Schema Documentation – Skill Exchange & Learning Platform

Data Dictionary

PK = Primary Key | FK = Foreign Key | NN = NOT NULL | U = UNIQUE | ENUM = fixed value set

1. University

Attribute	Data Type	Constraints	Description
UniversityID	INT	PK, NN, AUTO_INCREMENT	Primary key
UniversityName	VARCHAR(100)	NN	Full official name of the university
Address	VARCHAR(255)	NN	Physical address
ContactEmail	VARCHAR(100)	NN	Official contact email
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

2. User

Attribute	Data Type	Constraints	Description
UserID	INT	PK, NN, AUTO_INCREMENT	Primary key
Email	VARCHAR(100)	NN, UNIQUE	Login email (must be unique platform-wide)
PasswordHash	VARCHAR(255)	NN	Hashed password
FullName	VARCHAR(100)	NN	User name
PhoneNumber	VARCHAR(20)	NULL	Optional contact number
CreatedAt	DATETIME	NN, DEFAULT NOW()	Account creation timestamp
LastLogin	DATETIME	NULL	Last successful login

3. Student

Attribute	Data Type	Constraints	Description
StudentID	INT	PK, FK → User.UserID	Same value as UserID because of ISA inheritance
UniversityID	INT	FK → University.UniversityID, NN	Student's enrolled university
ReputationPoints	INT	NN, DEFAULT 0	Accumulated from positive reviews/exchanges
Bio	TEXT	NULL	Optional self description

IsAdminVerified	BOOLEAN	NN, DEFAULT FALSE	Becomes true after admin manually approves account
CreatedAt	DATETIME	NN, DEFAULT NOW()	Profile creation timestamp

4. Admin

Attribute	Data Type	Constraints	Description
AdminID	INT	PK, FK → User.UserID	Same value as UserID because of ISA inheritance
UniversityID	INT	FK → University.UniversityID, NULL	University this admin manages, NULL for Super Admin
AdminLevel	VARCHAR(20)	NN	'University Admin' or 'Super Admin'
CreatedAt	DATETIME	NN, DEFAULT NOW()	Admin account creation timestamp

5. SkillCategory

Attribute	Data Type	Constraints	Description
CategoryID	INT	PK, NN, AUTO_INCREMENT	Primary key
CategoryName	ENUM	NN, UNIQUE	One of these categories: Programming, Languages, Design, Music, Business, Academics, Creative, Life Skills
Description	TEXT	NULL	Description of the category
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

6. Skill

Attribute	Data Type	Constraints	Description
SkillID	INT	PK, NN, AUTO_INCREMENT	Primary key
CategoryID	INT	FK → SkillCategory.CategoryID, NN	Owning category

SkillName	ENUM	NN	Programming: Python, JavaScript, Java, C++, Web Development | Languages: English, Spanish, French, Arabic, Urdu | Design: Graphic Design, UI/UX, Figma, Adobe Photoshop | Music: Guitar, Piano, Music Production, Singing | Business: Digital Marketing, Excel, Presentation Skills | Academics: Mathematics, Calculus, Physics, Essay Writing | Creative: Creative Writing, Photography, Video Editing | Life Skills: Cooking, Public Speaking, Time Management
Description	TEXT	NULL	What the skill entails
DifficultyLevel	ENUM	NULL	Beginner / Intermediate / Advanced
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

7. OfferedSkill

Attribute	Data Type	Constraints	Description
OfferID	INT	PK, NN, AUTO_INCREMENT	Primary key
StudentID	INT	FK → Student.StudentID, NN	Offering student
SkillID	INT	FK → Skill.SkillID, NN	Skill being offered
IsPaid	BOOLEAN	NN, DEFAULT FALSE	Whether the offer charges a fee
PricePerHour	DECIMAL(10,2)	NULL, CHECK >= 0	Rate per hour, NULL if free
Description	TEXT	NULL	Details of what the student offers
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

8. RequestedSkill

Attribute	Data Type	Constraints	Description
RequestID	INT	PK, NN, AUTO_INCREMENT	Primary key
StudentID	INT	FK → Student.StudentID, NN	Requesting student
SkillID	INT	FK → Skill.SkillID, NN	Skill being requested
Description	TEXT	NULL	What the student wants to learn
PreferredTime	DATETIME	NULL	Preferred session datetime
PreferredMode	VARCHAR(20)	NULL	Exchange / Paid
Status	VARCHAR(20)	NN, DEFAULT 'Pending'	Pending / Matched / Completed

CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

9. SkillQuestion

Attribute	Data Type	Constraints	Description
QuestionID	INT	PK, NN, AUTO_INCREMENT	Primary key
SkillID	INT	FK → Skill.SkillID, NN	Skill this question tests
QuestionText	TEXT	NN	The question prompt
CorrectAnswer	VARCHAR(1)	NN	Option key of correct answer (e.g 'A')
Points	INT	NN, DEFAULT 1	Marks given for correct answer
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

10. QuestionOption [Weak Entity]

Attribute	Data Type	Constraints	Description
QuestionID	INT	PK, FK → SkillQuestion.QuestionID	Composite PK, identifies parent question
OptionKey	VARCHAR(1)	PK	Composite PK, e.g. 'A', 'B', 'C', 'D'
OptionText	VARCHAR(255)	NN	Text of this option
IsCorrect	BOOLEAN	NN, DEFAULT FALSE	True for the correct option

11. SkillEvaluation

Attribute	Data Type	Constraints	Description
EvaluationID	INT	PK, NN, AUTO_INCREMENT	Primary key
StudentID	INT	FK → Student.StudentID, NN	Student being evaluated
SkillID	INT	FK → Skill.SkillID, NN	Skill under evaluation
AdminID	INT	FK → Admin.AdminID, NULL	Admin who assigned/reviewed
StartedAt	DATETIME	NN	When the student began the evaluation
SubmittedAt	DATETIME	NULL	Submission time, NULL if still in progress
Score	DECIMAL(5,2)	NULL, CHECK 0–TotalPossible	Marks obtained, NULL until submitted
TotalPossible	INT	NN	Maximum marks achievable
Status	VARCHAR(20)	NN, DEFAULT 'InProgress'	InProgress / Submitted / Passed / Failed
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

12. EvaluationAnswer [Weak Entity]

Attribute	Data Type	Constraints	Description
EvaluationID	INT	PK, FK → SkillEvaluation.EvaluationID	Composite PK , links to evaluation
QuestionID	INT	PK, FK → SkillQuestion.QuestionID	Composite PK , links to question
StudentAnswer	VARCHAR(255)	NN	The option key the student selected

13. Exchange

Attribute	Data Type	Constraints	Description
ExchangeID	INT	PK, NN, AUTO_INCREMENT	Primary key
OfferID	INT	FK → OfferedSkill.OfferID, NN	The skill offer matched
RequestID	INT	FK → RequestedSkill.RequestID, NN	The skill request matched
ExchangeType	VARCHAR(10)	NN, CHECK IN ('Free','Paid')	Free / Paid
ConversationID	INT	FK → Conversation.ConversationID, NULL	Linked chat thread
Status	VARCHAR(20)	NN, DEFAULT 'Pending'	Pending / Active / Completed / Cancelled
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

14. PaidExchange

Attribute	Data Type	Constraints	Description
ExchangeID	INT	PK, FK → Exchange.ExchangeID, UNIQUE	1:1 with Exchange, also the FK
Price	DECIMAL(10,2)	NN	Agreed price for this exchange
Currency	VARCHAR(3)	NN, DEFAULT 'PKR'	Currency used for payment

15. Session

Attribute	Data Type	Constraints	Description
SessionID	INT	PK, NN, AUTO_INCREMENT	Primary key

ExchangeID	INT	FK → Exchange.ExchangeID, NN	Owning exchange
ScheduledStartTime	DATETIME	NN	Planned start
ScheduledEndTime	DATETIME	NN, CHECK > StartTime	Planned end
ActualStartTime	DATETIME	NULL	Actual start once session begins
ActualEndTime	DATETIME	NULL	Actual end once session closes
Venue	VARCHAR(255)	NULL	'Online' or physical location description
Status	VARCHAR(20)	NN, DEFAULT 'Scheduled'	Scheduled / InProgress / Completed / Cancelled
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

16. VideoSession

Attribute	Data Type	Constraints	Description
VideoSessionID	INT	PK, NN, AUTO_INCREMENT	Primary key
SessionID	INT	FK → Session.SessionID, NN, UNIQUE	1:1 with Session
MeetingLink	VARCHAR(255)	NN	URL for the video call
MeetingPassword	VARCHAR(50)	NULL	Password if required by platform
Platform	VARCHAR(50)	NN	Zoom / Google Meet / Other
RecordingURL	VARCHAR(255)	NULL	Link to session recording (if saved)
Status	VARCHAR(20)	NN, DEFAULT 'Scheduled'	Scheduled / Started / Ended / Cancelled
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

17. Payment

Attribute	Data Type	Constraints	Description
PaymentID	INT	PK, NN, AUTO_INCREMENT	Primary key
ExchangeID	INT	FK → Exchange.ExchangeID, NN, UNIQUE	1:1 with Exchange for paid type
Attribute	**Data Type**	**Constraints**	**Description**
Amount	DECIMAL(10,2)	NN	Total amount paid by learner
PaymentStatus	VARCHAR(20)	NN, DEFAULT 'Pending'	Pending / Completed / Failed / Refunded
PaymentMethod	VARCHAR(50)	NN	Credit Card / Wallet / Bank Transfer etc.

