数据库系统原理

第五章 PG：数据保护

第五章 PG：数据保护

5.1 数据保护

5.2 视图

5.3 访问控制

5.4 完整性约束

5.5 触发器

5.6 事务

5.7 加密

- 保密性
- 完整性
- 可用性

- 保密性：通常保护保密性就是指仅允许经授权的读数据。
  - ✓数据值的保密
  - ✓数据存在性的保密

- 完整性：指数据的可信度。通常保护完整性就是指仅允许合法授权的数据修改。
  - ✓数据值的完整性
  - ✓数据来源的完整性

- 可用性：指对数据的期望访问能力。
  - ✓并发
  - ✓故障

- 保密性：通常保护保密性就是指仅允许合法授权的数据读。
  - ✓数据值的保密————**访问控制**
  - ✓数据存在性的保密——**视图/访问控制**
- 完整性：通常保护完整性就是指仅允许合法授权的数据修改。
  - ✓数据值的完整性————**完整性约束、事务**
  - ✓数据来源的完整性——**访问控制**
- 可用性：指对数据的期望访问能力。
  - ✓并发————**事务(并发控制)**
  - ✓故障————**事务(恢复)**

第五章：PostgreSQL数据保护

5.1数据保护

5.2视图

5.3访问控制

5.4完整性约束

5.5触发器

5.6事务

5.7加密

```
CREATE VIEW vname(a1,a2,…)
        AS SELECT ….//查询表达式
        WITH CHECK OPTION;
```

eeexam		
eeid	eid	achieve
218811011013	0205000002	92
218811011013	0210000001	85
218811011013	0201020001	88
218811011116	0210000001	90
218811011116	0201020001	80

- 示例：
  - CREATE VIEW avgachieve(eeid, average) AS

```
        SELECT eeid,avg(achieve)
            FROM eeexam  GROUP BY eeid;
```

  - 视图属性名省略时，取select结果关系属性名

avgachieve	
eeid	average
218811011013	88.7
218811011116	85

- CREATE VIEW eeexamv1 AS
  SELECT examinee.eeid, examinee.eedepa,
  exampaper.ename, eeexam.achieve
  FROM examinee, eeexam,exampaper
  WHERE examinee.eeid=eeexam.eeid
  AND eeexam.eeid= exampaper.eid;

eeexam

eeid	eid	achieve
218811011013	0205000002	92
218811011013	0210000001	85
218811011013	0201020001	88
218811011116	0210000001	90
218811011116	0201020001	80

examinee

eeid	eename	eesex	eeage	eedepa
218811011013	刘诗诗	男	20	历史学院
218811011014	刘诗诗	男	21	历史学院
218811011219	王琳懿	女	18	文学院
218811011220	王琳懿	女	19	文学院
218811011221	刘慧杰	女	19	文学院
218811011117	刘慧杰	女	19	教育学部
218811011025	张立帆	男	20	心理学院
218811011027	张立帆	男	19	心理学院
218811011028	刘慧杰	男	20	心理学院

exampaper

eid	ename	etype	eduration
0205000002	中国近现代史纲要	4	100
0210000001	大学外语	2	180
0201020001	计算机应用基础	4	120
0211000001	大学美育	3	120
0219001014	普通物理学	4	100
0110001001	教育学	1	180
0110001002	心理学	1	180

- CREATE VIEW eeexamv2 AS

  SELECT eeid,ename,achieve

  FROM eeexamv1

  WHERE sdepa='历史学院';

- 视图不可以递归定义

eeexamv1

eeid	eedepa	ename	achieve
218811011013	历史学院	中国近现代史纲要	92
218811011013	历史学院	大学外语	85
218811011013	历史学院	计算机应用基础	88

eeexamv2

eeid	ename	achieve
218811011013	中国近现代史纲要	92
218811011013	大学外语	85
218811011013	计算机应用基础	88

