DBMS PROJECT REPORT

ON

Hostel Food Wastage Management System

by

Sehaj Verma (101703497)

Sehaj Pal Singh (101703498)

Saloni Malhotra (101703473)

Submitted To

Dr. Manisha Kaushal

Acknowledgements

We would like to express our deepest appreciation to Dr. Manisha Kaushal without whom every DBMS class would not be something to look forward to. She has always been there as a mentor and pillar of support throughout the making of the project without her guidance and persistent help, this dissertation would not have been possible. We are highly indebted for her guidance and constant supervision as well as for providing necessary information regarding the report.

In view of that, we worked as a team; each playing instrumental role with utmost enthusiasm and applied our knowledge and understanding of the different factors and elements related to schooling and education.

We also wish to express our gratitude to the officials and other staff members of Thapar Institute of Engineering and Technology who extended their help during the period of our project.

INDEX

REQUIREMENT ANALYSIS

Background of the area

Hostel Messes in various institutions serve food to the students. Though the food is made according to an estimation of the previous day. Most of the time food is not finished and the leftover food gets wasted. This food wastage cannot be used for the next day as the processed food starts to perish. Food wastage is massive as every institution have an average of 5 hostels.

Need of the Project

The Hostel Wastage Management project is required to minimize the food wastage and make everybody realise the wastage of food they do every day.

Objectives of the project

This project aims to minimize the food wastage by displaying the leftover food at the online portal for the needy to see it and collect the food within a certain time period.

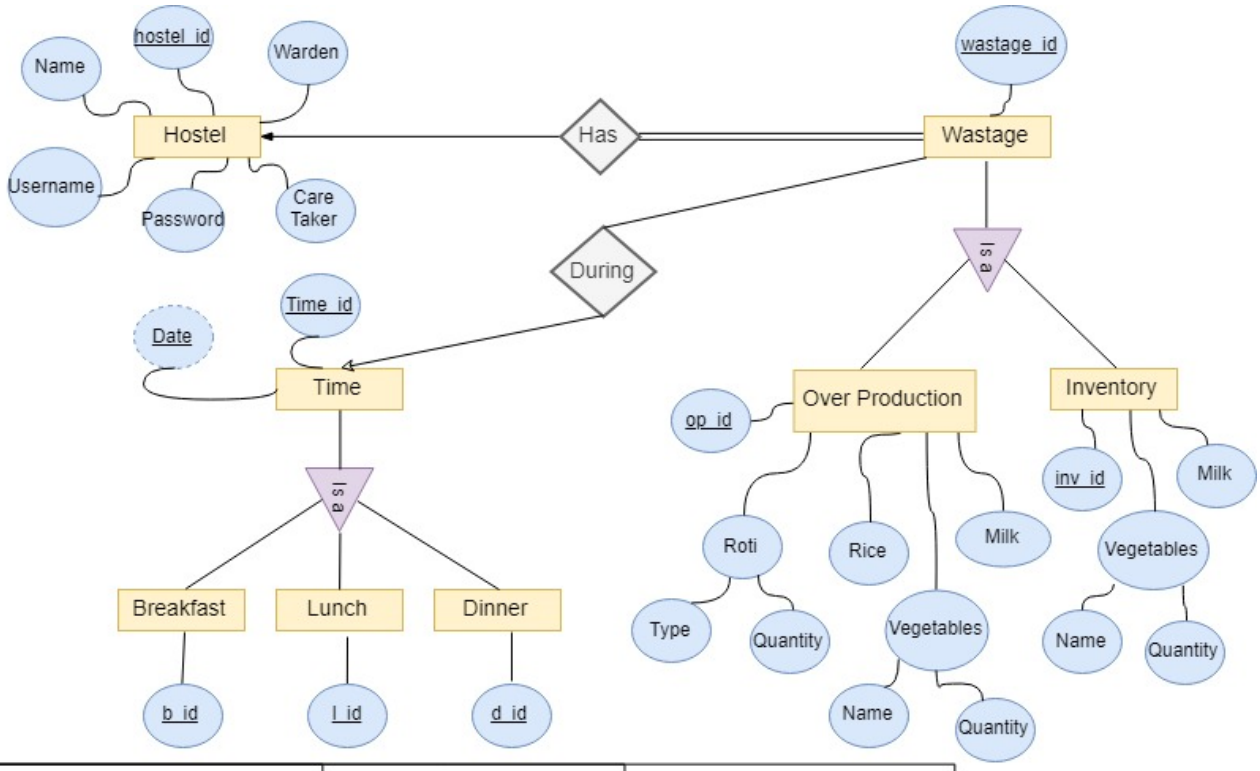
Project Outcomes

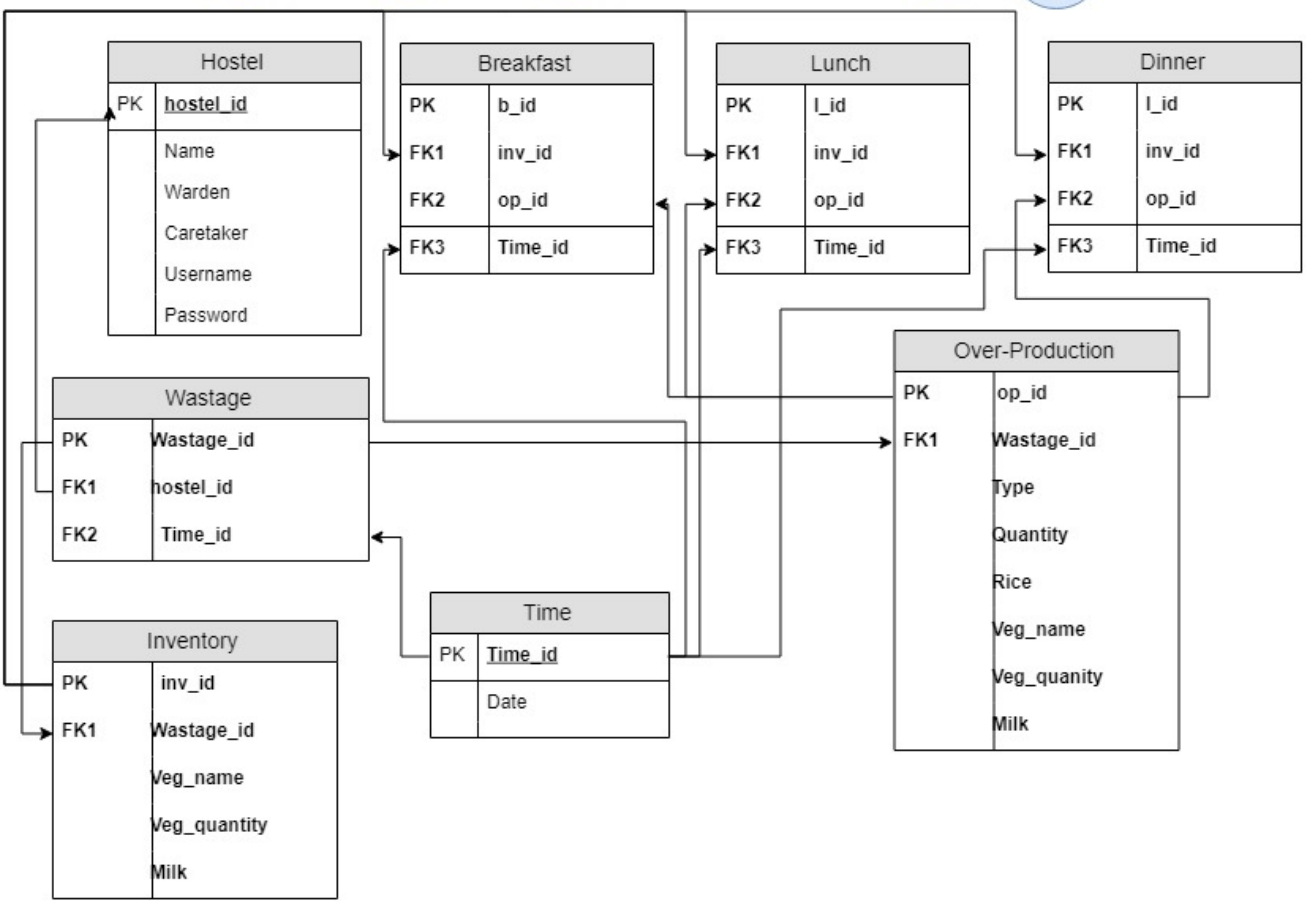
This project will minimize the food wastage to some extent and give an idea about the scarcity of food.

