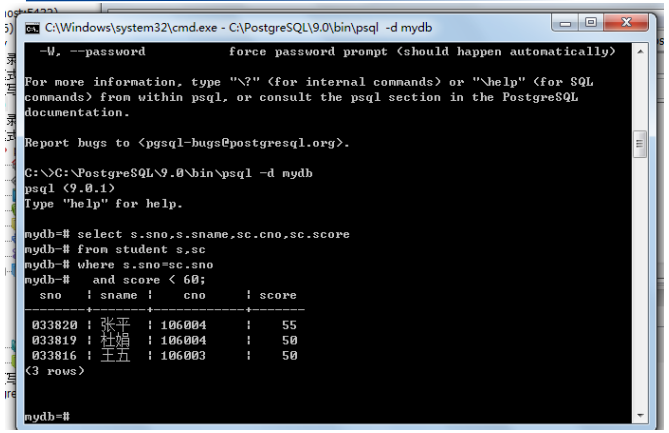
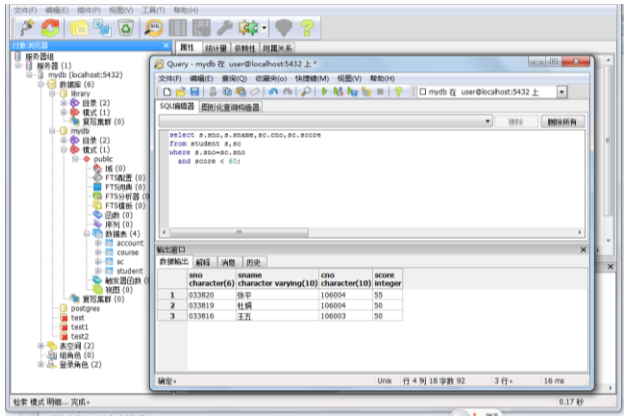
SQL应用

讲解人：陆伟 教授

数据库应用存在的问题思考





SQL的特点?

SQL

Lookup for all students which have failure records for any course.

```
Select sNo
From sc
Where score < 60
```

For each student which score < 60
    send a message to him about makeup
print name list for all students need makeup

如何表达用户业务?

数据库应用存在的问题

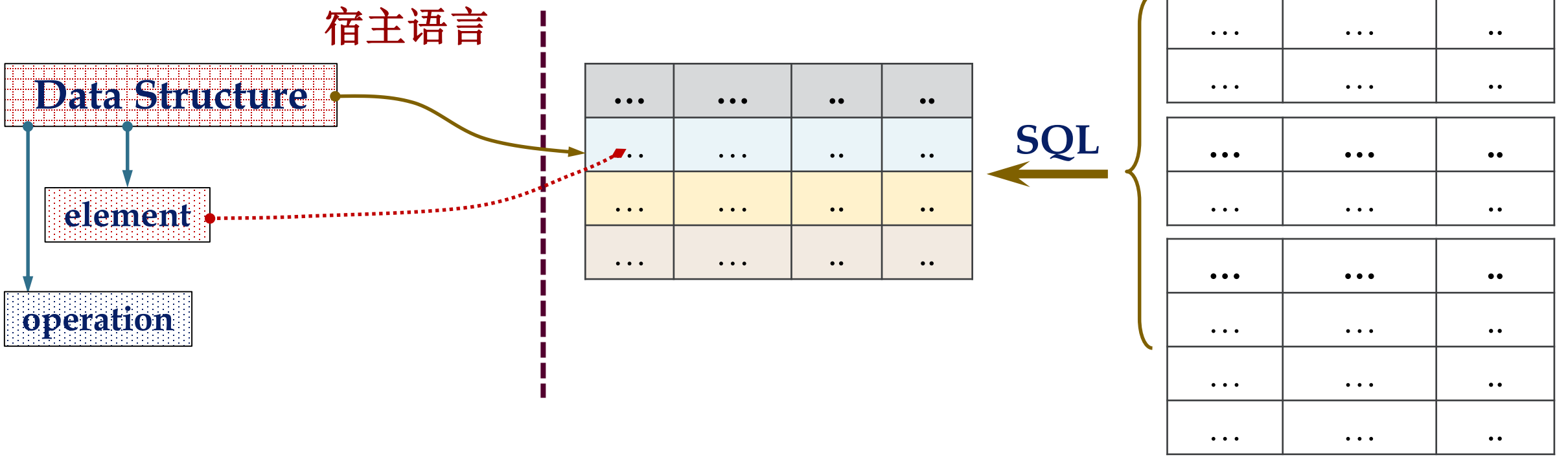
- SQL is a direct query language and it is very suit for data depositing and retrieving; as such, it has limitations.
  - Complex computational processing of the data.
  - Specialized user interfaces.
  - Access to more than one database at a time.

