Please past below
(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. changed the domain of a field, or constructed a new materialized table, etc.), insert the code you have used for this modification.

LEONARDO PLACIDI AND NEGIN AMININODOUSHAN
(1)

#Here we create the schema dmprojecthmk1 for HMK1

```
SHOW CHARACTER SET;
SET GLOBAL local_infile = 'ON';
SHOW GLOBAL VARIABLES LIKE 'local_infile';
#create database dmprojecthmk1;

create table candiv (
UNFLID VARCHAR(30),
CANCELLED DOUBLE,
CANCELLATION_CODE TEXT,
DIVERTED DOUBLE);



CREATE TABLE january(
UNFLID VARCHAR(30),
DAY_OF_MONTH INT,
DAY_OF_WEEK INT,
FL_DATE TEXT,
OP_CARRIER_AIRLINE_ID INT,
TAIL_NUM TEXT,
OP_CARRIER_FL_NUM INT,
ORIGIN_AIRPORT_ID INT,
ORIGIN_CITY_NAME TEXT,
ORIGIN_STATE_NM TEXT,
DEST_CITY_NAME TEXT,
DEST_STATE_NM TEXT,
CRS_DEP_TIME INT);


CREATE TABLE fsummary (
UNFLID VARCHAR(30),
CRS_ELAPSED_TIME DOUBLE,
ACTUAL_ELAPSED_TIME DOUBLE,
AIR_TIME DOUBLE,
FLIGHTS DOUBLE,
DISTANCE DOUBLE,
DISTANCE_GROUP INT
);


CREATE TABLE depperf (
UNFLID VARCHAR(30),
DEP_TIME DOUBLE,
DEP_DELAY DOUBLE,
DEP_DEL15 DOUBLE,
DEP_DELAY_GROUP DOUBLE
);

CREATE TABLE arrperf (
UNFLID VARCHAR(30),
CRS_ARR_TIME DOUBLE,
ARR_TIME DOUBLE,
ARR_DELAY DOUBLE,
ARR_DEL15 DOUBLE,
ARR_DELAY_GROUP DOUBLE);

-----------------------------------
(1forHMK2)
```

#Optimized database schema dmprojecthmk2 for HMK2;

```
#create database dmprojecthmk2;

CREATE TABLE january(
UNFLID VARCHAR(30) PRIMARY KEY,
YEARS INT,
DAY_OF_MONTH INT,
DAY_OF_WEEK INT,
FL_DATE TEXT,
OP_CARRIER_AIRLINE_ID INT,
TAIL_NUM TEXT,
OP_CARRIER_FL_NUM INT,
ORIGIN_AIRPORT_ID INT,
ORIGIN_CITY_NAME TEXT,
ORIGIN_STATE_NM TEXT,
DEST_CITY_NAME TEXT,
DEST_STATE_NM TEXT,
CRS_DEP_TIME INT);


create table candiv (
UNFLID VARCHAR(30) PRIMARY KEY,
YEARS INT,
CANCELLED DOUBLE,
CANCELLATION_CODE TEXT,
DIVERTED DOUBLE,
FOREIGN KEY (UNFLID) REFERENCES january(UNFLID));


CREATE TABLE fsummary (
UNFLID VARCHAR(30) PRIMARY KEY,
YEARS INT,
CRS_ELAPSED_TIME DOUBLE,
ACTUAL_ELAPSED_TIME DOUBLE,
AIR_TIME DOUBLE,
FLIGHTS DOUBLE,
DISTANCE DOUBLE,
DISTANCE_GROUP INT,
FOREIGN KEY (UNFLID) REFERENCES january(UNFLID)
);


CREATE TABLE depperf (
UNFLID VARCHAR(30) PRIMARY KEY,
YEARS INT,
DEP_TIME DOUBLE,
DEP_DELAY DOUBLE,
DEP_DEL15 DOUBLE,
DEP_DELAY_GROUP DOUBLE,
FOREIGN KEY (UNFLID) REFERENCES january(UNFLID));
```

```
 CREATE TABLE arrperf (
UNFLID VARCHAR(30) PRIMARY KEY,
YEARS INT,
CRS_ARR_TIME DOUBLE,
ARR_TIME DOUBLE,
ARR_DELAY DOUBLE,
ARR_DEL15 DOUBLE,
ARR_DELAY_GROUP DOUBLE,
FOREIGN KEY (UNFLID) REFERENCES january(UNFLID));
```

----

(2)

