



SCHOOL OF  
MATHEMATICS SCIENCES AND  
INFORMATION TECHNOLOGY

## OPTIMIZATION QUERY DATABASE

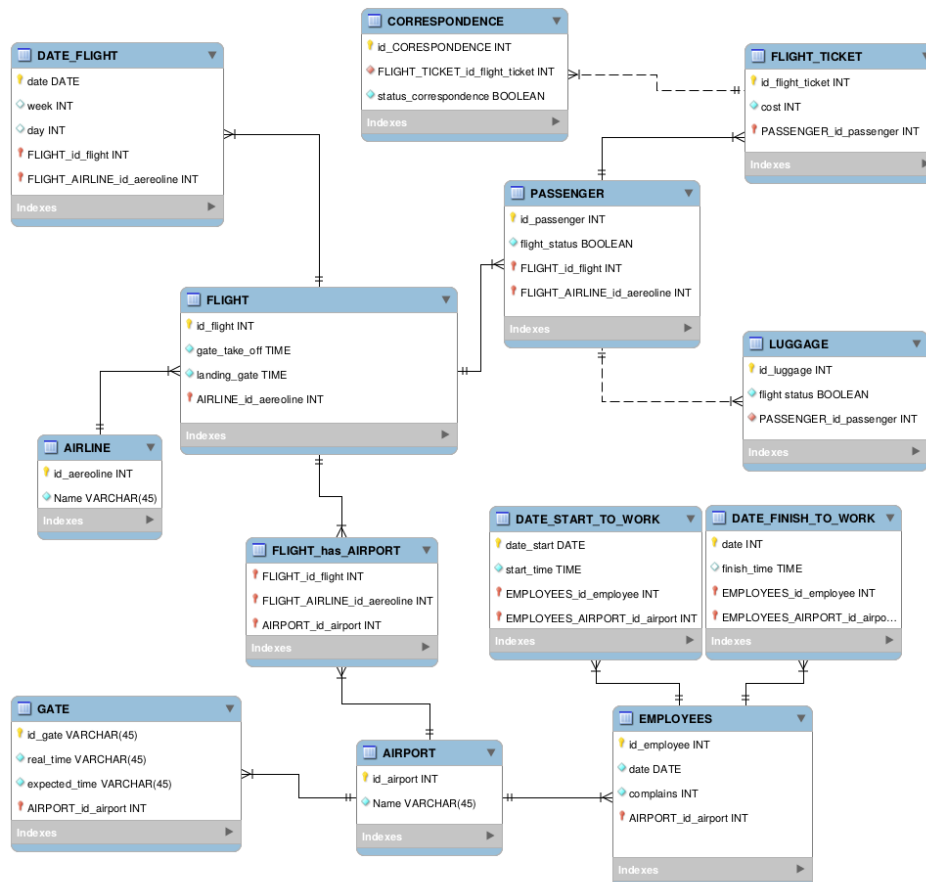
María Fernanda Molina \*

Andrés Riofrío †

Lenin Rueda ‡

Emil Vega §

August 3, 2017



\*maria.molina@yachaytech.edu.ec

†andres.riofrio@yachaytech.edu.ec

‡lenin.rueda@yachaytech.edu.ec

§emil.vega@yachaytech.edu.ec

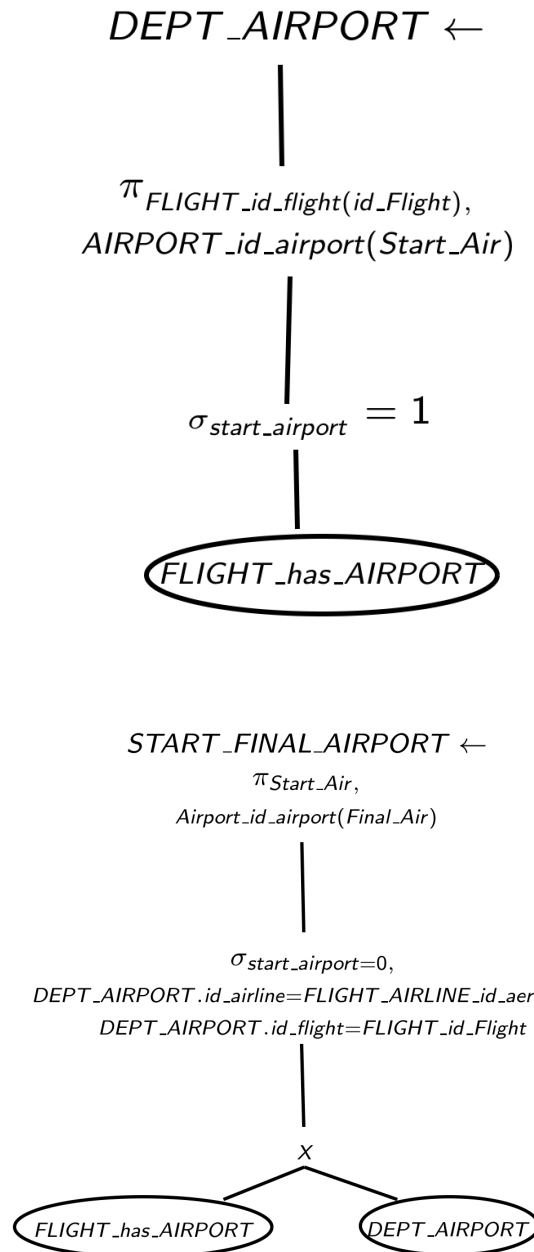
# 1 Each TAME group flight

## 1.1 the start and end airports?

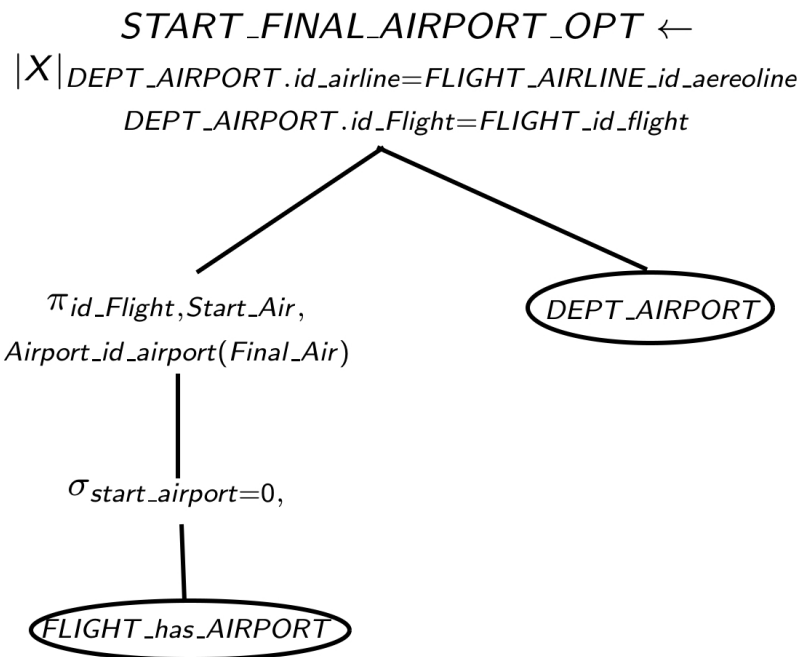
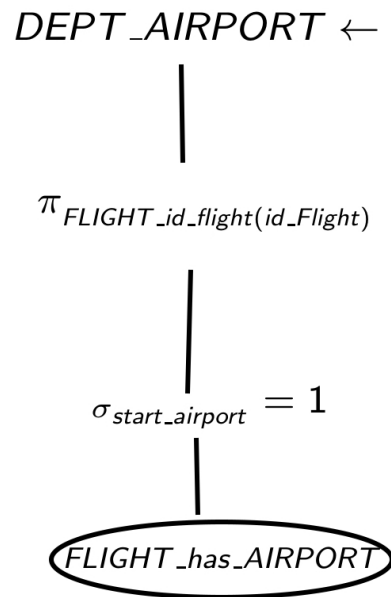
MySQL code without optimization:

```
SELECT Start_Air , AIRPORT_id_airport AS Final_Airport
FROM FLIGHT_has_AIRPORT,
(
  SELECT FLIGHT_AIRLINE_id_aereoline AS id_Airline ,
  FLIGHT_id_flight AS id_Flight , AIRPORT_id_airport AS Start_Air
