DATABASE'S REPORT

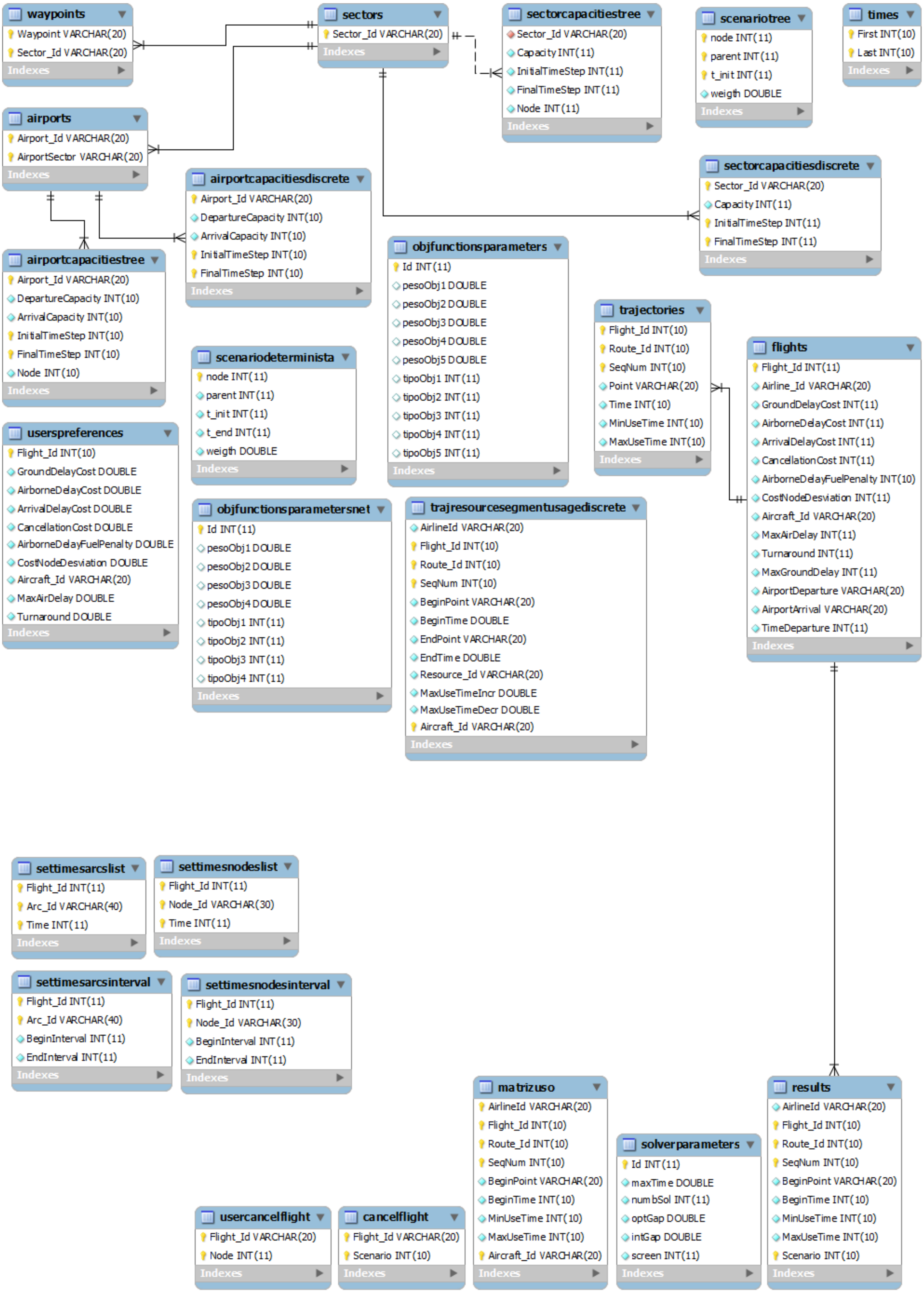
Author: Diego Ruiz Aguado

University Rey Juan Carlos
Madrid, Spain

Additional Information:

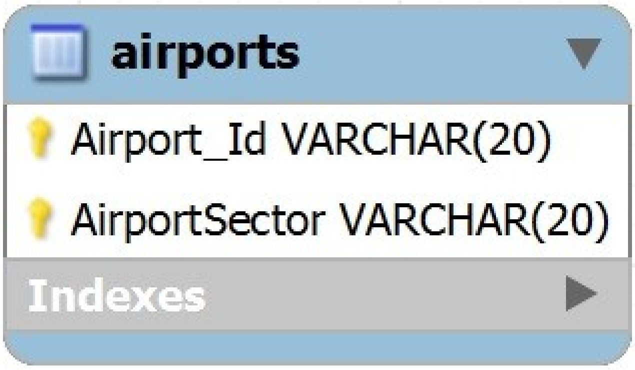
Database created with MySQL Server 5.5
in windows 7 Home Premium 64 bits.

EXTENDED ENTITY-RELATIONSHIP DIAGRAM



DATABASE'S TABLES

Airports



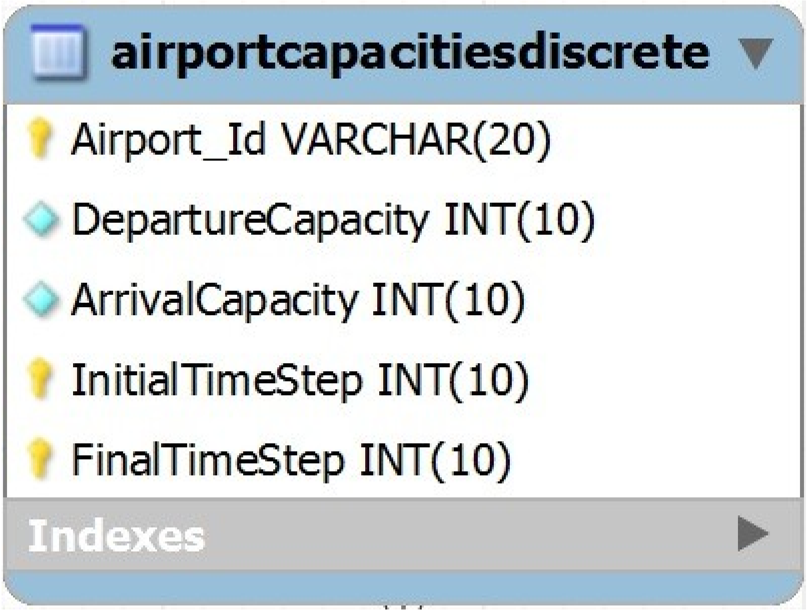
airports

Airport_Id VARCHAR(20)

AirportSector VARCHAR(20)

This table contains the identifier of each airport that is available for the model and the identifier of the sector to which it is connected. An airport can only be connected to a sector.

