms','whatsapp') and contacts.phone\_verified == 0: return (false, 'phone\_not\_verified')

return (true, 'ok')

function effective\_status(consents, channel, purpose, topic):

-- أولوية: موافقة topic المحدّد ثم العامة (topic IS NULL)

row = consents.find(c => c.purpose==purpose && c.topic==topic)

|| consents.find(c => c.purpose==purpose && c.topic is null)

if row exists: return row.status else:

-- افتراضات: transactional=granted, marketing=denied (قابلة للتهيئة)

return (purpose=='transactional') ? 'granted' : 'denied'

# **6) دمج القنوات (Integrations)**

## **6.1 WhatsApp**

* ممنوع إرسال outbound تسويقي بدون opt-in قابل للإثبات.
* خزّن نص الموافقة في evidence (مثلاً: “أوافق على استلام رسائل واتساب من زهراء…”).
* Webhook لإشعارات **STOP/الغاء** من المستخدم → تحدّث user\_consents إلى revoked.

## **6.2 SMS**

* دعم كلمات مفاتيح: STOP, UNSUB, الغاء, CANCEL, START, موافقة.
* كل حدث Keyword يُسجّل في user\_consent\_history ويحدّث user\_consents.

## **6.3 Email**

* **Unsubscribe link** لكل رسالة تسويقية + ترويسة List-Unsubscribe إن أمكن.
* Webhook لمزوّد البريد لتحديث revoked عند الإلغاء.

## **6.4 Push**

* يعتمد على **موافقة القناة** + **إذن الجهاز** (من جدول devices).
* لا ترسل Marketing إن كان push\_permission=0 حتى لو status='granted' في user\_consents.

# **7) سجل الإرسال (للتدقيق والتقارير)**

يوصى بجدول خفيف لتوثيق قرار الإرسال ونتيجته.

CREATE TABLE message\_deliveries (

id BIGINT NOT NULL AUTO\_INCREMENT,

user\_id BIGINT NOT NULL,

channel ENUM('push','sms','whatsapp','email') NOT NULL,

message\_type ENUM('transactional','marketing','survey','security') NOT NULL,

topic VARCHAR(64) NULL,

decision\_allowed TINYINT(1) NOT NULL,

decision\_reason VARCHAR(64) NOT NULL,

provider\_message\_id VARCHAR(128) NULL,

provider\_code VARCHAR(64) NULL, -- NOT\_REGISTERED, UNSUBSCRIBED, ...

send\_status ENUM('queued','sent','failed','skipped') NOT NULL,

created\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (id),

KEY idx\_deliveries\_user (user\_id),

KEY idx\_deliveries\_created (created\_at)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

عند الإرسال: خزّن **لقطة قرار** (decision\_allowed/reason) ونتيجة المزوّد.  
عند أخطاء مثل UNSUBSCRIBED أو NOT\_REGISTERED: حدّث user\_consents أو devices.token\_status فورًا.

# **8) تدفّقات العمل (Flows) مختصرة**

## **8.1 جمع موافقة Marketing عبر مركز التفضيلات**

1. المستخدم يفعّل خيار القناة (Checkbox).
2. الخادم:
   * يكتب في user\_consents (Upsert).
   * يضيف سجلًا في user\_consent\_history.
3. الواجهة تحدّث الحالة بصريًا.

## **8.2 إلغاء اشتراك Email عبر الرابط**

1. المستخدم يضغط Unsubscribe.
2. Webhook → تحديث user\_consents إلى revoked + سجل في user\_consent\_history.
3. أي إرسال لاحق عبر Email/Marketing يُمنع.

## **8.3 إرسال رسالة**

1. خدمة الإرسال تستدعي can\_send(...).
2. لو allowed=false: تسجّل message\_deliveries بـ skipped.
3. لو allowed=true: ترسل للمزوّد، ثم تحدّث message\_deliveries بنتيجة الإرسال.
4. معالجة الأخطاء: تحديث consent/device عند الحاجة.

# **9) الفهارس والأداء (Indexes & Perf)**

* user\_consents: مفتاح مركّب PK + فهرس على updated\_at لتقاطعات التقارير.
* user\_consent\_history: فهارس على (user\_id) و (changed\_at) لقراءة سريعة للتدقيق.
* message\_deliveries: فهرس زمني (created\_at) للتقارير الدورية.
* اجعل عمليات الإرسال Bulk-safe (دفعات) مع **Exponential Backoff** عند أخطاء الحصة.

# **10) الأمان والخصوصية**

* لا تُخزّن بيانات حسّاسة أكثر من اللازم في evidence.
* كل تغيّر موافقة يجب أن يكون **قابلًا للإثبات** (من/إلى، ومتى، وكيف).
* احترم اللغة (AR/EN) عند نصوص الموافقة لتطابق ما رآه المستخدم.

# **11) اختبارات (QA) أساسية**

* **Email Marketing revoked** ⇒ أي إرسال Email/Marketing → can\_send = false.
* **Push permission = 0** رغم موافقة push/marketing=granted ⇒ منع الإرسال التسويقي عبر Push.
* **SMS START/STOP** تحدّث user\_consents آنيًا.
* **Topic-specific**: granted على topic='promos' فقط ⇒ إرسال new\_arrivals يُمنع.
* **Transactional** مع قناة غير موثّقة (email\_verified=0) ⇒ منع وإرجاع سبب واضح.

# **12) قابلية التوسّع (Migration Notes)**

* إضافة قناة جديدة: أضف قيمة إلى ENUMs أو بدّلها إلى جداول مرجعية (channels, purposes) إن رغبت.
* إضافة Topic جديد: لا لزوم لتغيير المخطط (حقل topic حر).
* نقل الافتراضات إلى جدول إعدادات (config) إن أردت تخصيص defaults لكل قناة/غرض.

# **13) تنفيذ**

1. إنشاء الجداول الثلاثة + (اختياري) message\_deliveries.
2. ربط مركز التفضيلات UI بــ Endpoints: **GET/PUT consents**.
3. وصل Webhooks: email unsubscribe, SMS keywords, WhatsApp opt-in/out.
4. تنفيذ خدمة can\_send(...) واحترامها في **كل** مسارات الإرسال.
5. احترام devices.push\_permission وfcm\_token الصالح للـPush.
6. إضافة مراقبة (Dashboards): نسب opt-in، معدلات Unsub، أسباب الحجب، أثر الموافقات على الإيراد.
7. تغطية اختبارات QA المذكورة.

علامة التبويب 25

**الشرائح: تعريف شرائح (VIP، غير نشط، مهجور سلة…) وعضوية العملاء فيها.**، آلية احتسابها وتخزينها، وقواعد دخول/خروج العميل من كل شريحة بصورة **حتمية وقابلة للتدقيق**. تُستخدم لاحقًا في الاستهداف، مع احترام الموافقات (consents) عند الإرسال.

# **1) القاموس (Taxonomy) والتعاريف**

القيم الزمنية قابلة للتهيئة في جدول إعدادات، الافتراضات أدناه عملية لتجارة أزياء مع دفع عند الاستلام.

**A) شرائح دورة الحياة (Lifecycle) — متنافية داخل المجموعة**

* NEW\_7D: أول شراء خلال آخر 7 أيام.
* ACTIVE\_30D: آخر شراء خلال آخر 30 يومًا.
* DORMANT\_31\_90D: لا يوجد شراء منذ 31–90 يومًا.
* CHURN\_RISK\_91\_180D: لا يوجد شراء منذ 91–180 يومًا.
* LOST\_180P: لا يوجد شراء منذ >180 يومًا.

**B) قيمة العميل (Value)**

* VIP: أعلى 5% من إجمالي الإيراد طيلة العمر **أو** (≥ 5 طلبات & AOV ≥ P90 آخر 90 يومًا).
* HIGH\_VALUE\_NEW: أول طلب بقيمة ≥ P90 من أول طلبات آخر 90 يومًا.

**C) السلوك داخل الجلسة/القُمع (Behavioral)**

* CART\_ABANDONER\_24H: أضاف للسلة ولم يتم الشراء خلال 1–24 ساعة من آخر إضافة.
* CHECKOUT\_ABANDONER\_6H: بدأ الدفع ولم يكمل خلال 15 دقيقة–6 ساعات.
* BROWSE\_ABANDONER\_24H: ≥3 عرض منتج/تصنيف خلال 24 ساعة دون إضافة سلة.

**D) الجودة/المخاطر**

* HIGH\_RETURN\_RATE: نسبة المرتجعات ≥ 30% خلال 90 يومًا و≥3 طلبات.
* COD\_RISK\_HIGH: درجة مخاطرة COD ≥ 0.8 (من نموذجك)، أو ≥2 رفض استلام آخر 90 يومًا.
* COUPON\_HUNTER: ≥70% من الطلبات بخصم خلال 180 يومًا و≥3 طلبات.

ملاحظة: يمكن لعميل أن يكون في **عدة شرائح** بالتوازي، عدا مجموعة دورة الحياة فهي متنافية ويُختار منها واحدة وفق الأسبقية.

**أولوية عرض “الشريحة الأساسية (primary)”**

1. VIP
2. CHURN\_RISK / LOST
3. ACTIVE / NEW
4. Abandoners (Cart/Checkout)
5. باقي الشرائح

# **2) مخطط الجداول (DDL)**

-- 2.1 قاموس الشرائح

CREATE TABLE segment\_catalog (

segment\_id BIGINT AUTO\_INCREMENT PRIMARY KEY,

code VARCHAR(64) NOT NULL UNIQUE, -- مثال: VIP, CART\_ABANDONER\_24H

name\_ar VARCHAR(128) NOT NULL,

name\_en VARCHAR(128) NOT NULL,

group\_code ENUM('lifecycle','value','behavior','risk') NOT NULL,

is\_mutually\_exclusive TINYINT(1) NOT NULL DEFAULT 0, -- lifecycle=true

priority INT NOT NULL DEFAULT 100, -- للأولوية داخل المجموعة

criteria\_json JSON NOT NULL, -- تعريف قابل للقراءة الآلية (نوافذ، حدود…)

ttl\_hours INT NULL, -- صلاحية عضوية تلقائية (مثلاً مهجور سلة 24h)

is\_active TINYINT(1) NOT NULL DEFAULT 1,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

updated\_at DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 2.2 عضوية العملاء في الشرائح

CREATE TABLE segment\_membership (

user\_id BIGINT NOT NULL,

segment\_id BIGINT NOT NULL,

status ENUM('active','inactive') NOT NULL DEFAULT 'active',

score DECIMAL(6,3) NULL, -- لقيم مثل COD\_RISK, RFM score…

first\_join\_ts DATETIME NOT NULL,

last\_enter\_ts DATETIME NOT NULL,

last\_exit\_ts DATETIME NULL,

evidence JSON NULL, -- {rule:'A1', window:'24h', cart\_id:..., last\_event\_ts:...}

updated\_at DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (user\_id, segment\_id),

KEY idx\_segment (segment\_id),

KEY idx\_status (status),

KEY idx\_updated (updated\_at),

CONSTRAINT fk\_membership\_catalog FOREIGN KEY (segment\_id) REFERENCES segment\_catalog(segment\_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 2.3 مؤشرات مجمّعة لكل عميل (تتحدث يوميًا/شبه لحظي)

CREATE TABLE dim\_user\_metrics (

user\_id BIGINT PRIMARY KEY,

first\_order\_ts DATETIME NULL,

last\_order\_ts DATETIME NULL,

orders\_count\_365 INT NOT NULL DEFAULT 0,

revenue\_365 DECIMAL(12,2) NOT NULL DEFAULT 0,

revenue\_lifetime DECIMAL(12,2) NOT NULL DEFAULT 0,

aov\_90d DECIMAL(12,2) NULL,

returns\_rate\_90d DECIMAL(5,2) NULL, -- 0..100

last\_session\_ts DATETIME NULL,

last\_add\_to\_cart\_ts DATETIME NULL,

last\_checkout\_ts DATETIME NULL,

cart\_value\_last DECIMAL(12,2) NULL,

coupon\_usage\_rate\_180d DECIMAL(5,2) NULL, -- 0..100

cod\_risk\_score DECIMAL(6,3) NULL,

city VARCHAR(64) NULL,

updated\_at DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**ملاحظات**

* criteria\_json يصف القواعد (مثال بالأسفل) لتسهيل توليد SQL/Jobs.
* ttl\_hours يتيح **انقضاء عضوية** تلقائيًا (مثل مهجور سلة بعد 24h إن لم تتكرر الحالة).
* يمكن بناء View تُظهر “الشريحة الأساسية” لكل عميل وفق priority وis\_mutually\_exclusive.

# **3) أمثلة تعريف معيارية (criteria\_json)**

تُدرج في segment\_catalog وقت التهيئة الأولى.

// CART\_ABANDONER\_24H

{

"events\_required": ["add\_to\_cart"],

"window\_hours": 24,

"exit\_on": ["purchase"],

"min\_cart\_value": 1,

"mutual\_group": "behavior"

}

// CHECKOUT\_ABANDONER\_6H

{

"events\_required": ["begin\_checkout"],

"window\_hours": 6,

"exit\_on": ["purchase"],

"mutual\_group": "behavior"

}

// VIP

{

"rule": "percentile\_or\_threshold",

"percentile": 95,

"field": "revenue\_lifetime",

"or": {"orders\_min": 5, "aov\_90d\_percentile": 90},

"mutual\_group": "value"

}

// DORMANT\_31\_90D

{

"rule": "last\_order\_between\_days",

"min\_days": 31,

"max\_days": 90,

"mutual\_group": "lifecycle",

"exclusive\_group": true

}

# **4) منطق الاحتساب والتحديث (Jobs & Triggers)**

**A) دفعة يومية (Batch / BigQuery أو MySQL)**

1. تحديث dim\_user\_metrics من مصادر: orders, returns, events (sessions, add\_to\_cart, begin\_checkout).
2. **توليد عضوية** لكل شريحة حَسَب القواعد:
   * دورة الحياة: من last\_order\_ts.
   * VIP/Value: من revenue\_lifetime, aov\_90d, percentiles.
   * الجودة/المخاطر: returns\_rate\_90d, cod\_risk\_score, coupon\_usage\_rate\_180d.

**B) لحظي (Streaming / Near-real-time)**

* تشغيل Functions على أحداث:
  + add\_to\_cart → إدراج/تحديث CART\_ABANDONER\_24H (يدخل الآن مع ttl\_hours=24).
  + begin\_checkout → إدراج/تحديث CHECKOUT\_ABANDONER\_6H.
  + purchase → **إخراج** العميل فورًا من شرائح “مهجور” وتحديث دورة الحياة.

**C) قواعد الدخول/الخروج**

* عند تحقق شرط الدخول وكان السجل غير موجود → INSERT (active).
* عند زوال الشرط (شراء، انتهاء نافذة) → UPDATE status='inactive' وlast\_exit\_ts=NOW().
* عند إعادة التحقق ودخول جديد → تحديث last\_enter\_ts.

# **5) أمثلة SQL مختصرة**

**A) تحديد DORMANT\_31\_90D (دفعة يومية)**

-- افتراض: days\_since\_last\_order = DATEDIFF(CURDATE(), last\_order\_ts)

INSERT INTO segment\_membership (user\_id, segment\_id, status, first\_join\_ts, last\_enter\_ts, evidence)

SELECT m.user\_id, sc.segment\_id, 'active', NOW(), NOW(),

JSON\_OBJECT('days\_since\_last\_order', DATEDIFF(CURDATE(), m.last\_order\_ts))

FROM dim\_user\_metrics m

JOIN segment\_catalog sc ON sc.code = 'DORMANT\_31\_90D' AND sc.is\_active=1

WHERE m.last\_order\_ts IS NOT NULL

AND DATEDIFF(CURDATE(), m.last\_order\_ts) BETWEEN 31 AND 90

ON DUPLICATE KEY UPDATE

status='active',

last\_enter\_ts=VALUES(last\_enter\_ts),

evidence=VALUES(evidence),

updated\_at=CURRENT\_TIMESTAMP;

**B) إخراج من CART\_ABANDONER\_24H عند الشراء (لحظي)**

UPDATE segment\_membership sm

JOIN segment\_catalog sc ON sc.segment\_id = sm.segment\_id AND sc.code='CART\_ABANDONER\_24H'

SET sm.status='inactive', sm.last\_exit\_ts=NOW()

WHERE sm.user\_id = ? AND sm.status='active';

**C) VIP باستخدام Percentile (يُحسب في BigQuery أو MySQL مؤقتًا)**

* احسب P95 لإجمالي الإيراد، ثم:

INSERT INTO segment\_membership (...)

SELECT u.user\_id, sc.segment\_id, 'active', NOW(), NOW(),

JSON\_OBJECT('revenue\_lifetime', m.revenue\_lifetime)

FROM dim\_user\_metrics m

JOIN users u ON u.user\_id = m.user\_id

JOIN segment\_catalog sc ON sc.code='VIP' AND sc.is\_active=1

WHERE m.revenue\_lifetime >= @p95\_revenue

ON DUPLICATE KEY UPDATE status='active', last\_enter\_ts=NOW();

**D) تنظيف الشرائح منتهية الصلاحية (TTL)**

UPDATE segment\_membership sm

JOIN segment\_catalog sc ON sc.segment\_id=sm.segment\_id

SET sm.status='inactive', sm.last\_exit\_ts=NOW()

WHERE sm.status='active'

AND sc.ttl\_hours IS NOT NULL

AND sm.last\_enter\_ts < (NOW() - INTERVAL sc.ttl\_hours HOUR);

# **6) عرض “الشريحة الأساسية” (View اختياري)**

CREATE VIEW user\_primary\_segment AS

SELECT t.user\_id, t.segment\_id, t.code, t.name\_ar, t.group\_code

FROM (

SELECT sm.user\_id, sm.segment\_id, sc.code, sc.name\_ar, sc.group\_code,

ROW\_NUMBER() OVER (PARTITION BY sm.user\_id ORDER BY

(sc.group\_code='value') DESC, -- أعطِ VIP أولوية

sc.priority ASC, sm.last\_enter\_ts DESC) AS rn

FROM segment\_membership sm

JOIN segment\_catalog sc ON sc.segment\_id=sm.segment\_id

WHERE sm.status='active'

) t

WHERE t.rn=1;

# **7) ربط الشرائح بالاستخدامات (قواعد إرسال مبنية على الشرائح)**

* **CART\_ABANDONER\_24H**: يسمح برسالة تذكير واحدة خلال 24h (حد تكرار)، قناة وفق موافقات المستخدم.
* **CHECKOUT\_ABANDONER\_6H**: 2 رسائل (6h و24h) مع كود خصم اختياري (تحكم في “COUPON\_HUNTER”).
* **VIP**: وصول مبكر + حدّ تكرار أعلى، مع احترام consents.
* **DORMANT/CHURN**: برامج تنشيط، تفعيل WhatsApp فقط لذوي opt-in واضح.
* **COD\_RISK\_HIGH**: رسائل تحقق قبل الشحن أو منع الدفع عند الاستلام تلقائيًا.

**الموافقات** لا تغيّر العضوية؛ تُطبَّق فقط عند التنفيذ (can\_send).

# **8) مراقبة وجودة البيانات**

* لوحات: حجم كل شريحة، التحويلات لكل شريحة، تداخل الشرائح (overlap)، زمن البقاء.
* تنبيهات: ارتفاع مفاجئ في CART\_ABANDONER أو LOST.
* اختبارات:
  + شراء ↦ خروج فوري من Abandoners ودخول Lifecycle المناسب.
  + حساب VIP ثابت نسبيًا ولا يقفز يوميًا بدون سبب.
  + TTL يعمل لشرائح السلوك.

# **9) تنفيذ**

1. إنشاء الجداول الثلاثة وتعبئة segment\_catalog مع criteria\_json وpriority.
2. بناء Job يومي يُحدّث dim\_user\_metrics ثم يُطبّق قواعد الشرائح (INSERT/UPSERT/EXIT).
3. Functions لحظية لأحداث add\_to\_cart, begin\_checkout, purchase لتحديث شرائح السلوك فورًا.
4. بناء View للشريحة الأساسية (اختياري).
5. ربط محرّك الإرسال بشرائح الاستهداف + خدمة can\_send (الموافقات).
6. إضافة مراقبة وQA (الأحجام، معدلات التحويل، زمن البقاء، التداخل).

علامة التبويب 26

**جعل GA4/Firebase يعرّفان كل الأحداث باسم مستخدم واحد موحّد يساوي معرّف العميل في قاعدة البيانات (users.id)**.

* دعم الانتقال من ضيف → مسجّل (mapping)، وتوحيد القياس **عبر الأجهزة**.
* احترام الخصوصية: عدم استخدام أي PII (بريد/هاتف) كـ user\_id.

# **2) المعايير (Standards)**

* **صيغة user\_id في GA4**: نص قصير **غير حساس** مشتق من users.id (مثلاً: U123456 أو الـUUID كما هو).
  + ثابت/غير قابل لإعادة الاستخدام لعميل آخر.
  + لا تستخدم الهاتف/البريد مباشرة.
* **وقت التعيين**:
  + يعيَّن بعد **نجاح تسجيل الدخول/التسجيل** مباشرة.
  + يُزال عند **تسجيل الخروج**.
* **الأحداث قبل تسجيل الدخول** تُسجَّل على معرّف الجهاز (GA4 user\_pseudo\_id). بعد تسجيل الدخول، تستمر الجلسة مع user\_id.
* **الخصوصية**: في حال احتجت إظهار المعرّف خارج الأنظمة الداخلية (تقارير خارجية)، استخدم **تجزئة SHA-256** لمعرّف داخلي وليس لبريد/هاتف.

# **3) تعديلات قاعدة البيانات (اختيارية لكن موصى بها)**

* جدول devices:
  + ga\_app\_instance\_id VARCHAR(64) NULL — تخزين App Instance ID (لربط أحداث الخادم إن استخدمت Measurement Protocol).
* جدول users:
  + لا حاجة لحقل إضافي إن كنت ستستخدم users.id نفسه كـ user\_id في GA4، لكن يمكن إضافة ga\_user\_idVARCHAR(64) إن رغبت في طبقة تجريد.

سياسات: ga\_user\_id = دالة تحويل موحّدة: ga\_user\_id = 'U' || users.id.

# **4) تطبيق Flutter (Firebase Analytics)**

## **4.1 الحِزم**

* firebase\_core
* firebase\_analytics
* (اختياري) firebase\_auth لإدارة جلسة المستخدم

## **4.2 التهيئة الأساسية**

* فعّل Analytics في main().
* على أول تشغيل بعد الإذن/الموافقة (إن طبّقتها)، أبقِ التجميع مفعّلًا:  
  FirebaseAnalytics.instance.setAnalyticsCollectionEnabled(true);

## **4.3 تعيين user\_id عند تسجيل الدخول**

import 'package:firebase\_analytics/firebase\_analytics.dart';

Future<void> onLogin(String internalUserId) async {

// حوّل معرف DB إلى GA4 user\_id (اختياري إضافة بادئة)

final gaUserId = 'U$internalUserId';

// 1) عيّن user\_id على Analytics

await FirebaseAnalytics.instance.setUserId(id: gaUserId);

// 2) خصائص قياسية (اختياري لكنها مفيدة للتقارير)

await FirebaseAnalytics.instance.setUserProperty(name: 'app\_role', value: 'customer');

await FirebaseAnalytics.instance.setUserProperty(name: 'market', value: 'YE');

// 3) أرسل حدث login (اختياري)

await FirebaseAnalytics.instance.logLogin(loginMethod: 'password'); // أو 'otp', 'social'

}

## **4.4 إزالة user\_id عند تسجيل الخروج**

Future<void> onLogout() async {

// يزيل الربط بالمستخدم الحالي

await FirebaseAnalytics.instance.setUserId(id: null);

// مستحسن لمسح الوسوم المرتبطة بالمستخدم السابق

await FirebaseAnalytics.instance.resetAnalyticsData();

}

## **4.5 أفضل ممارسات إضافية**

* **لا تعيّن user\_id قبل التأكد من نجاح التوثيق**.
* لو سمحتِ بتبديل الحساب داخل التطبيق: نفّذ onLogout() ثم onLogin() للحساب الجديد.
* اربط هذا التدفق مع مواصفاتك السابقة: عند onLogin نفّذ أيضًا /v1/devices/associate.

# **5) الويب (إن وُجد WebView/موقع)**

* مع **gtag.js**:

<script>

// بعد نجاح تسجيل الدخول

gtag('config', 'G-XXXXXXX', {

'user\_id': 'U123456' // نفس صيغة التطبيق

});

// لتعيين خصائص مستخدم

gtag('set', 'user\_properties', {

app\_role: 'customer',

market: 'YE'

});

// عند تسجيل الخروج

// GA4 لا يوفّر "unset" مباشر عبر gtag، لكن بدّلي user\_id إلى undefined عبر إعادة config بدون user\_id

gtag('config', 'G-XXXXXXX'); // بدون user\_id

</script>

لو كان لديك **تدفق مشترك App + Web**، احرصي على استخدام نفس صيغة user\_id وموحِّد الدخول (SSO) لتحسين الربط عبر القنوات.

# **6) الخادم (Measurement Protocol v2) — اختياري متقدّم**

استخدمه فقط إن سترسلين **أحداث خادمية** (مثل تأكيد الدفع) وتريدين أن تُنسَب لنفس المستخدم والجلسة.

* لأحداث **تطبيقات الموبايل** أرسل app\_instance\_id (من التطبيق) و**نفس user\_id**:
  + خزّن ga\_app\_instance\_id في devices.
  + حمّلي قيمة app\_instance\_id لمرة واحدة من التطبيق (إن لزم) وارسليها لسيرفرك.
* لأحداث **الويب** استخدم client\_id (من \_ga) و**نفس user\_id**.

**هيكل JSON مبسّط:**

{

"api\_secret": "XXXX",

"measurement\_id": "G-XXXX",

"firebase\_app\_id": "1:123:web:abc", // لمسارات التطبيقات/الويب حسب نوع الحدث

"user\_id": "U123456",

"app\_instance\_id": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",

"events": [{

"name": "purchase\_server",

"params": {

"value": 25000,

"currency": "YER",

"transaction\_id": "ORD-9876"

}

}]

}

**مهم**: لا تُرسل نفس الحدث مرتين (عميل + خادم) بدون إزالة التكرار. إن احتجت الازدواج، أضف event\_id/timestamp\_micros موحّدين وخصّصي dedup في التحليلات.

# **7) BigQuery: التوافق وربط التقارير**

عند تفعيل **GA4 BigQuery Export**:

* ستجدي:
  + user\_id في المستوى العلوي للسجل.
  + user\_pseudo\_id (معرّف الجهاز/التثبيت).
* لربط تقارير الإيراد من DB:
  + اربطي user\_id في GA4 بـ users.id (أو ga\_user\_id) مباشرة.
  + احفظي order\_id نفسه كـ transaction\_id في حدث purchase (عميل/خادم)، لتسهيل الربط مع جدول orders.

**استعلام مثال (تلخيص مشتريات لكل user\_id):**

SELECT

user\_id,

COUNTIF(event\_name='purchase') AS purchases,

SUM(CAST(ep.value.int\_value AS INT64)) FILTER (WHERE ep.key='value') AS total\_value

FROM `project.dataset.events\_\*`,

UNNEST(event\_params) ep

WHERE user\_id IS NOT NULL

GROUP BY user\_id;

# **8) التكامل مع مواصفاتك السابقة**

* عند onLogin:
  + devices.associate (ربط الجهاز بالمستخدم)
  + analytics.setUserId
  + تحديث **user\_properties** (مثل first\_source/last\_source إن تحفظينها محليًا)
* عند purchase:
  + أرسل حدث purchase إلى Firebase Analytics **بنفس user\_id**.
  + خزِّن نفس user\_id داخل **orders** لتطابق 1:1 مع تقارير GA4.

# **9) اختبارات (QA)**

1. **تسجيل الدخول** → افتحي جلسة، نفّذي onLogin، أطلقي حدث purchase؛ في DebugView يجب أن يظهر user\_id=U....
2. **تسجيل الخروج** → setUserId(null) + resetAnalyticsData()؛ أحداث ما بعد الخروج لا تحتوي user\_id.
3. **تبديل الحساب** على نفس الجهاز → ينعكس user\_id الجديد على الأحداث التالية فقط.
4. **جهازان لنفس الحساب** → كلاهما يسجّل user\_id نفسه؛ تقارير GA4 توحّد المستخدم.
5. **خادم + عميل** (إن استخدمت MP): تأكدي أن حدث الخادم يحمل نفس user\_id و(للتطبيق) app\_instance\_id لتجنّب تشظّي المستخدم.

# **10) التنفيذ**

* اعتماد صيغة ga\_user\_id = تحويل ثابت من users.id (بدون PII).
* Flutter: setUserId عند الدخول، وsetUserId(null) + resetAnalyticsData() عند الخروج.
* إرسال purchase مع transaction\_id = order\_id.
* (اختياري) حفظ ga\_app\_instance\_id في devices إن ستُرسلين أحداث خادمة عبر MP.
* Web (إن وجد): gtag('config', ..., {user\_id}) بعد تسجيل الدخول.
* BigQuery: بناء لوحات تربط user\_id من GA4 بـ users.id وorders.
* QA: التحقق عبر DebugView + عينات BigQuery.

علامة التبويب 27

**قياس احتفاظ المستخدمين حسب دفعات اكتساب (Cohorts) على أساس تاريخ البدء.**

* دعم نوعين شائعين من “النشاط”:
  + **Activity Retention**: المستخدم قام بأي تفاعل معتبر.
  + **Purchase Retention**: المستخدم قام بعملية شراء.
* عرض مؤشرات: **W1/W4/W8** و **M1/M3/M6** + WAU/MAU.

# **2) التعاريف (Standards)**

## **2.1 مرساة الدُفعة (Cohort Anchor)**

* **تطبيق الموبايل (مستحسن)**: حدث first\_open من GA4 (أول فتح للتطبيق).
* **الويب** (إن وجد): حدث first\_visit.
* **تحليل الشراء**: يمكن تعريف دفعة بديلة على أساس first\_purchase\_ts (أول طلب).

احتفظ بالاثنين: **Cohort\_Acq** (first\_open/visit) و **Cohort\_Purch** (first\_purchase) حسب تقريرك.

## **2.2 تعريف “نشط”**

اختر واحدة (أو اعرض كلاهما بالتوازي):

* **نشاط عام (Activity)**: وجود أي من الأحداث: session\_start, user\_engagement, view\_item, add\_to\_cart, begin\_checkout, purchase.
* **نشاط شراء (Purchase)**: وجود حدث purchase فقط.

## **2.3 أنواع الاحتفاظ**

* **Classic Retention (Cohort-aligned):**أسبوع 0 = أسبوع الانضمام، أسبوع 1 = [اليوم 7..13] منذ الانضمام… (نوافذ ثابتة بالنسبة لتاريخ الانضمام).
* **Rolling Retention:**المستخدم يعتبر “محتفَظ به” في الأسبوع k إذا عاد **في أي وقت بعد اليوم 7k وحتى الآن**. (للتنشيط الكلي—استخدِمه عند الحاجة فقط).
* **Bracket Retention:**نوافذ تقويمية ثابتة (أسابيع ISO أو أشهر تقويمية)؛ مفيد لملاءمة التقارير الشهرية.

**الموصى به للتقارير التشغيلية:** Classic Retention.

## **2.4 توحيد المناطق الزمنية**

* GA4 يصدّر **UTC**؛ استخدم DATETIME(TIMESTAMP\_MICROS(event\_timestamp), "Asia/Aden") لضبط +03:00.
* أسبوع ISO يبدأ **الاثنين**؛ وثّق ذلك في المستند.

# **3) مخطط بيانات مساند (اختياري ولكنه مفيد)**

-- dim\_users: مرساة الدُفعة

CREATE TABLE dim\_users (

user\_key STRING PRIMARY KEY, -- COALESCE(user\_id, user\_pseudo\_id)

first\_open\_ts DATETIME, -- من GA4

first\_visit\_ts DATETIME, -- للويب إن وجد

first\_purchase\_ts DATETIME

);

-- dim\_calendar: أبعاد الوقت (يوم/أسبوع ISO/شهر)

-- إن لم تكن موجودة، أنشئها لتبسيط التقارير.

**user\_key**: في GA4/BQ نستخدم COALESCE(user\_id, user\_pseudo\_id) لضمان ربط حسابات مسجّلة مع الضيوف عند التعيين.

# **4) BigQuery (GA4 Export) — SQL أمثلة**

بدّلي project.dataset باسم مشروعك.  
ضعي مؤشرات منطقية في WHERE حسب قناتك (App/Web).  
**Activity list** قابلة للتعديل.

## **4.1 استخراج دفعة الاكتساب الأسبوعية (first\_open)**

DECLARE TZ STRING DEFAULT "Asia/Aden";

WITH firsts AS (

SELECT

COALESCE(user\_id, user\_pseudo\_id) AS user\_key,

MIN(DATETIME(TIMESTAMP\_MICROS(event\_timestamp), TZ)) AS first\_open\_dt

FROM `project.dataset.events\_\*`

WHERE event\_name = 'first\_open'

GROUP BY 1

),

cohorts AS (

SELECT

user\_key,

DATE\_TRUNC(DATE(first\_open\_dt), WEEK(MONDAY)) AS cohort\_week

FROM firsts

),

acts AS (

-- نشاط معتبر (Activity Retention)

SELECT

COALESCE(user\_id, user\_pseudo\_id) AS user\_key,

DATE(DATETIME(TIMESTAMP\_MICROS(event\_timestamp), TZ)) AS activity\_date

FROM `project.dataset.events\_\*`

WHERE event\_name IN ('session\_start','user\_engagement','view\_item','add\_to\_cart','begin\_checkout','purchase')

),

joined AS (

SELECT

c.cohort\_week,

a.user\_key,

a.activity\_date,

DATE\_DIFF(a.activity\_date, c.cohort\_week, DAY) AS days\_since\_cohort

FROM acts a

JOIN cohorts c USING (user\_key)

),

buckets AS (

-- حاويات الأسابيع الكلاسيكية: W0/W1/...

