



SQL Programming

Lecturer: Neena Thota

`neena.thota@it.uu.se`

Intended Learning Outcomes

Understand

- Rule based actions – Triggers and Stored Procedures
- Embedded SQL, APIs, and DB Programming languages
- Examples with JDBC

ACTIONS USING RULES

Actions using Rules

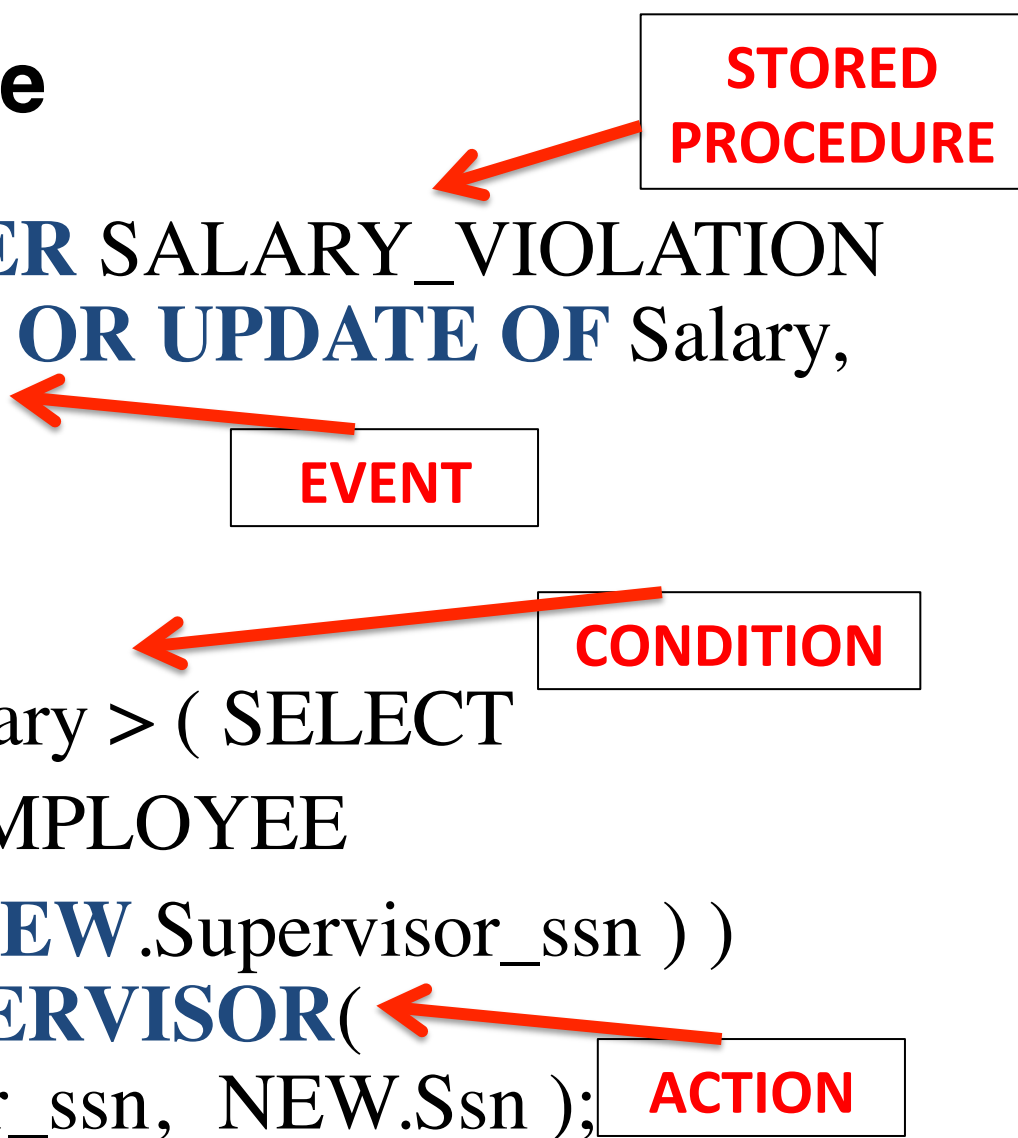
- In relational database systems, possible to enforce real world rules.
- **Trigger:**
 - used to specify automatic actions that database system will perform when certain events and conditions occur.
- **Stored procedures:**
 - part of overall database definition; invoked appropriately when certain conditions are met.