#Queries for HMK1

```
#Queries for HMK1
/*Our queries for HMK1 will propose an exploration of our dataset, so we will
start from simple ones, arriving to some hard interrogations of our database*/

/* Query1: One of the interesting queries is that which carriers had worse performance.
In this query we will see sum of delays normalized by number of flights for each carrier */

select j.OP_CARRIER_AIRLINE_ID,
         SUM(d.DEP_DELAY) as sum_of_delay ,
    count(*) as Num_flights ,
    SUM(d.DEP_DELAY) / count(d.DEP_DELAY)  as performance
from january as j , depperf as d
where j.UNFLID = d.UNFLID
group by j.OP_CARRIER_AIRLINE_ID
ORDER BY  performance DESC
LIMIT 5;

/* Now we see which carriers had worse performance than others and as we see the difference
of performance between the first 4 carriers and the fifth one is a lot*/

/* Query2: which routs(city-city) had the most number of arriving flights
 during January of 2020 with less than 15 mins arrival delay */

 select ORIGIN_CITY_NAME, DEST_CITY_NAME, count(*) as count
  from   january as j ,
              (select UNFLID
      from arrperf
      where ARR_DEL15 = 0) as a
 where j.UNFLID = a.UNFLID and
            YEAR(STR_TO_DATE(j.FL_DATE, '%Y-%m-%d')) = 2020
 group by ORIGIN_CITY_NAME, DEST_CITY_NAME
 order by count DESC
            LIMIT 10;

# As we can see New York- Chicago rout has the most number of flights


/* Query3: compare the percentage of departing canceled flights in Newyork airports: */

select ORIGIN_AIRPORT_ID,
    count(*) as total_flights,
        count(IF(CANCELLED = 1, 1, NULL)) as cancelled_flights  ,
        concat((COUNT(IF(CANCELLED = 1, 1, NULL))/count(*))*100,'%')  as percentage
from january as j, candiv as c
where j.UNFLID = c.UNFLID and  j.ORIGIN_CITY_NAME = 'New York, NY'
group by ORIGIN_AIRPORT_ID
order by percentage desc;

#We conclude that the quality of airports based on canceled flights in NY is actually good
considering the traffic of this great city.


/* Query4 : Carriers that had more delayed flights in 2010 than the worst carrier in 2020
 (carriers that had worse performance in 2010 than the worst carrier in 2020) */

  select OP_CARRIER_AIRLINE_ID, sum(DEP_DEL15)/count(*) as delayed_ratio
 from january as j ,  depperf as de
 where YEAR(STR_TO_DATE(j.FL_DATE, '%Y-%m-%d')) = 2020 and
            j.UNFLID = de.UNFLID
 group by OP_CARRIER_AIRLINE_ID
 having sum(DEP_DEL15)/count(*) >  (select  sum(DEP_DEL15) /count(*)
            from january as ja ,  depperf as d
            where YEAR(STR_TO_DATE(ja.FL_DATE, '%Y-%m-%d')) = 2010 and
            ja.UNFLID = d.UNFLID
            group by OP_CARRIER_AIRLINE_ID
            ORDER BY  count(*) DESC
            LIMIT 1);

/* An interesting thing about this query is that if we change year 2010 and 2020 the result
will be empty which shows that in 2020, flights have less delays than 2010 so the
performance of carriers have become better */
```

```
/*  Query 5: The tail number of the plane that has more averaged flied distance in year 2015 ?*/
select j.TAIL_NUM , AVG(f.DISTANCE)
        from january as j, fsummary as f
        where j.UNFLID = f.UNFLID and
        YEAR(STR_TO_DATE(j.FL_DATE, '%Y-%m-%d')) = 2015
        group by j.TAIL_NUM
        ORDER BY AVG(f.DISTANCE) DESC
        LIMIT 10 ;

/* As we see in the result the carrier ID 20409 has more flied distance and planes
with higher flied distance belong to this carrier */



/* 6)

QUERY 6: What day of the week we found more flights in 2010, 2015 and 2020?
to solve this a ordinate way would be to create 3 view of our dataset,for january2020,
 january2015 and january2010 with the days and the number of flight in those days. */

create view jan2020 as select day_of_week, sum(day_of_week) as sumdays
                                                from january
                where YEAR(STR_TO_DATE(january.FL_DATE, '%Y-%m-%d')) = 2020
                group by day_of_week;

create view jan2015 as select day_of_week, sum(day_of_week) as sumdays
                                                from january
                where YEAR(STR_TO_DATE(january.FL_DATE, '%Y-%m-%d')) = 2015
                group by day_of_week;

create view jan2010 as select day_of_week, sum(day_of_week) as sumdays
                                                from january
                where YEAR(STR_TO_DATE(january.FL_DATE, '%Y-%m-%d')) = 2010
                group by day_of_week;

#And now let's see the busiest day of the week for this three views

select a.day_of_week as day2020, b.day_of_week as day2015, c.day_of_week as day2010
 from jan2020 as a, jan2015 as b, jan2010 as c
 having max(a.sumdays)
```

```
        and max(b.sumdays)
        and max(c.sumdays);
```

#where 3 is wednesday, 4 is thursday, 5 is friday! so during the years the busiest day of the month changed!!! People tend to travel more in the week than in the weekend!!

