بناءً على:

- **Business Rules V1** 📘
- **State Machine V1** 🧠
- **README Flow – Availability Derived** 📅

هعمل لك الآن 📊 **Database Flow Section – Official V1**
بشكل هندسي مباشر قابل للتحويل إلى:

- DB Constraints
- Triggers
- RLS
- Edge Functions
- Audit Rules

📊 (Database Flow Section – V1 (Official

🎯 Core Database Principle

1️⃣ Source of Truth

Property Availability = Derived from Bookings

❌ لا يتم تخزين availability داخل properties
✅ يتم حسابها ديناميكياً من:

- bookings (confirmed, active)
- property_unavailability

كما هو مثبت رسمياً

🗄️ 1️⃣ Tables Responsibility Map

Table	مسؤول عن ماذا
profiles	المستخدمين وأدوارهم
properties	بيانات العقار فقط (ليس availability)
bookings	دورة حياة الحجز

payments	إثبات الدفع
revenues	عمولات المنصة
property_unavailability	فترات حظر المالك
activity_logs	Audit Log إجباري

Booking Flow → Database-Level 2 🧠 Mapping

A) Create Booking

شرط DB

(CHECK (verification_fee_paid = true

لا يسمح بإنشاء requested بدون 50 جنيه
كما هو محدد في Business Rules

Trigger

BEFORE INSERT ON bookings
()validate_availability →

التحقق يشمل:

('status IN ('confirmed','active
AND daterange overlap

طبقاً لقواعد Availability

B) Approve Booking

Transition:

requested → approved

Guard:

ONLY landlord_id = auth.uid()

ويسجل في:

activity_logs

كما تنص القاعدة التشغيلية 15.4

C) Electronic Payment Flow

approved → payment_pending → payment_uploaded → confirmed

DB Logic:

- payment_uploaded لا يسمح إلا لو payment_pending
- confirmed لا يسمح إلا عبر:
  - landlord
  - admin

طبقاً لـ State Machine الرسمي

D) Cash on Delivery

approved → confirmed مباشرة

لكن:

commission NOT collected yet

(E) Check-in (Critical Event

confirmed → active

Trigger:
AFTER UPDATE status TO active
→ collect_commission()
→ insert into revenues

تنفيذاً لقاعدة:

العمولة 10% من أول شهر فقط بعد الاستلام

F) Auto Expiry

payment_pending → expired

:Cron Job

'WHERE status = 'payment_pending
'AND created_at < now() - interval '24 hours

:Effects

- property becomes available
- EGP NOT refundable 50

G) Auto Complete

active → completed

WHERE end_date < CURRENT_DATE

Availability Query – Official SQL 3 📅17

*.SELECT p
FROM properties p
'WHERE p.status = 'approved

) AND NOT EXISTS
SELECT 1 FROM bookings b
WHERE b.property_id = p.id
('AND b.status IN ('confirmed','active
('[]' ,AND daterange(b.start_date, b.end_date
('[]' ,daterange(:req_start, :req_end &&

(

) AND NOT EXISTS
SELECT 1 FROM property_unavailability u

```
WHERE u.property_id = p.id
AND daterange(u.start_date, u.end_date, '[]')
&& daterange(:req_start, :req_end, '[]')
);
```

مطابق للـ README Flow

🔒 4 RLS Flow Enforcement

Bookings

Tenant

- INSERT → tenant_id = auth.uid()
- UPDATE → cancel فقط قبل active
- status لا يغير بعد confirmed

Landlord

- UPDATE → approved / rejected / active فقط

System

- expired / completed فقط عبر service role

💰 5 Commission Database Flow

Trigger: on_checkin_collect_commission

```
IF NEW.status = 'active'
```

Calculate:

```
commission = first_month_price * 0.10
```

Insert:

```
INSERT INTO revenues (
```

```
                    booking_id,
             service_fee_amount,
               total_commission
                              (
```

⚠ لا يتم تحصيل في:

- rejected
- cancelled
- expired

كما هو منصوص عليه

🚫 6 Hard – Overlapping Prevention Constraint

Index:

```
CREATE UNIQUE INDEX no_overlap
ON bookings (property_id, start_date, end_date)
WHERE status IN ('confirmed','active');
```

تنفيذ مباشر لقاعدة منع التداخل

🛡 7 Audit Guarantee (Non-Negotiable)

كل تغيير حالة:

```
AFTER UPDATE OF status
INSERT INTO activity_logs (...)
```

تنفيذ للقاعدة التشغيلية رقم 4

8️⃣ Property Status – Administrative 🧱 Only

<div dir="rtl">

properties.status يسمح فقط بـ:

draft
approved
blocked
archived

⚠️ لا يوجد booked
⚠️ لا يوجد rented

كما هو مثبت رسمياً

</div>

9️⃣ Cancellation DB Flow 🔁

Allowed only if:

```
status IN ('requested','approved','payment_pending','confirmed')
AND checkin_date > now()
```

If landlord_no_show:

status → cancelled
refund → tenant
penalty → landlord
rating--

<div dir="rtl">

مطابق للـ Flow Diagram

</div>

🔟 Database Guarantees (Final) 📌

✔ Availability never stored
✔ No overlapping confirmed bookings
✔ Commission collected only after active
✔ 50 EGP required before requested

✔ No cancellation after active
✔ Every state change logged
✔ Property never enters booked/rented

🎯 النتيجة

أصبح عندك الآن:

- 📊 Database-Level Booking Control
- 🧠 State Machine enforced at DB
- 💰 Commission event-driven
- 🔒 RLS secured
- 📅 Derived availability only
- 🧾 Audit-compliant

لو تحب نكمل بنفس المستوى الهندسي أعمل لك الآن:

- 🛡️ **RLS Security Flow – Full SQL Ready**
- ⚙️ **Event-Driven Architecture Section**
- 🧪 **(Edge Case Scenarios Matrix (Production Ready**
- 📊 **(Decision Matrix Table (Actors vs States**

قولّي نكمل على أي مستوى 🔥