Please Indicate below the group number and then past the following material:

(1) the SQL code you have used to create the schema of your database (only create table and alter table statements (if any), not statements for inserting values)

(2) the SQL code of the queries (possibly with an explanation)

(3) the SQL code used for query optimization for HW2. For each query, indicate the un-optimized version and the optimized one. In case the optimization has been realized through indexes, insert the SQL code for the index creation; in case you have modified the schema (e.g., defined constraints, changed the domain of a field, created a view, constructed a new materialized table, etc.), insert the code you have used for this modification.

GROUP 8
SIMONE BOESSO 1800408
TROMBONI GABRIELE 2088799

1) **GIVEN SEVERAL PROBLEMS IMPORTING THE CSV FILES AND THEN DUE TO TIME CONSTRAINTS, WE DECIDED TO IMMEDIATELY OPTIMIZE THE DATABASE:**

– CHANGING DATA TYPES (FROM BIGINT TO INT EXCEPT FOR ‘Num_Acc’)
– INDEX INSERTION
– PRIMARY KEYS

1A) TABLE **CHARACTERISTICS** ()

```
CREATE TABLE `characteristics` (
        `ID` int DEFAULT NULL,
        `Num_Acc` bigint NOT NULL,
        `year` int DEFAULT NULL,
        `month` int DEFAULT NULL,
        `day` int DEFAULT NULL,
        `hrmn` int DEFAULT NULL,
        `lighting` int DEFAULT NULL,
        `built_up_areas` int DEFAULT NULL,
        `atm` int DEFAULT NULL,
        `collision` int DEFAULT NULL,
        PRIMARY KEY (`Num_Acc`),
        UNIQUE KEY `ID_UNIQUE` (`ID`)
        ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
      COLLATE=utf8mb4_0900_ai_ci
```

1B) TABLE **PLACES**

```
CREATE TABLE `places` (
  `ID` int DEFAULT NULL,
  `Num_Acc` bigint NOT NULL,
  `type_road` int DEFAULT NULL,
  `traffic_regime` int DEFAULT NULL,
  `road_profile` int DEFAULT NULL,
  `road_plan` int DEFAULT NULL,
  `road_width` int DEFAULT NULL,
  `surface_condition` int DEFAULT NULL,
  `road_infrastructure` int DEFAULT NULL,
  `place_accident` int DEFAULT NULL,
  PRIMARY KEY (`Num_Acc`),
  UNIQUE KEY `ID_UNIQUE` (`ID`)
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
 COLLATE=utf8mb4_0900_ai_ci
```

1C) TABLE **USERS**

```
CREATE TABLE `users` (
  `ID` int NOT NULL,
  `Num_Acc` bigint DEFAULT NULL,
  `type_user` int DEFAULT NULL,
  `severity` int DEFAULT NULL,
  `sex` int DEFAULT NULL,
  `trip` int DEFAULT NULL,
  `safety_equipment` int DEFAULT NULL,
  `pedestrian_movement` int DEFAULT NULL,
  `year_of_birth` int DEFAULT NULL,
  `target` text,
  PRIMARY KEY (`ID`),
  KEY `Num_Acc` (`Num_Acc`)
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
 COLLATE=utf8mb4_0900_ai_ci
```

1D) TABLE **VEHICLES**

```
CREATE TABLE `vehicles` (
  `ID` int NOT NULL,
  `Num_Acc` bigint DEFAULT NULL,
  `category` int DEFAULT NULL,
  `target` text,
  PRIMARY KEY (`ID`),
  KEY `Num_Acc` (`Num_Acc`)
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
 COLLATE=utf8mb4_0900_ai_ci
```

2) QUERIES

-- **QUERY 1**. 'NUMBER OF NIGHT/NOT-NIGHT ACCIDENTS BASED ON ATMOSPHERIC CONDITIONS

```
SELECT atm as atmospheric_conditions, SUM( IF( lighting in (3, 4, 5), 1, 0)) as night, SUM( IF( lighting not in (3, 4, 5), 1, 0)) as not_night
FROM characteristics
GROUP BY atm
HAVING atm not in (7, 8, 9)
;
```

Atmospheric condition :
1 normal, 2 soft_rain, 3 heavy_rain, 4 snow, 5 fog, 6 heavy_wind
Lighting condition:
3,4,5 accidents by night in different situations (based on public illumination or not)

