

UNIVERSITY OF COMPUTER STUDIES, YANGON

Tour Booking System

Covered By

**DATABASE TRANSACTION MANAGEMENT**

Faculty of Information Science

University of Computer Studies, Yangon

Submitted By Semester – VI

Group No. –

Date

Group –

|  |  |  |
| --- | --- | --- |
| Roll No. | Name | Role |
|  | Htet Nadi | Leader |
|  | Ye Wont Aung | Co-Leader |
|  | Than Thar Linn Latt | Member |
|  | Phu Phu May Maung | Member |
|  | Shwe Sin Phu Lwin | Member |
|  | Pan Kay Thwel Oo | Member |
|  | May Thin Nwe | Member |

**Chapter 1**

**Introduction**

* 1. **Project Description**

The Tour Booking System is a comprehensive database-driven system designed to manage tour packages, customer bookings, and payment processing for a travel agency. The system provides an organized and scalable way to handle administrative tasks, customer interactions, and financial transactions while ensuring data integrity, efficiency, and security.

This database system enables administrators to manage tour packages, monitor bookings, and track payments, while customers can view available tours, make reservations, and check booking details. The system enforces role-based access control, ensuring that only authorized administrators can modify packages or confirm payments, while customers have restricted access for making bookings and viewing their own records.

* 1. **Objectives**

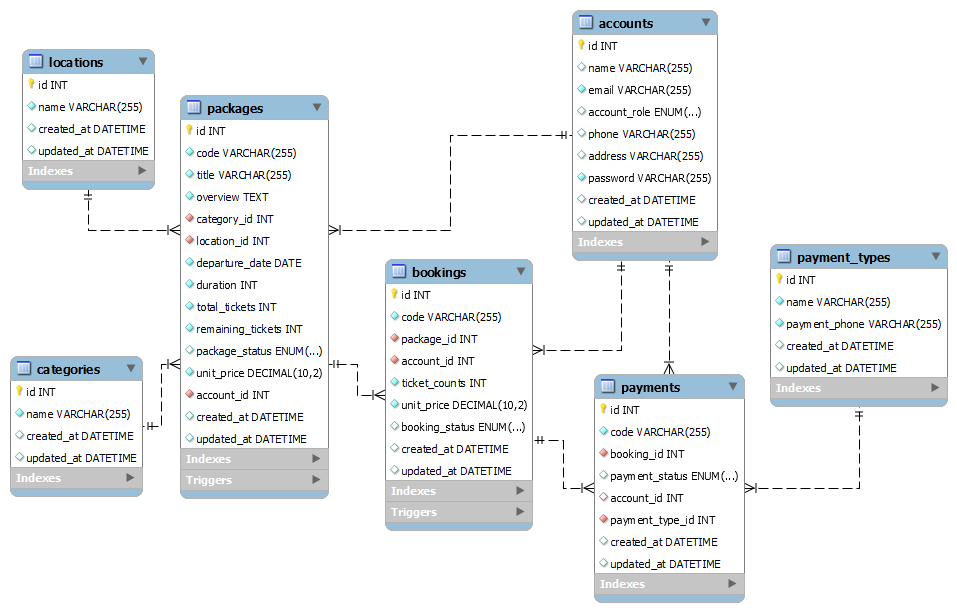
The objectives of the Tour Booking System are:

1. Efficient Tour Package Management
2. Streamlined Booking Process
3. Secure Payment Management
4. Role-Based Access Control
5. Data Integrity and Validation

