**LUYỆN TẬP SQL – PTIT VERSION**

Địa chỉ: <http://36.50.134.216:3000/>

**SQL132**: select \* from learnsql

**SQL101:** select \* from Cinema

where id % 2 = 1 and description != "boring"

order by rating desc

**SQL129:** select tweet\_id from Tweets where length(content) > 15

**SQL127:** SELECT teacher\_id, COUNT(DISTINCT subject\_id) AS cnt

FROM Teacher

GROUP BY teacher\_id;

**SQL126:** SELECT s.user\_id,

       ROUND(COALESCE(SUM(CASE WHEN c.action = 'confirmed' THEN 1 ELSE 0 END) / COUNT(c.user\_id), 0), 2) AS confirmation\_rate

FROM Signups s

LEFT JOIN Confirmations c ON s.user\_id = c.user\_id

GROUP BY s.user\_id;

**SQL130:** SELECT p.product\_name, s.year, s.price

FROM Sales s

JOIN Product p ON s.product\_id = p.product\_id;

**SQL131:** SELECT distinct customer\_id FROM Customer c

WHERE NOT EXISTS (

    SELECT 1 FROM Product p

    WHERE NOT EXISTS (

        SELECT 1 FROM Customer c2

        WHERE c2.customer\_id = c.customer\_id AND c2.product\_key = p.product\_key

    )

)

GROUP BY customer\_id;

**SQL133:** SELECT

    p.product\_id,

    ROUND(SUM(u.units \* p.price) / SUM(u.units), 2) AS average\_price

FROM Prices p

LEFT JOIN UnitsSold u

    ON p.product\_id = u.product\_id

    AND u.purchase\_date BETWEEN p.start\_date AND p.end\_date

GROUP BY p.product\_id

**SQL119**: ALTER TABLE Employees

MODIFY COLUMN employee\_id INT;

**SQL139:** SELECT

    e.name,

    b.bonus

FROM

    Employee e

LEFT JOIN

    Bonus b

ON

    e.empId = b.empId

WHERE

    b.bonus < 1000 OR b.bonus IS NULL;

**SQL128**: SELECT class FROM Courses

GROUP BY class HAVING COUNT(student) >= 5;

**SQL136:** SELECT

    s.student\_id,

    s.student\_name,

    sub.subject\_name,

    COUNT(e.subject\_name) AS attended\_exams

FROM

    Students s

CROSS JOIN

    Subjects sub

LEFT JOIN

    Examinations e

ON

    s.student\_id = e.student\_id AND sub.subject\_name = e.subject\_name

GROUP BY

    s.student\_id, s.student\_name, sub.subject\_name

ORDER BY

    s.student\_id, sub.subject\_name;

**SQL138:** UPDATE Users SET name = CONCAT(UPPER(SUBSTRING(name, 1, 1)), LOWER(SUBSTRING(name, 2)));

1. **SQL110**

select \* from class where title like "Introduction%".

1. **SQL103**

SELECT T.\* FROM Teaches T

JOIN Class C ON T.dept = C.dept AND T.number = C.number

JOIN Instructor I ON T.username = I.username

ORDER BY I.lname DESC

LIMIT 2;

1. **SQL117**

ALTER TABLE party\_guests

MODIFY age INT,

MODIFY drinks\_count INT;

1. **SQL100**

select product\_id from Products where low\_fats = "Y" and recyclable = "Y"

1. **SQL113**

SELECT dept, number, LEFT(title, 12) AS short\_title FROM Class;

1. **SQL111**

SELECT \* FROM Class WHERE LOWER(title) LIKE 'introduction%';

1. **SQL106**

select fname from Instructor where username = "zahorjan"

1. **SQL 105**

SELECT I.username, T.dept, T.number, C.title

FROM Instructor I

JOIN Teaches T ON I.username = T.username

JOIN Class C ON T.dept = C.dept AND T.number = C.number

ORDER BY I.lname ASC

LIMIT 2;

1. **SQL 107**

SELECT \* FROM Class WHERE dept = 'CSE' AND number BETWEEN 400 AND 499;

1. **SQL 95**

SELECT name, population, area

FROM World

WHERE area >= 3000000 OR population >= 25000000;

1. **SQL99**

select name from Customer where referee\_id != 2 or referee\_id IS NULL;

1. **SQL96** (SQL Server)

SELECT U.unique\_id, E.name

FROM Employees E

LEFT JOIN EmployeeUNI U ON E.id = U.id;

1. **SQL102**

SELECT W1.id

FROM Weather W1

JOIN Weather W2 ON W1.recordDate = DATE\_ADD(W2.recordDate, INTERVAL 1 DAY)

WHERE W1.temperature > W2.temperature;

1. **SQL94**

SELECT DISTINCT L1.num AS ConsecutiveNums

FROM Logs L1

JOIN Logs L2 ON L1.id = L2.id - 1

JOIN Logs L3 ON L1.id = L3.id - 2

WHERE L1.num = L2.num AND L2.num = L3.num;

1. **SQL112**

SELECT title, LENGTH(title) AS length FROM Class;

1. **SQL104**

//

1. **SQL 108**

select \* from Teaches where username = "levy" or username = "djw"

1. **SQL 114**

select \* from Instructor where started\_on < "1990-01-01"

order by username desc

1. **SQL125**

UPDATE SinhVien

SET TrangThai = CASE

WHEN DiemTB >= 5.0 THEN 'Đạt'

ELSE 'Không đạt'

END;

1. **SQL109**

//

1. **SQL120**

CREATE PROCEDURE GetEmployeeById(IN employeeId INT)

BEGIN

    SELECT \* FROM Employees WHERE id = employeeId;

END

1. **SQL98**

DELETE FROM Person

WHERE id NOT IN (

    SELECT MIN(id) FROM Person GROUP BY email

);

1. **SQL116**

SELECT \* FROM Instructor WHERE started\_on >= '2004-01-01';

1. SQL119

//

1. **SQL118**

CREATE TABLE Enrollments (

    enrollment\_id INT PRIMARY KEY,

    student\_id INT,

    course\_id INT,

    enrollment\_date DATE,

    FOREIGN KEY (student\_id) REFERENCES Students(student\_id),

    FOREIGN KEY (course\_id) REFERENCES Courses(course\_id)

);