EXECUTION TIME BETWEEN 1.016 S and 2.359 s

-- **QUERY 2.** 'SERIOUSLY INJURED OR DEAD DRIVERS INVOLVED IN ACCIDENTS ON NORMAL ROAD SURFACES

```
SELECT        DISTINCT u.ID, u.Num_Acc, u.severity, u.type_user
FROM users as u, places as p
WHERE u.Num_Acc = p.Num_Acc
and (u.severity = 2 or u.severity = 3)
and (u.type_user = 1)
and (p.surface_condition = 1)
;
```

Severity:
2 dead users, 3 heavily injured users
Type_user: 1 driver
Surface_condition: 1 normal

EXECUTION TIME 2.266 s with fetching of 0.062 s

-- **QUERY 3**. 'NUMBER OF ACCIDENTS DIVIDED BY VEHICLE CATEGORY

```
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category  = 1
UNION
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
```

```
        WHERE category = 2
UNION
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category  = 3
UNION
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category  = 7
UNION
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category  = 13
;
```

Category of vehicles:
1 bycycle, 2 scooter, 3 quad, 7 cars, 13 camion
(we have to take into account only these five types of vehicles because we didn't have an explanation for the others)

EXECUTION TIME 13.250 s (TO OPTIMIZE)

-- **QUERY 4.** 'DISPLAYS THE NUMBER OF ACCIDENTS INVOLVING THREE OR MORE VEHICLES (4, 5, 6) BROKEN DOWN BY ROAD TYPE (HIGHWAY(1), EXPRESSWAY(2), SUBURBAN(3), URBAN(4), COUNTRY ROADS(5)) IN BAD ATMOSPHERIC CONDITIONS (HEAVY RAIN(3), SNOW(4), STRONG WIND(6))'

```
SELECT type_road, COUNT(DISTINCT c.Num_Acc) as accidents
FROM characteristics as c
JOIN places as p ON c.Num_Acc = p.Num_Acc
WHERE c.collision in (4, 5, 6) and c.atm in (3, 4, 6)
GROUP BY p.type_road
HAVING p.type_road not in (6, 9)
ORDER BY accidents DESC
;
```

Collision: three or more vehicles involved
Atm: 3 heavy rain, 4 snow, 6 heavy wind
Type_road: 1 highway, 2 expressways, 3 suburban, 4 urban, 5 country roads

EXECUTION TIME 1.750 s

-- **QUERY 5**. 'DISPLAYS NUMBER OF ACCDìIDENTS WITH ONLY TWO VEHICLES OR NOT COLLISIONS IN WHICH THE DRIVER IS UNDER 35 YEARS OF AGE DIVIDED INTO THOSE AT NIGHT OR SUNRISE WHILE RETURNING FROM PARTIES OR EVENTS (5 or 9) AND DAYTIME ONES DURING TRAVEL TO WORK OR SCHOOL (1, 2, 4) AND ACCIDENTS UNDER OTHER CONDITIONS'

```
SELECT
   CASE WHEN c.lighting <> 1 and u.trip in (5, 9) THEN 'Notturni o all\'alba durante rientro
da feste o eventi'
       WHEN c.lighting = 1 and u.trip in (1, 2, 4) THEN 'Diurni durante spostamenti per lavoro
o scuola'
       ELSE 'Altre condizioni'
   END as type_accident,
   COUNT(DISTINCT c.Num_Acc) as accidents
FROM characteristics as c
JOIN users as u ON c.Num_Acc = u.Num_Acc
WHERE c.year - u.year_of_birth < 35
  and u.type_user = 1 and c.collision in (1,2,3,7)
GROUP BY type_accident
;
```

collision: in (1,2,3,7) only two vehicles involved or no collision(7)
lighting: by night is <> 1, on day = 1
trip: reason for journey (5,9) free-time or party, (1,2,4) from home to school or work

EXECUTION TIME BETWEEN 8.203 s and 10 s

-- QUERY 6. 'DISPLAYS THE NUMBER OF ACCIDENTS IN DESCENDING ORDER ON VARIOUS TYPES OF ROAD SURFACES (NORMAL, WET, FLOODED, SNOWY, ICY) IN WHICH ALL DRIVERS WERE WEARING SEAT BELTS'

```
SELECT p.surface_condition, COUNT(p.Num_Acc) as accidents
FROM  places as p, (SELECT DISTINCT u2.Num_Acc FROM users as u2 WHERE
(u2.type_user = 1 and u2.safety_equipment <> 11)) as b
WHERE p.surface_condition in (1, 2, 4, 5, 7) and p.Num_Acc <> b.Num_Acc
GROUP BY p.surface_condition
ORDER BY accidents DESC
;
```

surface condition: 1 normal, 2 wet, 4 flooded, 5 snowy, 7 icy)
safety equipment: drivers with belt is equal to 11

