-- drop table Property_Service cascade constraints;

-- 1) Using a create query, create the property service table. Ensure that primary key and foreign keys added. (Note: You must examine and see its connections to other tables) (3 points)

```
create table Property_Service(
   propertyserviceid number(5) not null,
   propertyid number(5) not null,
   serviceid number(5) not null,
   servicedate date not null,
   employeeid numbe(5) not null,
   hoursworked number(5,2) not null,
   constraint propertyserviceid_pk primary key(propertyserviceid)
   constraint ownedproperty_propertyserviceid_fk foreign key(propertyid) references Owner_Property(propertyid),
   constraint ggservice_propertyserviceid_fk foreign key(serviceid) references GG_Service(serviceid),
   constraint employee5_propertyserviceid_fk foreign key(employeeid) references Employee5(employeeid),
);
```

-- 2) Create at least one insert query to the Property_Service table. (2 points)

```
Insert into Property_Service(propertyserviceid, propertyid, serviceid, servicedate, employeeid, hoursworked) values(1006, 36, 28, 05/15/2019, 4, 2.75);

insert into Property_Service(propertyserviceid, propertyid, serviceid, servicedate, employeeid, hoursworked) values(1001, 29, 29, 05/05/2019, 1, 4.5);

insert into Property_Service(propertyserviceid, propertyid, serviceid, servicedate, employeeid, hoursworked) values(1002, 31, 29, 05/08/2019, 2, 4.5);
```

-- 3) Create at least one update query to the Property_Service table (2 points)

```
update Property_Service set hoursworked = 7 where propertyid = 36 and propertyserviceid = 1006;
```

-- 4) Retrieve the owner id, name, property id, property name and property city. Sort the results by owner name ascending. (2 points)

```
select ownerid, concat(firstname,' ',lastname) as name, propertyid, propertyname, propertycity from Owner_Table, Owned_Property order by 'ASC';
```

-- 5) Retrieve the employees who worked on May 8, 2019 (You should also include the date in the query results). Include their first name, last name, the number of hours they worked, and the property name they worked at that time. (2 points)

select firstname, lastname, propertyname, city, hoursworked, servicedate from Property_Service, Employee5, Owned_Property where servicedate To_Date('05/08/2019');

-- 6) Using a set operator, show only the employees who did not work on May 8, 2019. Show the employee id, last name. (3 points)

select employeeid, lastname from Employee5 intersect select employeeid from Property_Service;

-- 7) Show the total hours accumulated per service. Show the service id, service description and the hours worked. (3 points)

select distinct(serviceid), servicedescription, count(hoursworked) as hoursaccumulatedperservice from Property_Service, GG_Service;

-- 8) Display the total wage of all employees by computing the accumulated hours of their work and multiplying it with 25 dollars. Further filter the results by showing only those employees that have earned more than 125 dollars. Display the data according to the output below

select distinct(employeeid), count(hoursworked) as totalHoursWorked, totalhourworked * 25 as money from Employee5 where totalHoursWorked > 3;

-- 10) Display the sum of the total hours per service id that are greater than average hours per service id. Your output should be similar to the one below: (3 points)

select distinct(serviceid) count(hoursworked) as totalHoursWorked, avg(totalhourworked) as avgHours from Property_Service where totalHoursWorked > avgHours;

---------- PL/SQL

-- 1) Create an anonymous block that will create a new table from existing tables. The new table will contain property id, property name and hours worked on that property. The new table should be furthered filtered to only contain those with hours worked that is more than 4.

From this new table, the anonymous block should display the minimum number of hours. (3 points)

```
declare
   propertyid    number(5)
   propertyname  varchar2(50)
   hoursworked   number(5,2)
begin
   select
      propertyid into propertyid,
      propertyname into propertyname,
      hoursworked into hoursworked,
   from Property_Service, Owned_Property
end;
```

-- 2 You will create a function and procedure for this number (8 points).

a) Create a function that will accept the number of hours and multiply it by 25. It will return the result of the computation.

b) Create a procedure that will call that function and display the total wage of

employees based on the total number of hours of an employee. Output will be similar to the one below:

```
declare
   numOfHours    number(4)
begin
   select
      hoursworked into numOfHours,
      numOfHours = numOfHours * 25
   from Property_Service
end;
```