

Database Project: TAME's database state

M. Molina, A. Riofrío, E.Vega, L.Rueda

Yachay Tech University

Friday, 28th of July of 2017

Outline

- 1 Conceptual Model
 - Entity relationship diagram
- 2 Tables and dependencies
 - Functional Dependence
- 3 Normalization
 - Without Normalization
 - With Normalization
- 4 Explaining a Query
 - Tables
 - SQL Query
- 5 Optimization
 - Algebra optimization
 - MySql with optimization

Figure 1: Entity relationship diagram of airline

Tables and dependencies

- **Table FLIGHT**

Attributes of *FLIGHT*

$$FLIGHT = (\underline{id_flight}, landing_gate, gate_take_off)$$

Functional Dependences of *FLIGHT*

$$\{\underline{id_flight}\} \rightarrow \{landing_gate, gate_take_off\}$$

- **Table AIRLINE**

Attributes of *AIRLINE*

$$AIRLINE = (\underline{id_airline}, name)$$

Functional Dependences of *AIRLINE*

$$\{\underline{id_airline}\} \rightarrow \{name\}$$

Tables and dependencies

- **Table GATE**

Attributes of *GATE*

$$GATE = (\underline{id_gate}, real_time, expected_time)$$

Functional Dependence of *GATE*

$$\{\underline{id_gate}\} \rightarrow \{real_time, expected_time\}$$

- **Table AIRPORT**

Attributes of *AIRPORT*

$$AIRPORT = (\underline{id_airport}, name)$$

Functional Dependence of *AIRPORT*

$$\{\underline{id_airport}\} \rightarrow \{name\}$$

Conceptual/logical model without normalization

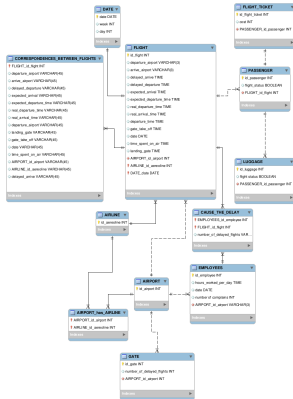


Figure 2: Model without normalization.

M. Molina, A. Riofrío, E.Vega, L.Rueda

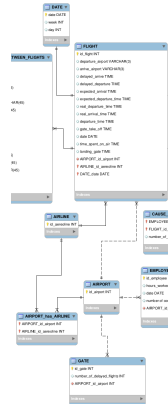


Figure 3: Tables with which we made the query

Database

Conceptual/logical model with normalization

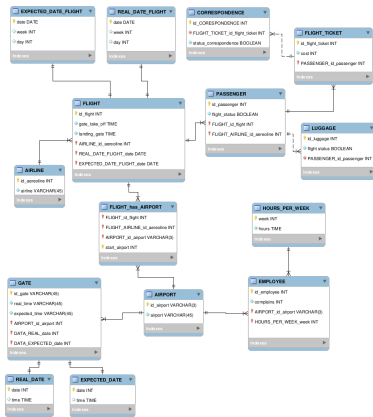


Figure 4: Model with normalization.

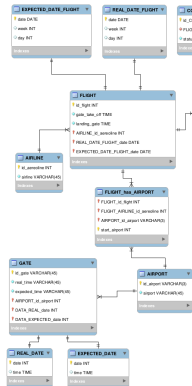


Figure 5: Tables with which we made the query

Database State

Query

Retrieve the start and end airports from every flight of TAME's group

FLIGHT-HAS-AIRPORT

| FLIGHT_id_flight | FLIGHT_AIRLINE_id_airline | AIRPORT_id_airport | start_airport |
|------------------|---------------------------|--------------------|---------------|
| 1 | 100 | ABL | 1 |
| 1 | 100 | ABQ | 0 |
| 2 | 104 | ABG | 1 |
| 2 | 104 | ABL | 0 |
| 3 | 105 | ABT | 1 |
| 3 | 105 | ABG | 0 |
| 4 | 106 | ABM | 1 |
| 4 | 106 | ABI | 0 |
| 5 | 107 | ABL | 1 |
| 5 | 107 | ABI | 0 |
| 6 | 108 | ABQ | 1 |
| 6 | 108 | ABT | 0 |

Figure 6: Possible database state for FLIGHT HAS AIRPORT table

SQL query

```
SELECT id_Flight, Start_Air, AIRPORT_id_airport AS Final_Air
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
) Start_A
where start_airport=0 AND Start_A.id_Airline=FLIGHT_AIRLINE_id_aereoline
```

Figure 7: MySql query.

| id_Flight | Start_Air | Final_Air |
|-----------|-----------|-----------|
| 1 | ABL | ABQ |
| 2 | ABG | ABL |
| 3 | AST | ABG |
| 4 | ABM | ABI |
| 5 | ABL | ABI |
| 6 | ABQ | AST |

Figure 8: MySql query.