About the features/functionalities

In order to create this particular database management system, the software used was sqlplus which offers great inbuilt features such as cursors, functions , procedures, indexes ,sequences and other various inbuilt features which are used to insert, update, delete and view data which helped us to the create the database management system in a very efficient manner.

ER DIAGRAM AND TABLES





TABLES AND ATTRIBUTES

TABLE NAME	ATTRIBUTES
HOSTEL	Hostel_id(Primary key), Name, Warden, Caretaker, Username, Password
BREAKFAST	B_id(Primary Key), inv_id (FK1), op_id(FK2), Time_id(FK3)
LUNCH	B_id(Primary Key), inv_id (FK1), op_id(FK2), Time_id(FK3)
DINNER	B_id(Primary Key), inv_id (FK1), op_id(FK2), Time_id(FK3)
WASTAGE	Wastage_id (Primary Key) Hostel_id(FK) Time_id(FK2)
INVNTORY	Inv_id, wastage_id, ,Veg_name, ,Veg_quantity, Milk
TIME	Time_id(Primary Key), Date
OVER-PRODUCTION	Op_id (Primary Key), Wastge_id(FK1), Type, Quantity, Rice, Veg_name, Veg_quantity, Milk

NORMALIZATION

The tables created are already in the normalized form.

PL/SQL CODE

TABLES

create table Hostel(hostel_id number(10),Name varchar(20),Warden varchar(20),Caretaker varchar(20),primary key(hostel_id));

create table Breakfast(b_id number(10), inv_id number(10), op_id number(10), Time_id number(10),primary key(b_id), foreign key(inv_id) references Inventory, foreign key(op_id) references OverProduction, foreign key(Time_id) references Time);

create table Lunch(l_id number(10), inv_id number(10), op_id number(10), Time_id number(10),primary key(l_id), foreign key(inv_id) references Inventory, foreign key(op_id) references OverProduction, foreign key(Time_id) references Time);

create table Dinner(d_id number(10), inv_id number(10), op_id number(10), Time_id number(10),primary key(d_id), foreign key(inv_id) references Inventory, foreign key(op_id) references OverProduction, foreign key(Time_id) references Time);

create table Wastage(Wastage_id number(10), hostel_id number(10), Time_id number(10),primary key(Wastage_id), foreign key(hostel_id) references Hostel, foreign key(Time_id) references Time);

create table Time(Time_id number(10), date Date, primary key(Time_id));

create table Inventory(inv_id number(10), Wastage_id number(10), Veg_name varchar(10),Veg_quantity number(10), milk number(10), primary key(inv_id), foreign key(Wastage_id) references Wastage);

create table OverProduction(op_id number(10), Wastage_id number(10),Type varchar(10), Quantity number(10), Rice number(10), Veg_name varchar(10),Veg_quantity number(10), milk number(10), primary key(op_id), foreign key(Wastage_id) references Wastage);

PL/SQL CODES

INSERT

Procedures and Functions

```
create or replace procedure deleteHostel()
        begin
                dbms_output.put_line(" Enter the Hostel ID to be deleted");
                p1:=&p1;
                delete from Hostel where hostel_id=p1;
        end;
create or replace procedure deleteInv()
                begin
                        dbms_output.put_line(' Enter the Inventory ID to be deleted');
                        p2:=&p2;
                        delete from Inventory where inv_id=p2;
                end;
create or replace procedure deleteOp()
                begin
                        dbms_output.put_line(" Enter the OverProduction ID to be deleted");
                        p3:=&p3;
                        delete from Inventory where op_id=p3;
                end;
```

```
create or replace procedure displayOp()
  declare
  opid OverProduction.op_id%type;
  wastageid OverProduction.Wastage_id%type;
  typeo OverProduction.Type%type;
  Quantity OverProduction.Quantity%type;
  Rice OverProduction.Rice%type;
  Veg_name2 Inventory.Veg_name%type;
  Veg_quantity2 Inventory.Veg_quantity%type;
  Milk2 Inventory.Milk%type;
  CURSOR c_op is
      SELECT op_id, Wastage_id,Type, Quantity,Rice,Veg_name,Veg_quantity,Milk FROM
OverProduction;
          begin
                OPEN c_op;
          LOOP
          FETCH c_op into opid, wastageid, typeo,Quantity;
                        EXIT WHEN c_op%notfound;
                        dbms_output.put_line(opid || '  ' || wastageid || '  ' || typeo || '  ' ||
Quantity|| '  ' || Rice|| '  ' || Veg_name2|| '  ' || Veg_quantity2|| '  ' || Milk2);
          END LOOP;
          CLOSE c_op;
         end;
```

```
create or replace function displayInv()
        begin
                OPEN c_inv;
                 LOOP
                 FETCH c_inv into invid, wastageid, Veg_name,Veg_quantity,Milk;
                EXIT WHEN c_inv%notfound;
                dbms_output.put_line(wastageid || '   ' || wastageid || '   ' || Veg_name || '   ' ||
Veg_quantity|| '   ' || Milk);
                 END LOOP;
                 CLOSE c_inv;
        end;
create function displayHostel()
  DECLARE
        begin
        OPEN h_hostel;
         LOOP
         FETCH h_hostel into hostelid, name, warden,caretaker;
        EXIT WHEN h_hostel%notfound;
        dbms_output.put_line(hostelid || '   ' || name || '   ' || warden || '   ' || caretaker);
         END LOOP;
         CLOSE h_hostel;
        end;
```

SHOW (Cursor)

```
CURSOR h_hostel is
    SELECT hostel_id, name, warden,caretaker FROM hostel;
CURSOR c_inv is
    SELECT inv_id, Wastage_id, Veg_name,Veg_quantity,Milk FROM Inventory;
CURSOR c_op is
    SELECT op_id, Wastage_id,Type, Quantity,Rice,Veg_name,Veg_quantity,Milk FROM
OverProduction;
```

TRIGGER

```
create or replace trigger caretaker_warden
before insert or update of Warden,Caretaker on Hostel
for each row
begin
if :NEW.Warden=:NEW.Caretaker then
  RAISE_APPLICATION_ERROR(-20004, 'A person cannot be caretaker and warden at
same time');
end if; end;
/
create or replace trigger timetrigger
before insert on Wastage
begin
insert into Time
(Time_id,cdate) values(time_seq.nextval,sysdate);
```

