

**THE BCS PROFESSIONAL EXAMINATION
Diploma**

April 2001

EXAMINERS' REPORT

Database Systems

QUESTION 1

a) What features would you expect to find with each of the following database tools:

- Command line interface;
- Form editor;
- Report writer?

Illustrate your answer with examples from database systems you have used. (9 marks)

b) SQL is not suitable as a language for writing database applications. Why is this? (6 marks)

c) To overcome the limitations of SQL, database application programmers can resort to one of the following three solutions:

- Embed SQL statements in a general-purpose programming language like C or COBOL;
- Use a call-level interface with a general-purpose programming language;
- Use an extension of SQL that provides the features of a general-purpose programming language.

Describe the features provided by each of these solutions. Illustrate your answer with examples. (10 marks)

Answer Pointers

a) Command-line interface => query language such as SQL
Enter a command and the result is displayed

Form => screen display version of a paper form

Form can be used for input and/or output

Form editor can create a default form for each table

Form can be used to enter data, to query the database, and to update data

Form editor can be used to create and display forms corresponding to the results of a join including one-to-many joins

Such forms can be used to update the database but you need to specify the exact semantics of this

Form editor can be used to create any arbitrary form as long as you specify the proper semantics for the form elements

A report writer is used to generate formatted output from a database

The contents of the report usually consists of the data from a table or the results of a query which are formatted according to the report specifications

Default reports can usually be generated

User-defined reports can be created using a report definition language and/or a graphical user interface

The report definition will usually specify how the data should be sorted, how the columns should be broken over page boundaries, how data should be summarised and where these summaries should be placed.

The report writer may also support the generation of graphics such as pie charts and bar charts

Examples illustrating above.

- b) A general-purpose programming language should include facilities for creating variables and procedures, and for writing iterative and/or recursive programs. SQL does not support any of these features.
- c) Embedded SQL allows you to place SQL commands into programs written in a general-purpose programming language. The result of the SQL queries can be retrieved into variables of that language. However, this is usually done one row at a time using the concept of a cursor, which is like an iterator over a set.

The architecture of most modern database systems is such that the system is broken down into a number of components, including forms editor and report writers. These components interact with the “database engine” through a set of procedures which provide access to the database. This set of procedures constitutes the call-level interface to the database. It can also be used to write database applications. It is usually a proprietary interface, although there are some standards.

Many database system vendors also provide proprietary programming languages for writing database applications. These languages provide the same facilities as a general-purpose programming language, but are augmented with special constructs for dealing with the set-based nature of the relational model.

Examples illustrating each of the above.

Examiners Comments

- a) Most candidates that answered this question were able to describe the purpose of each of the database tools. However, many students were not able to describe in detail the features available in a form editor. Very few students were able to describe the features of a report-writer. Very few students used examples of systems they had used in answering the question.
- b) Most students were able to recognize that SQL is a query language and not a programming language. However, many were not able to identify the features that are missing from SQL that would make it suitable as a programming language.
- c) Most students were able to describe the basic idea behind embedding SQL in a programming language like C or COBOL. However, not many were able to describe the concept of a cursor and result sets. In addition, not many students illustrated their answers with examples.

Very few students were able to describe what a call-level interface is. Those that did were able to identify how cursors and result sets are represented. They were also able to illustrate their answer with examples.

QUESTION 2

- a) What are the main distinctions between the database approach and the file-processing approach to data processing? (8 marks)
- b) Two of the main functions of a database administrator are authorising access to the database and managing the performance of the database system.
- i) Describe the features of SQL used by a database administrator for authorising access to the database. (8 marks)
- ii) Describe the basic principles managing the performance of the database system. (9 marks)

Answer Pointers

- a) Controlled redundancy
Controlled access
Persistent storage for program objects and data structures
Permitting inferencing and actions based on rules
Providing multiple user interfaces
Representing complex relationship among data
Enforcing integrity constraints
Providing backup and recovery
- b) i) A description of the GRANT and REVOKE statements of SQL along with their various options is required here. Should include a description of the different privileges that can be granted and the different levels at which they can be granted (that is, database, table, column).
- ii) A brief discussion of the principles of database performance management covering:
- The goals of tuning
 - The different statistics that can be gathered
 - The different problems that need to be managed: allocating locks, buffers, disks, RAM, processes/threads; managing the overhead of transactions and logging.
 - The use of indexes
 - The impact of database design
 - Query optimisation