SELECT

cohort\_week,

user\_key,

CASE

WHEN days\_since\_cohort BETWEEN 0 AND 6 THEN 0

WHEN days\_since\_cohort BETWEEN 7 AND 13 THEN 1

WHEN days\_since\_cohort BETWEEN 14 AND 20 THEN 2

WHEN days\_since\_cohort BETWEEN 21 AND 27 THEN 3

WHEN days\_since\_cohort BETWEEN 28 AND 34 THEN 4

ELSE NULL

END AS w\_bucket

FROM joined

)

SELECT

cohort\_week,

COUNT(DISTINCT user\_key) AS cohort\_size,

SAFE\_DIVIDE(COUNT(DISTINCT IF(w\_bucket=0, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS w0\_ret,

SAFE\_DIVIDE(COUNT(DISTINCT IF(w\_bucket=1, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS w1\_ret,

SAFE\_DIVIDE(COUNT(DISTINCT IF(w\_bucket=2, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS w2\_ret,

SAFE\_DIVIDE(COUNT(DISTINCT IF(w\_bucket=3, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS w3\_ret,

SAFE\_DIVIDE(COUNT(DISTINCT IF(w\_bucket=4, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS w4\_ret

FROM buckets

GROUP BY cohort\_week

ORDER BY cohort\_week DESC;

## **4.2 الاحتفاظ الشهري (Classic) على أساس first\_open**

DECLARE TZ STRING DEFAULT "Asia/Aden";

WITH firsts AS (

SELECT

COALESCE(user\_id, user\_pseudo\_id) AS user\_key,

MIN(DATETIME(TIMESTAMP\_MICROS(event\_timestamp), TZ)) AS first\_open\_dt

FROM `project.dataset.events\_\*`

WHERE event\_name='first\_open'

GROUP BY 1

),

cohorts AS (

SELECT

user\_key,

DATE\_TRUNC(DATE(first\_open\_dt), MONTH) AS cohort\_month

FROM firsts

),

acts AS (

SELECT

COALESCE(user\_id, user\_pseudo\_id) AS user\_key,

DATE(DATETIME(TIMESTAMP\_MICROS(event\_timestamp), TZ)) AS activity\_date

FROM `project.dataset.events\_\*`

WHERE event\_name IN ('session\_start','user\_engagement','view\_item','add\_to\_cart','begin\_checkout','purchase')

),

joined AS (

SELECT

c.cohort\_month,

a.user\_key,

a.activity\_date,

DATE\_DIFF(a.activity\_date, c.cohort\_month, DAY) AS days\_since\_cohort

FROM acts a

JOIN cohorts c USING (user\_key)

),

mbuckets AS (

SELECT

cohort\_month,

user\_key,

CASE

WHEN days\_since\_cohort BETWEEN 0 AND 29 THEN 0

WHEN days\_since\_cohort BETWEEN 30 AND 59 THEN 1

WHEN days\_since\_cohort BETWEEN 60 AND 89 THEN 2

WHEN days\_since\_cohort BETWEEN 90 AND 119 THEN 3

WHEN days\_since\_cohort BETWEEN 120 AND 149 THEN 4

ELSE NULL

END AS m\_bucket

FROM joined

)

SELECT

cohort\_month,

COUNT(DISTINCT user\_key) AS cohort\_size,

SAFE\_DIVIDE(COUNT(DISTINCT IF(m\_bucket=0, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS m0\_ret,

SAFE\_DIVIDE(COUNT(DISTINCT IF(m\_bucket=1, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS m1\_ret,

SAFE\_DIVIDE(COUNT(DISTINCT IF(m\_bucket=2, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS m2\_ret,

SAFE\_DIVIDE(COUNT(DISTINCT IF(m\_bucket=3, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS m3\_ret,

SAFE\_DIVIDE(COUNT(DISTINCT IF(m\_bucket=4, user\_key, NULL)), COUNT(DISTINCT user\_key)) AS m4\_ret

FROM mbuckets

GROUP BY cohort\_month

ORDER BY cohort\_month DESC;

## **4.3 احتفاظ الشراء (Purchase Retention) — أسبوعي**

بدّلي “acts” ليقتصر على purchase فقط:

...

acts AS (

SELECT

COALESCE(user\_id, user\_pseudo\_id) AS user\_key,

DATE(DATETIME(TIMESTAMP\_MICROS(event\_timestamp), TZ)) AS activity\_date

FROM `project.dataset.events\_\*`

WHERE event\_name = 'purchase'

),

...

## **4.4 مصفوفة Heatmap (Pivot) للاحتفاظ الأسبوعي**

WITH ret AS ( -- استخدمي استعلام 4.1 حتى حقل w\_bucket

...

)

SELECT

cohort\_week,

COUNT(DISTINCT user\_key) AS cohort\_size,

ROUND(100\*COUNT(DISTINCT IF(w\_bucket=0, user\_key, NULL))/COUNT(DISTINCT user\_key),1) AS W0,

ROUND(100\*COUNT(DISTINCT IF(w\_bucket=1, user\_key, NULL))/COUNT(DISTINCT user\_key),1) AS W1,

ROUND(100\*COUNT(DISTINCT IF(w\_bucket=2, user\_key, NULL))/COUNT(DISTINCT user\_key),1) AS W2,

ROUND(100\*COUNT(DISTINCT IF(w\_bucket=3, user\_key, NULL))/COUNT(DISTINCT user\_key),1) AS W3,

ROUND(100\*COUNT(DISTINCT IF(w\_bucket=4, user\_key, NULL))/COUNT(DISTINCT user\_key),1) AS W4

FROM ret

GROUP BY cohort\_week

ORDER BY cohort\_week DESC;

## **4.5 WAU/MAU (نسبة المرونة)**

DECLARE TZ STRING DEFAULT "Asia/Aden";

DECLARE today DATE DEFAULT CURRENT\_DATE(TZ);

WITH acts AS (

SELECT DISTINCT

COALESCE(user\_id, user\_pseudo\_id) AS user\_key,

DATE(DATETIME(TIMESTAMP\_MICROS(event\_timestamp), TZ)) AS d

FROM `project.dataset.events\_\*`

WHERE event\_name IN ('session\_start','user\_engagement','view\_item','add\_to\_cart','begin\_checkout','purchase')

AND DATE(DATETIME(TIMESTAMP\_MICROS(event\_timestamp), TZ)) BETWEEN DATE\_SUB(today, INTERVAL 30 DAY) AND today

)

SELECT

COUNT(DISTINCT IF(d BETWEEN DATE\_SUB(today, INTERVAL 7 DAY) AND today, user\_key, NULL)) AS WAU,

COUNT(DISTINCT user\_key) AS MAU,

SAFE\_DIVIDE(

COUNT(DISTINCT IF(d BETWEEN DATE\_SUB(today, INTERVAL 7 DAY) AND today, user\_key, NULL)),

COUNT(DISTINCT user\_key)

) AS WAU\_MAU\_ratio

FROM acts;

# **5) تقارير DB داخلية (بدون GA4) — فكرة مختصرة**

لو كان لديك جدول أحداث داخلي events\_app:

* أعمدة: user\_id, event\_date, event\_name
* اتبع نفس منطق التجميع أعلاه مع DATE\_DIFF على event\_date.

# **6) لوحات ومؤشرات (KPIs)**

* **Weekly Retention (Activity)**: W1, W4 (أساسيان).
* **Monthly Retention (Activity)**: M1, M3, M6.
* **Purchase Retention**: نسبة من cohort تعود وتشتري في W4 / M3.
* **WAU/MAU**: يقيس “لزاجة” التطبيق (Stickiness).
* **Median Time-to-2nd Purchase**: وسيط الزمن للشراء الثاني (مهم للتنشيط).

# **7) معايير جودة البيانات (QA)**

* الاحتفاظ **غير متزايد** منطقيًا عبر الأسابيع: W2 ≤ W1 … (راقبيه).
* تراجع ملحوظ في “Direct” بعد تحسين UTM (من تقارير القنوات) ↦ يعني القياس جيد.
* تأكدي من المنطقة الزمنية “Asia/Aden” في كل التقارير.
* الأحداث المكررة (عن طريق العميل والخادم) يجب **منع ازدواجها** أو تمييزها بـ event\_id.

# **8) ربط بالشرائح والموافقات**

* يمكنك تفصيل الاحتفاظ لكل **شريحة** (VIP, DORMANT…) بضم user\_id مع segment\_membership (status='active' لحظة الانضمام أو تراكميًا).
* عند إنشاء حملات “تنشيط”، استخدمي **can\_send** لاحترام الموافقات قبل الإرسال.

# **9) التنفيذ**

1. اختيار **Anchor**: first\_open (App) / first\_visit (Web) + بديل first\_purchase.
2. تعريف “نشط” (Activity vs Purchase) وتثبيته في المستند.
3. تفعيل **GA4 BigQuery Export** (إن لم يُفعّل).
4. بناء استعلامات 4.1–4.5 (كوّني Views أو Scheduled Queries).
5. ضبط **المنطقة الزمنية** إلى "Asia/Aden".
6. إنشاء لوحات Heatmap + بطاقات W1/W4/M1/M3/WAU/MAU.
7. QA: تحقق من خصائص التناقص، والهوية الموحّدة user\_id.

الكتالوج (Catalog)

# **الكتالوج (Catalog)**

## **1) الهدف**

* تنظيم المنتجات ونسخها (مقاسات/ألوان) ومحتواها لتجربة تسوّق سلسة.
* دعم البحث والتصفية والفرز بكفاءة.
* ربط الكتالوج بالمخزون والتسعير والعروض دون تعقيد.
* رفع معدّل التحويل عبر محتوى بصري ونصي مكتمل.

## **2) نطاق الكتالوج (Scope)**

* العلامات التجارية، المنتجات الأساسية، النسخ (Variants)، التصنيفات متعددة المستويات، الخصائص (Attributes)، الوسائط (صور/فيديو)، الأسعار والعروض، المجموعات (Collections)، أدلة المقاسات والعناية، شارات العرض، SEO، الربط بالمنتجات ذات الصلة.
* التكاملات: المخزون على مستوى النسخة، العروض/الخصومات، الواجهة الإدارية (لوحة التحكم)، البحث والفلاتر.

## **3) الكيانات الأساسية والحقول**

أدناه قائمة الكيانات والحقول المقترحة ووصف مختصر لكل حقل. (لا توجد أكواد؛ التنفيذ وتقنيات التخزين من صلاحية الفريق).

### **3.1 العلامة التجارية (Brand)**

* **brand\_id**: معرف داخلي.
* **name**: اسم العلامة.
* **country\_of\_origin**: بلد المنشأ (اختياري).
* **display\_order**: ترتيب العرض.
* **status**: نشط/مؤرشف.

### **3.2 المنتج الأساسي (Product)**

* **product\_id**: معرف داخلي.
* **name\_ar / name\_en**: اسم عربي (إلزامي) + إنجليزي (اختياري).
* **short\_description / long\_description**: وصف مختصر وموسع.
* **brand\_id**: مرجع العلامة.
* **product\_type**: فستان/أحذية/حقائب/لانجري/بيجامات… (قائمة مضبوطة).
* **collection\_code / season**: رمز التشكيلة/الموسم (اختياري).
* **country\_of\_origin**: بلد المنشأ (اختياري).
* **badges**: شارات عرض (جديد/الأكثر مبيعًا/قريب النفاد) — مع ملاحظة إن كانت تلقائية أم يدوية.
* **publish\_status**: مسودة/منشور/مؤرشف.
* **slug**: رابط ثابت فريد (أساسي في العربية).
* **seo\_title / seo\_description**: تحسين محركات البحث (اختياري).
* **content\_score**: درجة اكتمال المحتوى (0–100) للحكم على الجاهزية.
* **created\_at / updated\_at**: إدارة زمنية.

### **3.3 نسخ المنتج (Product Variant)**

* **variant\_id**: معرف داخلي.
* **product\_id**: مرجع المنتج.
* **sku**: رقم صنف فريد (مطلوب فريد عالميًا).
* **barcode**: باركود/QR (اختياري).
* **color**: اللون (من قاموس ألوان قياسي).
* **size**: المقاس (S/M/L/XL أو قياسات رقمية حسب الفئة).
* **material / fabric**: الخامة/القماش (اختياري).
* **weight / dimensions**: الوزن/الأبعاد (لأغراض شحن اختيارية).
* **regular\_price**: السعر الأساسي.
* **sale\_price**: سعر العرض (إن وجد).
* **currency**: العملة (YER/SAR…).
* **price\_valid\_from / price\_valid\_to**: جدول زمني للعرض (اختياري).
* **availability\_status**: متوفر/غير متوفر/طلب مسبق.
* **is\_default**: النسخة الافتراضية لعرضها أولًا.
* **created\_at / updated\_at**.

ملاحظة: مستوى النسخة هو مصدر الحقيقة للسعر والتوفر، ويرتبط بالمخزون.

### **3.4 التصنيفات (Category)**

* **category\_id**: معرف داخلي.
* **parent\_id**: تصنيف أب (لدعم مستويات متعددة).
* **name\_ar / name\_en**: اسم عربي + إنجليزي.
* **slug**: رابط ثابت فريد للتصنيف.
* **display\_order**: ترتيب العرض.
* **is\_visible**: الإظهار في الواجهة/الإخفاء.

### **3.5 ربط المنتج بالتصنيفات (Product–Category Link)**

* **product\_id**, **category\_id**: يسمح بوجود المنتج في أكثر من تصنيف.

### **3.6 المجموعات (Collections)**

* **collection\_id**: معرف.
* **name**: اسم المجموعة (مثل: “وصل حديثًا”، “فساتين سهرة”).
* **type**: يدوي/آلي.
* **rule\_definition**: قواعد المجموعة إن كانت آلية (مثال: فئة = فساتين && سعر ≥ X && شارة = جديد).
* **display\_order / is\_visible**.
* **products\_link**: ربط المنتجات بالمجموعة عند النوع اليدوي.

### **3.7 الخصائص (Attributes) والقيم**

* **attribute**: تعريف الخاصية (مثال: الطول، القصّة، الياقة، نوع الأكمام، النقشة، المناسبة، القماش…).
* **attribute\_values**: القيم الممكنة لكل خاصية (من قوائم مضبوطة).
* **product\_attributes**: خصائص على مستوى المنتج (تؤثر على الفلاتر العامة).
* **variant\_attributes**: خصائص على مستوى النسخة (عادة اللون/المقاس/الخامة).

الهدف: مرونة بدون فوضى؛ خصائص أساسية للأزياء جاهزة منذ البداية.

### **3.8 الوسائط (Media)**

* **media\_id**: معرف.
* **product\_id / variant\_id**: الربط (الأغلب على مستوى المنتج، ويمكن على مستوى النسخة عند اختلاف الألوان).
* **type**: صورة/فيديو.
* **url**: رابط الوسيط (يفضّل عبر CDN).
* **alt\_text**: نص بديل وصفي.
* **sort\_order**: ترتيب العرض (الأولى غلاف).
* **is\_primary**: تأكيد الصورة الأساسية عند الحاجة.
* **aspect\_ratio / width / height**: توصيف اختياري لضبط الجودة.

### **3.9 أدلة المقاسات والعناية (Size Guide / Care)**

* **size\_guide\_id**: مرجع جدول مقاسات لكل فئة/علامة.
* **care\_instructions**: تعليمات العناية (رموز أو نصوص مختصرة).
* ربط بالمنتج/الفئة حسب الحاجة.

### **3.10 العلاقات ذات الصلة (Related / Cross‑sell / Upsell)**

* **related\_products**: منتجات مرتبطة يدويًا.
* **cross\_sell / upsell**: اقتراحات تكميليّة (حقيبة/حذاء مع فستان).

### **3.11 شارات العرض (Badges / Flags)**

* **is\_new**: جديد (تلقائي خلال X أيام من النشر أو يدوي).
* **is\_best\_seller**: الأكثر مبيعًا (حسب مبيعات تاريخية/يدوي).
* **is\_low\_stock**: قريب على النفاد (وفق حد مخزون).
* **is\_on\_sale**: ضمن عرض (يتحدد من السعر/الجدولة).

### **3.12 SEO والتعريب**

* **slug فريد** لكل منتج وتصنيف (العربية أساس).
* **meta\_title / meta\_description** للمنتج والتصنيف (اختياري).
* دعم **name/description** بلغتين (على الأقل AR أساسي + EN اختياري).

## **4) قواعد العمل (Business Rules)**

* لا يُنشَر منتج حتى تتوفر: اسم عربي، تصنيف واحد على الأقل، صورة أساسية، نسخة واحدة على الأقل بسعر فعّال.
* يمنع تكرار (اللون + المقاس) داخل نفس المنتج.
* السعر والتوفر يُحدّدان على مستوى النسخة (Variant).
* أول صورة هي الغلاف؛ جودة وأبعاد موحّدة (نسبة 4:5 أو 3:4) وعدد صور مثالي 5–8.
* شارات العرض: إما تلقائية بالقواعد أو قابلة للضبط يدويًا مع أولوية واضحة (تلقائي → يدوي أو العكس).
* “أبلغني عند التوفر”: تُفعّل للنسخ غير المتاحة وتُخزن طلبات الإشعار (خارج نطاق هذا المستند).
* المنتجات المؤرشفة لا تظهر في البحث ولا في التصنيفات/المجموعات العامة.

## **5) البحث والتصفية (Search & Facets)**

* فلاتر أساسية: المقاس، اللون، السعر (مدى)، التصنيف، العلامة، الخامة/القماش، الطول، القصّة، الأكمام، الياقة، النقشة، المناسبة، بلد المنشأ، التوفر، الخصومات، “وصل حديثًا”.
* مراتب الفرز: الأحدث، السعر تصاعديًا/تنازليًا، الأكثر مبيعًا، الأكثر مشاهدة/إضافة للسلة.
* دعم **اقتراحات البحث** و**تصحيح الأخطاء الإملائية** لاحقًا (اختياري).

## **6) إدارة المحتوى والتصوير (Content Standards)**

* **الصور**: خلفية نظيفة، زووم على القماش/التفاصيل، إضاءة ثابتة، أبعاد موحّدة، ضغط مناسب.
* **النصوص**: اسم واضح + نقاط مختصرة (القماش/القصّة/الطول/المناسبة/العناية).
* **الألوان**: أسماء متسقة + عيّنات لونية (swatches) مطابقة قدر الإمكان.
* **الترجمة**: العربية أساس؛ الإنجليزية لتوسيع الأسواق لاحقًا.

## **7) الواجهة الإدارية (لوحة التحكم)**

* إنشاء/تعديل منتج مع عدّة نسخ بسرعة.
* رفع وترتيب الصور بالسحب والإفلات.
* استيراد/تحديث جماعي (CSV/Excel) للمنتجات والأسعار والخصائص.
* تعديلات جماعية: نشر/إلغاء نشر، تعيين تصنيف، تغيير أسعار، نقل لمجموعة.
* إدارة مجموعات الصفحة الرئيسية: يدويًا أو عبر قواعد آلية.
* لوحة **درجة اكتمال المحتوى** (اسم/صور/وصف/خصائص/سعر/تصنيف) لكل منتج.

## **8) الجودة والجاهزية (Catalog Health)**

* درجة اكتمال المحتوى لكل منتج (0–100) مع معايير واضحة لحسابها.
* نسبة المنتجات المكتملة > 90% كهدف.
* تغطية المقاسات الأساسية لكل منتج (حسب الفئة).
* نسبة النسخ غير المتوفرة مستهدفة < 20%.
* متابعة معدل الإرجاع لكل SKU لتحسين الوصف والمقاسات.

## **9) العلاقات مع الأنظمة الأخرى**

* **المخزون**: النسخة ترتبط بمخزون الموقع/المستودع. حالة التوفر تُستمد من المخزون.
* **العروض/الخصومات**: تُعرض تلقائيًا عند اقتران النسخة بعرض ساري.
* **التوصيات**: يمكن ربط واجهة التوصية (AI/قواعد) بقوائم cross‑sell/upsell.

## **10) ضوابط البيانات والخصوصية**

* توحيد قوائم القيم (الألوان/المقاسات/الأقمشة) لتجنّب الفوضى النصية.
* منع تكرار SKU.
* استخدام حقول زمنية للإنشاء/التحديث؛ ودعم الأرشفة الناعمة (Soft Archive) للمنتجات.
* التحكم في الوصول حسب الدور: المحتوى/التسعير/النشر.

## **11) مؤشرات الأداء المطلوبة**

* زمن تحميل قوائم المنتجات وتفاصيل المنتج (للمتابعة عبر Firebase Performance).
* نسبة النقر إلى تفاصيل المنتج من القوائم.
* تحويلات إضافة للسلة والشراء لكل فئة/تصنيف.
* أداء البحث (لا توجد نتائج/استعلامات شائعة/استعلامات تؤدي لمبيعات).

## **12) معايير القبول (Definition of Done)**

* يمكن إنشاء منتج كامل وربط نسخ وصور وتصنيفات ونشره.
* البحث والفلاتر يعملان بدقة مع فرز موثوق.
* تظهر خصائص اللون/المقاس بوضوح في تفاصيل المنتج ويُمكن الاختيار بينها بسهولة.
* الواجهة الإدارية تدعم الاستيراد والتعديلات الجماعية والمجموعات.
* شارات “جديد/خصم/الأكثر مبيعًا/قريب النفاد” تُعرض وفق القواعد.
* لوحة صحة الكتالوج تعرض درجة اكتمال المحتوى ونقاط التحسين.

## **13) خارطة الإصدارات (Release Plan)**

* **MVP**: العلامات، المنتجات، النسخ، التصنيفات، الصور، الأسعار، فلاتر أساسية، مجموعتان (وصل حديثًا/الأكثر مبيعًا)، معايير نشر.
* **المرحلة 2**: خصائص موسّعة للأزياء، مجموعات آلية بالقواعد، أدلة مقاسات ديناميكية، فيديو قصير، “أبلغني عند التوفر”.
* **المرحلة 3**: تعدد قوائم الأسعار حسب السوق/العملة، توصيات ذكية، تحسين البحث الدلالي.

### **ملاحظات ختامية**

* هذه المتطلبات تصف “ما يلزم” وظيفيًا للكتالوج بدون تقييد التنفيذ بتقنية محددة.
* يعتمد تحديد أنواع الحقول/الفهارس والقيود التفصيلية على قاعدة البيانات المختارة، مع الحفاظ على القواعد الوظيفية المذكورة أعلاه.
* يُفضّل توثيق قاموس للخصائص والقيم المسموح بها (Attributes Dictionary) قبل الاستيراد الأول للبيانات لتفادي التضارب لاحقًا.

**كتالوج المنتجات** المصمَّمة:

* **تنظيم محكم للبيانات:** المنتج الأساسي → نسخ (لون/مقاس) → تصنيفات وخصائص واضحة. النتيجة: إدخال وتحديث أسهل، ومنع تكرار الأكواد (SKU).
* **تجربة تسوّق أسرع ودقيقة:** بحث وفلاتر ذكية (مقاس/لون/سعر/مناسبة…) توصّل العميلة لما تريد بسرعة، فترتفع إضافة السلة والتحويل.
* **سعر وتوفر موثوقان:** التحديد على مستوى النسخة يضمن أن ما يظهر “متوفر” فعليًا، مع دعم “أبلغني عند التوفر” وتقليل الإحباط.
* **تحكّم تسويقي مرن:** مجموعات (Collections) وشارات تلقائية/يدوية (جديد/خصم/الأكثر مبيعًا) لعرض أقسام جذابة وتحديث الواجهة بسرعة.
* **تحسين جودة المحتوى:** معيار “درجة اكتمال المحتوى” وصحة الكتالوج يوجّهان الفريق لإكمال الصور/الوصف/الخصائص، ما يرفع التحويل ويقلّل المرتجعات.
* **SEO وصور منظّمة:** Slug وعناوين وأوصاف محسّنة + صور مرتبة، ما يزيد الظهور العضوي وجودة الهبوط من الإعلانات.
* **كفاءة تشغيلية:** استيراد/تعديلات جماعية وصلاحيات حسب الدور—وقت أقل لإدارة الكتالوج وأخطاء أقل.
* **تكامل سلس مع بقية النظام:** ارتباط مباشر بالمخزون والعروض والشحن، فتكون الأسعار والتقارير متسقة ودقيقة.
* **رفع متوسط قيمة الطلب:** علاقات “منتجات مرتبطة/تكميليّة” (cross-sell/upsell) تزيد فرص إضافة قطع مكمِّلة.
* **قابلية توسّع بلا فوضى:** قواميس ألوان/مقاسات/خصائص موحّدة، مع قابلية لاحقة لأسعار متعددة للأسواق.
* **قياس واتخاذ قرار أفضل:** مؤشرات مثل الأكثر مشاهدة، الأكثر إضافة للسلة، وقيمة السلال حسب الفئة—تساعد في التوريد، التسعير، وترتيب الواجهة.

**توقعات واقعية (تختلف حسب التنفيذ):**تحسّن معدل التحويل على صفحة المنتج **5–15%** مع اكتمال المحتوى، انخفاض المرتجعات **5–10%** بفضل أدلة المقاسات والوصف الأدق، وزيادة **3–8%** في متوسط قيمة الطلب عبر العروض التكاملية.

**الخلاصة:** هذا الكتالوج يمنحك قاعدة بيانات نظيفة وقابلة للتوسّع، وتجربة تسوّق أوضح وأسرع، وتحكّم تسويقي وتشغيلي يترجم مباشرة إلى مبيعات أعلى وتكاليف أقل.

المنتجات (Product)

# **المنتجات (Product)**

## **1) الهدف**

تجميع كل ما يخص المنتج الأساسي في سجل واحد: الاسم، الوصف، العلامة، التصنيفات، الوسائط، الخصائص العامة، وشارات العرض—مع فصل الأسعار والتوفر على مستوى **النسخ (Variant)**.

## **2) ما الذي يمثّله “المنتج”**

* “المنتج” = هوية العنصر في المتجر (فستان/حقيبة/حذاء…).
* تحته تأتي **النسخ** (لون/مقاس/خامة)، وهي مصدر **السعر** و**التوفر**.
* المنتج يرتبط بتصنيفات متعددة، وخصائص عامة للفلاتر، وصور/فيديو، ومجموعات العرض.

## **3) أهم الحقول**

* **التعريفات:** اسم عربي (أساسي) + إنجليزي (اختياري)، وصف مختصر وموسع، نوع/فئة سلعية (فستان/لانجري/أحذية…)، العلامة التجارية، موسم/تشكيلة (اختياري).
* **العرض:** شارات (جديد/خصم/الأكثر مبيعًا/قريب النفاد)، ترتيب العرض، حالة النشر (مسودة/منشور/مؤرشف).
* **التصنيفات:** ارتباط بواحد أو أكثر من التصنيفات المتعددة المستويات.
* **الخصائص العامة:** (طول/قصّة/ياقة/أكمام/نقشة/مناسبة/قماش… من قوائم مضبوطة).
* **الوسائط:** صورة أساسية + صور إضافية مرتّبة، (فيديو اختياري)، نص بديل.
* **SEO والتعريب:** رابط ثابت فريد (Slug)، عنوان/وصف SEO (اختياري).
* **جودة المحتوى:** درجة اكتمال المحتوى (0–100) تُحسب من توفر الاسم/الصورة/الوصف/الخصائص/التصنيف.

ملاحظة: **السعر والتوفر** ليسا على مستوى المنتج بل على **مستوى النسخة**.

## **4) العلاقات**

* منتج ↔ نسخ: واحد إلى متعدد.
* منتج ↔ تصنيفات: متعدد إلى متعدد.
* منتج ↔ مجموعات (Collections): يدوي أو قواعد آلية.
* منتج ↔ وسائط: واحد إلى متعدد (مع إمكانية ربط صور بألوان النسخ).
* منتج ↔ منتجات مرتبطة/تكميليّة (Related/Cross-sell/Upsell).

## **5) قواعد العمل**

* لا يُنشر المنتج إلا عند اكتمال: **اسم عربي + تصنيف + صورة أساسية + نسخة واحدة فعّالة بالسعر**.
* داخل المنتج الواحد: لا تتكرر **(لون + مقاس)** بين النسخ.
* الشارة **“خصم”** تظهر إذا كان للنسخة سعر عرض ساري؛ **“جديد”** خلال X أيام من النشر (أو تُحدَّد يدويًا).
* المنتج المؤرشف لا يظهر في البحث والتصنيفات.
* اللون يُعرض بـ **عينة لونية (swatch)** ويبدّل الصور عند التغيير.

## **6) معايير المحتوى**

* **الاسم:** واضح ويشير للفئة والتميّز (مثال: “فستان سهرة ساتان بقصّة ملفوفة”).
* **الوصف:** نقاط مختصرة: القماش، القصّة، الطول، المناسبة، تعليمات العناية.
* **الصور:** 5–8 صور بجودة موحّدة ونسبة 4:5 أو 3:4 + لقطة قماش قريبة.
* **المقاسات والعناية:** ربط بدليل مقاسات مناسب للفئة وتعليمات العناية.
* **التعريب:** العربية أساس؛ الإنجليزية اختيارية.

## **7) إدارة المنتج (لوحة التحكم)**

* إنشاء/تعديل منتج بسرعة مع إدارة النسخ والوسائط والسعر والعروض.
* استيراد/تعديل جماعي (CSV/Excel) للمنتجات والخصائص والترتيب.
* إدارة المجموعات (وصل حديثًا/الأكثر مبيعًا…) يدويًا أو عبر قواعد.
* لوحة **“اكتمال المحتوى”** لتنبيه الفريق بما ينقص كل منتج.

## **8) البحث والتصفية (على مستوى المنتج)**

* فلاتر: التصنيف، العلامة، الخصائص العامة (طول/قصّة/…)، “وصل حديثًا”، “ضمن خصم”.
* الفرز: الأحدث، السعر تصاعد/تنازل، الأكثر مبيعًا، الأكثر مشاهدة/إضافة للسلة.
* دعم اقتراحات البحث وتصحيح الإملاء لاحقًا.

## **9) قياس الأداء للقرار التجاري**

* مشاهدات المنتج → نسبة الدخول إلى التفاصيل (CTR من القوائم).
* **إضافة للسلة** ونسبة التحويل من صفحة المنتج.
* أكثر الألوان/المقاسات طلبًا.
* معدل المرتجعات لكل منتج/نسخة (لتحسين الوصف والمقاسات).
* نقاط اكتمال المحتوى مقابل التحويل (لاستهداف التحسين).

## **10) تعريف الاكتمال (Definition of Done)**

* يمكن نشر المنتج ويظهر صحيحًا بالتصنيف والصور والخصائص.
* النسخ (السعر/التوفر) تعمل وتظهر خيارات اللون/المقاس بوضوح.
* الشارات تظهر وفق القواعد، والمجموعات تُدار بسهولة.
* البحث/الفلاتر تُعيد نتائج دقيقة، والفرز موثوق.
* لوحة “اكتمال المحتوى” تعمل وتُظهر ما ينقص.

**الخلاصة:** “المنتج” يقدّم الهيكل والمحتوى والعلاقات، بينما “النسخة” تتكفّل بالسعر والتوفر. هذا الفصل يبقي البيانات نظيفة، ويجعل التسعير والمخزون والبحث والفلاتر تعمل بدقة—ويترجم ذلك إلى تجربة شراء أسرع ومبيعات أعلى.

الطلبات (Orders)

**الطلبات (Orders)**

## **1) الهدف**

* تسجيل الطلب بشكل دقيق من لحظة الإنشاء حتى التسليم/الإغلاق.
* دعم **COD** بقوة (سوق اليمن) مع تقليل الرفض والتحصيل المتأخر.
* ربط تلقائي بالمخزون، الشحن، المدفوعات، والمرتجعات.
* توفير بيانات واضحة التقارير والتحليلات والتسويق.

## **2) ما الذي نحتاج بناؤه (الكيانات الأساسية)**

* **الطلب (Order)**: رأس الطلب، الأرقام، العميل، العناوين، الإجماليات، الحالة.
* **بنود الطلب (Order Items)**: المنتجات/النسخ والكميات والأسعار والخصومات.
* **تاريخ الحالة (Order Status History)**: سجل انتقالات الحالة مع الوقت والسبب.
* **الدفع (Payments)**: مزوّد، حالة الدفع (النقدي/أونلاين)، مبالغ محصّلة/مستردّة.
* **الشحن (Shipments)**: شركة الشحن، رقم التتبع، حالات الشحن ومحاولات التسليم.
* **العناوين الملتقطة (Address Snapshot)**: نسخة ثابتة لعنوان الشحن داخل الطلب.
* **الخصومات والقسائم**: ربط بالرموز/العروض المستخدمة في الطلب.
* **الارتباطات**: المرتجعات/الاستبدالات لاحقًا + مرجع الحملات التسويقية (UTM).

## **3) الحقول الرئيسية في الطلب (مستوى الرأس)**

* **المعرفات**: رقم طلب مقروء (تسلسلي قصير)، معرف العميل (إن وجد)، قناة الإنشاء (التطبيق/إنستقرام/واتساب).
* **الزمن**: تاريخ الإنشاء، التحديث، تواريخ المعالم (التأكيد، الشحن، التسليم).
* **الحالة**:
  + حالة الطلب: pending → confirmed → packed → shipped → out\_for\_delivery → delivered
  + حالات جانبية: cancelled، returned، failed\_delivery
* **الدفع**: طريقة الدفع (COD/أونلاين)، حالة الدفع (غير مدفوع/مفوّض/محصّل/مسترد جزئيًا/كاملًا).
* **الإجماليات**: المجموع قبل الخصم، إجمالي الخصم، رسوم الشحن، رسوم COD (إن وجدت)، الضرائب (إن طبقت)، الإجمالي النهائي، العملة.
* **العناوين**: اسم المستلم، هاتف، مدينة/حي/علامة قريبة (دقة محلية)، ملاحظات التوصيل.
* **التسويق**: promo code المستخدم، UTM (المصدر/الوسيط/الحملة) إن توفر.
* **ملاحظات**: ملاحظة العميل، ملاحظة فريق العمليات.

## **4) بنود الطلب (مستوى السطر)**

* المنتج/النسخة (لون/مقاس) + **السعر الملتقط لحظة الطلب** (حتى لو تغيّر لاحقًا).
* الكمية، خصم السطر (إن وجد)، المجموع الصافي للسطر.
* سياسات: لا تُحجز كمية تتجاوز المخزون المتاح (أو يسمح بالطلب المسبق حسب الإعداد).

