



LẬP TRÌNH CSDL NÂNG CAO

ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn



MỤC TIÊU

● Cơ sở dữ liệu

- ✱ Sử dụng các toán tử đặc biệt
- ✱ Truy vấn kết nối dữ liệu từ nhiều bảng
- ✱ Truy vấn tổng hợp và thống kê dữ liệu
- ✱ Viết được Stored Procedure

● Lập trình JDBC

- ✱ Sử dụng PreparedStatement
- ✱ Sử dụng CallableStatement
- ✱ Điều khiển transaction
- ✱ JDBC Components



PHẦN 1: CƠ SỞ DỮ LIỆU



CÂU LỆNH TRUY VẤN

```
SELECT <danh sách cột>  
      FROM <bảng>  
          JOIN <bảng 1> ON <ĐK JOIN>  
          JOIN <bảng 2> ON <ĐK JOIN>...  
WHERE <ĐK lọc dữ liệu>  
GROUP BY <biểu thức nhóm>  
        HAVING <ĐK lọc nhóm>  
ORDER BY <biểu thức sắp xếp> [ASC] | DESC
```



VÍ DỤ

```
SELECT * FROM Products
  WHERE UnitPrice BETWEEN 5 AND 15
 ORDER BY UnitPrice DESC
```

```
SELECT * FROM Products
  WHERE year(ProductDate) IN (1996, 2000)
```

```
SELECT * FROM Products
  WHERE UnitPrice < 5 OR Quantity > 100
```

```
SELECT * FROM Products
  WHERE Name LIKE '%on%'
```



TOÁN TỬ ĐẶC BIỆT

- [NOT] LIKE
 - ✱ Tương tự
- [NOT] IN
 - ✱ Tập hợp
- [NOT] BETWEEN
 - ✱ Trong khoảng
- IS [NOT] NULL
 - ✱ Null



TOÁN TỬ LIKE

● [NOT] LIKE

- ✱ %: đại diện một nhóm ký tự bất kỳ
- ✱ _: đại diện một ký tự bất kỳ
- ✱ [xyz]: một ký tự thuộc nhóm x, y hoặc z
- ✱ [^xyz]: một ký tự không thuộc nhóm x, y và z

● Ví dụ

- ✱ Name **LIKE** N'Nguyễn%
- ✱ Name **LIKE** '%anh%'
- ✱ Name **LIKE** '%Tuấn'
- ✱ Name **LIKE** 'Pha[nm]%'
- ✱ Name **LIKE** '%AB_C%'



TOÁN TỬ ĐẶC BIỆT

● [NOT] IN

✱ SELECT* FROM Products WHERE

- Id **IN** (1005, 1007, 1018, 1025)
- CategoryId **IN** (SELECT Id FROM Categories)

● [NOT] BETWEEN

✱ SELECT* FROM Products WHERE

- ProductDate **BETWEEN** '12/31/2016' AND '12/31/2017'
- UnitPrice NOT **BETWEEN** 20 **AND** 80

● IS [NOT] NULL

✱ SELECT * FROM Products WHERE

- ProductDate **IS NULL**



HÀM XỬ LÝ THỜI GIAN

● Truy vấn theo thời gian

✱ **SELECT*** FROM Products WHERE

- ProductDate < **GetDate**()
- **Year**(ProductDate) = 2014
- **DatePart**(Quarter, ProductDate) = 3

● Hàm xử lý thời gian thường dùng

- ✱ **GetDate**() lấy thời gian hiện tại
- ✱ **DatePart**(<thành phần>, <thời gian>)
- ✱ **Year**(<thời gian>): lấy năm
- ✱ **Month**(<thời gian>): lấy tháng
- ✱ **Hour**(<thời gian>): lấy giờ
- ✱ **Minute**(<thời gian>): lấy phút
- ✱ **Second**(<thời gian>): lấy giây



