Practical Database Lab Assignment 4

Module 4 – SQL Applications & Hands-on Queries

# **Objective 1: Create a Learning Database & Pupils Table**

-- Initiate a fresh database for academic purposes  
CREATE DATABASE IF NOT EXISTS academy\_db;  
USE academy\_db;  
  
-- Define a table to maintain pupil information  
CREATE TABLE pupils (  
 pupil\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(40),  
 age INT,  
 grade VARCHAR(15),  
 residence VARCHAR(100)  
);

# **Objective 2: Insert Sample Data into Pupils Table**

-- Insert mock records into pupils table  
INSERT INTO pupils(name, age, grade, residence) VALUES  
('Aanya Mehta', 9, 'Grade 3', 'Blue Orchid Complex'),  
('Reyansh Iyer', 8, 'Grade 2', 'Sunshine Valley'),  
('Kiara Joshi', 6, 'KG', 'Meadow Residency'),  
('Vivaan Kapoor', 10, 'Grade 4', 'Hilltop Society'),  
('Myra Shah', 7, 'Grade 2', 'Riverfront Homes');

# **Objective 3: Data Retrieval Operations**

-- Select all records  
SELECT \* FROM pupils;  
  
-- Select specific fields  
SELECT name, age FROM pupils;  
  
-- Filter pupils older than 6  
SELECT \* FROM pupils WHERE age > 6;

# **Objective 4: Teachers Table with Relationships**

-- Create a table to store faculty details with relational constraint  
CREATE TABLE faculty (  
 faculty\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 full\_name VARCHAR(40) NOT NULL,  
 teaching\_subject VARCHAR(30) NOT NULL,  
 email\_id VARCHAR(60) UNIQUE,  
 FOREIGN KEY (faculty\_id) REFERENCES pupils(pupil\_id)  
);  
  
-- Add faculty records  
INSERT INTO faculty(full\_name, teaching\_subject, email\_id) VALUES  
('Sneha Trivedi', 'Mathematics', 'sneha.trivedi123@mail.com'),  
('Kunal Rao', 'Science', 'kunal.rao98@mail.com');

# **Objective 5: University DB and Programs Table**

-- Set up university schema  
CREATE DATABASE IF NOT EXISTS uni\_db;  
USE uni\_db;  
  
-- Create programs table  
CREATE TABLE programs (  
 prog\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 title VARCHAR(50),  
 credits INT  
);  
  
-- Modify table to include duration, then remove credits  
ALTER TABLE programs ADD COLUMN duration VARCHAR(30);  
ALTER TABLE programs DROP COLUMN credits;

# **Objective 6: Dropping and Updating Tables**

-- Remove teacher & pupil tables  
DROP TABLE IF EXISTS faculty;  
DROP TABLE IF EXISTS pupils;  
  
-- Insert, update, and delete entries  
INSERT INTO programs(title, duration) VALUES  
('AI Basics', '6 Months'),  
('UI/UX Design', '4 Months'),  
('Cybersecurity', '8 Months');  
  
UPDATE programs SET duration='5 Months' WHERE prog\_id=2;  
  
DELETE FROM programs WHERE prog\_id=1;

# **Objective 7: Display and Limit Records**

-- Show all programs  
SELECT \* FROM programs;  
  
-- Sort programs in reverse alphabetical order  
SELECT \* FROM programs ORDER BY title DESC;  
  
-- Show top 2 programs only  
SELECT \* FROM programs LIMIT 2;

# **Objective 8: User Roles and Permissions**

-- Creating new users and adjusting privileges  
CREATE USER 'learner1'@'localhost' IDENTIFIED BY 'pass123';  
CREATE USER 'learner2'@'localhost' IDENTIFIED BY 'pass456';  
  
GRANT SELECT ON programs TO 'learner1'@'localhost';  
REVOKE INSERT ON programs FROM 'learner1'@'localhost';  
GRANT INSERT ON programs TO 'learner2'@'localhost';

# **Objective 9: Transactions, Commits, and Savepoints**

-- Inserting records and using transaction controls  
INSERT INTO programs(title, duration) VALUES  
('Web Dev', '10 Months'),  
('Data Science', '12 Months');  
  
COMMIT;  
  
INSERT INTO programs(title, duration) VALUES  
('Ethical Hacking', '6 Months');  
  
ROLLBACK;  
  
SAVEPOINT sp1;  
  
INSERT INTO programs(title, duration) VALUES  
('Animation', '9 Months');  
  
ROLLBACK TO SAVEPOINT sp1;

# **Objective 10: Department & Employee Join Operations**

CREATE DATABASE IF NOT EXISTS biz\_corp;  
USE biz\_corp;  
  
CREATE TABLE departments (  
 dept\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 dept\_name TEXT  
);  
  
INSERT INTO departments(dept\_name) VALUES  
('Tech Support'), ('Design Team'), ('Marketing'), ('Human Resources');  
  
CREATE TABLE staff (  
 staff\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 emp\_name TEXT,  
 dept\_name TEXT,  
 salary FLOAT  
);  
  
INSERT INTO staff(emp\_name, dept\_name, salary) VALUES  
('Aarav Jain', 'Design Team', 60000),  
('Neha Gupta', 'Marketing', 70000),  
('Rishabh Sen', 'Tech Support', 65000),  
('Divya Rao', 'Human Resources', 72000);  
  
-- Join queries  
SELECT \* FROM departments INNER JOIN staff ON departments.dept\_name = staff.dept\_name;  
SELECT departments.dept\_name, staff.dept\_name FROM departments LEFT JOIN staff ON departments.dept\_name = staff.dept\_name;

# **Objective 11: Aggregate Functions**

-- Count employees by department  
SELECT dept\_name, COUNT(\*) FROM staff GROUP BY dept\_name;  
  
-- Average salary per department  
SELECT dept\_name, AVG(salary) FROM staff GROUP BY dept\_name;

# **Objective 12: Using Stored Procedures**

-- Procedure to fetch all employees  
DELIMITER //  
CREATE PROCEDURE fetch\_employees()  
BEGIN  
 SELECT emp\_name, dept\_name FROM staff;  
END //  
DELIMITER ;  
  
CALL fetch\_employees();  
  
-- Procedure to get course duration  
DELIMITER //  
CREATE PROCEDURE get\_program(IN id INT)  
BEGIN  
 SELECT title, duration FROM programs WHERE prog\_id = id;  
END //  
DELIMITER ;  
  
CALL get\_program(3);

# **Objective 13: Triggers for Logging Changes**

-- Trigger on insert  
DELIMITER //  
CREATE TRIGGER log\_insert  
AFTER INSERT ON staff  
FOR EACH ROW  
BEGIN  
 INSERT INTO log\_table(message, logged\_at) VALUES('Employee Added', NOW());  
END //  
DELIMITER ;  
  
-- Trigger on update  
DELIMITER //  
CREATE TRIGGER log\_update  
AFTER UPDATE ON staff  
FOR EACH ROW  
BEGIN  
 INSERT INTO log\_table(message, logged\_at) VALUES('Record Updated', NOW());  
END //  
DELIMITER ;