## **5) حالات الطلب والانتقالات (الماكينة المنطقية)**

* **pending**: تم الإنشاء، بانتظار تأكيد (ودفع إن كان أونلاين).
* **confirmed**: تأكيد داخلي/آلي، يتم حجز المخزون.
* **packed**: جُهّزت الشحنة.
* **shipped**: سُلّمت لشركة الشحن ورقم تتبع موجود.
* **out\_for\_delivery**: خرجت للتسليم.
* **delivered**: تسليم ناجح (ولـ COD: تحصيل المبلغ).
* **failed\_delivery**: فشل/عدم رد، تُرفع محاولات التسليم ويُعاد الجدولة أو يُلغى.
* **cancelled**: قبل الشحن (تحرير الحجز/الكمية) أو بعده وفق سياسة الاسترجاع.
* **returned**: معالجة مرتجع مرتبطة بنفس الطلب.

يُسجَّل كل انتقال مع: **من/إلى**، الوقت، الفاعل (نظام/موظف/API)، السبب.

## **6) الدفع: اونلاين وCOD**

* **أونلاين**: تفويض/تحصيل، فشل، رد جزئي/كامل.
* **COD**:
  + أعلام تشغيلية: مسموح COD؟ درجة مخاطر العميل؟
  + عند **delivered**: تسجيل المبلغ المحصَّل، ثم تسوية مع شركة الشحن (remittance) لاحقًا.
  + في **failed\_delivery**: زيادة عدّاد الفشل للعميل/العنوان لاتخاذ إجراءات مستقبلية.

## **7) المخزون والحجز**

* عند **confirm**: حجز الكمية (reservation).
* عند **cancelled** قبل الشحن: تحرير الحجز.
* عند **delivered**: خصم نهائي من المخزون.
* مهلة حجز (TTL) لعربات/طلبات غير مكتملة لتفادي حبس المخزون بلا داعٍ.

## **8) الشحن والتتبع**

* ربط الطلب بواحدة أو أكثر من الشحنات (يدعم **التجزئة** عند الحاجة).
* حقول الشحن: شركة الشحن، رقم التتبع، حالة الشحن، محاولات التسليم، أوقات الشحن/التسليم.
* موافقة على **SLA** داخلية:
  + من **confirmed → packed** خلال X ساعات.
  + من **packed → shipped** خلال X ساعات.
  + من **shipped → delivered** خلال Y أيام (حسب المدينة).

## **9) الإلغاء، الاسترجاع، الاستبدال**

* **الإلغاء**: قبل الشحن مجانًا غالبًا؛ بعد الشحن وفق السياسة.
* **الاسترجاع**: نافذة زمنية (مثلاً 7–14 يومًا)، سبب (مقاس/عيب/تأخر)، حالة العنصر المرتجع، **طريقة التعويض** (استرداد/رصيد متجر/استبدال).
* **الاستبدال**: إنشاء طلب جديد مرتبط بالمرتجع الأصلي.

## **10) قواعد التسعير والخصومات**

* الخصم قد يكون على السطر أو على الطلب كاملًا (promo code).
* تُحفظ **القيم الملتقطة** لحظة إنشاء الطلب (سعر/خصم/شحن) لضمان الاتساق.
* سياسة التقريب والكسور واضحة على مستوى العملة.
* رسوم COD إن طُبّقت تُحسب وتظهر للعميل قبل التأكيد.

## **11) مكافحة الاحتيال والجودة (خصوصًا COD)**

* إشارات تحذير: تكرار إلغاء/فشل تسليم، أرقام مكرّرة، عناوين عالية المخاطر، حجم طلب غير اعتيادي لأول طلب.
* إجراءات: تأكيد واتساب/اتصال قبل الشحن لطلبات محددة، **OTP للتأكيد** عند الحاجة.
* قوائم حظر أو **تقييد COD** تلقائيًا عند تجاوز عتبات محددة.

## **12) الإشعارات والتواصل مع العميل**

* رسائل عند: إنشاء الطلب، التأكيد، الشحن (برابط تتبع)، الخروج للتسليم، التسليم، الفشل/إعادة الجدولة.
* القنوات: Push / SMS / واتساب (حسب الموافقات)، مع **روابط عميقة** لصفحة الطلب.

## **13) التكاملات والتحليلات**

* إرسال أحداث رئيسية إلى **Firebase/GA4**: begin\_checkout, add\_payment\_info, purchase, وأحداث حالات الشحن.
* ربط الطلب بـ UTM المصدر/الوسيط/الحملة إن توفر لقياس القنوات.
* تقارير تشغيلية: **نسبة التحويل من confirmed→delivered**، معدل فشل COD، متوسط زمن المعالجة، الطلبات حسب المدينة/شركة الشحن/القناة.

## **14) معايير القبول (Definition of Done)**

* إنشاء طلب كامل ببيانات عميل وعنوان **ملتقطة** وإجماليات صحيحة.
* حالات الطلب تعمل بانتقالات صحيحة ويُسجل تاريخ الحالة.
* ربط بنود الطلب بالمنتجات/النسخ بأسعار **ملتقطة**.
* دفع COD/أونلاين يعمل مع سيناريوهات الفشل/الرد.
* الحجز/التحرير للمخزون يعمل دون أخطاء أو حجز دائم.
* شحنة واحدة على الأقل لكل طلب قابل للشحن، مع رقم تتبع وحالات شحن.
* إشعارات العميل تُرسل حسب الحالة ووفق موافقات القنوات.
* لوحات تقارير أساسية متاحة (تحويل، فشل COD، زمن المعالجة).

## **15) سيناريوهات حافة يجب دعمها**

* **انقسام الشحنة**: صنف غير متوفر يُشحن منفصلًا، مع تحديث الإجماليات والتواصل.
* **استرداد جزئي**: مرتجع جزئي بعد التسليم (أونلاين أو رصيد متجر).
* **فشل تحصيل COD مع تسليم**: استثناءات نادرة—تسوية داخلية وإجراءات صارمة لاحقة.
* **تعديل عنوان بعد التأكيد**: يسمح قبل الشحن فقط ومع تسجيل السبب.
* **إلغاء شركة الشحن** بعد محاولات فاشلة: إغلاق الطلب وتحديث ملف مخاطر العميل.

## **16) تشغيلات مختصرة (SOPs)**

* مراجعة يومية للطلبات **pending** لأكثر من X ساعات.
* متابعة شحنات **out\_for\_delivery** نهاية اليوم (تأكيد نجاح/جدولة إعادة محاولة).
* تسويات أسبوعية مع شركات الشحن لمبالغ COD.
* مراجعة أسباب **failed\_delivery** وتصحيحها (عناوين/شركة شحن/تواصل).

السلة (Cart)

# **السلة (Cart)**

## **لماذا نفصل السلة عن المفضلة؟**

* **اختلاف النية:** السلة = نية شراء فورية؛ المفضلة = حفظ للاطلاع لاحقًا.
* **الأداء والعمر الافتراضي:** السلة لها **TTL** ومراجعة مخزون/سعر لحظية؛ المفضلة طويلة العمر ولا تحجز موارد.
* **الرسائل والتحويل:** تذكير “سلة مهجورة” لا ينطبق على المفضلة.
* **التحليلات:** فصل مصدر الحقيقة لمعدلات التحويل والمهجورة عن سلوك “إعجاب”.
* **الصيانة:** تغييرات التسعير/المخزون تؤثر على السلة فقط، وتسهل الاختبارات.

## **1) الهدف من السلة**

* تمكين الإضافة والتعديل بسرعة مع **سعر وتوفّر دقيقين لحظيًا**.
* دعم الدمج بين الأجهزة عند التوثيق.
* تجهيز الدخول السلس إلى الدفع (checkout) وتقليل فقدان الطلبات.

## **2) الكيانات (بدون تفاصيل تقنية)**

### **أ) السلة (Cart)**

* الهوية: cart\_id، customer\_id (إن وُجد)، anonymous\_id/device\_id، القناة (app/…)
* الحالة: **active / converted / abandoned**
* العملة: currency
* المجاميع التقديرية: subtotal، discount\_total، shipping\_estimate، tax\_estimate (إن وُجد)، cod\_fee\_estimate (إن وُجد)، **grand\_total\_estimate**
* العنوان (اختياري): مرجع عنوان لإظهار تقدير الشحن
* الكوبونات: promo\_code المطبّق + سبب الرفض إن فشل
* التواريخ: created\_at، **last\_activity\_at**، **expires\_at** (إن لزم)
* ملاحظات: حقل حر (اختياري)

### **ب) بنود السلة (Cart Items)**

* التعريف: product\_id، **variant\_id** (مصدر اللون/المقاس)
* الكمية: qty (مع حد أقصى لكل صنف إن وُضع)
* التسعير: **السعر الحالي**، والسعر عند الإضافة (للمقارنة فقط)
* التوفّر: حالة لحظية تُعاد مراجعتها عند كل فتح سلة/تعديل
* ملاحظات اختيارية (مثلاً ملاحظة مقاس)

## **3) قواعد العمل (Business Rules)**

* **الإضافة والتعديل**
  + لا تُقبل كمية تتجاوز المتاح؛ تُقصّ تلقائيًا لأقصى حد مع تنبيه.
  + عند تغيّر السعر/التوفّر: تُعرض شارة “سعر محدّث/غير متوفر”.
* **التسعير**
  + السلة دائمًا تعكس **السعر الحالي**؛ لا “قفل سعر” إلا عند إنشاء الطلب.
  + العروض تُطبَّق تلقائيًا إن انطبقت قواعدها.
* **الكوبونات**
  + التحقّق فور الإدخال وأي تعديل بالسلة.
  + سياسة التراكم واضحة (مثلاً كوبون واحد فقط).
* **الشحن والضرائب/رسوم COD (تقديرية)**
  + تحتاج مدينة/حي لإظهار تقدير دقيق؛ بدونها يُعرض نطاق/متوسط.
* **الهوية والدمج**
  + يبدأ المستخدم **ضيفًا**؛ عند توثيق الهاتف تُدمَج سلة الضيف بسلة العميل (تجميع نفس الـvariant).
* **المهلة (TTL) والسلة المهجورة**
  + تعتبر السلة **مهجورة** بعد عدم نشاط لمدة 7 أيام (قابلة للضبط).
  + تنظيف سلال الضيوف الخاملة دوريًا (مثلاً بعد 30–60 يومًا).
* **الحجز**
  + لا حجز مخزون أثناء التصفح. يُسمح **بحجز خفيف** فقط أثناء الدفع وحتى التأكيد.

## **4) التدفقات التشغيلية**

1. **إضافة للسلة:** تحقق لحظي من السعر/المخزون → تحديث المجاميع.
2. **بدء الدفع (Begin Checkout):** إعادة تحقق شامل (سعر/توفر/كوبون/شحن).
3. **التوثيق/الدخول:** دمج سلة الضيف مع سلة العميل فورًا.
4. **إنشاء الطلب:** التحقق الأخير + حجز خفيف → عند النجاح تتحول السلة إلى **converted**.

## **5) التكاملات المطلوبة**

* **الكتالوج/المخزون:** جلب السعر والتوفر من **النسخة (Variant)** أثناء كل تعديل.
* **العروض/الكوبونات:** تطبيق القواعد فورًا على السلة.
* **الشحن:** خدمة تقدير التكلفة حسب الوجهة والوزن/الحجم (إن لزم).
* **المدفوعات:** تمرير المجاميع النهائية إلى الدفع (مع رسوم COD إن طُبّقت).
* **التحليلات/الرسائل:** إرسال أحداث add\_to\_cart, begin\_checkout; تفعيل تذكير مهجور عبر Push/SMS/WhatsApp حسب الموافقات.

## **6) مؤشرات الأداء (KPIs)**

* Add-to-Cart من صفحات المنتج والقوائم.
* Begin Checkout → Purchase Conversion.
* قيمة وعدد السلال المهجورة حسب القناة/الفئة.
* أكثر العناصر إضافةً، وأبرز أسباب فشل التحويل (نفاد/سعر/شحن).

## **7) معايير القبول (Definition of Done)**

* إضافة/إزالة/تعديل كميات تعمل بسلاسة وبدون تضارب مخزون.
* الأسعار والتوفّر **محدّثة** دائمًا مع تنبيهات واضحة عند تغيّرها.
* دمج سلة الضيف مع سلة العميل عند التوثيق دون فقد عناصر.
* الكوبونات تُقبل/تُرفض برسائل مفهومة وتؤثّر على المجاميع فورًا.
* تقدير الشحن/COD يظهر بدقة عند توافر العنوان.
* التحويل إلى طلب يتم بدون أخطاء، والسلة تُوسَم **converted**.

## **8) سيناريوهات حافة**

* تغيّر السعر بعد الإضافة: تحديث السعر وعرض شارة تنبيه.
* نقص المخزون عند الدفع: تقليل الكمية لأقصى حد متاح مع تنبيه.
* صنف أزيل/أرشف: يُزال من السلة مع إشعار المستخدم.
* سلتان لنفس العميل من جهازين: تُدمج عند أول مزامنة.
* تعارض كوبونات: اتباع سياسة ثابتة (مثلاً الأحدث يحل محل السابق).

## **9) خارطة إصدارات**

* **MVP:** سلة ضيف/عميل، دمج عند التوثيق، تحقق لحظي سعر/مخزون، كوبون واحد، تقدير شحن أساسي، تذكير مهجور بعد 24–72 ساعة.
* **المرحلة 2:** حدود شراء لكل SKU، قواعد كوبونات متقدمة، حفظ سلة أطول للضيوف.
* **المرحلة 3:** توصيات داخل السلة (Cross-sell/Bundle)، تقدير شحن دقيق متعدد الشركات، A/B لاختبار رسائل المهجورة.

علامة التبويب 28

**الهدف**

* تحديد المنتجات الأكثر إضافةً إلى السلة بدقة (SKU/SPU/فئة).
* تشخيص أسباب فشل التحويل داخل السلة مع نسب وتأثير مالي (نفاد المخزون، السعر، الشحن).
* إنتاج لوحة قياس عملية ترتّب “قائمة أولويات الإصلاح”.

# **مصادر البيانات المطلوبة (ملخّص)**

* سلوك المستخدم: add\_to\_cart، remove\_from\_cart، begin\_checkout، view\_shipping, shipping\_quote\_shown, apply\_coupon, price\_view, payment\_failed، exit\_step.
* حالة المخزون للحظة التفاعل: available\_qty, reserved\_qty, backorder\_flag, restock\_eta.
* التسعير: list\_price، current\_price، campaign\_price، competitor\_gap (إن وُجد)، price\_changed\_flag.
* الشحن: shipping\_zone, delivery\_eta, shipping\_fee, cod\_allowed, serviceable (y/n).
* بنية السلة: cart\_id، user\_id، line\_id، sku، qty، timestamps.

# **I) أكثر العناصر إضافةً (Top Added Items)**

## **التعاريف**

* “إضافة” = كل حدث add\_to\_cart ناجح (بعد تحقق القيود: المقاس/اللون…).
* **TAI\_Units**: إجمالي الوحدات المضافة لكل SKU خلال فترة محددة.
* **TAI\_Users**: عدد المستخدمين الفريدين الذين أضافوا SKU.
* **TAI\_Rate**: (عدد جلسات أضيف فيها SKU) ÷ (جلسات شاهدت SKU).
* **Net Adds**: الإضافات − الإزالات (remove\_from\_cart) لنفس الفترة.

## **التقارير الأساسية**

1. قائمة Top-N حسب:
   * TAI\_Units (افتراضي)، مع عرض TAI\_Users وTAI\_Rate وNet Adds.
2. تفصيل بالبعد:
   * اللون/المقاس، الفئة، المصدر التسويقي، المنطقة.
3. جودة المخزون لمفضّلات العملاء:
   * %النفاد خلال زيارات السلة لكل SKU، restock\_eta المتوسط.
4. حساسية السعر:
   * نسبة الإزالات بعد **price\_changed\_flag** أو بعد عرض الشحن.

## **قواعد العرض والفرز**

* الفرز الأساسي: TAI\_Units ثم TAI\_Rate.
* استبعد SKUs بنشاط احتيالي/اختبارات داخلية (internal traffic).
* طبّق حدًا أدنى للظهور (≥ 200 مشاهدة SKU أو ≥ 50 إضافة) لتجنّب الضجيج.

# **II) أسباب فشل التحويل في السلة (نفاد/سعر/شحن)**

السبب = أقوى عامل قابل للإثبات “عند آخر تفاعل ذي صلة” قبل الخروج أو قبل فشل الطلب.

## **1) نفاد المخزون (OOS)**

**تعريف السبب:**

* المنتج أو أي متغير منه أصبح available\_qty ≤ 0 عند:
  + محاولة “التوجّه للدفع” أو
  + تحديث الكمية في السلة أو
  + ظهور رسالة OOS/Not serviceable في الـUI.
* أو تم تقليص الكمية تلقائيًا (auto-reduce) قبل الدفع.

**إثبات الحدث (signals):**

* oos\_message\_shown=true، أو server\_response=“OUT\_OF\_STOCK”.
* اختلاف qty قبل/بعد التحديث (مثلاً من 2 إلى 1).
* restock\_eta معروف (اختياري للمعالجة اللاحقة).

**مؤشرات تعرض:**

* OOS Rate = عدد سلال شهدت OOS ÷ إجمالي سلال وصلت لمرحلة مراجعة السلة.
* Lost Units = الفرق بين qty المطلوبة والمتاحة.
* Lost Revenue (estimate) = Lost Units × current\_price.

**أولويات الإصلاح:**

* SKUs ذات TAI\_Units عالية + OOS Rate مرتفع + restock\_eta بعيد.

## **2) السعر (Price)**

**تعريف السبب:**

* price\_changed\_flag بين وقت الإضافة ووقت المراجعة (زيادة).
* غياب الخصم/القسيمة المتوقعة (coupon\_rejected، campaign\_not\_applied).
* فجوة سعرية كبيرة مقابل مرجع داخلي (مثلاً أعلى من P90 لفئة مماثلة) أو مقابل منافس إن متاح.

**إثبات الحدث:**

* remove\_after\_price\_change خلال ≤ 2 ساعات من price\_changed\_flag.
* abandon بعد apply\_coupon\_failed أو بعد view\_price\_rules.
* bounce عند صفحة “المجموع” دون بدء الدفع.

**مؤشرات تعرض:**

* Price-Abandon Rate = سلال شهدت price\_change أو coupon\_failed ثم لم تبدأ الدفع.
* Net Adds→Removals بعد عرض الرسوم النهائية.
* Elasticity Proxy: نسبة الإزالات عند زيادات سعرية صغيرة (مثلاً +5%).

**أولويات الإصلاح:**

* منتجات “Top Added” مع أعلى Price-Abandon Rate.
* سيناريوهات “خصم موعود غير مطبّق” (تصحيح قواعد الكوبون/الحملة).

## **3) الشحن (Shipping)**

**تعريف السبب:**

* shipping\_quote\_shown وبعده exit خلال ≤ 10 دقائق بدون بدء الدفع.
* shipping\_fee أعلى من عتبة المنطقة (threshold) أو delivery\_eta طويل.
* المنطقة غير مخدومة (serviceable=false) أو COD غير متاح لعنوان العميل.

**إثبات الحدث:**

* provider\_code=“ZONE\_NOT\_SERVICEABLE”/“COD\_NOT\_ALLOWED”.
* زيادة واضحة في remove\_from\_cart بعد عرض رسوم/مدة الشحن.
* انتقال المستخدم بين طرق شحن ثم خروج (change\_shipping\_method≥2 ثم exit).

**مؤشرات تعرض:**

* Shipping-Abandon Rate = سلال رأت عرض الشحن ثم غادرت قبل begin\_checkout.
* % Non-serviceable by city/zone.
* تأثير تكلفة الشحن: معامل ارتباط بين shipping\_fee وبدء الدفع لكل شريحة.

**أولويات الإصلاح:**

* مناطق ذات Non-serviceable أعلى وتأثير إيرادي كبير.
* SKUs ثقيلة/كبيرة مع أعلى Shipping-Abandon.

## **أسبقية السبب عند تزامن عوامل متعددة**

1. نفاد المخزون (سبب حتمي يمنع الشراء).
2. الشحن (عدم الخدمة/تكلفة/مدة تُعدّ مانعًا وظيفيًا).
3. السعر (عامل قرار، يُسجّل إن لم يوجد 1 أو 2).

سجّل السبب الرئيسي وسببًا ثانويًا إن وُجد، لكن اعتمد الرئيسي في التقارير الافتراضية.

# **III) لوحة “قائمة أولويات الإصلاح” (Fix Queue)**

* لكل SKU/فئة:
  1. حجم الطلب: TAI\_Units وTAI\_Users
  2. معدل فشل التحويل لكل سبب (OOS/Price/Shipping)
  3. الإيراد المفقود التقديري لكل سبب
  4. “درجة الأولوية” = تطبيع (0–100) من: حجم الإضافة × (معدل الفشل) × (الإيراد المفقود)
* فلترة بحسب: مصدر تسويقي، مدينة، جهاز، شريحة (VIP، مهجور سلة).

# **IV) قراءات قياسية/أهداف**

* OOS Rate في السلة < 2% لمنتجات Top-100.
* Price-Abandon بعد تغيّر السعر < 10%.
* Shipping-Abandon بعد عرض الرسوم < 12% (يتفاوت حسب السوق).
* استعادة 20–30% من مهجوري الشحن عبر بدائل الدفع/الشحن.

# **V) قرائن مساندة (لتحسين الدقة)**

* استبيان خروج سريع (اختياري): زر “لماذا لم تكملين؟” بخيارات (السعر/الشحن/الوقت/طريقة الدفع)، يُستخدم للمعايرة.
* دمج بيانات دعم العملاء (Tags: “غالي”، “لا يشحن لمنطقتي”).
* تتبّع رسائل الخطأ/التنبيهات المعروضة للمستخدم كـ signals رسمية.

# **VI) جودة البيانات والحوكمة**

* ضبط المنطقة الزمنية “Asia/Aden” لكل المقاييس.
* منع ازدواج الأحداث (مثلاً add\_to\_cart المتكرر لنفس line\_id).
* استبعاد حركة داخلية/اختبارات (قائمة allowlist للأجهزة/المستخدمين).
* مراقبة التناسق: أي SKU يظهر Top Added ثم صفر طلبات أسبوعًا كاملًا ⇒ تحقق من بوابة الشحن/الدفع.

# **VII) إجراءات سريعة بناءً على النتائج**

* نفاد: رفع نقاط إعادة الطلب، تمكين بدائل/ألوان، إظهار restock\_eta، تفعيل “إشعار إعادة التوفر”.
* السعر: تثبيت تسعير متّسق حتى نهاية الجلسة، مراجعة سياسات القسائم، إبراز القيمة (جودة/خامة/مقارنة).
* الشحن: تخفيض شحن مناطق رئيسية، تمكين COD انتقائي مع حد قيمة، إضافة خيار “استلام من نقطة تسليم” إن أمكن.

علامة التبويب 29

**آلية معيارية لإدارة الكوبونات المدخلة يدويًا والكوبونات التلقائية داخل السلة والدفع**،

# **الهدف**

* تطبيق أفضل خصم ممكن للمستخدم مع وضوح القواعد.
* منع التعارضات/التكديس غير المرغوب.
* تسجيل أسباب القبول/الرفض للتدقيق والتحليل.

# **1) التعاريف**

* **Manual Coupon**: كود يدخله المستخدم (مثل VIP20).
* **Auto Coupon/Promotion**: خصم يُحتسب تلقائيًا إذا تحققت شروط (أول طلب، حد سلة، فئة، شريحة…).
* **Benefit Type**: نوع المنفعة (نسبة %، مبلغ ثابت، شحن مجاني، هدية/BxGy، كاش باك).
* **Eligibility**: شروط الاستحقاق (قيمة سلة، منتجات/فئات مسموحة/مستثناة، قناة، شريحة، طريقة دفع، مدينة…).
* **Stacking**: إمكانية الجمع بين خصمين أو أكثر.
* **Suppression**: إيقاف عرض/تطبيق عرض تلقائي عندما يطبَّق عرض أعلى أولوية.

# **2) قواعد عامة (Policy)**

1. **ضمان أفضل سعر**: النظام يحسب جميع العروض المؤهَّلة ويختار مجموعة الخصومات المسموح جمعها التي تعطي **أقل إجمالي صافٍ**وفق قواعد التكديس.
2. **أولوية المقصد**:
   * (أ) **أكواد فريدة/دعوات** (Single-use/Referral/Influencer) أعلى أولوية، ولا تتكدّس مع خصومات نقدية أخرى.
   * (ب) **الكود اليدوي** المُدخل في الجلسة الحالية يسبق العروض التلقائية **إن قدّم سعرًا أفضل**.
   * (ج) العروض التلقائية تُطبَّق “أفضل خيار تلقائي” عند عدم وجود كود يدوي مؤهَّل أفضل.
3. **مصفوفة التكديس (Combinability Matrix)**:
   * خصم نقدي (٪ أو مبلغ) ⇢ **لا يتكدّس** مع خصم نقدي آخر.
   * “شحن مجاني” ⇢ **يمكن** أن يتكدّس مع **خصم نقدي واحد**.
   * “هدية/BxGy” ⇢ التكديس **مسموح** إذا لم تخصم من نفس البنود وبشرط ألا تتجاوز **الحد الأقصى للخصم**.
   * “كاش باك” ⇢ يمكن تكديسه مع واحد نقدي **أو** شحن مجاني (اختر واحد نقدي فقط).
4. **الحد الأقصى للخصم (Cap)**: لكل طلب حد أقصى نسبي/مطلق يقي القيود المالية حتى عند التكديس المسموح.
5. **الأولوية داخل الفئة الواحدة**: في العروض التلقائية ذات الطبقات (Threshold Tiers) طبّق **أعلى طبقة مؤهَّلة فقط**.
6. **شفافية الواجهة**: إن وجد كود يدوي أضعف من عرض تلقائي، أظهري للمستخدم خيارين:
   * “استخدام الكود اليدوي” (يُطبّق الكود ويُوقَف التلقائي)
   * “الحفاظ على أفضل خصم” (يُقمع الكود اليدوي ويُحفظ في الحقل لتجربته لاحقاً)
7. **طريقة الدفع/المخاطر**: يمكن تقييد خصومات معينة بـ **الدفع المسبق فقط** أو حجبها عن **COD\_RISK\_HIGH**.
8. **عدم الرجعية**: بعد إنشاء الطلب، لا يُعاد تسعيره تلقائيًا إذا تغيّر الكوبون/العرض.

# **3) نموذج البيانات (ملخّص حقول لازمة)**

* **promotions**: code/null, benefit\_type, value, max\_discount, start/end, channel, stacking\_rule, priority, scope (basket/item/shipping), audience (segment/region/payment), usage\_limits(per\_user/per\_campaign/global), eligible\_categories/skus, exclusions, is\_unique, owner(influencer/referral).
* **promotion\_ledger**: سجل محاولات التطبيق: cart\_id, code, auto\_flag, eligible(y/n), reject\_reason, estimated\_saving, applied(y/n), suppressed\_by.
* **cart\_lines snapshot** عند التسعير: الأسعار قبل/بعد الخصم، البنود المتأثرة، سبب التغيير.

# **4) ترتيب التقييم (Evaluation Pipeline)**

1. **جمع المرشّحين**:
   * الكود اليدوي (إن وُجد) + جميع العروض التلقائية المستوفية للشروط.
2. **التحقق من الأهلية** لكل مرشح: الشروط، الحدود، القنوات، الشريحة، طريقة الدفع، المنتجات المستثناة.
3. **تجميع المرشّحين** وفق **مصفوفة التكديس** لتكوين حالات تسعير محتملة (Scenarios).
4. **حساب صافي السلة** لكل سيناريو + تطبيق **Cap**.
5. **اختيار السيناريو الأفضل** (أقل إجمالي، مع كسر التعادل بالأولوية priority ثم المنفعة الأشمل “basket > shipping > item”).
6. **القمع/التوضيح**: وسم العروض غير المختارة كـ suppressed مع سبب واضح.
7. **تثبيت النتيجة** حتى تغيّر جوهري (تبديل كود، تغيير كمية، تغيير شحن/عنوان/طريقة دفع).

# **5) أسباب الرفض/القمع القياسية (Reason Codes)**

* not\_eligible\_min\_basket (لم يتحقق حد السلة)
* product\_excluded (منتج/فئة خارج النطاق)
* payment\_method\_blocked (طريقة الدفع غير مؤهَّلة)
* region\_not\_serviceable (المنطقة غير مدعومة/رسوم مخصّصة)
* usage\_limit\_reached\_user/global
* coupon\_expired/not\_started
* stacking\_conflict (تعارض مع خصم نقدي آخر)
* better\_auto\_applied (عرض تلقائي أفضل)
* risk\_blocked (حظر بسبب درجة مخاطرة COD)

تُسجَّل في promotion\_ledger لكل محاولة، وتُستخدم في التقارير وتحسين الحملات.

# **6) سيناريوهات معيارية**

* **أول طلب 10% (تلقائي)** + **يدوي 15%** ⇢ يطبَّق 15% ويُقمع التلقائي (better\_auto\_applied=false).
* **يدوي 20% + شحن مجاني تلقائي** ⇢ يُسمح بالتكديس (نقدي واحد + شحن مجاني).
* **BxGy تلقائي + يدوي 10%** ⇢ إذا كان BxGy يخصم من نفس البنود، اختاري **الأفضل** فقط؛ إن كانت الهدية على بنود أخرى، يمكن التكديس مع **Cap**.
* **كود مؤثر (Unique)** ⇢ يمنع أي خصم نقدي آخر؛ يبقى الشحن المجاني قابلًا للتكديس إن سمح إعداد الكود.
* **COD\_RISK\_HIGH** ⇢ يمنع خصومات نقدية > X%، ويسمح بشحن مجاني فقط (حسب السياسة).
* **تحويل طريقة الدفع إلى مسبق** ⇢ يفتح خصم Payment Boost تلقائيًا ويُعاد التقييم.

# **7) السلوك في الإرجاع/الإلغاء**

* **توزيع الخصم** على البنود بنسبة مساهمتها في الإجمالي **قبل** الخصم.
* عند الإرجاع الجزئي: يُعاد احتساب الحسم على البنود المعادة، ويُسترد المبلغ المناسب فقط.
* عند إلغاء كامل: تُزاد عدّادات الاستخدام (usage\_limits) إن كانت تُخصم عند إنشاء الطلب.

# **8) واجهة المستخدم (نقاط إلزامية)**

* شارة “**تم تطبيق أفضل خصم تلقائيًا**” مع إمكانية “استخدام كود آخر”.
* عند إدخال كود أضعف: تنبيه **“الكود يمنح خصمًا أقل من العرض الحالي”** مع خيار الاستبدال.
* كشف تفصيلي: (الخصم النقدي، الشحن المجاني، هدية) + إجمالي التوفير.
* رسائل سبب الرفض من **Reason Codes** أعلاه بصياغة مفهومة.

# **9) الضبط والحوكمة**

* لوائح **Limits** افتراضية:
  + لكل مستخدم: حد استخدام/حملة (مثلاً 1–2)
  + لكل طلب: خصم نقدي واحد + شحن مجاني + (اختياري) هدية
  + **Cap** افتراضي: لا يتجاوز X% من إجمالي السلة
* استثناءات تُدار عبر priority وis\_unique.
* استبعاد حركة داخلية/اختبار من العدّادات.

# **10) التقارير والمؤشرات**

* **Acceptance Rate** لكل كود/حملة (محاولات ⇢ تطبيقات).
* **Suppression Map**: ما أكثر العروض التي تُقمع ولماذا (يُرشد لتبسيط القواعد).
* **Avg Discount % / Order** و**Impact on AOV** و**Gross Margin**.
* **Leakage**: محاولات رفض بسبب usage\_limit\_reached أو not\_eligible\_min\_basket.
* **Cross-channel**: أداء أكواد المؤثرين مقابل التلقائي لكل مصدر/شريحة.

# **11) اختبار (QA) مختصر**

* إدخال كود صالح + تلقائي أقوى ⇒ يُحترم الأفضل وتظهر رسالة توضيح.
* طبقات تلقائية (100/200/300) ⇒ عند تغيّر قيمة السلة تُطبَّق أعلى طبقة فقط.
* منع جمع خصمين نقديين ⇒ تحقّق من تطبيق **Cap** وسبب stacking\_conflict.
* تغيير العنوان/المنطقة ⇒ إعادة تقييم الشحن وسبب region\_not\_serviceable.
* إرجاع جزئي ⇒ التوزيع النسبي للخصم سليم.

## **Check**

* جداول/حقول الحملات مع benefit\_type وstacking\_rule وpriority وusage\_limits.
* مسار تقييم موحّد (pipeline) يُعيد **السيناريو الأفضل** + تفسير (explain) + أسباب الرفض.
* تسجيل كامل في promotion\_ledger (eligible/applied/suppressed + reasons).
* مصفوفة تكديس مطبّقة كما في القسم (3).
* سياسات الإرجاع وتوزيع الخصم النسبي مفعّلة.
* واجهة رسائل واضحة للمستخدم.
* تقارير قبول/قمع وتأثير مالي أسبوعية.

المفضّلة (Wishlist)

# **المفضّلة (Wishlist)**

## **1) الهدف**

* حفظ المنتجات التي تهم العميلة للعودة لاحقًا.
* رفع العودة للتطبيق والتحويل عبر إشعارات **عودة التوفر** و**هبوط السعر**.
* دعم مشاركة قوائم مختارة (اختياري) لتحفيز الشراء.

## **2) النطاق**

* **قائمة مفضّلة** لكل عميلة (واحدة كحد أدنى)، مع خيار **عدّة قوائم** لاحقًا.
* **عناصر المفضّلة** على مستوى **المنتج أو النسخة** (الأفضل النسخة إذا كان اللون/المقاس مهمّين).
* **إعدادات إشعارات**: عند التوفّر، عند هبوط السعر (اختياري تشغيل/إيقاف لكل عنصر).
* **مشاركة القائمة** برابط عام للعرض فقط (اختياري في مرحلة لاحقة).

## **3) الهوية والربط**

* الافتراضي: المفضّلة **للمستخدم المسجّل** (مرتبطة بـ customer\_id).
* للضيف: مفضّلة مؤقتة على الجهاز؛ **تُدمَج** تلقائيًا مع حساب العميلة عند التوثيق (مع إزالة التكرارات).
* المزامنة بين الأجهزة بعد الدخول.

## **4) الحقول (مستوى المفهوم، بدون تفاصيل تقنية)**

### **أ) المفضّلة**

* المعرف، **customer\_id**، الاسم (لعدّة قوائم لاحقًا)، **is\_default**، **is\_public** (اختياري)، رابط مشاركة (token)، أوقات الإنشاء/التحديث، **items\_count**.

