Umeå University
Department of Computing Science

Introduction to Database Managment 7.5 p
5DV119

Exercises, Chapter/Topic 3

Submitted	2017-02-21
Author:	Lorenz Gerber (`dv15lgr@cs.umu.se` `lozger03@student.umu.se`)
Instructor:	Jan Erik Moström / Michael Minock / Filip Allberg / Carl-Anton Anserud

Query 1

Find the code of each airport which is located either in Greece or else in Germany

Relational Algebra

$X_1 \leftarrow \sigma_{(Country=\text{'}Germany\text{'})\wedge(Country=\text{'}Greece\text{'})}(Airport)$
$X_2 \leftarrow \pi_{(Code)}(X_1)$

Relational Tuple Calculus

$\{a.Code|Airport(a)$
$\wedge((a.Country = \text{'}Germany\text{'}) \vee (a.Country = \text{'}Greece\text{'}))\}$

Query 2

Find the name and abbreviation of each airline which has a flight with destination the airport with the code ‘TXL’ but no flight with destination the airport with code ‘SXF’

Relational Algebra

$X_1 \leftarrow \sigma_{Destination=\text{'}TXL\text{'}}(Flight)$
$X_2 \leftarrow \sigma_{Destination=\text{'}SXF\text{'}}(Flight)$
$X_3 \leftarrow \pi_{(Airline)}(X_1)$ $X_4 \leftarrow \pi_{(Airline)}(X_2)$ $X_5 \leftarrow X_3 \backslash X_4$
$X_6 \leftarrow X_5 \bowtie_{Airline=Abbreviation} Airline$
$X_7 \leftarrow \pi_{Name,Abbreviation}(X_6)$

Relational Tuple Calculus

$\{a.Name, a.Abbreviation|Airline(a)$
$\wedge(\exists f_1)(\forall f_2)(Flights(f_1) \wedge Flights(f_2)$
$\wedge(f_1.Airline = a.Abbreviation) \wedge (f_2.Airline = a.Abbreviation)$
$\wedge(f_1.Destination = \text{'}TXL\text{'})$
$\wedge(f_2.Destination \neq \text{'}SXF\text{'}))\}$

Query 3

Find the name and abbreviation of those airlines which do not have any flights to an airport in Germany or France

Relational Algebra

$X_1 \leftarrow Flight \bowtie_{(Destination=Code)} Airport$
$X_2 \leftarrow \sigma_{(Country=France)\vee(Country=Germany)}(X_1)$
$X_3 \leftarrow \pi_{Airline}(X_2)$
$X_4 \leftarrow \pi_{Abbreviation}(Airline)$

$X_5 \leftarrow X_4 \backslash X_3$
$X_6 \leftarrow X_5 \bowtie_{X_5.Abbreviation=Airline.Abbreviation} Airline$
$X_7 \leftarrow \pi_{Abbreviation,Name}(X_6)$

Relational Tuple Calculus

$\{a.Name, a.Abbreviation | Airline(a)$
$\wedge(\forall f)((Flight(f)$
$\wedge(a.Abbreviation = f.Airline))$
$\Rightarrow (\forall ap)(Airport(ap) \wedge (ap.Code = f.Destination)$
$\wedge((ap.Country \neq \text{'}Germany\text{'})$
$\vee(ap.Country \neq \text{'}France\text{'})))))\}$

Query 4

Find the codes of those airports which have flights to every airport in France. (Note that no French airport will normally qualify because, for example, there is no flight from 'CDG' to 'CDG'.)

Relational Algebra

$X_1 \leftarrow Airport \bowtie_{(Code=Destination)} Flight$
$X_2 \leftarrow \sigma_{(Country=\text{'}France\text{'})}(X_1)$
$X_3 \leftarrow \pi_{(Origin,Destination)}(X_2)$
$X_4 \leftarrow \sigma_{(Country=\text{'}France\text{'})}(Airport)$
$X_5 \leftarrow \pi_{(Origin)}(X_2)$
$X_6 \leftarrow \pi_{(Code)}(X_4)$
$X_7 \leftarrow X_5 \times X_6$
$X_8 \leftarrow X_7 \backslash X_3$
$X_9 \leftarrow \pi_{(Origin)}(X_8)$
$X_{10} \leftarrow X_5 \backslash X_9$

Relational Tuple Calculus

$\{a.Code | Airport(a) \wedge ((\forall ap)(\neg Airport(ap)) \vee \neg(ap.Country = \text{'}France\text{'})$
$\vee((\exists f)(Flight(f) \wedge (f.Origin = a.Code) \wedge (f.Destination = ap.Code))))\}$

Query 5

Find the codes of those airports which have departures (i.e. fligths with origin at that airport) for exactly two distinct airlines.

Relational Algebra

$X_1 \leftarrow \sigma_{F_1}(Flight)$
$X_2 \leftarrow \sigma_{F_2}(Flight)$
$X_3 \leftarrow X_1 \bowtie_{F_1.Airline \neq Airline) \wedge (F_1.Origin = Origin)} Flight$
$X_4 \leftarrow X_2 \bowtie_{F_2.Airline \neq Airline) \wedge (F_2.Origin = Origin)} X_3$
$X_5 \leftarrow X_3 \setminus X_4$
$X_6 \leftarrow \pi_{(Origin)} X_5$

Relational Tuple Calculus

$\{a.Code | Airport(a)$
$\wedge (\exists F_1)(\exists F_2)((Flight(F_1) \wedge Flight(F_2)$
$\wedge (F_1.Origin = a.Code)$
$\wedge (F_1.Origin = F_2.Origin))$
$\wedge (F_1.Airline \neq F_2.Airline))$
$\wedge (\forall F_1)(\forall F_2)(\forall F_3)(Flight(F_1) \wedge Flight(F_2) \wedge Flight(F_3)$
$\wedge (F_1.Origin = F_2.Origin) \wedge (F_1.Origin = F_3.Origin) \wedge (F_1.Origin = a.Code))$
$\Rightarrow ((F_1.Airline = F_2.Airline) \vee (F_1.Airline = F_3.Airline) \vee (F_2.Airline = F_3.Airline))\}$