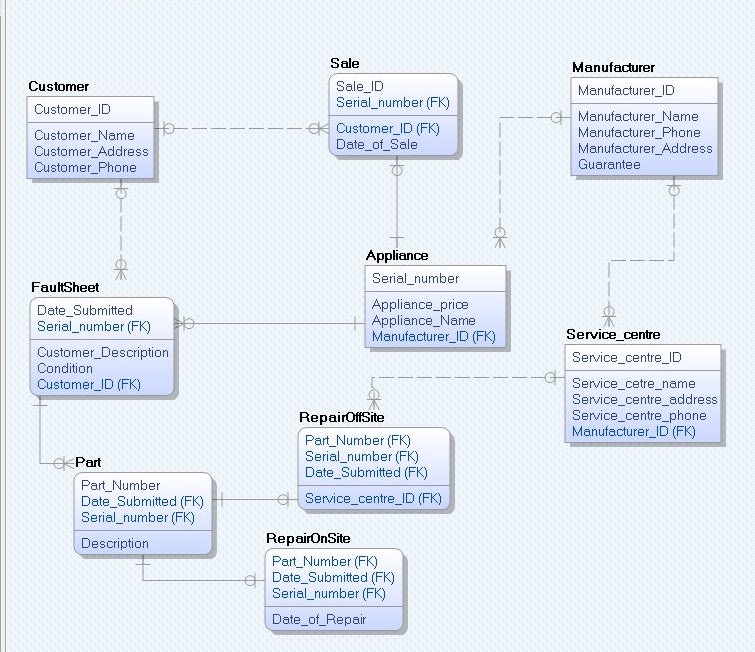
Database Assignment

Student Name: C13432152

Student Name: Jonathan Riordan

The group Erd.

Ed’s Electronic’s

We added an extra table called parts into our erd. This table is used to record the part number, serial number and the date submitted of the appliance. Other changes we made to our erd is in the repair on site table, Made the part number, serial number and date submitted primary keys. Also made changes to the repair on site to the part number, date submitted and serial number. Other changes made in our erd is that we removed stock attribute from the appliance table. We removed this because we made each serial number identical. We can count the stock of a certain product by counting the serial number where the application name equals a certain name such as “iphone 5s” or “Samsung TV”.

For my transaction, I did, the user enters in the serial number and the date submitted, I have a transaction that checks to make sure that the serial number and date entered exist in the fault sheet table, if not an error occurs, if the data entered exists, i check to see if the data exists in the part table, if it does, my procedure will get executed. In my procedure, I use a select statement to join the fault sheet on the part table where the serial number equals the serial number entered in by the user. After that, another select statement is used to join the service centre table to the manufacturer table using the manufacturer id, i then join the manufacturer table to the appliance table using the manufacturer id and return the service centre id.

I then use an if statement to check the condition of a serial number, if the condition is “ON\_NeedsRepair”, The data is inserted into the repair on site table and if the if the condition is “OFF\_NeedsRepair”, the data is entered into the repair off site table.

The tables i have manipulated is repair in site and repair of site depending of the condition. An insert will be applied to either one of these tables if the if statement is true, if not, an exception will occur.

**Anonymous Block:**

set SERVEROUTPUT on;

DECLARE

serial faultsheet.Serial\_number%type := &Enter\_in\_Serial\_Number;

BEGIN

if CHECKSERIALNUMBER(serial,'&date\_entered') THEN

dbms\_output.put\_line('Serial Number exists');

if CHECKPARTEXIST(serial,'&date\_entered') then

ADDPART(serial);

ELSE

dbms\_output.put\_line('No data found in the part table');

END IF;

ELSE

dbms\_output.put\_line('Serial Number and date do not exist');

END IF;

EXCEPTION

WHEN OTHERS THEN

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

END;

**Procedure:**

create or replace PROCEDURE ADDPART(serial in faultsheet.Serial\_number%type) is

serial\_num faultsheet.Serial\_number%type;

sNumber Part.Serial\_number%type;

date\_entered Part.Date\_Submitted%type;

cond faultsheet.Condition%type;

Service\_number Service\_centre.Service\_centre\_ID%type;

part\_num Part.Part\_Number%type;