KẾT NỐI NHIỀU BẢNG

```
SELECT c.Name Loai, p.Name SanPham, UnitPrice DonGia
FROM Products p
JOIN Categories c ON c.Id=p.CategoryId
ORDER BY c.Name
```

Loai	SanPham	DonGia
Điện thoại di động	Queso Cabrales	21
Điện thoại di động	Queso Manchego La Pastora	38
Điện thoại di động	Gorgonzola Telino	12.5
Điện thoại di động	Mascarpone Fabioli	32
Điện thoại di động	Geitost	2.5
Điện thoại di động	Raclette Courdavault	55
Điện thoại di động	Camembert Pierrot	34
Điện thoại di động	Gudbrandsdalsost	36
Điện thoại di động	Flotemysost	21.5
Điện thoại di động	Mozzarella di Giovanni	34.8
Đồ trang sức	Mishi Kobe Niku	97
Đồ trang sức	Alice Mutton	39
Đồ trang sức	Tharinger Rostbratwurst	123.79
Đồ trang sức	Perth Pasties	32.8
Đồ trang sức	Tourtiare	7.45
Đồ trang sức	Pacta chinois	24
Đồ trang sức	Mishi Kobe Niku	97
Đồng hồ	Aniseed Syrup	190



PHÂN LOẠI KẾT NỐI

INNER JOIN

```
SELECT * FROM Categories AS c  
INNER JOIN Products AS p  
ON c.Id = p.CategoryId
```

LEFT OUTER JOIN

```
SELECT * FROM Categories AS c  
LEFT OUTER JOIN Products AS p  
ON c.Id = p.CategoryId
```

RIGHT OUTER JOIN

```
SELECT * FROM Categories AS c  
RIGHT OUTER JOIN Products AS p  
ON c.Id = p.CategoryId
```

FULL OUTER JOIN

```
SELECT * FROM Categories AS c  
FULL OUTER JOIN Products AS p  
ON c.Id = p.CategoryId
```



INNER JOIN

```
SELECT
    c.Name AS Loại,
    p.Name AS [Hàng Hóa],
    p.UnitPrice AS [Đơn giá]
FROM Categories AS c
    INNER JOIN Products AS p ON c.Id = p.CategoryId
```

	Loại	Hàng Hóa	Đơn giá
1	Laptop	Dell 1014	2000
2	Laptop	Acer Xyz	700
3	Mobile	iPhone ...	1500



LEFT OUTER JOIN

```
SELECT
    c.Name AS Loại,
    p.Name AS [Hàng Hóa],
    p.UnitPrice AS [Đơn giá]
FROM Categories AS c
    LEFT OUTER JOIN Products AS p ON c.Id = p.CategoryId
```

	Loại	Hàng Hóa	Đơn giá
1	Fashion	NULL	NULL
2	Laptop	Dell 1014	2000
3	Laptop	Acer Xyz	700
4	Mobile	iPhone 5S	1500



THỐNG KÊ – GROUP BY

SELECT

```
CategoryId AS 'Loại',  
sum(UnitPrice * Quantity) AS 'Giá trị',  
count(*) AS 'Số lượng',  
min(UnitPrice) AS 'Giá TN',  
max(UnitPrice) AS 'Giá CN',  
avg(UnitPrice) AS 'Giá TB'
```

FROM Products

GROUP BY CategoryId

Results Messages

Loại	Giá trị	Số lượng	Giá TN	Giá CN	Giá TB
BAG	7601.05	12	6	62.5	20.6825
CAM	14004.83	14	9.2	81	24.8878571428571
FAS	6395.75	5	10	53	32.37
LAP	6996.8	11	10	43.9	23.2181818181818
LEW	37242.92	7	7.45	123.79	60.1485714285714
MOB	10175.7	10	2.5	55	28.73
PER	3678.75	7	7	38	20.25
WAT	109899...	13	4.5	263.5	49.75

● GROUP BY

✱ SUM()

✱ AVG()

✱ COUNT()

✱ MIN()

✱ MAX()