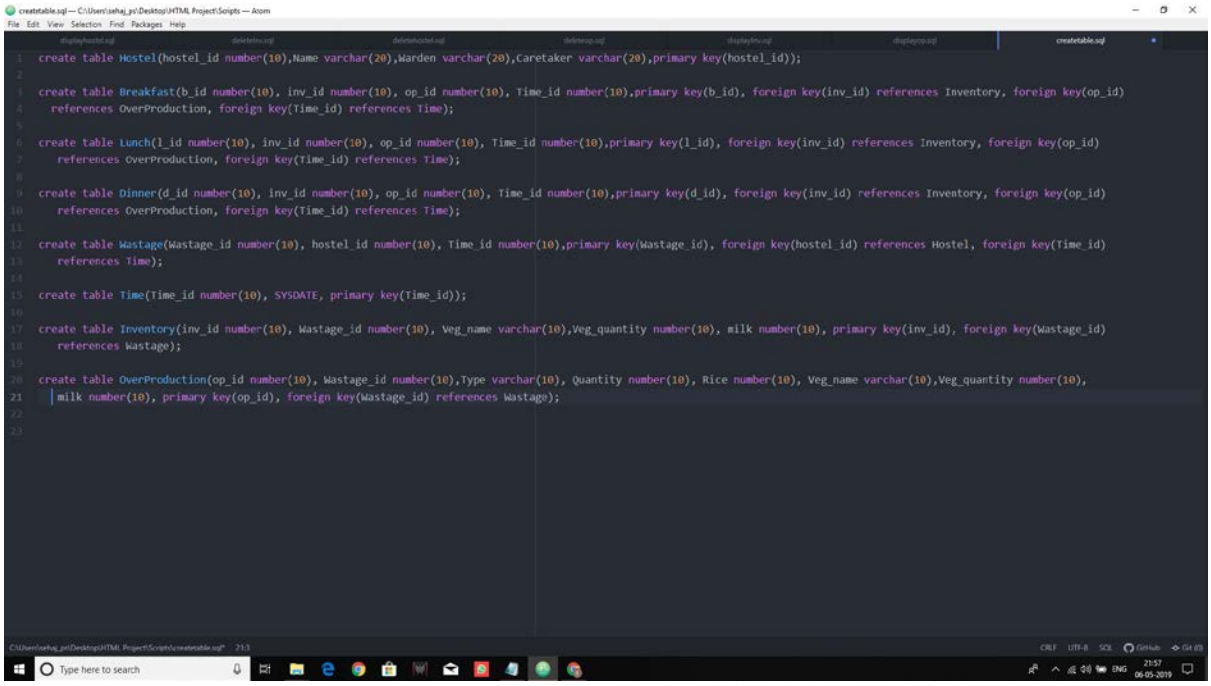
```
end;
/
CREATE OR REPLACE TRIGGER FOREIGN_KEY
BEFORE INSERT OR UPDATE OF hostel_id ON WASTAGE
FOR EACH ROW
 DECLARE DNO WASTAGE.hostel_id%TYPE;
 BEGIN
 SELECT hostel_id INTO DNO FROM Hostel WHERE hostel_id=:NEW.hostel_id;
 NULL;
 EXCEPTION
 WHEN NO_DATA_FOUND
 THEN RAISE_APPLICATION_ERROR(-20004, 'FOREIGN KEY VIOLATED BECAUSE
VALUE IS NOT FOUND IN THE PARENT TABLE');
 END;
/
```

SEQUENCE

```
create sequence time_seq
start with 1
maxvalue 100
minvalue 1
cycle
cache 20
/
```

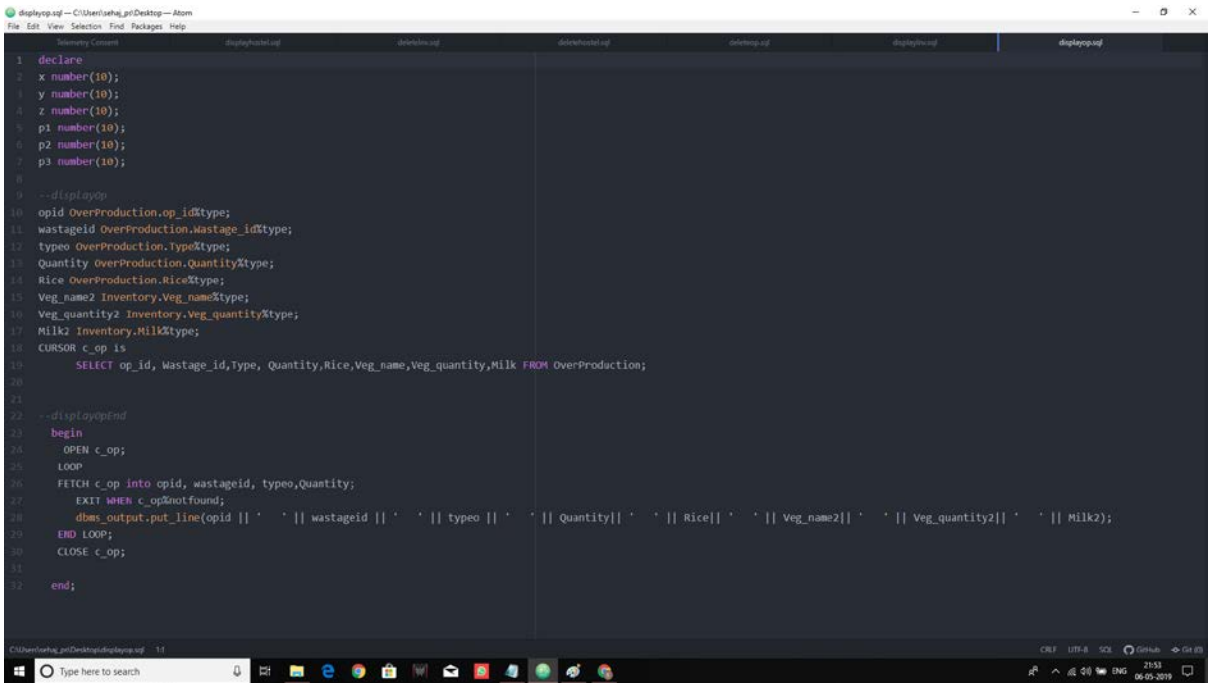
Snapshots

Create Table

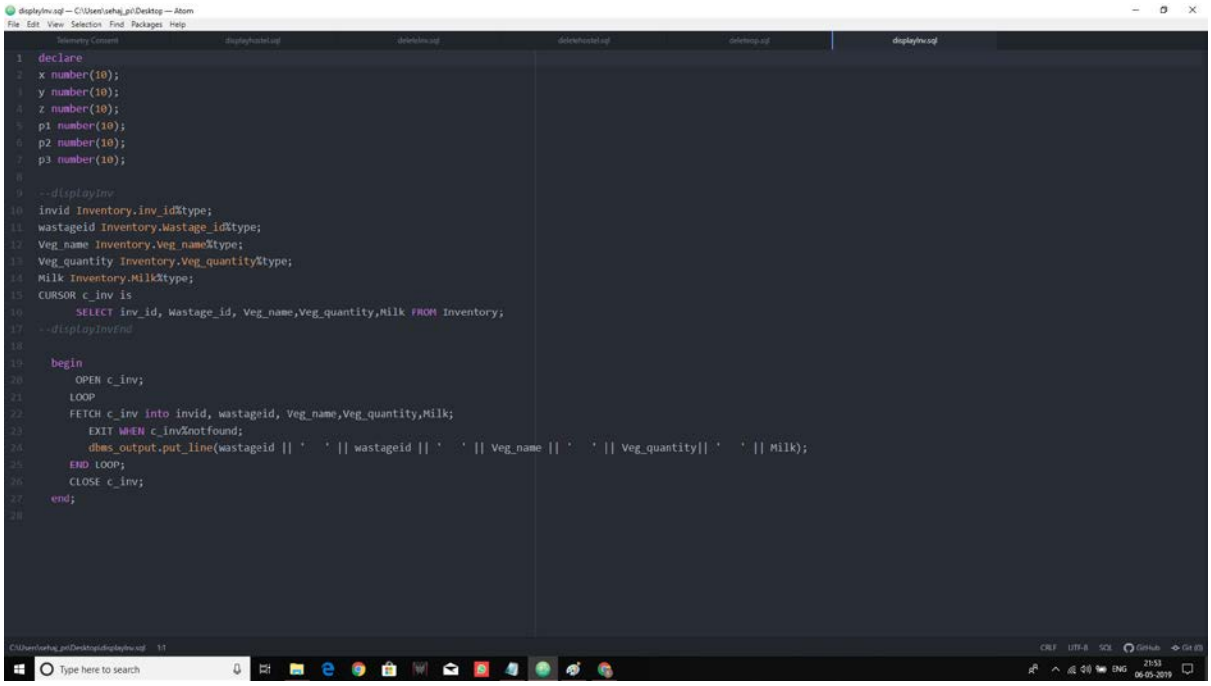


```
create table Hostel(hostel_id number(10),Name varchar(20),Warden varchar(20),Caretaker varchar(20),primary key(hostel_id));

create table Breakfast(b_id number(10), inv_id number(10), op_id number(10), Time_id number(10),primary key(b_id), foreign key(inv_id) references Inventory, foreign key(op_id)
  references OverProduction, foreign key(Time_id) references Time);

create table Lunch(l_id number(10), inv_id number(10), op_id number(10), Time_id number(10),primary key(l_id), foreign key(inv_id) references Inventory, foreign key(op_id)
   references OverProduction, foreign key(Time_id) references Time);

create table Dinner(d_id number(10), inv_id number(10), op_id number(10), Time_id number(10),primary key(d_id), foreign key(inv_id) references Inventory, foreign key(op_id)
   references OverProduction, foreign key(Time_id) references Time);

create table Wastage(Wastage_id number(10), hostel_id number(10), Time_id number(10),primary key(Wastage_id), foreign key(hostel_id) references Hostel, foreign key(Time_id)
   references Time);

create table Time(Time_id number(10), SYSDATE, primary key(Time_id));

create table Inventory(inv_id number(10), Wastage_id number(10), Veg_name varchar(10),Veg_quantity number(10), milk number(10), primary key(inv_id), foreign key(Wastage_id)
   references Wastage);

create table OverProduction(op_id number(10), Wastage_id number(10),Type varchar(10), Quantity number(10), Rice number(10), Veg_name varchar(10),Veg_quantity number(10),
  milk number(10), primary key(op_id), foreign key(Wastage_id) references Wastage);
```