**Chapter – 2**

**Database Management System**

**2.1 Entity Relationship (ER) Diagram**



**Figure 2.1 ER Diagram**

**2.2 Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tour Booking System | | | | | | |
| Table | **Field** | **Type** | **Null** | **Default** | **Key** | **Foreign Key Table** |
| accounts | id | int | No |  | PK |  |
| name | varchar(255) | Yes |  |  |  |
| email | varchar(255) | No |  | UNI |  |
| account\_role | enum | Yes | ‘CUSTOMER’ |  |  |
| phone | varchar(255) | Yes |  |  |  |
| address | varchar(255) | Yes |  |  |  |
| password | varchar(255) | No |  |  |  |
| created\_at | datetime | Yes | current\_timestamp |  |  |
| updated\_at | datetime | Yes | current\_timestamp |  |  |
|  |  |  |  |  |  |  |
| categories | id | int | No |  | PK |  |
| name | varchar(255) | No |  | UNI |  |
| created\_at | datetime | Yes | current\_timestamp |  |  |
| updated\_at | datetime | Yes | current\_timestamp |  |  |
|  |  |  |  |  |  |  |
| locations | id | int | No |  | PK |  |
| name | varchar(255) | No |  | UNI |  |
| created\_at | datetime | Yes | current\_timestamp |  |  |
| updated\_at | datetime | Yes | current\_timestamp |  |  |
|  |  |  |  |  |  |  |
| packages | id | int | No |  | PK |  |
| code | varchar(255) | No |  | UNI |  |
| title | varchar(255) | No |  |  |  |
| overview | text | Yes |  |  |  |
| category\_id | int | No |  | FK | categories |
| location\_id | int | No |  | FK | locations |
| departure\_date | date | No |  |  |  |
| duration | int | No |  |  |  |
| total\_tickets | int | No |  |  |  |
| remaining\_tickets | int | No |  |  |  |
| package\_status | enum | No | ‘AVAILABLE' |  |  |
| unit\_price | decimal(10.2) | No |  |  |  |
| account\_id | int | No |  | FK | accounts |
| created\_at | datetime | Yes | current\_timestamp |  |  |
| updated\_at | datetime | Yes | current\_timestamp |  |  |
|  |  |  |  |  |  |  |
| bookings | id | int | No |  | PK |  |
| code | varchar(255) | No |  | UNI |  |
| package\_id | varchar(255) | No |  | FK | packages |
| account\_id | int | No |  | FK | accounts |
| ticket\_counts | int | No |  |  |  |
| unit\_price | decimal(10.2) | No |  |  |  |
| booking\_status | enum | Yes | ‘PENDING' |  |  |
| created\_at | datetime | Yes | current\_timestamp |  |  |
| updated\_at | datetime | Yes | current\_timestamp |  |  |
|  |  |  |  |  |  |  |
| payment\_types | id | int | No |  | PK |  |
| name | varchar(255) | No |  | UNI |  |
| payment\_phone | varchar(255) | No |  |  |  |
| created\_at | datetime | Yes | current\_timestamp |  |  |
| updated\_at | datetime | Yes | current\_timestamp |  |  |
|  |  |  |  |  |  |  |
| payments | id | int | No |  | PK |  |
| code | varchar(255) | No |  | UNI |  |
| booking\_id | int | No |  | FK | bookings |
| payment\_status | enum | Yes | ‘PENDING' |  |  |
| account\_id | Int | Yes | NULL | FK | accounts |
| payment\_type\_id | int | No |  | FK | payment\_types |
| created\_at | datetime | Yes | current\_timestamp |  |  |
| updated\_at | datetime | Yes | current\_timestamp |  |  |

**2.3 Data Insertion in Relational Tables**

The following figures are data inserted into relational tables.

**2.3.1 Table Creation**

create table accounts (

id int auto\_increment primary key,

name varchar(255),

email varchar(255) not null unique,

account\_role enum('ADMIN', 'CUSTOMER') default 'CUSTOMER',

phone varchar(255) null,

address varchar(255) null,

password varchar(255) not null,

created\_at datetime default CURRENT\_TIMESTAMP,

updated\_at datetime default CURRENT\_TIMESTAMP on update CURRENT\_TIMESTAMP

);

create table categories (

id int auto\_increment primary key,

name varchar(255) not null unique,

created\_at datetime default CURRENT\_TIMESTAMP,

updated\_at datetime default CURRENT\_TIMESTAMP on update CURRENT\_TIMESTAMP

);

create table locations (

id int auto\_increment primary key,

name varchar(255) not null unique,

created\_at datetime default CURRENT\_TIMESTAMP,

updated\_at datetime default CURRENT\_TIMESTAMP on update CURRENT\_TIMESTAMP

);

create table packages (

id int auto\_increment primary key,

code varchar(255) not null unique,

title varchar(255) not null,

overview text not null,

category\_id int not null,

location\_id int not null,

departure\_date date not null,

duration int not null,

total\_tickets int not null,

remaining\_tickets int not null,

package\_status enum('AVAILABLE', 'UNAVAILABLE', 'FINISHED') default 'AVAILABLE',

unit\_price decimal(10, 2) not null,

account\_id int not null,

created\_at datetime default CURRENT\_TIMESTAMP,

updated\_at datetime default CURRENT\_TIMESTAMP on update CURRENT\_TIMESTAMP,

foreign key (category\_id) references categories (id),

foreign key (location\_id) references locations (id),

foreign key (account\_id) references accounts (id),

check (remaining\_tickets <= total\_tickets)

);

create table payment\_types (

id int auto\_increment primary key,

name varchar(255) not null unique,

payment\_phone varchar(255) not null,

created\_at datetime default CURRENT\_TIMESTAMP,

updated\_at datetime default CURRENT\_TIMESTAMP on update CURRENT\_TIMESTAMP

);

create table bookings (

id int auto\_increment primary key,

code varchar(255) not null unique,

package\_id int not null,

account\_id int not null,

ticket\_counts int not null,

unit\_price decimal(10, 2) not null,

booking\_status enum('PENDING', 'REQUESTING', 'RESERVED', 'CANCELLED') default 'PENDING',

created\_at datetime default CURRENT\_TIMESTAMP,

updated\_at datetime default CURRENT\_TIMESTAMP on update CURRENT\_TIMESTAMP,

foreign key (package\_id) references packages (id),

foreign key (account\_id) references accounts (id)

);

create table payments (

id int auto\_increment primary key,

code varchar(255) not null unique,

booking\_id int not null,

payment\_status enum('PENDING', 'SUCCESS', 'FAIL') default 'PENDING',

account\_id int null,

payment\_type\_id int not null,

created\_at datetime default CURRENT\_TIMESTAMP,

updated\_at datetime default CURRENT\_TIMESTAMP on update CURRENT\_TIMESTAMP,

foreign key (booking\_id) references bookings (id),

foreign key (account\_id) references accounts (id),

foreign key (payment\_type\_id) references payment\_types (id)