THỐNG KÊ – HAVING

[-] SELECT

```
CategoryId AS 'Loại',  
sum(UnitPrice * Quantity) AS 'Giá trị',  
count(*) AS 'Số lượng',  
min(UnitPrice) AS 'Giá TN',  
max(UnitPrice) AS 'Giá CN',  
avg(UnitPrice) AS 'Giá TB'
```

FROM Products

GROUP BY CategoryId

HAVING count(*) > 10

● GROUP BY

● **HAVING**

✱ SUM()

✱ AVG()

✱ COUNT()

✱ MIN()

✱ MAX()

Results Messages

Loại	Giá trị	Số lượng	Giá TN	Giá CN	Giá TB
BAG	7601.05	12	6	62.5	20.6825
CAM	14004.83	14	9.2	81	24.8878571428571
LAP	6996.8	11	10	43.9	23.2181818181818
WAT	109899.75	13	4.5	263.5	49.75



THỐNG KÊ

```
SELECT
    c.Name 'Loại',
    SUM(p.UnitPrice * p.Quantity) AS 'Giá trị',
    MIN(p.UnitPrice) 'Giá thấp nhất',
    AVG(p.UnitPrice) 'Giá trung bình'
FROM Products p
    JOIN Categories c ON c.Id=p.CategoryId
WHERE YEAR(ProductDate) < 2000
GROUP BY c.Name
    HAVING SUM(p.UnitPrice * p.Quantity) > 1000
```

Results

Messages

Loại	Giá trị	Giá thấp nhất	Giá trung bình
Điện thoại di động	3519.7	21	28.825
Đồ trang sức	18818	97	97
Đồng hồ	108940.75	7.75	72.65625
Máy ảnh	1643.5	9.5	16.216666666666667
Máy tính xách tay	3580.05	10	20.7071428571429
Túi xách du lịch	2870.55	9.5	17.56875



STORED PROCEDURE

- PROC là một thủ tục được tạo ngay trong lòng CSDL nhằm thực hiện một nhiệm vụ cụ thể nào đó.
- Tạo Stored Procedure

```
CREATE PROC TênProc([Tham số]) AS  
BEGIN  
    <Nội dung>  
END
```

- Sử dụng Stored Procedure

```
EXEC TênProc [Tham số]
```



VÍ DỤ 1 - PROC

● Tạo PROC thêm mới loại

```
CREATE PROC spInsertCategory
(
    @Id CHAR(3),
    @Name NVARCHAR(50)
) AS
BEGIN
    INSERT INTO Categories(Id, Name)
        VALUES(@Id, @Name)
END
```

● Sử dụng PROC

```
EXEC spInsertCategory 'MOB', N'Mobile'
```



VÍ DỤ 2 - PROC

- Tạo PROC truy vấn sản phẩm theo giá

```
CREATE PROC spSearchProducts
(
    @MaxPrice FLOAT = 0,
    @MinPrice FLOAT
) AS
BEGIN
    SELECT * FROM Products
        WHERE UnitPrice BETWEEN @MaxPrice AND @MaxPrice
END
```

- Sử dụng PROC

```
EXEC spSearchProducts 5, 10
EXEC spSearchProducts 7
```



VÍ DỤ 3 - PROC

● Tạo PROC tổng hợp dữ liệu

```
CREATE PROC spInventory
AS
BEGIN
    SELECT
        c.Name AS 'Loại',
        sum(UnitPrice * Quantity) AS 'Giá trị',
        count(*) AS 'Số lượng',
        min(UnitPrice) AS 'Giá TN',
        max(UnitPrice) AS 'Giá CN',
        avg(UnitPrice) AS 'Giá TB'
    FROM Products p
        JOIN Categories c ON c.Id = p.CategoryId
    GROUP BY c.Name
        HAVING count(*) > 10
END
```



LỢI ÍCH CỦA PROC

- Thực hiện nhanh hơn
- Dễ quản lý, nâng cấp
- Tái sử dụng
- Sử dụng gọn hơn trong ngôn ngữ lập trình