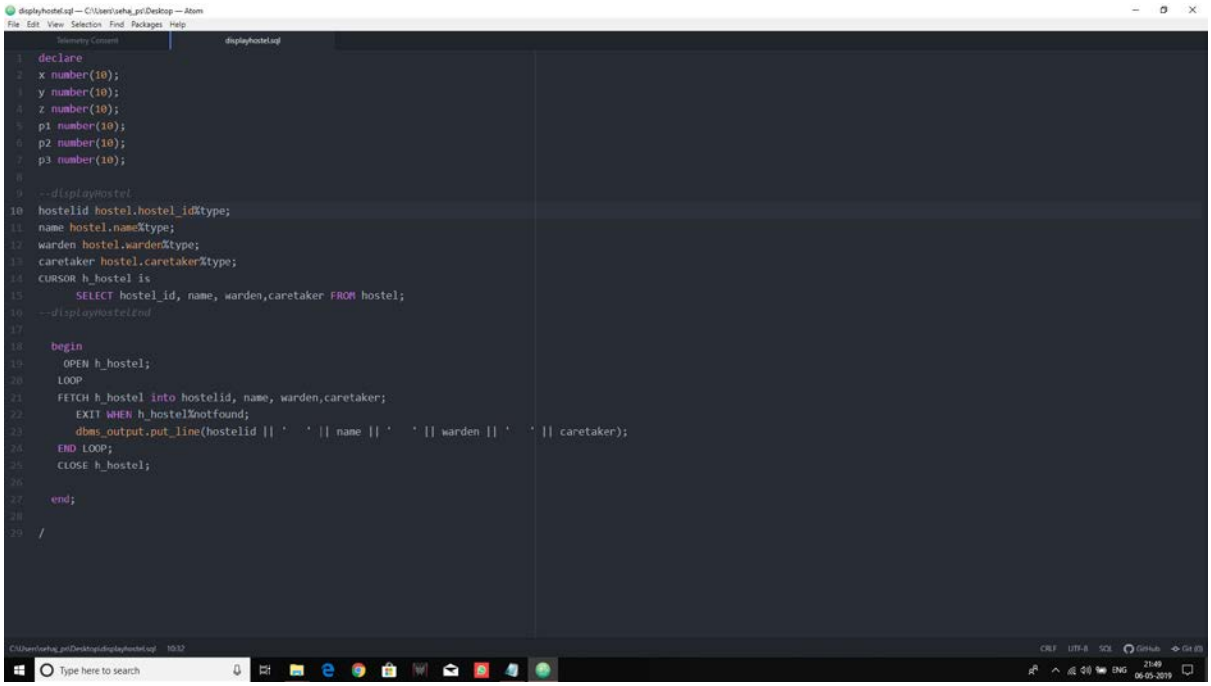
Display



```
declare
x number(10);
y number(10);
z number(10);
p1 number(10);
p2 number(10);
p3 number(10);

--displayOp
opid OverProduction.op_id%type;
wastageid OverProduction.Wastage_id%type;
typeo OverProduction.Type%type;
Quantity OverProduction.Quantity%type;
Rice OverProduction.Rice%type;
Veg_name2 Inventory.Veg_name%type;
Veg_quantity2 Inventory.Veg_quantity%type;
Milk2 Inventory.Milk%type;
CURSOR c_op is
      SELECT op_id, Wastage_id,Type, Quantity,Rice,Veg_name,Veg_quantity,Milk FROM OverProduction;


--displayOpEnd
  begin
    OPEN c_op;
   LOOP
   FETCH c_op into opid, wastageid, typeo,Quantity;
      EXIT WHEN c_op%notfound;
      dbms_output.put_line(opid || '   ' || wastageid || '   ' || typeo || '   ' || Quantity|| '   ' || Rice|| '   ' || Veg_name2|| '   ' || Veg_quantity2|| '   ' || Milk2);
   END LOOP;
   CLOSE c_op;

  end;
```

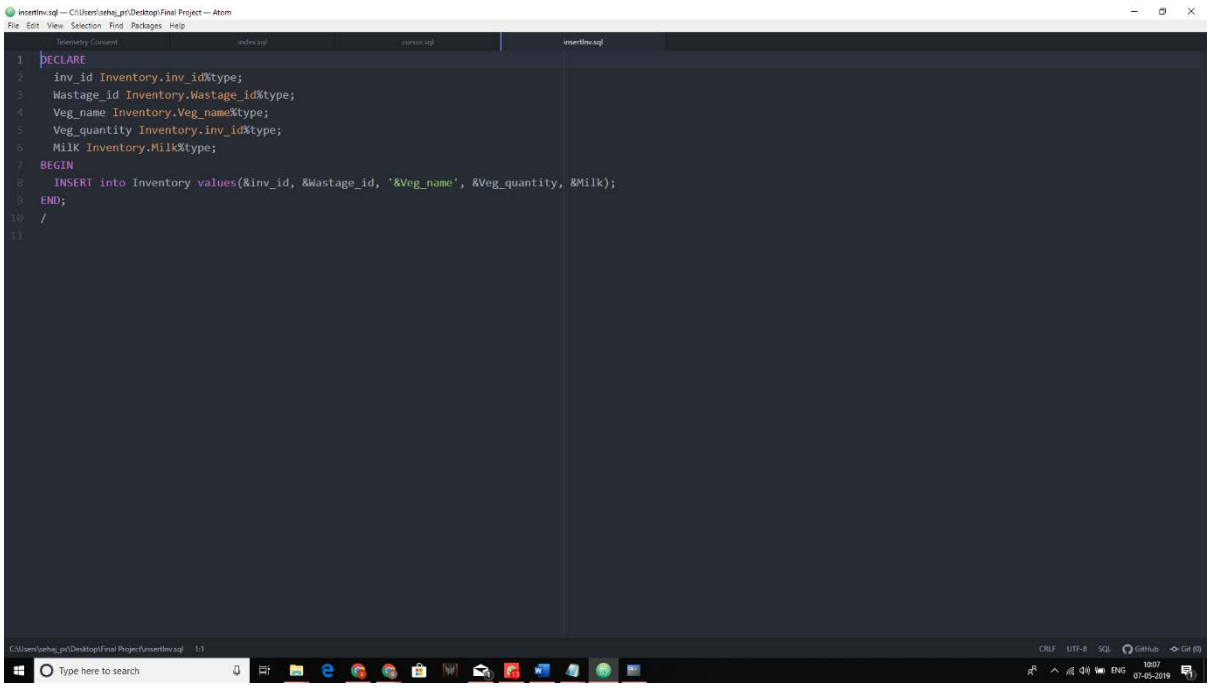


```
declare
x number(10);
y number(10);
z number(10);
p1 number(10);
p2 number(10);
p3 number(10);

--displayInv
invid Inventory.inv_id%type;
wastageid Inventory.Wastage_id%type;
Veg_name Inventory.Veg_name%type;
Veg_quantity Inventory.Veg_quantity%type;
Milk Inventory.Milk%type;
CURSOR c_inv is
      SELECT inv_id, Wastage_id, Veg_name,Veg_quantity,Milk FROM Inventory;
--displayInvEnd

  begin
      OPEN c_inv;
     LOOP
     FETCH c_inv into invid, wastageid, Veg_name,Veg_quantity,Milk;
        EXIT WHEN c_inv%notfound;
        dbms_output.put_line(wastageid || '   ' || wastageid || '   ' || Veg_name || '   ' || Veg_quantity|| '   ' || Milk);
     END LOOP;
     CLOSE c_inv;
  end;
```

