

account

id	username	password	role
11913558	Roswell	\$2a\$06\$/Frig8eAYHWDWByu5gVLgeLuCKkGkwTelrxVZI8L64yh2AwlsZy3e	School students
22011249	Robust	\$2a\$06\$eBDNuXeKSWI9bzJAwbA/ueDboZ23AG28dSrKXW6Lr4xYLqyZrvs1S	Exchange students
11937541	Ross	\$2a\$06\$2L/BHz9eUCqQcFqp/ZAG5uiXm1k0vi/s7eL0hnNkLnPc5o8HShMua	School students
11991451	Sirena	\$2a\$06\$s117nZN0YzsBoA5yLV/0y.2jFcIQddB6JpDwFLd9nxJXb7uTNoVjW	School students
11841390	Frederick	\$2a\$06\$shbBFzsNnLJCZpDQiWDqx.g2SPEu0NKPjbLm0MfZerW/I0uL9JTcm	School students
11751923	Wesley	\$2a\$06\$XTL0.fhzi.R55JHAiCGnmePAHYWAzyvHDgGbcCP4U3BLV09pMWECS	School students
50019827	Lloyd	\$2a\$06\$K7gDyK.AF2rZxqRxcC.mVeg6cfpo3N01QptZJLvlWYdSxsiGbnTQS	Teacher
30891203	Smith	\$2a\$06\$5dEcXsMJ64LzsXD58cE6teIs5PJ2KyzjhUXxgNSEtTLzKgug0uSVe	Teacher
30020826	Bright	\$2a\$06\$7P0F2PvQzufoJJGpSKvdJ0QBdyxyprM6ei8I2Fe/CcDwVdLLLHLW6	Teacher

account_log

id	username	password	role
11913558	Roswell	\$2a\$06\$/Frig8eAYHWDWByu5gVLgeLuCKkGkwTelrxVZI8L64yh2AwlsZy3e	School students
22011249	Robust	\$2a\$06\$eBDNuXeKSWI9bzJAwbA/ueDboZ23AG28dSrKXW6Lr4xYLqyZrvs1S	Exchange students
11937541	Ross	\$2a\$06\$2L/BHz9eUCqQcFqp/ZAG5uiXm1k0vi/s7eL0hnNkLnPc5o8HShMua	School students
11991451	Sirena	\$2a\$06\$s117nZN0YzsBoA5yLV/0y.2jFcIQddB6JpDwFLd9nxJXb7uTNoVjW	School students
11841390	Frederick	\$2a\$06\$shbBFzsNnLJCZpDQiWDqx.g2SPEu0NKPjbLm0MfZerW/I0uL9JTcm	School students
11751923	Wesley	\$2a\$06\$XTL0.fhzi.R55JHAiCGnmePAHYWAzyvHDgGbcCP4U3BLV09pMWECS	School students
50019827	Lloyd	\$2a\$06\$K7gDyK.AF2rZxqRxcC.mVeg6cfpo3N01QptZJLvlWYdSxsiGbnTQS	Teacher
30891203	Smith	\$2a\$06\$5dEcXsMJ64LzsXD58cE6teIs5PJ2KyzjhUXxgNSEtTLzKgug0uSVe	Teacher
30020826	Bright	\$2a\$06\$7P0F2PvQzufoJJGpSKvdJ0QBdyxyprM6ei8I2Fe/CcDwVdLLLHLW6	Teacher

初始化

安装 prcrypto

```
1 CREATE EXTENSION IF NOT EXISTS pgcrypto;
```

```
1 CREATE TABLE IF NOT EXISTS account1
2 (
3     id          INT PRIMARY KEY,
4     username    VARCHAR(20) NOT NULL,
5     password    VARCHAR(60) NOT NULL,
6     role        VARCHAR(20)
7 );
8 CREATE TABLE IF NOT EXISTS account_log
9 (
10    user_id      INT,
11    update_date  TIMESTAMP,
12    password     VARCHAR(60) NOT NULL,
13    PRIMARY KEY (user_id, update_date),
```

```
14 FOREIGN KEY (user_id) REFERENCES account (id)
15 );
```