- 数据库应用中如何弥补SQL的不足，进而解决业务表达问题?
  - **嵌入式**-将SQL直接嵌入到通用程序设计语言，各司其职。
  - **应用编程接口(API)**-通过库函数(接口)调用，实现与数据库交互。
  - **设计新语言(混合)**-支持SQL和通用程序设计语言特性。

} 程序式

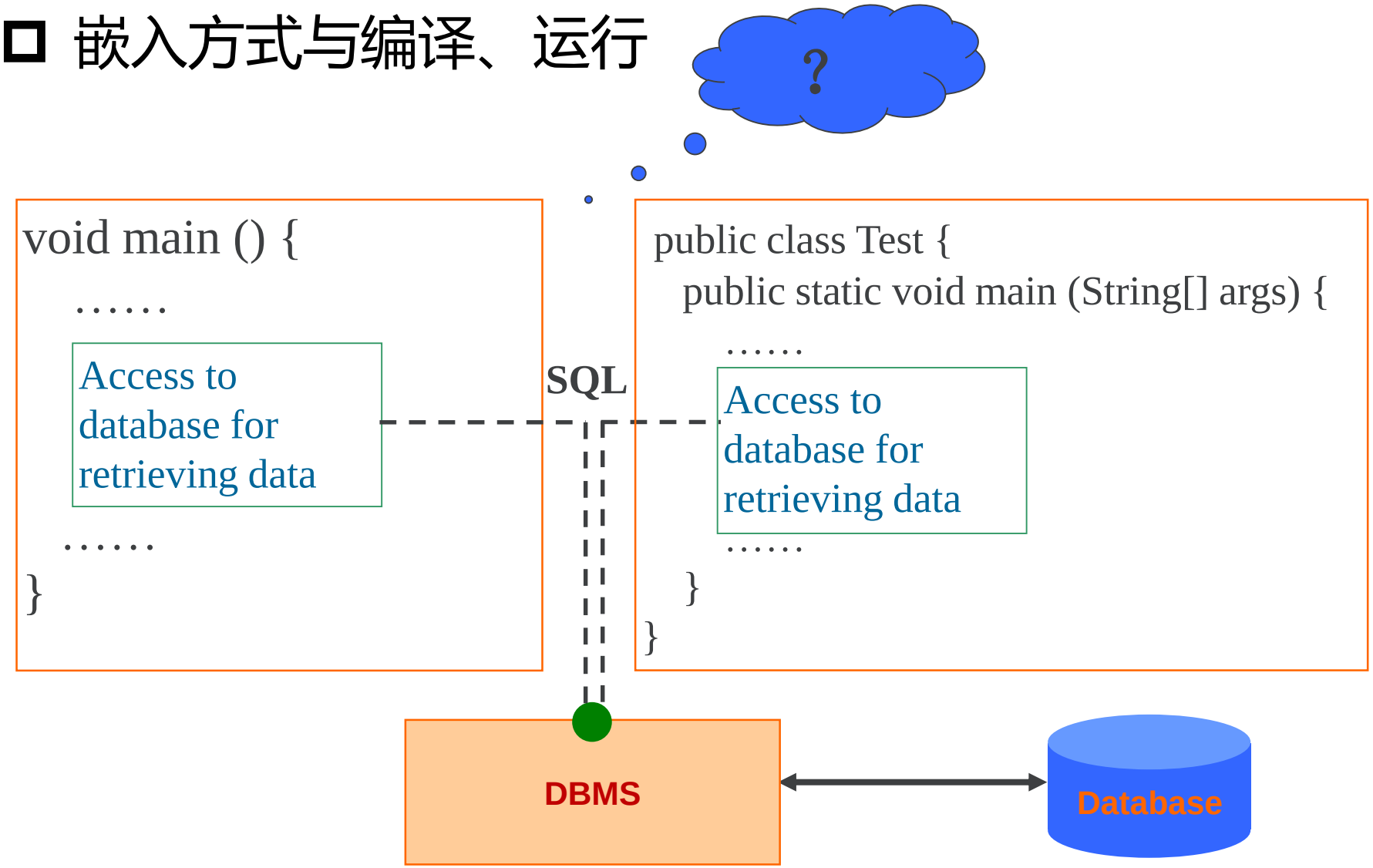
程序式SQL需要思考的问题-1

- 阻抗失配(impendance mismatch)：数据库模型与程序设计语言模型之间存在的差异而导致的不匹配问题。
  - 数据类型问题
  - 数据查询返回结果与程序设计语言数据类型匹配问题
  - 数据查询结果遍历问题