  FROM FLIGHT_has_AIRPORT
  WHERE start_airport=1
) DEPT_AIRPORT
WHERE start_airport=0 AND Start_A.id_Airline=FLIGHT_AIRLINE_id_aereoline
```

Algebra optimization without optimization:



Algebra with optimization:



MySQL code with optimization:

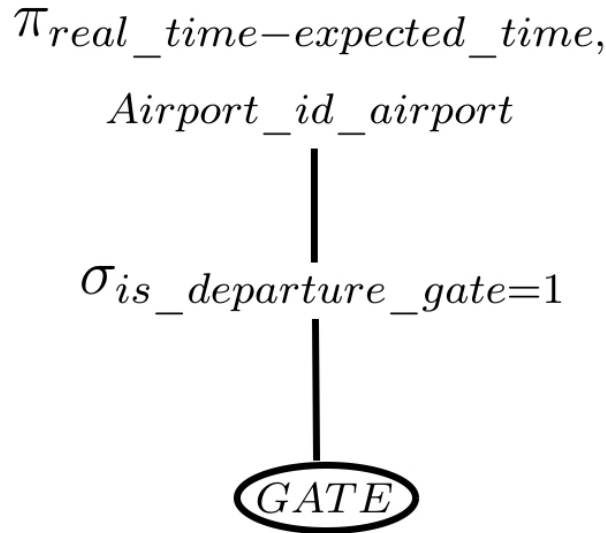
```
SELECT Start_Airport , Final_Airport
FROM      (
            SELECT FLIGHT_id_flight AS id_Flight ,
                   AIRPORT_id_airport AS Start_Airport
            FROM FLIGHT_has_AIRPORT
            WHERE start_airport=1
            ) START_AIR NATURAL JOIN
            (
            SELECT FLIGHT_id_flight AS id_Flight ,
                   AIRPORT_id_airport AS Final_Airport
            FROM FLIGHT_has_AIRPORT
            WHERE start_airport=0
            ) FINAL_AIR
```