EXECUTION TIME too slow, select clause in a from clause

-- **QUERY 7**. 'FOR EACH ACCIDENT WITH AT LEAST ONE SERIOUS OR FATAL INJURY, DISPLAYS THE TOTAL NUMBER OF USERS, USER'S TYPE, TOTAL NUMBER OF VEHICLES AND VEHICLE'S TYPE INVOLVED'

```
SELECT
       DISTINCT u.Num_Acc,
   COUNT(u.Num_Acc) as "Total users",
   SUM( CASE WHEN u.type_user = 1 THEN 1 ELSE 0 END) as "Drivers",
```

```
   SUM( CASE WHEN u.type_user = 2 THEN 1 ELSE 0 END) as "Passengers",
        SUM( CASE WHEN u.type_user = 3 THEN 1 ELSE 0 END) as "Pedestrians",
   SUM( CASE WHEN u.type_user = 4 THEN 1 ELSE 0 END) as "Rider",
   COUNT(DISTINCT v.target) as "Total vehicles",
   SUM( CASE WHEN v.category = 1 THEN 1 ELSE 0 END) as "Bycicle",
   SUM( CASE WHEN v.category = 2 THEN 1 ELSE 0 END) as "Scooter",
   SUM( CASE WHEN v.category = 3 THEN 1 ELSE 0 END) as "Quads",
   SUM( CASE WHEN v.category = 7 THEN 1 ELSE 0 END) as "Cars",
   SUM( CASE WHEN v.category = 13 THEN 1 ELSE 0 END) as "Trucks"
FROM users as u
JOIN vehicles v ON u.Num_Acc = v.Num_Acc
WHERE u.severity in (2, 3) and v.category in (1, 2, 3, 7, 13)
GROUP BY u.Num_Acc
;
```

Type user: 1 drivers, 2 passengers, 3 pedestrains, 4 riders
Category of vehicles: 1 bycycle, 2 scooter, 3 quad, 7 cars, 13 camion
Severity of users in the accident: 2 dead, 3 heavily injured
EXECUTION TIME 1.829 s with fetching of around 30 s

-- **QUERY 8.** 'NUMBER OF ACCIDENTS IN WHICH NO PEDESTRIANS OR CYCLISTS ARE INVOLVED (3 or 4) DISTINGUISHING BETWEEN THOSE OCCURRING IN URBAN AND NON-URBAN AREAS, BROKEN DOWN BY TIME OF DAY (DAYTIME, NIGHTTIME, SUNSET OR SUNRISE)'

```
CREATE VIEW accidents.accidentswithbikandped
AS
SELECT DISTINCT
   Num_Acc
FROM
   users as u
WHERE
        u.type_user in (3,4)
;
```

```
SELECT COUNT(DISTINCT c.Num_Acc) as 'diurno in periferia'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 1 and c.lighting = 1 and c.Num_Acc <> a.Num_Acc
UNION
SELECT COUNT(DISTINCT c.Num_Acc) as 'in periferia al tramonto o alba'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 1 and c.lighting = 2 and c.Num_Acc <> a.Num_Acc
UNION
SELECT COUNT(DISTINCT c.Num_Acc) as  'notturno in periferia'
```

```
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 1 and
c.lighting  not in (1, 2) and c.Num_Acc <> a.Num_Acc
UNION
SELECT COUNT(DISTINCT c.Num_Acc) as  'diurno in area urbana'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 2 and
c.lighting = 1 and c.Num_Acc <> a.Num_Acc
UNION
SELECT COUNT(DISTINCT c.Num_Acc) as 'in area urbana al tramonto o alba'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 2 and
c.lighting = 2 and c.Num_Acc <> a.Num_Acc
UNION
SELECT  COUNT(DISTINCT c.Num_Acc) as  'notturno in area urbana'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 2 and
c.lighting  not in (1, 2) and c.Num_Acc <> a.Num_Acc
;
```

Lighting: several band_time of the day
Type user: not pedestrains or riders(not in 3,4)
Areas of accident: 1 suburbs, 2 built-up
accidentswithbikandped view: table with all accidents involving pedestrians or riders
EXECUTION TIME too slow, using UNION and CREATE_VIEW