AirportCapacitiesDiscrete



airportcapacitiesdiscrete

Airport_Id VARCHAR(20)

DepartureCapacity INT(10)

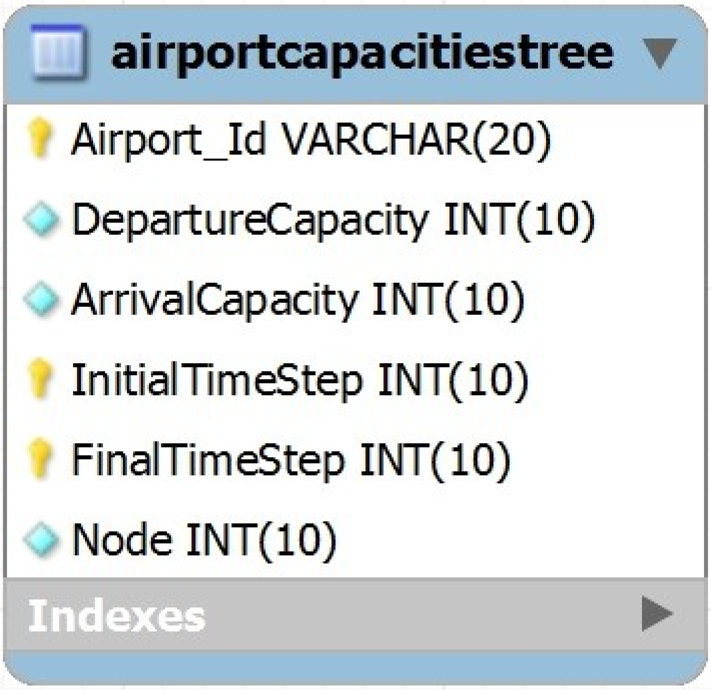
ArrivalCapacity INT(10)

InitialTimeStep INT(10)

FinalTimeStep INT(10)

This table contains all airports capacities involved in each route. Set a start time and end time for these airports also the values of landing capacities and take-off capacities can manage. These times can only be natural values including zero.

AirportCapacitiesTree



airportcapacitiestree

Airport_Id VARCHAR(20)

DepartureCapacity INT(10)

ArrivalCapacity INT(10)

InitialTimeStep INT(10)

FinalTimeStep INT(10)

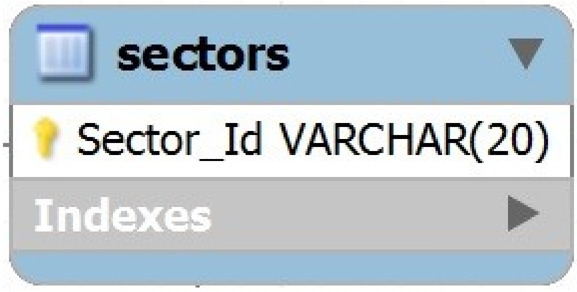
Node INT(10)

Indexes

This table contains the possible variation of the capacity of the airports of departure and arrival for different supernodes (variation occupies all periods of the supernode) or subsections of supernode (the variation which occupies only some of the periods of the supernode).

The values of DepartureCapacity and ArrivalCapacity can be negative, if you want decrease the original capacity or positive if you want increase it. This is the only case in which negative values can be entered into the fields of the database.

Sectors



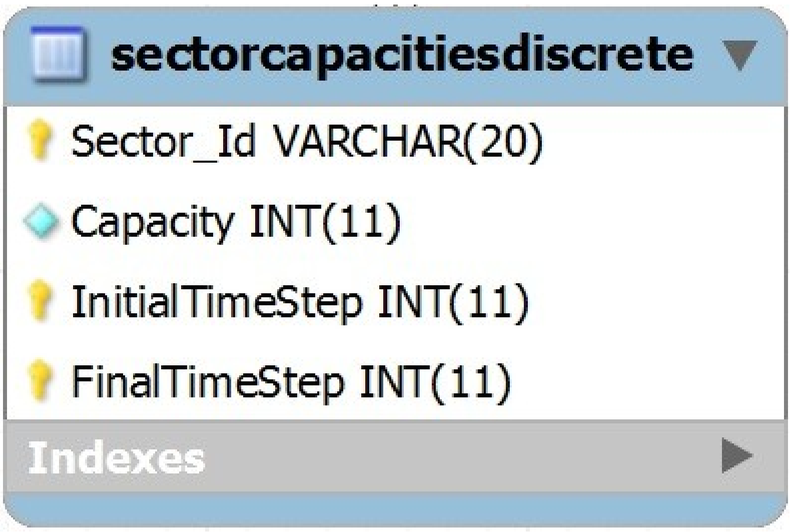
sectors

Sector_Id VARCHAR(20)

Indexes

This table collect the identifier of each sector available for the model.

SectorCapacitiesDiscrete



sectorcapacitiesdiscrete

Sector_Id VARCHAR(20)

Capacity INT(11)

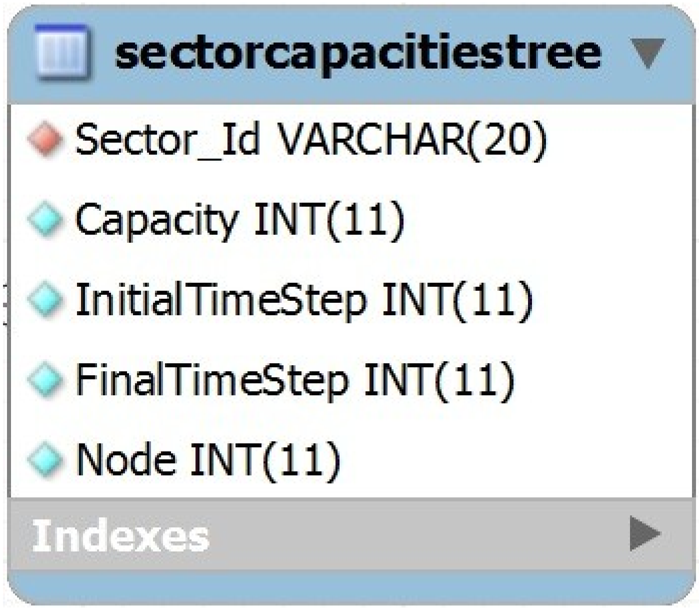
InitialTimeStep INT(11)

FinalTimeStep INT(11)

Indexes

This table contains the nominal capacity of the sectors.