## 1.2 the difference between the expected departure time and the real departure time?

MySQL code without optimization:

```
SELECT real_time - expected_time , Airport_id_airport
FROM GATE
WHERE is_departure_gate=1
```

Algebra without optimization:



Note: In this case the algebra without optimization is same that the algebra with optimization then the MySQL algorithm not change.

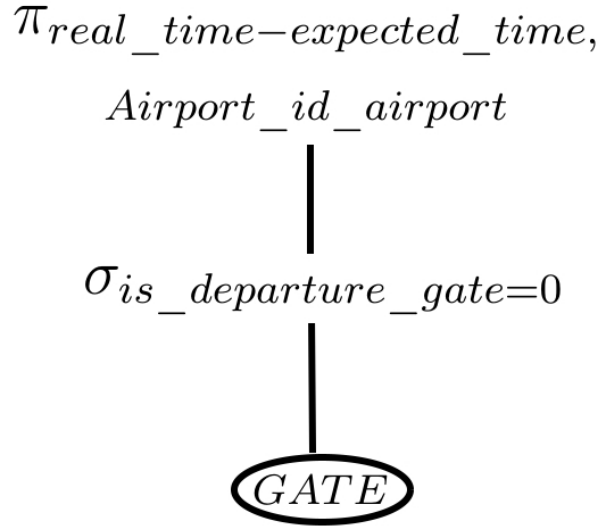
## 1.3 the difference between the expected arrival time and the real arrival time?

MySQL code without optimization:

```
SELECT real_time - expected_time , Airport_id_airport
FROM GATE
WHERE is_departure_gate=0
```

Algebra without optimization:

Note: In this case the algebra without optimization is same that the algebra with optimization then the MySQL algorithm not change.

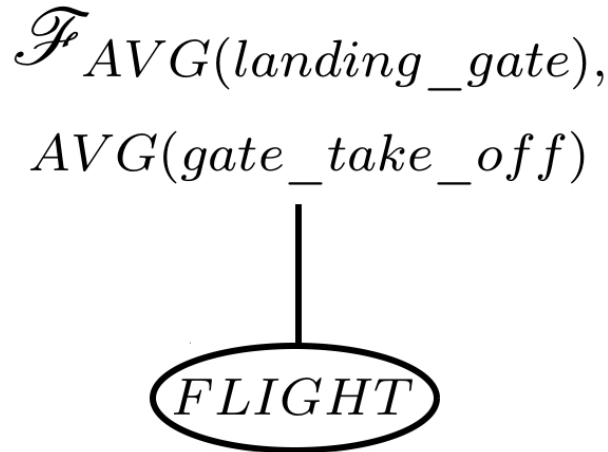


**1.4 the average time spent on the ground at the origin and the destination airports: i.e. from departure from the gate till take-off or from landing till complete stop?**

MySQL code without optimization:

```
SELECT AVG(landing_gate), AVG(gate_take_off)
FROM FLIGHT
```

Algebra without optimization:



Note: In this case the algebra without optimization is same that the algebra with optimization then the MySQL algorithm not change.

**1.5 the average length of the flight for each day of the week and each week in the year?**

MySQL code without optimization:

