

USER

- user_id (PK, UUID)
- email (STRING, unique)
- phone (STRING, unique, nullable)
- display_name (STRING)
- role (ENUM: student|tutor|admin)
- is_active (BOOL, default true)
- created_at (TIMESTAMP)
- updated_at (TIMESTAMP)

STUDENT_PROFILE

- user_id (PK, FK → USER.user_id)
- grade (STRING) // e.g., O/L 10, A/L 12
- area_code (FK → AREA.area_code)
- subjects_of_interest (ARRAY<STRING>) // optional

TUTOR_PROFILE

- user_id (PK, FK → USER.user_id)
- bio (TEXT)
- qualifications (TEXT)
- subjects_taught (ARRAY<STRING>) // subject codes
- area_code (FK → AREA.area_code)
- mode (ENUM: online|physical|hybrid)
- base_price (DECIMAL(10,2))
- rating_avg (FLOAT, default null)
- rating_count (INT, default 0)
- status (ENUM: pending|approved|rejected, default pending)
- reviewed_by (FK → USER.user_id, nullable)

- reviewed_at (TIMESTAMP, nullable)

ADMIN_PROFILE

- user_id (PK, FK → USER.user_id)
- role_type (ENUM: super|academic|finance)

SUBJECT (reference)

- subject_code (PK, STRING) // e.g., OLMATH, ALPHYS
- name (STRING)
- level (ENUM: OL|AL|Other)

AREA (reference)

- area_code (PK, STRING) // e.g., CMB05
- area_name (STRING)
- lat (FLOAT), lng (FLOAT) // optional, for distance buckets

2) Classes, sessions, enrollment

VENUE (for physical classes)

- venue_id (PK, UUID)
- name (STRING)
- address (STRING)
- area_code (FK → AREA.area_code)
- capacity (INT)

CLASS

- class_id (PK, UUID)
- tutor_id (FK → USER.user_id)
- subject_code (FK → SUBJECT.subject_code)
- grade (STRING)
- mode (ENUM: online|physical|hybrid)
- area_code (FK → AREA.area_code)

- venue_id (FK → VENUE.venue_id, nullable if online)
- fee (DECIMAL(10,2))
- price_band (ENUM: low|mid|high) // derived or stored
- capacity_seats (INT)
- status (ENUM: draft|published|archived, default draft)
- created_at (TIMESTAMP)
- published_at (TIMESTAMP, nullable)

CLASS_SESSION (expanded meetings for clash checks & calendars)

- session_id (PK, UUID)
- class_id (FK → CLASS.class_id)
- session_date (DATE)
- start_time (TIME)
- end_time (TIME)
- room (STRING, nullable)
- is_cancelled (BOOL, default false)
- cancel_reason (STRING, nullable)

Add a unique or validation rule to prevent overlapping sessions for the same (tutor_id, session_date, time range, venue/room).

ENROLLMENT

- enrollment_id (PK, UUID)
- class_id (FK → CLASS.class_id)
- student_id (FK → USER.user_id)
- status (ENUM: pending|active|completed|cancelled, default pending)
- enrolled_at (TIMESTAMP)
- cancelled_at (TIMESTAMP, nullable)
- cancel_reason (STRING, nullable)

- CONSTRAINT unique_student_class UNIQUE (student_id, class_id)

3) Payments (proof + verification + refunds)

INVOICE

- invoice_id (PK, UUID)
- enrollment_id (FK → ENROLLMENT.enrollment_id)
- amount_due (DECIMAL(10,2))
- due_date (DATE)
- status (ENUM: awaiting_proof|under_review|verified|rejected|refunded, default awaiting_proof)
- created_at (TIMESTAMP)

PAYMENT

- payment_id (PK, UUID)
- invoice_id (FK → INVOICE.invoice_id)
- paid_amount (DECIMAL(10,2))
- paid_at (TIMESTAMP)
- method (ENUM: bank_transfer|cash|other)
- proof_url (STRING)
- verify_status (ENUM: verified|rejected|pending, default pending)
- verified_by (FK → USER.user_id, nullable)
- verified_at (TIMESTAMP, nullable)
- verify_note (STRING, nullable)

REFUND (optional, when needed)

- refund_id (PK, UUID)
- payment_id (FK → PAYMENT.payment_id)
- refund_amount (DECIMAL(10,2))
- refunded_at (TIMESTAMP)

- reason (STRING)
- processed_by (FK → USER.user_id)

4) Content, announcements, messaging, notifications

MATERIAL

- material_id (PK, UUID)
- class_id (FK → CLASS.class_id)
- title (STRING)
- file_url (STRING) // or link_url
- allow_download (BOOL, default true)
- uploaded_by (FK → USER.user_id)
- uploaded_at (TIMESTAMP)

