Lab 3 Assignment

Mohamed Emary

February 12, 2025

1 Assignment Questions

1.1 Question 1

> Create a PL/pgSQL block using cursor to update employee commission percentages based on salary ranges:
>
> - SALARY < 7000: COMM = 0.1
> - 7000 <= SALARY < 10000: COMM = 0.15
> - 10000 <= SALARY < 15000: COMM = 0.2
> - 15000 <= SALARY: COMM = 0.25

```
DO $$
DECLARE
    v_emp_cursor CURSOR FOR
        SELECT employee_id, salary
        FROM employees;
    v_commission NUMERIC(2,2);
BEGIN
    FOR v_emp IN v_emp_cursor LOOP
        -- Determine commission based on salary range
        IF v_emp.salary < 7000 THEN
            v_commission := 0.10;
        ELSIF v_emp.salary < 10000 THEN
            v_commission := 0.15;
        ELSIF v_emp.salary < 15000 THEN
            v_commission := 0.20;
        ELSE
            v_commission := 0.25;
        END IF;

        -- Update employee commission
        UPDATE employees
        SET commission_pct = v_commission
        WHERE employee_id = v_emp.employee_id;
    END LOOP;
END $$;
```

1.2 Question 2

> Alter table employees to add column `retired_bonus` and create a PL/pgSQL block to:
> - Calculate retired bonus for employees with 18+ years service
> - Bonus = `working_months * (10% of current salary)`
> - Update the `retired_bonus` column

```
-- Add retired_bonus column
ALTER TABLE employees ADD COLUMN retired_bonus NUMERIC(12,2);

-- Calculate and update retired bonus
DO $$
DECLARE
    v_emp_cursor CURSOR FOR
        SELECT employee_id, salary, hire_date
        FROM employees
        WHERE EXTRACT(YEAR FROM AGE(NOW(), hire_date)) >= 18;
    v_months INT;
    v_bonus NUMERIC(12,2);
BEGIN
    FOR v_emp IN v_emp_cursor LOOP
        -- Calculate months of service
        v_months := EXTRACT(YEAR FROM AGE(NOW(), v_emp.hire_date)) * 12 +
                    EXTRACT(MONTH FROM AGE(NOW(), v_emp.hire_date));

        -- Calculate bonus amount
        v_bonus := v_months * (v_emp.salary * 0.10);

        -- Update employee's retired bonus
        UPDATE employees
        SET retired_bonus = v_bonus
        WHERE employee_id = v_emp.employee_id;
    END LOOP;
END $$;
```

1.3 Question 3

> Create a PL/pgSQL block to increase `salary` by 10% for all employees in the IT department.

```
DO $$
BEGIN
    UPDATE employees
    SET salary = salary * 1.10
    WHERE department_id = (
        SELECT department_id
        FROM departments
        WHERE UPPER(department_name) = 'IT'
    );
END $$;
```

1.4 Question 4

> Create a PL/pgSQL block to award a 500 bonus to employees who:
> - Have worked for more than 15 years
> - Have no `commission_pct`

```
DO $$
DECLARE
    v_emp_cursor CURSOR FOR
        SELECT employee_id
        FROM employees
        WHERE EXTRACT(YEAR FROM AGE(NOW(), hire_date)) > 15
        AND commission_pct IS NULL;
BEGIN
    FOR v_emp IN v_emp_cursor LOOP
        UPDATE employees
        SET salary = salary + 500
        WHERE employee_id = v_emp.employee_id;
    END LOOP;
END $$;
```

1.5 Question 5

> Create a PL/pgSQL block to give a 5% bonus to employees who:
> - Work in the Sales department
> - Have `salary` greater than 8000

```
DO $$
BEGIN
    UPDATE employees
    SET salary = salary * 1.05
    WHERE department_id = (
        SELECT department_id
        FROM departments
        WHERE UPPER(department_name) = 'SALES'
    )
    AND salary > 8000;
END $$;
```

1.6 Question 6

> Create a retirement check system:
> - Add `RETIRED INT` column to employees table
> - Create `CHECK_RETIRED` function that takes `employee_id` and `max_hire_years`
> - Return `1` if employee has worked >= `max_hire_years`, else return `0`
> - Create block to update retired status for all employees

```
-- Add RETIRED column
ALTER TABLE employees ADD COLUMN RETIRED INT DEFAULT 0;
```

Question 6

```sql
-- Create check_retired function
CREATE OR REPLACE FUNCTION check_retired(
    v_emp_id INT,
    v_max_years INT
)
RETURNS INT
LANGUAGE plpgsql
AS $$
DECLARE
    v_years INT;
BEGIN
    SELECT EXTRACT(YEAR FROM AGE(NOW(), hire_date))
    INTO v_years
    FROM employees
    WHERE employee_id = v_emp_id;

    RETURN CASE WHEN v_years >= v_max_years THEN 1 ELSE 0 END;
END $$;

-- Update retired status for all employees
DO $$
DECLARE
    v_emp_cursor CURSOR FOR
        SELECT employee_id
        FROM employees;
BEGIN
    FOR v_emp IN v_emp_cursor LOOP
        UPDATE employees
        SET retired = check_retired(v_emp.employee_id, 25)
        WHERE employee_id = v_emp.employee_id;
    END LOOP;
END $$;
```