程序式SQL需要思考的问题-2

- 嵌入方式与编译、运行

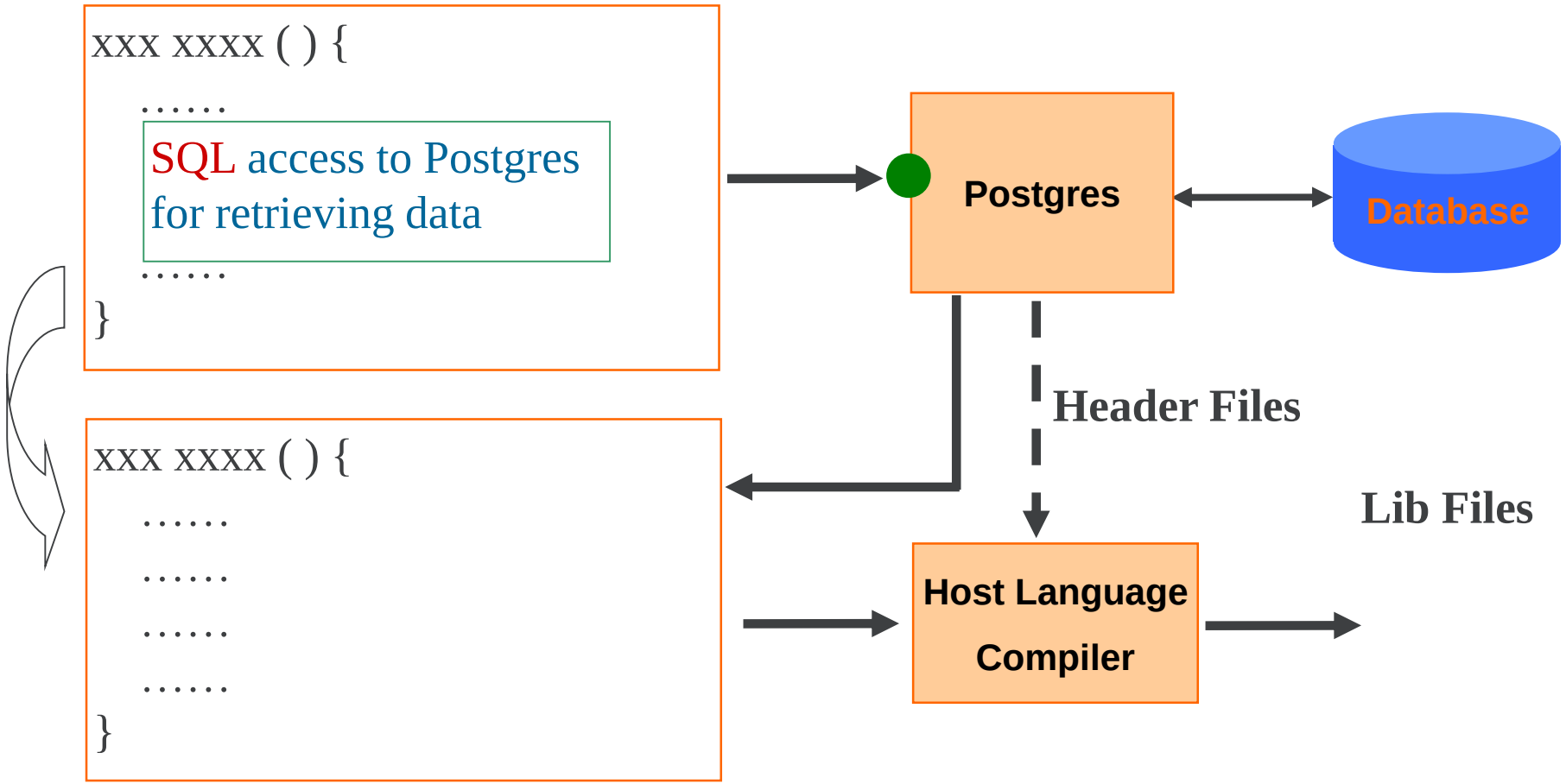


2001年，IBM高级副总裁Paul Horn在哈佛大学的报告中提出“自主计算（Autonomic Computing[1]）”概念，以应对计算复杂性危机的问题。“自主计算”也译作“自治计算”，其基本思想是参照生物领域自主神经系统的自我调节机制，以现有的理论和技术为基础构建计算系统，使得计算系统具有自我感知与管理的能力。他在报告中指出“It's time to design and build computing systems capable of running themselves, adjusting to varying circumstances, and preparing their resources to handle most effectively the workloads we put upon them. These autonomic systems must anticipate needs and allow users to concentrate on what they want to accomplish rather than figuring how to rig the computing systems to get them there.”

