Helpdesk Database

By: Tom Morse

Table of Contents

Executive Summary	**3**
Entity Relationship Diagram	4
Table Create Statements	5
• **Call Logs**	5
• **Products**	6
• **Troubleshoot**	7
• **Staff**	8
• **Computers**	9
• **Workstations**	10
• **Ratings**	11
• **Resolutions**	12
• **Cases**	13
View	14
Reports and Queries	17
Stored Procedures	20
Triggers	20
Security	21
Implementation Notes, Known Problems, Future Enhancements	23

Executive Summary

The purpose of this database is to keep track of all calls and issues coming into a helpdesk call center. The main benefits of the database are; that it keeps track of the staff members information, the caller's phone number and last name if available and the products they are having issues with. From there, the database puts all of this information together and creates a central table that connects all other tables. The main goal of this is to make it simple to find out what staff members handled which calls and how that staff member did with resolving the issue at hand. New calls, workstations, and products can be easily added or edited through the database, this includes; caller's last name, phone number, a workstations staff member, or computer.

Entity Relationship Diagram - Helpdesk Database

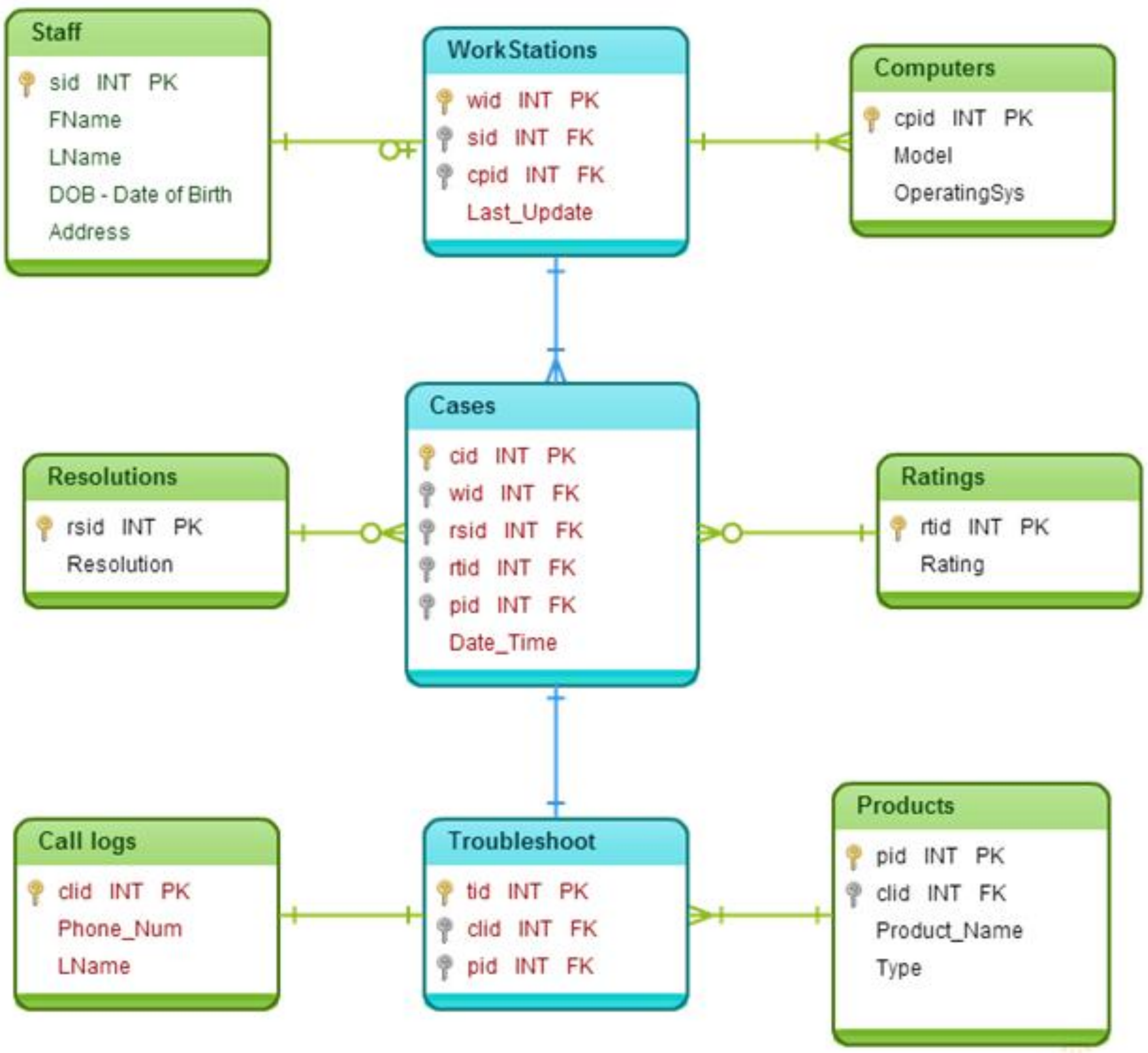


Table Create Statements

This section depicts each table's functional dependencies, create statements, and sample data

Call Logs:

Purpose: To log the phone number and last name of every caller, if available.