-- QUERY 9. 'DISPLAYS THE NUMBER OF ACCIDENTS IN DESCENDING ORDER INVOLVING THREE OR MORE VEHICLES WITH SERIOUS OR FATAL INJURIES BROKEN DOWN INTO THE VARIOUS ROAD INFRASTRUCTURES (TUNNELS 1 , BRIDGES 2 , RAILWAYS 4, PEDESTRIAN ZONE 6)'

```
SELECT p.road_infrastructure, COUNT(*) as accidents
FROM characteristics as c
JOIN users as u ON c.Num_Acc = u.Num_Acc
JOIN places as p ON c.Num_Acc = p.Num_Acc
WHERE c.collision  in (4, 5) and u.severity in (2, 3)
GROUP BY p.road_infrastructure
HAVING p.road_infrastructure in (1, 2, 4, 6)
ORDER BY accidents DESC
;
```

Road infrastructure of accident: (1 tunnel or galleries, 2 bridges, 4 rialways, 6 walkways)
Colllision: 4,5 to take accidents with three or more vehicles (road traffic disasters)
Severity: 2 user dead, 3 user heavily injured
EXECUTION TIME 12.375 s

-- **QUERY 10.** 'DISPLAYS THE NUMBER OF NIGHTTIME OR SUNRISE INCIDENTS WHILE RETURNING FROM PARTIES OR EVENTS (5 or 9) BROKEN DOWN BY AGE GROUP OF ALL PEOPLE INVOLVED (UNDER 30, OVER 30)'

```
CREATE VIEW accidents.nightpartyaccidents
AS
SELECT distinct
   u.Num_Acc
FROM
   users as u, characteristics as c
WHERE
        c.Num_Acc = u.Num_Acc and u.trip IN (5, 9) and c.lighting <> 1
;

CREATE VIEW accidents.usersages
AS
SELECT
   u.Num_Acc, COUNT(*) as 'TotUsers', SUM( CASE WHEN (c.year + 2000 -
u.year_of_birth) < 30  THEN 1 ELSE 0 END) as "Under30",
   SUM( CASE WHEN (c.year + 2000 - u.year_of_birth) >= 30  THEN 1 ELSE 0 END) as
"Over30"
FROM
   users as u, characteristics as c
WHERE
        c.Num_Acc = u.Num_Acc
GROUP BY u.Num_Acc
;

SELECT SUM( CASE WHEN ages.TotUsers = ages.Under30 THEN 1 ELSE 0 END) as
"Under30", SUM( CASE WHEN ages.TotUsers = ages.Over30 THEN 1 ELSE 0 END) as
"Over30"
FROM usersages ages, nightpartyaccidents acc
WHERE ages.Num_Acc = acc.Num_Acc
;
```

nightpartyaccidents view: table with accidents in the night with the party as the reason for journey
usersages view: table with info on ages of users involved in the accidents divided into under and over 30

EXECUTION TIME slow using VIEWS BETWEEN 32 s and 44 s (TO OPTIMIZE)

3) OPTIMIZED QUERIES

IT'S INTERESTING TO NOTE THAT USING THE VIEWS IN OUR SETTING IS NOT THE BEST WAY TO REDUCE THE EXECUTION TIME, ACTUALLY, A NESTED APPROACH WITH THE NOT EXIST CLAUSE HAS SHOWN BETTER PERFORMANCES IN ALL THE QUERIES.
WE HYPOTHESIZE THAT THIS BEHAVIOUR IS DUE TO THE LARGE NUMBER OF ROWS IN EACH VIEW.

-- **QUERY 3**. 'NUMBER OF ACCIDENTS DIVIDED BY VEHICLE CATEGORY

```
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category  = 1
UNION
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category = 2
UNION
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category  = 3
UNION
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category  = 7
UNION
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
        FROM vehicles
        WHERE category  = 13
;
```

Category of vehicles:
1 bycycle, 2 scooter, 3 quad, 7 cars, 13 camion
(we have to take into account only these five types of vehicles because we didn't have an explanation for the others)

EXECUTION TIME 13.250 s (TO OPTIMIZE)

-- **QUERY 3 OPTIMIZED**. 'NUMBER OF ACCIDENTS DIVIDED BY VEHICLE CATEGORY'

```
SELECT category, COUNT(DISTINCT Num_Acc) as accidents
FROM vehicles
GROUP BY category
HAVING category in (1, 2, 3, 7, 13)
;
```