触发器

Password_check

```
1 CREATE OR REPLACE FUNCTION password_check()
2     RETURNS TRIGGER
3 AS
4 $$
5 DECLARE
6     -- Q1 第一问 flag
7     uppercase INTEGER = 0;
8     lowercase INTEGER = 0;
9     digits    INTEGER = 0;
10
11     -- Q1 第四问 flag
12     passwordLen INTEGER;
13     pointer     INTEGER = 1;
14     asciiCode   INTEGER;
15
16     -- Q2
17     firstID     INTEGER = 0; -- id的第一位
18
19 BEGIN
20     -- ID checking
21     -- The length of student ID should be 8,
22     --
23     //////////////////////////////////////
24     //////////////////////////////////////
25     IF (NEW.id <= 9999999 OR NEW.id > 100000000) THEN
26         RAISE EXCEPTION 'The length of student ID should be 8';
27     END IF;
28
29     -- Password checking
30     -- 1.The password contains both uppercase letters,
31     lowercase letters, and digits.
32     --
33     //////////////////////////////////////
34     //////////////////////////////////////
```

```
30      -- 检查大写
31      IF (POSITION('A' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
32      IF (POSITION('B' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
33      IF (POSITION('C' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
34      IF (POSITION('D' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
35      IF (POSITION('E' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
36      IF (POSITION('F' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
37      IF (POSITION('G' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
38      IF (POSITION('H' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
39      IF (POSITION('I' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
40      IF (POSITION('J' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
41      IF (POSITION('K' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
42      IF (POSITION('L' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
43      IF (POSITION('M' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
44      IF (POSITION('N' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
45      IF (POSITION('O' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
46      IF (POSITION('P' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
47      IF (POSITION('Q' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
48      IF (POSITION('R' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
49      IF (POSITION('S' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
50      IF (POSITION('T' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
51      IF (POSITION('U' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
```

```
52     IF (POSITION('v' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
53     IF (POSITION('w' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
54     IF (POSITION('x' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
55     IF (POSITION('y' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
56     IF (POSITION('z' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
57     -- 检查小写
58     IF (POSITION('a' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
59     IF (POSITION('b' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
60     IF (POSITION('c' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
61     IF (POSITION('d' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
62     IF (POSITION('e' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
63     IF (POSITION('f' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
64     IF (POSITION('g' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
65     IF (POSITION('h' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
66     IF (POSITION('i' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
67     IF (POSITION('j' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
68     IF (POSITION('k' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
69     IF (POSITION('l' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
70     IF (POSITION('m' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
71     IF (POSITION('n' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
72     IF (POSITION('o' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
73     IF (POSITION('p' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
```

```
74     IF (POSITION('q' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
75     IF (POSITION('r' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
76     IF (POSITION('s' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
77     IF (POSITION('t' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
78     IF (POSITION('u' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
79     IF (POSITION('v' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
80     IF (POSITION('w' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
81     IF (POSITION('x' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
82     IF (POSITION('y' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
83     IF (POSITION('z' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
84     -- 检查数字
85     IF (POSITION('0' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
86     IF (POSITION('1' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
87     IF (POSITION('2' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
88     IF (POSITION('3' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
89     IF (POSITION('4' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
90     IF (POSITION('5' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
91     IF (POSITION('6' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
92     IF (POSITION('7' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
93     IF (POSITION('8' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
94     IF (POSITION('9' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
95
96     IF (uppercase = 0 OR lowercase = 0 OR digits = 0) THEN
```

```

97         RAISE EXCEPTION 'Password should contain both uppercase
letters, lowercase letters, and digits';
98     END IF;
99
100    -- 2. The length of the password is larger than 8 digits.
101    --
102    //////////////////////////////////////
103    //////////////////////////////////////
104    IF (LENGTH(NEW.password) < 8) THEN
105        RAISE EXCEPTION 'The length of the password is less
than 8 digits';
106    END IF;
107
108    -- 3. The password cannot contain user's username
109    --
110    //////////////////////////////////////
111    //////////////////////////////////////
112    IF (POSITION(NEW.username IN NEW.password) <> 0) THEN
113        RAISE EXCEPTION 'The password cannot contain user's
username';
114    END IF;
115
116    -- 4. Passwords cannot contain characters other than upper
and lower case letters, digits,
117    -- underscores( _ ), asterisks ( * ), and dollars ( $ ).
118    --
119    //////////////////////////////////////
120    //////////////////////////////////////
121    passwordLen = LENGTH(NEW.password);
122    pointer = 1;
123    WHILE (pointer <= passwordLen)
124        LOOP
125            asciiCode = ASCII(SUBSTR(NEW.password, pointer,
1261));
127            IF NOT ((asciiCode BETWEEN 48 AND 57) OR (asciiCode
128BETWEEN 65 AND 90) OR (asciiCode BETWEEN 97 AND 122)
129                OR (asciiCode = 42) OR (asciiCode = 95) OR
130                (asciiCode = 36)) THEN
131                RAISE EXCEPTION 'Passwords cannot contain
characters other than upper and lower case letters, digits,
132underscores( _ ), asterisks ( * ), and dollars ( $ ) for
133character: %, ascii: %' , SUBSTR(NEW.password, pointer, 1),
134                asciiCode;