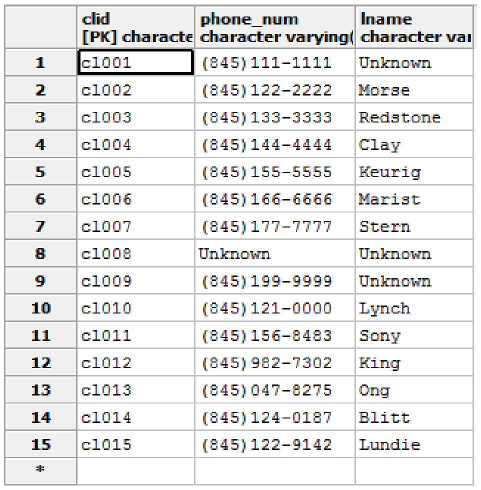
Functional Dependencies

Clid $\Longrightarrow$ Phone_num, lname

Create Statement

```
DROP TABLE IF EXISTS calllogs ;

CREATE TABLE calllogs (

    clid CHAR (5),

    phone_num VARCHAR (15),

    lname VARCHAR (15),

    PRIMARY KEY (clid)) ;
```

Sample Data



	clid [PK] characte	phone_num character varying(	lname character va
1	c1001	(845)111-1111	Unknown
2	c1002	(845)122-2222	Morse
3	c1003	(845)133-3333	Redstone
4	c1004	(845)144-4444	Clay
5	c1005	(845)155-5555	Keurig
6	c1006	(845)166-6666	Marist
7	c1007	(845)177-7777	Stern
8	c1008	Unknown	Unknown
9	c1009	(845)199-9999	Unknown
10	c1010	(845)121-0000	Lynch
11	c1011	(845)156-8483	Sony
12	c1012	(845)982-7302	King
13	c1013	(845)047-8275	Ong
14	c1014	(845)124-0187	Blitt
15	c1015	(845)122-9142	Lundie
*			

Products:

Purpose: Store different products that helpdesk could help a caller with and what type of product it is.

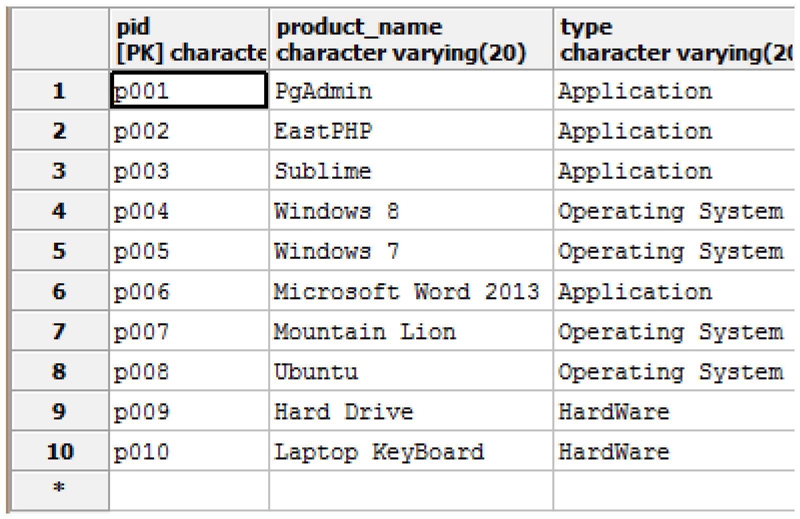
Functional Dependencies

pid $\Longrightarrow$ Product_name, type

Create Statement

```
 DROP TABLE IF EXISTS products ;

CREATE TABLE products(

        pid CHAR (4),

        product_name VARCHAR (20) NOT NULL,

         type VARCHAR (20) NOT NULL,

         PRIMARY KEY (pid)) ;
```

Sample Data



	pid [PK] characte	product_name character varying(20)	type character varying(2
1	p001	PgAdmin	Application
2	p002	EastPHP	Application
3	p003	Sublime	Application
4	p004	Windows 8	Operating System
5	p005	Windows 7	Operating System
6	p006	Microsoft Word 2013	Application
7	p007	Mountain Lion	Operating System
8	p008	Ubuntu	Operating System
9	p009	Hard Drive	HardWare
10	p010	Laptop KeyBoard	HardWare
*			

Troubleshoot:

Purpose: To match a call with a product, every time the hotline is called.

