



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: Howells

FORENAME(S): Adam

1	Student Number	AH180955
2	Title of Examination Paper	Database Management Systems
3	Module Code	ACCA5014
4	Date and Time of Examination	09/05/19 09:05am
5	Examination Venue	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	20
Total	123

Dec. A)

i) a)

i) CREATE TABLE TutorBooking (
 BookingRef IDENTITY(1,1) NOT NULL,
 BookingDate DATE NOT NULL,
 CHECK (BookingDate > getdate()),
 TutorNo int,
 StudentNo int,
 SubjectId int,
 SubjectLevel Varchar(10),
 HourlyRate decimal NOT NULL);

6 ✓

ii) ALTER TABLE TutorBooking (
~~BookingRef IDENTITY(1,1) NOT NULL~~
 PRIMARY KEY (BookingRef)
);

1 ✓

iii) ALTER TABLE TutorBooking (
 FOREIGN KEY (TutorNo) REFERENCES [dbo].Tutor.TutorNo,
 FOREIGN KEY (StudentNo) REFERENCES [dbo].Student.StudentNo,
 FOREIGN KEY (SubjectId) REFERENCES [dbo].Subject.SubjectId
);

3 ✓

iv) CREATE TABLE TutorSubject (
 TutorNo int NOT NULL,
 SubjectId int NOT NULL,
 KSLevel Varchar(4),
 ALevel Varchar(5),
 HLevel Varchar(4),
 MinHourlyRate decimal NOT NULL,
 MaxHourlyRate decimal NOT NULL,
 FOREIGN KEY (TutorNo) REFERENCES [dbo].Tutor.TutorNo,
 FOREIGN KEY (SubjectId) REFERENCES [dbo].Subject.SubjectId,
 PRIMARY KEY (TutorNo, SubjectId),
 CHECK (MinHourlyRate < MaxHourlyRate));

2 ✓

b)

- i) SELECT TutorNo, FirstName, Surname, Email, PhoneNo, DBSEpiryDate
FROM Tutor
~~WHERE (DBSEpiryDate BETWEEN '2019-06-01' AND '2019-06-31')~~
WHERE (DBSEpiryDate BETWEEN '2019-06-01' AND '2019-06-31')
GROUP BY DBSEpiryDate;

2 ✓

ii) UPDATE TABLE Tutor (

ADD Status 'PENDING DBS Check'

WHERE (DBSEpiryDate >= getdate(1)));

1 1/2 ✓

iii) SELECT MIN(HourlyRate), MAX(HourlyRate), AVG(HourlyRate)

FROM TutorBooking

WHERE (BookingDate BETWEEN '2019-01-01' AND '2019-12-31');

2 ✓

20 1/2

Sec. B)

Q2)

- a) The external schema defines the portion of Views that is used by the end user to interact with the database. Data is independent between external and Conceptual Schemas. Conceptual Schema ~~also~~ describes the make-up of the table, also known as community view. This Schema defines the rows (tuples) and columns (attributes). The internal Schema is part of the physical database and defines how tables are structured and it almost always consists of SQL code.

5 ✓

b)

- i) Indexes are a number used to quickly reference a column (attribute) in a table (relation). Primary keys are indexes, but secondary indexes can be applied to any column(s).
- ii) It's important to analyse transactions to discover what can benefit from indexes. Generally it's not a good idea to index tables which get updated frequently as it can cause bottlenecks and performance issues. This is because for each update action, the index must be updated too. Smaller tables also does not benefit from indexes as there isn't much data to gather in the first place. Transaction analysis allows you to view key bottlenecks and see which table and column(s) gets used a lot.

2 ✓

5 ✓

- iii) DBExpireDate can be used as a clustered index in order to sort the rows by earliest ~~expiry~~ date to latest expiry date.

CREATE CLUSTERED INDEX Tutor.DBExpireDate AS ExpireDate;

- iv) The phone number is an appropriate secondary index as it will be retrieved frequently and won't be updated heavily.

CREATE INDEX Tutor.PhoneNo AS PhoneNo;

- v) INSERT INTO TutorBooking
VALUES (500, getDate(), 66, 25, 100, Null, Null);

INSERT INTO BookedLesson

~~VALUES (500, '2019-05-30 10:00:00', 120, Null, Null, Null, Null);~~

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

(500, '2019-05-31 10:00:00', 120, Null, Null, Null, Null);

Q4)

- a) ACID is an acronym for individual properties that transactions should have.

A - acidity; describes the property of transaction either fully succeeding or fully failed, never both. Transactions must be complete.

C - consistency; database must be consistent before and after transactions.
E.g. adding 200 to ~~starting~~ a balance of 300 will result in 500.

I - isolation; Each transaction ~~must~~ must perform operations in a separate database state on execution. Changes are permanent after committing. Isolation methods include locking and timestamps.

D - durability; When changes are made to a database it must be permanent in non-volatile memory and persist even if a system failure occurs.

b) Concurrency control handles how simultaneous execution of transactions are managed. It ensures reliability and consistency as issues can otherwise occur when lacking concurrent control.

c) i) Lost update is an issue where, for example, two transactions updates the same data, whichever transaction is committed last will overwrite the other transaction's changes, losing the update.

Uncommitted dependencies is when a transaction updates a value ~~before~~ ^{whilst} another transaction reads it before the first transaction is committed.

Inconsistent analysis occurs when one transaction incorrectly reads a value, also known as a 'dirty read'. For example, T1 writes to a value which T2 reads the new value concurrently. However, T1 is rolled back due to an error which now means that T2 has read the incorrect value.

ii)

Time	T1	T2	balx (£500)
t1		BEGIN TRANS	500
t2	BEGIN TRANS	Read balx	500
t3	Read balx	balx = balx + 100	500
t4	balx = balx - 50	write balx	600
t5	...	COMMIT	600
t6	Write balx		~~600~~ 450
t7	Commit		450

d)

i) T1 must be redone as it crashed during its execution. T1 must also be ~~redone~~ redone as it was aborted.

ii) T3 will need to be undone as the checkpoint occurs during it.

iii) T2 executed successfully and is saved at checkpoint before failure; it is unaffected.

Student Number: AH180953

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	300 ✓
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.