……

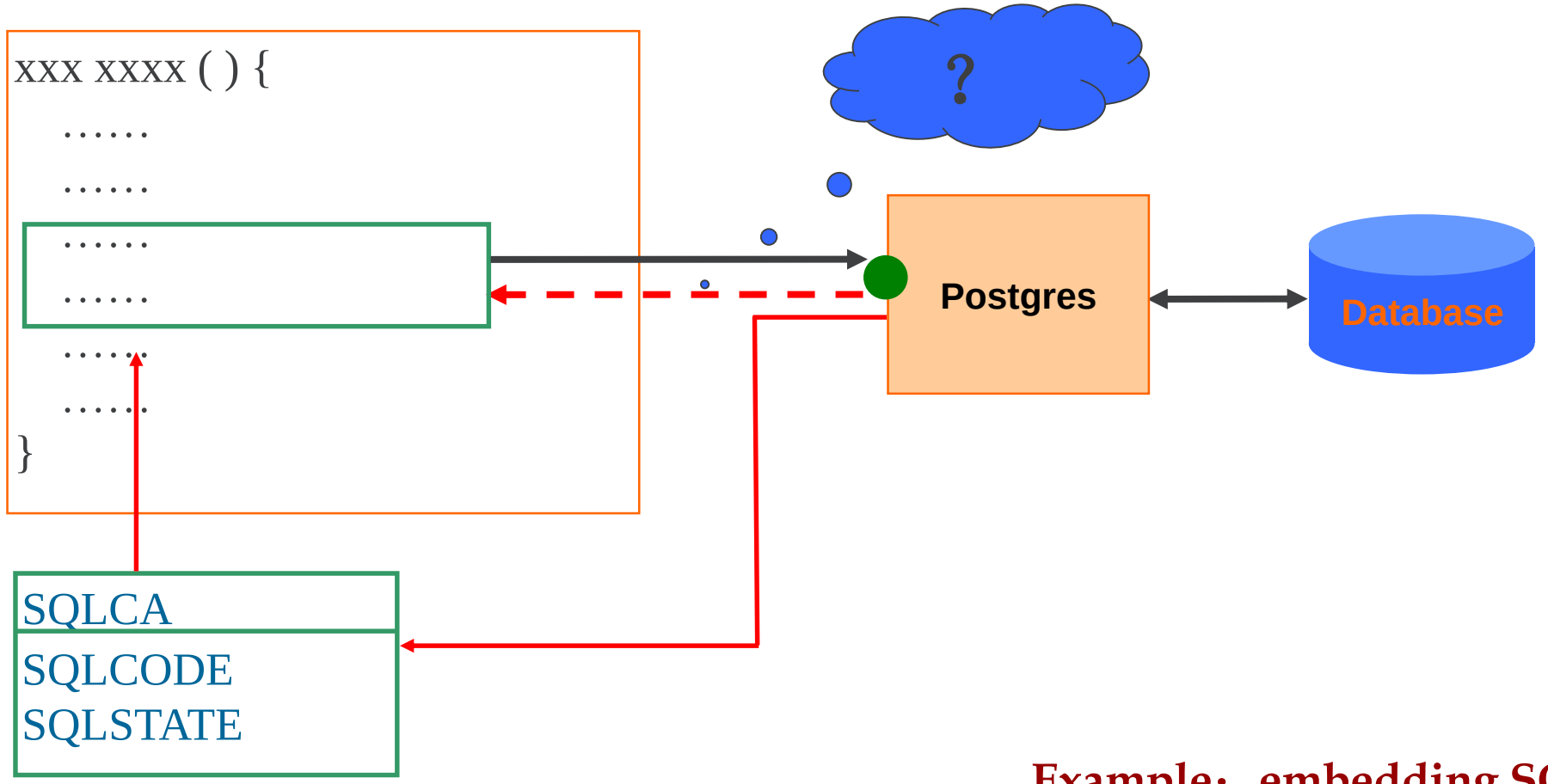
嵌入式SQL(Embedded SQL)

- SQL statements are embeded directly into the program source code and mixed with the host language statements.
- Need to be precompiled.



