



福昕PDF编辑器

· 永久 · 轻巧 · 自由

点击升级会员

点击批量购买



永久使用

无限制使用次数



极速轻巧

超低资源占用，告别卡顿慢



自由编辑

享受Word一样的编辑自由



扫一扫，关注公众号

- ResultSet
- getMetaData()
- ResultSetMetaData



查询结果集模式

```
ResultSetMetaData myrsmd = myrset.getMetaData();  
For(int i=1;i<=myrsmd.getColumnCount();i++)  
{  
    System.out.println(myrsmd.getColumnName(i));  
    System.out.println(myrsmd.getColumnTypeName(i));  
}
```



查询数据库模式

- Connection
- getMetaData()
- DatabaseMetaData



查询数据库模式

```
DatabaseMetaData mydbmd = myjconn.getMetaData();
ResultSet mydmrset = mydbmd. getCollumns(null, null, "eeexam" , "%" );
While (mydmrset.next())
{
System.out.println(mydmrset.getString( "COLUMN_NAME" ), mydmrset.getString( "TYPE_NAME" ));
}
```




```
ResultSetMetaData myrsmd = myrset.getMetaData();
For(int i=1;i<=myrsmd.getColumnCount();i++)
{
    System.out.println(myrsmd.getColumnName(i));
    System.out.println(myrsmd.getColumnTypeName(i));
}
```



JSP程序例子

```
try {
    mystmt.executeUpdate("insert into eeexam values('218811011116', '0201020001', 80)");
}
catch (SQLException sqle) {
    System.out.println("Could not insert tuple. " + sqle);
}
```



JSP程序例子

```
<html>
<head>
<title>This is a test of JSP JDBC.</title>
</head>
<body>
<h3><font size=16>
<% Class.forName("org.postgresql.Driver");
    Connection myjconn = DriverManager.getConnection
        ("jdbc:postgresql://192.168.1.101:5432/postgres","postgres","123456");
    Statement mystmt = myjconn.createStatement();
    mystmt.executeUpdate("insert into exam values(118, 'xxxx', 1987)");
    mystmt.close();
    myjconn.close();
    out.println("This is a test of JSP JDBC."); %>
</font> </h3>
</body>
</html>
```


JSP程序例子

```
<html>
<head>
<title>This is a test of JSP JDBC.</title>
</head>
<body>
<h3><font size=16>
This is a test of JSP JDBC.
</font> </h3>
</body>
</html>
```