Triggers

- Used in various applications, such as maintaining database consistency, monitoring database updates, and updating derived data automatically.
- **Example:** Check whenever an employee's salary is greater than salary of his or her direct supervisor
 - **Event:** inserting a new employee record, changing an employee's salary, or changing an employee's supervisor.
 - **Condition:**
 - if specified, evaluate first and if true, do action.
 - if none, execute action after event.
 - After event, optional condition may be evaluated.
 - **Action:** call an external stored procedure which will notify the supervisor.

Trigger: Example

```
CREATE TRIGGER SALARY_VIOLATION
BEFORE INSERT OR UPDATE OF Salary,
Supervisor_ssn
ON EMPLOYEE
FOR EACH ROW
WHEN ( NEW.Salary > ( SELECT
Salary FROM EMPLOYEE
WHERE ssn = NEW.Supervisor_ssn ) )
INFORM_SUPERVISOR(
NEW.Supervisor_ssn, NEW.Ssn );
```



The diagram illustrates the components of the SQL trigger code:

- STORED PROCEDURE**: Points to the `CREATE TRIGGER` statement.
- EVENT**: Points to the `BEFORE INSERT OR UPDATE OF` clause.
- CONDITION**: Points to the `WHEN (NEW.Salary > (SELECT ...))` clause.
- ACTION**: Points to the `INFORM_SUPERVISOR(...)` clause.

Stored Procedures

- Database program modules stored and executed by DBMS **at database server**.
- Useful:
 - Stored at server and invoked by any application programs. This **reduces duplication of effort and improves software modularity**;
 - Reduces **data transfer and communication cost** between client and server in certain situations;
 - Can enhance modeling power provided by *views* by allowing more complex types of derived data to be made available to database users;
 - Can be used to check for **complex constraints** that are beyond the specification power of *assertions* and *triggers*;

Stored Procedures: Example

- To create a stored procedure

CREATE PROCEDURE <procedure name>
(<parameters>)  **SQL data types; Mode –IN, OUT, INOUT**

LANGUAGE <programming language name>
EXTERNAL NAME <file path name>;

- To invoke a stored procedure

CALL <procedure or function name>
(<argument list>);

Example: Stored Procedure

- *GetDeptEmployees* - stored procedure that selects employees in a department.
- The *deptNumber* is IN parameter of stored procedure.
- Pass a value (1) to stored procedure to select all employees specified by *deptNumber* parameter

```

1  DELIMITER //
2  • CREATE PROCEDURE GetDeptEmployees (IN deptNumber INT)
3    BEGIN
4      SELECT *
5      FROM Employee
6      WHERE DEPARTMENT = deptNumber;
7    END //
8  DELIMITER ;
9
10 • CALL GetDeptEmployees(1);|
  
```

	EmpID	SSN	Name	Surname	Salary	Department	Supervisor
►	1	6808029376	John	McEnroe	53000	1	NULL
	5	NULL	Rafael	Codardus	43000	1	1
	6	6803036078	Obama	Virilus	34000	1	1
	7	7808178347	Karl	Gloriosus	43500	1	1
	8	6809099948	Carl	Macho	33000	1	1