```

```

124         END IF;
125         pointer = pointer + 1;
126
127     END LOOP;
128
129     -- Analyze the role of account
130     -- We can make sure that the first digit of ID in test
131     -- cases only can be 1,2,3 or 5.
132     --
133     //////////////////////////////////////
134     //////////////////////////////////////
135     firstID = NEW.id / 10000000;
136     IF (firstID = 1) THEN
137         NEW.role = 'School students';
138     ELSEIF (firstID = 2) THEN
139         NEW.role = 'Exchange students';
140     ELSEIF (firstID = 3 OR firstID = 5) THEN
141         NEW.role = 'Teacher';
142     END IF;
143
144     NEW.password = crypt(NEW.password, gen_salt('bf'));
145     RETURN NEW;
146 END ;
147 $$ LANGUAGE plpgsql;

```

password_trigger

```

1 CREATE TRIGGER password_trigger
2     BEFORE INSERT
3     ON account
4     FOR EACH ROW
5     EXECUTE PROCEDURE password_check();

```

测试样例

```

1 INSERT INTO account(id, username, password)
2 VALUES (30020824, 'Molly', 'abAB12e'); -- less than 8 digits
3
4
5 INSERT INTO account(id, username, password)
6 VALUES (50020825, 'Harmony', 'abcdefgh'); -- only lowercase

```

```
7
8
9 INSERT INTO account(id, username, password)
10 VALUES (30020826, 'Bright', 'Abcaac12'); -- success
11
12 INSERT INTO account(id, username, password)
13 VALUES (50020827, 'Firm', 'abc123asd'); -- no uppercase
14
15 INSERT INTO account(id, username, password)
16 VALUES (11913558, 'Roswell', '123578AbdeFa'); -- success
17
18 INSERT INTO account(id, username, password)
19 VALUES (22011249, 'Robust', 'asdaA12999999'); -- success
20
21 INSERT INTO account(id, username, password)
22 VALUES (11937541, 'Ross', 'Ba1sad19a'); -- success
23
24 INSERT INTO account(id, username, password)
25 VALUES (11991451, 'Sirena', 'ACzc901a'); -- success
26
27 INSERT INTO account(id, username, password)
28 VALUES (11841390, 'Frederick', 'asdasdzc12esLsadz'); -- success
29
30 INSERT INTO account(id, username, password)
31 VALUES (11751923, 'Wesley', 'aszCAsa12sd'); -- success
32
33 INSERT INTO account(id, username, password)
34 VALUES (50019827, 'Lloyd', 'zxcA21_9asd'); -- success
35
36 INSERT INTO account(id, username, password)
37 VALUES (50019981, 'Bob', '123Bob2ads'); -- include user's own
    name
38
39 INSERT INTO account(id, username, password)
40 VALUES (30891203, 'Smith', '_asd*91Ab'); -- success
41
42 INSERT INTO account(id, username, password)
43 VALUES (11913724, 'Zoo', 'Abc12&}asad'); -- include invalid
    character "}"
44
45 INSERT INTO account(id, username, password)
46 VALUES (1194567, 'HUANG', 'hasd19Hasd');
47 -- too short
```


