PL/SQL:

1. PL/SQL to find age of passenger using birthdate of passenger and to update the column age of passenger table.

```
CREATE OR REPLACE function find_age()
returns integer
language plpgsql
AS
$$
declare
  age1 integer;
  pass_id integer;
  cnt int;
  b_year int;
BEGIN
cnt:=(select count(*) from passenger);
pass_id:=101;
   while cnt>0 loop
            b_year:=(select EXTRACT(year from b_date) from passenger where p_id=pass_id);
            age1:=(date_part('year', CURRENT_DATE)-b_year);
     raise notice 'age is %',age1;
     update passenger set age=age1 where p_id=pass_id;

pass_id:=pass_id+1;
cnt:=cnt-1;
end loop;
return 0;
END;
$$;

select find_age();
```

	p_id [PK] integer	fname character varying (100)	lname character varying (100)	sex character varying (1)	email character varying (100)	b_date date	age integer
1	101	Aman	Walia	M	amanwalia@gmail.com	2001-01-21	20
2	102	Amit	Mistry	M	amitmistry@gmail.com	2001-04-23	20
3	103	Ahmed	Gamal	M	ahmed@gmail.com	2002-02-07	19
4	104	Shina	William	F	willam@gmail.com	2001-11-13	20
5	105	clara	Chopra	F	chopraclera@gmail.com	2001-05-07	20
6	106	Adarsh	Suthar	M	adarshsuthar@gmail.com	2001-03-03	20
7	107	Ayush	Mahera	M	Aayushmahra@gmail.com	2001-03-27	20
8	108	Omakar	moustafa	M	omkar@gmail.com	2001-02-02	20
9	109	Darshan	Mohamed	M	mohmed@gmail.com	1966-06-10	55
10	110	Avika	Maheta	F	Avikamaheta@gmail.com	1999-09-10	22
11	111	Mannat	Parekh	F	parekh@gmail.com	2001-01-22	20
12	112	Abdul	Raheman	M	abdulraheman@gmail.com	2009-04-26	12
13	113	Ramy	Gamal	M	ramy@gmail.com	1992-02-07	29
14	114	Krish	Devendra	M	Devendrakrish@gmail.com	1997-10-20	24
15	115	Salma	Ahmed	F	salma@yahoo.com	1998-05-20	23

FUNCTION:

2.function to find flight details including flight no, from, to, departure date, arrival date, available seats, route no and plane id between two given dates.

```
CREATE or REPLACE FUNCTION find_flight_between(first_date date, last_date date)
returns table (flight_no integer, fromm varchar(50), to0 varchar(50), class varchar(50), departure_date
date, arrival_date date, no_passenger integer, r_no integer, plane_id integer)
language plpgsql
as
$$
begin
  return query select * from flight f where f.departure_date >= first_date and f.departure_date <=
last_date;
end;
$$;

select find_flight_between('2021-10-1', '2021-10-30');
```

	find_flight_between record
1	(152305,MAHARASHTRA,DELHI,B,2021-10-02,2021-10-02,200,5,9168)
2	(121234,HESSE,"NEW YORK",A,2021-10-02,2021-10-03,160,4,9723)
3	(569876,CHANDIGARH,MAHARASHTRA,C,2021-10-01,2021-10-01,130,2,9165)
4	(561689,FLORIDA,CHANDIGARH,A,2021-10-05,2021-10-05,250,2,9235)
5	(784367,DELHI,"NEW YORK",A,2021-10-12,2021-10-12,190,1,9875)
6	(561902,DELHI,FLORIDA,B,2021-10-10,2021-10-10,200,3,9023)
7	(223056,HESSE,DELHI,A,2021-10-20,2021-10-20,120,4,9033)

TRIGGER:

1. **trigger to keep eye on ticket table & it makes new entry when changes occur in ticket table e.g. ticket added, cancelled ticket, updated ticket.**

```
create table ticket_audit(operation text, ticket_id integer, p_id integer, stamp timestamp);
drop trigger ticket_audit on ticket;

CREATE OR REPLACE FUNCTION do_ticket_audit()
RETURNS TRIGGER AS $ticket_audit$
begin
if(TG_OP='DELETE') then
insert into ticket_audit
        select 'CANCELLED TICKET',OLD.t_no,OLD.p_id,now();
        RETURN OLD;

elseif(TG_OP='UPDATE') then
        insert into ticket_audit select 'UPDATED TICKET',NEW.t_no,NEW.p_id,now();
        RETURN NEW;

elseif(TG_OP='INSERT') then
        insert into ticket_audit select 'TICKET ADDED',NEW.t_no,NEW.p_id,now();
        RETURN NEW;
END IF;
RETURN 'NULL';
END;
$ticket_audit$ LANGUAGE PLPGSQL;

create trigger ticket_audit
after INSERT or UPDATE or DELETE ON ticket
FOR EACH ROW EXECUTE PROCEDURE
do_ticket_audit();
```

insert into ticket values

(101001,'Y','N',784567,114,'J');

delete from ticket where t_no=101001;