PHẦN 2: LẬP TRÌNH JDBC



PREPAREDSTATEMENT – THAO TÁC

```
String sql = "INSERT INTO Categories(Id, Name) VALUES(?, ?)";  
Class.forName(driver);  
Connection connection = DriverManager.getConnection(url, user, password);  
PreparedStatement statement = connection.prepareStatement(sql);  
statement.setString(1, "301");  
statement.setString(2, "Cây cảnh");  
statement.executeUpdate();  
connection.close();
```

- ? đầu tiên có vị trí là 1
- Statement.setXYZ(index, value)



PREPAREDSTATEMENT – TRUY VẤN

```
String sql = "SELECT * FROM Products WHERE UnitPrice BETWEEN ? AND ?";  
Class.forName(driver);  
Connection connection = DriverManager.getConnection(url, user, password);  
PreparedStatement statement = connection.prepareStatement(sql);  
statement.setDouble(1, 5.0);  
statement.setDouble(2, 15.0);  
ResultSet resultSet = statement.executeQuery();  
while(resultSet.next()) {  
    String name = resultSet.getString("name");  
    System.out.println(name);  
}  
connection.close();
```




ƯU ĐIỂM CỦA PREPARED STATEMENT

- Mã viết rõ ràng
- Unicode (không sử dụng N'...')
- Tránh lỗi dấu nháy đơn
- Nhanh hơn khi thực hiện câu lệnh nhiều lần
- Làm việc được với dữ liệu nhị phân
- Tránh được hack bằng SQL injection



CALLABLESTATEMENT – THAO TÁC

```
String sql = "{call spInsertCategories(?, ?)}";  
Class.forName(driver);  
Connection connection = DriverManager.getConnection(url, user, password);  
CallableStatement statement = connection.prepareCall(sql);  
statement.setString(1, "301");  
statement.setString(2, "Cây cảnh");  
statement.executeUpdate();  
connection.close();
```



CALLABLESTATEMENT – TRUY VẤN

```
String sql = "{call spSearchProducts(?, ?)}";  
Class.forName(driver);  
Connection connection = DriverManager.getConnection(url, user, password);  
CallableStatement statement = connection.prepareCall(sql);  
statement.setDouble(1, 5.0);  
statement.setDouble(2, 15.0);  
ResultSet resultSet = statement.executeQuery();  
while(resultSet.next()) {  
    String name = resultSet.getString("name");  
    System.out.println(name);  
}  
connection.close();
```



TRANSACTION

```
try {  
    // hủy chế độ auto-commit  
    conn.setAutoCommit(false);  
    <thực thi khối lệnh thao tác dữ liệu>  
    conn.commit();  
}  
catch(SQLException ex) {  
    // hủy bỏ các câu lệnh đã thực hiện  
    conn.rollback();  
}  
// đặt lại chế độ auto-commit  
con.setAutoCommit(true);
```