#to not forget let's drop the views for now

```
drop view jan2010;
drop view jan2015;
drop view jan2020;
```

/* 7)

Query7: Now let's address a query about the ratio of the avg(arrival_delay) and avg(air_time) in 2020, and so on about average velocity, and average delay per distance*/

#This time we won't use a view, but just a select for the year 2020

```
select avdel/avtim, avtim/avdis, avdel/ avdis
from( select avg(air_time) avtim , avg(arr_delay)  avdel, avg(distance) avdis
                from fsummary as f, january as j, arrperf as a
                where YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) = 2020
                        and f.unflid = j.unflid
        and a.unflid = j.unflid) as s;
```

/* 8)

Query8: Now let's see in which city in 2015 the most cancelled flights were headed!*/

```
select *
from (select dest_city_name, sum(cancelled) as sumcancelled
                from january as j, candiv as c
                where YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) = 2015
                        and c.unflid=j.unflid and c.cancelled=1
                group by dest_city_name) as a
group by a.sumcancelled
order by sumcancelled desc
limit 10;
```

#So chicago had a real problem!!!!

```
/* 9)
Query9: Now, as we saw that in 2015 Chigago was a mess, let's list all the number
flights that were cancelled on Thursdays(4), the busiest day according to previous queries
 and confront it to the max/min number of flights cancelled!*/

select *
from (select sum(cancelled) as s, day_of_week
                    from january left outer join candiv
                    on january.unflid = candiv.unflid
       where dest_city_name = 'Chicago, IL'
       or origin_city_name = 'Chicago, IL'
                    group by day_of_week) as k
where k.day_of_week = 4
or  k.s >= all( select sum(cancelled) as s
from january left outer join candiv
on january.unflid = candiv.unflid
where dest_city_name = 'Chicago, IL'
or origin_city_name = 'Chicago, IL'
group by day_of_week)
or  k.s <= all( select sum(cancelled) as s
                    from january left outer join candiv
                                        on january.unflid = candiv.unflid
                    where dest_city_name = 'Chicago, IL'
                              or origin_city_name = 'Chicago, IL'
                  group by day_of_week)
order by k.day_of_week asc;

#a pretty busy day eh!!!


/* 10)
Query10: Let's now see how much distance in jan2010, jan2015, jan2020 flights covered in
the us confronted to how much distance covered from flights from or dest to NY and Chicago */

select Year, sum_distance, sumChicNY, sumChicNY/sum_distance
from (select YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) as Year, sum(distance) as
sum_distance
            from fsummary as f, january as j
      where j.unflid = f.unflid
             group by YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) ) as un,
                    (select YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) as ChicNY,
sum(distance) as sumChicNY
                    from fsummary as f, january as j
                    where j.unflid = f.unflid and (dest_city_name = 'Chicago, IL'
                                                    or origin_city_name = 'Chicago, IL'
                                                    or dest_city_name = 'New York, NY'
                                                    or origin_city_name = 'New York, NY')
          group by YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) ) du
where un.Year = du.ChicNY
```

#A pretty big slice pass and goes from those 2 cities!!! 20% of all the distance in flights is covered from and to NY and Chicago!!!

----

(3)

