# 459-设备加载7万人，在人员列表中操作时，设备反应比较慢

# 故障分析报告

对应master版本号：bf82b6f1ebfbf02ec6513e69c8f6256b6547d0ef

## 问题描述

设备加载7万人，在人员显示列表做查询、翻页，页面响应慢；

点击添加、修改按钮，进入人员详情页面后，返回到人员显示列表，页面响应慢。

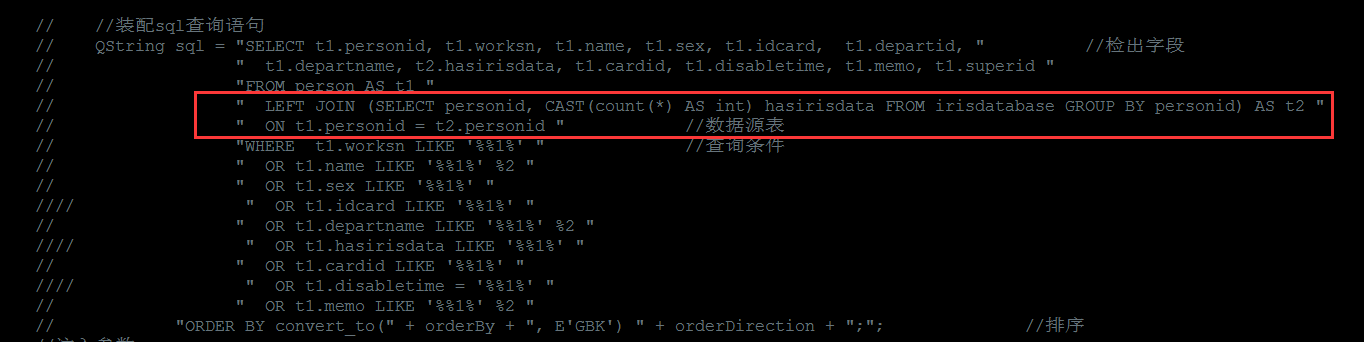
## 产生原因

之前的代码，显示每页人员时，先是把所有满足要求的人员查出来，之后再从这所有人员中，搜索到要显示的那一页人员（10个），然后显示出来。里面有两个导致慢的因素

1、人员是否有虹膜，通过person表与irisdatabase表联合查询得到结果；

2、查询全部人员，从中摘出10个来显示

原来的查询语句：



注意红框中的内容，虹膜表中所有特征按照人员分组，并与人员表中的所有人左联合。



查询出所有人员后，再从中摘出要显示的那一页的数据，进行显示。

当人员较少时，这种操作不会有太花时间的感觉，但是当总人数有7万人，特征有7\*6万，从中把所有人选出来，然后只要其中的10个人，花费时间就会非常明显。

## 问题解决方案

解决办法是：

A、启用person表中hasirisdata字段，增加触发器功能。一旦irisdatabase表中有虹膜变动（增加或删除），则对应修改person表中hasirisdata字段：增加时，该人对应hasirisdata=1；删除时，该人对应hasirisdata=0。避免大数据量下联合查询（person表与irisdatabase表之间），只查询person表即可，可大大缩短查询时间。

对应的代码包括创建触发器的语句，在main.cpp中增加：

-- 查询触发函数是否存在

SELECT proname FROM pg\_proc WHERE proname='change\_iris\_trigger';

-- 如触发函数不存在，则执行以下对设备数据库修改的语句

-- 如触发函数存在，则不需要执行以下语句

-- 1、创建触发函数

CREATE FUNCTION change\_iris\_trigger()

RETURNS trigger AS $$

BEGIN

IF TG\_OP = 'INSERT' THEN

UPDATE person SET hasirisdata=1 WHERE personid=NEW.personid;

RETURN NEW;

ELSIF TG\_OP = 'DELETE' THEN

UPDATE person SET hasirisdata=0 WHERE personid=OLD.personid;

RETURN OLD; --返回值要与ELSIF平齐,因为先插入后最好才执行返回

END IF;

END;

$$ LANGUAGE 'plpgsql' VOLATILE;

-- 2、创建触发器

CREATE TRIGGER "logging\_iris\_change"

AFTER INSERT OR DELETE

ON irisdatabase

FOR EACH ROW

EXECUTE PROCEDURE change\_iris\_trigger();

-- 3、删除原有v\_person\_detail视图

DROP VIEW v\_person\_detail;

-- 4、创建v\_person\_detail视图

CREATE OR REPLACE VIEW v\_person\_detail AS

SELECT p.personid, p.name, p.sex, p.cardid, p.idcard, p.memo,

p.departid, p.departname, p.disabletime, p.worksn, p.superid,

CASE

WHEN p.hasirisdata=1 THEN '是'::text

ELSE '否'::text

END AS hasirisdata

FROM person p;

ALTER TABLE v\_person\_detail

OWNER TO postgres;

-- 5、person表中hasirisdata字段更新为0

-- 当现有人员数据较多时，可能费时较长

-- 例如7万人数据，更新时间为5893ms

update person set hasirisdata=0;

-- 6、person表中有虹膜的人员的hasirisdata字段更新为1

-- 当现有人员及虹膜数据较多时，可能费时较长

-- 例如7万人中有2万人有虹膜数据，更新时间为7642ms

update person set hasirisdata=1 WHERE personid in (select personid from irisdatabase group by personid);

B、显示某一页时，只查询这一页的人员数据并显示，也可大大缩短查询时间

对应的SQL语句



经验证，在7万人的数据库下，大约原本10~60秒的过程缩短为不到1秒，已在可接受的范围内。

getSpecifiedPage函数是在页面前后翻页时调用的显示函数；

dispPageHasGivenRecord函数是从新增或修改人员界面退出后，显示这个人员所在页面时的显示函数。

dispPageHasGivenRecord花费时间会比getSpecifiedPage长一些，因为需要在所有人员中找出这个人员属于第几个页面，并显示。

目前在所有人员中找出这个人员属于第几个页面的花费时间约为2秒（7万人的库）。示例代码为：

select t.rn from

(select t1.personid, row\_number() over(ORDER BY convert\_to(name, E'GBK')) rn from v\_person\_detail t1

WHERE t1.worksn LIKE '%%1%'

OR t1.name LIKE '%%1%'

OR t1.sex LIKE '%%1%'

OR t1.idcard LIKE '%%1%'

OR t1.departname LIKE '%%1%'

--OR t1.hasirisdata LIKE '%%1%'

OR t1.cardid LIKE '%%1%'

OR t1.memo LIKE '%%1%'

) t

where t.personid ='400f83f3-d156-f143-96ff-fd7fd6fc1090'

## 相关功能影响

由于是否已注册的信息直接写在person表中，有些相关的查找可能需要修改。

## 故障引入版本

“历史遗留”

## 问题解决版本

对应版本号为vsdk4.0.4.0