### **ب) عنصر المفضّلة**

* **product\_id** و/أو **variant\_id** (الأولوية للـ variant عندما يغيّر اللون الصور).
* **added\_at**، **notes** (اختياري للعميلة).
* إعدادات الإشعارات: **notify\_when\_in\_stock**، **notify\_on\_price\_drop**.
* معلومات مرجعية للعرض: السعر الحالي، حالة التوفّر الحالية (تُحدَّث عند الفتح، ليست حقلًا ثابتًا).
* (اختياري) **price\_at\_add** لعرض “انخفض السعر منذ الإضافة”.

## **5) قواعد العمل**

* لا تأثير للمفضّلة على المخزون أو التسعير.
* **منع التكرار**: لا يتكرر نفس المنتج/النسخة داخل القائمة.
* **حد أقصى منطقي** لعناصر المفضّلة (مثلاً 500 عنصر للقائمة الواحدة).
* العناصر المرتبطة بمنتجات **مؤرشفة/محذوفة**: تُخفى مع رسالة توضيح، وخيار إزالة سريعة.
* المستوى الأفضل للإضافة: **variant** (حتى تظهر الصور الصحيحة مع اللون).
* الإشعارات:
  + **عودة التوفّر**: تُرسل مرة عند الانتقال من غير متاح → متاح، مع **فترة تهدئة** (مثلاً 7 أيام) لنفس العنصر لنفس العميلة.
  + **هبوط السعر**: تُرسل عند انخفاض ≥ نسبة محددة (مثلاً 10%) أو قيمة مطلقة، مع **فترة تهدئة** مشابهة.
  + تُحترم موافقات القنوات (Push/SMS/WhatsApp).
* المشاركة: القوائم العامة **قراءة فقط** عبر رابط، ويمكن للزائر **استنساخ** العناصر إلى مفضّلته بعد الدخول.

## **6) التدفقات التشغيلية**

1. **إضافة إلى المفضّلة** من صفحة المنتج/القائمة (زر قلب).
2. **إزالة/تنظيم**: إزالة فردية، إزالة الكل، فرز بحسب “أُضيفت مؤخرًا/السعر/التوفّر”.
3. **نقل إلى السلة**: “نقل” أو “إضافة” مع التحقق من اللون/المقاس المتوفر.
4. **دمج الضيف عند الدخول**: اتحاد القوائم وإزالة التكرارات.
5. **إشعارات**: مهمة يومية/فورية تتحقق من تغيّر التوفّر/السعر وتُرسل رسائل مقيّدة بموافقات القنوات.
6. **مشاركة (اختياري)**: توليد رابط للقائمة، مع صفحة عرض عامة، وخيار “نسخ إلى مفضلتي”.

## **7) التكاملات**

* **الكتالوج/المخزون**: قراءة حالة النسخة والتسعير لحظيًا لعرضها داخل المفضّلة.
* **العروض/التسعير**: مقارنة السعر الحالي و**price\_at\_add** لتحديد “هبوط السعر”.
* **الرسائل (FCM/SMS/WhatsApp)**: إرسال الإشعارات مع **Deep Links** مباشرة إلى المنتج/النسخة.
* **التحليلات (GA4/Firebase)**: add\_to\_wishlist, remove\_from\_wishlist, view\_wishlist, share\_wishlist, wishlist\_to\_cart, notif\_open\_back\_in\_stock, notif\_open\_price\_drop.

## **8) مؤشرات الأداء**

* نسبة المستخدمين الذين يستخدمون المفضّلة، ومتوسط العناصر لكل مستخدم.
* **تحويل من المفضّلة إلى الشراء** (Wishlist → Purchase) وزمن التحويل.
* أداء إشعارات **عودة التوفّر** و**هبوط السعر** (فتح/نقر/شراء).
* أكثر العناصر حفظًا، والعناصر التي تُحفظ طويلًا دون شراء.
* الرفع في **AOV** عندما تنتقل العناصر من المفضّلة إلى السلة.

## **9) معايير القبول (Definition of Done)**

* عميلة مسجّلة: تملك مفضّلة افتراضية ويمكنها الإضافة/الإزالة والتنظيم.
* ضيفة: تملك مفضّلة محلية تُدمج دون تكرار عند الدخول.
* عرض فوري لحالة **التوفّر** والسعر الحالي داخل المفضّلة.
* “نقل إلى السلة” يعمل ويطلب اختيار المقاس/اللون عند الحاجة.
* إشعارات عودة التوفّر/هبوط السعر تعمل، مع احترام الموافقات وفترات التهدئة.
* (اختياري) مشاركة قائمة تنشئ رابطًا للعرض فقط ويمكن استنساخها بعد الدخول.

## **10) سيناريوهات حافة**

* تغيّر الصور حسب اللون: عند حفظ variant مختلف، يجب تبديل الصور وفق اللون عند العرض.
* قيود إقليمية/عملة: لو تغيّرت العملة أو السوق، تُعرض الأسعار حسب واجهة السوق الحالي.
* عناصر اشتريت لاحقًا: خيار “إخفاء العناصر المشتراة” تلقائيًا X يومًا.
* قوائم ضخمة: صفحات/تحميل كسول مع بحث داخل المفضّلة.
* منتج أصبح مؤرشفًا: يظهر كتالي “لم يعد متاحًا” مع زر إزالة.

## **11) خارطة الإصدارات**

* **MVP (موصى به للإطلاق):**
  + مفضّلة واحدة خاصة لكل مستخدم مسجّل + مفضّلة ضيف تُدمج عند الدخول.
  + إضافة/إزالة/نقل إلى السلة، عرض السعر والتوفّر الحاليين.
  + إشعار **عودة التوفّر** (Push) مع Deep Link.
* **المرحلة 2:**
  + عدّة قوائم بأسماء (سهرة/يومي/هدايا).
  + إشعار **هبوط السعر** مع عتبة ونطاقات تهدئة.
  + مشاركة قائمة برابط للعرض فقط.
* **المرحلة 3:**
  + اقتراحات ذكية داخل المفضّلة (تكميلي/بدائل متاحة)، فرز متقدّم، تصفية داخل المفضّلة.

المدفوعات (Payments) — المتطلبات المعتمدة

# **المدفوعات (Payments)**

## **1) الهدف**

* قبول المدفوعات **أونلاين** و**الدفع عند الاستلام (COD)** بكفاءة وأمان.
* تقليل فشل الدفع ورفع نسبة التحويل.
* تسوية دقيقة للأموال: بوابة الدفع، و**تحصيلات شركات الشحن** للـCOD.
* دعم الاسترداد الجزئي/الكامل والاعتراضات (Chargebacks) عند الحاجة.

## **2) النطاق**

* طرق دفع: بطاقة/محافظ/تحويل محلي (أونلاين)، و**COD**.
* إدارة دورة الحياة: **محاولة دفع** → تفويض/تحصيل → استرداد/إلغاء.
* تسويات: تقارير تسوية بوابة الدفع، و**دفعات شركات الشحن** للـCOD.
* تكامل مع **الطلبات**، **الشحنات**، **المرتجعات**، **التقارير**، والإشعارات.

## **3) أنواع الدفع المدعومة**

* **أونلاين:** بطاقات (مع 3-D Secure)، محافظ (Apple/Google/محلي)، تحويل/فوترة بنكية (حسب المتاح).
* **COD:** تحصيل نقد/تحويل عند التسليم عبر شركة الشحن.
* **اختياري لاحقًا:** شراء الآن وادفع لاحقًا (BNPL).

## **4) الكيانات (مفهومية — بدون تفاصيل تقنية)**

* **Payment (الدفع):** رأس العملية المرتبطة بطلب واحد، يمكن أن يحتوي عدة محاولات.
* **Payment Attempt (محاولة):** كل محاولة عبر مزوّد/قناة معيّنة، مع نتيجتها.
* **Payment Method (وسيلة):** نوع الوسيلة/المعرّف المرمّز (Token) إن وُجد.
* **Refund (استرداد):** استرداد كامل/جزئي مرتبط بدفع ناجح.
* **COD Remittance (تسوية COD):** حركة تحصيل شركة الشحن حتى توريدها لنا (Remit).
* **Dispute/Chargeback (اعتراض):** في الأونلاين، إن حدث نزاع.
* **Settlement Batch (دفعة تسوية):** دفعات مجمّعة من المزوّد/شركة الشحن للمطابقة المحاسبية.

## **5) حالات الدفع (State Machine)**

### **أ) أونلاين**

* initiated → authorized (تفويض) → captured (تحصيل)
* مسارات جانبية: failed (فشل)، voided (إلغاء تفويض)، refunded\_partial / refunded\_full.
* ملاحظات:
  + قد نستخدم **التحصيل المؤجّل**: نفوّض أولًا ثم نُحصّل عند الشحن.
  + في حالة فشل بوست-باك أو انقطاع، يتم الاعتماد على **Webhook** لإثبات الحالة النهائية.

### **ب) COD**

* pending\_collection (بانتظار التحصيل) → collected\_by\_courier (تم التحصيل من العميل) → remitted (تم توريدها لنا)
* مسارات جانبية: failed\_delivery، discrepancy (فرق توريد)، write\_off (معالجة فرق نهائي).

## **6) قواعد العمل (Business Rules)**

* **ارتباط الطلب:** لا يُوسَم الطلب “مدفوع” إلا بعد captured (أونلاين) أو بعد remitted (COD) إذا أردنا دقة محاسبية، مع إبقاء “مسلم” على مستوى العمليات.
* **محاولات الدفع:** حد أقصى X محاولات لكل طلب؛ إظهار رسائل واضحة للأخطاء الشائعة.
* **التفويض/التحصيل:** التفويض صالح N أيام؛ إن فاتت يُعاد التفويض قبل الشحن.
* **العملة/الصرف:** تثبيت **سعر الصرف** لحظة الدفع/التحصيل وتخزين مصدره.
* **الاسترداد:** يسمح بجزئي/كامل؛ في COD يكون **نقدًا/تحويلًا** أو **رصيد متجر**.
* **الاعتمادية:** كل إنشاء/تحديث دفع يجب أن يكون **Idempotent** (مفتاح تكرار).
* **الامتثال:** لا تُخزَّن بيانات بطاقة خام؛ استخدام **Tokenization** فقط، وتفعيل 3DS للمعاملات التي تتطلب SCA.
* **COD والمخاطر:** ربط التحصيل بنتيجة الشحن؛ عند **failed\_delivery** تُرفع مؤشرات مخاطر العميل.

## **7) التدفقات التشغيلية**

### **7.1 أونلاين — تفويض ثم تحصيل**

1. العميل يضيف وسيلة دفع → **محاولة دفع** → 3DS (إن لزم).
2. عند authorized: نحجز المخزون ونجهّز الشحن.
3. عند الشحن: نُحوّل إلى captured.
4. في المرتجعات: **استرداد** جزئي/كامل حسب بنود المرتجع.

### **7.2 أونلاين — تحصيل فوري**

* من initiated إلى captured مباشرة عند إتمام الدفع (ملائم للسلع الرقمية/الشحن الفوري).

### **7.3 COD — التحصيل والتوريد**

1. عند إنشاء الطلب بـCOD: pending\_collection.
2. عند تسليم المندوب: collected\_by\_courier مع تسجيل المبلغ الفعلي.
3. عند توريد شركة الشحن أسبوعيًا/دفعات: remitted لكل شحنة/طلب، مع ربط **دفعة التسوية**.
4. **فروقات التوريد**: إن وُجد فرق (زيادة/نقص)، تسجيله في discrepancy ومعالجته.

### **7.4 الاسترداد/الإلغاء**

* قبل الشحن: إلغاء التفويض أو عكس التحصيل إن كان فوريًا.
* بعد التسليم: استرداد عبر البوابة (أونلاين) أو تحويل/رصيد متجر (COD).

### **7.5 الاعتراضات (Chargebacks)**

* عند استلام إشعار نزاع من البوابة: تحديث حالة الدفع، جمع أدلة (تتبع/تسليم/تواصل)، والرد ضمن المهلة.

## **8) التكاملات**

* **الطلبات:** تحديث حالة الدفع والإجماليات؛ لقطة الأسعار/العملات لحظة الطلب.
* **الشحنات:** تشغيل التحصيل عند الشحن (أونلاين)، وتحديث تحصيلات COD عند التسليم.
* **المرتجعات:** إنشاء استرداد متوافق مع بنود المرتجع.
* **بوابة الدفع:** صفحات الدفع/SDK + **Webhooks** موثوقة مع إعادة محاولات واستيثاق توقيع.
* **شركات الشحن (COD):** ملف/واجهة تسويات أسبوعية (CSV/API) تتضمّن: رقم الشحنة/الطلب، المبلغ المُحصّل، رسوم الشحن، صافي المبلغ، تاريخ التوريد.
* **المحاسبة:** ترحيل إلى دفاتر مزدوجة (إيرادات/خصومات/رسوم/حساب وسيط COD).
* **الإشعارات:** رسائل “تم الدفع/فشل الدفع/تم الاسترداد” حسب القناة المصرّح بها.

## **9) التقارير والمؤشرات (KPIs)**

* **نسبة نجاح الدفع أونلاين** (Authorization Rate / Capture Rate).
* **معدل فشل الدفع** حسب الوسيلة/الجهاز/الشبكة.
* **نسبة تحصيل COD** ومعدل **التوريد** خلال ≤7 أيام.
* **فروقات التوريد (Discrepancy Rate)** مع شركات الشحن.
* **متوسط زمن التسوية** لكل مزوّد/شركة شحن.
* **نسبة الاسترداد** و**الاعتراضات**، وقيمتها.
* **تكلفة الدفع** (رسوم مزوّد/رسوم COD) لكل طلب.

## **10) معايير القبول (Definition of Done)**

* يمكن إنشاء دفع مرتبط بطلب مع عدة محاولات، وتُعرض الحالات بدقة.
* دعم **تفويض ثم تحصيل** و**تحصيل فوري** للأونلاين.
* COD يعمل بكامل الدورة: تحصيل عند التسليم → توريد → إغلاق دفعة التسوية.
* الاسترداد الجزئي/الكامل يعمل ويرتبط بالمرتجع.
* التكامل مع Webhooks موثوق مع إعادة محاولات وفحص توقيع.
* تقارير تسوية: تطابق إجمالي المعاملات مع دفعات المزوّد/شركة الشحن دون فروقات غير مفسَّرة.
* إشعارات العميل تُرسل بالحالات الرئيسية.

## **11) سيناريوهات حافة**

* **انقطاع بعد الدفع الناجح:** يُستعاد الوضع عبر Webhook لاحقًا بدون ازدواج.
* **تفويض منتهي قبل الشحن:** إعادة تفويض تلقائي أو طلب إعادة الدفع.
* **تحصيل COD أقل/أكثر من الإجمالي:** تسجيل الفرق ومعالجته في التسوية.
* **استرداد أعلى من المبلغ المحصَّل:** منع وإظهار رسالة واضحة.
* **عملة مختلفة:** تثبيت الصرف وتوحيد التقارير بعملة الأساس.
* **تجزئة الطلب لشحنتين:** تحصيل جزئي مقابل كل شحنة (أونلاين) أو تحصيل COD لكل شحنة.

## **12) الامتثال والأمان**

* **PCI DSS:** عدم تخزين بيانات بطاقات خام؛ الاعتماد على **Tokenization**.
* **3-D Secure / SCA:** تفعيل حيث تتطلبه الأنظمة.
* **صلاحيات الوصول (RBAC):** فصل أدوار الدعم/المالية/التطوير.
* **سجلات تدقيق:** لعمليات الدفع/الاسترداد/تغيير الحالات (غير قابلة للتعديل).
* **حماية الاحتيال:** قواعد للطلبات عالية المخاطر، وحدود للمحاولات، وقوائم منع.

## **13) خارطة الإصدارات**

* **MVP:**
  + أونلاين: تحصيل فوري + Webhook موثوق + استرداد كامل.
  + COD: تدفّق كامل حتى remitted مع ملف تسوية أسبوعي.
  + تقارير أساسية (نجاح/فشل/تحصيل/توريد).
* **المرحلة 2:**
  + تفويض ثم تحصيل، استرداد جزئي، اختلاف عملات، تحصيل جزئي للشحنات المجزأة.
  + لوحات تسوية متقدمة وفروقات التوريد.
* **المرحلة 3:**
  + BNPL، نزاعات/Chargebacks مدمجة، كشف احتيال أعمق، أتمتة كاملة للتطابق المحاسبي.

**الخلاصة:** بهذه المتطلبات يصبح نظام الدفع لديك **متوافقًا مع المعايير**، يدعم واقع السوق (COD) دون التضحية بالدقة المحاسبية، ويُحسّن معدل التحويل مع حماية من الفشل والاحتيال—مع تقارير تسوية واضحة يمكن الوثوق بها.

الشحن (Shipping) — المتطلبات المعتمدة

# **الشحن (Shipping)**

## **1) الهدف**

* إدارة الشحن من **التجهيز** حتى **التسليم أو الإرجاع** بدقة وشفافية.
* دعم واقع **الدفع عند الاستلام (COD)** في اليمن مع تقليل فشل التوصيل.
* توحيد تتبّع الشحنات عبر شركات مختلفة وإظهار حالة واضحة للعميلة.

## **2) النطاق**

* إنشاء شحنة/شحنات للطلب (يدعم **تجزئة الطلب**).
* اختيار شركة الشحن وخدمة التوصيل (نفس اليوم/اليوم التالي/اعتيادي).
* طباعة/توليد **رقم بوليصة (AWB)** وملصق شحن.
* تتبّع الحالات وتلقي التحديثات من شركة الشحن (Webhook/CSV/API).
* محاولات التسليم، الفشل، **العودة إلى المرسل (RTO)**، والاستلام عند المرتجع.
* تقدير تكلفة الشحن أثناء السلة/الدفع حسب **المنطقة/الوزن**.
* ربط تحصيل **COD** بالشحنة، مع تمرير التوريدات (remittance) للمالية.

## **3) الكيانات (مفهومية—بدون أكواد)**

* **Shipment (الشحنة):** مرتبطة بطلب واحد وقد تكون هناك عدة شحنات للطلب.
* **Shipment Items (بنود الشحنة):** (اختياري) لربط أي بنود تم شحنها في شحنة معيّنة.
* **Courier (شركة الشحن):** تعريف الشركة وخدماتها ودعمها لـ COD.
* **Service Level (مستوى الخدمة):** نفس اليوم/اليوم التالي/اعتيادي وساعات العمل.
* **Zones & Rates (المناطق/التسعير):** تعريف مناطق المدن/الأحياء وجداول الأسعار حسب الوزن/الشريحة ورسوم COD/الوقود.
* **Tracking Events (أحداث التتبع):** طبقة توحيد لحالات الشحن من مختلف الشركات.
* **Return/RTO (الإرجاع/العودة):** تدفق إعادة القطع للمستودع والتقييم.

## **4) الحقول الرئيسية المطلوبة**

### **4.1 الشحنة (Shipment)**

* **المعرفات:** shipment\_id، order\_id، warehouse/location\_id، courier\_id، service\_level، **tracking\_no/AWB**.
* **الحالة الموحّدة:** ready → shipped → out\_for\_delivery → delivered  
  مسارات جانبية: failed\_delivery, rto\_in\_transit, rto\_received, cancelled.
* **التواريخ:** created\_at، ready\_at (جاهزة للتسليم للشركة)، shipped\_at، out\_for\_delivery\_at، delivered\_at، failed\_at، rto\_received\_at.
* **محاولات التسليم:** delivery\_attempts (عدد/تواريخ)، last\_attempt\_reason (لا يرد/عنوان غير واضح…).
* **العنوان الملتقط:** اسم/هاتف/مدينة/حي/علامة مميزة (لقطة من الطلب).
* **الأوزان/الأبعاد:** weight\_actual، weight\_dimensional، **chargeable\_weight** (المحاسبي).
* **التكلفة:** base\_rate، fuel\_surcharge، cod\_fee (إن وُجد)، other\_fees، **shipping\_cost\_total** (للتقارير).
* **COD:** cod\_amount\_expected، cod\_amount\_collected (يُحدّث عند التسليم)، remittance\_batch\_ref (يرتبط بالمالية).
* **الملاحظات التشغيلية:** handover\_method (pickup/dropoff)، packaging\_notes، courier\_notes، internal\_notes.

### **4.2 شركة الشحن (Courier)**

* **الهوية:** name، contact، supports\_cod (نعم/لا)، المدن المغطّاة، ساعات العمل/العطل.
* **الخدمات:** service\_levels المتاحة وتعريف SLA لكل مستوى.
* **التكامل:** نوع التكامل (API/Webhook/CSV يدوي)، مفاتيح/إعدادات (تُخزّن في إعدادات تكامل، لا في جداول عامة).

### **4.3 التسعير والمناطق (Zones & Rates)**

* **المنطقة:** country/region/city/district mapping (قوائم مضبوطة).
* **جداول السعر:** شريحة وزن (0–1كغ، 1–2كغ…)، سعر أساسي، زيادة لكل كغ، رسوم COD، رسوم نائية/بعيدة (remote area).

### **4.4 أحداث التتبع (Tracking Events)**

* **مصدر الحدث:** courier، raw\_status.
* **الحالة الموحّدة:** mapped\_status (ready/shipped/out\_for\_delivery/delivered/failed\_delivery/rto…).
* **الوقت/الموقع:** event\_at، event\_city، notes.

## **5) قواعد العمل**

* **اختيار شركة الشحن تلقائيًا** حسب: المدينة/الحي، دعم COD، **سعر أقل** + **نسبة نجاح** أعلى (معدّل التسليم من البيانات التاريخية)، و**SLA**.
* **تجزئة الشحنات:** إذا بند غير جاهز/غير متوفر، يُشحن الباقي وتُنشأ شحنة ثانية لاحقًا.
* **تطابق COD:** مبلغ COD في الشحنة = إجمالي الشحنة (أو حصة من الطلب عند التجزئة).
* **محاولات التسليم:** حد افتراضي 2–3 محاولات؛ بعد آخر محاولة → **RTO** ما لم يطلب العميل إعادة الجدولة.
* **تعديل العنوان:** مسموح قبل shipped. بعد الشحن: طلب تعديل عبر شركة الشحن وتوثيق السبب.
* **وزن محاسبي:** إن غيّرت الشركة الوزن عند الفوترة، نسجّل **chargeable\_weight** وفارق التكلفة.
* **الإشعارات:** آليًا عند shipped، out\_for\_delivery، delivered، failed\_delivery، مع **روابط تتبع** ووقت تسليم تقريبي.
* **RTO/Returns:** عند الاستلام في المستودع: فحص الحالة → قرار **إرجاع مخزون** أو **تخفيض/تلف**.

## **6) التدفقات التشغيلية**

### **6.1 تنفيذ الطلب (Fulfillment)**

1. **Pick & Pack**: اختيار وتجهيز الأصناف.
2. **توليد AWB** وطباعة الملصق.
3. **Ready → Shipped**: تسليم للشركة وتأكيد الاستلام.
4. **Out for Delivery**: تحديث يوم التسليم.
5. **Delivered / Failed**: تحديث الحالة والـCOD.

### **6.2 فشل التسليم**

* المحاولة 1: لا يرد/عنوان غير واضح → تواصل واتساب/اتصال وتحديث العنوان إن لزم.
* المحاولة 2: جدولة جديدة.
* المحاولة 3: إن فشلت → **RTO** مع سبب واضح.

### **6.3 RTO (عودة إلى المرسل)**

* عند وصول **rto\_received**:
  + تقييم الصندوق/القطعة.
  + تحديث المخزون (سليم/معيب).
  + تحديث ملف مخاطر العميل (لـCOD) وكتابة سبب الفشل.

### **6.4 المرتجعات (بعد التسليم)**

* **Pick-up** من العميل (إن كانت سياسة تسمح بذلك) أو التسليم في نقطة محددة.
* إنشاء **شحنة مرتجع** مرتبطة بالطلب الأصلي.

## **7) تقدير تكلفة الشحن (Cart/Checkout)**

* يعتمد على: **المدينة/الحي** + **الوزن المحسوب** (فعلي/حجمي) + شركة الشحن المقترحة.
* عرض **سعر دقيق** إن توفّر العنوان؛ خلاف ذلك **نطاق سعري**.
* إظهار **رسوم COD** إن طُبقت قبل تأكيد الطلب.

## **8) التكاملات**

* **شركات الشحن:**
  + API لإنشاء الشحنة/الملصق/التتبّع (أولوية).
  + **Webhook** لاستلام تحديثات الحالة.
  + بديل: **استيراد CSV** يومي للحالات إن لم تتوفر Webhooks.
* **المدفوعات:** تمرير cod\_amount\_collected عند delivered وربطه بـ **دفعة التوريد** (remittance).
* **الطلبات/المخزون:** تحديث الحالة والمخزون تلقائيًا عند الشحن/العودة.
* **الإشعارات:** Push/SMS/WhatsApp مع **Deep Links** لصفحة التتبع.
* **لوحة العمليات:** لوحة “Dispath/Tracking” لمتابعة الشحنات حسب الحالة/الشركة/المدينة.

## **9) مؤشرات الأداء (KPIs)**

* **First-Attempt Delivery Rate** (نسبة التسليم من أول محاولة).
* **On-Time Delivery** مقابل SLA لكل شركة/مدينة.
* **RTO Rate** وأسباب الفشل (لا يرد/عنوان/سعر COD…).
* **COD Collection Rate** ومعدل التوريد ≤ 7 أيام.
* **متوسط زمن التسليم** (من shipped إلى delivered).
* **تكلفة الشحن لكل طلب** وفروق الوزن المحاسبي.
* **نسبة الشحنات بلا تتبّع حديث** (>24 ساعة) لاتخاذ إجراء.

## **10) معايير القبول (Definition of Done)**

* إنشاء شحنة مع **AWB** وخدمة/شركة واضحة، وطباعة ملصق.
* تتبّع محدث (API/Webhook/CSV) ينعكس على الحالة الموحّدة داخل النظام.
* محاولات التسليم تُسجَّل مع الأسباب.
* RTO يعمل: استقبال، تقييم، تحديث مخزون وحالة الطلب.
* تقدير الشحن يظهر في السلة/الدفع بدقة عند توافر العنوان.
* إشعارات العميل تعمل بالحالات الرئيسية مع روابط تتبّع.
* تكاليف الشحن تُسجَّل وتظهر في التقارير، مع معالجة فروق الوزن.
* ربط **COD** بالشحنات وتغذية **دفعات التوريد** للمالية.

## **11) سيناريوهات حافة**

* **تجزئة الشحنة**: بنود تُشحن لاحقًا، مع COD موزّع.
* **منطقة نائية/غير مغطّاة**: اختيار شركة بديلة أو عرض خيار “استلام من نقطة”.
* **زيادة وزن محاسبي** بعد الفوترة: تسجيل الفرق وتمريره للتسوية.
* **عطلات/طقس**: تمديد SLA وتحديث إشعار العميل.
* **عنوان غير قياسي**: اعتماد **حقل “علامة قريبة”** وإرسال موقع/لوكيشن عند الإمكان.
* **انقطاع تكامل الشركة**: التحوّل إلى CSV مؤقتًا دون فقد تتبّع.

## **12) خارطة الإصدارات**

* **MVP:** شحنة واحدة لكل طلب، شركة واحدة مدمجة API، تتبّع أساسي، محاولتان للتسليم، RTO يدوي، تقدير شحن بالمدينة، إشعارات shipped/out\_for\_delivery/delivered.
* **المرحلة 2:** تعدد الشركات والاختيار الآلي، تجزئة الشحنات، Webhooks متعددة، لوحات تشغيل، RTO مؤتمت، رسوم COD مرِنة.
* **المرحلة 3:** تحسين توجيه الشحن (Routing) بالبيانات التاريخية، نوافذ تسليم مواعيد (Time Slots)، نقاط التسلّم/الإرجاع، تسعير ديناميكي.

## **13) ملاحظات تشغيلية للسوق المحلي**

* الاعتماد على **المدينة/الحي/المعلم** بدل الرمز البريدي، وتأكيد العنوان عبر واتساب قبل الشحن عالي القيمة.
* تفعيل **OTP للتسليم** في COD للحماية من تسليم خاطئ.
* بناء **قائمة مناطق** موحّدة داخليًا لتقليل فشل العنوان، مع تحديث دوري وفق تجربة الميدان.

**الخلاصة:** بهذه المتطلبات يصبح مسار الشحن واضحًا ومضبوطًا من التجهيز إلى التسليم أو العودة، مع تكامل قوي لـ COD، وتتبع موحّد، واختيار شركة شحن ذكي—ينعكس مباشرة على ارتفاع نسبة التسليم من أول محاولة وخفض معدّل RTO وتكاليف الشحن.

المرتجعات والاستبدالات RMA(نقطة مشتركة)

# **RMA — المرتجعات والاستبدالات (نقطة مشتركة)**

## **1) الهدف**

* إدارة كل حالات ما بعد البيع في مسار واحد موحّد: **استرداد** أو **استبدال**.
* تقليل زمن المعالجة ورفع رضى العميلة مع تتبّع واضح وشفاف.
* ضبط الأثر المالي والمخزني بدقة (خصوصًا مع **COD**).

## **2) النطاق**

* فتح طلب RMA بسبب محدد (مقاس/عيب صناعة/تأخر/تغيير رأي…).
* خياران للحل: **Refund** (استرداد) أو **Exchange** (استبدال).
* لوجستيات الإرجاع: استلام من العميلة/تسليم لنقطة/ترتيب شركة شحن.
* فحص القطعة وتحديد التصرف المخزني (يرجع للمخزون/تخفيض/تالف).
* التكامل مع: **الطلبات، المدفوعات، الشحن، المخزون، الإشعارات، التقارير**.

## **3) السجلات (مفهومية — بدون أكواد)**

* **RMA**: رقم، طلب أصلي، عميلة، قناة، سبب، حالة، تواريخ رئيسية.
* **RMA Items**: البنود والكميات، حالة العنصر عند الاستلام، قرار التصرف.
* **Resolution**: نوع الحل (Refund / Exchange)، الملاحظات.
* **Logistics**: طريقة الاستلام/التسليم، شركة الشحن، رقم التتبع، محاولات الاستلام.
* **Refund**: المبلغ، الطريقة (بوابة/تحويل/نقد/رصيد متجر)، التواريخ.
* **Exchange**: **طلب استبدال جديد** مرتبط، فروق السعر، رسوم شحن جديدة.
* **Reason Codes**: قاموس موحّد للأسباب (مقاس/عيب/تأخر/…).

## **4) قواعد العمل (Business Rules)**

* **الأهلية**: ضمن نافذة زمنية (مثلاً 30ساعة من التسليم)، مع شرط الحالة (غير مستخدم/بطاقته مثبتة… حسب السياسة).
* **الاسترداد**:
  + الأولوية لـ **رصيد متجر** (يعجّل الإغلاق ويقلّل السيولة الخارجة)، ويُتاح نقد/تحويل/بوابة عند الحاجة.
  + الجزئي/الكامل حسب البنود المقبولة.
* **الاستبدال**:
  + إنشاء **طلب جديد** للبند البديل، وحساب **فرق السعر** (تحصيل/رد).
  + رسوم الشحن للاستبدال حسب السياسة (مجاني لعيب الصناعة، مدفوع لتغيير الرأي…).
* **المخزون**:
  + عند الاستلام والفحص: تحديد مصير كل بند (**يعود للمخزون** إذا سليم، **تخفيض** إذا عبوة مفتوحة، **تالف** إذا غير قابل للبيع).
* **COD**:
  + عند الاسترداد: **تحويل/نقد/رصيد متجر**؛ ربط أي مبالغ مع **تسويات شركة الشحن** إذا كان التحصيل عبرهم.
* **الإشعارات**: تلقائية في كل مرحلة (قبول/رفض/في الطريق/استلمنا/تم الفحص/تم الاسترداد/أُنشئ الاستبدال…).
* **مكافحة سوء الاستخدام**: حد أقصى لطلبات RMA لكل عميلة/شهر، ورفع **درجة المخاطر** عند نمط مسيء.

## **5) سير العمل الموحّد**

1. **فتح RMA**: اختيار السبب والمسار (استرداد/استبدال) والتحقق من الأهلية.
2. **اللوجستيات**: تحديد طريقة الإرجاع (استلام/تسليم لنقطة/شركة شحن) وإنشاء رقم تتبع عند الحاجة.
3. **الاستلام والفحص**: تسجيل الحالة، تحديد التصرف المخزني.
4. **القرار**:
   * **Refund**: تنفيذ الاسترداد بالطريقة المناسبة وتأكيده.
   * **Exchange**: إنشاء طلب استبدال، معالجة فرق السعر، وترتيب شحن جديد.
5. **الإغلاق**: تحديث التقارير/المخزون/ملف العميل، وإرسال إشعار نهائي.

## **6) الحالات والانتقالات (Statuses)**

requested → approved → in\_transit\_return → received → inspected → resolved (refunded / exchanged) → closed  
مسارات جانبية: rejected (غير مؤهل)، no\_item\_received، partial.

## **7) التكاملات**

* **الطلبات**: ربط RMA بالطلب الأصلي وبنوده.
* **المدفوعات**: تنفيذ **Refund** (بوابة/تحويل) أو إضافة **Store Credit**.
* **الشحن**: شحنة مرتجع مع تتبّع موحّد.
* **المخزون**: تحديث الكمية/الحالة حسب قرار الفحص.
* **الإشعارات**: Push/SMS/WhatsApp وفق موافقات القنوات.
* **التقارير**: معدلات RMA، الأسباب، زمن المعالجة، أثرها على العائدات.

## **8) مؤشرات الأداء (KPIs)**

* **Return Rate** و**Exchange Rate** حسب الفئة/المقاس/المدينة.
* **متوسط زمن المعالجة** من الطلب إلى الإغلاق.
* **نسبة الإغلاق ضمن SLA** (مثلاً ≤ 7 أيام).
* **أكثر الأسباب تكرارًا** (لتعديل وصف/مقاسات/جودة).
* **نسبة الاسترداد كرصيد متجر** مقابل نقد/بوابة.
* **أثر RMA** على هامش الربح والمخزون.

