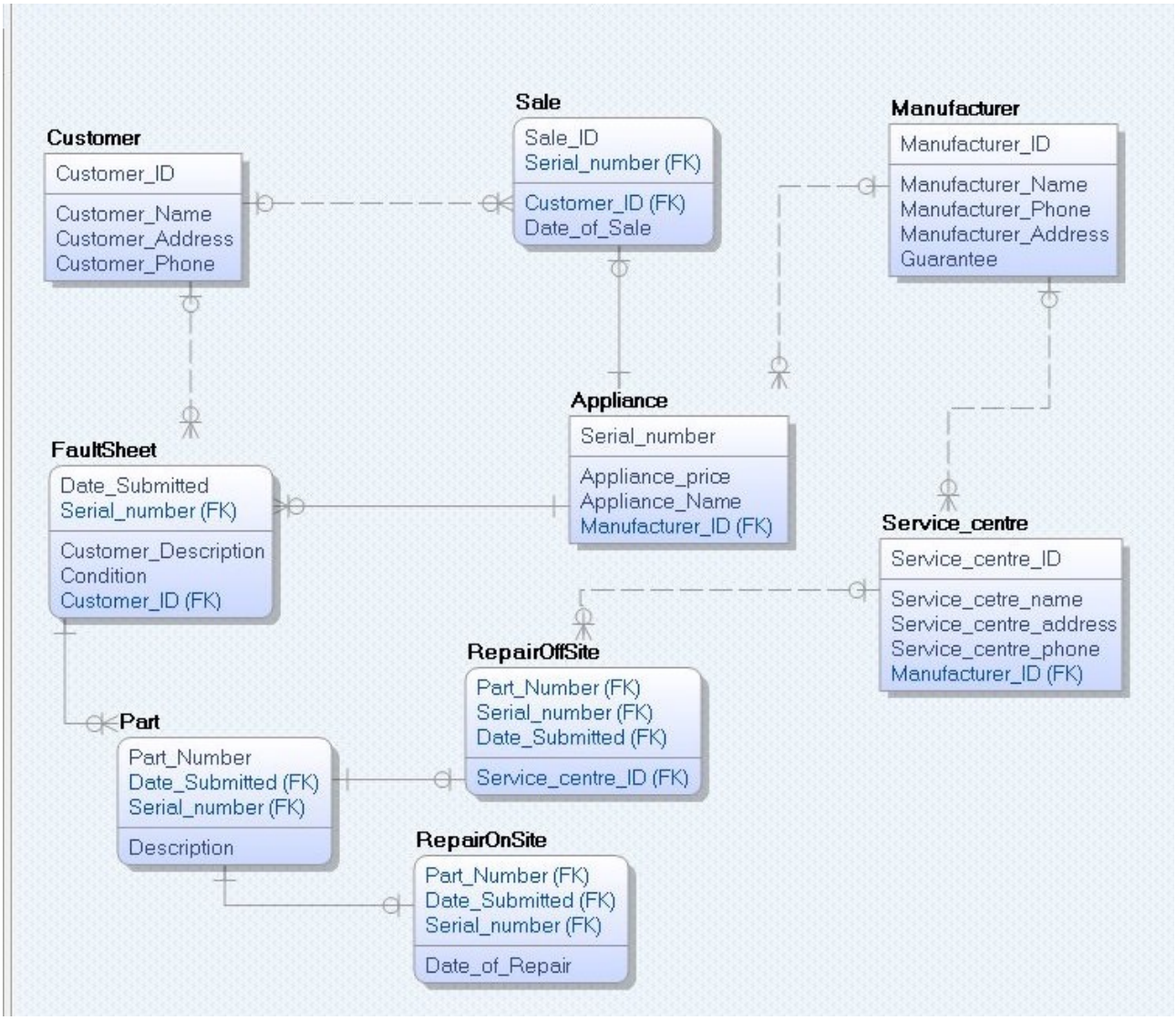
Database Assignment
Student Name: C13432152
Student Name: Jonathan Riordan

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;
```

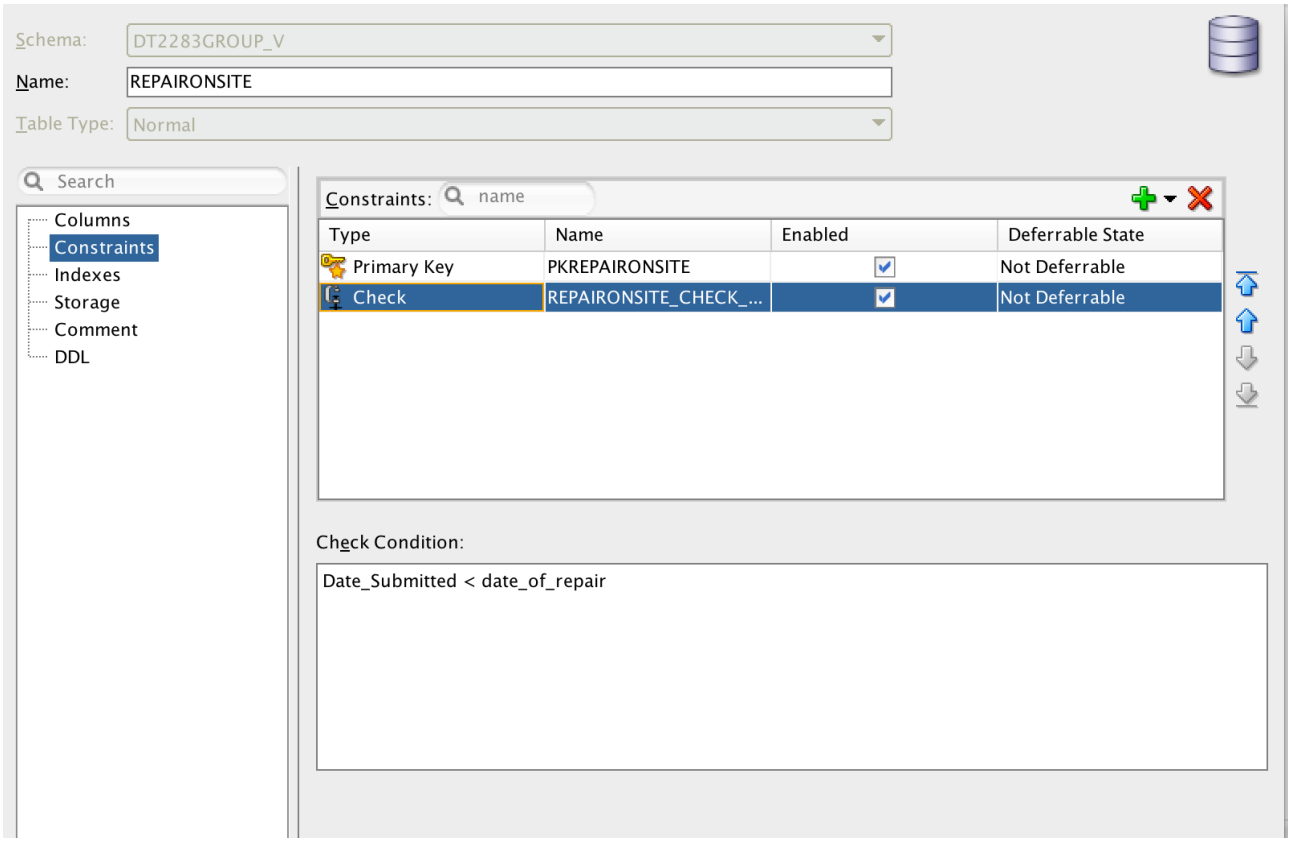
My second trigger

```
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;
```