BEGIN

select faultsheet.Serial\_number, faultsheet.Date\_Submitted, faultsheet.Condition, Part.Part\_Number into sNumber, date\_entered, cond,part\_num from faultsheet

join Part on faultsheet.Serial\_number = Part.Serial\_number where faultsheet.Serial\_number = serial;

select Service\_centre\_ID into Service\_number from Service\_centre join Manufacturer using(Manufacturer\_ID)

join Appliance using(Manufacturer\_ID) where Appliance.Serial\_number = serial;

IF(cond = 'ON\_NeedsRepair') then

DBMS\_OUTPUT.PUT\_LINE('insert to onsite');

Insert into RepairOnSite values(SYSDATE,part\_num,date\_entered,serial);

commit;

ELSE IF(cond = 'OFF\_NeedsRepair') then

DBMS\_OUTPUT.PUT\_LINE('insert to off site');

Insert into RepairOffSite values(part\_num,Service\_number,date\_entered,sNumber);

commit;

END IF;

END IF;

END ADDPART;

**My first function**

create or replace FUNCTION CHECKSERIALNUMBER

(sNumber Appliance.Serial\_number%type, date\_passed faultsheet.Date\_Submitted%type) RETURN BOOLEAN

IS

aname Appliance.Appliance\_Name%TYPE;

BEGIN

SELECT Appliance\_Name INTO aname FROM Appliance join faultsheet using(Serial\_number)

WHERE Serial\_number = sNumber and faultsheet.Date\_Submitted = date\_passed;

RETURN TRUE;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END CHECKSERIALNUMBER;

**My second function**

create or replace FUNCTION CHECKPARTEXIST(sNumber faultsheet.Serial\_number%type,date\_passed faultsheet.Date\_Submitted%type)

RETURN BOOLEAN IS

con faultsheet.Condition%type;

BEGIN

select condition into con from faultsheet join Part on faultsheet.SERIAL\_NUMBER = Part.Serial\_number where faultsheet.Serial\_number = sNumber;

return true;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END CHECKPARTEXIST;

**My first trigger**

create or replace TRIGGER REPAIRONSITE\_AI

AFTER INSERT ON RepairOnSite

FOR EACH ROW

BEGIN

INSERT INTO logtable VALUES

('RepairOnSite','INS',TO\_CHAR(USER),SYSDATE, :new.serial\_number);

DBMS\_OUTPUT.PUT\_LINE('Appliance been repaired on site');

END;

**My second trigger**

create or replace TRIGGER REPAIROFFSITE\_AI

AFTER INSERT ON RepairOffSite

FOR EACH ROW

BEGIN

INSERT INTO logtable VALUES

('RepairOffSite','INS',TO\_CHAR(USER),SYSDATE, :new.serial\_number);

DBMS\_OUTPUT.PUT\_LINE('Appliance sent off site to be repaired');

