RE_F14_177_Asst4

A [relational algebra] - Kevin Zheng

A.1 (i)

$\Pi_{\text{Patient.AdmitNo, PName}}(\sigma_{\text{Illness.Disease=COVID}}(\text{Patient} \bowtie_{\text{AdmitNo=AdmitNo}} \text{Illness})$

A.1 (ii)

$\Pi_{\text{Doctor.DStaffNo, Dname}}(\sigma_{\text{Insurance=NIB}}(\text{Patient} \bowtie_{\text{AdmitNo=AdmitNo}}$
$\text{Treatment} \bowtie_{\text{DStaffNo=DStaffNo}} \text{Doctor}))$

A.1 (iii)

$\Pi_{\text{NStaffNo, DStaffNo}}(\text{Treatment} \bowtie_{\text{AdmitNo=AdmitNo}} \text{Locate} \bowtie_{\text{Ward=Ward}} \text{Assign})$

A.1 (iv)

$\Pi_{\text{Patient.AdmitNo, DStaffNo}}(\text{Patient} \bowtie_{\text{AdmitNo=AdmitNo}} \text{Illness} \bowtie_{\text{Disease=Disease}} \text{Expertise})$

A.1 (v)

$\Pi_{\text{Nurse.StaffNo, NName, Doctor.DStaffNo, Doctor.DStaffNo}}(\sigma_{\text{Disease=Anthrax}}(\text{Doctor} \bowtie_{\text{DStaffNo=DStaffNo}}$
$\text{Treatment} \bowtie_{\text{AdmitNo=AdmitNo}} \text{Locate} \bowtie_{\text{Ward=Ward}} \text{Assign} \bowtie_{\text{NStaffNo=NStaffNo}} \text{Nurse})$

B [relational design theory] - Kuai Yu

B.1

B.1 (i)

(HospitalNumber)+

1. result = HospitalNumber
2. result = HospitalNumber, HospitalName (HospitalNumber → HospitalName)

B.1 (ii)

(HospitalName)+

1. result = HospitalName
2. result = HospitalName, HospitalNumber (HospitalName → HospitalNumber)

B.1 (iii)

(DrugName)+

1. result = DrugName
2. result = DrugName, DiseaseCode, ManufacturerName (DrugName → DiseaseCode, ManufacturerName)

B.1 (iv)

(DrugName, SizeOfDose)+

1. result = DrugName, SizeOfDose
2. result = DrugName, SizeofDose, Cost, DiseaseCode, ManufacturerName (DrugName, SizeofDose → Cost) and (DrugName → DiseaseCode, ManufacturerName)

3. result = DrugName, SizeofDose, Cost, DiseaseCode, ManufacturerName, ManufacturerAddress (ManufacturerName → ManufacturerAddress)

B.1 (v)

(ManufacturerName)+

1. result = ManufacturerName
2. result = ManufacturerName, ManufacturerAddress (ManufacturerName → ManufacturerAddress)

B2

No, relation DrugUsage is not in BCNF, because:

- HospitalNumber → HospitalName
- HospitalName → HospitalNumber
- DrugName → DiseaseCode, ManufacturerName
- DrugName, SizeofDose → Cost
- ManufacturerName → ManufacturerAddress

For each function dependency above, none of them:

- is trivial (the right side is included in the left side)
- the left side is a super key of the relation (determines all attributes)

We say a relation is in BCNF when any of those two conditions above hold for every functional dependency in the relation. Thus the DrugUsage is not in BCNF because all its functional dependencies violate the BCNF.

B3

Decompose DrugUsage into:

- Hospital (HospitalNumber, HospitalName) with
  - HospitalNumber → HospitalName
  - HospitalName → HospitalNumber

  Relation Hospital is in BCNF because all function dependencies left sides are the super keys (determines all attributes) of the relation Hospital
- Manufacture (ManufacturerName, ManufacturerAddress) with
  - ManufacturerName → ManufacturerAddress

  Relation Manufacture is in BCNF because all function dependencies left sides are the super keys (determines all attributes) of the relation Manufacture.
- Drug (DrugName, DiseaseCode, ManufacturerName) with
  - DrugName → DiseaseCode, ManufacturerName

  Relation Drug is in BCNF because all function dependencies left sides are the super keys (determines all attributes) of the relation Drug
- DrugCost (DrugName, SizeofDose, Cost)
  - DrugName, SizeofDose → Cost

  Relation DrugCost is in BCNF because all function dependencies left sides are the super keys (determines all attributes) of the relation DrugCost.

C [index choices] - Kelvin Chen

i)

```
CREATE INDEX ON T(b)
```

This index will speed up searches on T.b, but doesn’t cover the SELECT * because other attributes are not in the index will have to be accessed with the database

ii)

No need to create a new index because a is the primary key and a clustered index is implicit for primary keys. However, the index doesn’t cover this query as you need T.c.

iii)

```
CREATE INDEX ON T(c, a)
```

This index both covers and speeds up the query as there wasn’t an index on c before. A is also included in the index so that the index can answer this particular query but can be removed and will still speed up this query.

iv)

```
CREATE INDEX ON T(b, d)
```

The default clustered index doesn’t cover this query as the WHERE clause doesn’t involve the primary key the index is based on. The above index is multivalued and will speed up the query on b and d. However, it doesn’t cover the query as T.a is needed.

v)

```
CREATE INDEX ON T(d, b)
```

Since the original structure doesn’t contain an index on d and b (original only contains the clustered index on a), this index will speed it up. This index doesn’t cover the query as both a and c are not covered but are in the SELECT clause.