#Optimized Queries for HMK2

```
/* OLD Query3: compare the percentage of departing canceled flights in Newyork airports: */

select ORIGIN_AIRPORT_ID,
      count(*) as total_flights,
            count(IF(CANCELLED = 1, 1, NULL)) as cancelled_flights  ,
            concat((COUNT(IF(CANCELLED = 1, 1, NULL))/count(*))*100,'%')  as percentage
from january as j, candiv as c
where j.UNFLID = c.UNFLID and  j.ORIGIN_CITY_NAME = 'New York, NY'
group by ORIGIN_AIRPORT_ID
order by percentage desc;
```

```
#new QUERY3

/* we do 1) create materalized table for origin_city in newyork
           2) indexed the origin_airport_id
           3) indexed cancel on candiv*/

set profiling = 1;
create table Newyork AS
          SELECT UNFLID,ORIGIN_AIRPORT_ID FROM january WHERE ORIGIN_CITY_NAME
= 'New York, NY';

ALTER TABLE Newyork ADD INDEX airport (ORIGIN_AIRPORT_ID);

ALTER TABLE candiv ADD INDEX cancel (CANCELLED)  ;
```

```
select ORIGIN_AIRPORT_ID, count(*) as total_flights, COUNT(IF(CANCELLED = 1, 1, NULL))
as cancelled_flights  ,
concat((COUNT(IF(CANCELLED = 1, 1, NULL))/count(*))*100,'%')  as percentage
        from Newyork as n, candiv as c
        where n.UNFLID = c.UNFLID
        group by ORIGIN_AIRPORT_ID
        order by percentage desc;

show profiles;

#1.79 seconds versus 4.259 old one

/* OLD  query 5: The tail number of the plane that has more averaged flied distance in year
2015 ?*/
select j.TAIL_NUM , AVG(f.DISTANCE)
        from january as j, fsummary as f
        where j.UNFLID = f.UNFLID and
        YEAR(STR_TO_DATE(j.FL_DATE, '%Y-%m-%d')) = 2015
        group by j.TAIL_NUM
        ORDER BY AVG(f.DISTANCE) DESC
        LIMIT 10 ;

/* As we see in the result the carrier ID 20409 has more flied distance and planes
with higher flied distance belong to this carrier */

#new QUERY5

/* We do 1) create the January2015 table.
          2) alter the columns type and index by Tail_NUM in jan2015*/

create table jan2015 AS
        SELECT UNFLID, TAIL_NUM FROM january WHERE YEARS = 2015;

ALTER TABLE jan2015
  MODIFY TAIL_NUM varchar(30);

ALTER TABLE jan2015 ADD INDEX tail_numb (TAIL_NUM);


select j.TAIL_NUM , AVG(f.DISTANCE)
        from jan2015 as j, fsummary as f
        where j.UNFLID = f.UNFLID
        group by j.TAIL_NUM
        ORDER BY  AVG(f.DISTANCE) DESC
        LIMIT 10 ;
SHOW PROFILES;

#WOW 2.09 from 6.151 old Q5


# OLD QUERY 6: What day of the week we found more flights in 2010, 2015 and 2020?
to solve this a ordinate way would be to create 3 view of our dataset,for january2020,
 january2015 and january2010 with the days and the number of flight in those days. */

create view jan2020 as select day_of_week, sum(day_of_week) as sumdays
                                                from january
                where YEAR(STR_TO_DATE(january.FL_DATE, '%Y-%m-%d')) = 2020
                group by day_of_week;

create view jan2015 as select day_of_week, sum(day_of_week) as sumdays
                                                from january
                where YEAR(STR_TO_DATE(january.FL_DATE, '%Y-%m-%d')) = 2015
                group by day_of_week;

create view jan2010 as select day_of_week, sum(day_of_week) as sumdays
                                                from january
                where YEAR(STR_TO_DATE(january.FL_DATE, '%Y-%m-%d')) = 2010
                group by day_of_week;

#And now let's see the busiest day of the week for this three views

select a.day_of_week as day2020, b.day_of_week as day2015, c.day_of_week as day2010
 from jan2020 as a, jan2015 as b, jan2010 as c
 having max(a.sumdays)
        and max(b.sumdays)
        and max(c.sumdays);

 #where 3 is wednesday, 4 is thursday, 5 is friday! so during the years the busiest day of the
month changed!!! People tend to travel more in the week than in the weekend!!

#to not forget let's drop the views for now

drop view jan2010;
drop view jan2015;
drop view jan2020;
```

```
#new QUERY6

/* We do 1) We create tables with the days of the week, and index those table by year, so we
make simpler to retrieve the sum(day_of_the_week).
        2) index the materialized table we just wrote.
*/

create table ja2020 AS
        select day_of_week, count(day_of_week) as sumdays
                                                from january
                where YEARS = 2020
                group by day_of_week;
create table ja2015 AS
        select day_of_week, count(day_of_week) as sumdays
                                                from january
                where YEARS = 2015
                group by day_of_week;
create table ja2010 AS
        select day_of_week, count(day_of_week) as sumdays
                                                from january
                where YEARS = 2010
                group by day_of_week;

ALTER TABLE ja2020 ADD INDEX d_o_week (DAY_OF_WEEK);
ALTER TABLE ja2015 ADD INDEX d_o_week (DAY_OF_WEEK);
ALTER TABLE ja2010 ADD INDEX d_o_week (DAY_OF_WEEK);