Optimization of the query with algebra

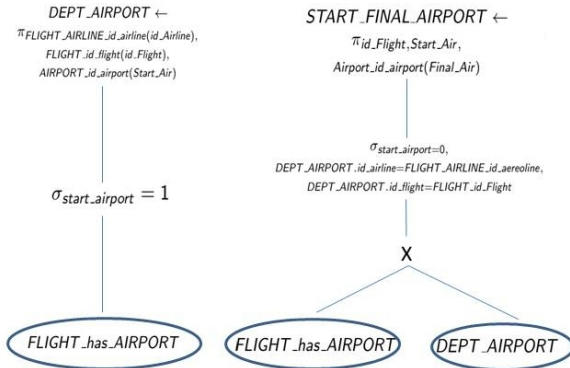


Figure 9: Optimization of start and final airports query.

Optimization of the query with algebra

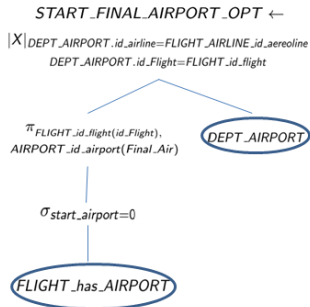
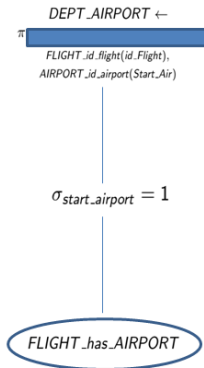
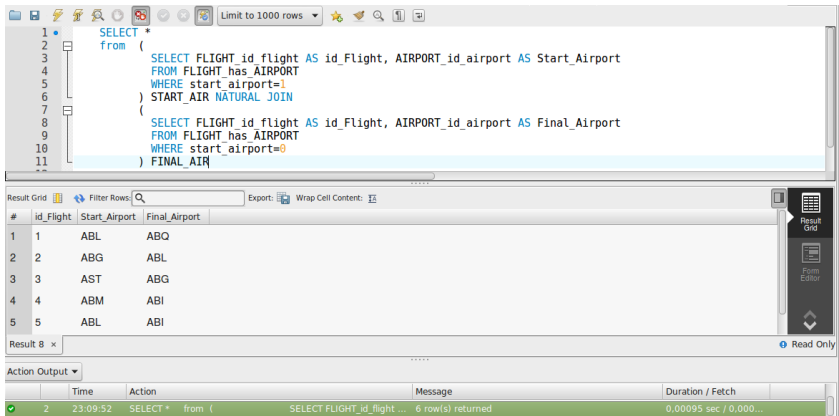


Figure 10: Optimization of start and final airports query.

Optimization of the query in MySQL



The screenshot shows the MySQL Workbench interface. The top pane contains a SQL query that uses a NATURAL JOIN to find flights starting and ending at different airports. The bottom pane displays the results of the query in a table format.

Query:

```

1 SELECT *
2 from (
3     SELECT FLIGHT_id flight AS id_Flight, AIRPORT_id_airport AS Start_Airport
4     FROM FLIGHT has_AIRPORT
5     WHERE start_airport=1
6 ) START_AIR NATURAL JOIN
7 (
8     SELECT FLIGHT_id flight AS id_Flight, AIRPORT_id_airport AS Final_Airport
9     FROM FLIGHT has_AIRPORT
10    WHERE start_airport=0
11 ) FINAL_AIR
    
```

Results:

| # | id_Flight | Start_Airport | Final_Airport |
|---|-----------|---------------|---------------|
| 1 | 1 | ABL | ABQ |
| 2 | 2 | ABG | ABL |
| 3 | 3 | AST | ABG |
| 4 | 4 | ABM | ABI |
| 5 | 5 | ABL | ABI |

The bottom pane also shows an "Action Output" table with the following data:

| | Time | Action | Message | Duration / Fetch |
|---|----------|---|-------------------|------------------------|
| ✓ | 23:09:52 | SELECT * from (SELECT FLIGHT_id flight ... | 6 row(s) returned | 0.00095 sec / 0.000... |

Figure 11: MySQL optimization of the start and final airports query.