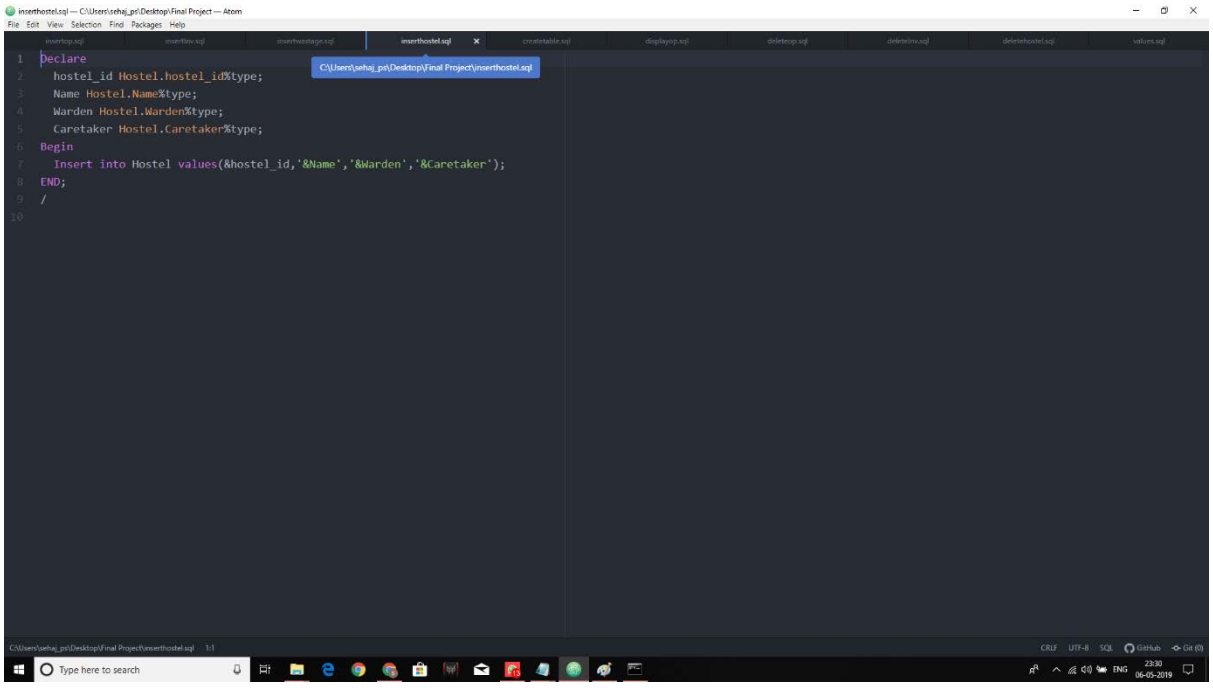


```
declare
x number(10);
y number(10);
z number(10);
p1 number(10);
p2 number(10);
p3 number(10);

--displayHostel
hostelid hostel.hostel_id%type;
name hostel.name%type;
warden hostel.warden%type;
caretaker hostel.caretaker%type;
CURSOR h_hostel is
      SELECT hostel_id, name, warden,caretaker FROM hostel;
--displayHostelEnd

  begin
    OPEN h_hostel;
   LOOP
   FETCH h_hostel into hostelid, name, warden,caretaker;
      EXIT WHEN h_hostel%notfound;
      dbms_output.put_line(hostelid || '   ' || name || '   ' || warden || '   ' || caretaker);
   END LOOP;
   CLOSE h_hostel;

  end;

/
```

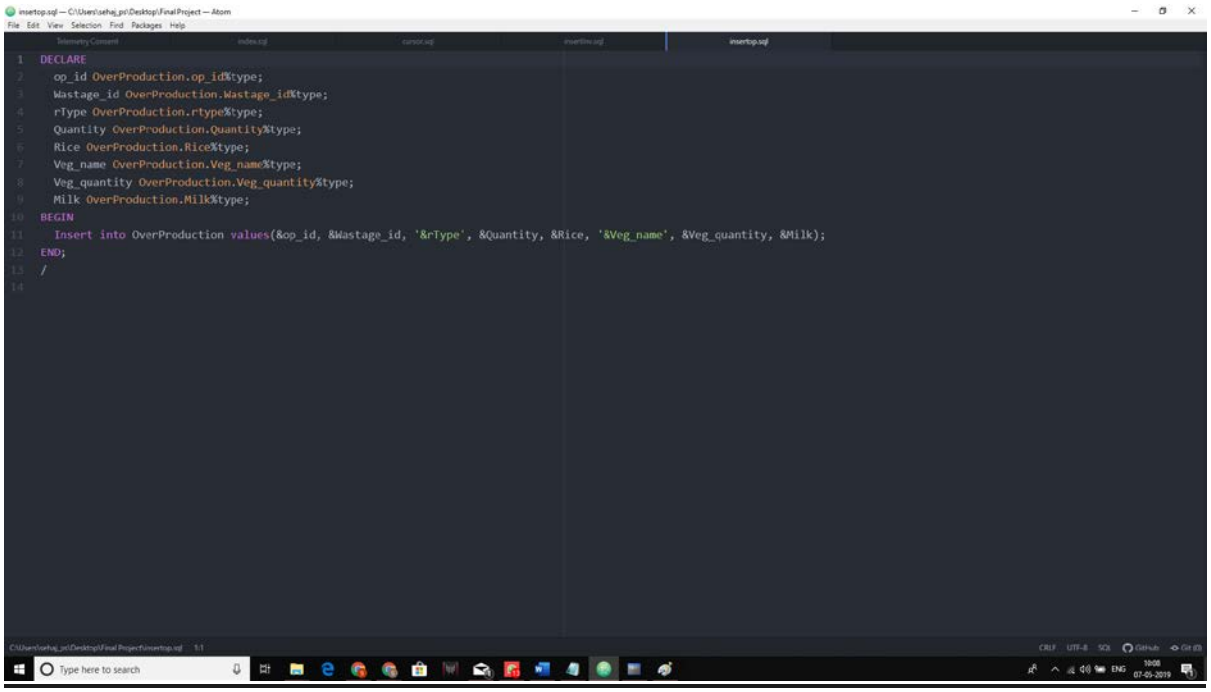
Insertion



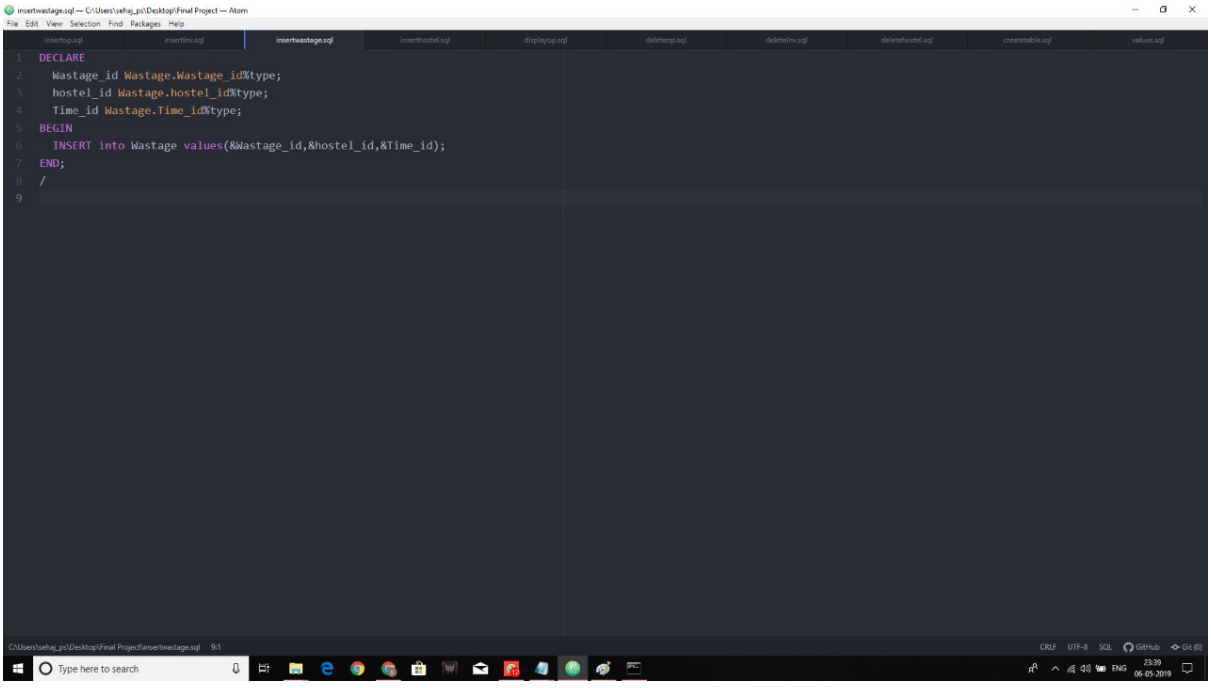
```
DECLARE
  inv_id Inventory.inv_id%type;
  Wastage_id Inventory.Wastage_id%type;
  Veg_name Inventory.Veg_name%type;
  Veg_quantity Inventory.inv_id%type;
  MilK Inventory.Milk%type;
BEGIN
  INSERT into Inventory values(&inv_id, &Wastage_id, '&Veg_name', &Veg_quantity, &Milk);
END;
/
```



```
Declare
  hostel_id Hostel.hostel_id%type;
  Name Hostel.Name%type;
  Warden Hostel.Warden%type;
  Caretaker Hostel.Caretaker%type;
Begin
  Insert into Hostel values(&hostel_id,'&Name','&Warden','&Caretaker');
END;
/
```