- 视图的使用：视图和表等价使用

```
SELECT count(*)
    FROM avgachieve
    WHERE average>=86
```

eeexam

eeid	eid	achieve
218811011013	0205000002	92
218811011013	0210000001	85
218811011013	0201020001	88
218811011116	0210000001	90
218811011116	0201020001	80

avgachieve

eeid	average
218811011013	88.7
218811011116	85

- 视图和表都是关系，都可在查询中直接应用
- DB中存储表的模式**定义和数据**
- DB中只存储视图的定义，**不存视图的数据**
- 视图数据在使用视图时临时计算

- PG只允许对可更新视图进行修改：
  - 视图是从单个关系只使用投影、选择操作导出
  - SELECT子句中只包含属性名，不包含其它表达式、聚集、DISTINCT声明，查询中不含有GROUP BY或HAVING子句
  - WHERE子句子查询不出现基础关系
  - 并且符合一般更新语句的规则（如插入时主键不能为空，需要插入操作的视图的投影列需包含基础关系的键）。

```
CREATE VIEW eemale AS
        SELECT eeid,eename,eeage
        FROM examinee
        WHERE eesex='男';
```

- 对视图eemale执行插入操作

```
INSERT INTO eemale
  VALUES('211610012938','王涛',24);
```

- 系统将执行以下语句

```
INSERT INTO examine(eeeid,eename,eeage)
  VALUES('211610012938','王涛',24);
```

第五章： PostgreSQL数据保护

5.1 数据保护

5.2 视图

5.3 访问控制

5.4 完整性约束

5.5 触发器

5.6 事务

5.7 加密

- 访问控制：定义给定角色拥有在给定数据库对象上的给定操作权限
  - 角色
  - 数据对象
  - 操作权限

- postgres
- CREATE ROLE yanni;
- CREATE ROLE yuxiaotong LOGIN;
- CREATE ROLE masu CREATEDB;
- CREATE ROLE lichen CREATEROLE;
- CREATE ROLE wangxi PASSWORD ‘654321’;
- CREATE ROLE nini SUPERUSER;

- ALTER ROLE yanni RENAME TO newyanni;
- ALTER ROLE yangchen CREATEROLE CREATEDB;
- ALTER ROLE yangchen NOCREATEROLE NOCREATEDB;
- CREATE USER lini

  与 CREATE ROLE lini login等价。
- DROP ROLE yangni;
- DROP USER wangni;

- GRANT *rolef* TO *rolet*;
- REVOKE *rolet* FROM *rolef*;

- SET ROLE
- INHERIT属性

- CREATE ROLE Alice LOGIN INHERIT;
- CREATE ROLE Bob NOINHERIT;
- CREATE ROLE Rose NOINHERIT;
- GRANT Bob TO Alice;
- GRANT Rose TO Bob;

PostgreSQL通过GRANT 语句和 REVOKE 语句实现数据库操作权限管理

- GRANT语句的一般格式：

  GRANT <权限>[,<权限>]...

  [ON <对象类型> <对象名>]

  TO <角色>[,<角色>]...

  [WITH GRANT OPTION];

- 赋予指定角色对指定对象的指定操作权限

把查询examiner表和修改考官号的权限授予角色uYing

```
GRANT UPDATE(eeid), SELECT
ON TABLE examiner
TO uYing;
```

- 授予的权限可以由DBA或授权者用REVOKE语句收回
- REVOKE语句的一般格式为：

  REVOKE <权限>[,<权限>]...

  [ON <对象类型> <对象名>]

  FROM <角色>[,<角色>]...;

第五章： PostgreSQL数据保护

5.1数据保护

5.2视图

5.3访问控制

5.4完整性约束

5.5触发器

5.6事务

5.7加密

- 主键值
  - 不能重复
  - 不能为空

- PRIMARY KEY
- 单属性构成主键
  - 属性级约束
  - 元组级约束
- 多个属性构成的主键
  - 元组级约束

将examinee表中的eeid属性声明为主键

(1)在属性级定义主键

```
CREATE TABLE examinee
    (   eeid  CHAR(9)  PRIMARY KEY，
        eename  CHAR(20) NOT NULL，
        eesex  CHAR(2) ，
         eeage  SMALLINT，
         eedepa  CHAR(20)
    );
```

