Randyll Bearer [RLB97@Pitt.edu](mailto:RLB97@Pitt.edu) CS1555 HW 3

Part 1:

A. The two expressions are equivalent. Since SELL.drug is a foreign key from the DRUG table, a join where SELL.drug = DRUG.name will not include or remove any entries that were not already in the SELL table. Therefore, the projections will display the same tuples.

B. The two expressions are NOT equivalent. In the first expression where the cross-product is concerned, the projection of “drug,price” from SELL is occurring BEFORE the cross-product. This means that the resultant table will have no “store” attribute as it was removed from SELL by the projection, and thus the expression “store=name” is invalid. In the second expression though, the projection does not occur until after the join and so no error will occur.

C. The two expressions are NOT equivalent. In the second expression the natural join will be between the DOCTOR table reduced to just the “specialty” attribute and the PRESCRIPTION table reduced to just the “drug” attribute. Since these tables do not share any attributes (i.e. drug != specialty) the natural join will simply become a cartesian product (i.e. the result will contain duplicates). In the first expression DOCTOR and PRESCRIPTION contain the common attributes of “Doctor\_Lic\_no” and “Doctor\_Lic\_State” so duplicates would be removed.

Part 2:

A. If we are assuming that all drugs already fall under the three price ranges, then the “σprice≥10” statement is pointless as it won’t eliminate any tuples. Therefore, the expression can be more efficiently written as Πformula(SELL **EQUIJOIN**Drug=Name DRUG);

B. The inclusion of the PHARMACY table in these sequential expressions is unnecessary as SELL.store (a foreign key) could be used to retrieve the pharmacy name as well, thus the first expression can be removed and the second to last expression modified.

PharmacyDrugs ← SELL **EQUIJOIN**SELL.Drug=DRUG.Name DRUG;

PharmacyPharmaceutical ← PharmacyDrugs **EQUIJOIN**PCM Registration=Manufacturer Registration PHARMACEUTICAL\_CO;

OaklandSells ← σPharmacyDrugs.Store=’Oaklandpharmacy’ ∧ PHARMACEUTICAL\_CO.Name=’Johnson’(PharmacyPharmaceutical);

RSLT ← ΠDRUG.Name(OaklandSells);

Part 3:

A. Assuming that PRESCRIPTION.Patient is a Foreign Key for PATIENT.SSN and that DOCTOR.Doctor\_Lic\_No is a Primary Key and that a patient can only get one prescription from the same doctor.

ValidPrescriptions ← σPatient = ‘123547689’(PRESCRIPTION);

DoctorsPrescriptions ← ValidPrescriptions \* DOCTOR;

RSLT ← ΠFirstName, LastName DoctorsPrescriptions;

B.

PatientTotals ← Patient **FUNCTION**count Drug (PRESCRIPTION);

ValidPatients ← σ count\_Drug >= 15 (PatientTotals);

PatientNames ← PATIENT **EQUIJOIN**PATIENT.SSN = ValidPatients.Patient (ValidPatients);

RSLT ← ΠFirstName, LastName (PatientNames);

C.

ValidDoctors ← DOCTOR **EQUIJOIN**DOCTOR.Doctor\_Lic\_No = PRESCRIPTION.Doctor\_Lic\_No PRESCRIPTION;

ValidDoctorsNames ← ΠLastName(AllDoctors);

AllDoctors ← ΠLastName(DOCTOR);

RSLT ← AllDoctors - ValidDoctorsNames;

D.

???

E.

TomPatient ← σFirstName = ‘Tom’(PATIENT);

TomSSN ← ΠSSN (TomPatient);

TomsPrescriptions ← TomSSN **EQUIJOIN**TomSSN.SSN = PRESCRIPTION.Patient PRESCRIPTION;

ValidPrescriptions ← PRESCRIPTION **DIVIDED\_BY** TomsPrescriptions;

ValidPatients ← PATIENT **EQUIJOIN**PATIENT.SSN = ValidPrescriptions.SSN PRESCRIPTION;

RSLT ← ΠFirstName, LastName (ValidPatients);

F.

CompanyContracts ← PHARMACEUTICAL\_CO **EQUIJOIN**PCM\_Registration = Manufacturer\_Registration CONTRACT;

RSLT ← ΠCompanyContracts.

Part 4:

Relation R

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **D** |
| Z | Q | A | 1 |
| X | W | B | 2 |
| C | E | C | 3 |
| V | R | D | 4 |
| B | T | E | 5 |
| N | Y | F | 6 |
| M | U | G | 7 |
| < | I | H | 8 |

Relation S

|  |  |  |
| --- | --- | --- |
| **D** | **E** | **F** |
| 1 | A | Y |
| 2 | S | U |
| 3 | D | I |
| 9 | F | O |
| 10 | G | P |

So R\*S will have 3 tuples, and R **RightOuterJoin** S will have have 5 tuples.

R **FullOuterJoin**R.D = S.D S

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **R.A** | **R.B** | **R.C** | **R.D** | **S.D** | **S.E** | **S.F** |
| Z | Q | A | 1 | 1 | A | Y |
| X | W | B | 2 | 2 | S | U |
| C | E | C | 3 | 3 | D | I |
| V | R | D | 4 | NULL | NULL | NULL |
| B | T | E | 5 | NULL | NULL | NULL |
| N | Y | F | 6 | NULL | NULL | NULL |
| M | U | G | 7 | NULL | NULL | NULL |
| < | I | H | 8 | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | 9 | F | O |
| NULL | NULL | NULL | NULL | 10 | G | P |