## **9) معايير القبول (Definition of Done)**

* فتح RMA بمسارَيه (Refund/Exchange) مع تحقّق أهلية واضح.
* لوجستيات مرتجع تعمل وتتبعها ظاهر للعميلة.
* فحص دقيق وتصرّف مخزني موثّق لكل بند.
* تنفيذ الاسترداد أو إنشاء طلب الاستبدال ومعالجة الفروق.
* إشعارات في كل محطة رئيسية.
* تقارير تشغيلية جاهزة تغطي KPIs أعلاه.

## **10) سيناريوهات حافة**

* **استلام جزئي**: قبول بعض البنود ورفض أخرى.
* **مرتجع بلا قطعة**: إغلاق no\_item\_received بعد مهلة/تحقيق.
* **استبدال بديل غير متوفر**: اقتراح بديل/رصيد متجر.
* **فرق COD**: تحصيل/رد فرق السعر في الاستبدال مع توثيق.
* **ضرر من الشحن**: تصنيف كعيب صناعة وتحمّل شركة الشحن/المورّد حسب السياسة.
* **عنوان مرتجع غير صحيح**: إعادة جدولة أو تحويل لنقطة استلام.

## **11) خارطة الإصدارات**

* **MVP**: RMA موحّد، استلام مرتجع يدوي/شركة شحن واحدة، فحص وتصرف مخزني أساسي، Refund رصيد متجر أو بوابة، Exchange بسيط، إشعارات أساسية، تقارير أولية.
* **المرحلة 2**: شركات شحن متعددة، سياسات متقدّمة حسب السبب/الفئة، فروق سعر تلقائية، لوحات متابعة SLA، أتمتة رفض/قبول وفق الصور.
* **المرحلة 3**: توصيات ذكية لتقليل المرتجعات (مقاسات/بدائل)، تحليلات توقعية للأسباب، أتمتة استرداد لحظي.

## **12) ملاحظات للسوق المحلي (خصوصًا COD)**

* تفضيل **رصيد متجر** في الاسترداد لتسريع إغلاق الحالات.
* عند فشل تسليم COD ثم RTO: تحديث **مخاطر العميل** لمنع تكرار الهدر.
* توثيق **علامة قريبة** بمدينة/حي لتسهيل استلام المرتجع.

**الخلاصة:** نقطة RMA الموحّدة تبسّط العمل (واجهات وعمليات وتقارير)، وتجمع 80% من البيانات المشتركة بين الاسترداد والاستبدال في مسار واضح واحد—فتقلّ الأخطاء، ينخفض الزمن، وتتحسّن تجربة العميلة والأثر المالي مع المحافظة على دقة المخزون.

العروض والخصومات — المتطلبات المعتمدة

# **العروض والخصومات**

## **1) الهدف**

* تحفيز الشراء وزيادة متوسط قيمة الطلب (AOV) والعودة للشراء.
* تقديم خصومات دقيقة وعادلة دون تعقيد على المخزون أو المحاسبة.
* قياس أثر كل عرض على المبيعات والهامش بدقة.

## **2) النطاق**

* **أكواد خصم (Promo Codes/Coupons)** تُدخلها العميلة يدويًا.
* **عروض تلقائية (Catalog/Cart Rules)** تُطبَّق دون كود (مثال: خصم على فئة معينة).
* **شحن مجاني** بشرط حد أدنى أو ضمن حملات.
* (لاحقًا) **اشترِ X واحصل على Y**، **Bundles**، **خصومات طبقية**.

## **3) أنواع العروض المدعومة (المرحلة الأولى)**

* **نسبة مئوية** (% على الطلب أو على أصناف مؤهلة).
* **قيمة ثابتة** (مبلغ يُخصم من الإجمالي أو من أصناف مؤهلة).
* **شحن مجاني** (كلي/جزئي أو سقف محدد).
* **تحديد النطاق**: على **الطلب كاملًا** أو على **بنود/تصنيفات/علامات تجارية** محددة.

## **4) التعريفات والحقول (مستوى المفهوم)**

### **أ) الحملة/الكوبون**

* الاسم، **الرمز** (لأكواد الخصم)، نوع الخصم وقيمته.
* **الجدولة**: تاريخ/وقت بداية ونهاية + المنطقة الزمنية.
* **الشروط**: حد أدنى للطلب، عملة، مدينة/منطقة، طريقة دفع (يمكن **استثناء COD** أو فرض رسوم COD بلا خصم)، قناة (App/Instagram/WhatsApp)، جهاز/إصدار تطبيق (اختياري).
* **النطاق**: يشمل/يستبعد فئات، منتجات، علامات تجارية، **منتجات مخفّضة مسبقًا** (تحديد السياسة).
* **الاستحقاق**: عميلة جديدة فقط/كل العملاء، شرائح محددة (VIP، غير نشطين…).
* **التراكم (Stacking)**: يسمح/لا يسمح بالجمع مع عروض أخرى؛ **أولوية/ترتيب** التطبيق عند التعارض.
* **حدود الاستخدام**: إجمالي مرات الاستخدام، **لكل عميلة**، لكل يوم، لكل قناة.
* **سقوف**: حد أقصى للخصم في الطلب، وحد أقصى **للصنف** إن كان بنديًّا.
* **الرسائل**: نص نجاح/رفض واضح (سبب الرفض)، ملاحظات قانونية مختصرة.
* **التفعيل**: حالة (مسودة/منشور/موقوف)، مالك الحملة.

### **ب) عمليات الاستخدام (Redemptions)**

* عميلة، طلب، قيمة الخصم النهائية، وقت الاستخدام، نتيجة (نجح/فشل) + السبب، القناة/المصدر (UTM إن وجد).

## **5) قواعد العمل (Business Rules)**

* **حساب الأهلية قبل الحساب**: يتحقق النظام من الشروط أولًا، ثم يطبّق الخصم على **الأصناف المؤهلة فقط**.
* **الأولوية عند التعارض**:
  + القاعدة: **“أفضل سعر للعميلة”** أو **“غير قابل للتجميع”** وفق إعداد الحملة.
  + إن كان التجميع ممنوعًا: يُطبّق العرض صاحب **الأولوية الأعلى** أو **الخصم الأكبر** حسب السياسة.
* **منتجات مخفّضة مسبقًا**: إمّا تُستثنى أو يسمح بتراكم حتى سقف محدد (سياسة واضحة وثابتة).
* **الشحن وCOD**: الخصم لا يُطبَّق على **رسوم COD** افتراضيًا؛ الشحن مجاني قد يلغي رسوم الشحن فقط.
* **التقريب والعملة**: قواعد تقريب موحّدة **للمجموع الكلي** وبنود السطر؛ نفس القواعد عند الاسترداد.
* **الالتقاط في الطلب**: تُحفَظ قيمة الخصم النهائية موزَّعة **بنِسَب** على بنود الطلب (لتسهيل المرتجعات).
* **الإرجاع/الاستبدال**: خصم **نسبي (pro-rata)** يُستعاد مع البند المرتجع فقط؛ لا تُرجَع قيمة خصم تخص بنود لم تُرجَع.
* **منع إساءة الاستخدام**: حد محاولات إدخال كود، منع مشاركة أكواد خاصة، ربط الأكواد الفردية بالعميلة عند اللزوم.

## **6) التدفقات التشغيلية**

1. **تطبيق الكود في السلة** → تحقق شروط/تعارض → رسالة فورية (نجاح/سبب رفض).
2. **عروض تلقائية** تظهر فور توفر الشروط (مثال: “أضف 10,000 YER لتحصل على شحن مجاني”).
3. **الدفع**: إعادة تحقق نهائي قبل إنشاء الطلب (السعر/المخزون/الخصم).
4. **الطلب**: التقاط الخصم وتوزيعه على البنود + تسجيل **redeption**.
5. **المرتجعات**: حساب الاسترداد مع خصم نسبي ومزامنة مع المدفوعات/رصيد المتجر.

## **7) الدمج والتكامل**

* **السلة/الدفع**: تحقق لحظي، إعادة حساب المجاميع، رسائل واضحة.
* **الطلبات**: لقطة نهائية للخصم وتوزيع بنود.
* **المرتجعات**: سياسة pro-rata تلقائية.
* **المدفوعات**: الشحن/COD لا يتأثران إلا إذا نص العرض.
* **التحليلات (GA4/Firebase)**: أحداث apply\_promo, promo\_error, view\_promo\_banner, purchase مع قيم الخصم.
* **القنوات**: إمكانية تقييد كود لقناة أو حملة (UTM) محددة.

## **8) التقارير وKPIs**

* **معدل استخدام الكوبونات** وعدد الطلبات المتأثرة.
* **الخصم/الإيراد**: إجمالي الخصم، **نسبة الخصم إلى المبيعات**.
* **الأثر الصافي**: الرفع في التحويل/AOV مقابل تكلفة الخصم.
* **التسريب**: نسبة محاولات الاستخدام الفاشلة وأسبابها.
* **الأداء حسب الشريحة/القناة/الفئة**.
* **تكلفة الشحن المجاني** وتأثيره على التحويل.

## **9) معايير القبول (Definition of Done)**

* إنشاء وتفعيل **حملة/كوبون** مع الشروط والحدود والجدولة.
* تطبيق الخصم بدقة على الأصناف المؤهلة وإظهار السبب عند الرفض.
* سياسة **التراكم/الأولوية** تعمل كما هو محدد.
* توزيع الخصم على بنود الطلب محفوظ؛ المرتجعات تُحسب **pro-rata** تلقائيًا.
* الشحن المجاني و/أو استثناء COD يعملان كما في السياسة.
* تقارير الاستخدام والأثر متاحة للفريق.
* رسائل العميلة واضحة في كل خطوة (نجاح/سبب الرفض/المتبقي للوصول للعرض).

## **10) سيناريوهات حافة**

* انتهاء صلاحية الكود أثناء الدفع → رسالة واضحة مع بدائل.
* اختلاف العملة/السوق → الكود صالح لأسواق محددة فقط.
* كود **“عميلة جديدة فقط”** مع رقم/جهاز مكرر → رفض برسالة واضحة.
* سلة فيها منتج مؤرشف/نفد → إعادة تحقق وإزالة الخصم المطبق عليه فقط.
* **BOGO/Bundle** (لاحقًا): التعامل مع المرتجعات بحيث يُعاد حساب خصم المجموعة عند فقدان أهلية البند المجاني.
* أكواد شخصيّة (1-time codes) → ترتبط بحساب العميلة وتُبطل بعد الاستخدام.

## **11) خارطة الإصدارات**

* **MVP**:
  + أكواد خصم (٪/قيمة ثابتة) + شحن مجاني، حد أدنى للطلب، قيود فئة/منتج، حدود استخدام (إجمالي/لكل عميلة)، جدولة، استثناء COD اختياري، تقرير استخدام أساسي.
* **المرحلة 2**:
  + عروض تلقائية ذكية، تلميحات تقدُّمية (“تبقى 3,000 YER للشحن المجاني”), أكواد فردية، شرائح عملاء، قنوات/UTM، A/B لرسائل الخصم.
* **المرحلة 3**:
  + **BOGO**، **Bundles**، **Tiered Discounts**، تسعير متعدد الأسواق، توصيات عرض شخصية.

## **12) ملاحظات للسوق المحلي**

* التركيز على **حد أدنى للشحن المجاني** يرفع AOV بكلفة مفهومة.
* غالبًا **استثناء رسوم COD** من الخصم لتفادي تكاليف زائدة غير متوقعة.
* رسائل عربية بسيطة وواضحة على سبب الرفض والكيفية للوصول للعرض.

**الخلاصة:** بهذه المتطلبات تحصل على نظام عروض مرن وواضح، يطبّق خصومات عادلة بلا تعارض، ويُظهر أثرًا يمكن قياسه—مع جاهزية للتوسّع لاحقًا إلى BOGO والباقات والشرائح المتقدمة، دون تعقيد على الطلبات أو المحاسبة.

المشتريات والتوريد (Procurement & Inbound)

# **المشتريات والتوريد (Procurement & Inbound)**

## **1) الهدف**

* تأمين توفر المنتجات بالكمّيّات والسعر الصحيحين وفي الوقت المناسب.
* تتبّع التوريد من **أمر الشراء** حتى **الاستلام والفحص والإدخال للمخزون**.
* احتساب **التكلفة الحقيقية (Landed Cost)** بدقّة لرفع هامش الربح واتخاذ قرار تسعير صحيح.
* قياس أداء المورّدين وتقليل زمن النفاد (Stockouts) ومرتجعات المورد.

## **2) النطاق**

* إدارة **المورّدين** وشروطهم (عملات/مدد توريد/حدود دنيا).
* **أوامر الشراء (PO)** وبنودها، والموافقات.
* **شحنات واردة** وإشعارات شحن مسبقة (ASN) وتتبع الوصول.
* **الاستلام (GRN)** والفحص والجودة ومعالجة الفروقات (نقص/زيادة/ضرر).
* **تكاليف إضافية**: شحن/تأمين/جمارك وتوزيعها على البنود (Landed Cost).
* **فواتير المورد** والمطابقة الثلاثية (PO ↔ GRN ↔ Invoice).
* **مرتجعات المورد (RTV)** ومذكرات الخصم (Debit Notes).
* **التخطيط وإعادة الطلب** (Reorder Rules) بصورة خفيفة.

## **3) الكيانات (مفهومية – بدون تفاصيل تقنية)**

* **Supplier (مورّد):** الاسم، جهة الاتصال، شروط الدفع (مقدم/NET 30)، العملة، **مدة التوريد (Lead Time)**، الحد الأدنى للطلب (MOQ)، شروط الشحن/إنكوترمز (Incoterms).
* **Purchase Order (PO):** رقم، مورد، عملة، تواريخ (الإنشاء/الموافقة/المتوقّع)، حالة (مسودة/موافق/جزئي/مكتمل/ملغى).
* **PO Items:** الصنف/النسخة، الكمية، **سعر الشراء**، الخصم، موعد التوريد المتوقّع لكل بند.
* **Inbound Shipment / ASN:** شركة الشحن، رقم بوليصة، حاوية/شحنة، **ETA/ETD**، ملاحظات جمركية.
* **GRN (إثبات استلام):** الكمية المستلمة فعليًا، الفروقات، حالة الجودة.
* **QC (الجودة):** نتيجة الفحص (سليم/عيب/تالف)، قرارات التصرف.
* **Landed Cost:** شحن/تأمين/جمارك/مناولة… وطريقة توزيعها (بالوزن/الحجم/القيمة/الكمية).
* **Supplier Invoice:** رقم، تاريخ، عملة، مبلغ، حالة (مُطابَق/فروقات).
* **RTV (Return to Vendor):** سبب الإرجاع، الكميات، مذكّرة خصم، شحنة عكسية.
* **Reorder Rules:** نقطة إعادة الطلب، كمية الطلب الاقتصادية، الحد الأدنى للمخزون.

## **4) الحقول عالية المستوى التي نحتاجها**

* **Supplier:** تقييم أداء، نسبة الالتزام بالمواعيد، نسبة العيوب، متوسط مهلة التوريد الفعلي.
* **PO:** إجمالي الأمر بالعملة، سعر الصرف المثبّت لحظة الإنشاء/الفاتورة، **حالة الموافقة**، مرفقات (عرض سعر/عقد).
* **PO Item:** SKU/Variant، الكمية المطلوبة/المستلمة/المتبقية، **سعر الشراء** قبل وبعد الهبوط/الخصم.
* **Shipment/ASN:** شركة الشحن، طريقة الشحن (جوي/بحري/بري)، الوزن/الحجم، رقم تتبع، ETA.
* **GRN/QC:** تاريخ الاستلام، مستلم، نتائج الفحص، **فروقات** (كمّية/جودة)، قرارات (إدخال مخزون/تخفيض/تلف/RTV).
* **Landed Cost:** قائمة التكاليف + **طريقة التوزيع** ونتيجة التكلفة للوحدة.
* **Invoice:** مبلغ الفاتورة، ضرائب/رسوم، **مطابقة ثلاثية** (تطابق السعر/الكمية)، فروقات السعر/الكمية.
* **RTV:** سبب، كميات، قيمة، طريقة الشحن العكسي، حالة الإغلاق.
* **Reorder:** نقطة إعادة الطلب، **Lead Time**، **Safety Stock**.

## **5) قواعد العمل (Business Rules)**

* **الموافقات على PO:** أوامر فوق حدّ مالي معيّن تتطلب موافقة ثنائية.
* **المُبالغة/النقص (Tolerance):** سماحية ±5–10% لكل بند تُقبل تلقائيًا، وما زاد يذهب لمراجعة.
* **الاستلام الجزئي:** يسمح باستلام أجزاء من الأمر؛ **تبقى الكمية المتبقية مفتوحة** أو تُغلق حسب قرار الشراء.
* **مطابقة ثلاثية (3-Way Match):** لا تُعتمد فاتورة إلا إن طابقت **كمّية وسعر** (PO/GRN/Invoice) أو فُسِّرَ الفرق.
* **تثبيت سعر الصرف:** يُثبَّت عند الفاتورة أو عند الدفع (سياسة موحّدة) ويُحفظ المصدر.
* **التكلفة القياسية:** طريقة التقييم: **متوسط متحرّك (Moving Average)** أو **FIFO** – تُعتمد واحدة وتوثَّق.
* **Landed Cost:** تُوزَّع قبل إدخال التكلفة النهائية للمخزون؛ أي تعديل لاحق يُسجَّل كتعديل تكلفة.
* **الجودة/العيوب:** المعيب لا يدخل مخزون البيع؛ إمّا **RTV** أو تحويله لتخفيض/تالف.
* **RTV:** يُنشأ لأصناف ناقصة/معيبة؛ تُصدر **مذكرة خصم** أو استبدال.
* **إعادة الطلب:** تنبيه تلقائي عندما يصل المخزون إلى **Reorder Point** آخذًا بالاعتبار **Lead Time** والطلب التاريخي.

## **6) التدفقات التشغيلية (Workflows)**

### **6.1 تعاقد/إضافة مورّد**

إدخال بيانات المورّد + شروط الدفع/العملة/الـIncoterms → توثيق **Lead Time** وMOQ → تقييم ابتدائي.

### **6.2 إنشاء PO والموافقة**

مشتريات تجمع بنود مطلوبة (احتياج/تخطيط/حملات) → مراجعة الأسعار والخصومات → **موافقة** حسب الصلاحيات → إرسال للمورّد.

### **6.3 شحنة واردة & ASN**

استلام ASN (رقم بوليصة/حاوية، ETA، محتويات) → ربطها بالـPO → تجهيز الاستلام.

### **6.4 الاستلام والفحص (GRN & QC)**

وصول الشحنة → عدّ فعلي لكل بند → تسجيل **GRN** والفروقات → **QC**: سليم/عيب/تالف → قرارات (إدخال/RTV/تخفيض).

### **6.5 التكاليف الإضافية & Landed Cost**

تجميع الشحن/التأمين/الجمارك/المناولة → توزيع على البنود بطريقة مختارة (قيمة/وزن/حجم/كمية) → تحديث **تكلفة الوحدة** الفعلية.

### **6.6 فاتورة المورد والمطابقة**

استلام الفاتورة → **مطابقة ثلاثية** مع PO وGRN → قبول/رفض/استثناء مع توثيق → تمرير للحسابات الدائنة.

### **6.7 مرتجع للمورد (RTV)**

إن وُجد عيب/فائض غير مقبول → فتح RTV مع سبب → شحنة عكسية → **مذكرة خصم** أو استبدال.

### **6.8 إعادة الطلب (Reorder)**

مراقبة مخزون SKU مقابل **Reorder Point** → توليد اقتراح PO → مراجعة وتوحيد بنود وفق MOQ/شحن → اعتماد.

## **7) التكاملات**

* **المخزون:** إدخال الكميات السليمة فور GRN، وتعديل حالة المعيب/التلف، وتحديث التكلفة (Moving Average/FIFO).
* **الشحن/التتبع (Inbound):** ربط أرقام التتبع والـETA، ودعم حالات الجمارك.
* **المالية/المحاسبة:** قيود تلقائية (مخزون، تكلفة بضائع مباعة، حسابات دائنة، فروق عملة، تكاليف شحن وجمارك)، وتسويات المورد.
* **التقارير/التحليلات:** أداء المورّدين، زمن التوريد الفعلي، انحراف التكلفة، تغطية المخزون.
* **الإشعارات:** تنبيهات تأخّر ASN/ETA، تجاوز تَحمّل الفروقات، وصول مخزون حرج.

## **8) التقارير وKPIs**

* **On-Time Delivery %** لكل مورد/فئة.
* **Lead Time Accuracy** (الفرق بين المتوقع والفعلي).
* **Fill Rate %** (نسبة التوريد مقابل المطلوب).
* **Defect Rate %** و**RTV Rate %**.
* **Cost Variance %** (فرق سعر/تكلفة هبوط/Landed Cost).
* **Days of Inventory** و**Stockout Days**.
* **Inventory Turns** و**Aging** (بضائع بطيئة الحركة).
* **Supplier Scorecard** شامل (زمن/جودة/تكلفة/تواصل).

## **9) معايير القبول (Definition of Done)**

* إنشاء **PO** وموافقته وتتبّع حالته حتى الإغلاق.
* تسجيل **ASN/Shipment** وربطها بأمر الشراء.
* **GRN** مع استلام جزئي وفروقات موثّقة ونتيجة **QC**.
* توزيع **Landed Cost** على البنود وتحديث تكلفة الوحدة.
* **مطابقة ثلاثية** تعمل وتظهر الاستثناءات بوضوح.
* إنشاء **RTV** وإغلاقه مع مذكّرة خصم أو استبدال.
* مؤشرات الأداء الأساسية ولوحة **Supplier Scorecard** متاحة.

## **10) سيناريوهات حافة**

* **Short/Over Shipment:** أقل/أكثر من المطلوب—تُسجّل وتُقرَّر (قبول/RTV/خصم).
* **ضرر أثناء النقل:** يتحملها الشحن/المورّد حسب الـIncoterms—تسجيل مطالبة.
* **رسوم جمركية غير متوقعة:** تُضاف للـLanded Cost وتوثَّق كملاحظة.
* **تقلب عملة قوي:** فروق تقييم عملة تُسجَّل محاسبيًا؛ سياسة تثبيت واضحة.
* **تجزئة PO على شحنات متعددة:** تحديث ETAs ومتابعة استلامات جزئية.
* **استبدال SKU (بديل):** قبول بديل مع ربط واضح ووسم “بديل مورّد”.

## **11) خارطة الإصدارات**

* **MVP:** مورّدون، PO + Items، استلام جزئي (GRN)، QC أساسي، RTV بسيط، تكلفة مخزون **متوسط متحرّك**، تقارير أساسية.
* **المرحلة 2:** ASN مفصّل، **Landed Cost** بتوزيع مرن، **مطابقة ثلاثية**، مؤشرات أداء الموردين، تنبيهات تأخّر.
* **المرحلة 3:** قواعد **Reorder** تلقائية، توقّعات طلب (Forecast)، عقود أسعار، مناقصات موردين، تسعير متعدد العملات.

## **12) ملاحظات للسوق المحلي (اليمن وما حولها)**

* التعامل غالبًا بعملات **USD/SAR** مع تثبيت صرف موثّق عند الفاتورة/الدفع.
* الاهتمام بنمذجة **الجمارك والرسوم** بدقة ضمن Landed Cost؛ والاحتفاظ بالمستندات.
* أولوية للشحن البري/البحري في التكاليف والـETA؛ تأكّد من دعم سيناريو **تأخر الميناء**.
* فرق COD في المرتجعات بعد التوريد لا يؤثر على **تكلفة الشراء**؛ يُعالج ضمن المبيعات/المرتجعات لا ضمن المشتريات.

بهذه المتطلبات يصبح مسار التوريد واضحًا من PO إلى الاستلام والتكلفة الفعلية، مع رقابة قوية على الجودة والتكاليف، ومطابقة مالية سليمة، ومؤشرات أداء تمكّنك من تحسين اختيار المورّدين وتقليل نفاد المخزون—وكل ذلك بدون تعقيد تقني زائد.

التحليلات (Analytics) — خلاصة تنفيذية

# **التحليلات (Analytics)**

## **1) الهدف**

* إجابة أسئلة العمل بسرعة: التحويل، الاحتفاظ، قيمة العميل، أداء المنتجات، فعالية القنوات، جودة التشغيل (شحن/COD/مرتجعات).
* توحيد مصدر الحقيقة للأرقام مع تعاريف ثابتة يمكن الوثوق بها.

## **2) النطاق والبيانات الداخلة**

* **أحداث التطبيق (GA4/Firebase):** view\_item, add\_to\_cart, begin\_checkout, purchase, session\_start…
* **المعاملات التشغيلية:** الطلبات، بنود الطلب، المدفوعات، الشحنات، COD، المرتجعات.
* **الكتالوج والمخزون:** منتجات/نسخ، تصنيفات، لقطة مخزون يومية.
* **العروض والرموز:** حملات، استخدام القسائم، الشحن المجاني.
* **التسويق/الإسناد:** UTM (source/medium/campaign)، القناة (App/Instagram/WhatsApp).
* **الرسائل:** فتح/نقر حملات Push/SMS/WhatsApp (اختياري).

## **3) نموذج البيانات**

**الأبعاد (Dimensions):**

* التاريخ، العميل، المنتج، النسخة، التصنيف، المنطقة/المدينة، الجهاز/منصة التطبيق، القناة/الحملة.

**الحقائق (Facts):**

* **الطلبات** (fact\_orders)
* **بنود الطلب** (fact\_order\_items)
* **المرتجعات** (fact\_returns)
* **الشحنات/COD** (fact\_shipments, fact\_cod\_collections)
* **جلسات/مستخدمين** (fact\_sessions)
* **مشاهدات/إضافات للسلة** (fact\_product\_views) تلخيص يومي
* **المخزون اليومي** (fact\_inventory\_daily)
* **استخدام القسائم** (fact\_promos)

**عروض جاهزة (Materialized Views):**

* mv\_funnel\_daily (عرض → سلة → دفع → شراء)
* mv\_abandoned\_carts\_7d
* mv\_repeat\_purchase\_30d (تكرار شراء)
* mv\_top\_skus (الأكثر مبيعًا/هامشًا)

## **4) أهم المؤشرات (KPIs)**

* **التحويل:** Add-to-Cart %, Checkout %, Purchase %, Conversion Funnel.
* **الإيراد:** Revenue, AOV, Discount %, (الهامش إن توفّرت التكلفة).
* **العملاء:** LTV، RFM، Cohorts احتفاظ أسبوعي/شهري، تكرار شراء.
* **الكتالوج:** الأكثر مشاهدة/إضافة/بيعًا، معدل الإرجاع لكل SKU، اكتمال المحتوى.
* **السلة:** حجم وقيمة السلال المهجورة، استرداد المهجورة.
* **الشحن/COD:** First-Attempt Delivery %, RTO %, COD Collection %, تسوية ≤7 أيام.
* **التسويق:** ROAS/CVR حسب القناة/الحملة، استخدام القسائم وفعاليّتها.
* **البحث:** معدل “لا توجد نتائج”، استعلامات الأعلى تحويلًا.

## **5) لوحات التقارير (Dashboards)**

* **نظرة تنفيذية يومية:** إيراد/طلبات/AOV/تحويل/خصم/مرتجعات/RTO.
* **تسويق:** قنوات/حملات، ROAS، قسائم، Funnel بالقناة.
* **merchandising:** أداء الفئات والـSKUs، الهامش، نفاد المخزون.
* **العمليات/الشحن:** SLA، محاولات التسليم، COD والتحصيل.
* **العملاء:** RFM، Cohorts، LTV، شرائح VIP/غير نشطين.
* **السلة والبحث:** مهجورة، نصوص بحث بلا نتائج وتحسيناتها.

## **6) التحديث والزمنية (Freshness)**

* شبه لحظي (كل 15–60 دقيقة): أحداث التطبيق، مؤشرات المبيعات الأساسية.
* يومي (Daily): لقطات المخزون، Cohorts، LTV، تقارير الشحن/COD.
* أسبوعي: لوحات المورد/التكلفة إن لزم.

## **7) الجودة والحَوْكمة**

* قاموس بيانات موحّد لتعريف كل مؤشر.
* فحوص صحة دورية: توازن طلبات/مدفوعات/شحنات، ازدواج/فجوات، حدود منطقية.
* صلاحيات وصول حسب الدور، تقليل بيانات شخصية، احترام الموافقات.

## **8) تعريف الاكتمال (Definition of Done)**

* تدفّق بيانات من المصادر المذكورة إلى مخزن تحليلي موحّد.
* طبقة أبعاد/وقائع جاهزة للاستعلام + العروض المادية الأربعة أعلاه.
* لوحات: تنفيذية، تسويق، Merchandising، عمليات الشحن/COD، عملاء، سلة/بحث.
* KPIs الأساسية محسوبة ومتطابقة مع الأرقام التشغيلية.
* تواتر التحديث مفعل (لحظي/ليلي) مع تنبيهات تعطل.

## **9) خارطة إصدارات مختصرة**

* **MVP (أسبوعان):** fact\_orders/order\_items/sessions/product\_views + لوحتين (تنفيذية، تسويق) + mv\_funnel\_daily.
* **المرحلة 2:** شحن/COD/مرتجعات/Promos + لوحتي العمليات والكتالوج + abandoned carts.
* **المرحلة 3:** Cohorts/LTV/RFM، لوح العملاء، تنبؤ نفاد مخزون، تحسين البحث.

بهذه الطبقة تحصل على مصدر أرقام موثوق، يربط الرحلة من التصفح حتى التسليم، ويُظهر ببساطة أين نكسب وأين نهدر—وبوتيرة تحديث تكفي لاتخاذ قرارات يومية بثقة.

المحفظة (Wallet / Store Credit)

# **المحفظة (Wallet / Store Credit)**

## **1) الهدف**

* تمكين **رصيد متجر** آمن وسريع للاستخدام في الشراء، والاسترداد من المرتجعات.
* تقليل زمن الإغلاق المالي لحالات RMA وطلبات COD.
* تحفيز التكرار والولاء عبر **Cashback** وبطاقات هدية قابلة للاسترداد.

## **2) النطاق**

* **رصيد متجر للعميل** (Store Credit) مرتبط بحسابه.
* **معاملات محفظة**: إضافة (Credit) وخصم (Debit) و**حجز مؤقت** أثناء الدفع.
* **سياسات**: صلاحية/انتهاء، حدود استخدام، مكافحة الاحتيال.
* **مصادر الرصيد**: مرتجعات، تعويضات، Cashback، (اختياري) تعبئة رصيد/بطاقات هدية.
* **استخدام الرصيد**: وسيلة دفع جزئية/كاملة للطلبات، مع الدمج مع بطاقة/COD.
* **(اختياري)** بطاقات هدية (Gift Cards) برموز قابلة للنقل.

## **3) الكيانات (مفهومية—بدون تفاصيل تقنية)**

* **Wallet Account**: لكل عميلة حساب محفظة بعملة العرض (YER أساسًا).
* **Wallet Transaction**: حركة مالية بقيَم موجبة/سالبة، مع مصدر/مرجع (طلب/مرتجع/تعويض).
* **Wallet Hold (Authorization)**: حجز مؤقت عند بدء الدفع، يُحرَّر أو يُخصَم عند إنشاء الطلب.
* **Gift Card (اختياري)**: بطاقة ذات رمز وقيمة وصلاحية، قابلة لتحويل الرصيد إلى محفظة العميلة عند الاسترداد.
* **Cashback Rule (اختياري)**: تعريف نسبة/قيمة تُكتسب بعد اكتمال الطلب.

## **4) الحقول الأساسية (مختصرة)**

### **Wallet Account**

* المعرّف، **customer\_id**، العملة، **balance\_available**، **balance\_on\_hold**، تواريخ الإنشاء/التحديث.

### **Wallet Transaction**

* المعرّف، **customer\_id**، النوع (credit/debit/adjustment)، المبلغ (+/-)، العملة.
* **source** (refund/cashback/manual/giftcard/recharge)، **reference\_id** (طلب/RMA/كوبون).
* **status** (pending/posted/reversed)، **reason\_code** (تعويض/مرتجع/عرض…).
* الفاعل (نظام/موظف)، ملاحظات، طابع زمني.

### **Wallet Hold**

* المبلغ المحجوز، مرجع السلة/الدفع، **expires\_at** (مهلة تلقائية)، الحالة (active/released/captured).

### **Gift Card (اختياري)**

* الرمز، الرصيد الأصلي/المتبقي، العملة، **expires\_at**، الحالة (نشطة/مستردّة/منتهية)، مرات الاستخدام.

### **Cashback Rule (اختياري)**

* نسبة/قيمة ثابتة، حدود قصوى، أهلية (فئات/قنوات)، **lock\_period** حتى الإتاحة (مثلاً بعد 7–14 يومًا من التسليم).

## **5) قواعد العمل (Business Rules)**

* **ترتيب التطبيق في الدفع**:
  + سعر البنود − الخصومات/الكوبونات →
  + **الشحن** و**رسوم COD** (حسب السياسة) →
  + **تطبيق رصيد المحفظة** →
  + الباقي يُدفع ببطاقة/COD.
* يفضّل استثناء **رسوم COD** من التغطية بالمحفظة لتبقى مفهومة (قرار تجاري).
* **المرتجعات**: الافتراضي **استرداد إلى المحفظة** (أسرع)، مع إمكان تحويل بنكي يدوي عند الطلب.
* **الدمج مع طرق الدفع**: يسمح بالدفع المختلط (محفظة + بطاقة أو محفظة + COD).
* **الحجز المؤقت**: عند Begin Checkout يُنشأ **Hold**؛ عند إنشاء الطلب يُحوَّل إلى **Debit**، وإلا يُحرَّر تلقائيًا عند انتهاء المهلة.
* **الصلاحية**: رصيد متجر من المرتجعات **لا ينتهي** (مستحسن). Cashback/بطاقات هدية قد تنتهي (مثلاً 12 شهرًا).
* **القيود**: حد أقصى يومي/شهري للاستخدام، وحد أقصى للرصيد المتراكم (مكافحة إساءة).
* **العملة**: المحفظة بعملة السوق النشطة (YER). إن تعددت الأسواق لاحقًا: إمّا محفظة لكل عملة أو تحويل بسعر صرف مثبت لحظة الحركة.
* **مكافحة الاحتيال**:
  + تفعيل OTP عند استخدام رصيد كبير أو استرداد بطاقة هدية.
  + قيود سرعة (Velocity) لعمليات الإضافة/الخصم، وتطابق الجهاز.
  + مراجعة يدوية لحركات **manual adjustment** مع سبب إلزامي.
* **دمج الحسابات**: عند دمج عميلَين، تُدمَج المحافظ وحركاتها مع **Audit Trail**.
* **السجلات والتوافق**: جميع الحركات تُسجَّل كسجل غير قابل للتعديل (append-only) مع ربط محاسبي (التزام/ذمم).

