CS2102/S and IT2002 Tutorial Solution

* Relational Algebra

*P* → Pizza(*code, pname, size*)

*S* → Store(*sname, area, phone*)

*L* → Sells(*sname, code, price*)

1. Find the names of pizzas that come in a 10 inch size.

πpname(σsize=10(P))

2. Find the names of pizzas that come in a 10 inch or a 12 inch size.

πpname(σsize=10∨size=12(P))

3. Find the names of pizzas that come in both a 10 inch and a 12 inch size.

πpname(σsize=10(P)) ∩ πpname(σsize=12(P))

4. Find the pairs of diﬀerent codes of pizzas with the same name and the same size (is there any?).

πP1.code,P2.code(σP1.code ≠ P2.code ∧ P1.name = P2.name ∧ P1.size = P2.size (ρ(P1,P) × ρ(P2,P)))

5. Find the names and phone numbers of the stores in “College Park” or “Greenbelt” that sell a 10 inch pizza named “pepperoni” for less than $8.

πS.sname,phone(

σsize=10 ∧ pname=‘pepperoni‘∧price<8∧(area=‘CollegePark‘ ∨ area=‘Greenbelt‘)(P ⋈n S ⋈n L))

please note that the attribute names are changed, as given at the beginning of this answer sheet. Otherwise, we cannot use nature join here.

6. Find the codes of the most expensive pizzas assume the scheme of the database is reduced to a relation P → Pizza(code, price) to simplify.

πP1.code,P2.code(σP1.price≥P2.price(ρ(P1,P) × ρ(P2,P)))/πcode(ρ(P1,P))

The intuition is: (i) Find all pairs (code1, code2) of pizza codes where the price of code1 is more or equal to the price of code2. (ii) For a speciﬁc c ∈ code1, if c is paired with all possible codes, it means that its price is more or equal to all prices. Therefore c is the most expensive pizza. Note that many such pizzas may exist (if all have the same high price).

Recall that, for two tables A(x, y), B(y), division is deﬁned as:

A/B = πx(A) − πx((πx(A) × B) − A)

7. Find the names of the stores that sell all the pizzas.

πsname,code(L)/πcode(P)=

πsname(L) − πsname((πsname(L) × πcode(P )) − πsname,code(L))

Observe. This is division. In this case:

A ≡ πsname,code(L)

B ≡ πcode(P )