1. Employee-Department Database

Create the following tables:

- Employee (EmpID, EmpName, DeptID, Salary, DOJ)
- Department (DeptID, DeptName, Location)

Sub-Questions:

1. Write a PL/SQL procedure to add a new employee. The procedure should raise an exception if the Salary is less than 10,000.
2. Create a function that takes DeptID as input and returns the total salary of all employees in that department.
3. Create a trigger to automatically update the department's total employee count when a new employee is added or removed.
4. Write a PL/SQL block that uses a cursor to increase the salary of all employees in a given department by 5%.
5. Modify the procedure to handle exceptions such as invalid department ID or salary constraints and raise appropriate error messages.
6. Create a package that contains the procedure for inserting a new employee and a function to calculate the total salary of employees in a department.

2. Library Management System

Create the following tables:

- Books (BookID, Title, Author, ISBN, PubYear)
- Borrowers (BorrowerID, BorrowerName, BookID, BorrowDate)

Sub-Questions:

1. Write a PL/SQL procedure to insert a new borrower and update the book’s availability status. Raise an exception if the borrower has already borrowed more than 3 books.
2. Create a function that takes BookID as input and returns whether the book is available or not.
3. Create a trigger to automatically update the book’s availability status when a book is borrowed or returned.
4. Write a PL/SQL block using a cursor to display all borrowers who have not returned a book for more than 30 days.
5. Modify the procedure to handle exceptions such as non-existent book ID or if a borrower tries to borrow an unavailable book.
6. Create a package that includes the procedure for borrowing a book, the function to check availability, and any necessary variables/constants.

3. Sales-Orders Database

Create the following tables:

- Orders (OrderID, CustomerID, ProductID, OrderDate, Quantity)
- Products (ProductID, ProductName, Price, Stock)

Sub-Questions:

1. Write a PL/SQL procedure to place a new order. If the ordered quantity exceeds available stock, raise an exception.
2. Create a function that takes OrderID as input and returns the total amount for that order (Quantity * Price).
3. Create a trigger to automatically update the stock level of the product after an order is placed.
4. Write a PL/SQL block using a cursor to calculate the total sales for each product at the end of the day.
5. Modify the procedure to handle exceptions, such as ordering more than available stock or attempting to place an order for a non-existent product.
6. Create a package that includes the procedure to place an order, the function to calculate the order total, and any necessary constants or exceptions.

4. University Database

Create the following tables:

- Students (StudentID, StudentName, DeptID, CGPA)
- Departments (DeptID, DeptName)

Sub-Questions:

1. Write a PL/SQL procedure to add a new student. The procedure should raise an exception if the CGPA is not between 0 and 10.
2. Create a function that returns the average CGPA of students in a specific department.
3. Create a trigger to automatically update the department's average CGPA when a student’s CGPA is updated.
4. Write a PL/SQL block using a cursor to list all students whose CGPA is below a specified value and print a warning message.
5. Modify the procedure to handle invalid DeptID or CGPA values and raise an appropriate error.
6. Package: Create a package that contains the student insertion procedure, the function to calculate average CGPA, and the exception handlers.

5. E-commerce Database

Create the following tables:

- Products (ProductID, ProductName, Price, Stock)
- Orders (OrderID, CustomerID, ProductID, OrderDate, Quantity)

Sub-Questions:

1. Write a PL/SQL procedure to add a new product. The procedure should raise an exception if the price is less than or equal to 0.
2. Create a function that takes ProductID as input and returns the total stock value of the product (Stock * Price).
3. Create a trigger that automatically reduces the stock level when an order is placed and raises an exception if stock falls below a threshold.
4. Write a PL/SQL block using a cursor to list all products that have low stock (less than 10 units) and print a message for each.
5. Modify the product insertion procedure to handle cases like negative stock or invalid price, raising a custom error message.
6. Create a package that includes the product insertion procedure, the stock value function, and a set of constants for minimum stock levels.

