Databases
Answers to selected exercises
Version: Tuesday 25th February, 2025, 14:30

1

It makes sense to identify books in general with the `isbn`. Each copy of a book will be identified with the attribute `bno`, representing a unique number without meaning and corresponding to the bar code in the book.

```
CREATE TABLE BOOK (
  isbn INTEGER,
  title CHAR(50),
  author CHAR(50),
  publisher CHAR(50),
  bno INTEGER,
  PRIMARY KEY (bno)
)
```

Unfortunately, this leads to data redundancy. If we have n copies of a book, the descriptive book data is repeated n times. Therefore, we should split.

```
CREATE TABLE BOOK (
  isbn INTEGER,
  title CHAR(50),
  author CHAR(50),
  publisher CHAR(50),
  PRIMARY KEY (isbn)
)
```

```
CREATE TABLE BOOK-COPY (
  isbn INTEGER,
  bno INTEGER,
  PRIMARY KEY (bno),
  FOREIGN KEY isbn REFERENCES BOOK(isbn)
)
```

Note that with this modification, the schema of `LOAN` may remain unchanged, apart from the foreign key `bno`.

2

Recall:

```
READER(rno, name, address,city)
```

For the additional table Reservation, we define the primary and foreign keys in the Book table from the previous exercise. Again by using SQL/DDL we create the following statement:

```
CREATE TABLE RESERVATION (
  resno INTEGER,
  rno INTEGER,
  isbn INTEGER,
  PRIMARY KEY (resno),
  FOREIGN KEY (rno) REFERENCES READER(rno),
  FOREIGN KEY (isbn) REFERENCES BOOK(isbn),
)
```

3

Q1: $\pi_{dno,name}(Driver \bowtie \sigma_{date='14-2-2017'}(Schedule))$

Q2:
$\pi_{dno,name}(Driver \bowtie \sigma_{date='14-2-2017'}(Schedule) \bowtie \sigma_{cap>60}(Bus))$
or
$\pi_{dno,name}(\sigma_{(cap>60 \wedge date='14-2-2017')}(Driver \bowtie Schedule \bowtie Bus))$

While this alternative approach leads to the same result, it requires more space and time resources (without optimization).

Q3:
$\pi_{dno,dname}((\pi_{dno}(Driver) -$
$\pi_{dno}(Schedule \bowtie (\sigma_{type='A'}(Bus)))) \bowtie Driver)$

Q4:
$\pi_{dno,dname}((\pi_{dno}(Driver) -$
$\pi_{dno}(Schedule \bowtie (\sigma_{type \neq 'A'}(Bus)))) \bowtie Driver)$

Q5: $\pi_{dno,name,rtid}(Driver \bowtie Schedule) \div \pi_{rtid}(\sigma_{nr-of-stops>10}(Route))$

Note that we make use of the convention that unary operators have a higher precedence than binary operators.

Q6:
$S1 := \sigma_{date='14-2-2017'}(Schedule);$
$S2 := \sigma_{date='14-2-2017'}(Schedule);$

$Result := \pi_{S1.dno}(S1 \bowtie_\theta S2)$
with $\theta : S1.dno = S2.dno \wedge S1.rtid \neq S2.rtid$

4

$R \cap S \equiv R - (R - S))$

$R[X, Y] \div S[Y] \equiv \pi_X(R) - \pi_X((\pi_X(R) \times S) - R)$

5

(i) No. The NOT can be expressed using a minus. The AND and OR can be expressed by intersection and union.

(ii) The algebra also serves as an intermediate language for query processing. For this purpose, the operators should reflect the physical operations to some extent. A selection is calculated basically by a single table scan, also in the case of more complicated selection predicates with AND, OR and NOT. We do not need minus, intersection or union to calculate a selection of this kind.

The same argument explains why we distinguish the (equi) join from a cartesian product followed by a selection. It can be calculated much more efficiently than a cartesion product.

(iii) Again, if we did, the evaluation of a selection would require in general more complicated physical operations than a single scan, for instance a join with other tables.

6

i

When an Entity-Relation diagram is not connected, this means that there are at least 2 entities in the diagram which are not connected, directly or indirectly. From this we can say that the ER-diagram models at least 2 independent parts which shouldn't be modeled in a single database.
For example, if we would have an ER-diagram that includes bank transactions and bus schedules, we will find that there is no relation, and therefore should be modeled in 2 separate databases.

ii

The ER-diagram forces relationships between (S, T), (S, C), and (C, T) in its relational representation. The information that student s follows course c, and teacher t teaches course c is already incorporated in this structural diagram. The additional link is therefore redundant when this link is interpreted as such, and must be avoided.

iii

When interpreting the relationship (S, T) as a student-tutor relationship, we are modeling additional information, and are not introducing redundancy. Therefore it makes sense to model the relationships as such.

iv

An ISA-hierarchy is based on generalisation and specialisation and is generally directed. In an ISA-hierarchy, the lower entity is more specified and/or has less instances. Supposing a cyclic relation leads to the contradictory observation that an entity is more specified and/or has less instances than itself.

8

ISA hierarchies: suppose we have a top entity set $R(K, A)$ and entity subsets $S(K, B)$ and $T(K, C)$. Be aware of the fact that all three tables share the key K.
Options:

- One table $R(K, A, B, C, sub)$. Attributes B and C are null if irrelevant. Attribute `sub` denotes to which subset the tuple belongs. Note that this solution is only applicable if the ISA is disjoint. It also has the disadvantage of introducing null values that are not necessary and occupying possibly much space for these null-vlues.
- Two tables: $S(K, A, B)$ and $T(K, A, C)$. This solution is ok in the case of a non-overlapping and covering ISA-hierarchy. In other cases it introduces null-values.
- Three tables: $R(K, A)$, $S(K, B)$ and $T(K, C)$. This is the most general solution, well suited for all other cases. We need two foreign key constraints.

