



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Examination Answer Books

12 Pages

Instructions to Candidates

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: Austin

FORENAME(S): Ashley Darren

1	Student Number	179591
2	Title of Examination Paper	Database Management Systems
3	Module Code	
4	Date and Time of Examination	Tuesday 9th May 9:05
5	Examination Venue	G42
6	Seat Number	N/A
7	Number of Answer Books handed in	1

List the numbers of the questions you have attempted	Office Use
Q1	21 1/2
Q2	20
Q4	21
Total	62 1/2

(Gweler yr ochr arall am y fersiwn Cymraeg)

Q1.

a. CREATE TABLE TutorBooking (

BookingRef int,

BookingDate datetime CHECK (CONVERT (date (GETDATE))),

TutorNo int, datetime

StudentNo int,

SubjectId varchar (5),

SubjectLevel ~~int~~ DEFAULT SubjectLevel = '0', char (2),

HourlyRate decimal (8, 2) / DEFAULT HourlyRate = '0.00';

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 SubjectId REFERENCES Subject (SubjectId)

iv. CREATE TABLE TutorSubject (

TutorNo int,

SubjectId varchar (5),

KSLevel char (25), varchar (255),

ALevel char (25), varchar (255),

HELevel char (25), varchar (255),

MinHourlyRate decimal (8, 2) CHECK MinHourlyRate > 0,

MaxHourlyRate decimal (8, 2) CHECK MaxHourlyRate > 0,

ADD CONSTRAINT PK_TutorSubject PRIMARY KEY (TutorNo, SubjectId)

ADD CONSTRAINT FK_TutorSubject FOREIGN KEY SubjectId

REFERENCES Subject (SubjectId)

);

b.

i. SELECT TutorNo, FirstName, Surname, Email, PhoneNo
FROM Tutor
WHERE DBSExpiryDate BETWEEN '01-05-2019' AND '30-05-2019'
SORT BY DBSExpiryDate

ii. ALTER TABLE Tutor
WHERE DBSExpiryDate < '01-05-2019'
UPDATE Status = 'Pending DBS Check'

iii. SELECT MIN (HourlyRate)
FROM TutorBooking

SELECT MAX (HourlyRate)
FROM TutorBooking

SELECT AVG (HourlyRate)
FROM TutorBooking

1 1/2

2 1/2

Q2.

- a. An external Schema is comprised of mostly views and one end user can use to read the data within one database. As stated this Schema would be mostly relevant to the users of the database.

Internal Schemas reference the internal workings of one database and its code. This is especially relevant to the programmers of the database.

A Conceptual Schema covers one database in a broader scale, referencing relations in one database. This information is especially important to the database administrator and programmers.

- b. Indexes can be used to improve the performance of queries within one database, this is especially important for queries with a large lookup return.

- ii. It's important to analyse all of your transactions on one database when concerning indexing to understand if an index is required. When deciding on an index for a relation, criteria such as the following would need to be considered;

- If the transaction is frequently used to search the same data.
- Dealing with a very large amount of data.

If either of these points apply then consideration for an index can be made. Indexes function by only searching a particular column within a 'SELECT' statement, therefore when working with large amounts of data the transactions will not need to query the entire relation.

- iii. An appropriate clustering index for the tutor table would be the FirstName and Surname columns. This could be used to order the tuples into alphabetic order, making the data in the table easier to look up. Additionally when dealing with a large quantity of records in this relation, the clustered index will improve query time.

```
ALTER TABLE Tutor  
ADD INDEX (FirstName, Surname)
```

We do not want to make this a unique index as there may be duplicate values within these columns.

- iv. Within the Tutor table I feel that it would be most beneficial to apply a secondary index onto the Email column. This could be uniquely indexed as nobody can have the same email as another member.

```
v. ALTER TABLE TutorBooking  
INSERT INTO BookingRef, BookingDate, TutorNo, StudentNo, SubjectId  
( '500', '30-04-2019 10:00:00 AM', '66', '25', '100' )  
( '500', '31-04-2019 10:00:00 AM', '66', '25', '100' );
```

Q4.

a. ACID refers to the Atomicity, Consistency, Isolation and Durability of the database. The details of these properties include;

- **Atomicity** - Transactions must be executed as one atomic unit. It can either be complete or not, there can be no in-between. Checks for this must be made before and after the execution / failure of a transaction.
- **Consistency** - Transactions in the database after being executed and committed should have no effect on the consistency of the data within the database. The data must remain in a consistent state both before and after a transaction.
- **Isolation** - This ensures that within a database of multiple, potentially overlapping, transactions that each one will be individually managed. In no scenario where isolation is in place will multiple transactions occur on the database.
- **Durability** - In the event that the database fails or restarts, durability ensures that any updates potentially committed during this time are not lost. When dealing with large amounts of data, this ensures that it will save in the event of a database failure and commit when it returns to an online state. The durability of the database must be maintained.

b. Concurrency control ensures that the database takes isolation into consideration. By not allowing the overlap of transactions within the database.

C.

- i. Lost updates occur when two transactions are attempting to alter one same data within a relation. Therefore one of these transactions will be working with incorrect data. Additionally, any update made to a particular field from one first transaction will be lost.

Inconsistent analysis covers when a transaction updates the data on a particular relation but a user is currently viewing it. Therefore one user will be looking at 'older' data that has not been updated with the new data from the transaction.

Uncommitted dependency occurs when transactions apply relational rules to a database, with multiple transactions ongoing, these may not be committed.

Time	T1	T2	bal x (£500)
T1		BEGIN TRANS	500
T2	BEGIN TRANS	Read bal x	500
T3	Read bal x	$bal x = bal x + 100$	600
T4	$bal x = bal x - 50$	Write bal x	450 450 600
T5	...	COMMIT	450 450 600
T6	Write bal x		450
T7	COMMIT		450

This last update problem occurred due to a mistimed transaction commit. Due to T2 altering the data first and then committing the value was not updated to what they set. Whereas T1's data is committed due to it being after T2's commit.

d.

- i. Transactions T1 and T2 would need to be undone redone as they were committed before the crash.
- ii. T3 would need to be undone as this was committed after the crash had occurred.
- iii. T4 would be unaffected as it was set to complete after T3 (when the crash happened) but it was not committed.

$$\begin{array}{r}
 21\frac{1}{2} \\
 20 \\
 \hline
 2\frac{1}{2} \\
 62\frac{1}{2}
 \end{array}$$