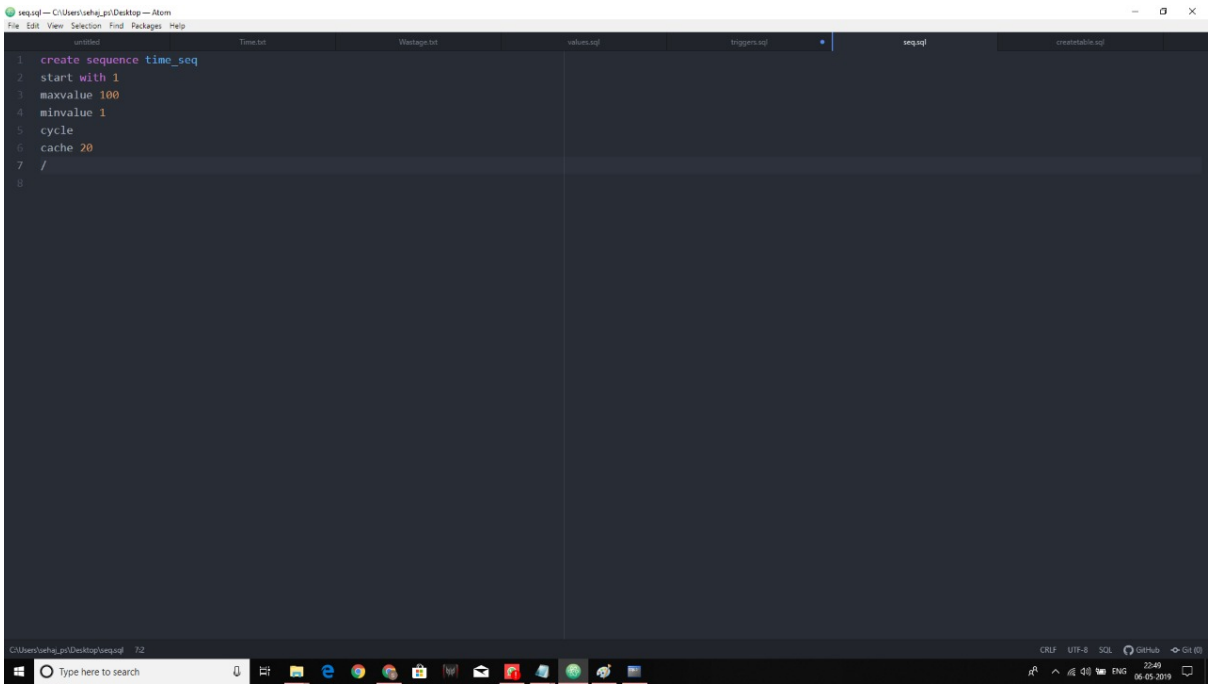


```
DECLARE
  op_id OverProduction.op_id%type;
  Wastage_id OverProduction.Wastage_id%type;
  rType OverProduction.rtype%type;
  Quantity OverProduction.Quantity%type;
  Rice OverProduction.Rice%type;
  Veg_name OverProduction.Veg_name%type;
  Veg_quantity OverProduction.Veg_quantity%type;
  Milk OverProduction.Milk%type;
BEGIN
  Insert into OverProduction values(&op_id, &Wastage_id, '&rType', &Quantity, &Rice, '&Veg_name', &Veg_quantity, &Milk);
END;
/
```



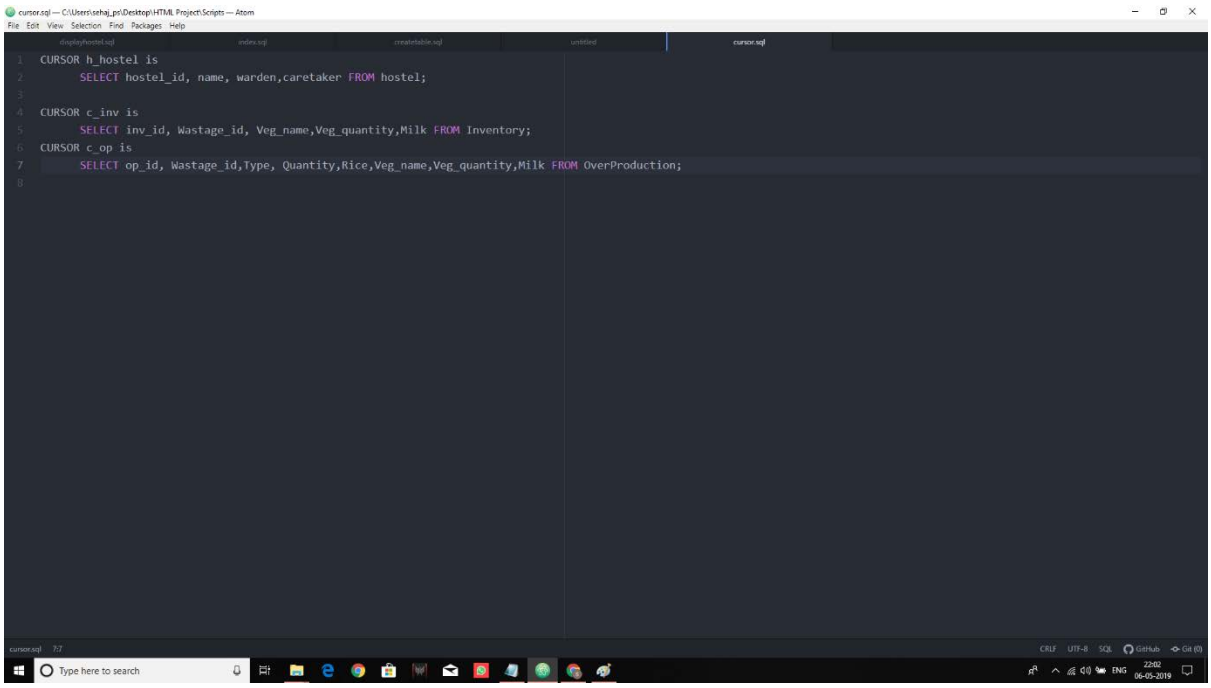
```
DECLARE
  Wastage_id Wastage.Wastage_id%type;
  hostel_id Wastage.hostel_id%type;
  Time_id Wastage.Time_id%type;
BEGIN
  INSERT into Wastage values(&Wastage_id,&hostel_id,&Time_id);
END;
/
```