);

**2.3.2 Data Insertion**

**Create Admin Accounts**

INSERT INTO accounts (name, email, password, phone, account\_role) VALUES

('admin1', 'admin1@gmail.com', 'password', '09999888777', 'ADMIN'),

('admin2', 'admin2@gmail.com', 'password', '09999888666', 'ADMIN');

**Create Customer Accounts**

INSERT INTO accounts (name, email, password, phone) VALUES

('Aung Aung', 'aung@gmail.com', 'password', '09777888777'),

('Su Su', 'su@gmail.com', 'password', '09777888666'),

('Moe Moe', 'moe@gmail.com', 'password', '09777888555'),

('Hla Hla', 'hla@gmail.com', 'password', '09777888444');

**Create Categories**

INSERT INTO categories (name) VALUES

('Relaxation'), ('Pagoda'), ('Beach'), ('History');

**Create Locations**

INSERT INTO locations (name) VALUES

('Yangon'),

('Bagan'),

('Chaung Tha'),

('Mandalay'),

('Naypyitaw'),

('Inle Lake');

**Create Payment Types**

INSERT INTO payment\_types (name, payment\_phone) VALUES

('KBZ Pay', '09990001111'),

('Wave Pay', '09990002222'),

('AYA Pay', '09990003333');

**Create Packages**

INSERT INTO packages

(code, title, overview, category\_id, location\_id, departure\_date, duration, total\_tickets, remaining\_tickets, unit\_price, account\_id, package\_status)

VALUES

-- Category 1 (Relaxation)

('PKG-001', 'Adventure in Mountains', 'Explore the majestic mountains with guided tours.', 1, 1, '2025-10-01', 5, 20, 18, 500.00, 1, 'AVAILABLE'),

('PKG-002', 'Mountain Hiking Challenge', 'A thrilling hiking experience for adventure seekers.', 1, 2, '2025-11-05', 7, 15, 12, 650.00, 1, 'AVAILABLE'),

-- Category 2 (Pagoda)

('PKG-003', 'Beach Relaxation Getaway', 'Relax on pristine beaches with all-inclusive amenities.', 2, 3, '2025-09-15', 4, 30, 29, 400.00, 1, 'AVAILABLE'),

('PKG-004', 'Sunset Cruise Escape', 'Enjoy sunset cruises and beach parties.', 2, 4, '2025-10-20', 3, 25, 24, 350.00, 1, 'AVAILABLE'),

-- Category 3 (Beach)

('PKG-005', 'City Cultural Tour', 'Discover historical landmarks and local culture.', 3, 5, '2025-12-01', 6, 20, 18, 300.00, 1, 'AVAILABLE'),

('PKG-006', 'Nightlife Exploration', 'Experience the vibrant city nightlife and local cuisine.', 3, 6, '2025-12-10', 5, 15, 15, 320.00, 1, 'AVAILABLE'),

-- Category 4 (History)

('PKG-007', 'Ancient Pagoda Tour', 'Visit historic pagodas and learn about Burmese history.', 4, 2, '2025-11-12', 4, 25, 25, 450.00, 1, 'AVAILABLE'),

('PKG-008', 'Historical Yangon Walk', 'A walking tour through Yangon''s historical sites.', 4, 1, '2025-12-05', 3, 20, 20, 280.00, 1, 'AVAILABLE'),

-- Extra Package: UNAVAILABLE (sold out)

('PKG-009', 'Hidden Lakes Adventure', 'A secret journey to hidden lakes with full bookings.', 1, 6, '2025-11-20', 3, 10, 0, 600.00, 1, 'UNAVAILABLE'),

-- Extra Package: FINISHED (departure already past)

('PKG-010', 'Old Kingdom Exploration', 'Historic kingdom tour, already departed.', 4, 2, '2025-08-01', 4, 12, 5, 700.00, 1, 'FINISHED');

**Create Bookings**

INSERT INTO bookings

(code, package\_id, account\_id, ticket\_counts, unit\_price, booking\_status)

VALUES

('BOOK-001', 1, 3, 2, 500.00, 'PENDING'),

('BOOK-002', 3, 4, 1, 400.00, 'RESERVED'),

('BOOK-003', 2, 3, 3, 650.00, 'REQUESTING'),

('BOOK-004', 5, 4, 2, 300.00, 'PENDING'),

('BOOK-005', 4, 3, 1, 350.00, 'CANCELLED');

**Create Payments**

INSERT INTO payments

(code, booking\_id, payment\_status, account\_id, payment\_type\_id)

VALUES

('PAY-001', 1, 'PENDING', NULL, 1),

('PAY-002', 2, 'SUCCESS', 4, 2),

('PAY-003', 3, 'FAIL', 3, 1),

('PAY-004', 4, 'PENDING', NULL, 3),

('PAY-005', 5, 'SUCCESS', 3, 2);