## **6) التدفقات التشغيلية (Workflows)**

### **أ) الدفع بالمحفظة**

1. المستخدم يختار “الدفع بالمحفظة” ويحدّد المبلغ → إنشاء **Hold**.
2. نجاح الدفع الإضافي (إن وُجد) → إنشاء الطلب → **Capture → Debit** ويُخصم من الرصيد.
3. فشل/إلغاء → **Release** للحجز.

### **ب) مرتجع → رصيد متجر**

1. إغلاق RMA ببنود مقبولة → إنشاء **Credit** بالمبلغ المناسب (بعد خصم جزء الخصم **pro-rata** ورسوم الشحن حسب السياسة).
2. يظهر الرصيد فورًا في المحفظة.

### **ج) Cashback (اختياري)**

1. بعد **delivered** وانتهاء نافذة المرتجع (مثلاً 7 أيام) → إنشاء **Credit** وفق القاعدة وحدودها.
2. إشعار العميلة وإظهار تاريخ الانتهاء إن وُجد.

### **د) بطاقات هدية (اختياري)**

* شراء بطاقة → إنشاء كيان Gift Card برمز يُرسل للمتلقّي.
* الاسترداد: إدخال الرمز → تحويل الرصيد إلى **Wallet Credit** مرتبط بالعميلة.
* الاستخدام جزئي/كلي، وتتبع الرصيد المتبقي.

### **هـ) تعويض/تعديل يدوي**

* فقط لأدوار مفوّضة، مع **سبب إجباري**، ومراجعة ثنائية لأكثر من حد محدد.

## **7) التكاملات**

* **الطلبات/الدفع**: الحجز/الخصم، ترتيب التطبيق، دعم الدفع المختلط، تحديث مبلغ COD المتوقَّع إذا استُخدمت المحفظة.
* **RMA**: إنشاء Credits تلقائيًا، وربط الحركة بملف المرتجع.
* **العروض**: Cashback مستقل عن الكوبونات (لا يتعارض)، ويظهر في الإيصال.
* **التحليلات**: أحداث wallet\_credit, wallet\_debit, wallet\_hold, wallet\_release, cashback\_earned, giftcard\_redeemed.
* **المحاسبة**: المحفظة = **التزام (Liability)**؛ قيود تلقائية عند كل Credit/Debit؛ تقارير تسوية يومية.

## **8) مؤشرات الأداء (KPIs)**

* **Wallet Usage %**: نسبة الطلبات التي استخدمت المحفظة.
* **Repeat Rate Lift** لمن استخدموا المحفظة مقابل من لم يستخدموها.
* **Time-to-Refund**: زمن من قبول RMA حتى ظهور الرصيد.
* **Breakage %**: رصيد منتهي الصلاحية (للكاش باك/بطاقات الهدايا) مقابل المكتسب.
* **Fraud Flags**: محاولات غير طبيعية/ضخمة، تعديلات يدوية فوق الحد.
* **COD Impact**: انخفاض المبلغ المُحصَّل نقدًا عند الباب بفضل المحفظة.

## **9) معايير القبول (Definition of Done)**

* لكل عميلة **Wallet Account** واضح برصيد متاح/محجوز.
* يمكن الدفع بالمحفظة جزئيًا/كليًا مع **Hold/Capture/Release** صحيح.
* المرتجعات تُنشئ **Credit** تلقائيًا بنِسَب pro-rata للخصومات.
* Cashback يعمل وفق القواعد بعد التسليم ونهاية مهلة المرتجع.
* (اختياري) بطاقات هدية قابلة للاسترداد إلى المحفظة برموز فريدة.
* تقارير تسوية يومية: مجموع الأرصدة = رصيد حساب الالتزامات، دون فروقات.
* سجلات تدقيق كاملة ومراجعة ثنائية للتعديلات اليدوية.

## **10) سيناريوهات حافة**

* **مزج محفظة + بطاقة** ثم مرتجع جزئي: تُعاد نسبة الدفع إلى البطاقة والمحفظة **pro-rata**.
* **شحنة مجزأة**: خصم المحفظة عند إنشاء الطلب، وليس عند كل شحنة.
* **فشل دفع خارجي بعد خصم المحفظة**: عكس حركة المحفظة تلقائيًا (Reverse) إن لم يُنشأ الطلب.
* **سالب بالخطأ**: منع الرصيد السلبي؛ معاملات متسلسلة ضمن قفل/معاملة واحدة.
* **بطاقة هدية مسروقة/مسرّبة**: OTP عند الاسترداد + حد يومي + إبطال رمز عند الإبلاغ.

## **11) خارطة الإصدارات**

* **MVP**: رصيد متجر من المرتجعات/التعويضات + دفع بالمحفظة (Hold/Capture) + تقارير تسوية + أحداث تحليلات أساسية.
* **المرحلة 2**: Cashback بقواعد بسيطة + حدود/صلاحيات + إشعارات.
* **المرحلة 3**: بطاقات هدية، تعبئة رصيد (Recharge) عبر البوابة، محفظة متعددة العملات، مركز نزاعات.

الحوكمة والتكامل (Governance & Integrations)

# **الحوكمة والتكامل (Governance & Integrations)**

## **1) الهدف**

* حماية البيانات وتشغيل النظام بثقة وموثوقية.
* ضمان جودة الأرقام واتساقها عبر كل الوحدات (كتالوج، عملاء، طلبات…).
* تكامل آمن ومرن مع مزوّدي الدفع والشحن والرسائل والتحليلات والمحاسبة.

## **2) الحوكمة (Governance)**

### **أ) الأدوار والصلاحيات (RBAC)**

* **الأدوار المقترحة:** Admin، عمليات الطلبات، الشحن، المالية، خدمة العملاء، المحتوى/الكتالوج، التسويق، التحليلات/BI، المطوّر.
* **مبدأ أقلّ صلاحية:** كل دور يرى ويعدّل فقط ما يحتاجه.
* **عمليات حساسة:** الاسترداد المالي، تغيير حالة الطلب، تعديل الأسعار—تتطلب صلاحيات عليا وتسجيل تدقيق.

### **ب) الخصوصية وPII والموافقات**

* **تقليل البيانات** (مطلوب فقط)، وإخفاء/إخفاء جزئي في واجهات الدعم.
* **الموافقات**: احترام قنوات Push/SMS/WhatsApp/Email في كل الإرساليات.
* **طلبات المستخدم (DSR):** تصدير/حذف بيانات العميل عند الطلب ضمن SLA داخلي (مثلاً 7 أيام).

### **ج) جودة البيانات وحَوْكمتها**

* **قاموس بيانات** يعرف المؤشرات والحقول المحورية (AOV، LTV، RTO…).
* **اختبارات صحة دورية:** فراغات/قيم شاذة/اتساق مراجع (FK)، تطابق طلبات↔مدفوعات↔شحنات.
* **سياسات التسمية والقيم المسموح بها** للألوان/المقاسات/المدن لتفادي الفوضى.

### **د) سجلات التدقيق (Audit Logs)**

* لكل تعديل على: الطلبات، المدفوعات، الشحن، RMA، الأسعار، الصلاحيات.
* **غير قابلة للتعديل** (append-only) وتُحفظ لمدة لا تقل عن 12 شهرًا.

### **هـ) الاحتفاظ والحذف (Retention)**

* سجلات التطبيق: 90 يومًا (عمليات)، **Audit**: 12–24 شهرًا، بيانات تحليلية المجمّعة أطول.
* حذف/أرشفة السلال والعناصر غير النشطة بعد فترة محددة (مثلاً 60 يومًا للضيوف).

### **و) الأمن**

* **تشفير أثناء النقل** (TLS 1.2+) و**عند السكون** لقواعد البيانات والنسخ.
* **إدارة أسرار** مركزية (Vault/KMS)، وتدوير دوري للمفاتيح.
* **2FA** للحسابات المميّزة، وقواعد كلمات مرور، وقوائم تحكم بالوصول الشبكي.
* **عدم تخزين بيانات بطاقات** (امتثال PCI DSS عبر توكنات مزوّد الدفع).

### **ز) النسخ الاحتياطي والتعافي من الكوارث (B/DR)**

* **أهداف**: RPO ≤ 15 دقيقة، RTO ≤ 2 ساعة.
* نسخ تلقائية (يومي كامل + لقطات دورية)، **اختبار استعادة** شهري.

### **ح) إدارة التغييرات (Change Management)**

* مسارات **Dev/Staging/Prod**، ومراجعة تغييرات، وهجرات قاعدة بيانات **متوافقة رجعيًا**.
* **Feature flags** للتدرّج في الإطلاق والتراجع السريع (rollback).

### **ط) المراقبة والموثوقية (Observability & SLOs)**

* **لوغز + مقاييس + تتبّع**: أخطاء، زمن استجابة، ازدحام الطوابير.
* **SLOs** مقترحة: Checkout p95 < 800ms، فشل Webhook < 0.5%، تزامن تتبع الشحن < 15 دقيقة.
* تنبيهات عند: ارتفاع فشل الدفع، زيادة RTO، تعطّل تكامل.

## **3) التكامل (Integrations)**

### **أ) واجهات وخدمات**

* **REST API داخلية/خارجية**: ترقيم إصدارات، ترقيم صفحات، **Rate Limits**، **Idempotency Keys** لعمليات الإنشاء/الدفع.
* **Webhooks** للأحداث: توقيع/تحقق، **إعادة محاولات بخوارزمية backoff**، و**Deduplication**.
* **حافلة أحداث** (Event Bus) داخلية للأحداث الموحّدة.

### **ب) أهم الأحداث (نماذج أسماء)**

* order.created, order.status\_changed
* payment.authorized, payment.captured, payment.refunded
* shipment.created, shipment.out\_for\_delivery, shipment.delivered, shipment.failed
* cod.collected, cod.remitted
* rma.requested, rma.resolved
* cart.abandoned, product.stock\_low

### **ج) مزوّدون نموذجيون والتكامل**

* **الدفع:** صفحات/SDK + Webhooks لحالات التفويض/التحصيل/الاسترداد. 3DS/SCA مفعّل.
* **الشحن:** إنشاء بوليصة/ملصق، تتبّع، Webhooks للحالات، **CSV بديل** عند تعطل التكامل.
* **الرسائل:** FCM (Push)، SMS/WhatsApp (مزود محلي/دولي) مع تتبّع التسليم/النقر.
* **التحليلات:** GA4/Firebase (user\_id = customer\_id)، تدفّق أحداث الشراء.
* **المحاسبة/ERP:** ترحيل الإيراد، الخصومات، المرتجعات، وحساب **وسيط COD**، وتسويات المزوّدين.
* **الإعلانات/الجماهير:** مزامنة شرائح (VIP/غير نشطين) عبر **Reverse ETL** (اختياري).

### **د) عقود البيانات والمعالجات**

* **عقد حقول** واضح لكل تكامل (مطلوب/اختياري/أنواع/نطاقات).
* **توحيد المناطق الزمنية** (Asia/Aden) والعملات (تثبيت الصرف لحظة الطلب/الدفع).
* **قنوات بديلة** عند الفشل (سقوط إلى CSV/Queue) دون فقدان البيانات.

### **هـ) الاختبار والبيئات**

* **Sandboxes** لمزوّدي الدفع والشحن.
* بيانات اختبار مجهولة (لا PII حقيقية).
* سيناريوهات فشل إجبارية: انقطاع Webhook، اختلاف مبالغ COD، تتبع مفقود.

## **4) مؤشرات الحوكمة والتكامل**

* دقة التطابق بين **طلبات↔مدفوعات↔شحنات** (اختلاف = 0).
* زمن معالجة Webhook المتوسط < 2 ثانية؛ نسبة إعادة المحاولات ضمن حد مقبول.
* نجاح مزامنة جماهير التسويق > 99%.
* اختبارات استعادة النسخ ناجحة شهريًا.
* التزام الوصول حسب الدور (لا خروقات صلاحيات).

## **5) تعريف الاكتمال (Definition of Done)**

* أدوار وصلاحيات مُفعّلة، وسجلات تدقيق لكل العمليات الحساسة.
* سياسات خصوصية/احتفاظ/DSR موثّقة ومجرّبة.
* نسخ احتياطي قائم مع **اختبار استعادة** حديث.
* مراقبة وتنبيهات فعّالة، وSLOs مضبوطة.
* واجهات/ويبهوكس بإصدار ثابت، Idempotency، توقيع، وإعادة محاولات.
* تكاملات الدفع/الشحن/الرسائل/التحليلات/المحاسبة تعمل من طرف لطرف مع سيناريوهات الفشل.

بهذه الحزمة تضمن تشغيلًا آمنًا وقابلًا للتوسع، وأرقامًا موثوقة، وتكاملات مستقرة— مع قدرة سريعة على اكتشاف الأعطال والتعافي منها، وامتثال واضح للخصوصية والمالية.

Architecture Decisions (ADR)

* **الترميز/الترتيب**  
  النوع: utf8mb4 مع collation = utf8mb4\_0900\_ai\_ci على مستوى السيرفر/القاعدة/الجداول.  
  السبب: يدعم الإيموجي واللغات ويعطي فرزًا دقيقًا غير حساس للحركات وحسن الأداء في MySQL 8.
* **المفاتيح الأساسية والمعرّف العام**  
  النوع: BIGINT UNSIGNED AUTO\_INCREMENT كـ PK داخلي + حقل public\_id = ULID (CHAR(26)).  
  السبب: أداء عالٍ للـPK المُجمّع، وULID قابل للفرز زمنيًا وآمن للمشاركة خارجيًا.
* **الحذف المنطقي**  
  النوع: عمود deleted\_at DATETIME NULL + فهرس + Views/Scopes تستثنيه افتراضيًا.  
  السبب: اتساق مع الأُطر الشائعة وسهولة الاسترجاع والتدقيق.
* **حجز المخزون**  
  النوع: جدول reservations (sku\_id, qty, state, reservation\_expires\_at, order\_id nullable) + Job دوري لتحرير المنتهي.  
  السبب: يمنع البيع الزائد ويضمن تحرير الحجوزات غير المُتمَّمة.
* **تسويات الدفع عند الاستلام (COD)**  
  النوع: cod\_settlements و cod\_settlement\_items مع ربط للأوامر/المدفوعات/الشحن وحالات (open/submitted/reconciled/closed).  
  السبب: تتبّع مالي دقيق وقابل للمطابقة البنكية/الناقل.
* **إدارة المرتجعات (RMA)**  
  النوع: rmas و rma\_items بحالات واضحة (requested/approved/received/refunded/closed) وربط بأسباب ledger (return\_to\_stock/damaged/lost).  
  السبب: ربط محاسبي/مخزني مُحكم وتقليل الفقد.
* **سلسلة التوريد (المشتريات)**  
  النوع: suppliers, purchase\_orders, purchase\_order\_items, asn, grn, qc, landed\_cost\_allocations.  
  السبب: مصدر واحد للحقيقة وتوزيع تكلفة الوصول (Landed Cost) بشكل عادل (وزن/قيمة/كمية).
* **سياسات الاحتفاظ والأرشفة**  
  النوع: Orders/RMAs: 5 سنوات – Payments/Settlements: 7 سنوات – Shipments/Fulfillment: 24 شهرًا – Inventory Ledger: 36 شهرًا – App Logs: 12 شهرًا – Raw Analytics: 6 أشهر مع طبقات مجمّعة دائمة.  
  السبب: توازن امتثال/تكلفة/أداء مع مسار أرشفة واضح.
* **أنواع الحالة/الأكواد**  
  النوع: جداول مرجعية Lookup + مفاتيح خارجية (لا تُستخدم ENUM).  
  السبب: مرونة التعديل بدون تغييرات مخطط/نشر كود، مع سلامة مرجعية.
* **الفهارس وSLO للأداء**  
  النوع: أهداف p95 لقراءات OLTP الأساسية ≤ 30ms وعمليات الكتابة ≤ 50–80ms، ومراجعة شهرية لأهم 10 استعلامات مع EXPLAIN وخريطة فهارس مُلزمة.  
  السبب: أداء مُتنبَّأ به وتقليل تضخم الفهارس والاستعلامات المنحرفة.
* **التقسيم (Partitioning)**  
  النوع: inventory\_ledger تقسيم RANGE شهريًا على txn\_ts؛ نافذة بيانات “حارة” 18 شهرًا مع أرشفة أقدم.  
  السبب: أحجام كبيرة/كتابة كثيفة واستعلامات زمنية أسرع وصيانة أسهل.
* **الصلاحيات والبيانات الحساسة والتدقيق**  
  النوع: audit\_log ملاحق فقط (append-only) مع JSON diff، Views مقنّعة للـPII، تشفير بالحفظ، احتفاظ للتدقيق 5 سنوات، Least-Privilege على مستوى الدور.  
  السبب: امتثال وتتبع تغييرات غير قابل للعبث وتقليل سطح المخاطر.
* **القاموس البياني وERD**  
  النوع: قاموس بيانات حي (Editable) + ERD مُحدَّث ضمن المستودع مع مصفوفة القيود.  
  السبب: توحيد الفهم، سهولة الانضمام للفريق، تقليل أخطاء النمذجة.
* **ضوابط الإنتاج**  
  النوع: حظر phpMyAdmin كتابةً في الإنتاج، تغييرات المخطط عبر Migrations فقط، أذون قراءة مُقيّدة عند الحاجة.  
  السبب: حوكمة تغيير صارمة وتتبّع كامل وتقليل أخطاء بشرية.

Operational Controls

* **ترحيلات مُعَنوَنة**  
  اعتماد أداة migrations مع **checksums**، وGate في CI يمنع أي DDL خارج الترحيلات.
* **بيانات مرجعية مُصدَّرة**  
  Versioning لـ seed/reference data وترحيلها بين البيئات تلقائيًا.
* **حواجز اتصال قاعدة البيانات**  
  حدود صارمة لـ max connections/queue لكل خدمة + backoff وتوقيت p95 للحماية من الانهيار.
* **مهلات الاستعلام والقتل الآلي**  
  تعريف max\_execution\_time وسياسة kill للـ long-running (عبر ProxySQL/pt-kill).
* **سياسة القراءة من النسخ المتماثلة**  
  SLO لتأخر النسخ (<2s)، وRead-after-write fallback للـ primary تلقائيًا.
* **خطة تراجع مُختبرة**  
  Point-in-Time Recovery مُجرَّب دوريًا + Playbook واضح بـ RPO/RTO.
* **إخفاء/تجزئة البيانات لغير الإنتاج**  
  Pipeline لتوليد عينات فرعية مع **إخفاء حتمي** يحافظ على العلاقات.
* **تطبيع اليونيكود**  
  فرض **NFC** ومنع الأحرف غير المرئية/ثنائية الاتجاه في الإدخال، مع فحص دوري.
* **سياسة المنطقة الزمنية**  
  تخزين **UTC فقط** على مستوى القاعدة، وتعامل التحويل في طبقة التطبيق.
* **أنماط العمليات الضخمة**  
  Batching (مثلاً 1k صف/معاملة)، chunking حسب PK، واستخدام SKIP LOCKED لتقليل التعارض.
* **ترحيلات عالية الخطورة بأسلوب Blue/Green**  
  جداول/عروض ظلّية (shadow) مع تبديل تدريجي وقياسات قبل/بعد.
* **حراسة جودة البيانات تلقائيًا**Checks يومية: nulls غير متوقعة، ازدواج فريد، أيتام مرجعية، وانحراف مخطط (schema drift) مع تنبيه.
* **نافذة صيانة وخط اتصال**  
  تعريف نافذة صيانة ثابتة، وخطوات تبليغ/تراجع موحدة للفِرق المعنية.
* **سياسة الإجراءات المخزّنة**  
  منع المنطق التجاري داخل SP/Triggers (باستثناء تدقيق/حفظ سلامة بسيطة) مع بدائل واضحة في التطبيق.

ملخص آلية العمل واعتمادياتها

# **ملخص آلية العمل واعتمادياتها**

* **ADR** = يحدد “ماذا ولماذا” (قرارات البنية والمعايير).
* **Operational Controls** = يحدد “كيف ومتى ومن” (تشغيل/حوكمة/اختبارات).
* الاعتماديات: CI/CD، أذونات DB، نسخ احتياطي/استعادة، مراقبة (Monitoring)، تنبيهات، نسخ متماثل (إن وُجد).

## **1) اربطها بالتنفيذ (CI/CD)**

* **Gate** يمنع أي DDL خارج *migrations*.
* **Checklists في الـCI**:
  + فحص collation/charset الافتراضي + أي جدول شاذ.
  + فحص وجود public\_id (ULID) لكل الجداول المستهدفة.
  + فحص تغطية deleted\_at للجداول التي تتطلب soft-delete.
  + فحص وجود فهارس الحقول الحرجة (FKs/التواريخ/الحالات).
  + فحص سياسة التقسيم للجداول المحددة وإنشاء قسم الشهر القادم مسبقًا.
* **Artifacts إلزامية مع كل نشر**: تقرير EXPLAIN لأهم 10 استعلامات + diff للمخطط + نتائج اختبارات الاستعادة.

## **2) عبِّئ “حقول الإثبات” داخل الوثيقتين**

لكل بند رئيسي املأ: **المالك (DRI)**، **SLO/SLA**، **أداة/آلية التنفيذ**، **أثر الفشل**، **دليل تحقق (Evidence)**، **تاريخ المراجعة**.

## **3) فحص اليوم صفر (Day-0 Validation)**

نفّذ هذا التحقق السريع قبل اعتبار المعيار مُعتمدًا:

1. **Collation/Charset**: ثابت على مستوى السيرفر/القاعدة/الجداول (لا استثناءات).
2. **PK + public\_id**: BIGINT PK داخلي + ULID موجود ومفهرس حيث يلزم.
3. **Soft-delete**: deleted\_at مفهرس + جميع الـViews/Scopes تستثنيه افتراضيًا.
4. **الحجز (Reservations)**: سيناريو إنشاء/انتهاء مهلة/تحرير تلقائي ناجح.
5. **COD Settlements**: دورة حياة كاملة (open→reconciled→closed) مع الربط بالفواتير.
6. **RMA**: انتقالات الحالات وتحديث ledger والأرصدة المالية تعمل.
7. **Procurement/Landed Cost**: PO→ASN→GRN→QC→Allocation تعمل والأرقام موزعة كما اخترتم (وزن/قيمة/كمية).
8. **Inventory Ledger**: يُسجل كل الحركات، والتقسيم مفعل، وأرشفة >18 شهر جاهزة.
9. **RBAC/PII**: Views مقنّعة تعمل، وأقل صلاحيات مفروضة فعليًا.
10. **Backups & PITR**: استعادة تجريبية ناجحة داخل الزمن الهدف (RTO/RPO).
11. **Replication/Read-policy**: تأخر النسخ ضمن الحد + fallback للقراءة من primary مثبت.
12. **Monitoring/Alerts**: تنبيهات p95 للأداء، تأخر النسخ، نمو الجدول/القسم، فشل نسخ احتياطي.

## **4) تعريف “اكتملنا” (Definition of Done)**

* كل بند في ADR له **تطبيق مؤكد** في البيئة (بدون TODOs).
* كل بند في Operational Controls له **Runbook** مختصر + **تنبيه** عند الانحراف.
* تقرير واحد من CI يثبت **نجاح Day-0** + أسماء الملاّك لكل عنصر.

## **5) ترتيب التنفيذ (سريع)**

* تفعيل Gates + فحوص CI + Day-0.
* اختبار استعادة كامل + تحميل بيانات مرجعية بنسخها + ضبط التنبيهات.
* مراجعة شهرية للأداء والفهارس + ربع سنوية لسياسات الاحتفاظ.

# 

# **ملخص آلية العمل وروابطها/اعتمادياتها**

* ADR = قرارات البنية (repo).
* Operational Controls = ضوابط التشغيل (repo + CI/CD + DB + Monitoring).

## **1) في مستودع الكود (Repo)**

* **القرارات**:  
  docs/adr/\*.md (كل بند رقم/مالك/تاريخ).
* **ضوابط التشغيل/التشغيلية**:  
  docs/runbooks/\*.md وdocs/checklists/\*.md.
* **المخطط والتوثيق**:  
  docs/erd/ (ERD) + docs/data-dictionary.xlsx.
* **الهجرة (migrations) والبيانات المرجعية**:  
  Laravel: database/migrations/, database/seeders/  
  (أو Flyway/Liquibase: db/migrations/, db/seeds/).
* **استعلامات الفحص الآلي**:  
  ops/sql/schema\_checks/\*.sql (فحوص collation/PK/soft-delete/فهارس).
* **اسكربتات اليوم صفر**:  
  ops/day0/\*.sh|sql.

## **2) في CI/CD**

* **ملفات الأنابيب**:  
  GitHub Actions: .github/workflows/db-ci.yml  
  (أو GitLab: .gitlab-ci.yml).
* **Gates إلزامية**:
  + منع أي DDL خارج *migrations*.
  + تشغيل ops/sql/schema\_checks/\*.sql ضد قاعدة اختبار.
  + توليد تقارير EXPLAIN لأهم 10 استعلامات وحفظها كـ Artifacts.
* **خطوات نشر DB**:
  + تشغيل *migrations* + *seeders* على Staging ثم Production.
  + نشر جداول التقسيم للشهر القادم مسبقًا.
* **مستند إثبات**:  
  يخرج من الـCI ملف artifacts/db-compliance-report.html.

## **3) على خادم قاعدة البيانات (MySQL)**

* **إعدادات ثابتة (my.cnf)**:  
  character\_set\_server=utf8mb4  
  collation\_server=utf8mb4\_0900\_ai\_ci  
  default\_time\_zone='+00:00'
* **صلاحيات/أدوار**:  
  إنشاء أدوار *read/write/admin* وتطبيق *least-privilege* للحسابات.
* **نسخ متماثل/تقسيم**:  
  تفعيل تقسيم inventory\_ledger (شهري) وخطة أرشفة.
* **مهلات وحدود**:  
  ضبط مهلة الاستعلام والاتصالات (timeouts/limits) حسب الـSLO.

## **4) في طبقة التطبيق (Jobs/Cron)**

* **حجز المخزون**:  
  Job ينفّذ كل دقيقة: تحرير reservations منتهية الصلاحية.
* **أرشفة/Retention**:  
  Jobs لسياسات الاحتفاظ (نقل البارد إلى أرشيف).
* **فحوص جودة البيانات**:  
  Job يومي لفحص *nulls/duplicates/orphans* وإرسال تنبيه.

## **5) طبقة البروكسي/الوسيط (اختياري لكن مُستحسن)**

* ProxySQL / pgbouncer-style:
  + توزيع قراءات على Replica مع fallback للـPrimary.
  + قتل الاستعلامات الطويلة تلقائيًا.
  + تحديد *connection pooling/limits* لكل خدمة.

## **6) المراقبة والتنبيهات (Monitoring)**

* Grafana/PMM/CloudWatch (حسب بيئتك):  
  لوحات وAlerts لـ:
  + p95 للقراءة ≤ 30ms والكتابة ≤ 80ms.
  + تأخر النسخ (replication lag).
  + نمو الجداول والأقسام.
  + نجاح/فشل النسخ الاحتياطي وPITR.
  + معدّلات الأخطاء في Jobs التشغيلية.

## **7) النسخ الاحتياطي والاستعادة (Backups/PITR)**

* Managed (RDS/Aurora) أو Self-hosted (XtraBackup):
  + جدولة نسخ يومية + سجلات للـPITR.
  + **اختبار استعادة شهري** وتوثيقه في docs/runbooks/restore.md.

## **8) (Day-0 Validation)**

* تشغيل ops/day0/validate.sh من CI ضد Staging:
  + يختبر collation/charset/UTC/PK/public\_id/soft-delete/الفهارس/التقسيم.
  + يختبر دورة RMA وCOD وReservation end-to-end ببيانات عينة.
  + يخرج تقرير نجاح واحد موقّع من Dev/DBA/Ops.

## **9) أين أضع التنبيهات والملكية (DRI)**

* **جدول مُوحّد للملكية**: docs/owners.yml  
  لكل بند (ADR/Control): *مالك*، *SLO/SLA*، *أداة التحقق*، *التنبيه*، *تاريخ المراجعة*.

### **تنفيذ سريع (اقتراح)**

1. أضِف المجلدات/الملفات المذكورة في **Repo** الآن.
2. أنشئ Workflow واحد للـCI باسم db-ci.yml يطبق Checks + Artifacts.
3. فعّل إعدادات MySQL في **my.cnf** وأعد تشغيل الخدمة.
4. أضف 3 Jobs في التطبيق: **reservations**, **retention**, **dq-checks**.
5. أنشئ لوحات Grafana وتنبيهاتها، ثم اختبر **PITR** فعليًا وسجّله في الـRunbook.

قوالب لـ:

* db-ci.yml،
* schema\_checks.sql،
* owners.yml,
* restore.md,
* reservations\_expiry\_job (Laravel Command).

سياسة حذف (PII Erasure & Transaction Retention)

# **سياسة حذف حساب العميل (PII Erasure & Transaction Retention)**

## **ملخص آلية العمل واعتمادياتها**

* **المبدأ:** حذف/تمويه بيانات التعريف الشخصية (PII) مع **الاحتفاظ بالمعاملات** (طلبات/شحن/محفظة/فواتير) لأغراض المحاسبة والتدقيق.
* **الاعتماديات:** DB (MySQL)، نظام المحفظة/المدفوعات (يشمل COD)، الشحن، نظم الهوية (Firebase/Auth)، التحليلات (GA4/BigQuery)، قنوات الرسائل (Email/SMS/Push)، النسخ الاحتياطي/الاستعادة، CI/CD.

## **1) نطاق السياسة**

تنطبق على: **users، orders، shipments، invoices، wallet\_ledger، rmas، returns، analytics identity** وكل ما يخزّن PII (الاسم، الهاتف، الإيميل، العنوان التفصيلي، مُعرّفات الأجهزة/الدفع).

## **2) قرارات معمارية (ADR مختصرة)**

1. **الاحتفاظ بالمعاملات وإخفاء PII:** تبقى سجلات الطلبات والدفعات والمرتجعات، مع تمويه حقول الاتصال والعنوان التفصيلي.
2. **استبدال الهوية:** بعد الإزالة، يُستبدل الربط بـ **anonymized\_customer\_id** بدلاً من user\_id في السجلات التاريخية.
3. **شرط التسوية قبل الإزالة:** أي **طلب/RMA مفتوح** أو **رصيد محفظة ≠ 0** يوقف الإزالة النهائية حتى التسوية.
4. **عدم الرجوع:** تخزين **pii\_hash** (hash مُملّح لقنوات التعريف) لأغراض منع إساءة الاستخدام بدون قابلية استرجاع PII.
5. **حذف المعرّفات التشغيلية:** إلغاء الجلسات، OAuth، Push/device IDs، وإصدار طلب حذف لأنظمة الطرف الثالث.

## **3) تغييرات المخطط المطلوبة (DDL المعياري)**

* **جدول anonymized\_customers**
  + الأعمدة: anonymized\_customer\_id (ULID), created\_at.
* **users**
  + أعمدة جديدة: deleted\_at, erased\_at, erase\_job\_id, pii\_hash, anonymized\_customer\_id (nullable).
  + تفريغ/تمويه أعمدة PII عند الإزالة (name/email/phone/address\*).
* **الجداول المرتبطة بالعميل** (orders, shipments, invoices, rmas, wallet\_ledger, …)
  + إضافة عمود: anonymized\_customer\_id (nullable, indexed).
  + **قاعدة التكامل:** عند الإزالة النهائية: user\_id = NULL وanonymized\_customer\_id مُعبّأ. (يُفرض بقيود تحقق/Triggers خفيفة أو عبر Jobs).
* **wallet\_ledger**
  + إضافة نوع حركة: wallet\_closure لإقفال الرصيد إلى صفر قبل الإزالة.
* **Views/Scopes**
  + عروض قراءة تُخفي PII افتراضيًا وتعرض المدن/المناطق فقط إن لزم محاسبيًا.

ملاحظة تصميمية: تُنشأ **فهارس** على anonymized\_customer\_id في كل الجداول، وفهرس مركّب على (state, reservation\_expires\_at) إن وُجدت حجوزات مرتبطة.

## **4) تمويه/إخفاء PII (معايير)**

* **الاسم:** يُمسح أو يُستبدل بقيمة ثابتة ([Erased]).
* **الهاتف/الإيميل:** تمويه غير قابل للعكس (مثل \*\*\*\*1234 للهاتف وآلية Hash مُملّحة داخليًا للحظر/التحقق دون تخزين الأصل).
* **العنوان:** حذف السطور التفصيلية، إبقاء **المدينة/المنطقة/الدولة** فقط عند الحاجة المحاسبية.
* **المعرّفات التشغيلية:** حذف Tokens والجلسات وPush/device IDs والموافـقات التسويقية.

## **5) تدفق التنفيذ (Runbook مختصر)**

### **A) عند تقديم طلب حذف**

1. **فحص مسبق**:
   * طلبات/RMAs مفتوحة؟
   * **رصيد محفظة** موجب/سالب؟
   * نزاعات/Chargebacks؟
2. **لو توجد حالات مفتوحة:**
   * **تعطيل الحساب فورًا** (لا تسجيل دخول، لا تسويق)، وجدولة الإزالة بعد (تسليم/إلغاء/إغلاق RMA).
3. **رصيد محفظة > 0:**
   * صرف آلي لنفس وسيلة الدفع إن أمكن؛ وإلا: تسوية بديلة (تحويل بنكي/محفظة جوال/قسيمة).
   * عدم استجابة العميل **14 يومًا** ⇒ صرف افتراضي محدّد مسبقًا.
4. **رصيد محفظة < 0:**
   * تحصيل/إلغاء وفق السياسات حتى يصل إلى **0**، ثم المتابعة.

### **B) الإزالة النهائية**

1. إنشاء anonymized\_customer\_id وتعبئته في users وفي كل الجداول التابعة.
2. **تفريغ/تمويه PII** في users (واللقطات snapshot في الطلبات/الشحن/الفواتير).
3. وضع user\_id = NULL في الجداول التابعة، وتعبئة anonymized\_customer\_id.
4. **wallet\_ledger:** قيد wallet\_closure لرصيد = 0 وقفل المحفظة.
5. **طرف ثالث:** حذف من Firebase/Auth، إرسال **user deletion** لـ GA4/BigQuery بالمعرّف، إزالة من Email/SMS/Push.
6. تسجيل **audit\_log** لا يُمكن التلاعب به (من/متى/ماذا).