在元组级定义主键

```
CREATE TABLE examinee
    (eeid  CHAR(9)，
     eename  CHAR(20) NOT NULL，
     eesex  CHAR(2) ，
     eeage  SMALLINT，
     eedepa  CHAR(20)，
     PRIMARY KEY (eeid)
);
```

将eeexam表中的eeid，eid属性组定义为主键

```
CREATE TABLE eeexam
  (eeid   CHAR(9) NOT NULL，
   eid  CHAR(4) NOT NULL，
   achieve    SMALLINT，
   PRIMARY KEY(eeid，eid)     /*只能在元组级*/
);
```

插入元组或对主键列进行更新操作时，RDBMS按照实体完整性规则自动进行检查。包括：

- 检查主键的各个属性是否为空，只要有一个为空就拒绝插入或修改
- 检查主键值是否唯一，如果不唯一则拒绝插入或修改

外键约束声明一个字段(或者一组字段)的数值必须匹配另外一个表中出现的数值。

- 在CREATE TABLE中用FOREIGN KEY子句定义哪些列为外键
- 用REFERENCES子句指明这些外键引用哪个表的哪个属性

```
CREATE TABLE eeexam
        (eeid    CHAR(9)   REFERENCES examinee(eeid)，
         eid     CHAR(4)   REFERENCES exampaper(eid)，
         achieve   SMALLINT，
       );
```

- 引用表操作导致违背外键约束
    - 拒绝相应操作

- 被引用表操作导致违背外键约束
  - NO ACTION拒绝相应操作
  - RESTRICT操作
  - 级联(CASCADE)操作
  - 设置为空值或默认值

- 列值非空（NOT NULL）

```
CREATE TABLE eeexam
      ( eeid  CHAR(9),
        eid  CHAR(4),
        achieve SMALLINT NOT NULL,
        PRIMARY KEY(eeid, eid)
       ) ;
```

- 列值唯一（UNIQUE）

```
CREATE TABLE department
      ( dname  CHAR(9),
        dloca  CHAR(10) ,
        dtele   CHAR(10) UNIQUE ,
       PRIMARY KEY (dname)
      );
```

```
CREATE TABLE examinee
        ( eeid  CHAR(9) PRIMARY KEY,
          eename CHAR(8) NOT NULL,
          eesex  CHAR(2),
          eeage  SMALLINT CHECK (0<eeage AND eeage<60) ,
          eedept  CHAR(20)
        );
```

```
CREATE TABLE examinee
        ( eeid  CHAR(9) PRIMARY KEY,
          eename CHAR(8) NOT NULL,
          eesex  CHAR(2),
         eeage  SMALLINT ,
          eedept  CHAR(20),
          CHECK (0<eeage AND eeage<60)
        );
```

- **属性CHECK约束**
  - 插入元组或修改属性的值时，检查属性上的约束条件
  - 如果不满足则操作被拒绝执行
- **元组CHECK约束义**
  - 插入元组或修改（该元组的任一）属性的值时，检查元组上的约束条件
  - 如果不满足则操作被拒绝执行

```
CREATE TABLE department
      ( dname  CHAR(9),
      dloca  CHAR(10) ,
      dtele   CHAR(10) ,
      PRIMARY KEY (dname),
      CONSTRAINT dnu UNIQUE(dtele)
      );
```

```
ALTER TABLE department DROP CONSTRAINT dnu;

ALTER TABLE department ADD CONSTRAINT dnu1 UNIQUE(dloca);
```

