1) Total shop revenue/rent at each airport

```
create type tot_rev as (airport_code char(3),total integer);
create or replace function get_revenue() returns setof tot_rev as $BODY$
declare
      fsum integer:=0;
      res tot_rev;
      r commercial_shop%rowtype;
      d airport%rowtype;
begin
      for d in select * from airport
      loop
            fsum:=0;
            for r in select * from commercial_shop where airport_code=d.airport_code
            loop
                  fsum:=fsum+r.revenue;
            end loop;
            res.airport_code:=d.airport_code;
            res.total:=fsum;
            return next res;
      end loop;
      return ;
end;
$BODY$ language 'plpgsql';
select * from get_revenue();
```

2)Average delay for each flight

```
create type my_type as (flight_code varchar , late_time time,cnt integer);
create or replace function get_delay() returns setof my_type as $BODY$
declare    ff actual_arr_dep%rowtype;
     cnt time;
     total integer;
     r my_type;
     dd flight_time_table%rowtype;

begin
     for dd in select * from flight_time_table
     loop
          cnt := (dd.sch_dep_time::time+(dd.duration_in_hours||' hours')::interval) -
(dd.sch_dep_time::time+(dd.duration_in_hours||' hours')::interval);
               total :=0;
          for ff in select * from actual_arr_dep where flight_code = dd.flight_code
               loop
                    if (ff.dep_timestamp::time+(dd.duration_in_hours||' hours')::interval) <=
(ff.arr_timestamp::time) then
                         total := total + 1;
                         cnt := cnt -(ff.dep_timestamp::time+(dd.duration_in_hours||' hours')::interval) +
(ff.arr_timestamp::time);
                         end if;
               end loop;
               if total>0 then
                    r.flight_code = dd.flight_code;
                    r.late_time = cnt/total;
                    r.cnt = total;
                         return next r;
               end if;
     end loop;
     return ;
end;
$BODY$ language 'plpgsql';
select * from get_delay();
```

3) Airline with the maximum number of delayed flights

```
Create type air_delay as (airline_code varchar,total integer)
Create type yyyy as(dep_timestamp timestamp,duration_in_hours integer,arr_timestamp timestamp)
create or replace function delay_airline() returns setof air_delay as $BODY$
declare
      r yyyy;
      air airlines%rowtype;
      total integer;
      res air_delay ;
begin
      for air in select * from airlines
      loop
            total :=0;
            for r in select dep_timestamp,duration_in_hours,arr_timestamp from flight_time_table natural join
actual_arr_dep where airline_code = air.airline_code
            loop
                  if (r.dep_timestamp::time+(r.duration_in_hours||' hours')::interval) < (r.arr_timestamp::time) then
                              total := total + 1;
                  end if;
            end loop;
            res.airline_code := air.airline_code;
            res.total = total;
            return next res;
      end loop;
      return;
end;
$BODY$ LANGUAGE 'plpgsql';
select * from delay_airline() order by total desc limit 1;
```

4)Rank all airport on the basis of total passenger traffic between any two given dates.

```
create type airport_type as(airport_code varchar ,total_p integer);
create type count_passenger as(flight_code varchar, cnt integer,src char(3),des char(3));
create type flight_detail as (flight_code varchar ,src_airport_code char(3),des_airport_code char(3));
create or replace function busy_airport(start_date date, end_date date) returns setof airport_type as $BODY$
declare
     r flight_detail;
          u airport_type;
          ff flight_time_table.flight_code%type;
          res count_passenger;
          air airport%rowtype;
          total integer;
          counter integer;
begin
     for air in select * from airport
     loop
          counter:=0;
          for r in select distinct (flight_code),src_airport_code,des_airport_code from actual_arr_dep natural join
flight_time_table
          loop
               total :=0;
                    for ff in select flight_code from booking where flight_code = r.flight_code and
exp_dep_date>=start_date and exp_dep_date<=end_date
                         loop
                              total :=total +1;
                         end loop;
                         if r.src_airport_code = air.airport_code or r.des_airport_code = air.airport_code then
                              counter:=counter+total;
                    end if;
          end loop;
          u.airport_code= air.airport_code;
          u.total_p = counter;
               return next u;
     end loop;
     return;
end;
```

```
$BODY$ LANGUAGE 'plpgsql';
select * from busy_airport('2019-10-10','2019-10-20') order by  total_p desc;
```

5)Available seats on any given date of each flight.

```
create type my_type_func as (flight_code varchar, vacancy integer);
create type dummy as (flight_code varchar, aircraft_type varchar,capacity integer);
create or replace function get_available_seats(given_date date) returns setof my_type_func as $BODY$
declare
     total integer;
     b booking%rowtype;
     res my_type_func;
     r dummy;
begin
     for r in select flight_code,aircraft,capacity from flight_time_table natural join aircraft
     loop
          total :=0;
          for b in select * from booking where flight_code=r.flight_code and exp_dep_date=given_date
          loop
               total:=total+1;
          end loop;
          res.flight_code = r.flight_code;
          res.vacancy = r.capacity - total;
          return next res;
     end loop;
     return;
end;
$BODY$ LANGUAGE 'plpgsql';
select * from get_available_seats('2019-12-10');
```