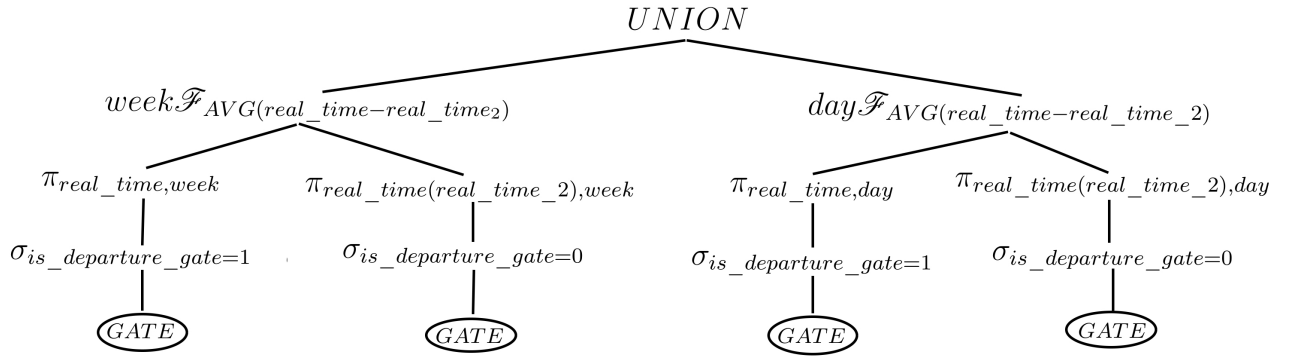
```
SELECT AVG(real_time-real_time_2),
FROM GATE,
(
  SELECT real_time AS real_time_2
  FROM GATE
  WHERE is_departure_gate=0
```

```

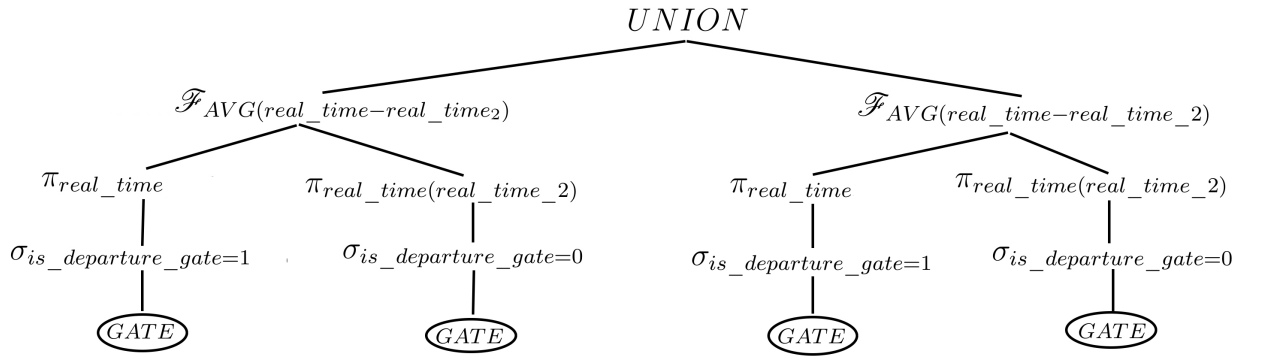
GROUP BY week
) R_2
WHERE is_departure_gate=1
GROUP BY week
UNION
SELECT AVG(real_time-real_time_2)
FROM GATE,
(
SELECT real_time AS real_time_2
FROM GATE
WHERE is_departure_gate=0
GROUP BY day
) R_d2
WHERE is_departure_gate=1
GROUP BY day

```

Algebra optimization without optimization:



Algebra with optimization:



MySQL code with optimization:

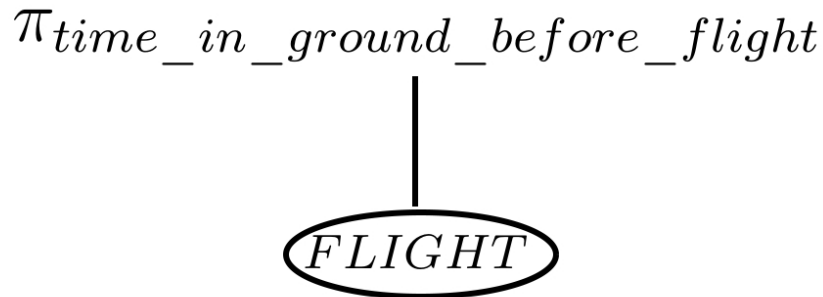
```
SELECT AVG(real_time-real_time_2),
FROM GATE,
    (
        SELECT real_time AS real_time_2
        FROM GATE
        WHERE is_departure_gate=0
        GROUP BY real_time2
    ) R_2
WHERE is_departure_gate=1
GROUP BY real_time
UNION
SELECT AVG(real_time-real_time_2)
FROM GATE,
    (
        SELECT real_time AS real_time_2
        FROM GATE
        WHERE is_departure_gate=0
        GROUP BY real_time_2
    ) R_d2
WHERE is_departure_gate=1
GROUP BY real_time
```

### 1.6 the time each SAS group plane spends on the ground between two consecutive flights for any given day and airport?

MySQL code without optimization:

```
SELECT AVG(time_in_ground_before_flight),
FROM FLIGHT
```

Algebra without optimization:



Note: In this case the algebra without optimization is the same that the algebra with optimization. Hence, the MySQL algorithm not change.

## 1.7 The number of passengers who have lost their flight transfer (either luggage only or passenger and luggage) after that flight?