Examiner's Comments

- a) Most students were able to answer the first part of this question well. The differences between the file-processing approach and the database approach were well understood, and the advantages of the database approach were identified by most students.
- b)i) Most students were able to identify the role played by the GRANT and REVOKE statements. There was a wide range of variation in how much detail was given in describing the various options available. Many students were not able to describe the different privileges that are available or the different levels at which they can be granted.
- ii) This part of the question was poorly answered by most students. The question sought a discussion that referred to the goals of tuning, the different statistics that can be gathered, the resources that need to be managed, the role of indexes, the impact of database design, and the role of query optimisation.

QUESTION 3

A bank requires a database designed to keep track of the accounts of its customers. Each customer may have a number of accounts. Each of these accounts may be in the name of one or more customers. Each account is located in a particular branch of the bank. In addition, each customer may have zero or more loans with the bank. Each loan must be with one or more customer. Each loan is also located in a particular branch.

The database must record the following information:

- The name of the bank, its address, and its main telephone number;
- The address of each branch, the sort code for each branch;
- The name of each customer, the customer's address and the customer's telephone number;
- For each bank account, the account number, its balance, and the type of account;
- For each loan, the loan account number, the outstanding balance, and the interest rate.

The database must also record the various relationships between the different entities in the database.

- (a) Draw an entity-relationship diagram for the database. In your diagram, illustrate which attributes are key attributes, which relationships are optional and mandatory, and the cardinality of the relationships. (9 marks)
- (b) Define a set of database tables that implement the entity-relationship diagram. Indicate which attributes will represent the primary key for each table, which attributes are foreign keys, and which attributes cannot have a null value. (9 marks)
- (c) The referential integrity rule does not allow unmatched foreign key values. What would be the consequences of this rule for your database design? (7 marks)

Answer Pointers

- a) The diagram should contain all the entities mentioned in the specification, along with their attributes and the relationships between the entities. It should indicate the cardinality of each attribute and relationship, which relationships are mandatory and which are optional. It should also indicate the key attributes of each entity.
- b) The set of tables should be capable of storing all the data specified by the ERD diagram from (a). In particular, it should represent all the relationships and any multi-valued attributes in an appropriate manner. It should indicate which attributes are primary and foreign keys, and which attributes do not allow null values.
- c) When rows are added, updated or deleted from the database, no unmatched foreign key values must remain. If there are such unmatched foreign keys, these will have to be dealt with. The database definition must define how this is done, or the programs updating the database must not allow it to happen.

Give examples of where this would arise in your database design.

Examiners Comments

- a) Most students were able to draw an ER diagram that represented the required database. Most students were able to represent the attributes of the entities correctly. Some students had difficulty in identifying the correct cardinality of the many-to-many relationships between customers and accounts and between customers and loans. Many students had difficulty in properly identifying the optional relationships between customers and accounts and between customers and loans.

- b) Most students were able to identify that the entities in the ER diagram became tables in the database. However, many students were not able to properly represent many-to-many relationships using an intermediate table. Often these relationships became one-to-one relationships. Very few students indicated which attributes were foreign keys, and which attributes could have null values.
- c) A lot of students had difficulty in identifying the consequences of the referential integrity rule for database updates. The restrictions placed on insertions, deletions and updates were not properly identified by most students.

QUESTION 4

Consider the following two database platforms:

PLATFORM 1:

File server platform: with a file store deployed on a server running network software and many client PCs each running database packages such as MS-Access.

PLATFORM 2:

A web based client server platform: with a database system deployed on a web server and many client PCs querying and updating a database over the Internet using a web browser such as Netscape or Internet Explorer.