第五章：PostgreSQL数据保护

5.1数据保护

5.2视图

5.3访问控制

5.4完整性约束

5.5触发器

5.6事务

5.7加密

- 触发器
  - 事件驱动
  - 服务器自动激活
  - 复杂的检查和操作

- 触发器函数的语法格式

```
CREATE FUNCTION function_name() RETURNS TRIGGER AS $$

DECLARE  变量声明;

BEGIN

        函数执行代键;

  END;

$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER name {BEFORE|AFTER}  {event [OR…] }
   ON TABLE  YYY
   [FOR [EACH] {ROW|STATEMENT}]
    [ WHEN ( condition ) ]
   EXECUTE PROCEDURE function_name();
```

```
CREATE FUNCTION examineeid() RETURNS TRIGGER AS $examineeid$
BEGIN
        IF(CHAR_LENGTH(new.eeid)<>10)   THEN
                RAISE EXCEPTION '考号格式错误！';
                RETURN NULL;
        ELSE
                RETURN NEW;
        END IF;
END;
$examineeid$ LANGUAGE plpgsql;

CREATE TRIGGER examineeid_insert AFTER INSERT ON examinee FOR EACH
ROW EXECUTE PROCEDURE examineeid();
```

```
CREATE FUNCTION examinee_up() RETURNS TRIGGER AS $examinee_up$
BEGIN
        IF(new.eeid<>old.eeid)
                THEN
                        UPDATE eeexam SET eeexam.eeid=new.eeid
                        WHERE eeexam.eeid=old.eeid;
        END IF;
        RETURN NULL;
END;
$examinee_up$ LANGUAGE plpgsql;


CREATE TRIGGER examineeid_update AFTER UPDATE ON examinee FOR EACH
ROW EXECUTE PROCEDURE examinee_up();
```

```
CREATE FUNCTION eedele() RETURNS TRIGGER AS $eedele$
BEGIN
        DELETE FROM eeexam
        WHERE eeexam.eeid=old.eeid;
        RETURN NULL;
END;
$eedele$ LANGUAGE plpgsql;


CREATE TRIGGER ee_delete AFTER DELETE ON examinee FOR EACH ROW
EXECUTE PROCEDURE eedele();
```

- 执行该表上语句级BEFORE触发器;
- 执行该表上行级BEFORE触发器;
- 执行激活触发器的SQL语句;
- 执行该表上的行级AFTER触发器;
- 执行该表上的语句级AFTER触发器。

DROP TRIGGER <触发器名> ON <表名>;

第五章：PostgreSQL数据保护

5.1数据保护

5.2视图

5.3访问控制

5.4完整性约束

5.5触发器

5.6事务

5.7加密

5.6事务

数据库管理系统以事务为单位执行操作

```
UPDATE examinee SET seatno = (SELECT MIN(seatno) FROM examroom WHERE seatstatus='empty')
WHERE eeid = '218811001166' AND rid = 'room001';

UPDATE examroom SET seatstatus = 'full'
WHERE seatno=(SELECT MIN(seatno) FROM examroom WHERE seatstatus = 'empty');
```

```
BEGIN;
UPDATE examee SET seatno = (SELECT MIN(seatno) FROM examroom WHERE seatstatus='empty')
WHERE eeid = '218811001166' AND rid = 'room001';

UPDATE examroom SET seatstatus = 'full'
WHERE seatno=(SELECT MIN(seatno) FROM examroom WHERE seatstatus = 'empty');
COMMIT;
```