SectorCapacitiesTree



sectorcapacitiestree

Sector_Id VARCHAR(20)

Capacity INT(11)

InitialTimeStep INT(11)

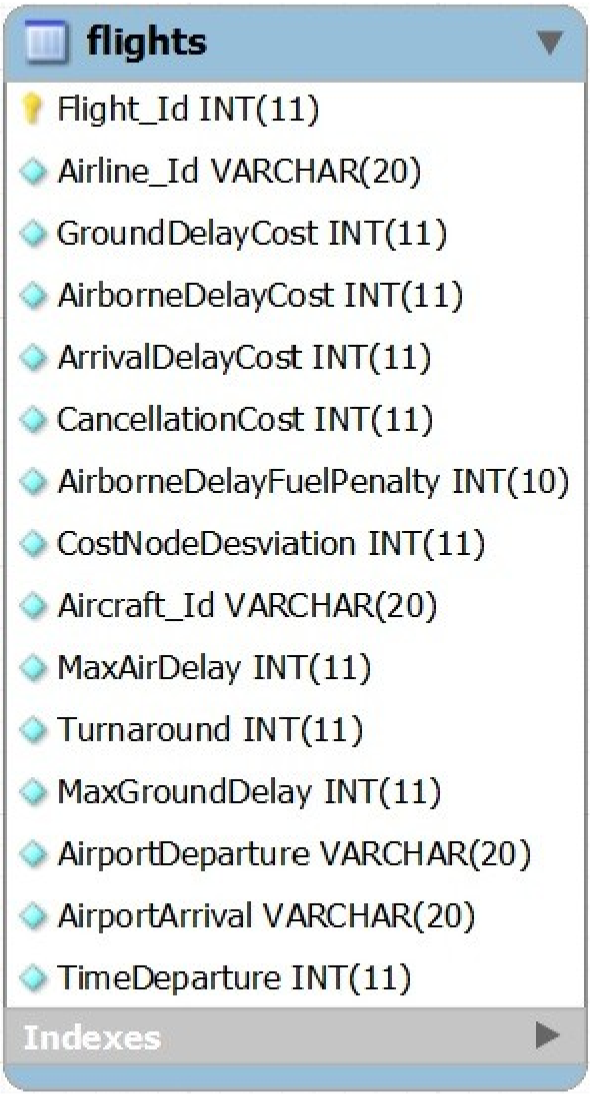
FinalTimeStep INT(11)

Node INT(11)

Indexes

This table contains the possible capacity variation of different sectors for supernodes (variation occupies all periods of supernode) or supernode's subsections (variation occupies only some of the periods of the supernode).

Flights

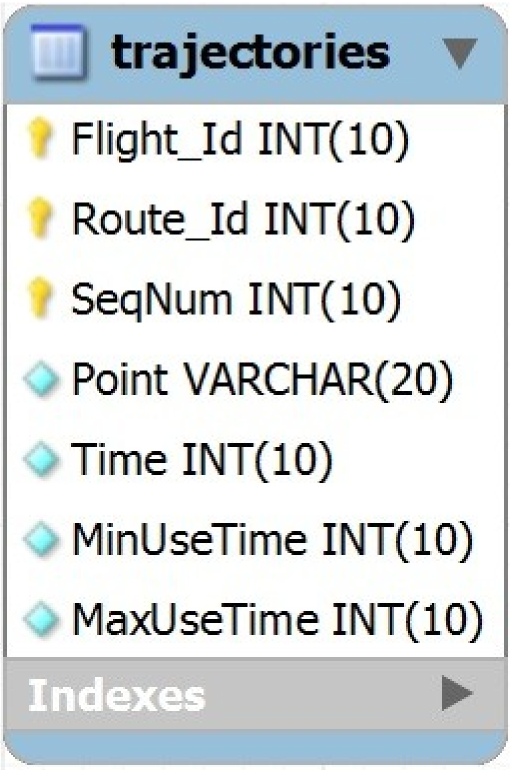


flights

Flight_Id INT(11)
Airline_Id VARCHAR(20)
GroundDelayCost INT(11)
AirborneDelayCost INT(11)
ArrivalDelayCost INT(11)
CancellationCost INT(11)
AirborneDelayFuelPenalty INT(10)
CostNodeDesviation INT(11)
Aircraft_Id VARCHAR(20)
MaxAirDelay INT(11)
Turnaround INT(11)
MaxGroundDelay INT(11)
AirportDeparture VARCHAR(20)
AirportArrival VARCHAR(20)
TimeDeparture INT(11)

This table contains information on each flight, such as its ID, the registration of the aircraft, the time of takeoff, besides delays and costs that could have the flight.

Trajectories

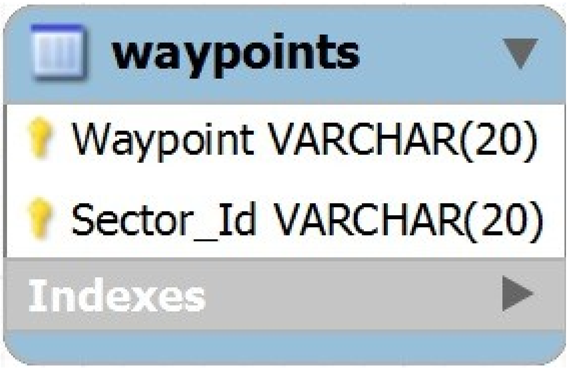


trajectories

Flight_Id INT(10)
Route_Id INT(10)
SeqNum INT(10)
Point VARCHAR(20)
Time INT(10)
MinUseTime INT(10)
MaxUseTime INT(10)

This table contains information on each route, such as its ID, the flight goes through that route and waypoints with times minimum, maximum and expected.

Waypoints

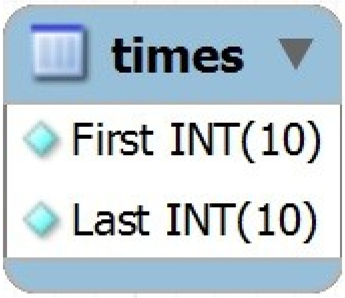


waypoints

Waypoint VARCHAR(20)
Sector_Id VARCHAR(20)

This table contains the information of the waypoints, such as its ID and sector where is it located.

Times



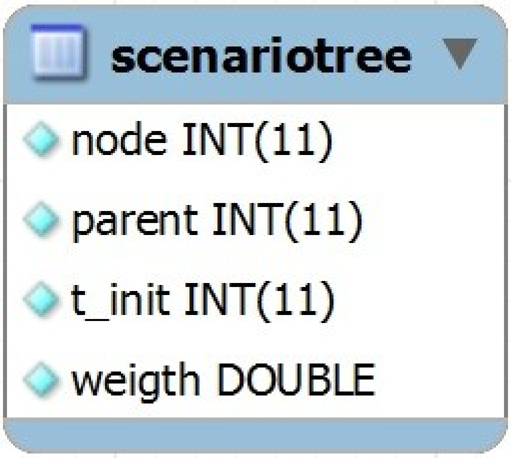
times

First INT(10)

Last INT(10)

This table contains the information of the start and end stages of the model. These values can only be natural including zero.