SQL PROGRAMMING

SQL Programming Techniques

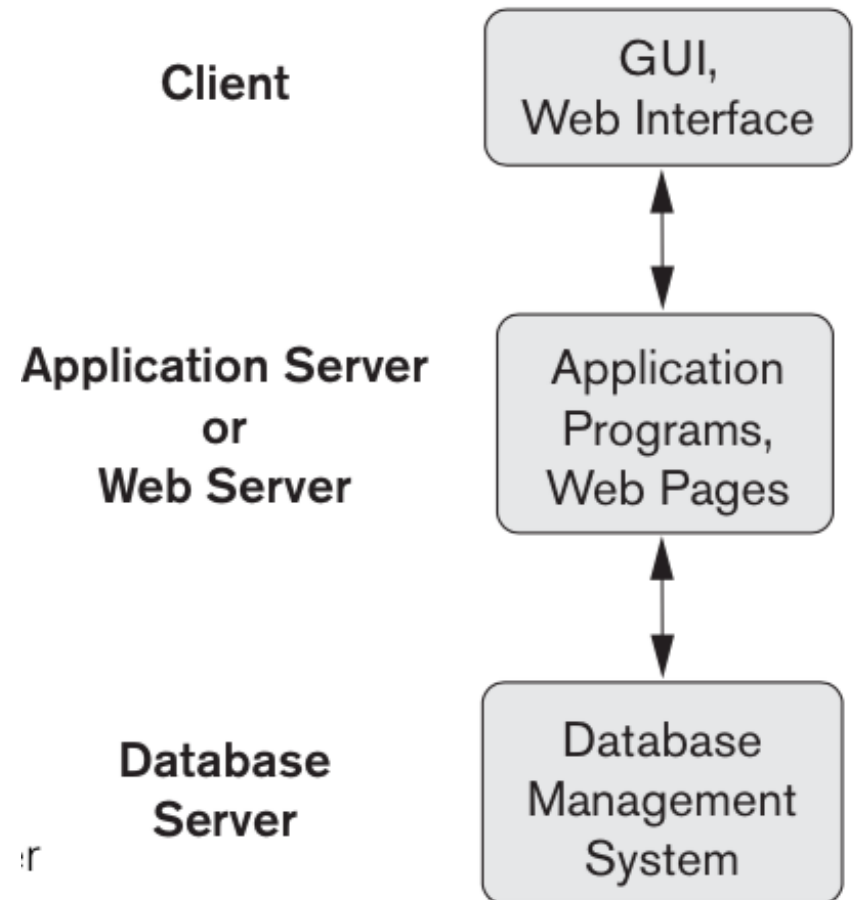
- **Database applications**
 - Host language
 - Java, C/C++/C#, COBOL etc.
 - Scripting languages - PHP and JavaScript
- **SQL standards**
 - Continually evolving
 - Each DBMS vendor may have some variations from standard

Database Programming: Techniques

- **Interactive interface**
 - SQL commands typed directly into a monitor
- **Execute file of commands**
 - *@<filename>*
- **Application programs or database applications**
 - Used as canned transactions by the end users access a database (e.g. banking, airlines)
 - May have **Web interface** (online purchases)

In 3-Tier Client/Server Architectures for DBMS

1. User interface programs and application programs run on the client side.
2. Program establishes connection to DBMS: **Open Database Connectivity (ODBC)** provides **API**.
3. Client program connects, send queries and transaction requests.
4. Query results sent back to client program, which can process and display results as needed.



Approaches to Database Programming

Embedded SQL

- Query text checked for syntax errors and validated against database schema at compile time
- For complex applications where queries have to be generated at runtime.

Library of functions

API (application programming interface)

- More flexibility
- More complex programming
- No checking of syntax done at compile time

Database programming language

Design brand new language e.g. Oracle's PL/SQL

- Does not suffer from impedance mismatch problem.
- Programmers must learn a new language.