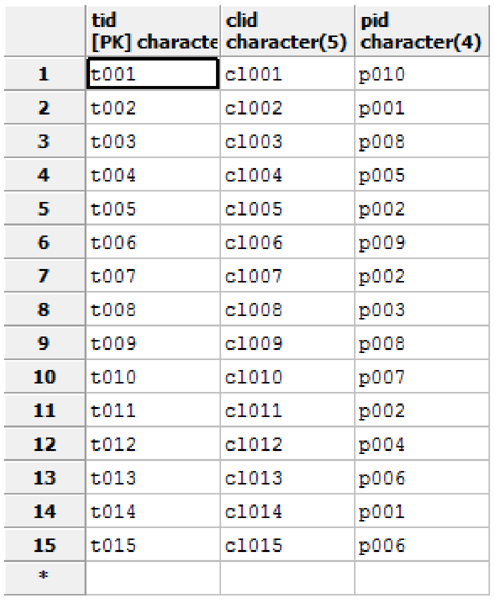
Functional Dependencies

tid $\Longrightarrow$ clid, pid

Create Statement

```
    DROP TABLE IF EXISTS troubleshoot ;

    CREATE TABLE troubleshoot(

        tid CHAR (4),

        clid CHAR (5) REFERENCES calllogs(clid),

        pid CHAR (4) REFERENCES products(pid),

         PRIMARY KEY (tid)) ;
```

Sample Data



	tid [PK] characte	clid character(5)	pid character(4)
1	t001	cl001	p010
2	t002	cl002	p001
3	t003	cl003	p008
4	t004	cl004	p005
5	t005	cl005	p002
6	t006	cl006	p009
7	t007	cl007	p002
8	t008	cl008	p003
9	t009	cl009	p008
10	t010	cl010	p007
11	t011	cl011	p002
12	t012	cl012	p004
13	t013	cl013	p006
14	t014	cl014	p001
15	t015	cl015	p006
*			

Staff:

Purpose: To store all necessary information of all current staff members.

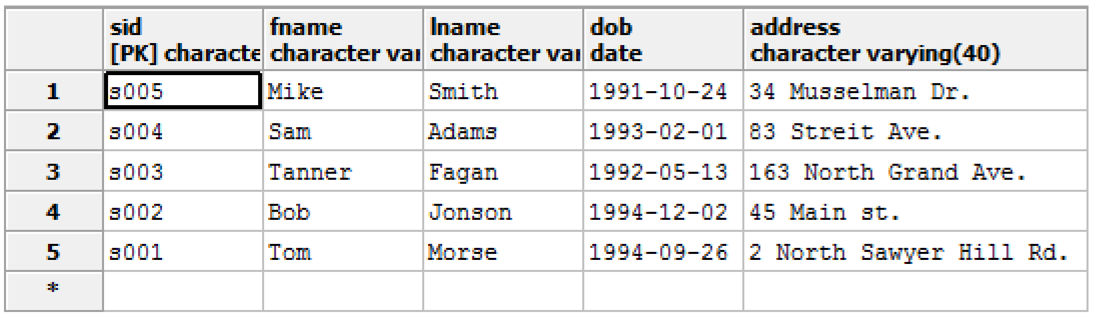
Functional Dependencies

sid $\Longrightarrow$ Fname, lname, dob, address

Create Statement

```
DROP TABLE IF EXISTS staff ;

CREATE TABLE staff(

    sid CHAR (4),

    fname VARCHAR (15) NOT NULL,

    lname VARCHAR (15) NOT NULL,

    dob DATE NOT NULL,

    address VARCHAR (40) NOT NULL,

    PRIMARY KEY (sid)) ;
```

Sample Data



	sid [PK] characte	fname character va	lname character va	dob date	address character varying(40)
1	s005	Mike	Smith	1991-10-24	34 Musselman Dr.
2	s004	Sam	Adams	1993-02-01	83 Streit Ave.
3	s003	Tanner	Fagan	1992-05-13	163 North Grand Ave.
4	s002	Bob	Jonson	1994-12-02	45 Main st.
5	s001	Tom	Morse	1994-09-26	2 North Sawyer Hill Rd.
*					

Computers:

Purpose: Store all computers that the helpdesk office has available even if they are currently not being used by a workstation.

