1.9 Databases and SQL: Test Pack

AQA 7517 · Calculators not allowed · about 20 minutes

Total Marks

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 Elm Grove Music Academy's booking system is used by several members of staff at once.

(a) Explain what is meant by record locking in a multi-user database system. **(2)**

(b) Two members of staff both try to change the Time of the same booking record (S04, I04) at the same moment. Explain, using pessimistic record locking, how the DBMS would prevent this causing a lost update. **(3)**

(Total for Question 1 is 5 marks)

© Revision Library | revisionlibrary.co.uk

2 A separate part of Elm Grove Music Academy's system processes online payments. When a parent pays a £50 deposit, the system must (1) subtract £50 from the parent's stored account balance and (2) add £50 to the academy's income record, using two SQL UPDATE statements grouped together.

(a) Define the term 'transaction' in the context of a database system. **(2)**

(b) State what is meant by COMMIT and ROLLBACK. **(2)**

(c) Discuss how each of the four ACID properties (Atomicity, Consistency, Isolation, Durability) contributes to the reliable processing of the deposit payment described above, particularly if the system were to fail partway through. **(6)**

(Total for Question 2 is 10 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 1.9 Databases and SQL: Test Pack

Question 1

(a) when a user/process is updating a record, the DBMS locks that record so no other user/process can access (or at least cannot update) it at the same time (oe) **B1**

(a) the lock is released once the update/transaction is complete, allowing other users to then access the record (oe) **B1**

(a) Answer: Record locking is when the DBMS prevents more than one user or process from updating the same record at the same time. When a user begins updating a record, the DBMS locks it; the lock is released once their transaction is complete, after which other users can access the record.

(b) the DBMS allows the first member of staff's request to lock the (S04, I04) record as soon as they begin editing it (oe) **B1**

(b) the second member of staff's request is blocked/queued (or given an error) and cannot open the same record for editing until the first lock is released (oe) **B1**

(b) once the first update is committed and the lock released, the second staff member's request can proceed, now working from the up to date record, so neither change is silently overwritten/lost (oe) **B1**

(b) Answer: The DBMS lets whichever staff member accesses the (S04, I04) record first place a lock on it. The second staff member's request to edit the same record is blocked (or shown an error) until that lock is released. Once the first update is committed and the lock released, the second request can then proceed, working from the now up to date record, so neither person's change silently overwrites the other's.

Question 2

(a) a transaction is a logical unit of work made up of one or more database operations, e.g. SQL statements (oe) **B1**

(a) it is treated as a single, indivisible action: either every operation within it completes successfully, or none of them are applied (oe) **B1**

(a) Answer: A transaction is a logical unit of work made up of one or more database operations (such as SQL statements) that is treated as a single, indivisible action: either every operation in it completes successfully, or none of them are applied to the database.

(b) COMMIT: permanently saves/applies all the changes made during the transaction to the database (oe) **B1**

(b) ROLLBACK: cancels/undoes all the changes made during the transaction, returning the database to the state it was in before the transaction began (oe) **B1**

(b) Answer: COMMIT permanently saves all the changes made during a transaction to the database. ROLLBACK cancels/undoes all the changes made during a transaction, returning the database to the state it was in before the transaction started.

(c) **L3** Clear, accurate explanation of all four ACID properties, each explicitly applied to the deposit scenario (e.g. what happens if the system fails between the two UPDATE statements). **5-6**

(c) **L2** Sound explanation of two or three ACID properties, with some application to the deposit scenario. **3-4**

(c) **L1** Basic, undeveloped reference to one or two ACID properties, with little or no link to the deposit scenario. **1-2**

(c) **L0** No creditable content. **0**

(c) Indicative content:

- Atomicity: the two UPDATE statements are treated as one all-or-nothing unit; if the system fails after the balance is subtracted but before the income record is updated, the whole transaction is rolled back so the deposit is not lost or applied only halfway.
- Consistency: the transaction must leave the database in a valid state, e.g. the combined total of (parent balance + academy income) is preserved before and after; any rule violation (such as a resulting negative balance) prevents the transaction from completing.
- Isolation: if another transaction is reading or updating the same parent account balance at the same time, each transaction's intermediate steps are hidden from the other, so a concurrent read cannot see a partly-completed deposit.
- Durability: once the transaction has been committed, the £50 deposit is permanently recorded (e.g. written to non-volatile storage) and will survive a subsequent power failure or system crash.

(c) Answer: Extended, levels-marked response: award marks holistically against the levels of response grid, using the indicative content as a guide to creditable points; there is no single correct wording.

© Revision Library | revisionlibrary.co.uk