Example of Embedded Code in C language

- Database statements **embedded** into host programming language
- Identified by **EXEC SQL**, which precedes all SQL commands in host language program.

```
loop = 1 ;
while (loop) {
    prompt("Enter a Social Security Number: ", ssn) ;
    EXEC SQL
        select Fname, Minit, Lname, Address, Salary
        into :fname, :minit, :lname, :address, :salary
        from EMPLOYEE where Ssn = :ssn ;
    if (SQLCODE == 0) printf(fname, minit, lname, address, salary)
        else printf("Social Security Number does not exist: ", ssn) ;
    prompt("More Social Security Numbers (enter 1 for Yes, 0 for No): ", loop) ;
}
```

Example of Embedded Code in Java

- **SQLJ** standard adopted by several vendors for embedding SQL in Java; **#sql**
- SQLJ translator converts SQL statements into Java, which then executed through **JDBC** interface

```
//Program Segment J1:
1) ssn = readEntry("Enter a Social Security Number: ") ;
2) try {
3)     #sql{ select fname, Minit, Lname, Address, Salary
4)         into :fname, :minit, :lname, :address, :salary
5)         from EMPLOYEE where Ssn = :ssn} ;
6) } catch (SQLException se) {
7)     System.out.println("Social Security Number does not exist: " + ssn) ;
8)     Return ;
9) }
10) System.out.println(fname + " " + minit + " " + lname + " " + address
    + " " + salary)
```


Retrieving Multiple Tuples in Java Using Iterators

- **Iterator**

- Object associated with a collection (set or multiset) of records in a query result

- **Named iterator**

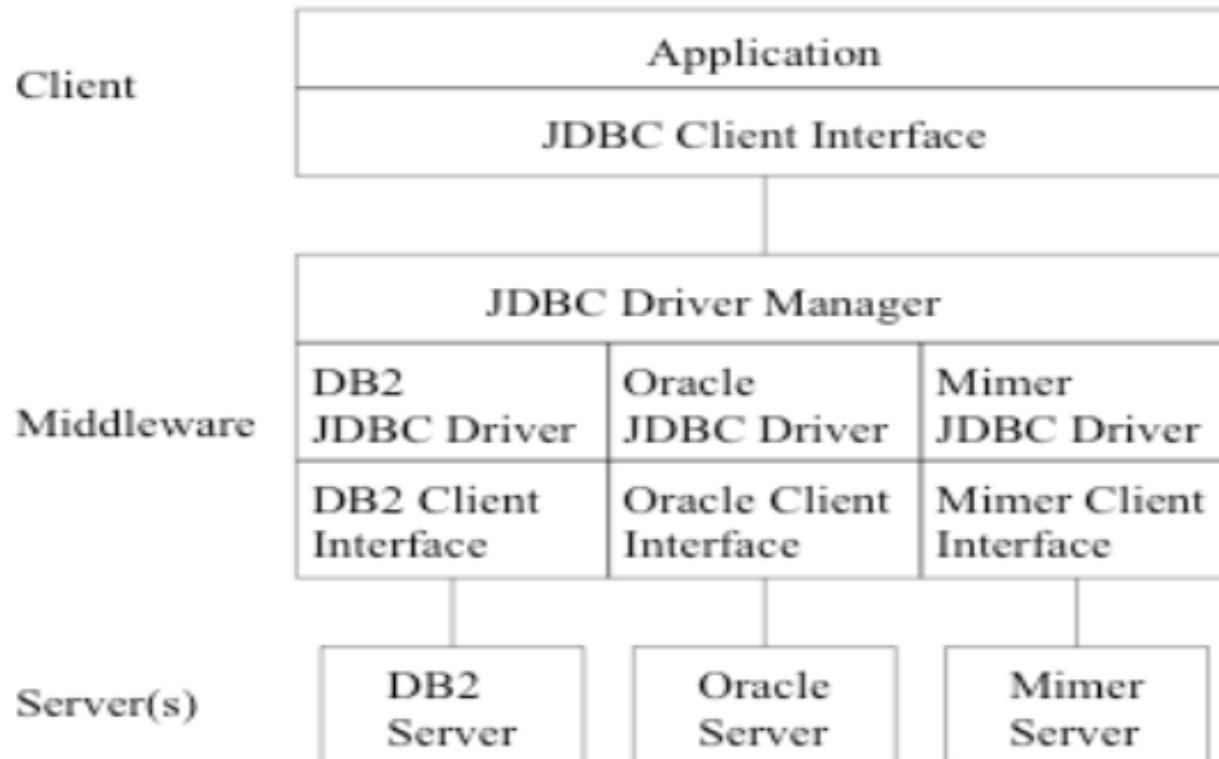
- Associated with a query result by listing attribute names and types in query result