On_update

```
1  CREATE OR REPLACE FUNCTION on_update()
2      RETURNS TRIGGER
3  AS
4  $$
5  DECLARE
6      -- Q1 第一问 flag
7      uppercase    INTEGER = 0;
8      lowercase    INTEGER = 0;
9      digits       INTEGER = 0;
10
11      -- Q1 第四问 flag
12      passwordLen  INTEGER;
13      pointer      INTEGER = 1;
14      asciiCode    INTEGER;
15
16
17      updateTime   INTEGER;
18      -- 最近 3 次 密码
19      lastPassword1 VARCHAR; -- 倒数第一次密码
20      lastPassword2 VARCHAR; -- 倒数第二次密码
21      lastPassword3 VARCHAR; -- 倒数第三次密码
22      boo1         BOOLEAN;
23      boo2         BOOLEAN;
24      boo3         BOOLEAN;
25
26
27  BEGIN
28      -- ID checking
29      -- The length of student ID should be 8,
30      --
31      //////////////////////////////////////
32      //////////////////////////////////////
33      IF (NEW.id <= 9999999 OR NEW.id > 100000000) THEN
34          RAISE EXCEPTION 'The length of student ID should be 8';
35      END IF;
36
37      -- Password checking
38      -- 1.The password contains both uppercase letters,
39      lowercase letters, and digits.
```

```

37      --
      //////////////////////////////////////
      //////////////////////////////////////
38      -- 检查大写
39      IF (POSITION('A' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
40      IF (POSITION('B' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
41      IF (POSITION('C' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
42      IF (POSITION('D' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
43      IF (POSITION('E' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
44      IF (POSITION('F' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
45      IF (POSITION('G' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
46      IF (POSITION('H' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
47      IF (POSITION('I' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
48      IF (POSITION('J' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
49      IF (POSITION('K' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
50      IF (POSITION('L' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
51      IF (POSITION('M' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
52      IF (POSITION('N' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
53      IF (POSITION('O' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
54      IF (POSITION('P' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
55      IF (POSITION('Q' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
56      IF (POSITION('R' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;
57      IF (POSITION('S' IN NEW.password) <> 0) THEN uppercase = 1;
END IF;