嵌入式SQL(Embedded SQL)

- 返回结果的处理(通信区Communication Area)



Example：embedding SQL in C program

嵌入式SQL(Embedded SQL)

- 静态嵌入SQL
  - In static embedded SQL statement, the pattern of database access is fixed and can be 'hard-coded' into the program.
  - Static SQL does not allow host variables to be used in place of table names or column names.
  - In static Embedded SQL, the follow elements must be fixed:
    - Reserved words (SELECT,UPDATE,DELETE…)
    - The number of host variables
    - The data type of each host variable
    - The database object will be accessed in SQL(table,column,view,index,…)

嵌入式SQL(Embedded SQL)

- 动态嵌入SQL
  - In many situations where the pattern of database access is not fixed and is known only at runtime.This requires more flexibility than static SQL.
  - The basic idea of dynamic SQL is to place the complete SQL statement to be executed in a host variable. The host variable is then passed to the DBMS to be executed.

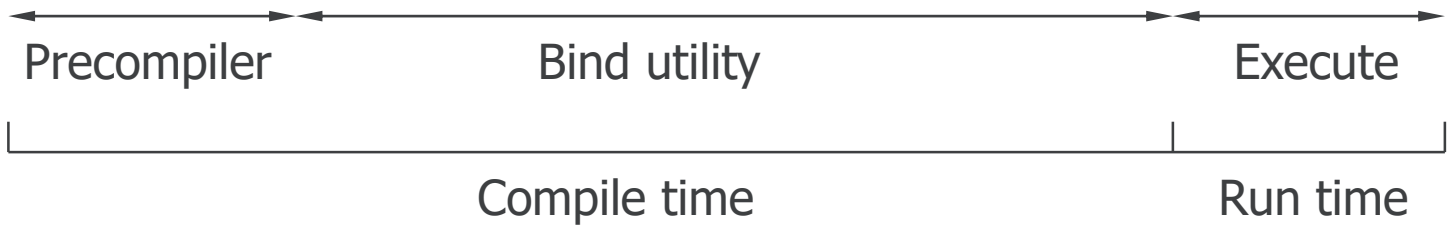
嵌入式SQL(Embedded SQL)

- 静态嵌入 VS 动态嵌入SQL

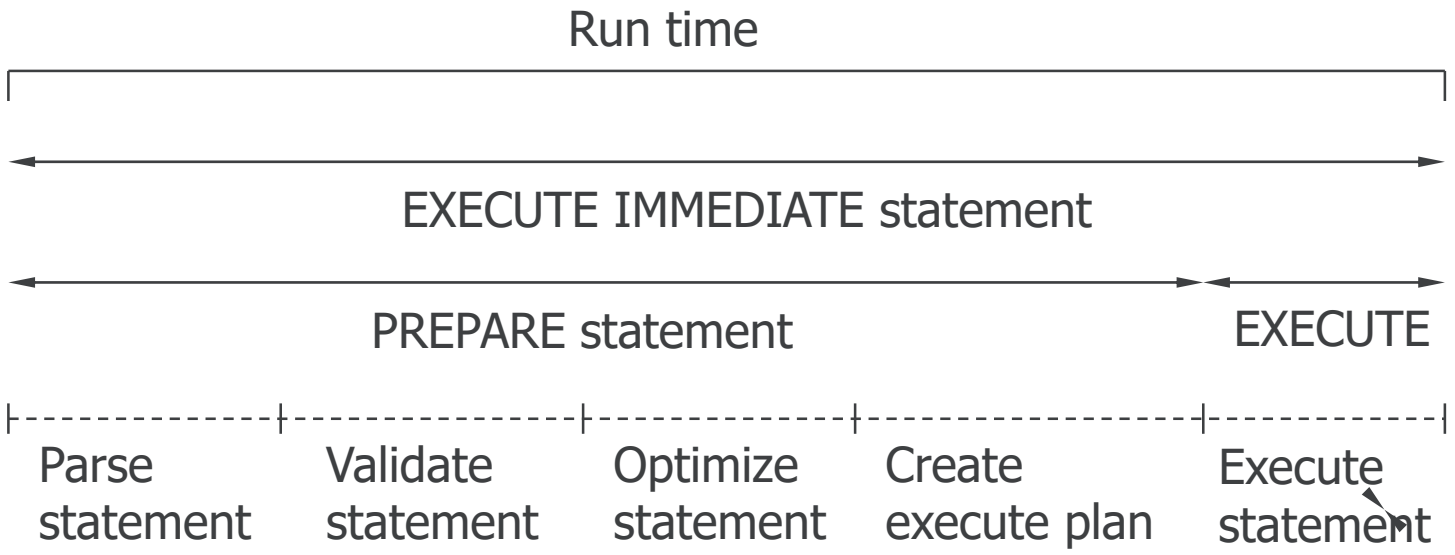
```
EXEC SQL BEGIN DECLARE SECTION;
   float increment;
EXEC SQL END DECLARE SECTION;
EXEC SQL UPDATE Staff SET
     salary=salary+:increment WHERE
     staffNo='SL21';

EXEC SQL BEGIN DECLARE SECTION;
 char buffer[100];
EXEC SQL END DECLARE SECTION;
 sprintf(buffer,"UPDATE Staff SET salary=salary+%f WHERE
    staffNo='SL21'",increment;);
EXEC SQL EXECUTE IMMEDIATE :buffer;
```

