🎯 Hackathon Management System: Project Report 📝

🎉 Project Overview

This ***Hackathon Management System*** is built using **Oracle SQL** to streamline the management of hackathons, participants, and ticketing. Organizers can manage hackathon details, track participant registrations, handle payments, and enforce event capacity limits. Key features include the ability to manage multiple hackathons, track ticket statuses, and run queries for insightful data analysis.

🌟 Features

- **Manage Hackathons**: Create and manage multiple hackathon events, including setting capacity and registration fees.
- **Participant Management**: Register participants, track personal details, and monitor registration statuses.
- **Ticketing System**: Organize ticket distribution, track payment status, and ensure the event stays within capacity limits.
- **Dynamic Status Updates**: Update participant registration statuses, such as "Pending," "Confirmed," or "Cancelled."
- **Queries for Insight**: Retrieve details of upcoming events and registered participants with easy-to-execute queries.

🗂️ Database Schema

1. Hackathons

This table manages hackathon events. It tracks the event name, dates, location, capacity, and registration fee.

sql

```
CREATE TABLE Hackathons (
  hackathon_id NUMBER(5) PRIMARY KEY,
  name VARCHAR2(100),
  start_date DATE,
  end_date DATE,
  location VARCHAR2(150),
  capacity NUMBER(5) CHECK (capacity > 0),
  registration_fee NUMBER(7,2) DEFAULT 0
);
```

Columns:

- hackathon_id: Unique identifier for each hackathon.
- name: Name of the event.
- start_date & end_date: Event duration.
- location: Venue of the hackathon.
- capacity: Maximum number of participants allowed.
- registration_fee: Registration cost for participants.

2. Participants

Stores details of participants including their contact information and registration date.

sql

```
CREATE TABLE Participants (
  participant_id NUMBER(5) PRIMARY KEY,
  first_name VARCHAR2(50),
  last_name VARCHAR2(50),
  email VARCHAR2(100),
  phone_number VARCHAR2(15),
  registration_date DATE DEFAULT SYSDATE
);
```

Columns:

- participant_id: Unique identifier for each participant.
- first_name & last_name: Participant's name.
- email: Contact email.
- phone_number: Contact phone number.
- registration_date: Date of registration.

3. Organizers

This table stores information about hackathon organizers and their roles in the event.

sql

```
CREATE TABLE Organizers (
   organizer_id NUMBER(5) PRIMARY KEY,
   first_name VARCHAR2(50),
   last_name VARCHAR2(50),
   email VARCHAR2(100),
   phone_number VARCHAR2(15),
   role VARCHAR2(50)
);
```

Columns:

- organizer_id: Unique identifier for each organizer.
- first_name & last_name: Name of the organizer.
- email: Organizer's email address.
- phone_number: Contact number.
- role: The role of the organizer in the event (e.g., Event Manager, Technical Lead).

4. Tickets

Tracks participants' registration and payment status for specific hackathons.

sql

```
CREATE TABLE Tickets (
   ticket_id NUMBER(5) PRIMARY KEY,
   hackathon_id NUMBER(5),
   participant_id NUMBER(5),
   registration_status VARCHAR2(20) DEFAULT 'Pending',
   payment_status VARCHAR2(20) DEFAULT 'Unpaid',
   CONSTRAINT fk_hackathon FOREIGN KEY (hackathon_id) REFERENCES Hackathons(hackathon_id),
   CONSTRAINT fk_participant FOREIGN KEY (participant_id) REFERENCES Participants(participant_id)
```

);

Columns:

- ticket_id: Unique ticket identifier.
- hackathon_id: References the hackathon being registered for.
- participant_id: References the participant.
- registration_status: Tracks the participant's registration status (Pending, Confirmed, Cancelled).
- payment_status: Tracks the payment status (Paid, Unpaid).

🔄 Key SQL Operations

📅 List Upcoming Hackathons:

Retrieve all hackathons that have not yet started.

sql

```
SELECT * FROM Hackathons WHERE start_date > SYSDATE;
```

👤 Find Participants for a Hackathon:

List participants registered for a specific hackathon (e.g., hackathon_id = 101).

sql

```
SELECT Participants.first_name, Participants.last_name, Tickets.registration_status
FROM Tickets
JOIN Participants ON Tickets.participant_id = Participants.participant_id
WHERE Tickets.hackathon_id = 101;
```

✅ Display Confirmed and Paid Participants:

Find participants who have confirmed their registration and made payment.

sql

```
SELECT Participants.first_name, Participants.last_name, Hackathons.name, Tickets.registration_status, Tickets.payment_status
FROM Tickets
```

JOIN Participants ON Tickets.participant_id = Participants.participant_id

JOIN Hackathons ON Tickets.hackathon_id = Hackathons.hackathon_id

WHERE Tickets.registration_status = 'Confirmed' AND Tickets.payment_status = 'Paid';

❌ Cancel a Registration:

Update the status of a ticket to "Cancelled."

sql

UPDATE Tickets

SET registration_status = 'Cancelled'

WHERE ticket_id = 402;

🧪 Sample Data

Hackathons:

sql

INSERT INTO Hackathons VALUES (101, 'AI Revolution Hackathon', TO_DATE('2024-12-01', 'YYYY-MM-DD'), TO_DATE('2024-12-03', 'YYYY-MM-DD'), 'TechPark, City', 200, 500);

INSERT INTO Hackathons VALUES (102, 'Blockchain Bonanza', TO_DATE('2024-11-10', 'YYYY-MM-DD'), TO_DATE('2024-11-12', 'YYYY-MM-DD'), 'Innovation Center, City', 150, 300);

Participants:

sql

INSERT INTO Participants VALUES (201, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', TO_DATE('2024-10-15', 'YYYY-MM-DD'));

INSERT INTO Participants VALUES (202, 'Jane', 'Smith', 'jane.smith@example.com', '098-765-4321', TO_DATE('2024-10-16', 'YYYY-MM-DD'));

Tickets:

sql

INSERT INTO Tickets VALUES (401, 101, 201, 'Confirmed', 'Paid');

INSERT INTO Tickets VALUES (402, 102, 202, 'Pending', 'Unpaid');

🚀 How to Use

1. **Create the schema** using the SQL commands provided.
2. **Insert sample data** to populate the hackathon system.
3. **Run queries** to manage events, participants, and tickets.

🔮 Future Enhancements

- **Triggers**: Automate checks for capacity limits and payment status updates.
- **Web Interface**: Provide an interface for participants and organizers to interact with the system.
- **Detailed Reporting**: Generate comprehensive reports for event organizers to track event performance and participant statistics.