MySQL code without optimization:

```
CREATE VIEW LUG_LOST AS
SELECT id_luggage ,PASSENGER_id_passenger
FROM LUGGAGE
WHERE flight_status=FALSE;
```

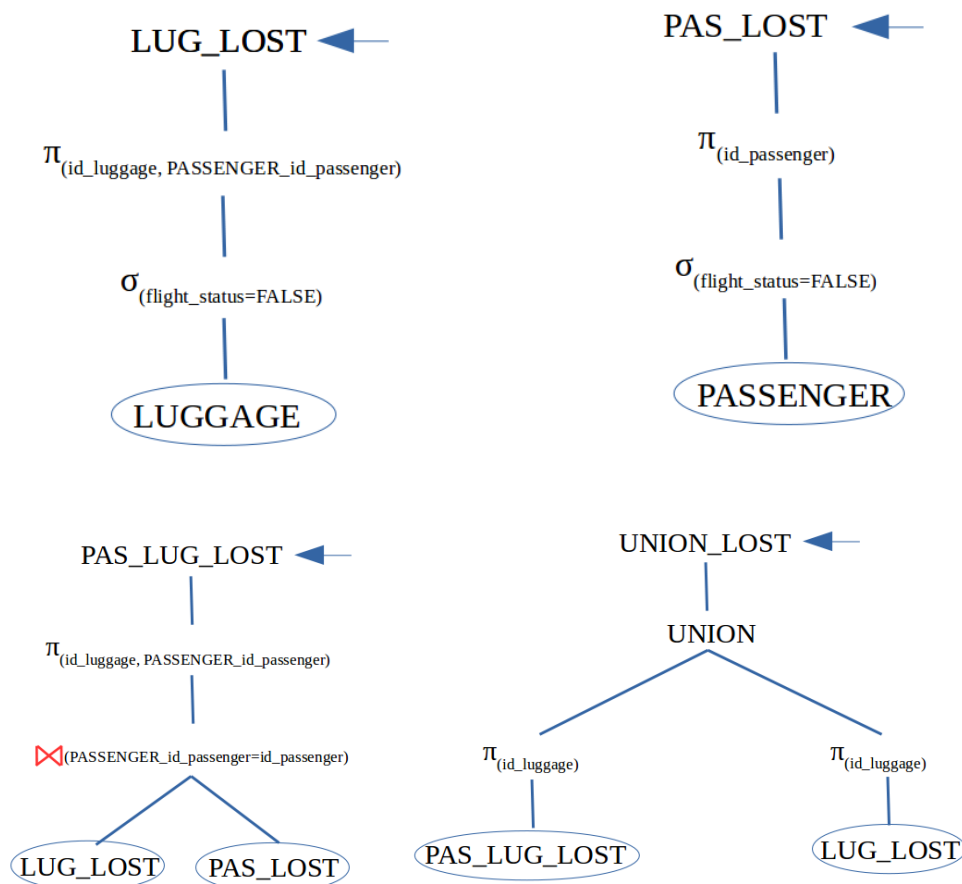
```
CREATE VIEW PAS_LOST AS
SELECT id_passenger
FROM PASSENGER
WHERE flight_status=FALSE;
```

```
CREATE VIEW PAS_LUG_LOST AS
SELECT id_luggage ,PASSENGER_id_passenger
FROM LUG_LOST
JOIN PAS_LOST
ON LUG_LOST.PASSENGER_id_passenger=PAS_LOST.id_passenger;
```

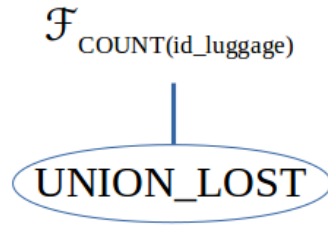
```
CREATE VIEW UNION_LOST AS
SELECT id_luggage FROM PAS_LUG_LOST
UNION
SELECT id_luggage FROM LUG_LOST;
```

```
SELECT COUNT(id_luggage)
FROM UNION_LOST
```

Tree of relational algebra without optimization







Note: In this case the algebra without optimization is the same that the algebra with optimization. Hence, the MySQL algorithm not change.

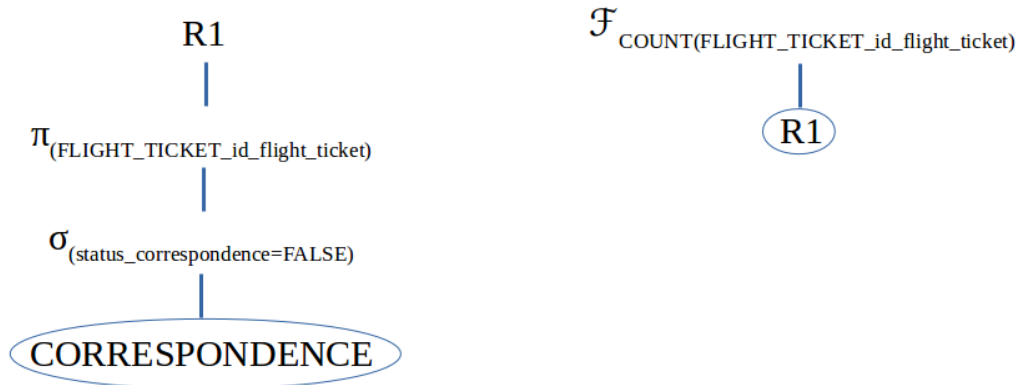
## 1.8 The total amount of the flight tickets of the passengers who have lost their correspondence after that flight?