嵌入式SQL(Embedded SQL)

- 静态嵌入 VS 动态嵌入SQL



（a）static SQL



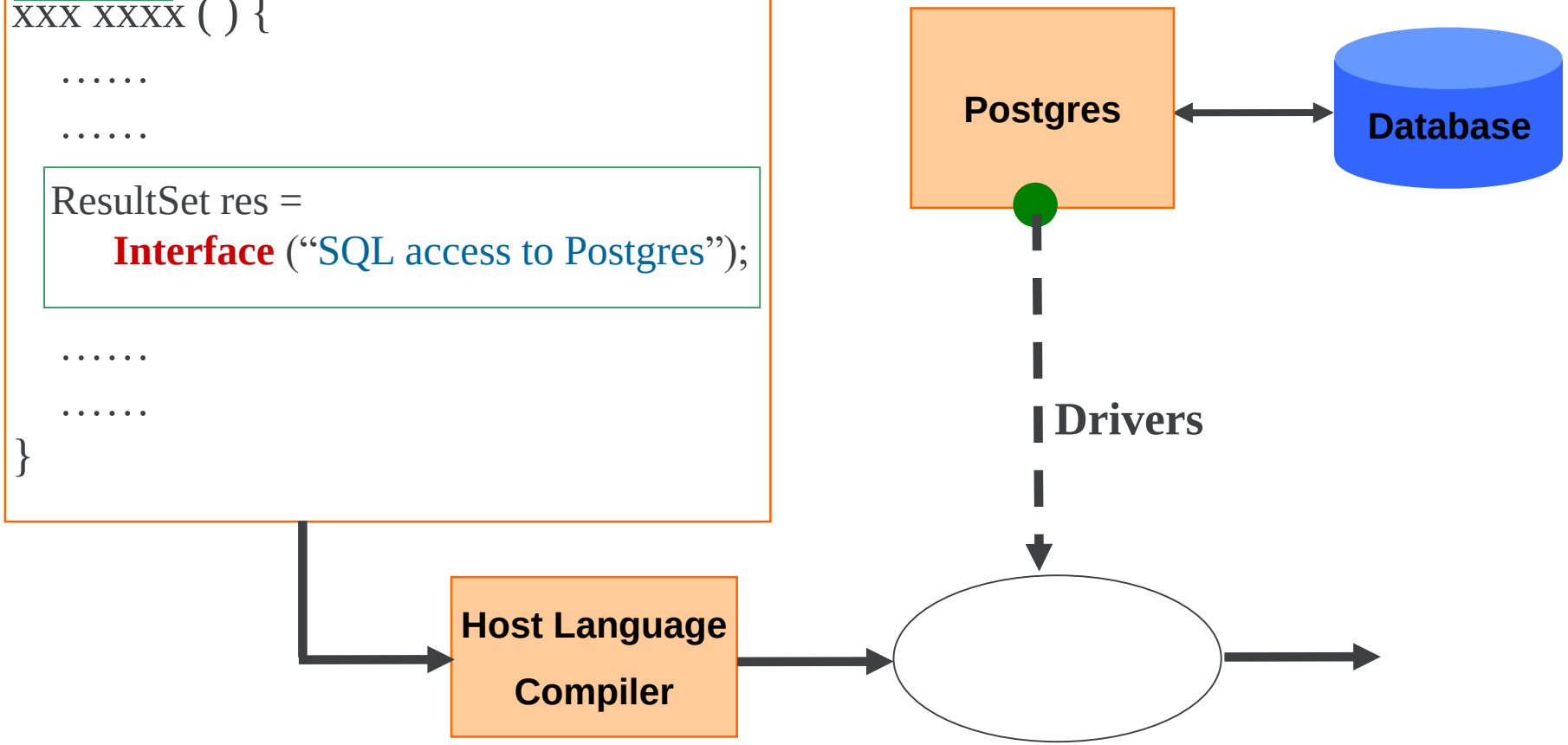
（b）Dynamic SQL

应用编程接口(API)

