1. Cab Ride History

Create a view containing attributes Driver ID, Driver Name, Start Location, Destination Location, Date, Time

CREATE VIEW my\_cab\_ride\_history AS

SELECT Driver\_ID, Payment, Start\_Location, Dest\_Location, Date FROM cab\_ride

SELECT \* FROM my\_cab\_ride\_history WHERE Customer\_ID = My\_ID;

1. Find all the bad drivers in order to get know whom to fire

SELECT \* FROM driver WHERE Rating<2 AND No\_of\_leaves>9;

CREATE FUNCTION bad\_drivers(min\_rate integer, max\_leaves integer) RETURNS record AS $BODY$

DECLARE

rate Driver.Rating%TYPE;

leave Driver.No\_of\_leaves%TYPE;

c1 refcursor;

ID Driver.License\_No%TYPE;

BEGIN

OPEN c1 FOR SELECT Rating, No\_of\_leaves, License\_No INTO rate, leave, ID FROM driver;

LOOP

END LOOP;

END;

$BODY$ LANGUAGE plpgsql;

1. Frequently locations
2. Surge Factor
3. Available Cars -> trigger

CREATE TRIGGER change\_status\_update\_area AFTER INSERT ON Cab\_Ride

FOR EACH ROW EXECUTE PROCEDURE status\_area\_func();

CREATE OR REPLACE FUNCTION status\_area\_func()

RETURNS TRIGGER AS $$

DECLARE

ID Cab\_Ride.Cab\_ID%TYPE;

D\_ID Cab\_Ride.Driver\_ID%TYPE;

Date date;

///extract date from timestamp

BEGIN

//// Cab Status Update

SELECT date (Cab\_Ride.Timestamp\_Current) AS Date;

SELECT NEW.Driver\_ID as D\_ID FROM Cab\_Ride;

SELECT Cab\_ID as ID FROM Assignation WHERE Driver\_ID = D\_ID AND Assigned\_date = date;

UPDATE Cab SET Status=” Booked” WHERE Cab\_Number\_Plate=ID;

END;

DECLARE

//// Count of Area

c1 refcursor;

x integer;

y integer;

a integer;

b integer;

BEGIN

LOOP

OPEN c1 FOR SELECT GpsA, GpsC FROM Area WHERE ;

END LOOP;

END;

$$ LANGUAGE plpgsql;

1. Biggest trigger

delimiter //

CREATE TRIGGER main\_trigger AFTER INSERT ON

for each row

begin

declare x int ;

if(new.number\_of\_axies > 0 ) then

set x = 1 ;

while x < new.number\_of\_axies do

insert into axle (train\_id,axle)

values

(new.train\_id,x);

set x=x+1;

end while ;

end if ;

end;//

delimiter ;

8)