JSP

- html和Java代码可以任意交替出现
- Java代码用<% 和 %>括起来
- html代码：能最终生成合适的页面
- Java代码：符合Java语法

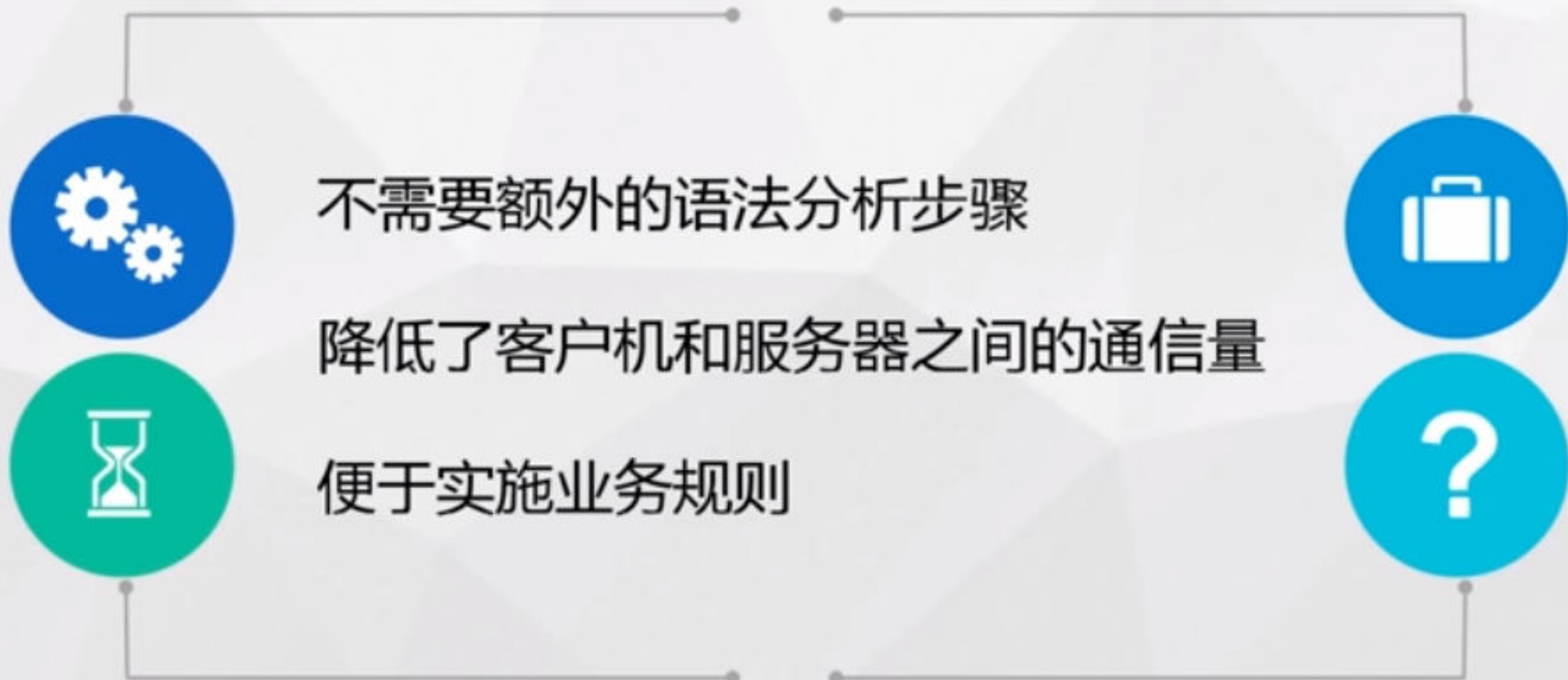




存储函数的优点

下午4:01

75% 中国大学MOOC



_____ **PI / hdsol** _____

- pgSQL是非过程化
- PL/pgSQL 是为PG扩展的过程语言
- 数据操作和流程控制
- 用于创建存储函数
- 基于块结构



- 可以使用pgSQL的所有数据类型
- uid integer;
- achieve numeric(5,2);



变量名 数据类型[:=初始表达式]



常量名 CONSTANT 数据类型:=常量表达式



赋值

变量名称: =表达式



IF 逻辑表达式 THEN
 语句序列;
END IF;

则执行 THEN

```
IF 逻辑表达式 THEN  
    语句序列1;  
ELSE  
    语句序列2 ;  
END IF;
```



```
IF 逻辑表达式1 THEN  
    语句序列1;  
ELSE  
    IF 逻辑表达式2 THEN  
        语句序列2;  
    END IF;  
END IF;
```

```
IF 逻辑表达式1 THEN  
    语句序列1;  
[ELSEIF 逻辑表达式2 THEN  
    语句序列2;  
[ELSEIF 逻辑表达式3 THEN  
    语句序列3;  
...]]  
[ELSE  
    语句序列n]  
END IF;
```

IF 逻辑表达式1 THEN

IF 逻辑表达式1 THEN

语句序列1;

[ELSEIF 逻辑表达式2 THEN

语句序列2;

[ELSEIF 逻辑表达式3 THEN

语句序列3;

...]]

[ELSE

语句序列n]

END IF;

IF 逻辑表达式1 THEN

语句序列1;

[ELSIF 逻辑表达式2 THEN

语句序列2;

[ELSIF 逻辑表达式3 THEN

语句序列3;

...]]

