



12 Pages

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right, fold over and seal down (using the label provided) to hide your name.
3. The answers must be legibly written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. **Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.**

Complete, fold and seal
(see note 2 below)

SURNAME: Morris

FORENAME(S): William James

1	Student Number	WM185927
2	Title of Examination Paper	Database Management Systems
3	Module Code	ACCA5014
4	Date and Time of Examination	9th May 2019 9:05AM
5	Examination Venue	Grang - G42
6	Seat Number	
7	Number of Answer Books handed in	1

List the numbers of the questions you have attempted	Office Use
1	22 $\frac{1}{2}$
2	20 $\frac{1}{2}$
4	24
Total	67

(Gweler yr ochr arall am y fersiwn Cymraeg)

Q1.

i) CREATE TABLE TutorBooking (
 BookingRef VARCHAR(6) NOT NULL,
 BookingDate DATETIME NOT NULL,
 TutorNo VARCHAR(6) NOT NULL,
 StudentNo VARCHAR(6) NOT NULL,
 SubjectId VARCHAR(6) NOT NULL,
 SubjectLevel VARCHAR(15) ~~NOT~~ NULL, -- Subject may not have level
 HourlyRate DECIMAL(18, 2) NOT NULL,
 CONSTRAINT CHK_BookingDate CHECK BookingDate > GETDATE()
)

ii) ALTER TABLE TutorBooking
 ADD CONSTRAINT PK_TutorBooking PRIMARY KEY BookingRef.

iii) ALTER TABLE TutorBooking
 ADD CONSTRAINT FK_TutorBooking-Tutor FOREIGN KEY TutorNo
 REFERENCES Tutor (TutorNo),
 ADD CONSTRAINT FK_TutorBooking-Student FOREIGN KEY StudentNo
 REFERENCES Student (StudentNo),
 ADD CONSTRAINT FK_TutorBooking-Subject FOREIGN KEY
 StudentNo REFERENCES Student (StudentNo)

iv) CREATE TABLE TutorSubject (
 TutorNo VARCHAR(6) NOT NULL,
 SubjectId VARCHAR(6) NOT NULL,
 KsLevel ~~VARCHAR~~ BIT NOT NULL, -- BIT used to
 ALevel ~~VARCHAR~~ BIT NOT NULL, -- denote whether
 HsLevel ~~VARCHAR~~ BIT NOT NULL, -- a tutor teaches this level.
 MinHourlyRate DECIMAL(18, 2) NOT NULL,
 MaxHourlyRate DECIMAL(18, 2) NOT NULL,
 CONSTRAINT PK_TutorSubject PRIMARY KEY (TutorNo, SubjectId),
 CONSTRAINT FK_TutorSubject-Subject ~~Subject~~ FOREIGN KEY
 SubjectId REFERENCES Subject (SubjectId),
 CONSTRAINT CHK_HourlyRate CHECK MaxHourlyRate >
 MinHourlyRate.

b.

i) SELECT
TutorNo, TutorName
CONCAT(FirstName, ' ', Surname) AS ~~TutorName~~
~~EMAIL~~
~~PhoneNo~~
PhoneNo,
DBSExpiryDate
FROM
Tutor
WHERE
DBSExpiryDate BETWEEN '01/06/19' AND '31/06/19'
ORDER BY
DBSExpiryDate.

ii. UPDATE Tutor
SET Status = 'Pending DBS Check'
WHERE DBSExpiryDate < GETDATE()

iii) SELECT
MIN(HourlyRate) AS MinRate,
MAX(HourlyRate) AS MaxRate,
AVG(HourlyRate) AS AvgRate
FROM
TutorBooking
WHERE
BookingDate BETWEEN '01/01/19' AND '31/12/19'

Q2

a) External Schema describe end user interaction with a DB. They are ^{usually} consist of several subschema. Each subschema is concerned with what data a group of users will see. The rest is hidden.

Internal Schema use the physical data model to describe complex, low-level structure of the database. They are concerned with how data will be stored in a block.