```

```
58     IF (POSITION('T' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
59     IF (POSITION('U' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
60     IF (POSITION('V' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
61     IF (POSITION('W' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
62     IF (POSITION('X' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
63     IF (POSITION('Y' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
64     IF (POSITION('Z' IN NEW.password) <> 0) THEN uppercase = 1;
    END IF;
65     -- 检查小写
66     IF (POSITION('a' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
67     IF (POSITION('b' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
68     IF (POSITION('c' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
69     IF (POSITION('d' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
70     IF (POSITION('e' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
71     IF (POSITION('f' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
72     IF (POSITION('g' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
73     IF (POSITION('h' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
74     IF (POSITION('i' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
75     IF (POSITION('j' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
76     IF (POSITION('k' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
77     IF (POSITION('l' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
78     IF (POSITION('m' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
79     IF (POSITION('n' IN NEW.password) <> 0) THEN lowercase = 1;
    END IF;
```

```
80     IF (POSITION('o' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
81     IF (POSITION('p' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
82     IF (POSITION('q' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
83     IF (POSITION('r' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
84     IF (POSITION('s' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
85     IF (POSITION('t' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
86     IF (POSITION('u' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
87     IF (POSITION('v' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
88     IF (POSITION('w' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
89     IF (POSITION('x' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
90     IF (POSITION('y' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
91     IF (POSITION('z' IN NEW.password) <> 0) THEN lowercase = 1;
      END IF;
92     -- 检查数字
93     IF (POSITION('0' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
94     IF (POSITION('1' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
95     IF (POSITION('2' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
96     IF (POSITION('3' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
97     IF (POSITION('4' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
98     IF (POSITION('5' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
99     IF (POSITION('6' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
100    IF (POSITION('7' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
101    IF (POSITION('8' IN NEW.password) <> 0) THEN digits = 1;
      END IF;
```

```

102     IF (POSITION('9' IN NEW.password) <> 0) THEN digits = 1;
    END IF;
103
104     IF (uppercase = 0 OR lowercase = 0 OR digits = 0) THEN
105         RAISE EXCEPTION 'Password should contain both uppercase
letters, lowercase letters, and digits';
106     END IF;
107
108     -- 2. The length of the password is larger than 8 digits.
109     --
////////////////////////////////////
////////////////////////////////////
110     IF (LENGTH(NEW.password) < 8) THEN
111         RAISE EXCEPTION 'The length of the password is less
than 8 digits';
112     END IF;
113
114     -- 3. The password cannot contain user's username
115     --
////////////////////////////////////
////////////////////////////////////
116     IF (POSITION(NEW.username IN NEW.password) <> 0) THEN
117         RAISE EXCEPTION 'The password cannot contain user's
username';
118     END IF;
119
120     -- 4. Passwords cannot contain characters other than upper
and lower case letters, digits,
121     -- underscores( _ ), asterisks ( * ), and dollars ( $ ).
122     --
////////////////////////////////////
////////////////////////////////////
123     passwordLen = LENGTH(NEW.password);
124     pointer = 1;
125     WHILE (pointer <= passwordLen)
126         LOOP
127             asciiCode = ASCII(SUBSTR(NEW.password, pointer,
1));
128             IF NOT ((asciiCode BETWEEN 48 AND 57) OR (asciiCode
BETWEEN 65 AND 90) OR (asciiCode BETWEEN 97 AND 122)
129                 OR (asciiCode = 42) OR (asciiCode = 95) OR
(asciiCode = 36)) THEN

```

```

130             RAISE EXCEPTION 'Passwords cannot contain
characters other than upper and lower case letters, digits,
131 underscores( _ ), asterisks ( * ), and dollars ( $ ) for
character: %, ascii: %' , SUBSTR(NEW.password, pointer, 1),
asciiCode;
132         END IF;
133         pointer = pointer + 1;
134
135     END LOOP;
136
137
138
139
140     -- 第二问部分代码
*****
*****
141     -- write your sql here
142     SELECT COUNT(*)
143     FROM account_log
144     WHERE user_id = NEW.id
145     INTO updateTime;
146
147     -- 超过 3 次更新
148     IF updateTime >= 3 THEN
149         SELECT temp.password
150         FROM (
151             SELECT ROW_NUMBER() OVER (ORDER BY update_date
DESC) AS rank, password
152             FROM account_log
153             WHERE user_id = NEW.id
154             LIMIT 3) temp
155         WHERE temp.rank = 1
156         INTO lastPassword1;
157
158         SELECT temp.password
159         FROM (
160             SELECT ROW_NUMBER() OVER (ORDER BY update_date
DESC) AS rank, password
161             FROM account_log
162             WHERE user_id = NEW.id
163             LIMIT 3) temp
164         WHERE temp.rank = 2
165         INTO lastPassword2;