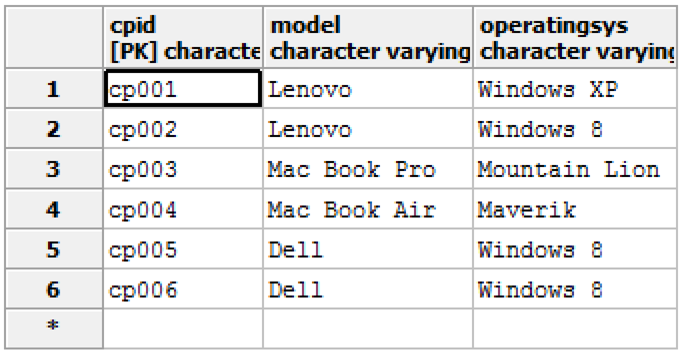
Functional Dependencies

cpid $\longrightarrow$ model, operatingsys

Create Statement

```
DROP TABLE IF EXISTS computers ;

CREATE TABLE computers(

    cpid CHAR (5),

     model VARCHAR (20) NOT NULL,

     operatingsys VARCHAR (20) NOT NULL,

     PRIMARY KEY (cpid)) ;
```

Sample Data



	cpid [PK] characte	model character varying	operatingsys character varying
1	cp001	Lenovo	Windows XP
2	cp002	Lenovo	Windows 8
3	cp003	Mac Book Pro	Mountain Lion
4	cp004	Mac Book Air	Maverik
5	cp005	Dell	Windows 8
6	cp006	Dell	Windows 8
*			

Workstations:

Purpose: To keep a record of which staff member and computer are at each workstation.

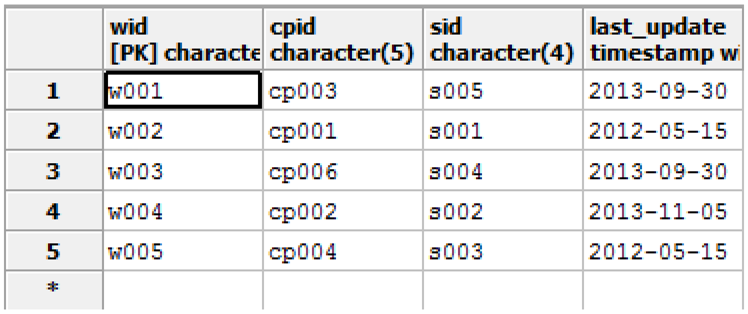
Functional Dependencies

wid $\Longrightarrow$ Cpid, sid, last_update

Create Statement

```
DROP TABLE IF EXISTS workstations ;

CREATE TABLE workstations(

    wid CHAR (4),

    cpid CHAR (5) REFERENCES computers(cpid),

    sid CHAR (4) REFERENCES staff(sid),

    last_update TIMESTAMP NOT NULL,

   PRIMARY KEY (wid)) ;
```

Sample Data



	wid [PK] characte	cpid character(5)	sid character(4)	last_update timestamp wi
1	w001	cp003	s005	2013-09-30
2	w002	cp001	s001	2012-05-15
3	w003	cp006	s004	2013-09-30
4	w004	cp002	s002	2013-11-05
5	w005	cp004	s003	2012-05-15
*				

Ratings:

Purpose: Stores the available ratings that a caller can give their interaction with helpdesk.

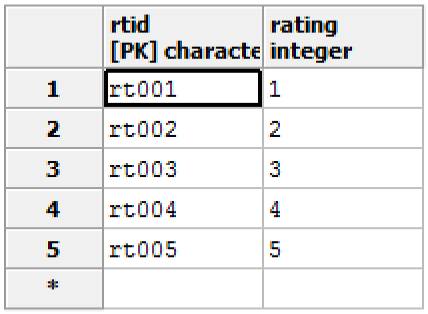
Functional Dependencies

$$\begin{tikzcd} \text{rtid} \arrow[r] & \text{rating} \end{tikzcd}$$

Create Statement

```
DROP TABLE IF EXISTS ratings ;

CREATE TABLE ratings(

    rtid CHAR (5),

    rating INT NOT NULL,

    PRIMARY KEY (rtid)) ;
```

Sample Data



	rtid [PK] characte	rating integer
1	rt001	1
2	rt002	2
3	rt003	3
4	rt004	4
5	rt005	5
*		

Resolutions:

Purpose: Stores the available options of whether or not the caller issue was resolved.

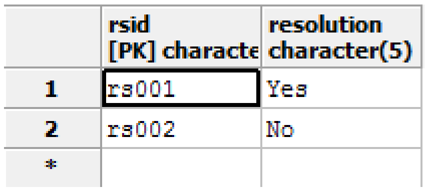
Functional Dependencies

$$\text{rsid} \longrightarrow \text{resolution}$$

Create Statement

```
DROP TABLE IF EXISTS resolutions ;

CREATE TABLE resolutions(

    rsid CHAR (5),

    resolution CHAR (5) NOT NULL,

    PRIMARY KEY (rsid)) ;
```

Sample Data



	rsid [PK] characte	resolution character(5)
1	rs001	Yes
2	rs002	No
*		

Cases:

Purpose: Bring all the data together by combining it and assigning each set of data a case number for easy record keeping and look up.