MySQL code without optimization:

```
CREATE VIEW R1 AS
SELECT DISTINCT FLIGHT_TICKET_id_flight_ticket
FROM CORRESPONDENCE
WHERE status_correspondence=FALSE;
```

```
SELECT COUNT(FLIGHT_TICKET_id_flight_ticket)
FROM R1
```

Tree of relational algebra without optimization



Note: In this case the algebra without optimization is the same that the algebra with optimization. Hence, the MySQL algorithm not change.

## 2 Each TAME group ground employee

### 2.1 Which airport is he or she assigned to?

MySQL code without optimization

```
SELECT id_employee, Airport_id_airport
FROM EMPLOYEE
```

Tree of relational algebra without optimization



Note: In this case the algebra without optimization is the same that the algebra with optimization. Hence, the MySQL algorithm not change.

## 2.2 How many hours per week does he/she work for each given airport?

MySQL code without optimization

```
SELECT id_employee , Airport_id_airport , hours_per_week
FROM EMPLOYEE
```

Tree of relational algebra without optimization



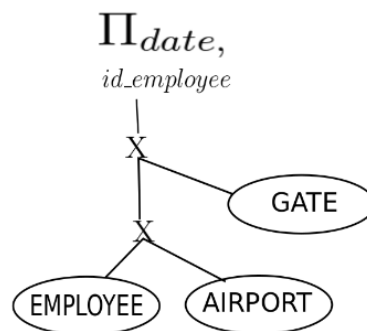
Note: In this case the algebra without optimization is the same that the algebra with optimization. Hence, the MySQL algorithm not change.

## 2.3 the number of flight delays he/she has been involved with?

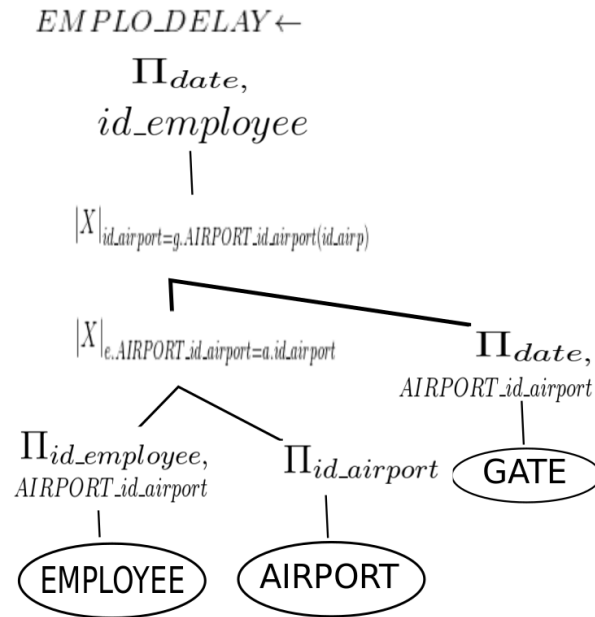
MySQL code without optimization:

```
SELECT id_employee , complains , hours_per_week , AIRPORT_id_airport
FROM EMPLOYEE
UNION
SELECT id_airport , airport
FROM AIRPORT
UNION
SELECT id_gate , real_time , expected_time , is_departure_gate , date , week
FROM GATE,
(
    SELECT id_employee , date
    FROM GATE
) NEW VIEW
```

Algebra optimization without optimization:



Algebra with optimization:



MySQL code with optimization:

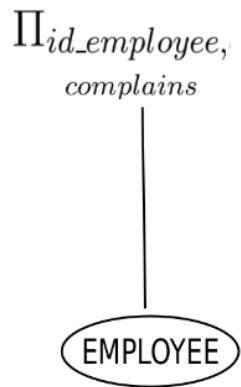
```
SELECT id_employee , AIRPORT_id_airport
FROM EMPLOYEE AS E
JOIN AIRPORT AS A,
    (
        SELECT id_airport
        FROM AIRPORT
    )
ON E.AIRPORT_id_airport=A.id_airport
JOIN GATE AS G,
    (
        SELECT id_employee , date
        FROM AIRPORT
    )
ON id_airport=G.AIRPORT_id_airport
```

## 2.4 the number of formal complaints about that employee?

```
SELECT id_employee , complains
FROM EMPLOYEE
```

Tree of relational algebra without optimization

Note: In this case the algebra without optimization is the same that the algebra with optimiza-



tion. Hence, the MySQL algorithm not change.

