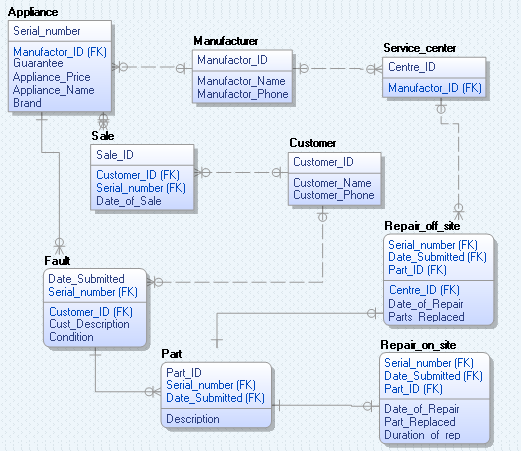
Databases Submission 3

Individual Work

Owen Kane - C13383511

Edmunds Electrics - Group B

Given the Feedback from the the lab here is our current iteration of the ERD diagram.



We removed Fault\_ID from the fault table, we replaced it with Date\_submitted as a primary key used to track a fault. We also added a new table called part to which it has a one to one relationship to both repair\_off\_site and repair\_on\_site as it previously made separate appearances in both tables.

We also removed “stock” from the appliance table as this was not a realistic way to track the stock of an item, instead we will use a count function to count the times the appliance name appears in the appliance table to get the stock of a appliance.

PL / SQL:

SET SERVEROUTPUT ON

DECLARE

useName APPLIANCE.APPLIANCE\_NAME%TYPE:='&Enter\_Appliance\_Name';

useNum APPLIANCE.SERIAL\_NUMBER%TYPE:=&Enter\_Appliance\_Serial\_Num;

useMan APPLIANCE.MANUFACTOR\_ID%TYPE;

useGuar APPLIANCE.GUARANTEE%TYPE;

usePrice APPLIANCE.APPLIANCE\_PRICE%TYPE;

useBrand APPLIANCE.BRAND%TYPE;

BEGIN

IF stock\_exists(useName) THEN

dbms\_output.put\_line('Appliance exists');

get\_Appliance(useName, useMan, useGuar, usePrice, useBrand);

add\_Appliance(useNum, useMAN, useGuar, usePrice, useName, useBrand);

ELSE

dbms\_output.put\_line('Appliance does not exists');

END IF;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line('Error occurred '||SQLCODE||SQLERRM);

END;

------------

CREATE OR REPLACE FUNCTION stock\_exists

(userName APPLIANCE.APPLIANCE\_NAME%type) RETURN BOOLEAN

IS

AppName APPLIANCE.APPLIANCE\_NAME%TYPE;

BEGIN

SELECT APPLIANCE.APPLIANCE\_NAME INTO AppName FROM APPLIANCE

WHERE APPLIANCE.APPLIANCE\_NAME LIKE userName;

RETURN TRUE;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END stock\_exists;

---------------

CREATE OR REPLACE PROCEDURE get\_Appliance(

useName in APPLIANCE.APPLIANCE\_NAME%TYPE,

useMan out APPLIANCE.MANUFACTOR\_ID%TYPE,

useGuar out APPLIANCE.GUARANTEE%TYPE,

usePrice out APPLIANCE.APPLIANCE\_PRICE%TYPE,

useBrand out APPLIANCE.BRAND%TYPE

) AS

BEGIN

SELECT MANUFACTOR\_ID, GUARANTEE, APPLIANCE\_PRICE, BRAND

INTO useMan, useGuar, usePrice, useBrand

FROM APPLIANCE

WHERE APPLIANCE.APPLIANCE\_NAME LIKE useName;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

useMan:=000; useGuar:=000; usePrice:=000; useBrand:='Error';

END get\_Appliance;

-----------

CREATE OR REPLACE PROCEDURE add\_Appliance (

useNum in APPLIANCE.SERIAL\_NUMBER%TYPE,

useMan in APPLIANCE.MANUFACTOR\_ID%TYPE,

useGuar in APPLIANCE.GUARANTEE%TYPE,

usePrice in APPLIANCE.APPLIANCE\_PRICE%TYPE,

useName in APPLIANCE.APPLIANCE\_NAME%TYPE,

useBrand in APPLIANCE.BRAND%TYPE)

AS

BEGIN

insert into APPLIANCE VALUES (useNum, useMAN, useGuar, usePrice, useName, useBrand);

COMMIT;

END add\_Appliance;

------------

CREATE TABLE LOGTABLE

(

tablename VARCHAR2(20),

opname CHAR(3) CHECK (opname IN ('INS', 'UPD', 'DEL')),

Uname VARCHAR2(20),

Sdate DATE,

Keyfield VARCHAR2(20)

);

-----------

CREATE OR REPLACE TRIGGER APPLIANCE\_AI

AFTER INSERT ON APPLIANCE

FOR EACH ROW

BEGIN

INSERT INTO logtable VALUES('APPLIANCE', 'INS', TO\_CHAR(USER), SYSDATE, :new.serial\_number);

END;

-------------