PaidAt	DATETIME	NULL	Timestamp of successful payment
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

18. Conversation

Attribute	Data Type	Constraints	Description
ConversationID	INT	PK, NN, AUTO_INCREMENT	Primary key
Student1ID	INT	FK → Student.StudentID, NN	First participant
Student2ID	INT	FK → Student.StudentID, NN, CHECK ≠ Student1ID	Second participant (must differ from Student1ID)
CreatedAt	DATETIME	NN, DEFAULT NOW()	Thread creation timestamp

19. Message

Attribute	Data Type	Constraints	Description
MessageID	INT	PK, NN, AUTO_INCREMENT	Primary key
ConversationID	INT	FK → Conversation.ConversationID, NN	Owning conversation
SenderID	INT	FK → Student.StudentID, NN	Student who sent the message
Content	TEXT	NN	Message body
IsRead	BOOLEAN	NN, DEFAULT FALSE	True once the recipient has seen it
CreatedAt	DATETIME	NN, DEFAULT NOW()	Send timestamp

20. Review

Attribute	Data Type	Constraints	Description
ReviewID	INT	PK, NN, AUTO_INCREMENT	Primary key
ExchangeID	INT	FK → Exchange.ExchangeID, NULL	Set when review is for a whole exchange
SessionID	INT	FK → Session.SessionID, NULL	Set when review is for a specific session
ReviewerID	INT	FK → Student.StudentID, NN	Student writing the review
RevieweeID	INT	FK → Student.StudentID, NN	Student being reviewed
Rating	DECIMAL(2,1)	NN, CHECK 0.0–5.0	Star rating to one decimal place
Feedback	TEXT	NULL	Optional written text

Attribute	Data Type	Constraints	Description
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

21. Portfolio

Attribute	Data Type	Constraints	Description
PortfolioID	INT	PK, NN, AUTO_INCREMENT	Primary key
StudentID	INT	FK → Student.StudentID, NN	Owning student
SkillID	INT	FK → Skill.SkillID, NN	Skill being showcased
Description	TEXT	NULL	What this portfolio item demonstrates
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp
UpdatedAt	DATETIME	NN, DEFAULT NOW()	Last update timestamp

Attribute	Data Type	Constraints	Description
OfferID	INT	PK, FK → OfferedSkill.OfferID	Composite PK , parent offer
TimeSlotID	INT	PK	Composite PK , slot number within the offer
SlotStart	DATETIME	NN	Availability window start
SlotEnd	DATETIME	NN	Availability window end
Status	VARCHAR(20)	NN, DEFAULT 'Available'	Available / Booked / Cancelled
CreatedAt	DATETIME	NN, DEFAULT NOW()	Record creation timestamp

22. OfferTimeSlot [Weak Entity]

There are 22 entities in total, 3 of them are weak entities and 3 are strong and independent meaning they have no FKs, rest are all strong and dependent. We had 20 entities pre normalization but afterwards 2 more entities: PaidExchange and QuestionOption were added. This is the whole normalization summary:

1) SkillQuestion → QuestionOption

We had OptionA, OptionB, OptionC, OptionD as separate columns, which is basically repeating the same attribute multiple times and if we wanted to change the number of options we cant do that here either because of the fixed number of attributes of an entity. That breaks 1NF, so we moved options into a new table (QuestionOption) so each option becomes a row and we can support any number of options.

2) Payment – Removed InstructorID and LearnerID

InstructorID and LearnerID were already indirectly determined by ExchangeID, so storing them again was redundant. This breaks 3NF (transitive dependency), so we removed them and now we just get them using JOINs when needed during runtime.

3) SkillEvaluation – Removed Passed

The Passed column was completely calculatable from Score and TotalPossible. Since it depends on non key attributes, it violates 3NF, so we removed it and compute it when required.

4) EvaluationAnswer – Removed IsCorrect

IsCorrect can always be figured out by comparing StudentAnswer with the correct answer in SkillQuestion. Because it's derivable from another relation, it violates 3NF as a non key attribute depends on another non key attribute, so we now calculate it during queries instead of storing it.