ScenarioTree



scenariotree

node INT(11)

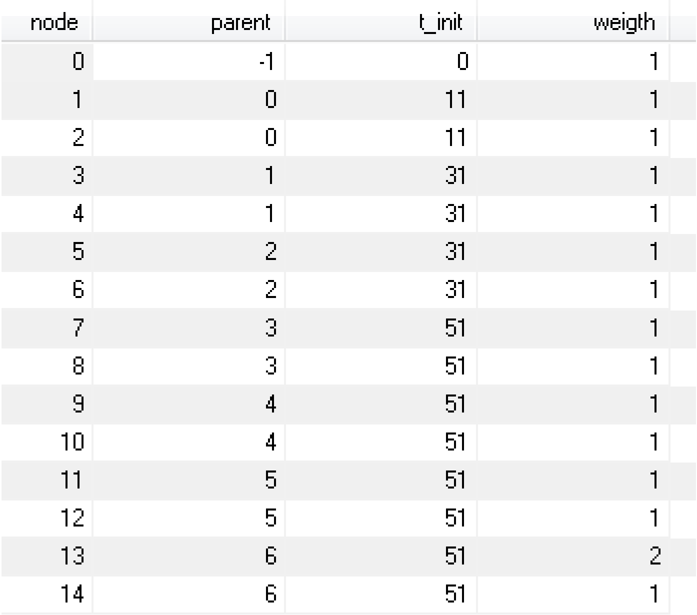
parent INT(11)

t_init INT(11)

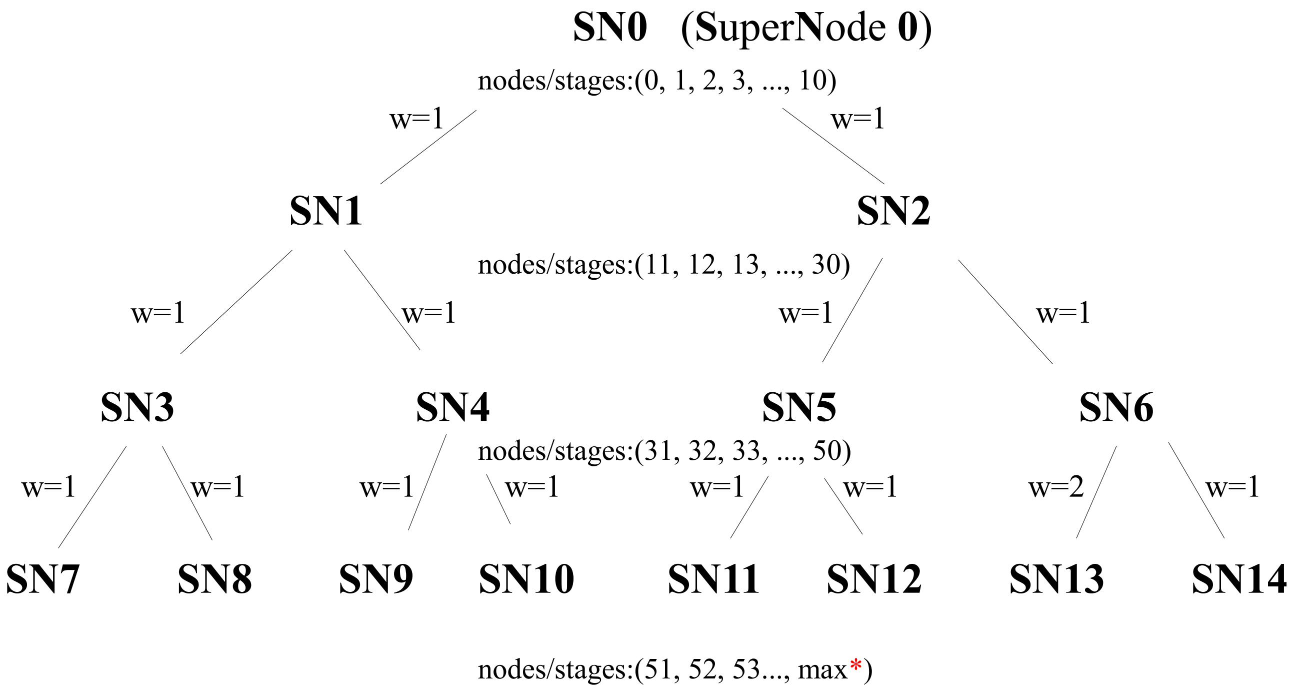
weigth DOUBLE

This table contains the information required for construction of the scenario tree model, such as the nodes that form, what are the parents of the node, the time step in initiating and weight is in the tree.

Example of binary tree for *linear programming* model → ATFMPUU.java:

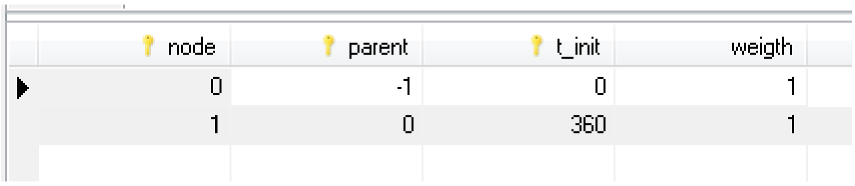


node	parent	t_init	weigth
0	-1	0	1
1	0	11	1
2	0	11	1
3	1	31	1
4	1	31	1
5	2	31	1
6	2	31	1
7	3	51	1
8	3	51	1
9	4	51	1
10	4	51	1
11	5	51	1
12	5	51	1
13	6	51	2
14	6	51	1


SN0 (SuperNode 0)
nodes/stages:(0, 1, 2, 3, ..., 10)
w=1
w=1
SN1
SN2
nodes/stages:(11, 12, 13, ..., 30)
w=1
w=1
w=1
w=1
SN3
SN4
SN5
SN6
nodes/stages:(31, 32, 33, ..., 50)
w=1
w=1
w=1
w=1
w=1
w=1
w=2
w=1
SN7
SN8
SN9
SN10
SN11
SN12
SN13
SN14
nodes/stages:(51, 52, 53..., max*)

* max is the value contained in the field *Last* of the table *Times*

Example of linear tree for *Network* model → NetsMain.java:



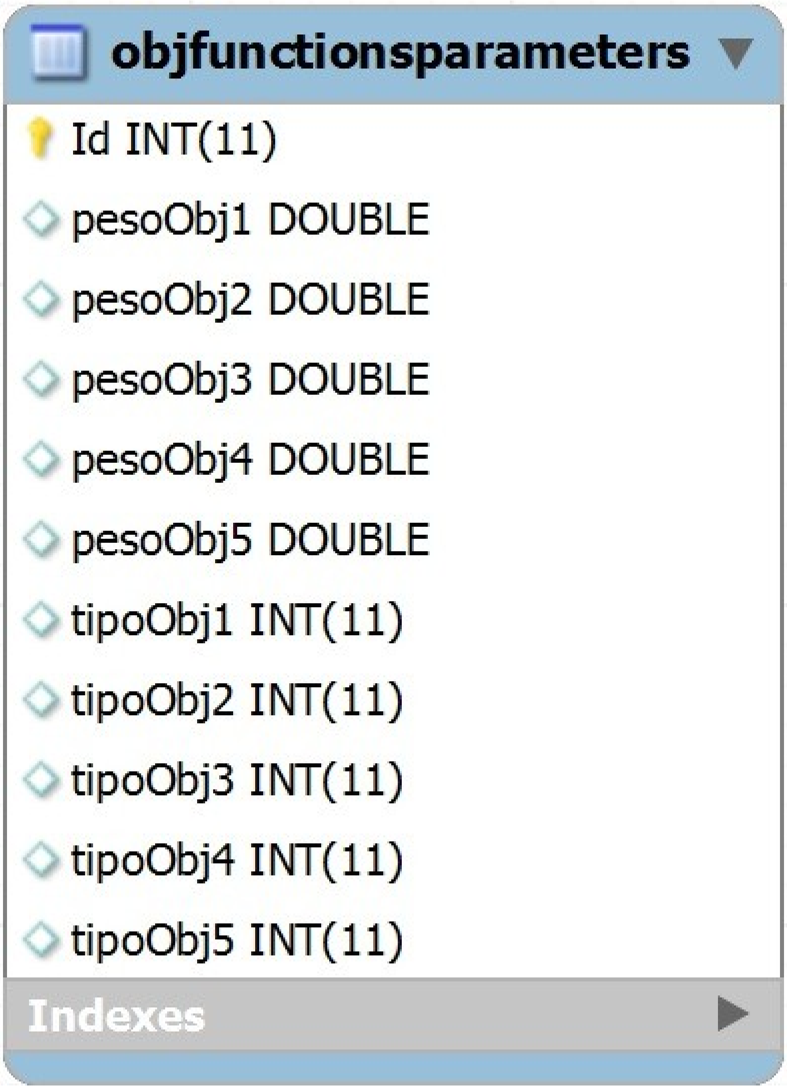
node	parent	t_init	weigth
0	-1	0	1
1	0	360	1

SN0 (Super**N**ode **0**) ------------------- **SN1** (Super**N**ode **1**)
nodes/stages:(0, 1, 2, 3, ..., 360) nodes/stages:(360*)
w=1 w=1

In case of a linear tree add a row where t_init equals max (field *Last* of the table *Times*).

* In this case 360 = max

ObjFunctionsParameters

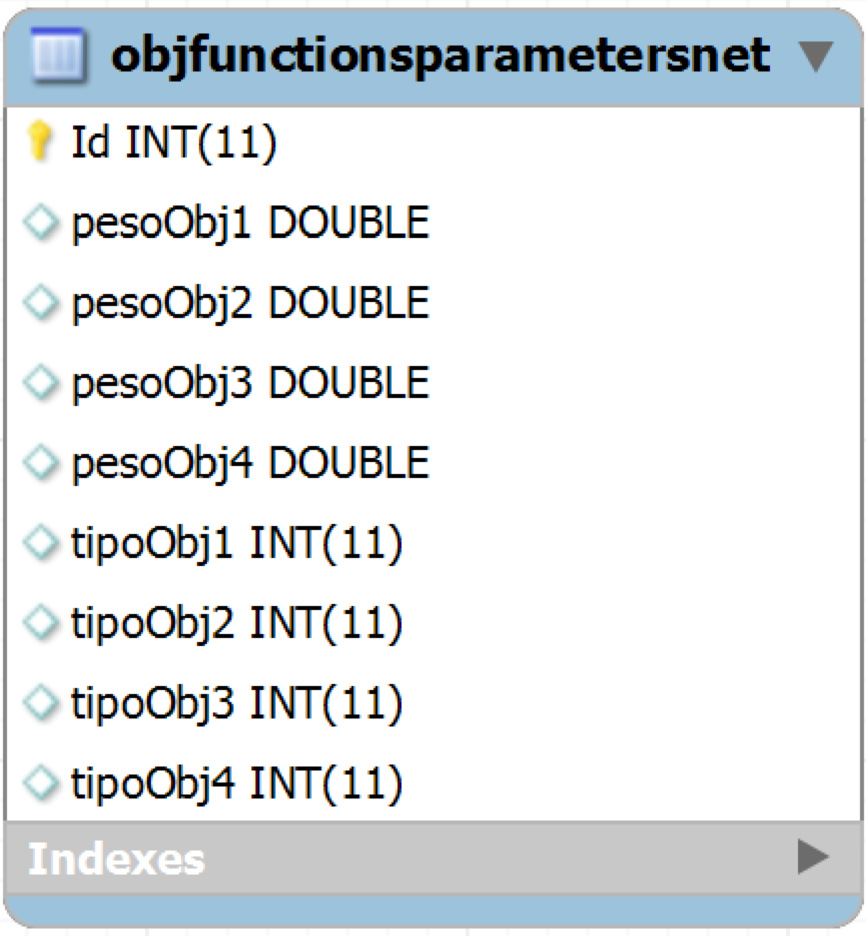


objfunctionsparameters

Id INT(11)
pesoObj1 DOUBLE
pesoObj2 DOUBLE
pesoObj3 DOUBLE
pesoObj4 DOUBLE
pesoObj5 DOUBLE
tipoObj1 INT(11)
tipoObj2 INT(11)
tipoObj3 INT(11)
tipoObj4 INT(11)
tipoObj5 INT(11)

This table contains the information required to parameterize the objective function of the model (Only for *linear programming* model → ATFMPUU.java).

ObjFunctionsParametersNet



objfunctionsparametersnet

Id INT(11)

pesoObj1 DOUBLE

pesoObj2 DOUBLE

pesoObj3 DOUBLE

pesoObj4 DOUBLE

tipoObj1 INT(11)

tipoObj2 INT(11)

tipoObj3 INT(11)

tipoObj4 INT(11)

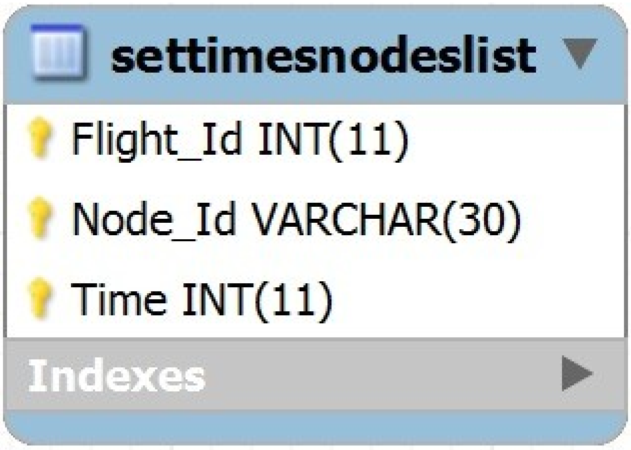
This table contains the information required to parameterize the objective function of the model (Only for *Network* model → NetsMain.java)

The four tables below may not appear initially exist in the database which are created within the execution of the program, so you can use a database without these four tables and have the program automatically create and fill.