- a) Outline the functional differences between each of the above platforms in terms of processing distributed data. (8 marks)
- b) Describe an application that would use PLATFORM 2 above. In particular describe a range of querying and updating activities that your application requires. (8 marks)
- c) Describe a technology that supports the processing of dynamic data content in web pages, based on PLATFORM 2 above. (9 marks)

Answer Pointers

- a) This part of the question requires a brief explanation of these two different platforms. Then the question seeks to determine the candidate's practical understanding (perhaps as users) of various differences, namely:

PLATFORM 1 (based on a file server). Here the database software represented by the database is hosted on PC clients with data access implemented by the issuing of requests for *files* held in a user account on a file server. Clearly the concept of sharing the same file is controlled by the operating system/netware so the opportunity for data sharing/concurrent access is limited to read only snapshots of the database file (note for MS-Access this is a .mdb file). It is worth noting that the client needs the same processing power as if the database file was hosted on the same machine as the database software package.

PLATFORM 2 (based on a client server). In contrast the database is hosted on a dedicated database server which controls the access to the data by issuing *SQL* requests either through a meta-language (such as Transact-SQL) or via the ODBC protocols which identify drivers for connecting different databases to clients. Clearly the DBMS plays a similar role to the operating system/netware in platform1 and thus the opportunity for data sharing and concurrent access is vastly improved. It is worth noting that the number of users supported can easily exceed that of Platform 1 by simply allocating more resources to the server (usually faster CPU, more memory, RAID disks etc). The issue of whether the client supports a web browser or other client tools (such as Oracle Forms) has little bearing on the answer to this part, rather the web based client server was presented to focus on the answer to part b) below.

- b) This part of the question concentrated on Platform2 due to its greater significance. Thus candidates had to be familiar with dynamic data content in applications associated with databases. A typical application would be on-line ordering of products (eg amazon.com) or accessing rail timetables (eg railtrack.co.uk). In each case these web sites reflect the current state of a constantly changing database (ie stock levels change or the train-timetable is changed due to a de-railment). Candidates should explain the basic user interface for these types of web-sites, that is a Forms based presentation tool with various GUI elements such as drop-downs, radio buttons and the all important SUBMIT command button. Some idea of the interaction with the chosen application should be outlined.
- c) This part seeks to determine if the candidate is aware of the technology implications of CGI (Common Gateway Interface) and similar technologies (ie ASP-Active Server Pages, JSP Java Server Pages). Only the general principles are expected at this level, namely: a cgi script request is initiated by a user running a web-browser such as IE/Netscape. The embedded tag is decoded as a database command and runs a CGI script on the web server. The database access will require access to a database server possibly on another machine. Therefore the processing of the CGI script will extract the SQL string (embedded in the script code) and initiate a process on the database server. The result of the SQL will be passed back to the client as HTML and the user will see a result set appear as a Form. Clearly each CGI program represents a single connection to the database and thus the database connection is usually closed after the CGI program has run to allow other connections to the server in a concurrent fashion.

Examiner's Comments

- a) For some reason most candidates could provide an informed description of only one of these platforms, rarely both. More often than not candidates could not perceive any differences or their explanations were decidedly weak and confused. The examiner finds it disappointing that candidates could not reason about the essential differences between client server and file server and this seems to indicate a lack of practical computing experience.
- b) As would be expected most people could describe an application. However the differentiation between querying (requiring a refresh operation) and update (with data integrity) was missing in most cases. Also missing was the database role in many of the applications that were described. There were a few cases where candidates misunderstood the question and assumed an application meant ASP and thus they in effect answered part c) in the process!
- c) This part of the question was poorly answered and often not attempted. Very few candidates could explain the CGI mechanism at all even though it is a fundamental web-database technology which many candidates with web access would have direct experience. Practically everyone chose ASP as the technology to describe, not the simpler and more fundamental CGI scripting.

QUESTION 5

Examine the two Tables AUTHORITY and CONFLICTS given below.

Table AUTHORITY shows which user is authorised to run a transaction. For example, Bill is authorised to run transaction Ta.

Table CONFLICTS below contains pairs of transactions that conflict. For example, Tb conflicts with transaction Tc.