ANNOUNCEMENT

- announcement_id (PK, UUID)
- scope (ENUM: class|grade|area|all)
- class_id (FK → CLASS.class_id, nullable)
- grade (STRING, nullable)
- area_code (FK → AREA.area_code, nullable)
- title (STRING)
- body (TEXT)
- created_by (FK → USER.user_id) // tutor/admin
- created_at (TIMESTAMP)

MESSAGE (1:1 or class thread)

- message_id (PK, UUID)
- sender_id (FK → USER.user_id)
- recipient_id (FK → USER.user_id) // for 1:1
- class_id (FK → CLASS.class_id, nullable) // for class thread

- text (TEXT)
- sent_at (TIMESTAMP)
- is_deleted (BOOL, default false)

NOTIFICATION (inbox + push)

- notification_id (PK, UUID)
- recipient_id (FK → USER.user_id)
- type (ENUM:
enrollment_status|payment_status|schedule_change|announcement|system)
- title (STRING)
- body (TEXT)
- is_read (BOOL, default false)
- created_at (TIMESTAMP)

RATING (optional but useful for AI + UX)

- rating_id (PK, UUID)
- student_id (FK → USER.user_id)
- tutor_id (FK → USER.user_id)
- class_id (FK → CLASS.class_id)
- stars (INT, 1..5)
- comment (TEXT, nullable)
- created_at (TIMESTAMP)

5) AI & analytics (events + rollups + views)

EVENT_INTERACTION (critical for recommendations & insights)

- event_id (PK, UUID)
- student_id (FK → USER.user_id, nullable for anonymous)
- tutor_id (FK → USER.user_id, nullable)
- class_id (FK → CLASS.class_id, nullable)

- event_type (ENUM: search|impression|view_tutor|view_class|click|bookmark|enrol)
- query_text (STRING, nullable) // for search
- ts (TIMESTAMP)

Index: (student_id, ts), (class_id, ts), (tutor_id, ts), (event_type, ts)

WEEKLY_DEMAND (materialized view or table for forecasting)

- week_start (DATE) // Monday of week
- subject_code (FK → SUBJECT.subject_code)
- area_code (FK → AREA.area_code)
- views (INT)
- clicks (INT)
- enrolls (INT)
- PRIMARY KEY (week_start, subject_code, area_code)

(Recommended) Feature views/tables for ML (derived)

- RECS_TRAINING_PAIRS (one row per (student_id, class_id))
 - student_id, class_id, label_clicked_or_enrolled (BOOL)
 - features: subject_match (BOOL), grade_match (BOOL), time_overlap (BOOL), distance_bucket (ENUM), price_band_fit (ENUM), tutor_popularity (FLOAT), past_clicks_30d (INT), past_enrolls_90d (INT), etc.
- SEARCH_TERMS (parsed from events)
 - student_id, ts, subject_code, day_of_week, area_code, fee_band, results_count

6) Governance & auditing (optional but handy)

AUDIT_LOG

- audit_id (PK, UUID)
- actor_type (ENUM: student|tutor|admin)
- actor_id (FK → USER.user_id)
- action (STRING) // approve_tutor, publish_class, verify_payment, issue_refund, etc.

- entity (STRING) // TUTOR_PROFILE, CLASS, PAYMENT, etc.
- entity_id (STRING) // the target PK
- ts (TIMESTAMP)
- note (STRING, nullable)

7) Indexes & constraints (quick list)

- USER: unique(email), unique(phone)
- ENROLLMENT: UNIQUE(student_id, class_id)
- CLASS: index(subject_code, grade, mode, fee), index(area_code), index(status)
- CLASS_SESSION: index(class_id, session_date, start_time); add clash-preventing validation in app/CF
- PAYMENT: index(invoice_id), index(verify_status), index(verified_by)
- MATERIAL: index(class_id, uploaded_at)
- MESSAGE: index(class_id, sent_at), index(recipient_id, sent_at)
- NOTIFICATION: index(recipient_id, is_read, created_at)
- EVENT_INTERACTION: indexes noted above
- WEEKLY_DEMAND: PK(week_start, subject_code, area_code)

8) Data type notes (portable choices)

- IDs: UUID (STRING) or Firestore doc IDs
- Money: DECIMAL(10,2)
- Times: TIMESTAMP (UTC), DATE, TIME
- Enums: store as STRING values; validate in app/CF
- Arrays: if using SQL, use join tables instead of arrays; in Firestore, arrays are fine

9) MVP vs Later (suggested)

- **MVP:** USER, STUDENT_PROFILE, TUTOR_PROFILE, SUBJECT, AREA, CLASS, CLASS_SESSION, ENROLLMENT, INVOICE, PAYMENT, MATERIAL, ANNOUNCEMENT, MESSAGE, NOTIFICATION, EVENT_INTERACTION, WEEKLY_DEMAND

- **Later:** VENUE (if physical locations matter), REFUND, RATING, AUDIT_LOG