¡Attention!, only two cases can occur:

1. the four tables are created and fill,
2. or that any of them does not exist in the database and therefore the database must pass through a first program execution

SetTimesNodesList



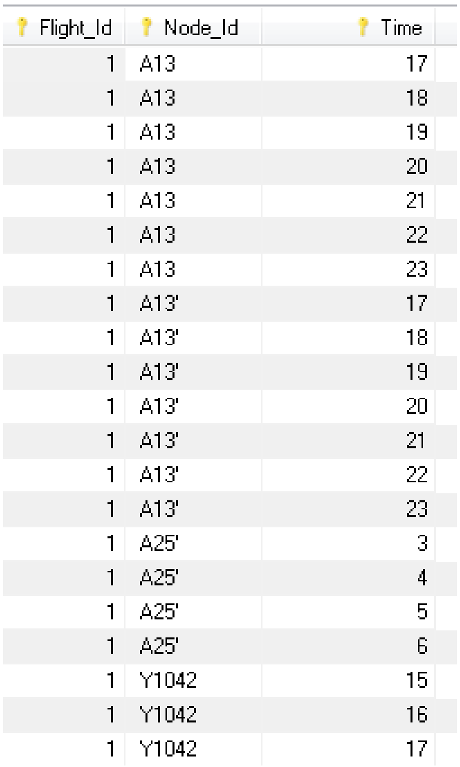
settimesnodeslist

Flight_Id INT(11)

Node_Id VARCHAR(30)

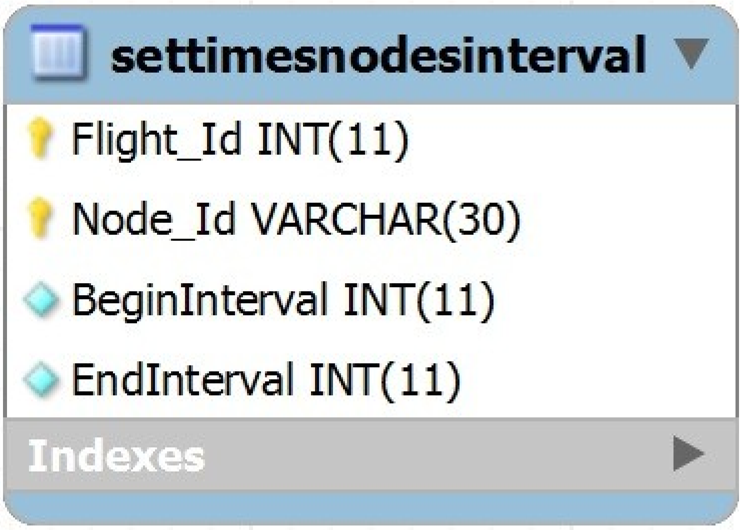
Time INT(11)

This table contains the values calculated from the sets of time associated with the nodes ordered in the graph.



Flight_Id	Node_Id	Time
1	A13	17
1	A13	18
1	A13	19
1	A13	20
1	A13	21
1	A13	22
1	A13	23
1	A13'	17
1	A13'	18
1	A13'	19
1	A13'	20
1	A13'	21
1	A13'	22
1	A13'	23
1	A25'	3
1	A25'	4
1	A25'	5
1	A25'	6
1	Y1042	15
1	Y1042	16
1	Y1042	17

SetTimesNodesInterval



settimesnodesinterval

Flight_Id INT(11)

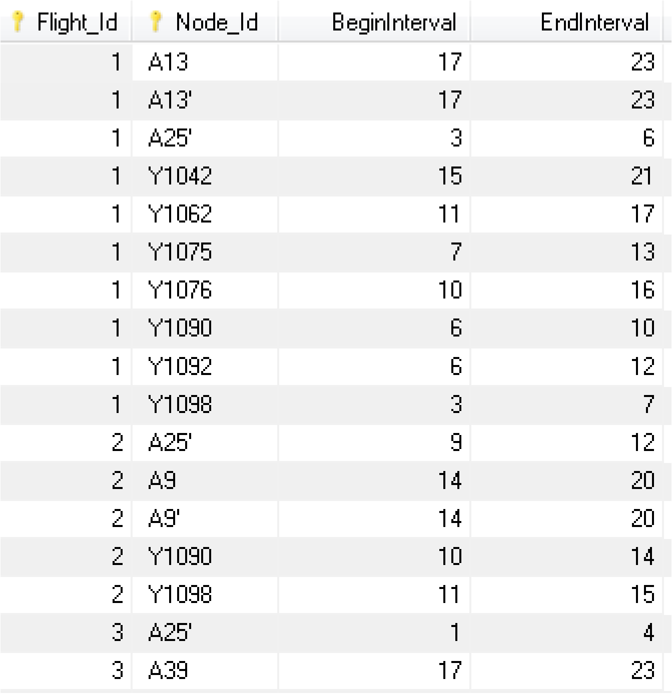
Node_Id VARCHAR(30)

BeginInterval INT(11)

EndInterval INT(11)

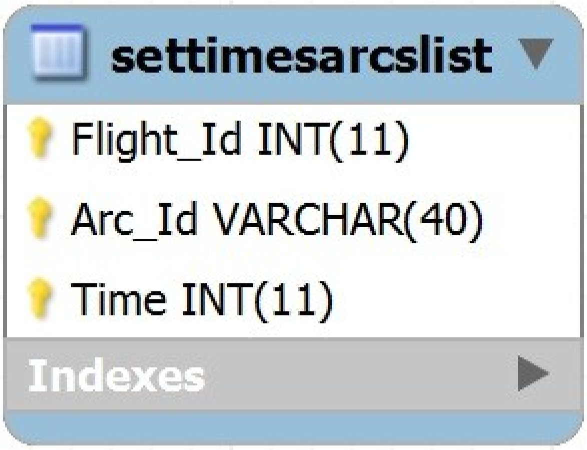
Indexes

This table contains the interval associated with the nodes of graph. The values were calculated from that time's sets.



