**Event Ticket Reservation – Database Design**

ERD Overview

Main Entities:

* users
* events
* ticket\_rules
* ticket\_batches
* bookings

Entity Relationships

* One event has one ticket\_rule
* One event has many ticket\_batches
* One user can make many bookings as long as quota max per user limit buyout quota
* One booking is tied to one event and optionally to one ticket\_batch

**Table Structures & Descriptions**

users

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Unique identifier for each user |
| username | VARCHAR(100) | Unique username |
| email | VARCHAR(100) | Unique email address |
| created\_at | timestamp | Created date time |

Users table used for store user information to help track booking per user

event

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Unique event ID |
| name | VARCHAR(255) | Name of the event |
| description | TEXT | Full description of the event |
| location | VARCHAR(255) | Where the event will be held |
| event\_datetime | TIMESTAMP | When the event happens |
| total\_tickets | INT | Total tickets available for all batches |
| created\_at | TIMESTAMP | Auto-generated creation time |

Event table used for stored event information

ticket\_rules

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Rule id |
| event\_id | BIGINT (FK) | Linked to event table PK |
| is\_batched | BOOLEAN | Whether the event uses batches or not |
| is\_time\_limited | BOOLEAN | Whether the event uses time limited buy event or not |
| max\_tickets\_per\_user | INT | Maximum tickets one user can book |

Ticket\_rules used for define each ticket even behave. So Bussiness flow like time-limited buy event, quota for each user, batching or not (regular sell or there is like event buy), etc..

**ticket\_batches**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Ticket batch id |
| event\_id | BIGINT (FK) | Linked to event table PK |
| batch\_name | VARCHAR(100) | Name for batches like "Early Bird", "Presale", “Member Sale” etc. |
| reserved\_quota | INT | Reserved number of tickets in this batch |
| sold\_count | INT | Auto-updated count of sold tickets |
| price | DECIMAL(10,2) | Price per ticket |
| start\_time | TIMESTAMP | When this batch is open for booking |
| end\_time | TIMESTAMP | When this batch closes |
| batch\_type | ENUM | LIMITED or REGULAR  LIMITED is for event, REGULAR is for normal sale mode |

ticket\_batches used for supporting ticket\_rules and make configuration for sale for each event

bookings

|  |  |  |
| --- | --- | --- |
| Colum | Type | Description |
| id | BIGINT (PK) | Booking id |
| user\_id | BIGINT (FK) | Linked to user table PK |
| event\_id | BIGINT (FK) | Linked to event table PK |
| ticket\_batch\_id | BIGINT (FK) | Linked to ticket\_batches table PK |
| quantity | INT | Number of ticket booked |
| status | ENUM | Status for booking, it supposed to be like invoice that already paid (if integrate with payment) can be PENDING, CONFIRMED, or EXPIRED |
| reserved\_at | TIMESTAMP | Date when user doing book ticket event |

Bookings used for track user booked ticket event, reservation, and payment status (if already integrated with payment)

**Relationships Summary**

* events ↔ ticket\_rules: 1-to-1
* events ↔ ticket\_batches: 1-to-many
* events ↔ bookings: 1-to-many
* users ↔ bookings: 1-to-many
* ticket\_batches ↔ bookings: many-to-1 (optional)