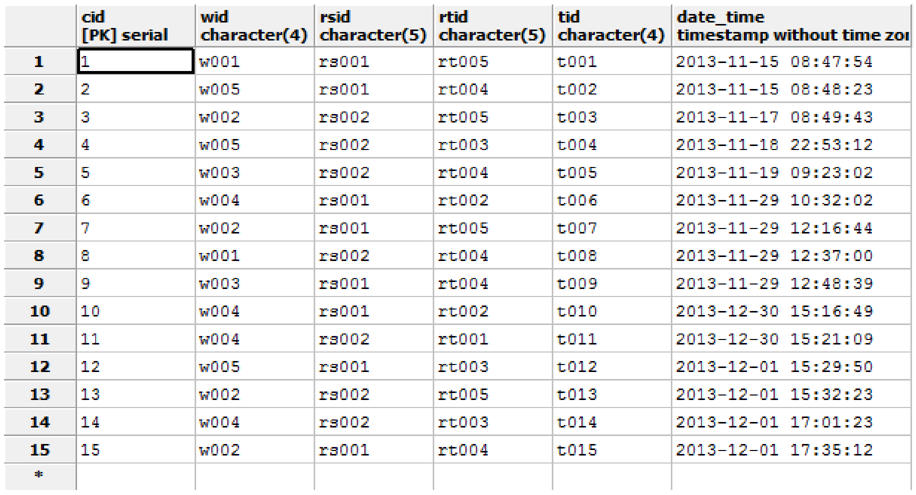
Functional Dependencies

cid $\Longrightarrow$ Wid, rsid, rtid, tid, date_time

Create Statement

```
DROP TABLE IF EXISTS cases ;

CREATE TABLE cases(

    cid SERIAL,

     wid CHAR (4) REFERENCES workstations(wid),

    rsid CHAR (5) REFERENCES resolutions(rsid),

    rtid CHAR (5) REFERENCES ratings(rtid),

     tid CHAR (4) REFERENCES troubleshoot(tid),

    date_time TIMESTAMP NOT NULL,

     PRIMARY KEY (cid)) ;
```

Sample Data



	cid [PK] serial	wid character(4)	rsid character(5)	rtid character(5)	tid character(4)	date_time timestamp without time zo
1	1	w001	rs001	rt005	t001	2013-11-15 08:47:54
2	2	w005	rs001	rt004	t002	2013-11-15 08:48:23
3	3	w002	rs002	rt005	t003	2013-11-17 08:49:43
4	4	w005	rs002	rt003	t004	2013-11-18 22:53:12
5	5	w003	rs002	rt004	t005	2013-11-19 09:23:02
6	6	w004	rs001	rt002	t006	2013-11-29 10:32:02
7	7	w002	rs001	rt005	t007	2013-11-29 12:16:44
8	8	w001	rs002	rt004	t008	2013-11-29 12:37:00
9	9	w003	rs001	rt004	t009	2013-11-29 12:48:39
10	10	w004	rs001	rt002	t010	2013-12-30 15:16:49
11	11	w004	rs002	rt001	t011	2013-12-30 15:21:09
12	12	w005	rs001	rt003	t012	2013-12-01 15:29:50
13	13	w002	rs002	rt005	t013	2013-12-01 15:32:23
14	14	w004	rs002	rt003	t014	2013-12-01 17:01:23
15	15	w002	rs001	rt004	t015	2013-12-01 17:35:12
*						

Views

This section depicts each views, which are tables that show large amount of data in a way that is simple to look at and understand.

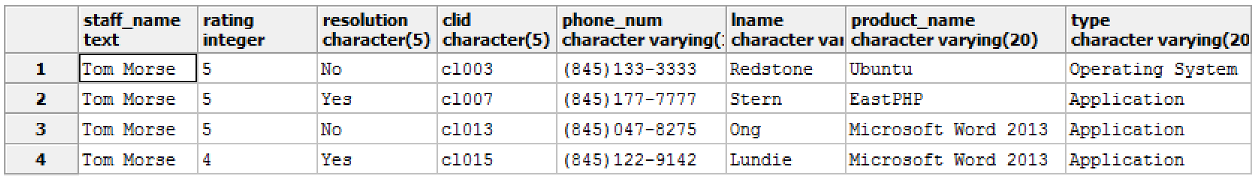
Main Info for staff member "s001":

Purpose: To show all of the callers that staff member "s001" has dealt with and the main information pertaining to each caller in an easy to view way.