EXECUTION TIME 4.234 s

-- QUERY 6. 'DISPLAYS THE NUMBER OF ACCIDENTS IN DESCENDING ORDER ON VARIOUS TYPES OF ROAD SURFACES (NORMAL, WET, FLOODED, SNOWY, ICY) IN WHICH ALL DRIVERS WERE WEARING SEAT BELTS'

```
SELECT p.surface_condition, COUNT(p.Num_Acc) as accidents
FROM  places as p, (SELECT DISTINCT u2.Num_Acc FROM users as u2 WHERE
(u2.type_user = 1 and u2.safety_equipment <> 11)) as b
WHERE p.surface_condition in (1, 2, 4, 5, 7) and p.Num_Acc <> b.Num_Acc
GROUP BY p.surface_condition
ORDER BY accidents DESC
;
```

surface condition: 1 normal, 2 wet, 4 flooded, 5 snowy, 7 icy)
safety equipment: drivers with belt is equal to 11
EXECUTION TIME too slow, select clause in a from clause **(TO OPTIMIZE)**

-- **QUERY 6 OPTIMIZED**. 'DISPLAYS THE NUMBER OF ACCIDENTS IN DESCENDING ORDER ON VARIOUS TYPES OF ROAD SURFACES (NORMAL, WET, FLOODED, SNOWY, ICY) IN WHICH ALL DRIVERS WERE WEARING SEAT BELTS'

```
SELECT p.surface_condition, COUNT(u.Num_Acc) as accidents
FROM users as u, places as p
WHERE u.Num_Acc = p.Num_Acc and p.surface_condition in (1, 2, 4, 5, 7)
        and NOT EXISTS
                (SELECT u2.Num_Acc
                        FROM users as u2
                        WHERE u2.type_user = 1 and u2.safety_equipment <> 11
                        and u2.Num_Acc = u.Num_Acc
                        and u2.ID <> u.ID)
GROUP BY p.surface_condition
ORDER BY accidents DESC
;
```

EXECUTION TIME 32.953 s

-- **QUERY 8.** 'NUMBER OF ACCIDENTS IN WHICH NO PEDESTRIANS OR CYCLISTS ARE INVOLVED (3 or 4) DISTINGUISHING BETWEEN THOSE OCCURRING IN URBAN AND NON-URBAN AREAS, BROKEN DOWN BY TIME OF DAY (DAYTIME, NIGHTTIME, SUNSET OR SUNRISE)'

```
CREATE VIEW accidents.accidentswithbikandped
AS
SELECT DISTINCT
   Num_Acc
FROM
   users as u
```

```
WHERE
        u.type_user in (3,4)
;

SELECT COUNT(DISTINCT c.Num_Acc) as 'diurno in periferia'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 1 and
c.lighting = 1 and c.Num_Acc <> a.Num_Acc
UNION
SELECT COUNT(DISTINCT c.Num_Acc) as 'in periferia al tramonto o alba'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 1 and
c.lighting = 2 and c.Num_Acc <> a.Num_Acc
UNION
SELECT COUNT(DISTINCT c.Num_Acc) as  'notturno in periferia'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 1 and
c.lighting  not in (1, 2) and c.Num_Acc <> a.Num_Acc
UNION
SELECT COUNT(DISTINCT c.Num_Acc) as  'diurno in area urbana'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 2 and
c.lighting = 1 and c.Num_Acc <> a.Num_Acc
UNION
SELECT COUNT(DISTINCT c.Num_Acc) as 'in area urbana al tramonto o alba'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 2 and
c.lighting = 2 and c.Num_Acc <> a.Num_Acc
UNION
SELECT  COUNT(DISTINCT c.Num_Acc) as  'notturno in area urbana'
FROM characteristics as c, accidentswithbikandped as a  WHERE c.built_up_areas = 2 and
c.lighting  not in (1, 2) and c.Num_Acc <> a.Num_Acc
;
```

Lighting: several band_time of the day
Type user: not pedestrains or riders(not in 3,4)
Areas of accident: 1 suburbs, 2 built-up
EXECUTION TIME too slow, using UNION and CREATE_VIEW **(TO OPTIMIZE)**