[ELSE

语句序列n]

END IF;



LOOP、WHILE、FOR、EXIT、CONTINUE



[<<label >>]

LOOP

循环体;

```
END LOOP[ label];
```



[<<label >>]

WHILE 逻辑表达式 LOOP

循环体;

END LOOP[label];



FOR循环

[<<label>>]

FOR count IN [REVERSE] bound1 .. bound2 [BY expression] LOOP

循环体;

END LOOP [label];

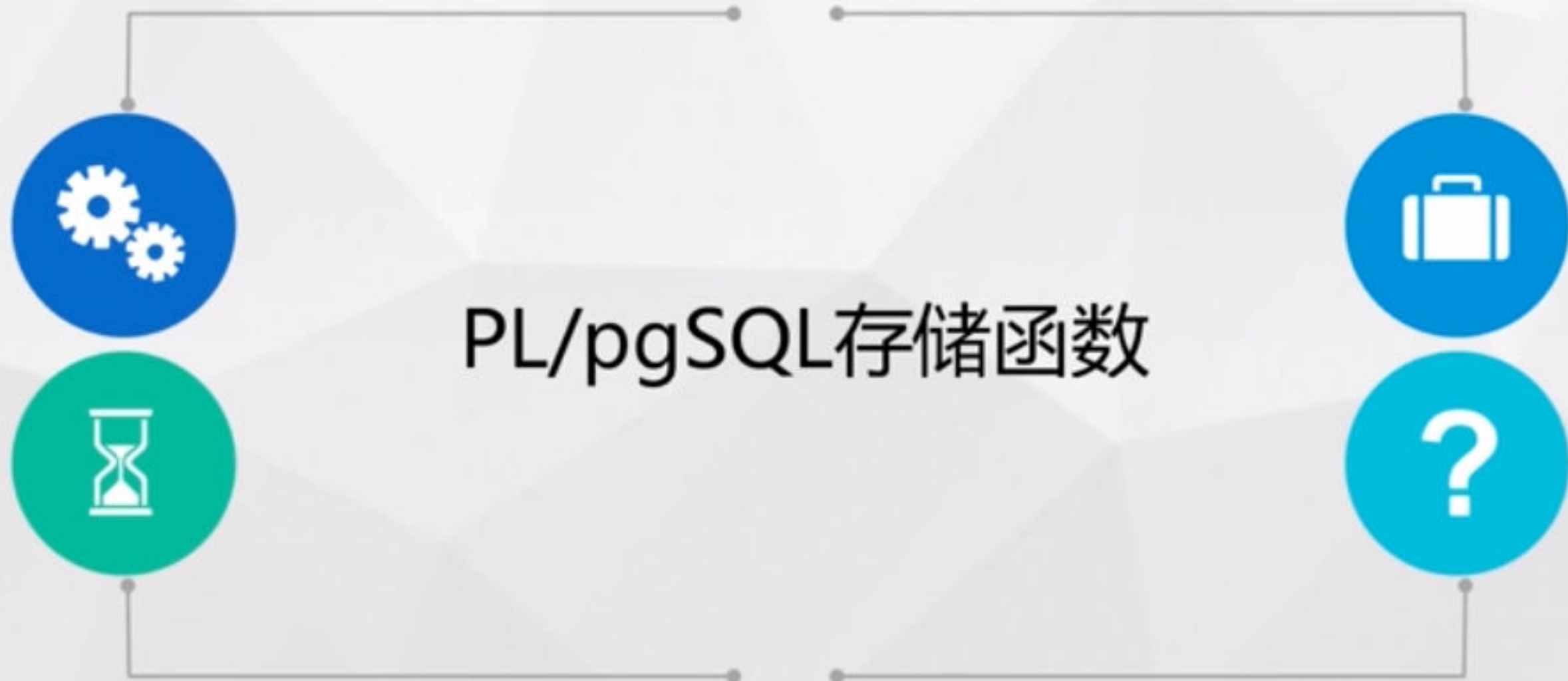



```
EXIT [label] [WHEN expression];
```



- CONTINUE [label] [WHEN expression];

如果没有label



创建存储函数

```
CREATE OR REPLACE FUNCTION add(INT, INT)
```

```
RETURNS INT
```

```
AS
```

```
$ $
```

```
DECLARE intsum INT;
```

```
BEGIN
```

```
    SELECT $1 + $2 INTO intsum;
```

```
    RETURN intsum;
```

```
END;
```

```
$ $
```

```
LANGUAGE plpgsql
```


SELECT 函数名 ([参数1, 参数2.....]);

PERFORM 函数名 ([参数1, 参数2.....]);



DROP FUNCTION 函数名 ([参数1, 参数2.....]);