5) Exchange – Removed Price → Added PaidExchange

Price only made sense when the exchange type was "Paid", otherwise it was null. That violates 3NF, so we removed Price from Exchange and created a separate PaidExchange table only for paid cases.

Index Specification with Rationale:

Index Name	Table	Column(s)	Type	Rationale
idx_user_created	User	CreatedAt	B-Tree	Filter users by registration date in admin dashboard
idx_student_university	Student	UniversityID	B-Tree	Filter students by university (admin views, matching)
idx_student_reputation	Student	ReputationPoints	B-Tree	Leaderboard ORDER BY reputation descending
idx_offer_student_skill	OfferedSkill	StudentID, SkillID	B-Tree	Check if student already offers a skill (composite)
idx_offer_skill	OfferedSkill	SkillID	B-Tree	Find all offers for a given skill (matching engine)
idx_request_student	RequestedSkill	StudentID	B-Tree	Find all requests by a student
idx_request_skill_status	RequestedSkill	SkillID, Status	B-Tree	Filter open requests for a skill (matching engine)
idx_slot_offer_status	OfferTimeSlot	OfferID, Status	B-Tree	Filter available slots for booking
idx_exch_offer	Exchange	OfferID	B-Tree	Join Exchange → OfferedSkill (derive instructor)
idx_exch_request	Exchange	RequestID	B-Tree	Join Exchange → RequestedSkill (derive learner)
idx_exch_status_created	Exchange	Status, CreatedAt	B-Tree	Dashboard queries filter by Active/Pending and date
idx_session_exch	Session	ExchangeID	B-Tree	List all sessions for an exchange
idx_session_time	Session	ScheduledStartTime	B-Tree	Calendar/timeline queries sorted by date
idx_payment_exch_status	Payment	ExchangeID, PaymentStatus	B-Tree	Payment status check inside ACID transaction
idx_review_reviewee	Review	RevieweeID	B-Tree	Load profile ratings — very common lookup
idx_review_reviewer	Review	ReviewerID	B-Tree	Reviews written by a specific student

idx_msg_conv_time	Message	ConversationID, CreatedAt	B-Tree	Load all messages in a conversation chronologically
idx_eval_student_skill	SkillEvaluation	StudentID, SkillID	B-Tree	Student transcript — check duplicate evaluations
idx_eval_status	SkillEvaluation	Status	B-Tree	Filter in-progress vs submitted evaluations
idx_portfolio_student	Portfolio	StudentID	B-Tree	Load a student's portfolio items
idx_portfolio_skill	Portfolio	SkillID	B-Tree	Find portfolios for a skill

User Roles:

1) Student

This is our normal university user. A student can offer skills, request skills, join exchanges, send messages, take evaluations, and build their portfolio.

What Students CAN Do:

- Create, update, and delete their own OfferedSkill and RequestedSkill
- Read Skills, SkillCategories, and other students' OfferedSkills
- Create and read Exchanges they are part of (both as instructor or learner)
- Create and read Sessions linked to their Exchanges
- Send and read Messages in their own Conversations
- Create Reviews for Exchanges/Sessions they participated in
- Manage their own Portfolio
- Take SkillEvaluations and view their own results

What Students CANNOT Do:

- Access other students' private messages or payment details
- Approve or reject student accounts
- Create or modify Skill or SkillCategory
- Manage SkillQuestion or QuestionOption
- Access Payments of other exchanges

2) Admin

Admin is one single entity in the database, but it has an attribute called AdminLevel.

- If AdminLevel = UniversityAdmin then they manage only their assigned university.
- If AdminLevel = SuperAdmin then they are the only top level admin with system wide access.

What University Admin CAN Do:

- Read all student profiles in their university

- Approve/reject student accounts
- Create/update/delete SkillQuestion and QuestionOption
- Assign themselves to moderate SkillEvaluations
- Update Exchange and SkillEvaluation status
- Read all Reviews in their university

What University Admin CANNOT Do:

- Access other universities' data
- Read private student messages
- Directly process payments
- Delete user accounts (platform-level action)
- Add/update/delete Skill or SkillCategory

3) Platform Admin

This is the SuperAdmin which is an admin on platform level. There is only one, and they have almost full system control across all universities.

What SuperAdmin CAN Do:

- Full CRUD on University
- Create Admin accounts and assign UniversityID
- Read all data across all universities
- Access full Payment records (including TransactionID & PaymentMethod)
- Generate financial reports
- Delete user accounts (after due process)
- Manage Skill and SkillCategory structure
- View all SkillEvaluation results
- Override Exchange status in disputes

What SuperAdmin CANNOT Do:

- Directly impersonate student accounts
- Modify PasswordHash manually