Database Management (COIY028H6) – 2019

Model Answers

1. (a) A data model should comprise a structural part, an integrity part and a manipulative part. The relational model provides standard definitions for all three parts, whereas the entity-relationship model has no standardised manipulative part.

(5 marks)

(b) i. The cardinality of the PaysFor relationship type is many-to-one.

(1 mark)

ii. Premium is a weak entity type. This means that it doesn't have sufficient attributes of its own to act as a primary key.

(3 marks)

iii. PaysFor is an identifying relationship type. This means that it connects a weak entity type to its identifying or owner entity type.

(3 marks)

iv. Premium has mandatory participation in the PayFor relationship type, while Premium has optional participation.

(2 marks)

v. The relation schema would be Premium(PolicyNumber, DueDate) with primary key (PolicyNumber, DueDate). There would be a foreign key constraint for PolicyNumber in Premium, referencing the primary key of Policy.

(5 marks)

(c) The OUTER JOIN operator in SQL allows rows from one table that would be excluded in a normal join because they don't match rows in the other table to be included in the join, padded with null values. The LEFT OUTER JOIN includes rows from the left table that do not join, the RIGHT OUTER JOIN includes those from the right table, and the FULL OUTER JOIN includes rows from both that do not join.

(6 marks)

2. (a) i. The database system will ensure that every non-null B-value in S also appears as an A-value in R.

(3 marks)

ii. If the A-value is not referenced by a B-value in S.

(2 marks)

iii. A user can always delete a row from S.

(2 marks)

iv. Since each row in S can join with at most one row in R (B may be null), there will be at most m rows in the join.

(3 marks)

(b) A set of transactions is serialisable if they behave as if they had been run serially, that is, the effect of running them concurrently is the same as that when running them serially.

An example of non-serialisable execution is two transactions running concurrently and booking the same seat on a flight. The sequence in time might be as follows:

i. Transaction 1 finds seat 22A empty.
ii. Transaction 2 finds seat 22A empty.
iii. Transaction 1 books seat 22A.
iv. Transaction 2 books seat 22A.

(6 marks)

(c) First normal form requires that attributes have atomic values. Examples of features violating 1NF include the built-in array and multiset types, or the use of user-defined types to create attribute types that comprise many components.

(4 marks)

(d) XML databases and document stores both address the criticism. Neither require a fixed schema to be defined before storing data, and both allow for nested and varying structures to be stored.

(5 marks)

3. (a) i. (A) UserID, Book $\rightarrow$ Rating, and (B) Book $\rightarrow$ Author.

(2 marks)

ii. The author of each book is repeated for each rating.

(2 marks)

iii. A key is UserID, Book, because that determines Rating, and Book determines Author, so we get all attributes. No subset is a key because users can rate multiple books and each book can be rated by multiple users.

(3 marks)

iv. The FD Book $\rightarrow$ Author violates BCNF because its left side is not a superkey. Hence we decompose into (Book, Author) and (UserID, Book, Rating). No FDs violate BCNF in either of these schemas.

(5 marks)

v. Such a decomposition is not lossless: if we perform the natural join of the decomposed relations, we get back two tuples that were not in the original relation. It is also not dependency-preserving, since Book $\rightarrow$ Author is not preserved.

(4 marks)

(b) We start by adding HS. From the 5th FD, we add R. Now from the 2nd FD, we can add C, and from the fourth we can add G. Finally from the first FD we can add T.

(5 marks)

(c) Null values are used to represent missing values or inapplicable attributes. They are ignored by `max` and collected into a single group by `group by`.

(4 marks)

4. (a) i. {Code,Textbook} and {Name,Textbook} are keys. Using the closure algorithm, once we have Code in the closure, we can add Name and Level. Similarly if we start with Name.

(4 marks)

ii. The first step removes any redundant FDs, but there aren't any. The second forms a relation schema from the attributes comprising each FD, so we get: {Code, Name}, {Code, Level} and {Name, Code}. The third step removes any schema that is a subset of another, so we can remove {Name, Code}. The fourth step adds a schema comprising a key if one is not contained in the schemas so far. So we add {Code, Textbook} or {Name, Textbook}. The final 3 relation schemas are therefore {Code, Name}, {Code, Level} and {Code, Textbook}, or {Code, Name}, {Code, Level} and {Name, Textbook}.

(6 marks)

(b) The minimum number is 3. Since the node is full, it needs to be split into two nodes, with $(17, 19, 22)$ in one node, $(25, 26, 28)$ in the other node, and the median value 23 inserted into the parent node. If the parent is full, then splitting needs to be repeated.

(8 marks)

(c) Placeholders are used in SQL to allow for user input at runtime. There are named and anonymous placeholders. The query "select price from Sells where pub=? and beer=?" uses anonymous placeholders, while "insert into Pubs (name, location) values (:name, :location)" uses named placeholders.

(7 marks)

5. (a) i. They are aliases or tuple variables. They are needed because `s_id` and `m_code` each appear in two relation schemas and so an unqualified reference would be ambiguous. (Of course, one could prefix the attribute names with the relation names, so stating that aliases allowed for shorter or more readable queries was also accepted as an answer.)

(3 marks)

ii.
```
from Student S join Result R on S.s_id=R.s_id
     join Module M on M.m_code=R.m_code
```

(3 marks)

iii. More than one relation name is mentioned in the `from` clause.

(2 marks)

iv. The three values given are all that is needed to insert a new row in the Result relation. If `s_id` and `m_code` already exist in the Student and Module relations, we are done. If not, only a single row needs to be inserted in each, since `s_id` and `m_code` are primary keys for the Student and Module relations respectively. This will be possible if both `s_name` and `m_name` can be set to null.

(7 marks)

(b) i. A natural join joins two relations based on all attributes with common names.

ii. Data independence is the property that changes at a lower level of a database system do not affect the higher levels.

iii. Transaction isolation level can be set to specify to what extent a transaction is isolated from others, e.g., whether or not to allow uncommitted read operations.

iv. Three-valued logic is employed to handle evaluating conditions involving null values, when a third value of "unknown" is needed.

v. A correlated subquery is one in which a condition in an inner query refers to an attribute from an outer query.

(10 marks)