## **6) حالات الحافة**

* **COD قيد التسليم:** لا إزالة نهائية قبل “مُسلّم/ملغى”.
* **نزاع/احتيال:** احتفاظ بالحد الأدنى اللازم للدفاع القانوني لمدة محددة (مثلاً 12–24 شهرًا)، مع PII مُموّهة.
* **عودة العميل لاحقًا:** يُعامل كـ “حساب جديد”، لا ربط بالبيانات الممحوة، ويُمنع الربط الالتفافي عبر pii\_hash (منع تعدد الحسابات المؤذي).

**7) الامتثال والاحتفاظ**

* **المعاملات:** الاحتفاظ حسب سياسة المؤسسة (مثلاً: طلبات/فواتير 5–7 سنوات).
* **النسخ الاحتياطية:** لا تعديل رجعي؛ تُزال PII من **الأنظمة الحيّة** ويُكتفى بانقضاء فترات الاحتفاظ على النسخ.
* **الوصول:** تقييد صارم لأدوار القراءة على السجلات الممحوة/المُمَوّهة.

## **8) المراقبة وقياس النجاح (KPIs)**

* **Pass rate** لسيناريو “Erase User” في الـCI/Staging.
* **زمن المعالجة** من طلب الحذف حتى الإزالة النهائية ضمن حد أقصى (مثلاً ≤ 30 يومًا لو توجد حالات مفتوحة).
* **% تغييرات DB عبر migrations** (هدف 100%).
* **صفر** إرسال تسويقي بعد deleted\_at.
* **فشل/نجاح** مهام صرف المحفظة وإلغاء المعرّفات الطرفية.

## **9) الملكية والتشغيل**

* **DRI:** فريق البيانات/المنصّة.
* **Runbooks:** موجودة في تبويب Operational Controls (سيناريوهات: طلب حذف، صرف رصيد، إزالة طرف ثالث، استعادة/تراجع).
* **مراجعة دورية:** ربع سنوية لتحديث المدد ومعايير التمويه.

## **10) تعريف**

* لا أوامر/نزاعات مفتوحة + **محفظة = 0**.
* تنفيذ Job الإزالة أنهى: تمويه PII، نقل الهوية إلى anonymized\_customer\_id، إلغاء Tokens والأطراف الثالثة.
* سِجِل تدقيق كامل + مؤشرات المراقبة الخضراء.

سياسة دمج الحسابات المكررة (Account Merge)

# **سياسة دمج الحسابات المكررة (Account Merge)**

## **ملخص آلية العمل واعتمادياتها**

* **المبدأ:** توحيد السجلات المتكرّرة لنفس العميل دون فقدان أي معاملات أو أرصدة، مع سجّل تدقيق كامل وقابلية تراجع.
* **الاعتماديات:** قاعدة البيانات، نظام الهوية (Firebase/Auth)، المحفظة/المدفوعات (يشمل COD)، الشحن، التحليلات (GA4/BigQuery)، قنوات الرسائل، CI/CD والمراقبة.

## **1) نطاق السياسة**

تنطبق على: **users، addresses، orders، shipments، invoices، wallet\_ledger، coupons/loyalty، rmas/returns، consents، sessions/tokens، analytics identity**.

## **2) قرارات معمارية (ADR مختصرة)**

1. **حساب مستهدف واحد (Target User):** يُحدَّد وفق أعلى ثقة تحقق + أقدمية + نشاط/رصيد المحفظة.
2. **دمج آمن وقابل للتراجع:** تنفيذ عبر Job مُراقَب بمرحلة **Dry-Run** ثم **Commit** مع **account\_merge\_links** وAudit لا يُمكن العبث به.
3. **حماية الخصوصية:** لا يُدمَج حسابان دون **تحقق هوية** (OTP للهاتف أو Proof للـEmail/مزود الدخول).
4. **مبدأ الأقل خطورة:** عند تعارضات PII أو عناوين/موافقات، تُحفَظ النسخ ويفوز الحقل المُوثَّق/المُتحقَّق؛ الباقي يُؤرشف.
5. **حماية المحفظة والمعاملات:** تُجمَع أرصدة المحفظة وقسائم/نقاط الولاء وتُنقَل جميع العلاقات للـTarget بدون أي فقد.

## **3) معايير التطابق (Matching Signals)**

* **قوية (لازمة للدمج التلقائي):** هاتف مُوثّق مطابق، أو بريد مُوثّق مطابق، أو موحِّد خارجي (Firebase UID / Apple/Google).
* **مساندة (ترفع النقاط):** تطابق device fingerprint، تكرار عناوين الشحن (مدينة/منطقة + اسم)، طريقة دفع/رمز IBAN متكرر، سجلات طلب لنفس الاسم مع أرقام قريبة، **pii\_hash** مطابق.
* **عتبة قرار:** Score ≥ 0.85 ⇒ دمج تلقائي؛ 0.6–0.85 ⇒ قائمة مراجعة بشرية؛ < 0.6 ⇒ رفض.

## **4) تغييرات المخطط (DDL)**

* **account\_merge\_jobs**: حالة الدمج، العتبة، نتائج dry-run، المالك (DRI)، الطوابع الزمنية.
* **account\_merge\_links**: mapping من source\_user\_id إلى target\_user\_id مع نسخة PII أرشيفية وتمييز الحقول المتضاربة.
* **users**: أعمدة merged\_into\_user\_id (nullable), merge\_at, merge\_job\_id.
* **فهارس**: فهارس على merged\_into\_user\_id وعلى مفاتيح الربط المتأثرة لضمان انتقال سريع.
* **قيود تكامل**: السماح بإعادة إسناد FK من source→target عبر Job (لا CASCADE تلقائي على الإنتاج؛ التنفيذ مُتحكَّم به).

## **5) تدفق التنفيذ (Runbook مختصر)**

### **A) اكتشاف المرشحين**

* Job دوري يحسب **merge\_score** وفق الإشارات أعلاه ويُنتج قائمة مرشحين.
* حالات 0.6–0.85 تُدفع لصف مراجعة بشرية (Backoffice).

### **B) التحقق**

* إرسال OTP للهاتف/البريد أو التحقق عبر مزود الدخول للحسابين لإثبات الملكية.
* توثيق الموافقة الداخلية (Tick في النظام) قبل الدمج.

### **C) الدمج (مرحلتان)**

1. **Dry-Run:**
   * إحصاء العلاقات التي ستُعاد إسنادها (Orders/Wallet/Shipments/…)، كشف التعارضات (Addresses/Consents/PII).
   * Guardrails: حد أقصى للصفوف المتأثرة/المدة؛ إن تجاوزه → إيقاف للمراجعة.
2. **Commit:**
   * نقل جميع العلاقات إلى **target** (دفعات صغيرة مع معاملات).
   * **Wallet:** تجميع الأرصدة وتسجيل قيد توضيحي wallet\_merge.
   * **Loyalty/Coupons:** ترحيل الرصيد ووسم القسائم المُكررة كـ منتهية.
   * **Sessions/Tokens:** إبطال Sessions المصدر وتوحيد Push Tokens في الهدف.
   * تحديث **users**: وضع merged\_into\_user\_id على المصدر وتعطيله.
   * تحديث مزوّد الهوية: ربط المزوّدين بالـtarget فقط.

### **D) التراجع (Rollback)**

* ضمن **نافذة قصيرة** (مثلاً 24–48 ساعة):
  + استخدام **account\_merge\_links** لعكس إعادة الإسناد.
  + إعادة الأرصدة كما كانت وتفعيل حساب المصدر.

## **6) قواعد فضّ النزاعات (Conflict Resolution)**

* **PII:** يفوز المُوثَّق (Verified). إن كان كلاهما موثّقًا ومختلفًا ⇒ الاحتفاظ بكلا القيمتين مؤرشفتين، ويُطلب اختيار المستخدم لاحقًا.
* **العناوين:** إبقاء جميع العناوين ووسم غير النشط منها. العنوان الافتراضي = الأكثر استخدامًا.
* **الموافقات (Consents):** تطبيق **OR** (إن وافق على قناة في أي حساب تُعتبر موافقًا)، مع إرسال إشعار مراجعة للمستخدم.
* **الإعدادات/التفضيلات:** آخر تعديل يربح، مع سجل تغييرات.

## **7) المراقبة وقياس النجاح (KPIs)**

* **Duplicate rate** شهريًا (قبل/بعد).
* **Merge accuracy** (نسبة عدم وجود تراجعات).
* **Avg merge duration p95**.
* **Tickets عن دمج خاطئ** (يُراد تقليلها إلى صفر).
* **أثر الإيراد/التحويل** لزوار كانوا مجزئين عبر حسابات متعددة.

## **8) حالات الحافة**

* **طلبات أو RMAs مفتوحة في المصدر:** الدمج مسموح؛ تُنقل الحالة كما هي وتُكمَّل على الهدف.
* **محفظة سالبة/نزاع دفع:** الدمج يتطلب موافقة مسؤول مالي؛ وإلا يُؤجَّل.
* **بلدان/عملات مختلفة:** الإبقاء على السجلات كما هي؛ لا تحويل عملة آليًا—تُعرض تقارير موحّدة على المستوى التحليلي.
* **حساب مشتبه بالاحتيال:** لا دمج تلقائي؛ مسار مراجعة خاصة.

## **9) الملكية والتشغيل**

* **DRI:** فريق البيانات/المنصّة.
* **المراجعة:** أسبوعية لحالات المراجعة المؤجلة، ورُبع سنوية لمعادلات الـscoring.
* **التوثيق:** Runbook مفصّل وPlaybook دعم للتعامل مع بلاغات الدمج الخاطئ.

## **10) تعريف**

* تنفيذ Dry-Run ثم Commit دون تجاوز الـGuardrails.
* تحويل كل العلاقات والظهور بحساب واحد فعّال فقط.
* إبطال جميع Tokens المصدر، وتحديث مزوّدي الهوية.
* سجلات Audit كاملة، وصفحة **Merge Report** مرفقة في الـCI Artifacts.

سياسات اخرئ

# **سياسة اخرئ للحسابات، بشكل مختصر من الضروري التركيز علئ كل سياسة بشكل منفصل**

* النطاق: سياسات تشغيل لح lifecycle حساب العميل.
* الاعتماديات: قاعدة البيانات، نظام الهوية (Firebase/Auth أو مزود خارجي)، المحفظة/المدفوعات (يشمل COD)، قنوات الرسائل، التحليلات (GA4/BigQuery)، CI/CD والمراقبة.

## **1) سياسة تعطيل الحساب المؤقت (Account Freeze)**

**الهدف:** إيقاف الوصول والطلبات مؤقتًا دون حذف أو كسر التدفق المالي/اللوجستي.  
**المحفزات:** اشتباه احتيال، نزاع مالي، طلب العميل، تحقيق أمني.  
**الإجراءات:**

* users.status = frozen مع freeze\_reason, freeze\_until.
* إبطال الجلسات/التوكينات ومنع **تسجيل الدخول، الطلبات، تغييرات PII**؛ السماح بردّ الأموال وخدمة العملاء.
* إيقاف التسويق (consents موقوفة مؤقتًا) وإظهار لافتة في التطبيق.
* رفع التجميد يدويًا أو تلقائيًا عند freeze\_until.  
  **تعريف الاكتمال (DoD):** لا إنشاء طلبات أثناء التجميد، كل التوكينات مُبطلة، سجل تدقيق مكتمل.  
  **KPIs:** زمن التجميد p95، صفر طلبات/دفعات جديدة خلال التجميد.

## **2) سياسة تصحيح بيانات العميل (Rectification)**

**الهدف:** تصحيح PII بأمان مع أثر تدقيقي كامل وتزامن للأطراف الثالثة.  
**المحفزات:** طلب العميل، خطأ إدخال، مطلب قانوني.  
**الإجراءات:**

* تحقق هوية (OTP/مزود دخول).
* تحديث آمن لحقول PII + تحديث pii\_hash، وتسجيل diff في audit\_log.
* مزامنة مع Auth/Email/SMS/Analytics، وإعادة تقييم مطابقة/دمج الحسابات إذا تغيّر الهاتف/الإيميل.  
  **DoD:** البيانات محدثة ومتسقة في جميع الأنظمة، إشعار نجاح للعميل، سجل تدقيق موجود.  
  **KPIs:** SLA ≤ 24 ساعة، معدل أخطاء مزامنة = 0%.

## **3) سياسة تغيير/استرجاع رقم الهاتف الأساسي**

**الهدف:** تغيير الرقم مع حماية قوية من اختطاف الحساب.  
**المحفزات:** فقدان الرقم، تبديل شركة، خطأ سابق.  
**الإجراءات:**

* تحقق مزدوج: OTP للرقم القديم **والجديد** (أو بديل هوية إذا القديم غير متاح).
* نافذة أمان (cool-off) قصيرة لتعطيل العمليات الحساسة.
* تحديث الهاتف في Auth وDB، تجديد pii\_hash، تحديث خرائط التحليلات.
* حفظ سجل في phone\_history، منع إعادة استخدام الرقم لفترة محدودة إذا رُصدت مخاطر.  
  **DoD:** الرقم الجديد مُوثق، الجلسات الحرجة مُجدَّدة، لا تنبيهات احتيال.  
  **KPIs:** معدل نجاح التحقق، صفر بلاغات استحواذ بعد التغيير.

## **4) سياسة نقل الهوية بين المنصّات (Provider Re-link)**

**الهدف:** ربط/فصل مزود دخول (Apple/Google/Email) بأمان مع الحفاظ على هوية تحليلات موحّدة.  
**المحفزات:** رغبة العميل بتغيير طريقة الدخول، توحيد مزودين، حل مشاكل وصول.  
**الإجراءات:**

* تحقق ملكية الحساب الحالي والمزود الهدف.
* تحديث جدول auth\_providers (provider, provider\_uid) وربط واحد “أساسي”.
* نقل الجلسات الآمنة، التحقق من عدم وجود حساب مكرر؛ إن وُجد ⇒ **دمج حسابات** وفق سياسة الدمج.
* تحديث هوية التحليلات وربط user\_id الموحد.  
  **DoD:** تسجيل دخول يعمل عبر المزود الجديد فقط، لا حسابات مكررة، سجلات تدقيق مكتملة.  
  **KPIs:** نجاح الربط من أول مرة، صفر حالات فقدان وصول، انخفاض الازدواجية.

ملخص آلية العمل

# **ملخص آلية العمل واعتمادياتها**

* السياسات: حذف الحساب، دمج المكرّر، تعطيل مؤقت، تصحيح PII، تغيير/استرجاع رقم الهاتف، نقل الهوية بين المنصّات.
* الاعتماديات: DB (MySQL)، Auth (Firebase/Apple/Google/Email)، Wallet/Payments (يشمل COD)، Notifications، Analytics (GA4/BigQuery)، CI/CD، Monitoring.

## **1) تغييرات قاعدة البيانات (DDL الموحّد)**

* users:  
  status ENUM('active','frozen','deleted'), deleted\_at, erased\_at, pii\_hash, anonymized\_customer\_id, merged\_into\_user\_id, merge\_at.
* phone\_history(user\_id, phone, verified\_at, changed\_at, reason)
* auth\_providers(user\_id, provider, provider\_uid, is\_primary)
* account\_merge\_jobs(id, target\_user\_id, score, state, dry\_run\_report, committed\_at)
* account\_merge\_links(source\_user\_id, target\_user\_id, merge\_job\_id)
* فهارس على: anonymized\_customer\_id, merged\_into\_user\_id, و(provider, provider\_uid).

DoD (للـDDL): الترحيلات مُعتمدة عبر CI، ولا استثناءات خارج migrations.

## **2) واجهات الـAPI (عقود مختصرة)**

* حذف: POST /account/erase → يتحقق من الرصيد/الطلبات ثم يشغّل Job الإزالة.
* تجميد/إلغاء: POST /account/freeze, POST /account/unfreeze.
* تصحيح PII: POST /account/rectify (OTP مسبق).
* تغيير رقم الهاتف: POST /account/phone/change (OTP للقديم والجديد أو بديل هوية).
* نقل الهوية: POST /account/identity/relink (تحقق مزوّدين + منع ازدواج).
* دمج: GET /account/merge/dry-run, POST /account/merge/commit (requires DRI approval).

DoD (للـAPI): مصادقة قوية، أخطاء مفسّرة، جميع العمليات idempotent.

## **3) Jobs/Cron**

* erase\_user\_job: تمويه PII، فك الربط للطرف الثالث، نقل user\_id→anonymized\_customer\_id.
* wallet\_closure\_job: تصفير المحفظة قبل الإغلاق/الحذف/الدمج.
* merge\_executor\_job: نقل العلاقات على دفعات + سجل تدقيق + إمكانية rollback خلال 24–48 ساعة.
* consent\_suppression\_job: قمع الإرسال لمن حالته frozen/deleted.

DoD: كل Job يكتب نتائج تشغيل قابلة للتدقيق + تنبيه عند الفشل.

## **4) أحداث التحليلات (Analytics Events)**

* account\_deleted, account\_frozen, account\_unfrozen, account\_rectified, phone\_changed, provider\_relinked, account\_merged.
* الخصائص الأساسية: reason, actor (user/agent/system), latency\_ms, success.

DoD: الأحداث تظهر في BigQuery خلال ≤ 5 دقائق وتُطابق عدّادات الـDB.

## **5) المراقبة والتنبيهات (Monitoring & Alerts)**

* **SLO:**
  + p95 تنفيذ erase\_user\_job ≤ 2m بعد التسوية.
  + p95 دمج الحساب ≤ 5m (بناء على الحجم).
* **تنبيهات:** فشل أي Job، ارتفاع merge rollback، رسائل تسويقية خرجت بعد deleted\_at, اختلاف العدّ بين GA4 وDB > 3%.

## **6) حوكمة التشغيل (Runbooks مختصرة)**

لكل سياسة صفحة 1–2: **الهدف، متى تُستخدم، خطوات دقيقة، رسائل جاهزة للعميل، نقاط تراجع (Rollback)، مالك DRI**.

## **7) اختبارات “اليوم صفر” (Day-0)**

* **Erase**: عميل برصيد محفظة>0 + طلب مفتوح ⇒ منع الإزالة حتى التسوية، ثم نجاح التمويه وربط anonymized\_customer\_id.
* **Merge**: حسابان بموفّق قوي (هاتف موثّق) ⇒ Dry-run يعطي diff صحيح ثم Commit ينقل كل العلاقات ويُبطل توكنات المصدر.
* **Freeze/Unfreeze**: لا طلبات جديدة أثناء frozen؛ رفع التجميد يُعيد الوصول.
* **Rectify**: تغيير بريد/اسم ⇒ يظهر diff في audit ويتزامن مع Auth/تحليلات.
* **Phone Change**: OTP مزدوج + نافذة أمان قصيرة + تحديث phone\_history.
* **Re-link**: ربط Apple→Google ⇒ لا ازدواج، وإلزام دمج إذا اكتُشف حساب آخر.

DoD: جميع سيناريوهات Day-0 “Pass” في Staging وتقرير واحد artifact من الـCI.

## **8) رسائل جاهزة (Customer-Facing)**

* قوالب مختصرة (عربي/إنجليزي) للحذف، التجميد، نجاح الدمج، تغيير الهاتف، نقل الهوية—تُحفظ في docs/runbooks/messages.md.

سياسة تنظيف الأجهزة (Device & Session Purge)

# **سياسة تنظيف الأجهزة والجلسات (Device & Session Purge)**

## **1) الهدف**

إلغاء وصول العميل فورًا عبر مسح/إبطال **الجلسات** و**توكينات OAuth** و**رموز الدفع بالإشعارات (Push Tokens)** و**بصمات الأجهزة**، مع سجل تدقيق كامل وإمكانية التكرار بدون آثار جانبية.

## **2) المحفزات**

* حذف الحساب النهائي، تجميد الحساب، دمج الحسابات، تغيير/استرجاع الهاتف، نقل الهوية بين المنصات.
* اشتباه استحواذ (ATO)، بلاغ أمن/اختراق، طلب العميل “تسجيل الخروج من كل الأجهزة”.

## **3) النطاق**

* الطبقات: Auth (جلسات/Refresh Tokens/OAuth)، Push (FCM/APNS/WebPush)، التطبيق (بصمات الأجهزة)، التحليلات (user\_pseudo\_id عند الحاجة).
* لا تُغيّر بيانات الطلبات/المعاملات؛ التأثير “وصول فقط”.

## **4) تغييرات قاعدة البيانات (DDL مختصر)**

التزم بـ utf8mb4\_0900\_ai\_ci وUTC. المعرّفات ULID (CHAR(26)). استخدم **Lookup tables** بدل ENUM (مثل: lk\_revoke\_reason, lk\_push\_token\_status, lk\_platform).

* **auth\_sessions**session\_id, user\_id, created\_at, expires\_at, last\_seen\_at,  
  ip, user\_agent, revoked\_at, revoke\_reason\_id, revoked\_by (system/user/admin).  
  *فهارس:* (user\_id), (expires\_at), (revoked\_at).
* **oauth\_tokens**token\_id, user\_id, provider, scope, issued\_at, expires\_at,  
  revoked\_at, revoke\_reason\_id, revoked\_by.  
  *فهارس:* (user\_id,provider), (expires\_at), (revoked\_at).
* **push\_devices**device\_id, user\_id, platform\_id, push\_token, push\_token\_status\_id,  
  invalidated\_at, invalidated\_reason\_id, last\_notified\_at, last\_seen\_at, app\_version, notifications\_opt\_in.  
  *فهارس:* (user\_id), (push\_token\_status\_id), (invalidated\_at).
* **devices** (اختياري لبصمة الجهاز)  
  device\_hash, user\_id, first\_seen\_at, last\_seen\_at, trust\_score, blocked\_at, block\_reason\_id.  
  *فهارس:* (user\_id), (device\_hash UNIQUE).
* **audit\_log**يسجل كل عمليات الإبطال/المسح (من/متى/ماذا/لماذا).
* **Retention**حذف سجلات الجلسات/التوكينات المنتهية بعد **90 يومًا** (Purging Job) مع بقاء سجل التدقيق لمدة **5 سنوات**.

## **5) عقود الـAPI (مختصر وواضح)**

* POST /account/sessions/purge (عميل): يتطلب **OTP**؛ يُبطل جميع الجلسات/التوكينات/Push Tokens.
* POST /admin/users/{id}/sessions/purge (مسؤول): مع سبب الإبطال.
* POST /devices/purge (عميل): مسح الأجهزة المرتبطة وتوكينات الـPush.
* **خصائص مشتركة:** idempotent، ردّ يحصي ما أُبطل، يسجّل Audit، يُطلق أحداث Analytics.

## **6) التنفيذ (الترتيب مهم)**

1. **Revoke OAuth/Refresh Tokens** أولًا (تمنع إصدار جلسات جديدة).
2. **Revoke App/Web Sessions**: وضع revoked\_at + سبب، ومسح Sessions Cache.
3. **Invalidate Push Tokens**: تعيين push\_token\_status\_id = invalid + invalidated\_at ومسحها من مزوّد الإرسال.
4. **Purge Devices**: مسح/حظر device\_hash ذات الثقة المنخفضة، وتصفير trust\_score عند ATO.
5. **Third-parties**: إبطال عبر SDK/واجهة المزود (قنوات Auth/Push).
6. **Analytics (اختياري حسب السياسة)**: إعادة تعيين user\_pseudo\_id إذا كان الهدف قطع التتبع.
7. **Audit & Notify**: سجل تدقيق + رسالة قصيرة للعميل (“تم تسجيل خروجك من كل الأجهزة”).

ملاحظة: التنفيذ **قابل للتكرار** (idempotent). أي سجل مُبطَل مسبقًا يُتجاهل دون خطأ.

## **7) Jobs/Cron**

* purge\_expired\_sessions\_job (كل 15 دقيقة): إبطال/حذف الجلسات/التوكينات المنتهية.
* push\_token\_cleanup\_job (ساعة): استخدام ردود FCM/APNS (مثل NotRegistered/InvalidToken) لتحديث push\_token\_status\_id.
* device\_quarantine\_job (ساعة): حظر الأجهزة ذات أنماط شاذة (Failed OTP متكرر، مواقع متعددة في دقائق).

## **8) المراقبة والتنبيهات**

* **SLOs**:
  + p95 زمن تنفيذ API/purge ≤ **30 ثانية**.
  + p95 زمن استكمال Jobs ≤ **5 دقائق**.
* **تنبيهات**:
  + فشل Job متكرر، ارتفاع **Active Sessions** بعد عملية purge > 0 خلال 1 دقيقة،
  + معدل Push Tokens غير الصالحــة،
  + فروقات بين عدّ الـDB وأحداث Analytics > 3%.

## **9) اختبارات Day-0 (على Staging)**

* حساب بثلاث جلسات + Push Token + OAuth:
  1. استدعاء /account/sessions/purge → تتصفر كل الجلسات والتوكينات، وتصبح حالة Push = invalid.
  2. محاولة استدعاء API بالتوكين القديم ⇒ **401**.
  3. وصول إشعار Analytics session\_revoked, device\_invalidated.
  4. تقرير CI يثبت أعداد ما قبل/بعد.

## **10) تعريف**

* جميع الجلسات/التوكينات/Push Tokens **revoked/invalid** للمستخدم المستهدف خلال **≤ 30 ثانية**.
* لا جلسات نشطة للمستخدم بعد دقيقة واحدة.
* سجل تدقيق كامل، وتنبيه نجاح في الـCI/Monitoring.
* عمليات الإبطال **idempotent** وقابلة للتدقيق.

## **11) حالات الحافة**

* **حذف حساب مع طلبات مفتوحة**: يُنفَّذ Purge فورًا (الوصول فقط)؛ تبقى المعاملات.
* **دمج حسابات**: يُنفَّذ Purge لحساب المصدر، وتُنشأ جلسة واحدة جديدة للهدف فقط عند تسجيل دخوله.
* **تجميد مؤقت**: Purge فوري + منع إنشاء جلسات جديدة حتى رفع التجميد.
* **NAT/Shared Devices**: لا نحظر IP؛ الاعتماد على device\_hash وOTP.

## **12) الملكية وSLA**

* **DRI**: فريق المنصّة/الأمن.
* **SLA**: استجابة API فورية؛ معالجة Jobs ضمن النافذة الزمنية المحددة.
* **مراجعة ربع سنوية** لإشارات “device\_quarantine” ومعايير الثقة.

سياسة إدارة الطلبات (Orders Management)

# **سياسة إدارة الطلبات (Orders Management)**

## **1) ملخص آلية العمل واعتمادياتها**

* **المبدأ:** دورة حياة طلب واضحة وقابلة للتدقيق، تفصل الطلب عن الشحنة/الدفع، وتربط المخزون عبر **inventory\_ledger** و**reservations**.
* **الاعتماديات:** DB (MySQL)، بوابة الدفع/المحفظة (يشمل COD)، الشحن/التتبع، المخزون (PIM/OMS)، التنبيهات، التحليلات، CI/CD والمراقبة.

## **2) نطاق السياسة**

تنطبق على: **orders, order\_items, order\_addresses (snapshots), order\_payments, shipments, inventory\_ledger, reservations, adjustments/discounts, ndr/rto, returns (RMA مذكورة بسياسة منفصلة).**

## **3) قرارات معمارية (ADR مختصرة)**

1. **فصل المفاهيم:** الطلب ≠ الشحنة ≠ الدفع (جداول وروابط منفصلة).
2. **حالات الطلب (lookup لا ENUM):** draft, placed, confirmed, allocated, packed, shipped, delivered, closed, cancelled.
3. **الترقيم:** order\_number مستقل عن PK وبصيغة قابلة للقراءة (سنة/بلد اختياريًا).
4. **Snapshots ثابتة:** عنوان/أسعار/ضرائب تُؤخذ **لحظة التأكيد**؛ لا تتأثر بتغيّر الكتالوج لاحقًا.
5. **المخزون:** حجز عند checkout (TTL)، **تخصيص** عند confirmed/allocated، وخصم عند shipped.
6. **الدفع:** prepaid تُلتقط/تُسجّل قبل الشحن؛ COD تُسوّى لاحقًا (سياسة COD).
7. **الإلغاء الذكي:** حسب مرحلة المعالجة (تفصيل بالأسفل) مع أثر تلقائي على الحجز/التخصيص/الخصم.
8. **التعديلات:** أي خصم/قسيمة/فرق شحن يُسجَّل في **order\_adjustments** (دائن/مدين).
9. **Idempotency:** لكل عمليات الكتابة (Header Idempotency-Key).
10. **المرونة التشغيلية:** السماح بالشحنات المتعددة (Split), الدمج للشحن فقط (لا دمج طلبات ماليًا).
11. **نوافذ زمنية:** تعريف واضحة لـ cancel/change address/ndrsla.
12. **الشفافية:** سجل أحداث **order\_events** إلزامي لكل انتقال حالة.

**4) تغييرات المخطط (DDL المعياري – مختصر)**

* **orders**id (PK), order\_number (UNIQUE), user\_id|null, anonymized\_customer\_id|null,  
  state\_id, sub\_state\_id|null,  
  placed\_at, confirmed\_at, allocated\_at, packed\_at, shipped\_at, delivered\_at, closed\_at, cancelled\_at,  
  مبالغ: subtotal, shipping, tax, discount\_total, grand\_total, currency, fx\_rate,  
  أعلام: is\_cod, is\_preorder, is\_backorder,  
  مراجع: shipping\_profile\_id, payment\_profile\_id.  
  *فهارس:* (state\_id), (placed\_at), (confirmed\_at), (shipped\_at).
* **order\_items**order\_id (FK), sku\_id, qty\_ordered, qty\_allocated, qty\_shipped, qty\_cancelled,  
  أسعار لقطة: unit\_price, unit\_discount, unit\_tax.  
  *فهارس:* (order\_id), (sku\_id).
* **order\_addresses** (Snapshot)  
  order\_id, type ('shipping'/'billing'), name, phone\_masked, city, region, country, address\_lines (masked).
* **order\_payments**order\_id, payment\_method, provider\_txn\_id, amount, status, captured\_at|refunded\_at, wallet\_ledger\_id|null.
* **order\_adjustments**order\_id, type (coupon/promo/shipping\_diff/price\_adjust), amount (+/-), reason, actor.
* **order\_events**order\_id, event\_type, payload(JSON), created\_at, actor.
* **shipments** / **shipment\_items** (منفصلة)  
  تربط بـ order\_id مع carrier, tracking\_no, state.
* **ndr / rto**جداول لحالات فشل التسليم والعودة للمرسل مع سبب/قرار.
* **reservations / inventory\_ledger**كما اعتمدتم سابقًا: reservation TTL، ledger لحركات allocate/ship/release/return.

**ملاحظة:** كل حالات/أنواع تُدار عبر **lookup tables** + FKs (لا ENUM)، وفهارس زمنية للحالة.

## **5) عقود الـAPI (مختصرة)**

* POST /orders (create draft/placed) — به **Idempotency-Key**.
* POST /orders/{id}/confirm — يجمّد الأسعار/العناوين ويبدأ **allocate**.
* POST /orders/{id}/cancel — إلغاء ذكي حسب الحالة.
* POST /orders/{id}/address/change — ضمن نافذة محددة قبل packed.
* POST /orders/{id}/split — تكوين **shipment** جزئي.
* POST /orders/{id}/merge-for-shipping — دمج شحني (لنفس العميل/العنوان/الإطار الزمني).
* POST /orders/{id}/ndr — تسجيل فشل تسليم مع قرار (إعادة شحن/استرداد).
* POST /orders/{id}/rto — معالجة عودة للمرسل.
* POST /orders/{id}/price-adjust — تعديل بعد الشراء بسياسة محكومة.
* **Webhook/Events:** order\_confirmed/shipped/delivered/cancelled/ndr/rto.

## **6) نوافذ وسياسات التشغيل**

### **6.1 الإلغاء الذكي (Smart Cancel)**

* **قبل confirmed:** إلغاء فوري → **release reservation** + عكس أي حجز دفع.
* **بين confirmed وpacked:** إلغاء إن لم يبدأ الالتقاط (picking)؛ إن بدأ: إلغاء جزئي حسب **items** المتاحة.
* **بعد shipped:** لا إلغاء؛ يتحول إلى **NDR/RTO/RMA**.
* **Preorder/Backorder:** مسموح حتى **X أيام** قبل ETA.

### **6.2 تغيير العنوان**

* مسموح حتى **قبل packed**؛ بعد ذلك عبر مركز الشحن (إن أمكن) مع فرق كلفة في **order\_adjustments**.

### **6.3 Split Shipment**

* إذا كانت **بعض العناصر جاهزة**: يُنشأ **shipment** جزئي؛ يظل الطلب مفتوحًا للباقي.

### **6.4 NDR (لم يتم التسليم) / RTO (عاد للمرسل)**

* **NDR:** تسجيل السبب (لا يجيب، عنوان خاطئ…)، قرار آلي وفق قواعد: إعادة محاولة/تأكيد عنوان/تحويل لـ RTO.
* **RTO:** عند العودة، **استرجاع** للعميل (أصل الدفع أو محفظة) بعد فحص الحالة، وإرجاع المخزون إلى **ledger** بالسبب المناسب.

### **6.5 COD**

* محاولات تسليم محدودة (n محاولات). تكرار فشل **→ قائمة مراقبة/حجب COD** لفترة.
* تسوية السائق/الناقل ترتبط بـ **cod\_settlements** (موجودة في سياساتكم).

### **6.6 تعديل السعر بعد الشراء**

* نافذة محدودة (مثلاً 72 ساعة) مع حدود قصوى وعدالة (لا تُخالف العروض). تُسجّل في **order\_adjustments**.

## **7) المخزون والدفاتر (Rules)**

* **Cart → Reservation (TTL)**
* **Confirm → Allocate** (تعبئة qty\_allocated)
* **Pack/Ship → Ledger: ship** (خصم فعلي)
* **Cancel → Ledger: release**
* **RTO/Return → Ledger: return\_to\_stock/damaged** حسب الحالة.