-- **QUERY 8 OPTIMIZED**. 'NUMBER OF ACCIDENTS IN WHICH NO PEDESTRIANS OR CYCLISTS ARE INVOLVED (3 or 4) DISTINGUISHING BETWEEN THOSE OCCURRING IN URBAN AND NON-URBAN AREAS, BROKEN DOWN BY TIME OF DAY (DAYTIME, NIGHTTIME, SUNSET OR SUNRISE)'

```
SELECT urban_area, time_day, COUNT(*) as accidents
FROM (
  SELECT
    CASE c.built_up_areas
      WHEN 1 THEN 'periferia'
      WHEN 2 THEN 'urbana'
    END as urban_area,
    CASE
```

```
    WHEN c.lighting = 1 THEN 'diurno'
    WHEN c.lighting = 2 THEN 'tramonto o alba'
    WHEN c.lighting not in (1, 2) THEN 'notturno'
  END as time_day,
  c.Num_Acc
 FROM characteristics as c
 WHERE NOT EXISTS (
  SELECT DISTINCT u.Num_Acc
  FROM users as u
  WHERE u.type_user  in (3, 4) and u.Num_Acc = c.Num_Acc
 )
) as incidents_table
GROUP BY urban_area, time_day
;
```

EXECUTION TIME 8.297 s

```
-- QUERY 10.  'DISPLAYS THE NUMBER OF NIGHTTIME OR SUNRISE INCIDENTS
WHILE RETURNING FROM PARTIES OR EVENTS (5 or 9) BROKEN DOWN BY AGE
GROUP OF ALL PEOPLE INVOLVED (UNDER 30, OVER 30)'

CREATE VIEW accidents.nightpartyaccidents
AS
SELECT distinct
   u.Num_Acc
FROM
   users as u, characteristics as c
WHERE
        c.Num_Acc = u.Num_Acc and u.trip IN (5, 9) and c.lighting <> 1
;

CREATE VIEW accidents.usersages
AS
SELECT
   u.Num_Acc, COUNT(*) as 'TotUsers', SUM( CASE WHEN (c.year + 2000 -
u.year_of_birth) < 30  THEN 1 ELSE 0 END) as "Under30",
   SUM( CASE WHEN (c.year + 2000 - u.year_of_birth) >= 30  THEN 1 ELSE 0 END) as
"Over30"
FROM
   users as u, characteristics as c
WHERE
        c.Num_Acc = u.Num_Acc
GROUP BY u.Num_Acc
;

SELECT SUM( CASE WHEN ages.TotUsers = ages.Under30 THEN 1 ELSE 0 END) as
"Under30", SUM( CASE WHEN ages.TotUsers = ages.Over30 THEN 1 ELSE 0 END) as
"Over30"
```

```
FROM usersages ages, nightpartyaccidents acc
WHERE ages.Num_Acc = acc.Num_Acc
;
```

nightpartyaccidents view: table with accidents in the night with the party as the reason for journey
usersages view: table with info on ages of users involved in the accidents divided into under and over 30

EXECUTION TIME slow using VIEWS BETWEEN 32 s and 44 s (TO OPTIMIZE)

-- **QUERY 10 OPTIMIZED**. 'DISPLAYS THE NUMBER OF NIGHTTIME OR SUNRISE INCIDENTS WHILE RETURNING FROM PARTIES OR EVENTS (5 or 9) BROKEN DOWN BY AGE GROUP OF ALL PEOPLE INVOLVED (UNDER 30, OVER 30)'

```
SELECT  COUNT( DISTINCT c.Num_Acc) as TOT_under30_TOT_over30
FROM characteristics as c
JOIN users as u ON c.Num_Acc = u.Num_Acc
WHERE u.trip IN (5, 9)
  and c.lighting <> 1
  and NOT EXISTS (
     SELECT DISTINCT u2.Num_Acc
     FROM users as u2
     WHERE u2.Num_Acc = u.Num_Acc
       and (c.year + 2000 - u2.year_of_birth) >= 30
  )
union all
SELECT COUNT( DISTINCT c1.Num_Acc)
FROM characteristics as c1
JOIN users as u3 ON c1.Num_Acc = u3.Num_Acc
WHERE u3.trip IN (5, 9)
  and c1.lighting <> 1
  and NOT EXISTS (
     SELECT DISTINCT u4.Num_Acc
     FROM users as u4
     WHERE u4.Num_Acc = u3.Num_Acc
       and (c1.year + 2000 - u4.year_of_birth) < 30
  )
;
```

EXECUTION TIME AROUND 24 s