第五章：PostgreSQL数据保护

5.1数据保护

5.2视图

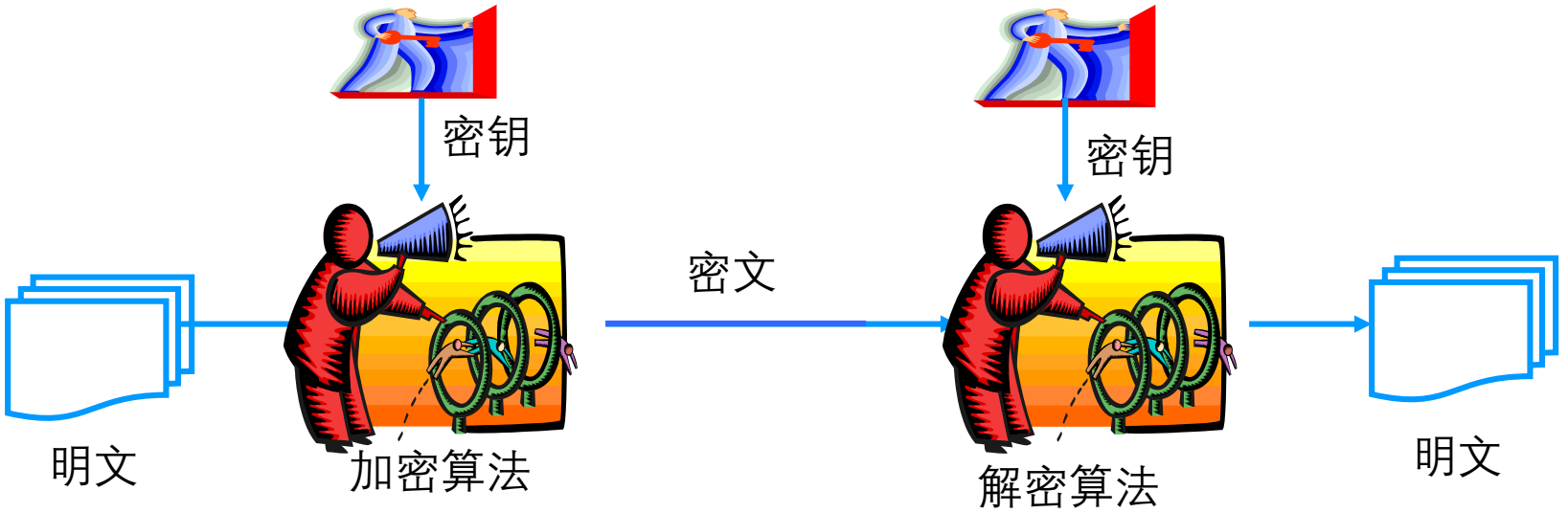
5.3访问控制

5.4完整性约束

5.5触发器

5.6事务

5.7加密


密钥
密钥
明文
加密算法
密文
解密算法
明文

- 加密算法

对称加密	非对称加密	单向加密
DES	RSA	MD5
AES	DSA	SHA

- create extension pgcrypto;

```
select MD5('43543');
```

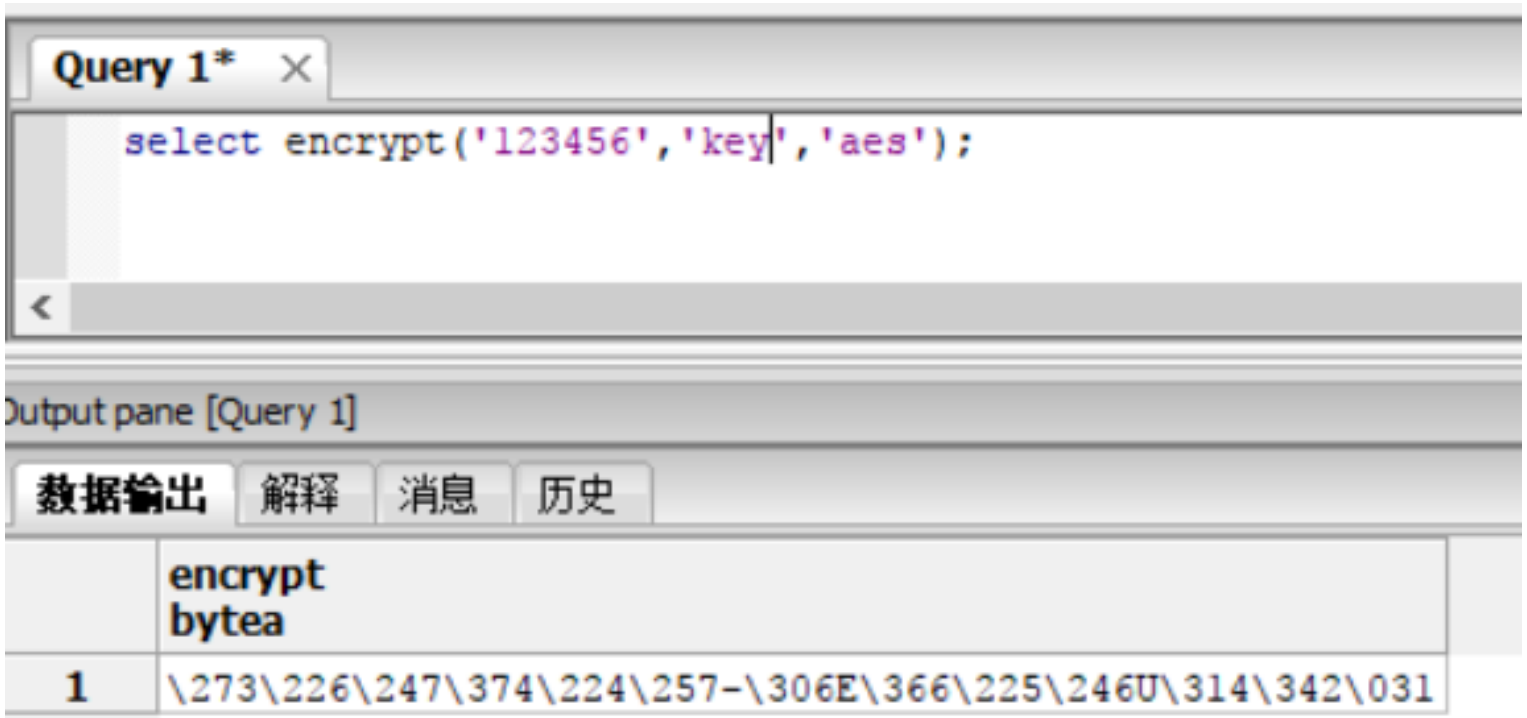
输出窗口

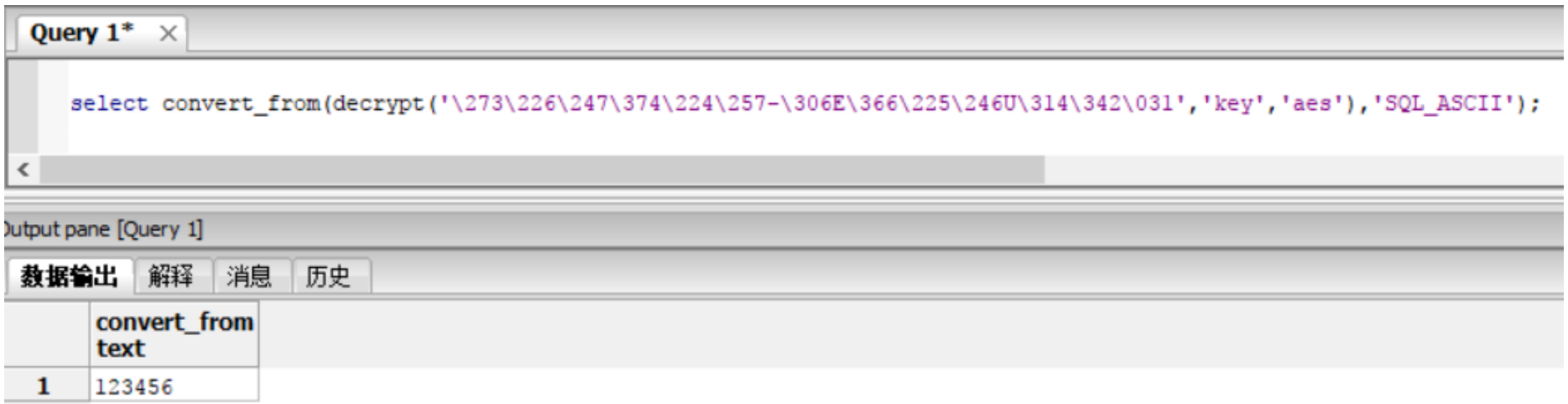
数据输出 | 解释 | 消息 | 历史

	md5 text
1	6fcdb2951819a022c6c46c51f89df49a

```
insert into examiner (erid,password)
values (‘28113699’, MD5(‘82180588’) );
```

select MD5(:pw)=(select password from examiner where erid='28113699');


Query 1*
select encrypt('123456','key','aes');
Output pane [Query 1]
数据输出
解释
消息
历史
encrypt
bytea
1
\273\226\247\374\224\257-\306E\366\225\246U\314\342\031


Query 1*
select convert_from(decrypt('\273\226\247\374\224\257-\306E\366\225\246U\314\342\031','key','aes'),'SQL_ASCII');
utput pane [Query 1]
数据输出
解释
消息
历史
convert_from
text
1
123456

- 再见