Sequences



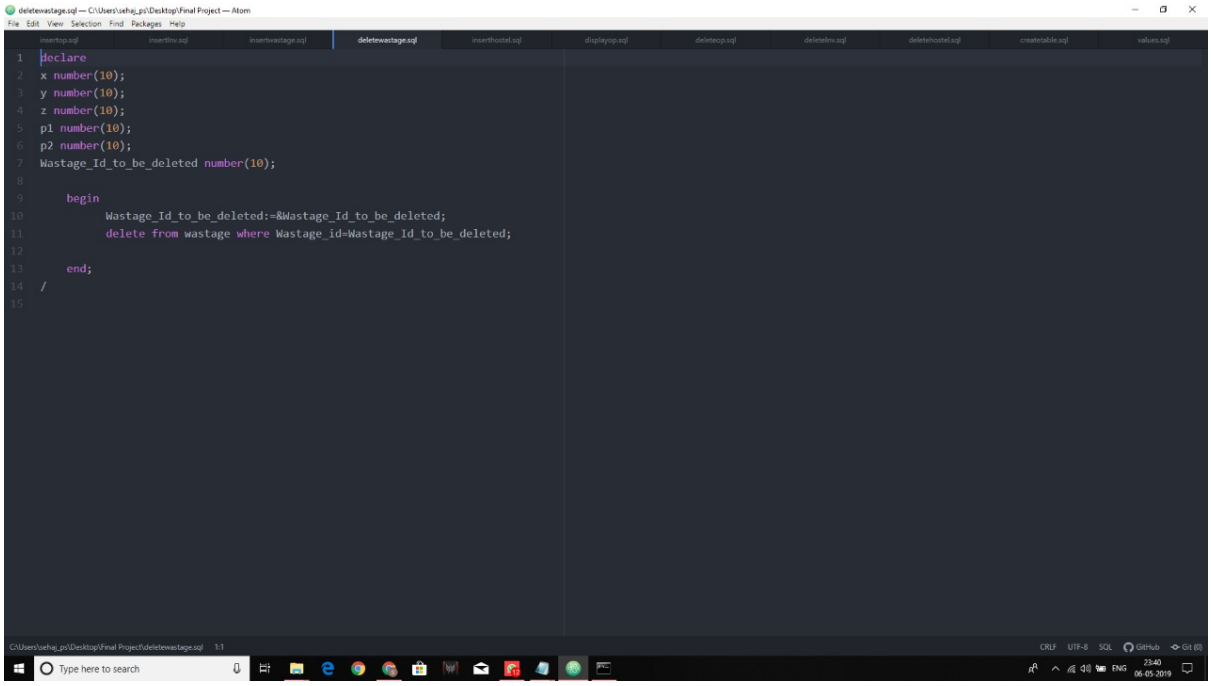
```
create sequence time_seq
start with 1
maxvalue 100
minvalue 1
cycle
cache 20
/
```

Cursors

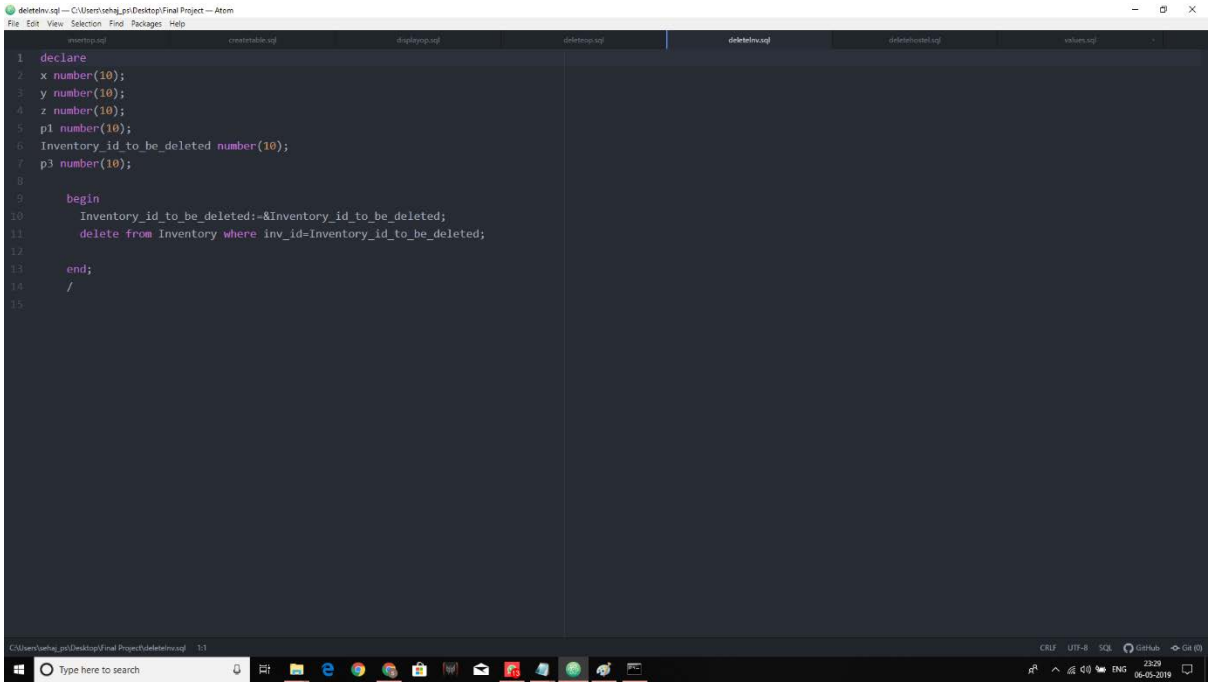


```
CURSOR h_hostel is
      SELECT hostel_id, name, warden,caretaker FROM hostel;

CURSOR c_inv is
      SELECT inv_id, Wastage_id, Veg_name,Veg_quantity,Milk FROM Inventory;
CURSOR c_op is
      SELECT op_id, Wastage_id,Type, Quantity,Rice,Veg_name,Veg_quantity,Milk FROM OverProduction;
```