UPDATE ticket set debit='Y',credit='N' where t_no=705001;

	operation text	ticket_id integer	p_id integer	stamp timestamp without time zone
1	TICKET ADDED	101001	114	2021-10-16 16:37:07.166782
2	CANCELLED TICKET	101001	114	2021-10-16 16:37:07.166782
3	UPDATED TICKET	705001	102	2021-10-16 16:41:40.635115

insert into ticket values

(101003,'Y','N',152305,113,'J');

insert into ticket values

(101004,'Y','N',152305,115,'J');

	operation text	ticket_id integer	p_id integer	stamp timestamp without time zone
1	TICKET ADDED	101001	114	2021-10-16 16:37:07.166782
2	CANCELLED TICKET	101001	114	2021-10-16 16:37:07.166782
3	UPDATED TICKET	705001	102	2021-10-16 16:41:40.635115
4	TICKET ADDED	101001	114	2021-10-16 17:26:08.890067
5	CANCELLED TICKET	101001	114	2021-10-16 17:28:55.640168
6	TICKET ADDED	101002	114	2021-10-16 17:31:44.798451
7	TICKET ADDED	101001	114	2021-10-16 17:34:57.391431
8	CANCELLED TICKET	101001	114	2021-10-16 17:34:57.391431
9	CANCELLED TICKET	101002	114	2021-10-16 17:35:47.375865
10	TICKET ADDED	101003	113	2021-10-17 14:03:27.550975
11	TICKET ADDED	101004	115	2021-10-17 14:03:27.550975

2. **trigger to keep eye on available seats in particular flight when new ticket is generated or ticket is cancelled. (e.g., available seats are 100 for flight x. after when new ticket is generated, then available seats will be 99 for flight x)**

```
CREATE OR REPLACE FUNCTION no_passenger_update()
RETURNS TRIGGER as $$
declare
old_no_passenger integer;
begin
if(TG_OP = 'DELETE') then
old_no_passenger := (select no_passenger from flight where flight_no in(select flight_no from ticket where
t_no = OLD.t_no));
old_no_passenger := old_no_passenger + 1;
update flight set no_passenger = old_no_passenger where flight_no in(select flight_no from ticket where
t_no = OLD.t_no);
        return old;

elseif(TG_OP='INSERT') then
old_no_passenger := (select no_passenger from flight where flight_no in(select flight_no from ticket where
t_no = NEW.t_no));
        old_no_passenger := old_no_passenger - 1;
update flight set no_passenger = old_no_passenger where flight_no in(select flight_no from ticket where
t_no = NEW.t_no);
        return new;
end if;
END;
$$ LANGUAGE PLPGSQL;

create trigger no_passenger_up
before INSERT or DELETE ON ticket
FOR EACH ROW EXECUTE PROCEDURE
no_passenger_update();
```

insert into ticket values

(101001,'Y','N',784567,114,'J');

	flight_no [PK] integer	fromm character varying (50)	to0 character varying (50)	class character varying (50)	departure_date date	arrival_date date	no_passenger integer	r_no integer	plane_id integer
1	121234	HESSE	NEW YORK	A	2021-10-02	2021-10-03	160	4	9723
2	152305	MAHARASHTRA	DELHI	B	2021-10-02	2021-10-02	200	5	9168
3	223056	HESSE	DELHI	A	2021-10-20	2021-10-20	120	4	9033
4	561689	FLORIDA	CHANDIGARH	A	2021-10-05	2021-10-05	250	2	9235
5	561902	DELHI	FLORIDA	B	2021-10-10	2021-10-10	200	3	9023
6	569576	MAHARASHTRA	NEW YORK	C	2021-11-01	2021-11-01	130	1	9167
7	569876	CHANDIGARH	MAHARASHTRA	C	2021-10-01	2021-10-01	130	2	9165
8	672015	NEW YORK	DELHI	A	2021-09-30	2021-09-30	150	4	9155
9	784367	DELHI	NEW YORK	A	2021-10-12	2021-10-12	190	1	9875
10	784567	HESSE	NEW YORK	A	2021-11-02	2021-11-03	189	4	9248

delete from ticket where t_no=101001;

	flight_no [PK] integer	fromm character varying (50)	to0 character varying (50)	class character varying (50)	departure_date date	arrival_date date	no_passenger integer	r_no integer	plane_id integer
1	121234	HESSE	NEW YORK	A	2021-10-02	2021-10-03	160	4	9723
2	152305	MAHARASHTRA	DELHI	B	2021-10-02	2021-10-02	200	5	9168
3	223056	HESSE	DELHI	A	2021-10-20	2021-10-20	120	4	9033
4	561689	FLORIDA	CHANDIGARH	A	2021-10-05	2021-10-05	250	2	9235
5	561902	DELHI	FLORIDA	B	2021-10-10	2021-10-10	200	3	9023
6	569576	MAHARASHTRA	NEW YORK	C	2021-11-01	2021-11-01	130	1	9167
7	569876	CHANDIGARH	MAHARASHTRA	C	2021-10-01	2021-10-01	130	2	9165
8	672015	NEW YORK	DELHI	A	2021-09-30	2021-09-30	150	4	9155
9	784367	DELHI	NEW YORK	A	2021-10-12	2021-10-12	190	1	9875
10	784567	HESSE	NEW YORK	A	2021-11-02	2021-11-03	190	4	9248

5. Cursor to find details of all male passengers with boarding and destination details who have flight at particular given date.

```
create or replace function passenger_flight_details(departure_date1 date)
returns varchar
as $$

declare
details text default '';
passenger_record record;
c1 CURSOR(departure_date1 date) FOR SELECT p.fname,p.lname,f.fromm,f.to0,p.sex FROM ticket
t,passenger p,flight f where t.p_id = p.p_id and t.flight_no = f.flight_no and f.departure_date =
departure_date1;

begin
open c1(departure_date1 date);
```

```
loop
fetch c1 into passenger_record;
exit when not found;
if passenger_record.sex = 'M' then
details := details || '  => ' ||passenger_record.fname ||  ' ' || passenger_record.lname ||' ,
'||passenger_record.fromm ||' to ' || passenger_record.to0;
end if;
end loop;
close c1;

return details;
end;
$$
language plpgsql;

select passenger_flight_details('2021-10-2');
```

	passenger_flight_details character varying
1	=> Ahmed Gamal , HESSE to NEW YORK => Amit Mistry , MAHARASHTRA to DELHI => Ramy Gamal , MAHARASHTRA to DELHI