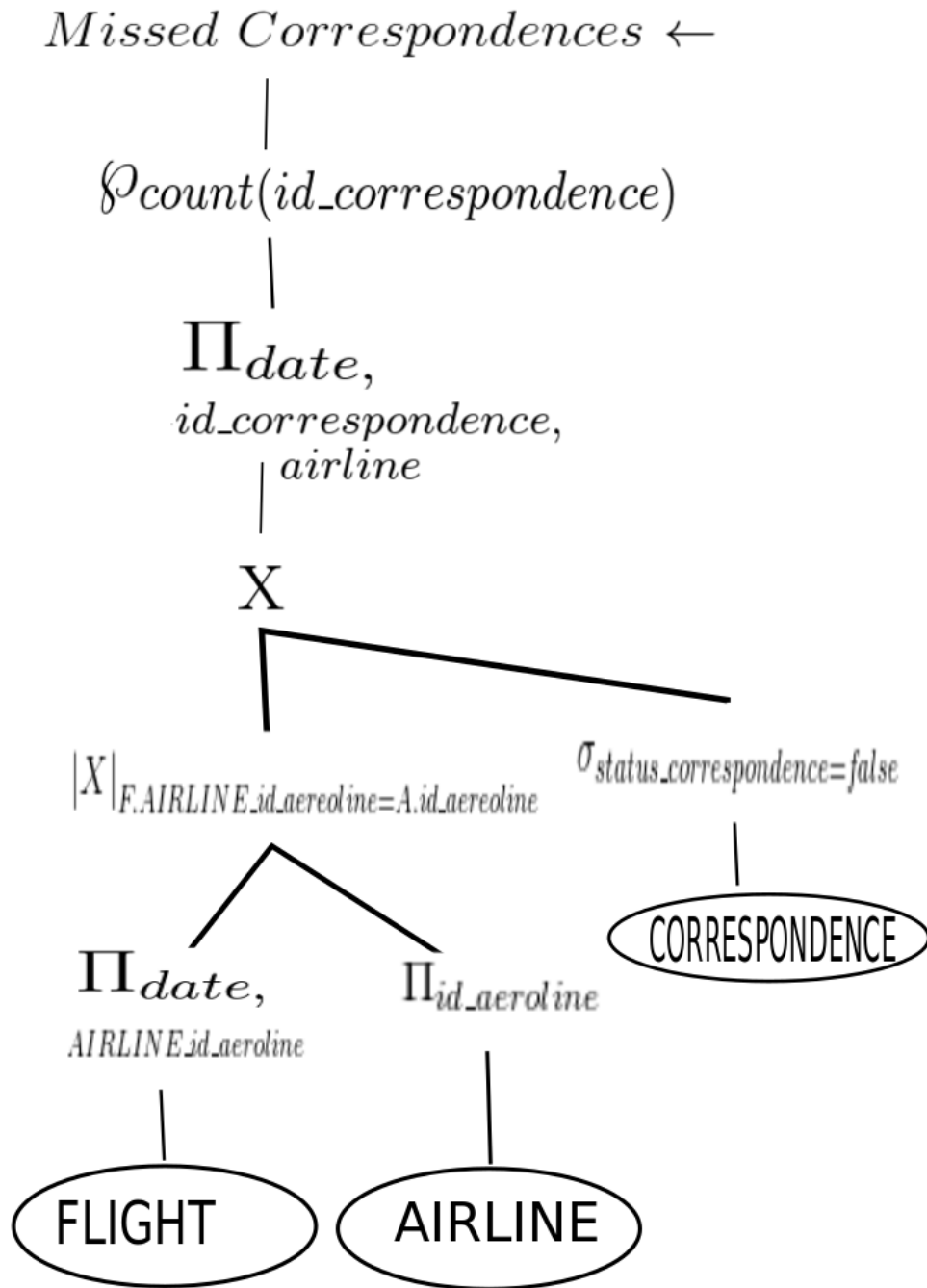
### 3 Each TAME group hub:

#### 3.1 how many missed correspondences between flights of the same airline are there for any given day, week, or year?

MySQL code without optimization:

```
CREATE VIEW MISSED CORRESPONDENCE AS
SELECT date , AIRLINE_id_aereoline
FROM FLIGHT AS F
JOIN AIRLINE,
    (
        SELECT id_aereoline
        FROM AIRLINE AS A
    )
ON F.AIRLINE_id_aereoline=A.id_aereoline
UNION
SELECT status_correspondence
FROM CORRESPONDENCE
WHERE status_correspondence=FALSE
SELECT COUNT(*)
UNION
SELECT id_correspondence , date , id_aereoline
```

Tree of relational algebra without optimization



Note: In this case the algebra without optimization is the same that the algebra with optimization. Hence, the MySQL algorithm not change.