Create Statement

```
CREATE VIEW all_callers_for_s001 AS

SELECT

            (s.fname||' '||s.lname) AS staff_name,

            rt.rating,

            rs.resolution,

            cl.*,

           p.product_name,

           p.type

FROM

          staff s,

         workstations w,

         cases c,

         ratings rt,

         resolutions rs,

         troubleshoot t,
         products p,
        calllogs cl
WHERE
        s.sid = 's001'
        AND
        s.sid = w.sid
        AND
        w.wid = c.wid
        AND
        c.rsid = rs.rsid
        AND
        c.rtid = rt.rtid
       AND
       c.tid = t.tid
       AND
       t.pid = p.pid
       AND
       t.clid = cl.clid
```

Sample Data



	staff_name text	rating integer	resolution character(5)	clid character(5)	phone_num character varying(	lname character va	product_name character varying(20)	type character varying(20
1	Tom Morse	5	No	cl003	(845)133-3333	Redstone	Ubuntu	Operating System
2	Tom Morse	5	Yes	cl007	(845)177-7777	Stern	EastPHP	Application
3	Tom Morse	5	No	cl013	(845)047-8275	Ong	Microsoft Word 2013	Application
4	Tom Morse	4	Yes	cl015	(845)122-9142	Lundie	Microsoft Word 2013	Application

Staff with rating above average:

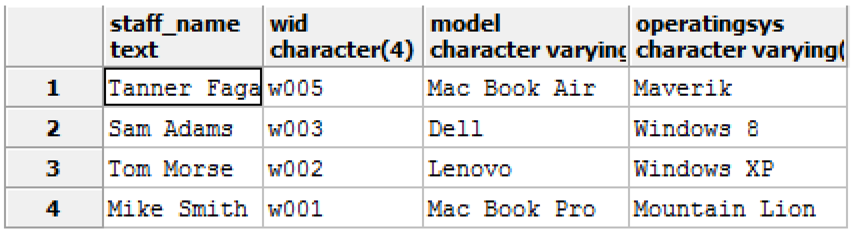
Purpose: To show all staff members who have had at least one caller interaction with a rating above the average.

Create Statement

```
CREATE VIEW staff_name_above_avg_rating AS

SELECT DISTINCT

      (s.fname||' '||s.lname) AS staff_name,

       w.wid,

       cp.model,

       cp.operatingsys

FROM

      staff s, workstations w, computers cp, cases c, ratings rt

WHERE

     s.sid = w.sid

     AND

     w.cpid = cp.cpid

     AND

      w.wid = c.wid

      AND

     c.rtid = rt.rtid

     AND
```

```
    rt.rating >(SELECT AVG(rt.rating) FROM ratings rt)
```

Sample Data



	staff_name text	wid character(4)	model character varying	operatingsys character varying(
1	Tanner Faga	w005	Mac Book Air	Maverik
2	Sam Adams	w003	Dell	Windows 8
3	Tom Morse	w002	Lenovo	Windows XP
4	Mike Smith	w001	Mac Book Pro	Mountain Lion

Reports and Queries

This section depicts reports and queries, these are an easy way to put small amount of data together in order to make quick and easy reports of the data

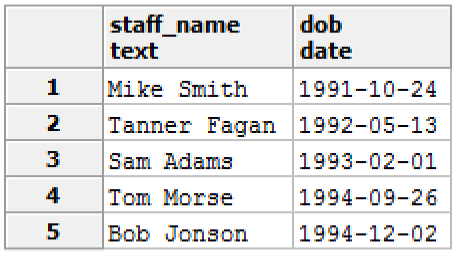
Staff Members DOB:

Purpose: To show when all the staff members were born.

Create Statement

```
  SELECT
      (fname||' '||lname) AS staff_name,
      dob
  FROM
      staff
  ORDER BY
      dob
```

Sample Data



	staff_name text	dob date
1	Mike Smith	1991-10-24
2	Tanner Fagan	1992-05-13
3	Sam Adams	1993-02-01
4	Tom Morse	1994-09-26
5	Bob Jonson	1994-12-02

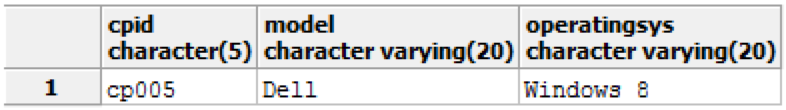
Computers not in use:

Purpose: To show which computers are not being used by a workstation.