select a.day_of_week as day2020, b.day_of_week as day2015, c.day_of_week as day2010
 from ja2020 as a, ja2015 as b, ja2010 as c
 having max(a.sumdays)
        and max(b.sumdays)
        and max(c.sumdays);
show profiles;

#0.000415 seconds from 4.42 old one!!!

#OLD Query9: Now, as we saw that in 2015 Chigago was a mess, let's list all the number
flights that were cancelled on Thursdays(4), the busiest day according to previous queries
 and confront it to the max/min number of flights cancelled!*/

select *
from (select sum(cancelled) as s, day_of_week
            from january left outer join candiv
            on january.unflid = candiv.unflid
    where dest_city_name = 'Chicago, IL'
    or origin_city_name = 'Chicago, IL'
            group by day_of_week) as k
where k.day_of_week = 4
or  k.s >= all( select sum(cancelled) as s
from january left outer join candiv
on january.unflid = candiv.unflid
where dest_city_name = 'Chicago, IL'
or origin_city_name = 'Chicago, IL'
group by day_of_week)
or  k.s <= all( select sum(cancelled) as s
            from january left outer join candiv
                        on january.unflid = candiv.unflid
            where dest_city_name = 'Chicago, IL'
                  or origin_city_name = 'Chicago, IL'
           group by day_of_week)
order by k.day_of_week asc;

#a pretty busy day eh!!!


#new QUERY9

/* We do 1) We already indexed for ORIGIN_CITY_NAME january, now on january we index the
DEST_CITY_NAME
       2) We create in index for DAY_OF_WEEL for january*/

create table chic as
     select unflid, day_of_week
                  from january
                  where dest_city_name = 'Chicago, IL'
                        or origin_city_name = 'Chicago, IL';

alter table chic add index  d_o_w (day_of_week);

select *
from (select sum(cancelled) as s, day_of_week
      from chic left outer join candiv
      on chic.unflid = candiv.unflid
      group by day_of_week) as k
where k.day_of_week = 4 or k.s >= all( select sum(cancelled) as s
from chic left outer join candiv
on chic.unflid = candiv.unflid
group by day_of_week)
        or  k.s <= all(select sum(cancelled) as s
                from chic left outer join candiv
                on chic.unflid = candiv.unflid
                group by day_of_week)
order by k.day_of_week asc;
show profiles;

#9.28 from >400 seconds!!!

#OLD Query10: Let's now see how much distance in jan2010, jan2015, jan2020 flights covered
in
the us confronted to how much distance covered from flights from or dest to NY and Chicago */

select Year, sum_distance, sumChicNY, sumChicNY/sum_distance
from (select YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) as Year, sum(distance) as
sum_distance
         from fsummary as f, january as j
     where j.unflid = f.unflid
          group by YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) ) as un,
                (select YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) as ChicNY,
sum(distance) as sumChicNY
                from fsummary as f, january as j
                where j.unflid = f.unflid and (dest_city_name = 'Chicago, IL'
                                                or origin_city_name = 'Chicago, IL'
                                                or dest_city_name = 'New York, NY'
                                                or origin_city_name = 'New York, NY')
        group by YEAR(STR_TO_DATE(j.fl_date, '%Y-%m-%d')) ) du
where un.Year = du.ChicNY

#A pretty big slice pass and goes from those 2 cities!!! 20% of all the distance in flights is
covered from and to NY and Chicago!!!


#new QUERY10

/* We create a chicnew Materialized table and add some indexes for years in chichnew and
january to optimize the select statement */

create table chicnew as
      select unflid, years
```

```
                     from january
                     where dest_city_name = 'Chicago, IL'
                            or origin_city_name = 'Chicago, IL'
          or dest_city_name = 'New York, NY'
                            or origin_city_name = 'New York, NY';

alter table chicnew add index yearz (years);
alter table january add index yy (years);

select Year, sum_distance, sumChicNY, sumChicNY/sum_distance
from          (select j.YEARS as YEAR, sum(distance) as sum_distance
                      from fsummary as f, january as j
                      where j.unflid = f.unflid
                      group by YEAR ) as un,
                     (select c.YEARS as ChicNY, sum(distance) as sumChicNY
                      from fsummary as f, chicnew as c
                      where c.unflid = f.unflid
                      group by ChicNY) du
where un.Year = du.ChicNY;
show profiles;

#9.7 seconds from 15.951

#The execution times we showed are bounded by the executing machine, so will be a similar
number every time but not precisely the ones we showed, it depends on outer factors but we
proved anyway a huge optimization.
```