- **Positional iterator**

- Lists only attribute types in query result

API: JDBC Architecture

- **JDBC API** is independent of (relational) DBMS and operating system.
- **Connector/J** provides driver support for connecting to **MySQL** from Java applications.



Sequence of Interaction in Database Programming

1. Open a connection to database server

1. (specify the Internet address (URL) of the machine where database server is located, plus provide login account name and password for database access).

2. Interact with database

1. by submitting queries, updates, and other SQL database commands.
2. Fetch and manage the result of the SQL statements

3. Terminate or close connection to database.

Steps in Java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
```

←
Import class
libraries

```
Connection con = DriverManager.getConnection(
    "jdbc:myDriver:myDatabase",
    username,
    password);
```

← Connect to
(registered) Data
Source

```
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");
```

← Send Queries

```
while (rs.next()) {
    int x = rs.getInt("a");
    String s = rs.getString("b");
    float f = rs.getFloat("c");
```

← Retrieve and process
results

```
if (stmt != null) {
    try {
        stmt.close();
    } catch (SQLException sqlEx) { } // ignore

    stmt = null;
```

← Release connection

JDBC: Statement classes

- Has two subclasses:
 - **PreparedStatement** and **CallableStatement**
- Question mark (?) symbol
 - Represents a statement parameter; query can be executed multiple times
 - Determined at runtime;
- JDBC functions
 - **execute / executeUpdate / executeQuery**
- **ResultSet object**
 - Holds results of query

Examples: Statements

```
String stmt1 = "select Lname, Salary from EMPLOYEE where Ssn = ?" ;
PreparedStatement p = conn.prepareStatement(stmt1) ;
ssn = readentry("Enter a Social Security Number: ") ;
p.clearParameters() ;
p.setString(1, ssn) ;
ResultSet r = p.executeQuery() ;
while (r.next()) {
    lname = r.getString(1) ;
    salary = r.getDouble(2) ;
    system.out.println(lname + salary) ;
}
```

Prepared Statement
Example

```
dno = readentry("Enter a Department Number: ") ;
String q = "select Lname, Salary from EMPLOYEE where Dno = " +
dno.toString() ;
Statement s = conn.createStatement() ;
ResultSet r = s.executeQuery(q) ;
while (r.next()) {
    lname = r.getString(1) ;
    salary = r.getDouble(2) ;
    system.out.println(lname + salary) ;
}
```

Callable Statement
Example

Examples using SQL and web pages

- Java Server Pages
- In three-tier architecture
 - DBMS is at the **bottom-tier database server**.
 - JSP runs at **middle-tier Web server**, where the program commands manipulate HTML files to create customized dynamic Web pages.
 - HTML is sent to **client tier** for display and interaction with user.

Summary of SQL programming techniques

- **Embedded SQL Approach**
 - Query text checked for syntax errors and validated against database schema at compile time
- **Library of Function Calls Approach**
 - More flexibility –
 - queries can be generated at runtime if needed.
 - More complex programming
 - program variables that match the columns in the query result may not be known in advance; Checking and query validation has to be done at runtime
- **Database Programming Language Approach**
 - Does not suffer from the impedance mismatch problem
 - Programmers must learn a new language
 - some database programming languages are vendor-specific