DBMS LAB EXAM VIVA 21-10-2024

1. Command to switch databases in MongoDB?
2. Command to list all databases in MongoDB?
3. Command to display the current database?
4. Command to drop a database in MongoDB?
5. Data format used to store records in MongoDB?
6. Method to insert a single document?
7. Method to insert multiple documents?
8. Method to retrieve documents in MongoDB?
9. Method to update a single document?
10. Operation to delete a collection in MongoDB?
11. Command to commit a transaction?
12. Command to undo changes in a transaction?
13. Command to set a savepoint within a transaction?
14. Command to release a savepoint?
15. Command to revert to a specific savepoint?
16. Command to grant privileges to a user?
17. Command to revoke privileges from a user?
18. Command to begin a transaction explicitly?
19. Command to temporarily prevent changes until a transaction is committed?
20. Command to allow read consistency during a transaction?
21. What is the format MongoDB uses to store data?
22. What command is used to create a new MongoDB database?
23. Which method is used to create a collection in MongoDB?
24. Which field specifies if a collection is capped?
25. What command drops a collection in MongoDB?
26. insertOne method is used to insert a in MongoDB?
27. What method is used to retrieve documents from a collection?
28. What method returns only one document from a MongoDB collection?
29. Which keyword is used for the AND condition in MongoDB queries?
30. Which method is used to update a single document in MongoDB?

1. Hospital Management System

Create the following tables:

- Patients (PatientID, Name, Age, Diagnosis)
- Doctors (DoctorID, Name, Specialization)

Sub-Questions:

1. Write a procedure to assign a patient to a doctor based on specialization.
2. Write a function to count the number of patients under a specific doctor.
3. Create a trigger to update the number of patients assigned to a doctor when a new patient is added.
4. Write a cursor to list all patients diagnosed with a specific disease.
5. Modify the procedure to handle exceptions like assigning a non-existent doctor.
6. Create a package with procedures for patient assignment and functions to track doctor workload.

2. Hotel Management System

Create the following tables:

- Rooms (RoomID, RoomType, Rate, Status)
- Bookings (BookingID, CustomerID, RoomID, BookingDate)

Sub-Questions:

1. Write a procedure to book a room and update the room status.
2. Create a function to return the total earnings for a given room.
3. Create a trigger to automatically mark a room as unavailable when it is booked.
4. Write a cursor to display all available rooms of a certain type.
5. Handle exceptions for booking unavailable rooms or invalid customer IDs.
6. Create a package with room booking procedures and functions to calculate room earnings.

3. Attendance System

Create the following tables:

- Attendance (AttendanceID, StudentID, Date, Status)
- Students (StudentID, StudentName, DeptID)

Sub-Questions:

1. Write a procedure to mark attendance for students.
2. Create a function to calculate the total attendance percentage for a student.
3. Create a trigger to update attendance records when a student is marked absent.
4. Write a cursor to list all students with attendance below a certain percentage.
5. Handle exceptions for marking attendance for invalid student IDs.
6. Create a package with procedures for managing attendance and functions to calculate attendance percentage.

4. Online Exam System

Create the following tables:

- Exams (ExamID, ExamName, MaxMarks)
- Results (ResultID, StudentID, ExamID, MarksObtained)

Sub-Questions:

1. Write a procedure to add new exam results for a student.
2. Create a function to calculate the percentage of marks obtained by a student.
3. Create a trigger to update the student's total score when a new result is added.
4. Write a cursor to display students who have scored above a certain percentage.
5. Handle exceptions for invalid ExamID or MarksObtained exceeding the maximum marks.
6. Create a package with procedures for adding results and functions to calculate percentages.

5. Flight Reservation System

Create the following tables:

- Flights (FlightID, FlightName, Departure, Arrival, SeatsAvailable)
- Reservations (ReservationID, CustomerID, FlightID, Date)

Sub-Questions:

1. Write a procedure to book a flight and update available seats.
2. Create a function to return the total number of seats booked for a specific flight.
3. Create a trigger to prevent booking if no seats are available.
4. Write a cursor to list all available flights on a given date.
5. Handle exceptions like overbooking or invalid flight IDs.
6. Create a package with procedures for flight booking and functions to check available seats.

6. Student Management System

Create the following tables:

- Students (StudentID, Name, DeptID, Age)
- Courses (CourseID, CourseName, DeptID)

Sub-Questions:

1. Write a procedure to enroll a student in a course.
2. Create a function to count the number of students enrolled in a specific course.
3. Create a trigger to prevent a student from enrolling in more than 5 courses.
4. Write a cursor to display students enrolled in a specific department.
5. Handle exceptions for invalid course IDs or student enrollments.
6. Create a package with procedures for course enrollment and functions for counting students