

```

-- =====
-- Co-op Salary Explorer Database Schema
-- =====

-- Drop old tables if they exist (for testing)
DROP TABLE IF EXISTS Blacklist;
DROP TABLE IF EXISTS Salary;
DROP TABLE IF EXISTS JobPosting;
DROP TABLE IF EXISTS Employer;

-- -----
-- Employer Table
-- -----
CREATE TABLE Employer (
    employer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL UNIQUE,
    blacklist_flag BOOLEAN DEFAULT FALSE,
    CHECK (name <> "")
);

-- -----
-- JobPosting Table
-- -----
CREATE TABLE JobPosting (
    job_id INT AUTO_INCREMENT PRIMARY KEY,
    employer_id INT NOT NULL,
    title VARCHAR(255) NOT NULL,
    location VARCHAR(100),
    term VARCHAR(20),
    FOREIGN KEY (employer_id)
        REFERENCES Employer(employer_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CHECK (title <> "")
);

-- -----
-- Salary Table
-- -----
CREATE TABLE Salary (
    salary_id INT AUTO_INCREMENT PRIMARY KEY,
    job_id INT NOT NULL,
    hourly_rate DECIMAL(6,2) NOT NULL CHECK (hourly_rate >= 0),
    hours_per_week INT CHECK (hours_per_week BETWEEN 0 AND 80),
    notes TEXT,
    FOREIGN KEY (job_id)
        REFERENCES JobPosting(job_id)
        ON DELETE CASCADE

```

```

        ON UPDATE CASCADE
    );

-- -----
-- Blacklist Table
-- -----

CREATE TABLE Blacklist (
    blacklist_id INT AUTO_INCREMENT PRIMARY KEY,
    employer_id INT NOT NULL,
    reason TEXT NOT NULL,
    date_added DATE DEFAULT CURRENT_DATE,
    added_by VARCHAR(100),
    FOREIGN KEY (employer_id)
        REFERENCES Employer(employer_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- =====
-- Example Stored Procedure
-- =====
-- Adds a new blacklist entry and automatically updates the employer's blacklist flag
DELIMITER $$

CREATE PROCEDURE AddToBlacklist(
    IN emp_id INT,
    IN reason_text TEXT,
    IN added_by_user VARCHAR(100)
)
BEGIN
    -- Insert a new blacklist record
    INSERT INTO Blacklist (employer_id, reason, added_by)
    VALUES (emp_id, reason_text, added_by_user);

    -- Mark the employer as blacklisted
    UPDATE Employer
    SET blacklist_flag = TRUE
    WHERE employer_id = emp_id;
END$$

DELIMITER ;

-- =====
-- Trigger: Automatically update blacklist_flag
-- =====
DELIMITER $$

CREATE TRIGGER UpdateBlacklistFlag

```

```

AFTER INSERT ON Blacklist
FOR EACH ROW
BEGIN
    UPDATE Employer
    SET blacklist_flag = TRUE
    WHERE employer_id = NEW.employer_id;
END$$

DELIMITER ;

-- =====
-- Example Data Inserts (optional for testing)
-- =====
INSERT INTO Employer (name) VALUES ('Google'), ('Meta'), ('Amazon');

INSERT INTO JobPosting (employer_id, title, location, term)
VALUES
    (1, 'Software Engineer Intern', 'Waterloo, ON', 'Winter 2025'),
    (2, 'Data Analyst Co-op', 'Toronto, ON', 'Spring 2025');

INSERT INTO Salary (job_id, hourly_rate, hours_per_week)
VALUES
    (1, 38.50, 40),
    (2, 32.00, 37);

-- Example usage of the stored procedure
CALL AddToBlacklist(2, 'Reported unfair interview process', 'AdminUser');

```