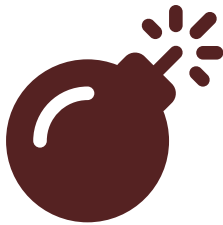


08 Data Structure

Kingdom Wife Blueprint — Data Structure

The MySQL schema · entity relationships · index strategy. Source of truth for the platform's data model.

Entity relationship diagram



Syntax error in text
mermaid version 10.9.5

Tier semantics (the central data concept)

The `kwb_users.current_tier` column drives nearly every gating decision. The values:

Value	Tier	What unlocks
0	None (signed up but not paid)	Public landing pages only · email queue starts at Tier 1 only after first purchase
1	Book Reader	Day 0/3/7/14/30 email sequence · sample chapter · intro module

Value	Tier	What unlocks
2	Course Member	Tier 1 + full Four Realms course content · journal prompts · scripture library
3	Cohort Member	Tier 2 + monthly cohort call · accountability partner · cohort-only resources

Upgrade path: 0 → 1 → 2 → 3 (always forward; downgrades expire by setting `tier_expires_at`).

Data dictionaries

`kwb_users.current_tier`

- 0 = None
- 1 = Book Reader
- 2 = Course Member
- 3 = Cohort Member

`kwb_purchases.status`

- `pending` — Stripe PaymentIntent created, not yet confirmed
- `succeeded` — payment cleared, user upgraded
- `failed` — payment failed, user not upgraded
- `refunded` — payment refunded post-fact, may trigger downgrade

`kwb_email_queue.status`

- `pending` — scheduled, not yet attempted
- `sent` — sent successfully
- `failed` — send failed (will retry up to 3x then mark cancelled)
- `cancelled` — manually cancelled by Sandi via `/admin`

`kwb_email_template` (controlled vocabulary)

- `day-0-welcome`

- `day-3-introduction-tier1` , `day-3-introduction-tier2` , `day-3-introduction-tier3`
- `day-7-reflection`
- `day-14-deeper-realm`
- `day-30-celebrate-tier1` , `day-30-upgrade-tier2` , `day-30-cohort-confirm-tier3`
- `quarterly-newsletter`
- `cohort-call-reminder`
- `cohort-call-followup`

Index strategy

Table	Index	Purpose
<code>kwb_users</code>	<code>email</code> (unique)	Login lookup
<code>kwb_users</code>	<code>current_tier</code>	Tier-segmented queries
<code>kwb_purchases</code>	<code>stripe_payment_intent</code> (unique)	Webhook idempotency
<code>kwb_purchases</code>	<code>status</code> , <code>created_at</code>	Recent purchase reports
<code>kwb_cohort_membership</code>	<code>cohort_number</code> , <code>left_at</code>	Active members per cohort
<code>kwb_journal_entries</code>	<code>user_id</code> , <code>created_at</code>	User's journal history
<code>kwb_email_queue</code>	<code>status</code> , <code>scheduled_for</code>	Cron picks pending sends

Privacy & encryption

- **Passwords:** bcrypt hashed (cost 12). Never stored plain.
- **Journal entries:** plain text in DB initially. **Plan:** encrypt at rest with per-user key once budget allows (column-level encryption via PHP openssl + key in `.env`).
- **Email + first_name:** plain text (needed for outbound email rendering).
- **Stripe data:** never stored beyond `payment_intent_id` . No card data touches the platform.
- **Cohort accountability content:** lives in `journal_entries` with `is_shared_with_partner=TRUE` . Partner can read; no other user can.

Backup strategy

What	Frequency	Retention
Full DB dump	Nightly	30 days
Full DB dump	Weekly	1 year
Code (php files)	On every git commit	indefinite via GitHub
User uploads (none currently)	n/a	n/a
Stripe data	Stripe-side (canonical)	indefinite

Recovery procedure documented in `_knowledge/BACKUP_RECOVERY_PLAN.md` .

Schema versioning

```
database/schema.sql           ← canonical schema (latest)
database/migrations/
  2026-04-29_initial.sql      ← v1.0 baseline
  2026-MM-DD_<change>.sql     ← incremental migrations as schema evolves
```

Every schema change goes through: 1. Write migration in `database/migrations/` 2. Test on local copy (or staging if available) 3. Run on production · capture timestamp + git SHA 4. Update `schema.sql` to reflect new state

Never edit production schema directly via phpMyAdmin without a migration file.

Cross-references

- `[[04_system_architecture]]` — folder structure + auth flow
- `[[05_automation_plan]]` — what queries this data
- `[[../..../sandi-system-engine/_knowledge/SECURITY_PRIVACY|SECURITY_PRIVACY]]` — what stays out of git