Flight_Id	Node_Id	BeginInterval	EndInterval
1	A13	17	23
1	A13'	17	23
1	A25'	3	6
1	Y1042	15	21
1	Y1062	11	17
1	Y1075	7	13
1	Y1076	10	16
1	Y1090	6	10
1	Y1092	6	12
1	Y1098	3	7
2	A25'	9	12
2	A9	14	20
2	A9'	14	20
2	Y1090	10	14
2	Y1098	11	15
3	A25'	1	4
3	A39	17	23

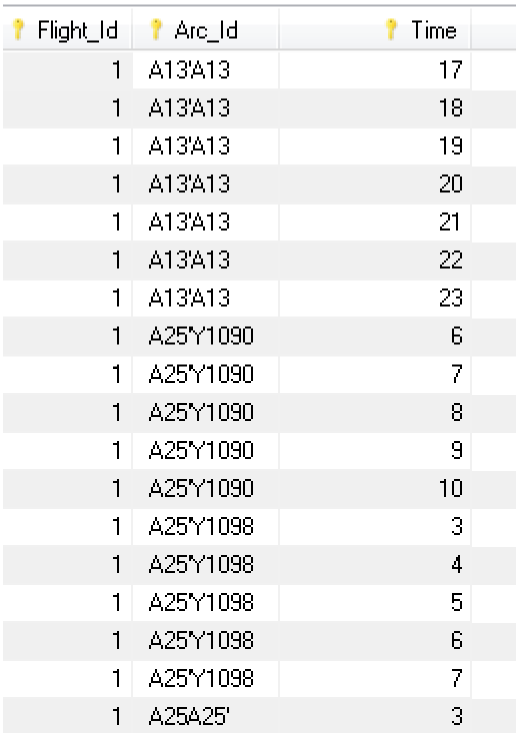
SetTimesArcsList



settimesarcslist

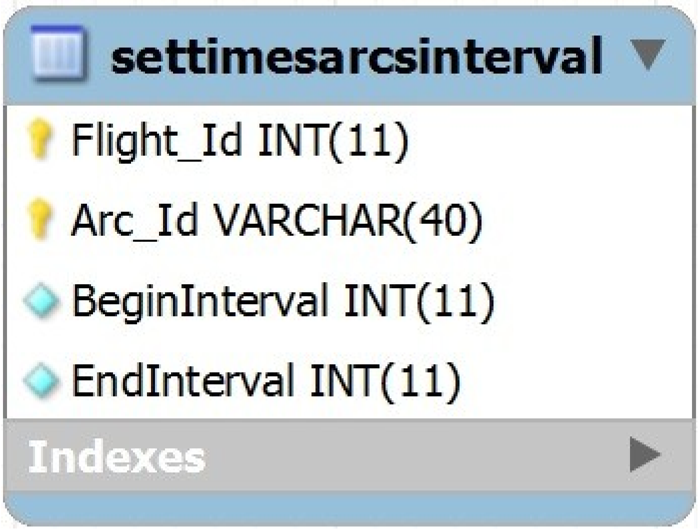
- Flight_Id INT(11)
- Arc_Id VARCHAR(40)
- Time INT(11)

This table contains the values calculated from the sets of time associated with the arcs in the graph.

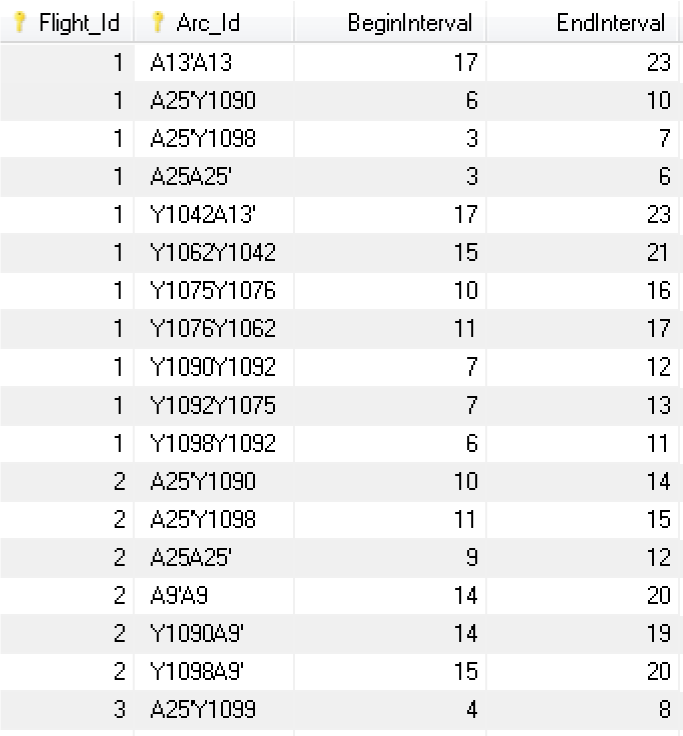


Flight_Id	Arc_Id	Time
1	A13'A13	17
1	A13'A13	18
1	A13'A13	19
1	A13'A13	20
1	A13'A13	21
1	A13'A13	22
1	A13'A13	23
1	A25''Y1090	6
1	A25''Y1090	7
1	A25''Y1090	8
1	A25''Y1090	9
1	A25''Y1090	10
1	A25''Y1098	3
1	A25''Y1098	4
1	A25''Y1098	5
1	A25''Y1098	6
1	A25''Y1098	7
1	A25A25'	3

SetTimesArcsInterval



This table contains the interval associated with the arcs of graph. The values were calculated from that time's sets.

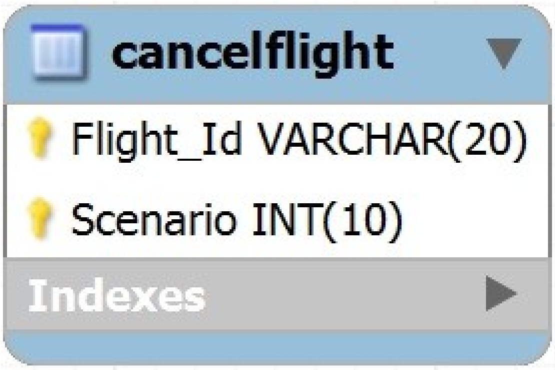


Flight_Id	Arc_Id	BeginInterval	EndInterval
1	A13'A13	17	23
1	A25'Y1090	6	10
1	A25'Y1098	3	7
1	A25A25'	3	6
1	Y1042A13'	17	23
1	Y1062Y1042	15	21
1	Y1075Y1076	10	16
1	Y1076Y1062	11	17
1	Y1090Y1092	7	12
1	Y1092Y1075	7	13
1	Y1098Y1092	6	11
2	A25'Y1090	10	14
2	A25'Y1098	11	15
2	A25A25'	9	12
2	A9'A9	14	20
2	Y1090A9'	14	19
2	Y1098A9'	15	20
3	A25'Y1099	4	8

OTHES TABLES

These tables come from the previous model of the database, mostly are output tables, namely, where information was stored after running the program. For the new model have not used but have remained useful in the future. Then their descriptions:

CancelFlight



cancelflight

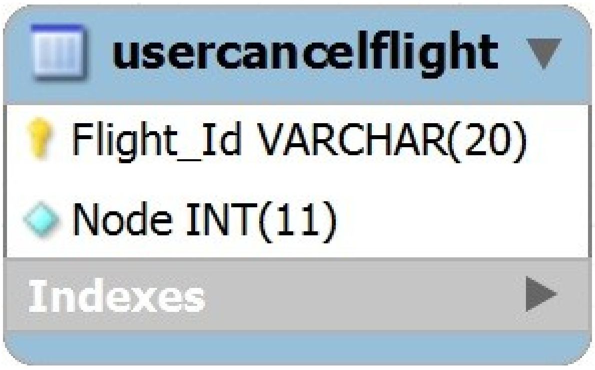
Flight_Id VARCHAR(20)

Scenario INT(10)

Indexes

It's an output table. We indicate the flights were canceled for each scenario. It starts as empty table.