Note that the disjointness property is not expressible using keys and foreign keys.

9

In a scheme $R(K, A_1, ..., A_n)$ the key property of K is expressed by the FD $K \rightarrow A_1, ..., A_n$. So the key property is a special case of a FD.

10

$A^+ = ABCDHGE$
$(BC)^+ = BCDHGAE$
$BC \rightarrow F$ is not valid, because $F \notin (BC)^+$.
$BC \rightarrow G$ is valid, because $G \in (BC)^+$.

11

In the previous exercise, A turned out to have a large closure. By adding F and K, we create a key AFK.
We might extend BC the same way. However, when we add F, B turns out to be superfluous! So CFK suffices.
Note that F en K will always be part of a candidate key (why?).

12

- $X \rightarrow Y \Rightarrow Y \rightarrow X$ (symmetry)
  Counterexample:

X	Y
a	c
b	c

- $X \rightarrow Y, U \rightarrow V \Rightarrow XU \rightarrow YV$ (augmentation 2)
  Valid:

  (1) $X \rightarrow Y$ (given)
  (2) $XU \rightarrow YU$ (augmentation of (1) with U)
  (3) $U \rightarrow V$ (given)
  (4) $YU \rightarrow YV$ (augmentation of (3) with Y)
  (5) $XU \rightarrow YV$ (transitivity with (2) and (4))

- $XY \rightarrow Z \Rightarrow X \rightarrow Z$ (left decomposition)
  Counterexample:

X	Y	Z
a	b	c
a	d	e

13

We have a relation r on schema $R(XYZ)$ with the FD $X \rightarrow Y$.
We have to prove: $r = \pi_{XY}(r) \bowtie \pi_{XZ}(r)$.
In order to prove that two sets A and B are equal, we show that A is a subset of B and vice versa. In this case : $r \subseteq \pi_{XY}(r) \bowtie \pi_{XZ}(r)$ en $\pi_{XY}(r) \bowtie \pi_{XZ}(r) \subseteq r$.

So we need a two step proof for a some tuple t:

1. if $t \in r$ then $t \in \pi_{XY}(r) \bowtie \pi_{XZ}(r)$, and
2. if $t \in \pi_{XY}(r) \bowtie \pi_{XZ}(r)$ then $t \in r$.

Proof:

1. Suppose we have a $t = (x, y, z) \in r$. By projection, we have a $t_1 = (x, y)$ in $\pi_{XY}(r)$ and a $t_2 = (x, z)$ in $\pi_{XZ}(r)$. Because t_1 and t_2 match on X, we will see a tuple (x, y, z) in $\pi_{XY}(r) \bowtie \pi_{XZ}(r)$.

2. Suppose we have a $t = (x, y, z) \in \pi_{XY}(r) \bowtie \pi_{XZ}(r)$. This has originated from a $t_1 = (x, y)$ in $\pi_{XY}(r)$ and a $t_2 = (x, z)$ in $\pi_{XZ}(r)$. t_1 and t_2 must have been created by projection from the following tuples in r: $t_1' = (x, y, z')$ and $t_2' = (x, y', z)$, for some z' and y'. Because of FD $X \to Y$ and $t_1[X] = t_2[X]$, we know that $y = y'$. So $t_2' = (x, y, z) = t$, in other words $t \in r$. (Replace y' by y in t_2).

14

Only for the brave!

For the proof, we need the soundness of the Armstrong-axioms.
Soundness:
A short inductive argument. Let X^i be the set that we calculated in round i.
Note that $X^0 = X$, so $X \to X^0$ holds because of reflexivity.
Induction hypothesis: $X \to X^i$.
Note that the calculation of X^{i+1} is based on reflexivity and transitivity, so we can conclude immediately (after applying the union-rule) that $X \to X^{i+1}$ holds.
Completeness:
Suppose we have a set of fds F. We call the result of the closure-algorithm on left side X: X^*. Note that $X^* \subset X^+$, due to the soundness.
Hypothesis:
Suppose we have an attribute A in X^+ (so: $X \to A \in F^+$) such that $A \notin X^*$. We construct a relation r with two tuples t_0 en t_1 as follows, where $A \in Z$.

```
      X*      Z = remaining attrs
=========   =======
0 0 ... 0   0 ... 0
0 0 ... 0   1 ... 1
```

Claim 1: each fd $V \to W \in F$ holds on r.
Proof: suppose r violates $V \to W$. Then we see that

- $V \subset X^*$, otherwise the left sides would be different, and
- $W \cap Z \neq \emptyset$, otherwise the left sides would be the same.

We now have $V \subset X^*$. The closure-algoritme should have taken $V \to W$ into account and should have added W to X^*. This contradicts $W \cap Z \neq \emptyset$. Claim 1 has been proven.

Observation 2: each fd $V \to W$ in F^+ holds on r. This follows immediately from claim 1 and the soundness of the Armstrong-axioms.

Conclusion: we see that $X \to A$ also holds on r. Because $X \subset X^*$ and $A \in Z$, we see a contradiction for the hypothesis that there exists such an A.

15

When you have a scheme (AB), the only possible non-trivial fd's are $A \to B$ and $B \to A$ (or fd's derived from them). In both cases, the left side of the fd is a superkey.

16

Every decomposition that the algorithm chooses obeys the conditions of the decomposition theorem (slide FD 17).