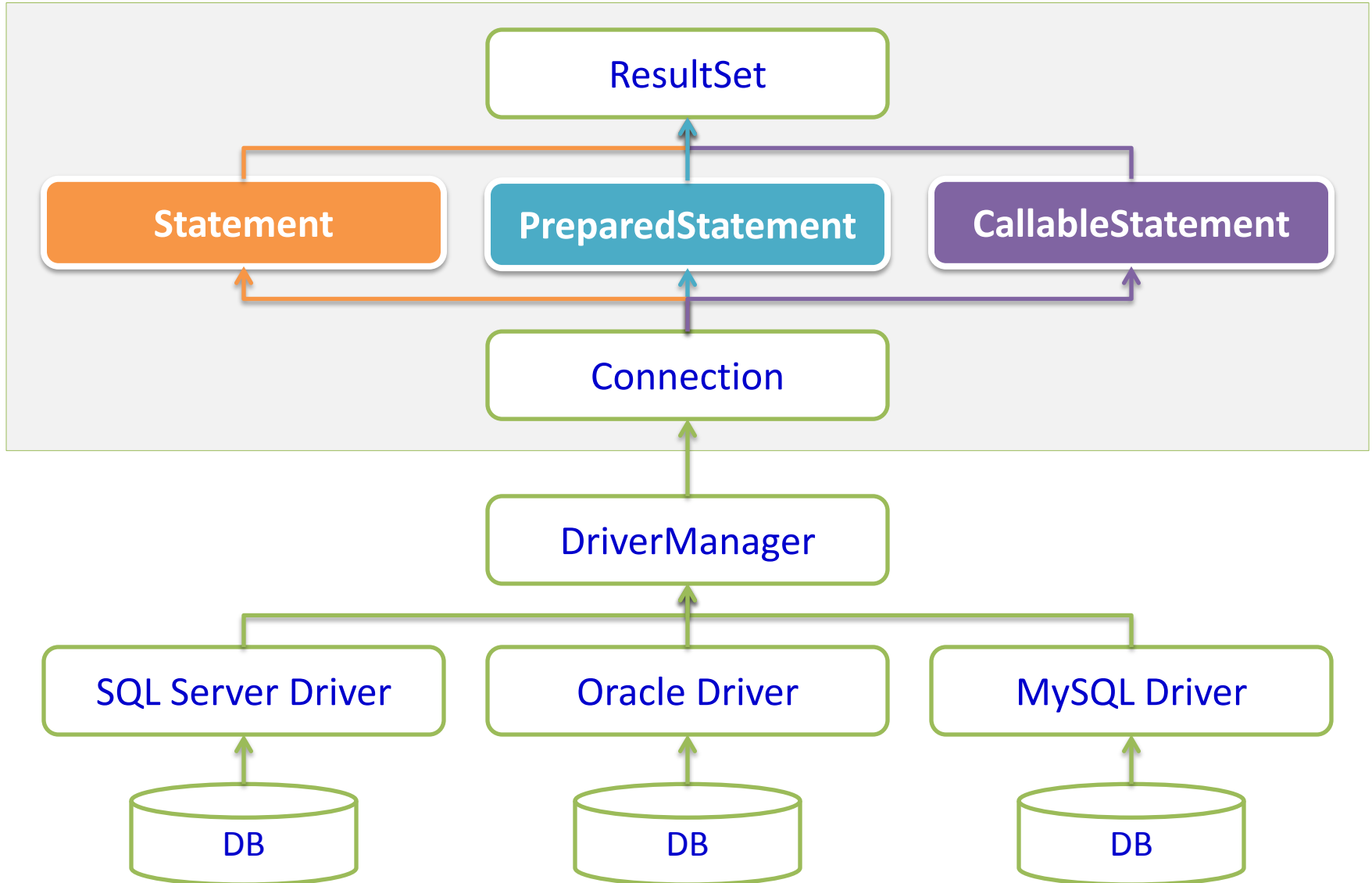


TRANSACTION

```
connection.setAutoCommit(false);
try {
    String sql = "INSERT INTO Categories(Id, Name) VALUES(?, ?)";
    PreparedStatement pstmt = connection.prepareStatement(sql);
    pstmt.setString(1, "MOB");
    pstmt.setString(2, "Mobile");
    pstmt.executeUpdate();
    pstmt.clearParameters();
    pstmt.setString(1, "MOB");
    pstmt.setString(2, "Điện thoại di động");
    pstmt.executeUpdate();
    connection.commit();
}
catch (Exception e) {
    connection.rollback();
}
```



JDBC COMPONENTS





● SQL

- ✱ Sử dụng các toán tử và hàm
- ✱ Kết nối dữ liệu từ nhiều bảng
- ✱ Tổng hợp và thống kê dữ liệu
- ✱ Viết được Stored Procedure

● JDBC

- ✱ Xác định rõ các thành phần JDBC
- ✱ Sử dụng PreparedStatement
- ✱ Sử dụng CallableStatement
- ✱ Điều khiển Transaction