PL/SQL Assignments on HR Schema

1. Employee Bonus Calculation (Procedures, Exceptions, Cursors)

Create a procedure that calculates a bonus for each employee based on salary and job title.
- Use an explicit cursor to iterate over employees.
- If salary > 10,000, bonus = 10%; otherwise, bonus = 15%.
- Handle exceptions where salary is NULL and log errors into an error table.

2. Department Salary Budget (Functions, Records, Loops, Exception Handling)

Develop a function that accepts a department ID and returns the total salary budget.
- Use %ROWTYPE to fetch employee details.
- Loop through employee records using a FOR LOOP and sum their salaries.
- If department ID does not exist, raise a custom exception.

3. Employee Promotion (Triggers, Cursors, Dynamic SQL, Bulk Collect)

Implement a trigger that updates an employee's job title and salary when their performance rating is updated.
- Use explicit cursors to fetch employee details.
- Apply bulk collect to handle multiple employee records efficiently.
- Ensure salary updates follow the company's dynamic business rules using Dynamic SQL.

4. Employee Hiring System (Procedures, Dynamic SQL, Autonomous Transactions)

Create a procedure to insert new employee records.
- Use Dynamic SQL to insert into different tables based on department type (e.g., IT, HR, Sales).
- Implement an autonomous transaction to commit data independently.

5. Employee Hierarchy Report (Recursion, Cursors, Table Functions)

Develop a table function that returns the hierarchy of an employee up to the CEO level.
- Use recursive logic to traverse the manager-subordinate hierarchy.
- Utilize cursors to fetch employee-manager relationships.

6. Department Reorganization (Collections, Bulk Collect, Updates, Exception Handling)

Move all employees from one department to another.
- Store their previous department details in a history table.
- Use collections and bulk collect for faster processing.
- Implement exception handling for scenarios where a department does not exist.

7. Employee Performance History (Triggers, Collections, Procedures, Exception Handling)

Implement a trigger that logs salary updates into a salary_history table.
- Store old and new values in a collection before inserting into the log.
- Handle exceptions when an employee has no salary record.

8. Dynamic Employee Reports (Dynamic SQL, Records, Cursors, CASE Statement)

Write a procedure that generates a report based on user inputs:
- Department
- Job title
- Hire date range
- Use Dynamic SQL to build queries dynamically and store fetched data into records.

9. Top Performers by Department (Bulk Collect, Table Functions, Cursors, Loops)

Develop a table function that returns the top 3 highest-paid employees in each department.
- Use bulk collect and cursor loops to process multiple records efficiently.

10. Employee Transfer Log (Autonomous Transactions, Exception Handling, Triggers)

Implement a trigger that logs department transfers into a transfer_log table.
- Use autonomous transactions to ensure logs are committed independently.
- Handle exceptions where an employee is transferred to a non-existent department.

11. Salary Adjustment by Job Type (Procedures, Dynamic SQL, Cursors, Records, Exception Handling)

Create a procedure to adjust salaries based on job title:
- Technical jobs: Increase by 5%
- Managerial jobs: Increase by 10%
- Use explicit cursors to fetch employee details and apply Dynamic SQL for flexible query execution.

12. Employee Leave Management (Collections, Procedures, Exception Handling, Bulk Collect)

Implement a procedure to process employee leave requests.
- Store leave requests in a collection.
- Validate leave balance before approving.
- Use bulk collect for performance optimization.
- Handle exceptions where employees exceed leave limits.

13. Departmental Headcount Analysis (Functions, Collections, Bulk Collect, Loops)

Write a function that returns department-wise headcount.
- Use collections and bulk collect to store employee data efficiently.
- Display only departments where employee count > 10.

14. Employee Resignation Process (Procedures, Cursors, Dynamic SQL, Exception Handling)

Write a procedure to process employee resignations.
- The procedure should:
- Update employee status
- Remove login credentials
- Transfer responsibilities
- Use Dynamic SQL and cursors to update records.

15. Dynamic Table Creation for Employee Logs (Dynamic SQL, Procedures, Triggers, Autonomous Transactions)

Create a procedure that dynamically creates log tables based on current year (e.g., employee_log_2025).
- Implement a trigger to insert employee updates into the correct table.

16. Employee Appraisal System (Functions, Cursors, CASE Statements, Bulk Collect)

Develop a function to evaluate employee appraisals based on:
- Performance scores
- Current salary
- Use CASE statements to determine the salary increase.
- Fetch records using bulk collect.

17. Payroll System with Error Handling (Procedures, Cursors, Exception Handling, Bulk Collect)

Implement a payroll system using a procedure that calculates net salary after deductions.
- Use bulk collect to process multiple employees.
- If an employee has missing salary details, log errors separately and continue processing.

18. Department Budget Forecasting (Table Functions, Bulk Collect, Aggregations, Cursors)

Create a table function that forecasts department budgets for the next 5 years.
- Use bulk collect and cursors to fetch and process salary trends.

19. Employee Role Change Process (Procedures, Triggers, Dynamic SQL, Exception Handling, Autonomous Transactions)

Write a procedure to process employee role changes:
- Update job title
- Modify salary based on role
- Trigger a notification system
- Use Dynamic SQL and autonomous transactions for data integrity.

20. Cross-Database Employee Audit (Database Links, Cursors, Procedures, Exception Handling)

Implement a procedure that fetches employee details from a remote HR database via a database link.
- Insert fetched records into a local audit table.
- Handle connection errors and missing data scenarios.