Create Statement

```
SELECT

    *

FROM

    computers

WHERE

    cpid NOT IN (SELECT cpid FROM workstations)
```

Sample Data



	cpid character(5)	model character varying(20)	operatingsys character varying(20)
1	cp005	Dell	Windows 8

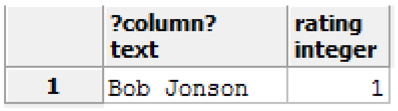
Staff with Low rating:

Purpose: To show who got a rating of 1.

Create Statement

```
SELECT
    (s.fname||' '||s.lname),
    rt.rating
FROM
    staff s, workstations w, cases c, ratings rt
WHERE
    s.sid = w.sid
    AND
    w.wid = c.wid
    AND
    c.rtid = rt.rtid
    AND
    rt.rating = 1
```

Sample Data

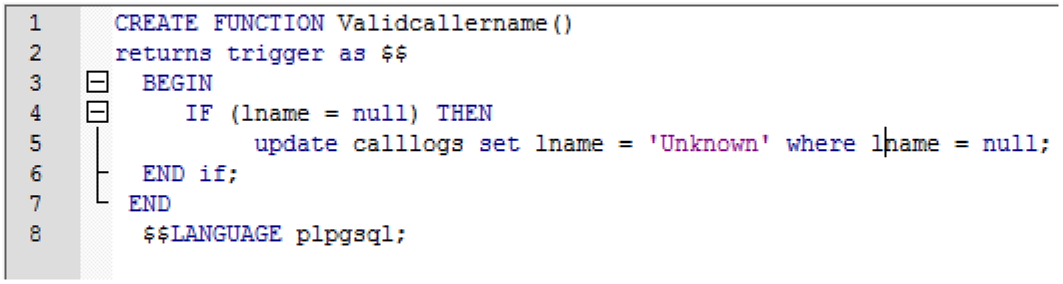


	?column? text	rating integer
1	Bob Jonson	1

Stored Procedures

This section depicts Stored Procedures, which are used to allow ease of use of certain needed functions, this allows these functions to always be stored and called upon when needed

Purpose: To validate and correct "null" in the last name field of the call logs table

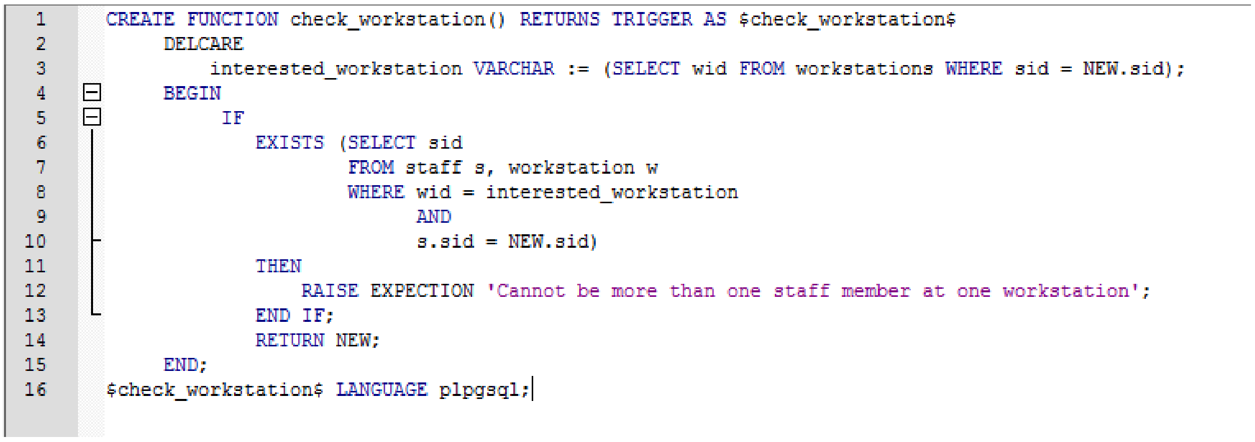


```
CREATE FUNCTION Validcallername()
returns trigger as $$
  BEGIN
     IF (lname = null) THEN
          update calllogs set lname = 'Unknown' where lname = null;
  END if;
 END
  $$LANGUAGE plpgsql;
```

Triggers

This section depicts Triggers, which allow the admin to set constrants on certain parts of the data base.