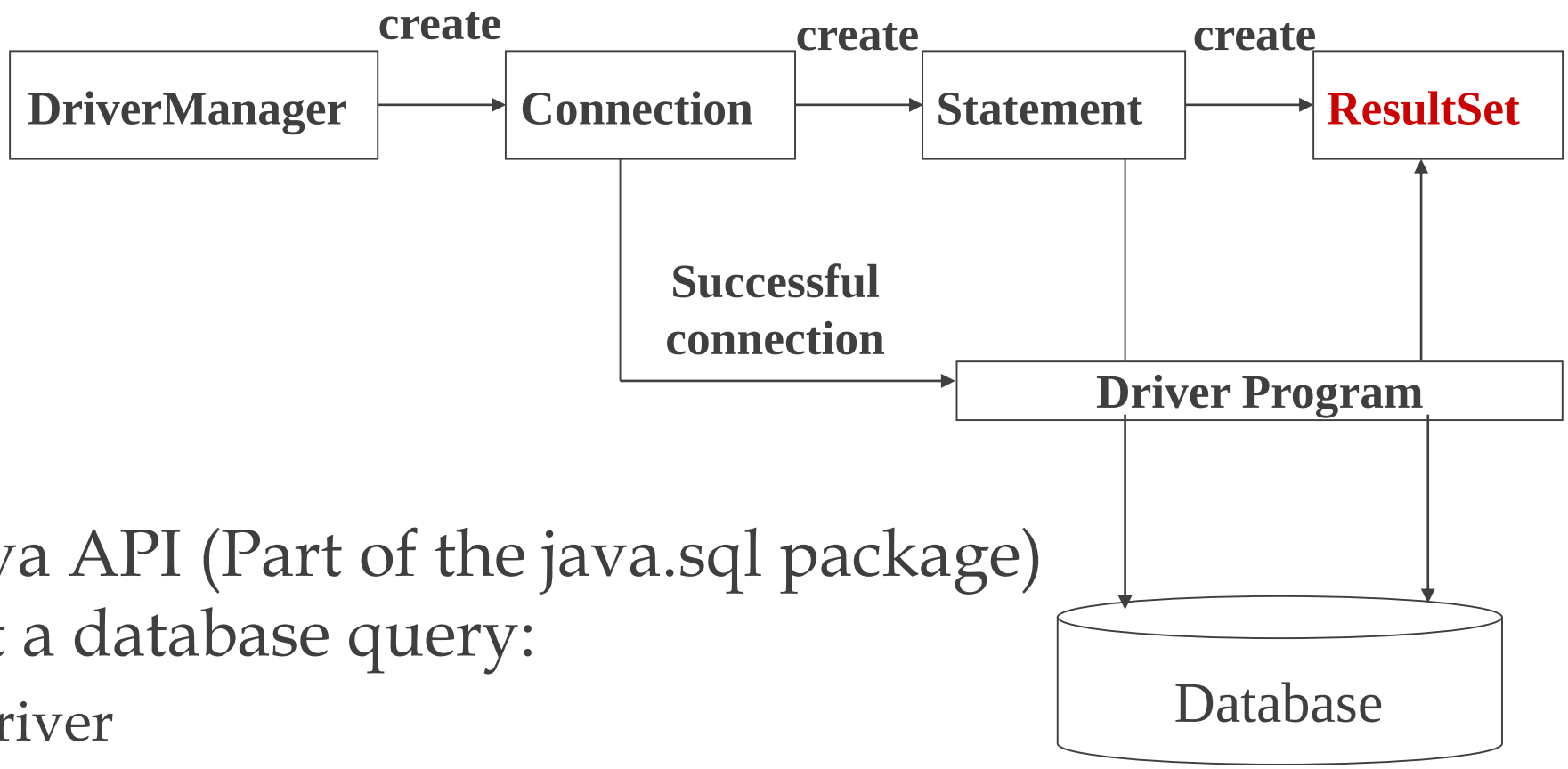
- Provide the programmer with a standard set of functions that can be invoked from the software.
- Need not to be precompiled.
- Open Database Connectivity (ODBC) standard.