UserCancelFlight



usercancelflight

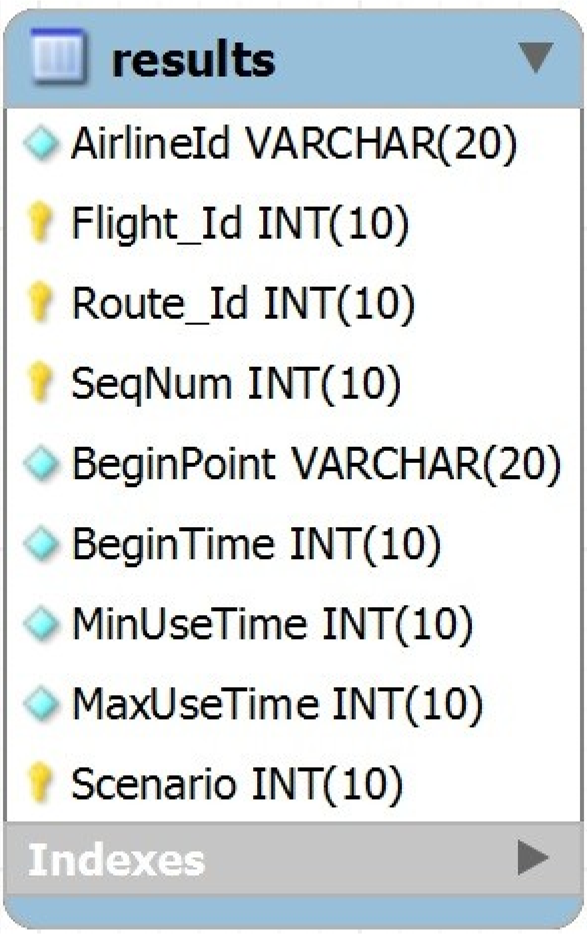
Flight_Id VARCHAR(20)

Node INT(11)

Indexes

It's an output table. We indicate the flights were canceled for each node. It starts as empty table.

Results



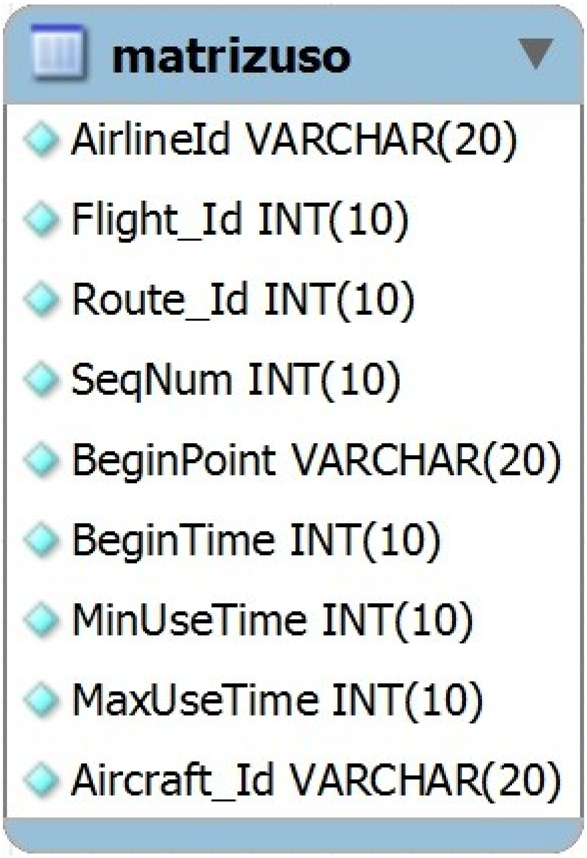
results

AirlineId VARCHAR(20)
Flight_Id INT(10)
Route_Id INT(10)
SeqNum INT(10)
BeginPoint VARCHAR(20)
BeginTime INT(10)
MinUseTime INT(10)
MaxUseTime INT(10)
Scenario INT(10)

Indexes

It's an output table. We indicate the flight path for each scenario. It starts as empty table.

MatrizUso



matrizuso

AirlineId VARCHAR(20)

Flight_Id INT(10)

Route_Id INT(10)

SeqNum INT(10)

BeginPoint VARCHAR(20)

BeginTime INT(10)

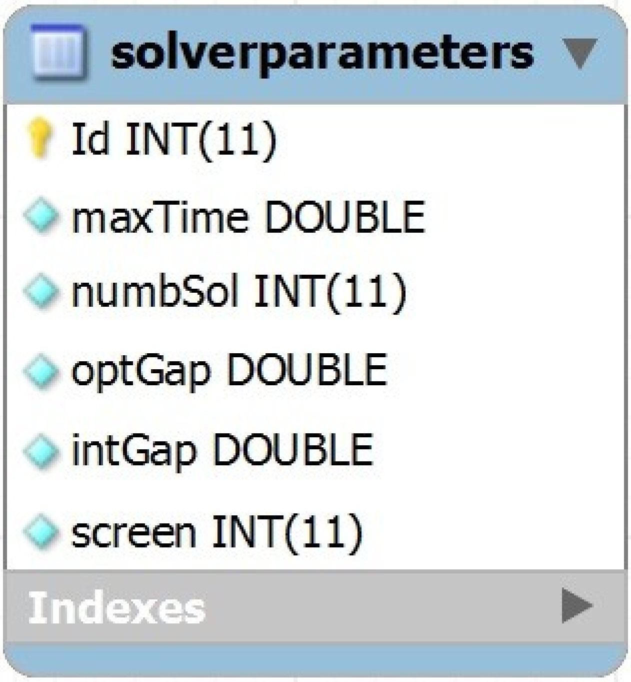
MinUseTime INT(10)

MaxUseTime INT(10)

Aircraft_Id VARCHAR(20)

It is an output table. Support table used in the previous model to test programming. The information was stored during program execution. Initially would be empty. The table would give us information where are the flights when there were no capacity issues.

SolverParameters



solverparameters

Id INT(11)

maxTime DOUBLE

numbSol INT(11)

optGap DOUBLE

intGap DOUBLE

screen INT(11)

Indexes

This table contains the variable parameters to solve the problem with CPLEX.