Conceptual Schema is the level at which programmers and admins work. They are used to describe the entire DB, what data will be ~~used~~ stored and the relationships between them. Implementation of the data structure is hidden.

i) An Index is a performance enhancing method which can be used to speed up read operations on large tables in a DB.

ii) It is important to analyse transactions before implementing indexes as they can sometimes slow down write and delete operations. This is because data has to be updated or deleted from both the index and the physical table. When considering their implementation it is best to look for large tables with many columns, some of which won't always be read. It is also best to identify columns which won't be updated often.

v)

05-09-2019^{*}

```
INSERT INTO TutorBooking
VALUES ('500', 07-05, '66', '25', '100', NULL, '25')
```

```
INSERT INTO BookedLesson
```

```
VALUES ('500', '30-05-2019 10:00:00', '120', NULL, NULL,
        NULL, NULL)
```

```
VALUES ('500', '31-05-2019 10:00:00', '120', NULL, NULL,
        NULL, NULL)
```

5 ✓

ii

iii)

I would index the surname column so that you could return a list of tutors by name quickly. These values will not be updated often.

```
ALTER TABLE Tutor
```

```
CREATE INDEX IDX_surname (surname)
```

1 1/2 ✓

iv)

I would choose to index the DBS Expiry Date column to allow for quicker lookup of these dates when needed as this would probably be checked often but only updated once a year.

2 ✓

20 1/2 ✓

Q4)

a) The four ACIDity properties are:

Atomicity

Consistency

Isolation

Durability.

Atomicity is the idea that each transaction should be treated as one atomic unit. Transactions should always fully succeed or fail and a transaction should never be left partially completed.

Consistency is the idea that the DB should always remain in a consistent state, both before and after a transaction. Transactions should not affect the integrity & consistency of data.

Isolation states that every transaction should be treated as the only active transaction. The existence of a transaction should never affect other concurrent transactions.

Durability states that the DB should be durable enough to remember recent events in the case of a failure. When a failure does occur, any uncommitted transactions should be completed as soon as the system is back up.

b) Concurrency control plays a role in ensuring serialisation and ACIDity of transactions to prevent errors and inconsistent data by managing concurrent transactions.

c)

concurrent

- i) Lost updates occur when two transactions attempt to update the same data item. If both transactions read at similar times and then each begin to update that value, the first value can be written and committed but may then be overwritten by the write and commit of the second transaction.

An uncommitted dependency ~~is~~ is when one transaction can see the intermediate results of another transaction. If this value is read by another transaction in order to update it then this can cause inconsistency in data as the first transaction has not yet finished.

Inconsistent analysis or a non-repeatable read occurs when a transaction is reading a data-item multiple times but the value of the data-item is updated by another transaction during this time. This results in an inconsistent read of data for the first transaction.

ii)

Time	T_1	T_2	bal _x (£500)	
t1		BEGIN TRANS	500	✓
t2	BEGIN TRANS	Read bal _x	500	✓
t3	Read bal _x	bal _x = bal _x + 100	500	✓
t4	bal _x = bal _x - 50	Write bal _x	600	✓
t5	...	COMMIT	600	✓
t6	Write bal _x		450	✓
t7	COMMIT		450	✓

The lost update problem has occurred here because ~~both transactions~~ while T_1 reads at the same time as T_2 updates bal_x, this new value has not been written yet. This means that T_2 will commit with its new balance at t5 but shortly after, T_1 will commit its new value which did not take into account T_2 's update. Hence the lost update - the value of T_2 is lost and only

T_1 's value is saved.

This problem could be prevented by using Concurrency Control protocols such as locking protocols. These include exclusive, shared, simple or preclaiming locking protocols. Timestamp based protocols could also be used. A locking protocol prevents a transaction from reading or updating a data item until it has the ~~appropriate~~ appropriate lock.

d)

i) T_4 needs to be redone as it ~~is~~ was started after the checkpoint and never committed.

ii) T_3 needs to be undone as it was only partially completed ~~at~~ at the checkpoint. So the uncommitted changes need to be redone.

iii) T_2 was not affected by the crash as it was fully committed before the checkpoint.

T_1 was also not affected by the crash, ~~as~~ ^{as} it was ~~started~~ finished before the checkpoint and was aborted anyway.

Student Number: WM185927

Time	T ₁	T ₂	bal _x (£500)
t1		BEGIN TRANS	500
t2	BEGIN TRANS	Read bal _x	500 ✓
t3	Read bal _x	bal _x = bal _x + 100	500 ✓
t4	bal _x = bal _x - 50	Write bal _x	600 ✓
t5	...	COMMIT	600 ✓
t6	Write bal _x		450 ✓
t7	COMMIT		450 ✓

Table 1: Table for Question 4c.ii.