Purpose: Checks to make sure there is only one staff member assigned to a workstation



```
CREATE FUNCTION check_workstation() RETURNS TRIGGER AS $check_workstation$
     DELCARE
         interested_workstation VARCHAR := (SELECT wid FROM workstations WHERE sid = NEW.sid);
     BEGIN
          IF
             EXISTS (SELECT sid
                     FROM staff s, workstation w
                     WHERE wid = interested_workstation
                           AND
                           s.sid = NEW.sid)
             THEN
                 RAISE EXPECTION 'Cannot be more than one staff member at one workstation';
             END IF;
             RETURN NEW;
     END;
$check_workstation$ LANGUAGE plpgsql;
```

Security

This section depicts the 3 different default users and what privileges they have. The first user, the admin, has the ability to insert, update, delete, or select any on any table, they have complete access to the database. The next user is the staff user which gets privileges to insert, update, delete, or select on certain tables but not all, and lastly the user does not get privileges to insert, update, delete, or select on any tables in the database.

```
CREATE USER helpdeskadmin WITH PASSWORD 'staff';

Revoke all on

Revoke all on Staff                    From helpdeskadmin
Revoke all on Computers                From helpdeskadmin
Revoke all on Workstations             From helpdeskadmin
Revoke all on Ratings                  From helpdeskadmin
Revoke all on Resolutions              From helpdeskadmin
Revoke all on Calllogs                 From helpdeskadmin
Revoke all on Products                 From helpdeskadmin
Revoke all on Troubleshoot             From helpdeskadmin
Revoke all on Cases                    From helpdeskadmin


Grant insert, update, delete, select on Staff           To helpdeskadmin
Grant insert, update, delete, select on Computers       To helpdeskadmin
Grant insert, update, delete, select on Workstations    To helpdeskadmin
Grant insert, update, delete, select on Ratings         To helpdeskadmin
Grant insert, update, delete, select on Resolutions     To helpdeskadmin
Grant insert, update, delete, select on Calllogs        To helpdeskadmin
Grant insert, update, delete, select on Products        To helpdeskadmin
Grant insert, update, delete, select on Troubleshoot    To helpdeskadmin
Grant insert, update, delete, select on Cases           To helpdeskadmin


CREATE USER  helpdeskstaff WITH PASSWORD 'staff';

Revoke all on Staff                    From helpdeskstaff
Revoke all on Computers                From helpdeskstaff
Revoke all on Workstations             From helpdeskstaff
Revoke all on Ratings                  From helpdeskstaff
Revoke all on Resolutions              From helpdeskstaff
Revoke all on Calllogs                           From helpdeskstaff
Revoke all on Products                           From helpdeskstaff
Revoke all on Troubleshoot                       From helpdeskstaff
Revoke all on Cases                              From helpdeskstaff


Grant insert, update, delete, select on Resolutions     To helpdeskstaff
Grant insert, update, delete, select on Calllogs        To helpdeskstaff
Grant insert, update, delete, select on Products        To helpdeskstaff
Grant insert, update, delete, select on Troubleshoot    To helpdeskstaff
Grant insert, update, delete, select on Cases           To helpdeskstaff
Grant select on Staff                                   To helpdeskstaff
Grant select on Computers                               To helpdeskstaff
Grant select on Workstations                            To helpdeskstaff
Grant select on Ratings                                 To helpdeskstaff


CREATE USER  helpdeskuser WITH PASSWORD 'password';

Revoke all on Staff                              From helpdeskuser
Revoke all on Computers                          From helpdeskuser
Revoke all on Workstations                       From helpdeskuser
Revoke all on Ratings                            From helpdeskuser
Revoke all on Resolutions                        From helpdeskuser
Revoke all on Calllogs                           From helpdeskuser
Revoke all on Products                           From helpdeskuser
Revoke all on Troubleshoot                       From helpdeskuser
Revoke all on Cases                              From helpdeskuser


Grant select on Staff                                   To helpdeskuser
Grant select on Computers                               To helpdeskuser
Grant select on Workstations                            To helpdeskuser
Grant select on Ratings                                 To helpdeskuser
Grant select on Resolutions                             To helpdeskuser
Grant select on Calllogs                                To helpdeskuser
Grant select on Products                                To helpdeskuesr
Grant select on Troubleshoot                            To helpdeskuser
Grant select on Cases                                   To helpdeskuser
```

Implementation Notes, Known Problems, and Future Enhancements

Implementation Notes:

- Implementation should be simple as long the basic structure is mostly unchanged
- Views and Reports could be added to show different part of the database if preferred

Known Problems:

- Limited space, in the call logs, the system as of right now can only handle up to 999 calls in total and if the helpdesk is a regularly used entity then more space may need to be added.
- Another known issue, is the cases id can also only handle up to 999, with that more space would need to be add in future

Future Enhancements:

- As of right now, the database can handle the major parts of a helpdesk but pieces could be added or expanded upon, such as certain products could be handled by certain workstations and not other, by doing this, the amount of positive resolution could increase do to more knowledge focused staff and system.