**WINAIM - Backend Database Programming Assignments**

**SQL Assessment**

**### Exercise 1: Employee Management System**

\*\*Objective\*\*: To test the ability to design relational databases, write complex SQL queries, and handle data relationships.

#### Requirements:

1. \*\*Database Schema Design\*\*:

- Create tables for `employees`, `departments`, and `salaries`.

- `employees` table should include: `employee\_id`, `first\_name`, `last\_name`, `department\_id`, `hire\_date`.

- `departments` table should include: `department\_id`, `department\_name`.

- `salaries` table should include: `employee\_id`, `salary`, `from\_date`, `to\_date`.

2. \*\*SQL Queries\*\*:

- Write a query to find all employees who have been hired in the last year.

- Write a query to calculate the total salary expenditure for each department.

- Write a query to find the top 5 highest-paid employees along with their department names.

---

**### Exercise 2: Online Retailer Database**

\*\*Objective\*\*: To evaluate the ability to work with transactional data, aggregate functions, and subqueries.

#### Requirements:

1. \*\*Database Schema Design\*\*:

- Create tables for `customers`, `orders`, `products`, and `order\_items`.

- `customers` table should include: `customer\_id`, `first\_name`, `last\_name`, `email`.

- `orders` table should include: `order\_id`, `customer\_id`, `order\_date`.

- `products` table should include: `product\_id`, `product\_name`, `price`.

- `order\_items` table should include: `order\_item\_id`, `order\_id`, `product\_id`, `quantity`, `price`.

2. \*\*SQL Queries\*\*:

- Write a query to find the total number of orders placed by each customer.

- Write a query to list the products that have never been ordered.

- Write a query to find the customer who has spent the most money in the last month.

---

**### Exercise 3: University Course Management**

\*\*Objective\*\*: To assess skills in managing complex data relationships, normalization, and advanced SQL queries.

#### Requirements:

1. \*\*Database Schema Design\*\*:

- Create tables for `students`, `courses`, `enrollments`, and `professors`.

- `students` table should include: `student\_id`, `first\_name`, `last\_name`, `enrollment\_date`.

- `courses` table should include: `course\_id`, `course\_name`, `professor\_id`.

- `enrollments` table should include: `enrollment\_id`, `student\_id`, `course\_id`, `grade`.

- `professors` table should include: `professor\_id`, `first\_name`, `last\_name`, `department`.

2. \*\*SQL Queries\*\*:

- Write a query to find the total number of students enrolled in each course.

- Write a query to list the courses taught by professors in the 'Computer Science' department.

- Write a query to calculate the average grade for each course.

- Write a query to find students who are enrolled in more than three courses.

**Programming Assessment**

**### Exercise 1: Customer Relationship Management (CRM) System**

\*\*Objective\*\*: To evaluate skills in managing customer data, handling complex queries, and implementing business logic.

\*\*Requirements\*\*:

- \*\*Backend\*\*: Design a database schema for a CRM system with tables for customers, contacts, opportunities, and interactions.

- \*\*API\*\*: Develop RESTful APIs to manage customer records, track interactions, and update opportunities.

- \*\*Frontend\*\*: Create a web interface for sales representatives to manage customer relationships.

- \*\*Functionalities\*\*:

- Add, edit, delete, and view customer records.

- Log interactions (calls, meetings, emails) with customers.

- Track sales opportunities and their stages.

- Implement a dashboard to show key metrics like number of opportunities, stages, and customer interaction history.

**### Exercise 2: Human Resources Management System (HRMS)**

\*\*Objective\*\*: To assess knowledge in managing employee data, role-based access control, and complex data relationships.

\*\*Requirements\*\*:

- \*\*Backend\*\*: Design a database schema for an HRMS with tables for employees, departments, roles, and performance reviews.

- \*\*API\*\*: Develop RESTful APIs for managing employee records, department assignments, and performance reviews.

- \*\*Frontend\*\*: Build a web interface for HR managers to manage employee data.

- \*\*Functionalities\*\*:

- Add, edit, delete, and view employee records.

- Assign employees to departments and roles.

- Conduct and record performance reviews.

- Implement role-based access control for different user types (HR managers, employees).

**### Exercise 3: Inventory Management System**

\*\*Objective\*\*: To test skills in inventory tracking, real-time updates, and reporting.

\*\*Requirements\*\*:

- \*\*Backend\*\*: Design a database schema for an inventory management system with tables for products, suppliers, warehouses, and stock levels.

- \*\*API\*\*: Develop RESTful APIs to manage product inventory, track stock levels, and update warehouse information.

- \*\*Frontend\*\*: Create a web interface for warehouse managers to monitor and manage inventory.

- \*\*Functionalities\*\*:

- Add, edit, delete, and view product records.

- Track stock levels and update in real-time.

- Generate reports on stock levels, low inventory alerts, and product movements.

- Integrate barcode scanning for quick updates.

---

**### Exercise 4: Project Management System**

\*\*Objective\*\*: To evaluate the ability to manage projects, tasks, timelines, and team collaboration.

\*\*Requirements\*\*:

- \*\*Backend\*\*: Design a database schema for a project management system with tables for projects, tasks, teams, and timelines.

- \*\*API\*\*: Develop RESTful APIs to manage projects, assign tasks, and track progress.

- \*\*Frontend\*\*: Build a web interface for project managers and team members to collaborate on projects.

- \*\*Functionalities\*\*:

- Create, update, delete, and view projects and tasks.

- Assign tasks to team members and set deadlines.

- Track project progress with Gantt charts or Kanban boards.

- Implement notifications and reminders for upcoming deadlines and task updates.