**Databases (**[**INF.01014UF**](javascript:subTopicClickM(%22c0508141242_34.htm%22,%22Examination%20(INF.01014UF%20Databases)%22,%22%22))**) – Report**

Hojun, Cho ( )

Dayeong, Lee ( 11739321 )

Table of Content

1. Database schema **1**
2. Functional dependencies 2
3. Current State of the database3
4. Database queries in terms of the Relational Algebra**4**
5. Database queries in terms of the Relational Calculus 5
6. Database queries in terms of the SQL6
7. Practical implementation of the database with SQL **4**
8. Database queries in terms of the Relational Calculus 5
9. Database queries in terms of the SQL6
10. Database schema

**Domain :**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| CID | Int | The client ID |
| LID | Int | The airline ID |
| TDTime | Datetime | The departure time |
| TATime | Datetime | The arrival time |
| PDID | Int | The departure airport ID number |
| PAID | Int | The arrival airport ID nubmer |
| TPrice | Int | The price of the airticket |
| TSeat | Int | The seat |
| PID | Int | The airport ID number |
| Pname | String | The airport name |
| PCity | String | The airport location city |
| LID | Int | The airline ID number |
| LName | String | The airline name |
| LUnion | String | The air union of airline |
| LNation | String | The nationality of airline |
| CName | String | The name of client |
| CAge | Int | The age of client |
| CGender | Boolean | The gender of client |
| CNation | String | The nationality of client |

\*\*\* The domain of PID, PDID and PAID are same. \*\*\*

* **Relation : airticket** ( CID, LID, TDTime, TATime, TPrice, TSeat, PDID, PAID )
* **Relation : airline** ( LID, LName, LUnion, LNation )
* **Relation : airport** ( PID, PName, PCity )
* **Relation : client** ( CID, CName, CAge, CGender, CNation )

1. Functional Dependencies

* **Relation : airticket** ( CID, LID, TDTime, TATime, TPrice, TSeat, PDID, PAID )  
  (CID, LID, TDTIme) → TATime  
  (CID, LID, TDTIme) → TPrice  
  (CID, LID, TDTIme) → TSeat  
  (CID, LID, TDTIme) → TDID  
  (CID, LID, TDTIme) → TAID
* **Relation : airline** ( LID, LName, LUnion, LNation )  
  LID → LName  
  LID → LUnion  
  LID → LNation
* **Relation : airport** ( PID, PName, PCity )  
  PID → PName  
  PID → PCity
* **Relation : client** ( CID, CName, CAge, CGender, CNation )  
  CID → CName  
  CID → CAge  
  CID → CGender  
  CID → CNation

**AirTicketDB is in the 3rd normal form:**

* To satisfy the 3rd normal form, non-primary keys should be dependent only on the primary key set, not on the subset of the primary key. That is, there should be no dependency between non-primary keys. Also, it should be in the the 2nd normal form.
* In the case of client, non-primary keys are CName, CAge, CGender, CNation. They are dependent only on CName. Also, there is no dependency between them.
* In the case of airticket, airline and airport, all of them satisfies the condition of 3rd normal form like the case of client.
* Therefore, Our DB, AirTicketDB, satisfies the condition of the 3rd normal form.

1. Current state of the database
2. Database queries in terms of the Relational Algebra

**Get the the locations of airport, which the clients, whose name are ‘Hojun’, visit.**

Select client Where CName = ‘Hojun’ Giving A;

Join A and airticket over CID Giving B;

Join B and airport over airport.PID = B.PDID Giving D;

Join B and airport over airport.PID = B.PAID Giving E;

D Union E Giving F;

Project F over PCity Giving Result;

**Get the names of client, who have ever used both of airline ‘korean air’ and ‘Japan air’.**

Select airline Where LName = ‘korean air’ Giving A;

Join A and airticket over LID Giving B;

Select airline Where LName = ‘Japan air’ Giving C;

Join C and airticket over LID Giving D;

B Intersect D Giving E;