```

```

166
167         SELECT temp.password
168     FROM (
169         SELECT ROW_NUMBER() OVER (ORDER BY update_date
DESC) AS rank, password
170         FROM account_log
171         WHERE user_id = NEW.id
172         LIMIT 3) temp
173     WHERE temp.rank = 3
174     INTO lastPassword3;
175
176     SELECT (lastPassword1 = crypt(NEW.password,
lastPassword1)) INTO boo1;
177     SELECT (lastPassword2 = crypt(NEW.password,
lastPassword2)) INTO boo2;
178     SELECT (lastPassword3 = crypt(NEW.password,
lastPassword3)) INTO boo3;
179
180     --RAISE EXCEPTION 'boo2: %, boo3: %', boo2, boo3;
181     IF (boo2 = TRUE OR boo3 = TRUE) THEN
182         RAISE EXCEPTION 'fail, the new password is the same
as the previous two passwords';
183     END IF;
184
185     ELSEIF updateTime = 2 THEN
186         SELECT temp.password
187     FROM (
188         SELECT ROW_NUMBER() OVER (ORDER BY update_date
DESC) AS rank, password
189         FROM account_log
190         WHERE user_id = NEW.id
191         LIMIT 3) temp
192     WHERE temp.rank = 1
193     INTO lastPassword1;
194
195     SELECT temp.password
196     FROM (
197         SELECT ROW_NUMBER() OVER (ORDER BY update_date
DESC) AS rank, password
198         FROM account_log
199         WHERE user_id = NEW.id
200         LIMIT 3) temp
201     WHERE temp.rank = 2

```

```

202         INTO lastPassword2;
203
204         SELECT (lastPassword1 = crypt(NEW.password,
lastPassword1)) INTO boo1;
205         SELECT (lastPassword2 = crypt(NEW.password,
lastPassword2)) INTO boo2;
206
207         IF (boo2 = TRUE) THEN
208             RAISE EXCEPTION 'fail, the new password is the same
as the previous passwords';
209         END IF;
210
211
212     END IF;
213
214
215     INSERT INTO account_log SELECT NEW.id, CURRENT_TIMESTAMP,
crypt(NEW.password, gen_salt('bf'));
216     RETURN NEW;
217 END
218 $$ LANGUAGE plpgsql;

```

updatePwd_trigger

```

1 CREATE TRIGGER updatePwd_trigger
2     BEFORE UPDATE
3     ON account
4     FOR EACH ROW
5     -- unaware of the new table structure
6     EXECUTE PROCEDURE on_update();

```

测试样例

```

1 -- 初始化
2 INSERT INTO account(id, username, password)
3 VALUES (30020826, 'HAHAHA', 'Abcaac12');
4 -- 初始化 UPDATE
5 UPDATE account
6 SET password = 'Abcaac12'
7 WHERE id = 30020826;
8

```



```
9
10 -- Initial Password (id=30020826): Abcaac12
11 UPDATE account
12 SET password = ''
13 WHERE id = 30020826; -- fail, the new password is too short
14
15 UPDATE account
16 SET password = 'Absazxc1213'
17 WHERE id = 30020826; -- success, the new password is different
  from the old "Absazxc1213"
18
19 UPDATE account
20 SET password = 'Abcaac12'
21 WHERE id = 30020826; -- fail, the new password is the same as
  one of the previous three passwords
22
23 UPDATE account
24 SET password = 'Absazxc1213'
25 WHERE id = 30020826; -- success, the new password is the same as
  last password, but log table has added a new row
26
27 UPDATE account
28 SET password = 'Absazxc1213'
29 WHERE id = 30020826; -- fail, the new password is the same as
  the previous two passwords
30
31 UPDATE account
32 SET password = 'Abasd12_a'
33 WHERE id = 30020826; -- success, the new is different from the
  previous three
34
35 UPDATE account
36 SET password = 'asd1sadA8z'
37 WHERE id = 30020826; -- success
38
39 UPDATE account
40 SET password = 'Absazxc1213'
41 WHERE id = 30020826; -- fail
42
43 UPDATE account
44 SET password = 'Abcaac12'
45 WHERE id = 30020826; -- success, the new is different from the
  previous three
```

```
46
47 UPDATE account
48 SET password = 'onjkn98_Q'
49 WHERE id = 30020826; -- success
50
51 UPDATE account
52 SET password = 'Absazxc1213'
53 WHERE id = 30020826;
54 -- success
55 -- Finally, password is equals to "Absazxc1213"
```