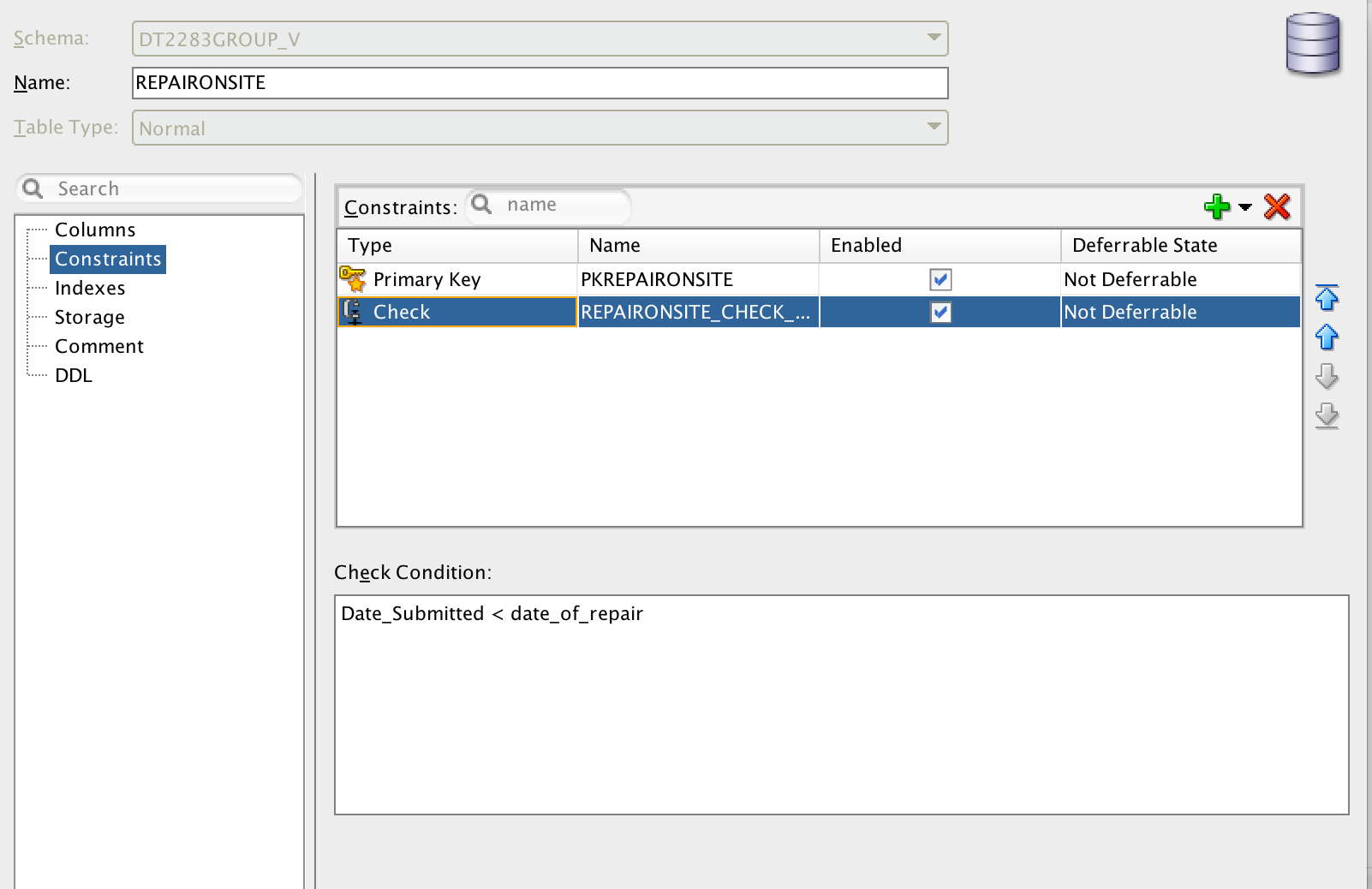
END;

For my constraint trigger, i put a constraint on the table. I put a constraint on the repair on site table. When the information is being entered into the table, the date submitted must be before the date of repair, if not, an error will occur when entering in data.

The following sql will cause an error while entering data into the table.

-- This will run the contraint causing an error because the date of repair ws before date submitted causing an error

INSERT INTO RepairOnSite (Part\_Number,Date\_Submitted,Serial\_number, Date\_of\_Repair)

VALUES (232324, '28-DEC-2015', 76548, ’27-DEC-2015’);

My sql Query's

— Selection and Projection

-- select all the appliance where the manufacturer equals a cerain ID.

select \* from appliance where Manufacturer\_ID = 3;

— Intersection

-- shows the serials numbers from the appliance table that have been sold and are in the sale table

select Serial\_number from appliance intersect select Serial\_number from Sale;

-- inner join

-- selects the manufacturer name and the serivce centre associated with that manufacturer using the manufacturer id.

select Manufacturer.Manufacturer\_ID, Manufacturer.Manufacturer\_Name, Service\_centre.Service\_cetre\_name from Manufacturer inner join

Service\_centre on Manufacturer.Manufacturer\_ID = Service\_centre.Manufacturer\_ID;

-- full outer join

-- select the customer names form the customer table and joins the sale table based on the customer id.

select Customer.Customer\_name, Sale.Date\_of\_sale from Customer full outer join sale on customer.customer\_id = sale.customer\_id;

-- Aggregation

-- sql to select the serial number where the application name equals sont tv, this is used to count the stock of a produck in the appliance table.

select count(serial\_number) from appliance where appliance\_name = 'Sony\_TV';

-- SubQueries

-- select all the maufacturer id that are less than 3.

select manufacturer\_id from Manufacturer where manufacturer\_id in (select manufacturer\_id from Manufacturer where Manufacturer\_id < 3);

-- Union

-- returns the partn numbern the serial number of the appliance and the date submitted.

select part\_number,serial\_number,date\_submitted from part union all

select part\_number,serial\_number,date\_submitted from REPAIRONSITE;

-- Difference

-- returns the serial number from the appliance table that are not in the fault sheet

select serial\_number from appliance minus select serial\_number from faultsheet;