Join E and client over CID Giving F;

Project F over CName Giving Result;

**Get the names of client, who have ever arrived at the airport ‘Vienna’ and have not ever arrived at the airport ‘Graz’**

Select airport Where PName = ‘Vienna’ Giving A;

Join A and airticket Over A.PID = airticket.PAID Giving B;

Select airport Where PName = ‘Graz’ Givng C;

Join C and airticket Over C.PID = airticket.PAID Giving D;

B Minus D Giving E;

Join E and client over CID Giving F;

Project F Over CName Giving Result;

1. Database queries in terms of the Relational Calculus

**Get the seat number of the ticket, of which the client is ‘hojun’, departure time is ‘2018-07-07’ and airline is ‘korean air’.**

T -> ticket

C -> client

L -> airline

(T.TSeat) : ( T.CID = C.CID & T.TDTime = ‘2018-07-07’ & T.LID = L.LID & C.CName = ‘hojun’ & L.LName = ‘korean air’)

**Get the lowest price of the ticket, of which the owner is ‘hojun’ and the airline is ‘korean air’**

T1 -> airticket

T2 -> airticket

C -> client

L -> airline

(T1.TPrice) : ∀T2 ∃C ∃L ( C.CName = ‘hojun’ & L.LName = ‘korean air’ & T1.CID = C.CID & T1.LID = L.LID & T2.CID = C.CID & T2.LID = L.LID & T1.TPrice <= T2.TPrice)

**Get the highest seat number of the ticket, of which the departure airport is ‘Vienna’, the arrival airport is ‘Graz’ and the airline is ‘LOT’.**

T1 -> airticket

T2 -> airticket

P1 -> airport

P2 -> airport

L -> airline

(T1.TSeat) : ∀T2 ∃P1 ∃P2 ∃L ( T1.PDID = P1.PID & T2.PDID = P1.PID & T1.PAID = P2.PID & T2.PAID = P2.PID & T1.LID = L.LID & T2.LID = L.LID & P1.PName = ‘Vienna’ & P2.PName = ‘Graz’ & L.LName = ‘LOT’ & T1.TSeat >= T2.TSeat)

1. Database queries in terms of the SQL

**Get the client names, who has a ticket of which Korean air mileage is 4000.**

Select CName from client where CID IN

( Select CID from airticket where LID IN

( Select LID from airline where LName = 'Korean Air')

AND TMileage =4000);

**Get the client names, whose ‘SKYTEAM’ Union mileage sum is over 5000;**

Select CName from client where CID IN

( Select A.CID from

( Select \* from airticket where LID IN

( Select LID from airline where LUnion = 'SKYTEAM')

) A GROUP BY CID having sum(TMileage) >=5000

);

**Get the number of male and female, who arrive at the airport 'Incheon International Airport’.**

Select CGender, count(CID) from client where CID IN

( Select CID from airticket where PDID IN

( Select PID from airport

where PName = 'Incheon International Airport')

)GROUP BY CGender;

1. Database queries in terms of the SQL without nested SQL Blocks

**Get the client names, whose Korean air mileage is 4000.**

Select CName from airline, airticket, client where

client.CID = airticket.CID AND airticket.LID = airline.LID

AND airline.LName = 'Korean Air' AND airticket.TMileage = 4000;

**Get the client names, whose ‘SKYTEAM’ Union mileage sum is over 5000;**

Select CName from airline, airticket, client where

airline.LUnion = 'SKYTEAM' AND airticket.LID = airline.LID

AND airticket.CID = client.CID

GROUP BY client.CID having sum(TMileage) >= 5000;

**Get the number of male and female, who arrive at the airport 'Incheon International Airport’.**

Select CGender, count(DISTINCT client.CID)

from airticket, airport, client where

airticket.PDID = airport.PID AND airticket.CID = client.CID

AND airport.PName = 'Incheon International Airport'

GROUP BY CGender;

1. Practical implementation of the database with SQL

CREATE

Dsaf

Asdfsa