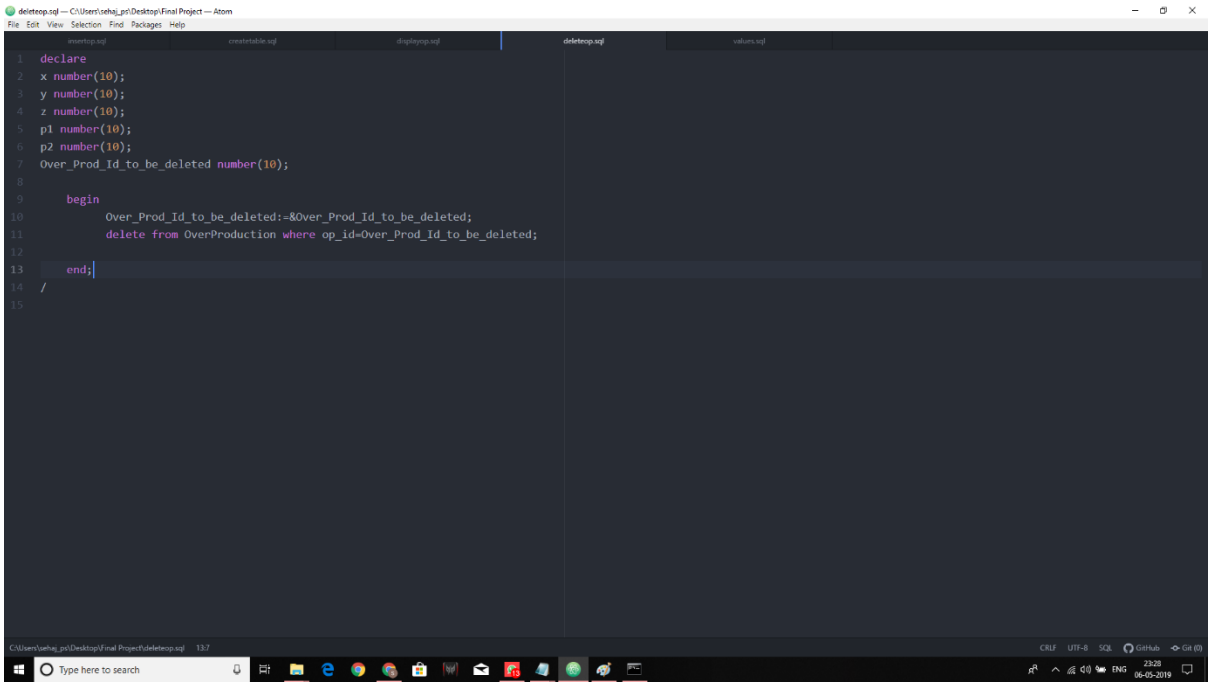
Deletion



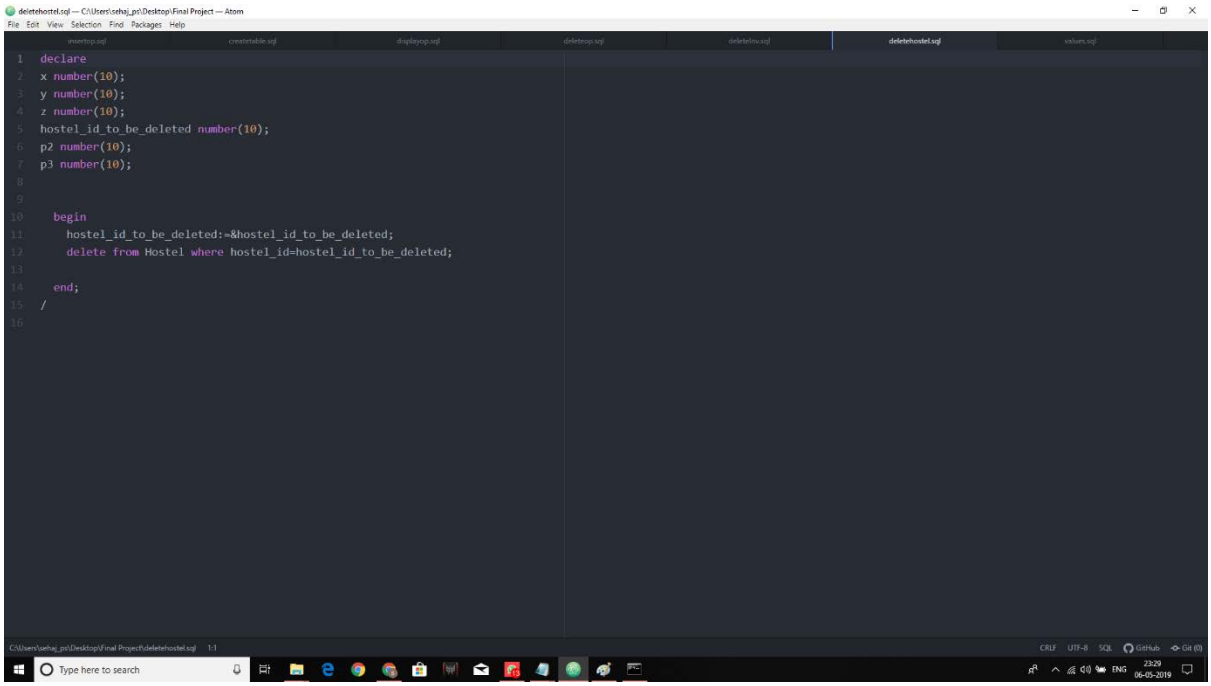
```
declare
x number(10);
y number(10);
z number(10);
p1 number(10);
p2 number(10);
Wastage_Id_to_be_deleted number(10);

    begin
         Wastage_Id_to_be_deleted:=&Wastage_Id_to_be_deleted;
         delete from wastage where Wastage_id=Wastage_Id_to_be_deleted;

    end;
/
```



```
declare
x number(10);
y number(10);
z number(10);
p1 number(10);
Inventory_id_to_be_deleted number(10);
p3 number(10);

    begin
      Inventory_id_to_be_deleted:=&Inventory_id_to_be_deleted;
      delete from Inventory where inv_id=Inventory_id_to_be_deleted;

    end;
    /
```

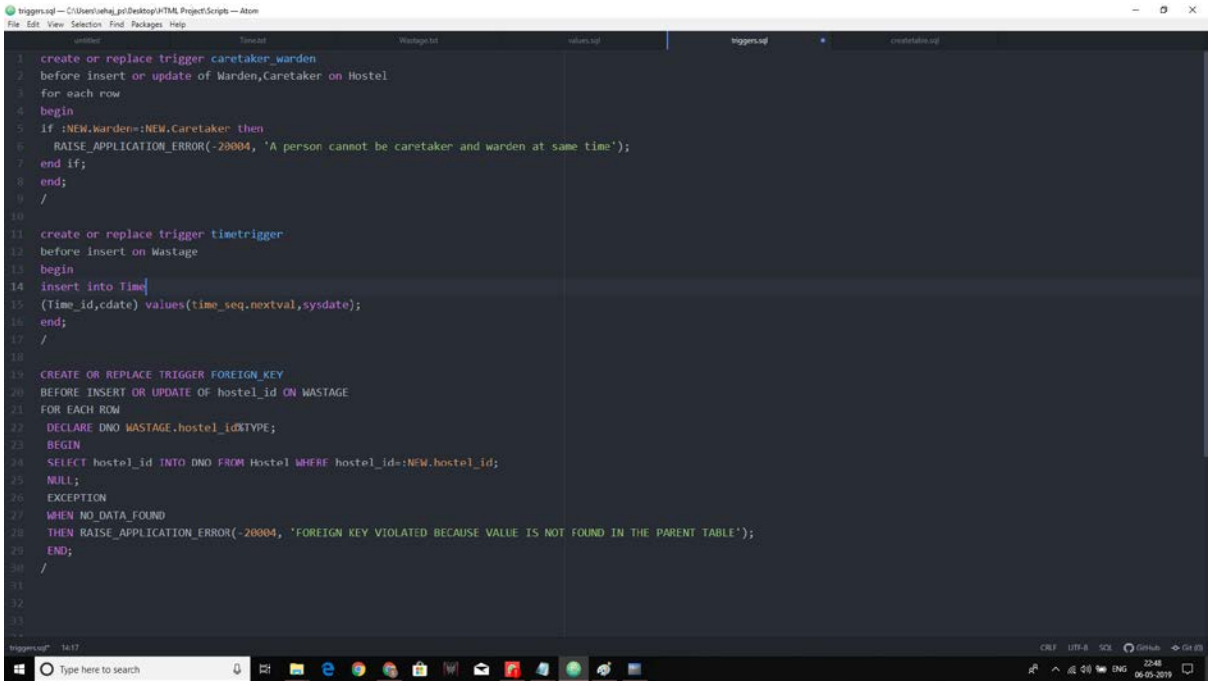


```
declare
x number(10);
y number(10);
z number(10);
p1 number(10);
p2 number(10);
Over_Prod_Id_to_be_deleted number(10);

   begin
        Over_Prod_Id_to_be_deleted:=&Over_Prod_Id_to_be_deleted;
        delete from OverProduction where op_id=Over_Prod_Id_to_be_deleted;

   end;
/
```



```
declare
x number(10);
y number(10);
z number(10);
hostel_id_to_be_deleted number(10);
p2 number(10);
p3 number(10);


  begin
    hostel_id_to_be_deleted:=&hostel_id_to_be_deleted;
    delete from Hostel where hostel_id=hostel_id_to_be_deleted;

  end;
/
```

Trigger



```
create or replace trigger caretaker_warden
before insert or update of Warden,Caretaker on Hostel
for each row
begin
if :NEW.Warden=:NEW.Caretaker then
  RAISE_APPLICATION_ERROR(-20004, 'A person cannot be caretaker and warden at same time');
end if;
end;
/

create or replace trigger timetrigger
before insert on Wastage
begin
insert into Time
(Time_id,cdate) values(time_seq.nextval,sysdate);
end;
/

CREATE OR REPLACE TRIGGER FOREIGN_KEY
BEFORE INSERT OR UPDATE OF hostel_id ON WASTAGE
FOR EACH ROW
 DECLARE DNO WASTAGE.hostel_id%TYPE;
 BEGIN
 SELECT hostel_id INTO DNO FROM Hostel WHERE hostel_id=:NEW.hostel_id;
 NULL;
 EXCEPTION
 WHEN NO_DATA_FOUND
 THEN RAISE_APPLICATION_ERROR(-20004, 'FOREIGN KEY VIOLATED BECAUSE VALUE IS NOT FOUND IN THE PARENT TABLE');
 END;
/
```

REFERENCES

- https://sites.google.com/site/dbms310cse/lecture-notes
- www.stackoverflow.com
- www.oracle.com
- Oracle 9i SQL.pdf