应用编程接口(API)

```
Sql.*
xxx xxxx ( ) {
    ……
    ……
    ResultSet res =
        Interface ("SQL access to Postgres");
    ……
    ……
}
```



应用编程接口(API)



- Sun's JDBC: Java API (Part of the java.sql package)
- Steps to submit a database query:
  - Load the JDBC driver
  - Connect to the data source
  - Execute SQL statements

Example：通过JDBC访问数据库

过程语言/SQL

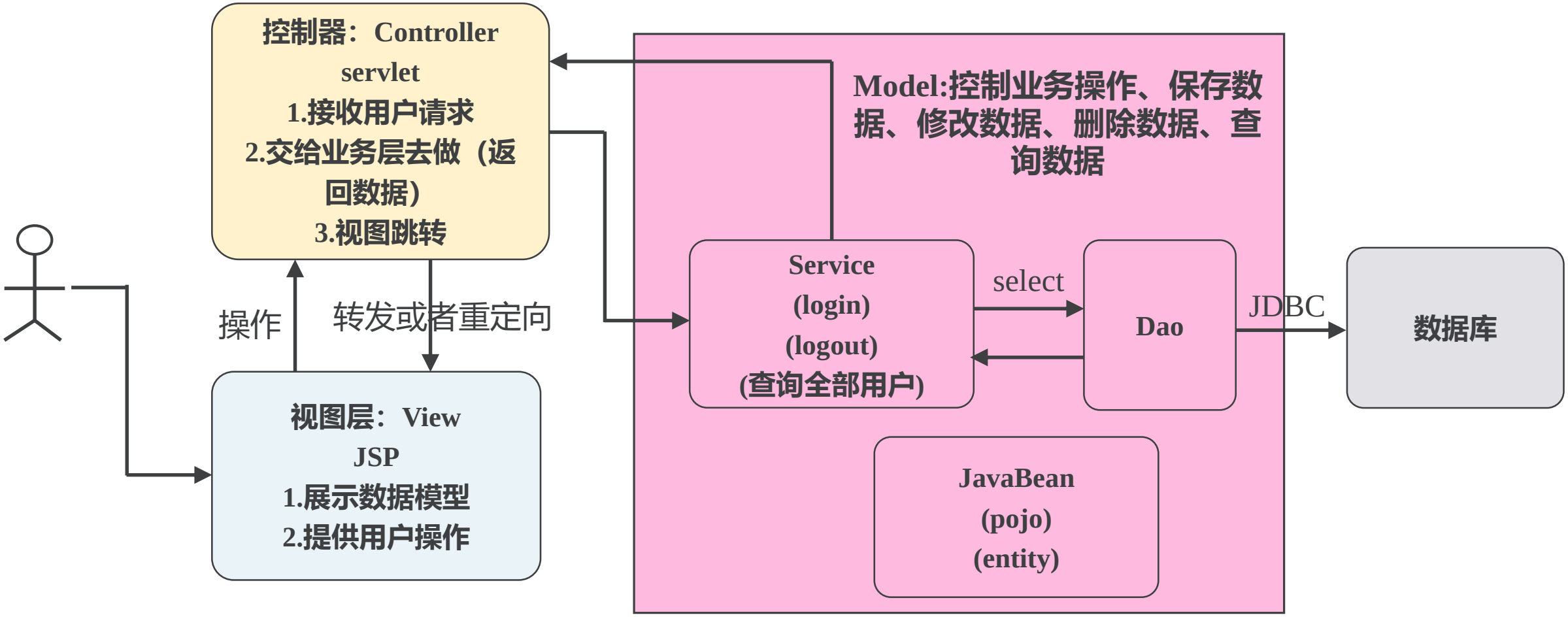
```
DECLARE
  weightedScore NUMERIC;
BEGIN
  SELECT CAST(SUM(CAST(score*credit AS
     NUMERIC))/SUM(credit) AS NUMERIC) INTO weightedScore
  FROM sc, course c
  where sc.cNo=c.cNo
    and score is not null and credit is not null
    and sc.sNo=sNoIn
    group by sc.sNo;
  RETURN weightedScore;
END;
```

Example：参照第10讲—存储过程和触发器

模型-视图-控制器体系结构(MVC)

- The Model View Controller (MVC) architecture describes a way to organize and separate the tasks of an application into three distinct parts: Model, View, and Controller.
- The View manages the output of a user interface.
- The Controller processes the user's input.
- The Model represents the data and logic of the subset of the external world used in the program

模型-视图-控制器体系结构(MVC)



关于本讲内容

祝各位学习愉快！

感谢观看！

讲解人：陆伟 教授