### 3.2 how many missed transfers between flights of airlines of SAS group are there for any given day, week, or year?

MySQL code without optimization:

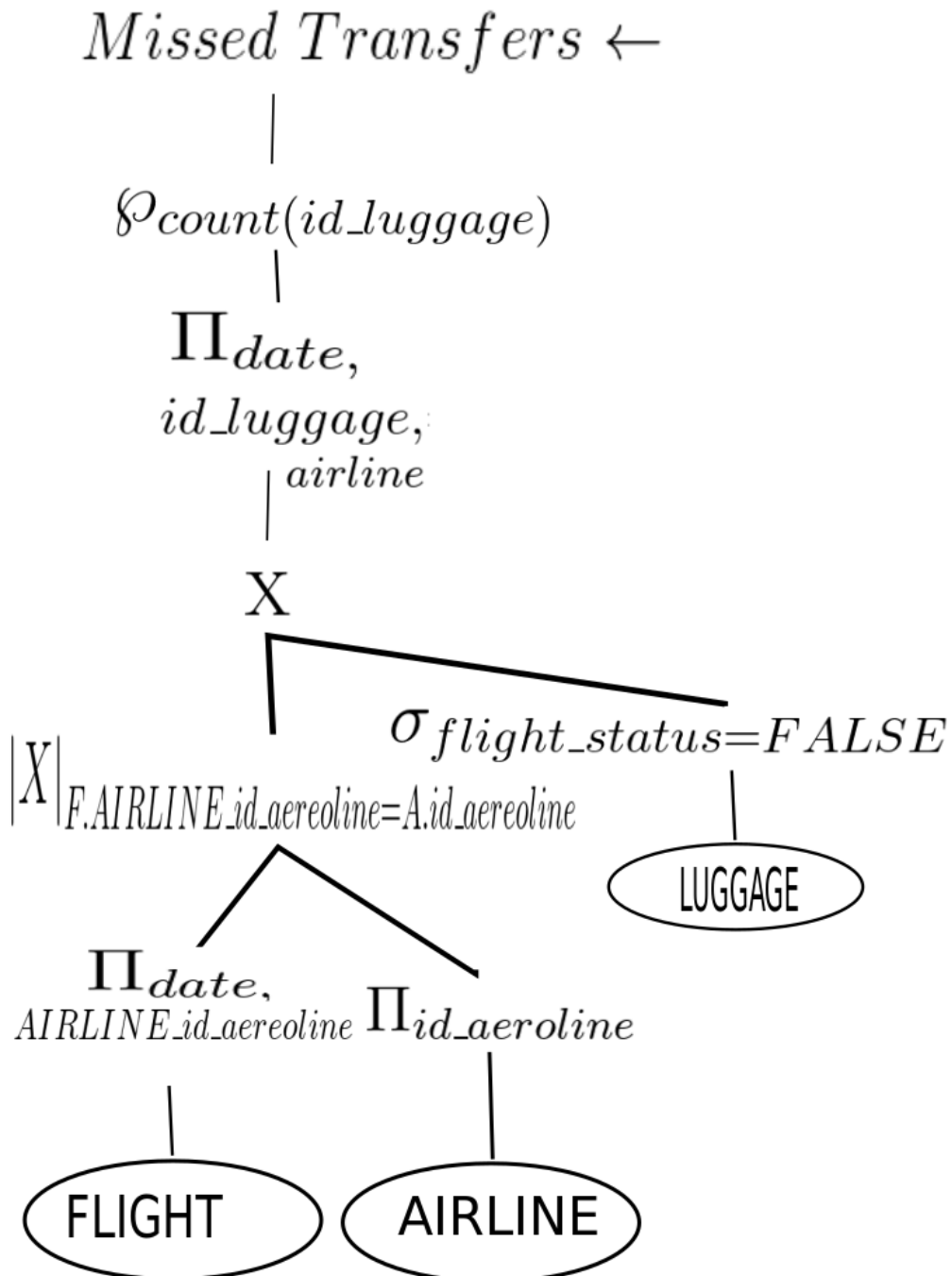
```
CREATE VIEW MISSED TRANSFERS AS
SELECT date , AIRLINE_id_aeroline
FROM FLIGHT AS F
JOIN AIRLINE,
(
```

```

SELECT id_aereoline
FROM AIRLINE AS A
)
ON F.AIRLINE_id_aereoline=A.id_aereoline
UNION
SELECT flight_status
FROM LUGGAGE
WHERE flight_status=FALSE
SELECT COUNT(*)
UNION
SELECT id_luggage, date, id_aereoline

```

Tree of relational algebra without optimization



Note: In this case the algebra without optimization is the same that the algebra with optimization. Hence, the MySQL algorithm not change.

**3.3 the total of all flight departure delays?**

**3.4 the total of all flight arrival delays?**

**3.5 the gates that have accepted all the airlines in the SAS group?**

**3.6 the gate with the highest total time of flight delays?**