**8) الاتصالات والتحليلات**

* إشعارات: order\_confirmed, order\_shipped (مع تتبع), out\_for\_delivery, delivered, ndr\_prompt, rto\_refund\_issued.
* Analytics Events: order\_created/confirmed/fulfilled/cancelled/ndr/rto/split/price\_adjust مع خصائص (latency\_ms, reason, cod, preorder).

## **9) المراقبة وSLO/KPI**

* **SLOs:**
  + p95 **confirm→allocate** ≤ 2m، **packed SLA** ≤ 12h، **ship→deliver** حسب شركة الشحن.
  + أخطاء API الحرجة < 0.1%.
* **KPIs:**
  + **Cancel rate** قبل الشحن، **NDR rate**، **RTO rate**، **On-time delivery %**، **Split ratio**، **Avg lead time**، **COD failure rate**، **Refund TAT**.
* **تنبيهات:** ارتفاع NDR/RTO فوق العتبة، تأخر تحديث المخزون، انحراف p95.

**10) اختبارات Day-0 (على Staging)**

1. **Confirm→Allocate→Pack→Ship→Deliver** (مسار سعيد) مع تطابق ledger.
2. **Cancel قبل و/أو بعد confirm** (تحقق release/partial cancel).
3. **Split Shipment** لعنصرين (واحد جاهز/واحد متأخر).
4. **NDR→RTO** مع استرداد وقيود ledger المناسبة.
5. **COD فشل مرتين** → إدراج في قائمة المراقبة.
6. **Address Change قبل packed** ينعكس على الشحنة.
7. **Price Adjust** يظهر في adjustments ويؤثر على الإجمالي بدقة.

**DoD:** كل السيناريوهات **Pass**، وتقرير CI يضم: حالات قبل/بعد، أرقام ledger، زمنيات SLO.

**11) تعريف**

* حالات الطلب تعمل **نهاية إلى نهاية** مع سجل أحداث كامل.
* الحجز/التخصيص/الخصم/الإرجاع في **ledger** صحّتهم مؤكدة.
* الانعكاس على الدفع/المحفظة/الشحن يتم تلقائيًا حسب الحالة.
* SLO/KPI قيد المراقبة مع تنبيهات فعّالة.

**12) الملكية والحَوْكمة**

* **DRI:** فريق العمليات/المنصّة.
* **Runbooks:** إلغاء ذكي، NDR/RTO، Split، تغيير عنوان، تسوية COD.
* **مراجعة ربع سنوية:** ضبط النوافذ الزمنية، عتبات NDR/RTO، قواعد COD، وأسعار الشحن.

سياسة إدارة طلبات المرتجعات والاستبدال (RMA)

**سياسة إدارة طلبات المرتجعات والاستبدال (RMA)**

* **المبدأ:** معالجة مرتجعات/استبدالات بعدالة وسرعة، مع فصل واضح بين الطلب والشحنة والدفع، وربط محكم بالمخزون عبر **inventory\_ledger** و**QC**.
* **الاعتماديات:** DB (MySQL)، الدفع/المحفظة (يشمل COD)، الشحن والتتبّع، PIM/Inventory، Notifications، Analytics، CI/CD، Monitoring.

## **1) نطاق السياسة**

تنطبق على: **rmas, rma\_items, rma\_events, returns, exchanges, order\_items, shipments, inventory\_ledger, wallet\_ledger, coupons/loyalty**.

## **2) قرارات معمارية (ADR مختصرة)**

1. **فصل المفاهيم:** RMA ككيان مستقل؛ لا تعدّل الطلب الأصلي مباشرة.
2. **حالات RMA (lookup لا ENUM):** requested, approved, in\_transit\_to\_dc, received\_qc, approved\_refund, approved\_exchange, rejected, refunded, exchanged, closed.
3. **Snapshots ثابتة:** أسعار/ضرائب/خصومات تُلتقط لحظة إنشاء RMA لضمان دقة الحسابات.
4. **المخزون:** كل مرتجع يمر عبر **QC**؛ المقبول الجاهز للبيع → return\_to\_stock، التالف → damaged/scrap.
5. **الاستبدال:** شحنة بديلة **بعد** استلام/قبول المرتجع، أو **Cross-Ship** بشروط (مذكورة أدناه).
6. **الماليات:** الحسابات **Pro-Rata** (بنود/كوبونات/شحن) + تسوية الولاء/الكوبونات + أولوية ردّ الأموال على وسيلة الدفع الأصلية ثم المحفظة.
7. **Idempotency & Audit:** لكل عمليات الكتابة؛ سجل أحداث إلزامي.
8. **إطارات زمنية قابلة للتهيئة:** نوافذ طلب RMA، مهلة شحن المرتجع، سقوف الاسترداد، إلخ.

## **3) شروط الأهلية وقبول RMA**

* **نافذة الطلب:** خلال **X يومًا** من الاستلام (قابلة للتهيئة؛ اقترح 7–14).
* **الحالة:** غير مُستخدَم/بقِطعِه وملحقاته وملصقاته.
* **ممنوعات لأسباب صحية:** الملابس الداخلية/الجوارب/مستحضرات التجميل المفتوحة (إلا عيب مصنعي موثّق).
* **أدلة:** صور/فيديو عند الطلب وسبب واضح (مقاس/عيب/وصف مختلف…).
* **الاستثناءات:** العيب المصنعي خارج النافذة الزمنية يُقبل ضمن ضمان الصنع إن وُجد.

## **4) تغييرات المخطط (DDL مختصر)**

* **rmas**id, order\_id, user\_id|null, anonymized\_customer\_id|null, state\_id, reason\_id, created\_at, approved\_at, received\_at, closed\_at, resolution ('refund'|'exchange'), notes.
* **rma\_items**rma\_id, order\_item\_id, sku\_id, qty, item\_condition\_id ('new','opened','defective','wrong\_item'), qc\_result\_id ('pass','repair','reject'), refund\_amount, restock\_fee\_amount.
* **rma\_events**rma\_id, event\_type, payload(JSON), created\_at, actor.
* **returns** (شحنة المرتجع)  
  rma\_id, carrier, tracking\_no, state\_id, label\_issued\_at, expected\_by, received\_at.
* **exchanges** (البديل)  
  rma\_id, new\_sku\_id, qty, shipment\_id|null, cross\_ship(bool), deposit\_wallet\_ledger\_id|null.
* **lookup tables**: lk\_rma\_state, lk\_rma\_reason, lk\_item\_condition, lk\_qc\_result.
* **فهارس أساسية:** على state\_id, created\_at, received\_at, وorder\_id.

**Inventory Ledger:** أسباب حركات: rma\_return\_to\_stock, rma\_damaged, rma\_missing\_parts, rma\_scrap, exchange\_out.

## **5) عقود الـAPI (مختصر)**

* POST /rma إنشاء RMA (سلة بنود + سبب + صور).
* POST /rma/{id}/approve / .../reject.
* POST /rma/{id}/label إصدار ملصق شحن مرتجع (إن متاح).
* POST /rma/{id}/receive استلام المرتجع وتسجيل **QC**.
* POST /rma/{id}/resolve/refund حساب/تنفيذ الاسترداد.
* POST /rma/{id}/resolve/exchange إنشاء/إرسال البديل.
* Webhooks/Events: rma\_approved/received/qc\_pass/qc\_fail/refunded/exchanged/rejected/closed.

## **6) التدفق التنفيذي (مختصر)**

1. **طلب RMA:** التحقق من الأهلية + إنشاء بنود + التقاط Snapshot الأسعار/الخصومات.
2. **الموافقة وإصدار الملصق (اختياري):** حالة approved مع مهلة شحن المرتجع **Y يومًا**.
3. **الشحن إلى مركز الإرجاع:** حالة in\_transit\_to\_dc.
4. **الاستلام وQC:** حالة received\_qc → لكل بند: qc\_result.
5. **قرار:**
   * **Refund:** approved\_refund → احتساب استرداد (Pro-Rata) → تنفيذ ردّ الأموال → refunded.
   * **Exchange:** approved\_exchange → تحضير البديل → شحن → exchanged.
   * **Reject:** لعدم الأهلية/استخدام/ضرر غير مصنعي → إخطار العميل بخيارات (إرجاع على حسابه/إتلاف).
6. **إغلاق:** closed بعد اكتمال الماليات/المخزون وإشعارات العميل.

## **7) قواعد الحساب المالي (مختصر ودقيق)**

* **Pro-Rata للخصومات/الكوبونات:** توزيع الخصم الإجمالي على البنود حسب **قيمة السطر**، واسترداد الجزء الموافق للبنود المرتجعة فقط.
* **الشحن:**
  + خطأ منّا/عيب مصنعي: ردّ **الشحن ذهابًا وإيابًا**.
  + تغيير رأي/مقاس: ردّ قيمة السلعة فقط (يمكن خصم الشحن/ملصق المرتجع).
* **COD:** الاسترداد إلى **المحفظة** افتراضيًا؛ بديل تحويل بنكي خلال **≤ Z يومًا**.
* **رسوم إعادة التخزين (Restock):** % ثابت أو حدّ مقطوع للبنود غير المعيبة (اختياري)، لا تُطبَّق على العيوب.
* **الولاء/القسائم:** خصم النقاط/القسائم التي كُسبت من البنود المرتجعة، وإرجاع القسائم المستخدمة وفق شروطها (متاحة/منتهية).
* **الأولوية:** نفس وسيلة الدفع → المحفظة → تحويل بنكي.

## **8) الاستبدال (Exchange)**

* **عادي:** شحن البديل بعد received\_qc (قبول).
* **Cross-Ship (اختياري):** شحن البديل **قبل** الاستلام مقابل **تأمين** (حجز بطاقة/إيداع محفظة)، وإلا يُحوّل إلى RMA قياسي.
* **فرق السعر:** تحصيل/ردّ الفرق عبر **order\_adjustments**.

## **9) المخزون والدفاتر (Rules)**

* **QC Pass:** زيادة إلى **stock\_available** عبر ledger: rma\_return\_to\_stock.
* **QC Fail – Damaged:** ledger: rma\_damaged (مستودع تالِف/سكراب).
* **Missing Parts:** ledger: rma\_missing\_parts + إمكانية خصم **restock\_fee**.
* **Exchange Out:** خصم من المخزون البديل ledger: exchange\_out.
* **RTO ↔ RMA:** إن عاد الطلب للمرسل ثم طلب RMA، تُحسب كـRMA جديدة على البنود المُستلمة.

## **10) نوافذ وسياسات التشغيل**

* **طلب RMA:** خلال **X** يومًا من الاستلام.
* **إرسال المرتجع بعد الموافقة:** خلال **Y** يومًا، وإلا تُغلق تلقائيًا.
* **TAT المعالجة بعد الاستلام:** ≤ **2 يوم عمل** لـQC، ≤ **5 أيام** للاسترداد البنكي.
* **Cross-Ship:** مسموح لفئات محدّدة وبحد أقصى **N** وحدات.

## **11) مكافحة إساءة الاستخدام**

* كشف نمط **High RMA Rate**، **Wardrobing** (استخدام ثم إرجاع)، **Size-cycling**؛ تطبيق:
  + تقليل حقوق الاسترجاع، أو طلب **رسوم شحن/إيداع** مسبق، أو **حجب COD** مؤقتًا.
* قائمة مراقبة + مراجعة بشرية للحالات الحدّية.

## **12) الاتصالات والتحليلات**

* إشعارات: rma\_approved, label\_issued, rma\_received\_qc, refund\_issued, exchange\_shipped/ delivered, rma\_rejected.
* Analytics Events: rma\_requested/approved/received/refunded/exchanged/rejected بخصائص (reason, items, latency\_ms, refund\_amount).

## **13) المراقبة وSLO/KPI**

* **SLOs:** p95 **استلام→QC** ≤ 48h، p95 **QC→Refund** ≤ 24h، أخطاء API < 0.1%.
* **KPIs:** **RMA rate%**، **Refund TAT**، **QC pass%**، **Resale lead time** للمرتجعات المقبولة، **Shrinkage%** (تالف/مفقود)، **Cross-Ship failure%**.
* **تنبيهات:** ارتفاع **RMA rate** فوق العتبة، تأخّر الاسترداد، تزايد **damaged%**.

## **14) اختبارات Day-0 (Staging)**

1. **RMA Refund**: بندان مع كوبون مشترك → استرداد Pro-Rata + تحديث الولاء.
2. **RMA Exchange**: استبدال مقاس لنفس SKU → خصم/تحصيل فرق إن وُجد.
3. **QC Pass/Fail**: بند يمرّ/يفشل → ledger صحيح، ورسوم restock على غير المعيب فقط.
4. **COD Refund**: استرداد لمحفظة ثم تحويل بنكي.
5. **Cross-Ship**: حجز تأمين، ثم استلام المرتجع؛ في حالة عدم وصول المرتجع خلال المهلة تُحوَّل العملية إلى بيع نهائي.
6. **Reject**: بند مستخدم/بدون ملصقات → رفض مُعلّل وخيارات عميل.

## **15) تعريف**

* حالات RMA تعمل من الطلب إلى الإغلاق مع **audit trail**.
* الحساب المالي/Pro-Rata والولاء/الكوبونات يتطابقان مع الإجماليات.
* تحديثات **inventory\_ledger** دقيقة لكل نتيجة QC.
* SLO/KPI قيد المراقبة والتنبيهات مفعّلة.

## **16) الملكية والحَوْكمة**

* **DRI:** فريق العمليات/المنصّة.
* **Runbooks:** سيناريوهات (Refund/Exchange/Cross-Ship/Reject).
* **مراجعة ربع سنوية:** ضبط النوافذ، سياسات الرسوم، عتبات إساءة الاستخدام، وفئات Cross-Ship.

سياسة المحفظة (Wallet Policy)

# **سياسة المحفظة (Wallet Policy)**

* **المبدأ:** المحفظة مصدر رصيد آمن وشفاف للعميل، تُدار عبر دفتر قيود **Append-Only** (wallet\_ledger) مع حجز مبالغ (holds) وتسويات صرف (payouts).
* **الاعتماديات:** DB (MySQL)، بوابة الدفع، COD، الشحن/RMA، الطلبات، الإشعارات، التحليلات، CI/CD، المراقبة.

## **1) النطاق (Scope – DB)**

* **wallet\_accounts**, **wallet\_ledger**, **wallet\_holds**, **wallet\_payouts** (+ جداول lookup للأسباب/الحالات/الوسائل).
* الربط مع: **orders / rmas / cod\_settlements / payment\_disputes / audit\_log**.

## **2) قرارات معمارية (ADR مختصرة)**

1. **مصدر الحقيقة = wallet\_ledger فقط** (Append-Only). يمكن استخدام balance\_cached كـ Snapshot يُعاد التحقق منه دوريًا.
2. **Idempotency إلزامي** لكل Credit/Debit/Hold/Capture/Payout (مفتاح فريد لكل عملية خارجية).
3. **العملة:** حقل currency وfx\_rate على مستوى القيد؛ افتراضيًا عملة واحدة، مع دعم متعدّد عند الحاجة.
4. **الرصيد السالب:** ممنوع افتراضيًا؛ يُسمح فقط بسيناريوهات **Chargeback/Fees** مع علم واضح allow\_negative=false (إلا إن فُعّل صراحة).
5. **الأولوية في الاسترداد:**
   * مدفوعات مسبقة: إلى وسيلة الدفع الأصلية، وإن تعذّر ⇒ المحفظة.
   * COD: إلى **المحفظة** افتراضيًا، مع خيار تحويل بنكي لاحق.
6. **تمييز الأموال:** **نقد/مكافآت**: أرصدة ترويجية Expirable تُسجَّل بأسباب مميّزة وتنتهي تلقائيًا (TTL).
7. **حجز/تجميد (Holds):** عند الحاجة (Cross-Ship/تحقق)، تُحوَّل إلى **Capture (Debit)** أو **Release** تلقائيًا مع TTL.
8. **التسوية والصرف (Payouts):** تُخصم من المحفظة كـ Debit مع سجل حالة (requested→processing→succeeded|failed) ومطابقة بنكية.
9. **الامتثال والأمان:** حدود يومية/شهرية، فحوص سرعات/أجهزة، متطلبات KYC للصرف.

## **3) تغييرات المخطط (DDL مختصر)**

* **wallet\_accounts**id (PK), user\_id|null, anonymized\_customer\_id|null, status ENUM('active','frozen','closed'),  
  currency, balance\_cached DECIMAL(18,2), allow\_negative BOOL DEFAULT 0,  
  طوابع: created\_at, updated\_at, closed\_at.  
  *فهارس:* (user\_id), (anonymized\_customer\_id), (status).
* **wallet\_ledger** (Append-Only)  
  id, wallet\_id (FK), entry\_type ('credit'|'debit'), reason\_id (FK lookup),  
  amount DECIMAL(18,2), currency, fx\_rate DECIMAL(12,6),  
  related\_type (order|rma|payout|settlement|dispute|adjustment), related\_id,  
  idempotency\_key UNIQUE, actor ('system'|'user'|'admin'), created\_at.  
  *فهارس:* (wallet\_id,created\_at), (idempotency\_key), (related\_type,related\_id).
* **wallet\_holds**id, wallet\_id, amount, reason\_id, created\_at, expires\_at,  
  captured\_at|null, released\_at|null, related\_type, related\_id, idempotency\_key UNIQUE.  
  *فهارس:* (wallet\_id), (expires\_at), (captured\_at).
* **wallet\_payouts**id, wallet\_id, method\_id (bank|mobile\_money|voucher), amount, fee\_amount,  
  status\_id (requested|processing|succeeded|failed|reversed),  
  requested\_at, processed\_at|null, provider\_ref|null, failure\_reason|null, idempotency\_key UNIQUE.  
  *فهارس:* (wallet\_id,status\_id), (requested\_at).
* **lookups**: lk\_wallet\_reason, lk\_payout\_method, lk\_payout\_status.
* **audit\_log**: يسجّل كل عمليات الائتمان/الخصم/الحجز/الصرف والتراجع.

**قاعدة:** لا **UPDATE/DELETE** على ledger؛ أي تصحيح يتم بقيد عكسي (Reverse Entry).

## **4) القواعد (Invariants)**

* **I-1:** مجموع **Credits − Debits − Captured Holds** = رصيد المحفظة (≈ balance\_cached مع إعادة مزامنة).
* **I-2:** لا قيود Debit تُسمح إن كان الناتج < 0 (إلا إذا allow\_negative=1 والسبب ضمن قائمة مسموحة).
* **I-3:** idempotency\_key فريد عالميًا لكل عملية خارجية (منع التكرار).
* **I-4:** لا **Close** لمحفظة ورصيدها ≠ 0.
* **I-5:** لا **Erase Account** قبل قيد wallet\_closure ورصيد = 0 (متكاملة مع سياسة الحذف).

## **5) التدفقات التشغيلية (Flows)**

* **Top-Up (Credit):** via بوابة الدفع ⇒ قيد credit:topup بـ idempotency + تحديث balance\_cached.
* **Refund In (Credit):** من RMA/Order ⇒ credit:refund مع related\_type='rma'|'order'.
* **Purchase/Adjustment (Debit):** debit:purchase أو debit:adjustment.
* **Holds:** إنشاء Hold (TTL) ⇒ لاحقًا **Capture** (يُنشئ Debit + يغلق الـHold) أو **Release** عند الإلغاء/الانتهاء.
* **Promotions/Rewards (Credit Expirable):** credit:promo مع expires\_at; Job يُنشئ debit:promo\_expire عند الانتهاء.
* **Payout (Withdrawal):** إنشاء سجل في wallet\_payouts (requested) + قيد debit:payout؛ عند فشل المزود ⇒ credit:payout\_reversal.
* **Chargeback/Dispute (Debit):** عند خسارة نزاع دفع ⇒ debit:chargeback وقد ينتج رصيد سالب + تجميد محفظة.
* **COD Refund:** إلى المحفظة افتراضيًا؛ طلب تحويل بنكي يخصم debit:payout عند التنفيذ.

## **6) واجهات الـAPI (عقود مختصرة)**

* POST /wallet/topup (idempotent)
* POST /wallet/refund (من RMA/Order)
* POST /wallet/hold → POST /wallet/hold/{id}/capture | /release
* POST /wallet/payout → متابعة /wallet/payout/{id} (status)
* GET /wallet/ledger (ترشيح حسب التاريخ/النوع)

كل العمليات تتطلب **Idempotency-Key** وتمر بمعاملة مع **SELECT … FOR UPDATE** على صف المحفظة لمنع السباق.

## **7) المخاطر والضوابط (Risk & Controls)**

* **حدود** يومية/شهرية لـ Top-Up/Payout/Refund-In.
* **كشف سرعة/أجهزة**: حد لعدد العمليات/جهاز/ساعة؛ ربط بـ device\_hash.
* **KYC/AML للصرف:** وثائق مطلوبة للمبالغ الكبيرة؛ قوائم حظر.
* **تجميد المحفظة** عند رصيد سالب لفترة أو نمط احتيالي.

## **8) المراقبة والمؤشرات (SLO/KPIs)**

* **SLOs:** p95 تنفيذ عمليات المحفظة ≤ 200ms، p95 معالجة Payout ≤ 2 يوم عمل.
* **KPIs:** رصيد مطلوب (Wallet Liability)، **Refund TAT**، **Payout Success%**، **Dispute Rate**، نسبة **Expired Promo%**، تبنّي المحفظة % من الاستردادات.

## **9) اختبارات Day-0 (Staging)**

1. **Idempotency:** إرسال Top-Up مكرر بنفس المفتاح ⇒ قيد واحد فقط.
2. **Concurrency:** عمليتا Capture متزامنتان على نفس الـHold ⇒ واحدة فقط تنجح.
3. **Refund→Payout:** ردّ RMA إلى المحفظة ثم طلب Payout؛ عند فشل المزود تُسجَّل credit:payout\_reversal.
4. **Promo Expiry:** إنشاء credit:promo مع TTL قصير ⇒ ينشأ debit:promo\_expire تلقائيًا.
5. **Negative Guard:** Debit يجعل الرصيد سالبًا ⇒ يُرفض إلا مع allow\_negative=1 وسبب مسموح.
6. **Erase/Close Gate:** محاولة إغلاق/حذف مع رصيد ≠ 0 ⇒ مرفوضة.

**DoD:** كل السيناريوهات Pass، وتقرير CI يحوي: صحة الرصيد، عدم وجود ازدواج، أزمنة التنفيذ.

## **10) تعريف (Definition of Done)**

* المخطط (DDL) والجداول المرجعية والقيود مُطبّقة، و**ledger** يعمل Append-Only.
* عمليات API idempotent، وقفل تنافسي صحيح، وتنبيهات مراقبة مفعّلة.
* ربط كامل مع الطلبات/RMA/COD/Disputes، وسياسة الحذف (Closure قبل Erase).
* تقارير الـAnalytics تعكس أحداث المحفظة خلال ≤ 5 دقائق.

## **11) حالات الحافة (Edge Cases)**

* **دمج حسابات:** نقل محفظة المصدر إلى الهدف (wallet\_merge قيد توضيحي).
* **تجميد الحساب/Legal Hold:** منع Payout/Erase؛ يسمح Refund-In فقط.
* **تغيّر العملة:** تُحفظ القيم بالعملة الأصلية + fx\_rate; التقارير توحّد عبر تحويل تحليلي.
* **انقطاع مزود الدفع:** Queue داخلي وإعادة محاولة مع Backoff وتتبّع idempotency\_key.

## **12) الملكية والحَوْكمة**

* **DRI:** فريق المنصّة/المالية.
* **Runbooks:** Top-Up Failures، Payout Reconciliation، Chargeback، Promo Expiry، Wallet Merge/Closure.
* **مراجعة ربع سنوية:** حدود/سياسات المخاطر، رسوم الصرف، مدد Expiry.

سياسة التحليلات والإحصائيات (Analytics )

**سياسة التحليلات والإحصائيات (Analytics & Statistics Policy)**

* **المبدأ:** سلوك المستخدم يُقاس عبر أحداث مُوحّدة وقابلة للتدقيق، مع فصلٍ واضح بين “مصدر الحقيقة المالي” (الـDB) و”تحليلات السلوك” (الحدث/الجلسات).
* **الاعتماديات:** DB (MySQL)، SDKs (Firebase/GA4/Web), Server-side Events، BigQuery (Raw/Staging/Marts)، ETL/ELT، Consent/DSAR، CI/CD، Monitoring.

## **1) النطاق (Scope – Data/DB)**

* **جمع الأحداث:** التطبيق (Firebase), الويب (GA4/gtag), الخادم (server-events).
* **المستودع التحليلي:** BigQuery (أو بديله) بثلاث طبقات: raw\_\*، clean\_\*، marts\_\*.
* **جداول الربط مع الـDB:** user\_identity\_map, product\_dim, order\_fact, rma\_fact, wallet\_fact, marketing\_dim (UTM).
* **حوكمة الخصوصية:** consent\_logs, dnt\_flags, deletion\_queue (طلبات erase).

## **2) قرارات معمارية (ADR مختصرة)**

1. **مصدر الحقيقة للإيراد/الطلبات = DB** (order\_fact/wallet\_fact). التحليلات لقياس السلوك/القمع (funnels).
2. **هوية موحّدة:** استخدام user\_id أينما أمكن + ربط user\_pseudo\_id المجهول عبر user\_identity\_map (anonymous→known).
3. **Taxonomy للأحداث:** قاموس أحداث مُوحّد (أسماء/حقول إلزامية)، لا أحداث خارج القائمة.
4. **Idempotency & De-dup:** كل حدث يحمل event\_id (ULID)؛ تُزال الازدواجات في clean\_\*.
5. **الطابع الزمني:** الاعتماد على server\_timestamp، وتخزين client\_timestamp للمعلومة فقط؛ المنطقة الزمنية **UTC**.
6. **تقسيم/عنقدة (DW):** التقسيم بالـdate، العنقدة حسب (event\_name, user\_pseudo\_id, source).
7. **الامتثال:** عدم إرسال PII (بريد/هاتف) في الأحداث؛ يُسمح بـ hash مُملّح داخليًا فقط عند الضرورة.
8. **الهوية التسويقية:** UTM موحَّدة (lowercase/normalized)، نماذج Attribution محددة (First/Last touch).

## **3) مخطط البيانات (DDL مختصر – Data Warehouse)**

صِف المخطط داخل المستند؛ التنفيذ الفعلي في الـDW.

* **raw\_events\_app / raw\_events\_web**event\_id, event\_name, client\_ts, server\_ts, event\_date, user\_pseudo\_id, device\_id, session\_id, user\_id\_nullable, source(app/web), params(JSON)  
  *Partition:* event\_date — *Cluster:* (event\_name, user\_pseudo\_id)
* **clean\_events**نفس الحقول + is\_duplicate, is\_bot, schema\_valid. تُفلتر الازدواج/الروبوت.
* **sessions**session\_id, user\_pseudo\_id, user\_id\_nullable, source, start\_ts, end\_ts, session\_duration, screen\_views, bounces\_flag
* **user\_identity\_map**user\_pseudo\_id, user\_id, first\_seen\_ts, last\_seen\_ts, link\_reason (login/purchase/sdk)
* **marketing\_dim (utm)**utm\_source, utm\_medium, utm\_campaign, utm\_content, utm\_term, normalized\_flags, first\_touch\_ts, last\_touch\_ts, attribution\_window\_days
* **order\_fact / rma\_fact / wallet\_fact** *(مأخوذة من الـDB عبر ELT)*مفاتيح الربط: user\_id, order\_id, … + مبالغ/حالات مع طوابع زمنية.
* **funnels\_mart**يومي/أسبوعي: خطوات القمع (view\_item → add\_to\_cart → begin\_checkout → add\_payment → purchase).
* **kpi\_daily**date, users\_active, sessions, cvrs, aov, revenue\_db, revenue\_events, delta\_revenue\_pct, cac, roas, ndr\_rate, rto\_rate
* **consent\_logs / dnt\_flags / deletion\_queue**لتطبيق القبول/عدم التتبع وطلبات المحو على المسارات التحليلية.

## **4) قاموس الأحداث (Taxonomy مختصر)**

**أحداث أساسية (أمثلة وحقول إلزامية):**

* app\_open (app\_version, os, locale)
* view\_item (sku\_id, price, currency)
* add\_to\_cart (sku\_id, qty, price)
* begin\_checkout (cart\_value, items\_count)
* add\_payment\_info (method\_type)
* purchase (order\_id, value, currency, items[]) — **يجب** إرسالها server-side أيضًا.
* بعد الشراء: rma\_requested/refunded, ndr/rto, wallet\_credit/debit, coupon\_applied/failed

**قاعدة:** أي حدث جديد يتطلب تحديث القاموس + اختبار Day-0 قبل الإرسال للإنتاج.

## **5) القواعد (Invariants)**

* **I-1:** event\_id فريد عالميًا في clean\_events.
* **I-2:** لا PII في params؛ المسموح فقط معرفات داخلية/Hashات غير قابلة للعكس.
* **I-3:** purchase.value في الأحداث ≈ إيراد الـDB لنفس الفترة مع إنحراف ≤ **3%** (يُفسَّر بالضرائب/الخصومات/الوقت).
* **I-4:** كل purchase server-side يولِّد حدثًا مطابقًا client-side أو يُوسم source='server'.
* **I-5:** جلسة واحدة لا تتجاوز 30 دقيقة عدم نشاط (sessionization ثابت).

## **6) Attribution (اختصار واضح)**

* **First Touch (90d)** + **Last Non-Direct (7d)**.
* تطبيع UTM (قائمة بيضاء للـmedium: cpc, social, email, affiliate, display, direct).
* التنازع يُحل حسب “آخر لمسة” إن كانت ضمن النافذة؛ وإلا “أول لمسة”.

## **7) الخصوصية والامتثال**

* **Consent Mode:** لا تُجمع أحداث التسويق أو تُخفَّض الدقة دون موافقة.
* **DNT:** عند تفعيل dnt\_flags تُوقف الـSDKs وتُحذف هوية التحليلات (identity reset).
* **DSAR/Erase:** حذف/إخفاء هوية التحليلات المرتبطة بـ user\_id وuser\_pseudo\_id في clean\_\*، مع إبقاء المجاميع المجهولة.
* **Non-Prod Masking:** عينات مُقنَّعة فقط لبيئات الاختبار.

## **8) جودة البيانات (DQ) والاختبارات**

**اختبارات يومية تلقائية:**

* **Schema Conformance:** جميع الأحداث تمرّ بـ JSON schema.
* **Volume Anomaly:** إنخفاض/ارتفاع > 20% مقارنة بمتوسط 7 أيام.
* **Join Rates:** معدل ربط view\_item↔product\_dim ≥ 99%، وpurchase↔order\_fact ≥ 99.5%.
* **Dedup Ratio:** نسبة الازدواج < 1%.
* **Revenue Delta:** |revenue\_events − revenue\_db| / revenue\_db ≤ 3%.

## **9) SLO/KPIs والمراقبة**

* **SLO Latency:** من جمع الحدث إلى توفره في clean\_\* ≤ **15 دقيقة**، وفي marts\_\* ≤ **60 دقيقة**.
* **KPIs:** DAU/WAU/MAU، CVR per step، AOV، ROAS، CAC، Retention (D1/D7/D30)، NDR/RTO Rates، Funnel Drop-offs.
* **Alerts:** تأخير خط الأنابيب > 30 دقيقة، delta الإيراد > 3%، Join Rate < 98.5%، تكلفة BigQuery خارجة عن الحد، فشل تحميل يومي.

## **10) التدفقات التشغيلية (ELT/Jobs مختصرة)**

* **Ingest:** Streaming إلى raw\_\* (idempotent).
* **Clean & Dedup:** تحويل إلى clean\_\* مع قواعد إزالة الروبوت/الازدواج.
* **Sessionization:** بناء sessions يوميًا + تحديث تزايدي.
* **Attribution Build:** تحديث marketing\_dim (first/last touch).
* **Marts:** تجميع funnels\_mart وkpi\_daily.
* **DSAR/Erase:** Job يطبّق الحذف على clean\_\* والخرائط الهووية.
* **Dead-Letter Queue:** تخزين أي حدث فاشل المعالجة مع سبب، ومتابعة آلية.

## **11) واجهات/تكاملات (اختصار)**

* **Server Events API:** نهاية واحدة موحّدة لاستقبال أحداث الخادم (idempotent + توقيع).
* **Exports:** لوحات/تقارير (Metabase/Looker) تعتمد جداول marts\_\*.
* **Row-Level Security:** الوصول للمجاميع فقط لغير المصرّح لهم بالبيانات الخام.

## **12) اختبارات (على Staging)**

1. إطلاق سلسلة قمع كاملة (view→add\_to\_cart→checkout→purchase) ثم التحقق من:
   * وجود الأحداث في clean\_\* خلال ≤ 15 دقيقة.
   * بناء جلسة صحيحة وUTM منسّق.
   * purchase يرتبط بـ order\_fact.
2. حساب **CVR/AOV** في marts\_\* يساوي النتائج اليدوية لنفس الفترة.
3. تشغيل DSAR لعميل اختباري: اختفاء الهوية من clean\_\* وبقاء kpi\_daily كما هو.
4. Revenue delta ≤ 3%.

**DoD:** كل الاختبارات Pass، وتنبيهات المراقبة خضراء، وتقارير الـKPIs تعرض أرقامًا متّسقة مع الـDB.

## **13) القالب المعياري للأحداث**

* **الاسم:** event\_name
* **الهدف:** …
* **الحقول الإلزامية:** …
* **المصدر:** app/web/server
* **الهوية:** يتطلب user\_id؟ (Y/N) — يحفظ user\_pseudo\_id دائمًا
* **الملاحظات:** حظر PII، وحدود الطول، القيم المسموحة
* **مالِك الحدث (DRI):** فريق ××
* **اختبار Day-0:** …

## **14) الملكية والحَوْكمة**

* **DRI:** فريق البيانات/المنصّة.
* **مراجعة ربع سنوية:** تحديث القاموس، ضبط نوافذ الـAttribution/Retention، مراجعة التكاليف.
* **CI Gates:** منع نشر أحداث غير مُعرفة في القاموس، وفحوص schema قبل الإنتاج.

### **ملاحظات**

* عيّن user\_id في SDK فور تسجيل الدخول (مع الاحتفاظ بـ user\_pseudo\_id).
* أرسل purchase **من الخادم** كمرجعية، ويمكن تكراره من العميل مع سمة source.
* دوّن جميع التعاريف/الصيغ (CVR, AOV, ROAS…) في docs/analytics/metrics.md لضمان اتساق التقارير.