AUTHORITY

<u>USER</u>	<u>TRANS</u>
Bill	Ta
Bill	Tb
John	Tc
John	Tb
John	Tf
Alex	Tf

CONFLICTS

<u>T1</u>	<u>T2</u>
Tb	Tc
Tf	Tc
Ta	Tc

- a) Using SQL list those Users who have authority to access more than one transaction. (8 marks)
- b) Using SQL, construct a View called 'PAIRS' that finds all the pairs of transactions that each User can access. (10 marks)

The View PAIRS would contain the following sample data:

PAIRS

<u>USER</u>	<u>TRANS1</u>	<u>TRANS2</u>
Bill	Ta	Tb
Bill	Tb	Ta
John	Tc	Tb
John	Tb	Tc
John	Tb	Tf
John	Tf	Tb
John	Tf	Tc
John	Tc	Tf

- c) Using SQL, create another View that finds which user has access to each conflicting pair of transactions.

For example the View should return:

USER John *has access to the conflicting transaction pairs* Tb,Tc and Tc,Tf (7 marks)

Answer Pointers

[The following code in italics is SQL-92 compliant and has been tested on SQLServer7. The SQL code can be run as a script or broken down into individual parts. Remember though to follow the order of execution.]

```
drop table authority;  
drop table conflicts; /* drop used in case these Tables are already created!  
/* Now create sample Tables and populate with sample data */
```

```
create table authority (userid char (6), trans char(2));
create table conflicts(t1 char(2),t2 char(2));
```

```
insert into authority values('Bill','Ta');
insert into authority values('Bill','Tb');
insert into authority values('John','Tc');
insert into authority values('John','Tb');
insert into authority values('John','Tf');
insert into authority values('Alex','Tf');
```

```
insert into conflicts values('Tb','Tc');
insert into conflicts values('Tf','Tc');
insert into conflicts values('Ta','Tc')
```

```
/* part a) */
select count(*) as Count_access,userid FROM authority
GROUP BY userid
HAVING COUNT(*) >1
```

```
/* RESULTS (run on SQLServer7) */
Count_access userid
```

```
-----
2      Bill
3      John
```

```
/*part b) */
drop view pairs
CREATE VIEW PAIRS AS (
SELECT DISTINCT x.UserId, x.trans as trans1, y.trans as trans2
FROM authority x, authority y
WHERE x.userid = y.userid
and x.trans <> y.trans)
```

```
/* RESULTS */
select * from pairs;
```

```
/*
UserId trans1 trans2
```

```
-----
Bill  Ta  Tb
Bill  Tb  Ta
John  Tb  Tc
John  Tb  Tf
John  Tc  Tb
John  Tc  Tf
John  Tf  Tb
John  Tf  Tc
*/
```

```
/* Part c) */
Create view AnotherView as(
SELECT userid,pairs.trans1,pairs.trans2
FROM Conflicts,Pairs
WHERE pairs.trans1 = conflicts.t1
AND pairs.trans2=conflicts.t2)
```

```
SELECT * FROM AnotherView
/*
```

```
userid trans1 trans2
```

```
-----
John  Tb  Tc
```


John Tf Tc
*/

Examiners Comments

Candidates had a lot of trouble solving these queries. Marks were awarded even if the SQL was not expressed correctly. Candidates are expected to be fairly fluent in SQL at this level. Dealing with each part:

Part a) The use of GROUP BY (for generating the set by aggregating the COUNT(*) is critical in this query. HAVING (rather than a WHERE clause) is necessary as the qualification COUNT>1 can only be applied to a set in this context.

Part b) Two join conditions are required, an equijoin and a nonequijoin. The latter join is used to delimit the result to those transactions that are not the same in the two Tables x,y. Note the use of alias names for a single Table so that the Join is in effect between two copies (x,y) of the original Table Authority.

Part c) The formulation of part b) used the View PAIRS from part c). This is a straightforward Join though it does require a bit of working out before the realisation of the simplicity of the solution becomes apparent.

QUESTION 6

a) Describe the features of a commercial visual software development tool/environment supporting the generation of screen Forms, used for developing user interfaces to database applications. (7 marks)

b) Draft out the design for a sequence of screen Forms that are required to meet the specification given below.

Your design should include:

- Diagrams of screen shots showing how the user would interact with your Forms.
- The processing requirements of data input and output between the database and the User Interface.
- An indication of how your design ensures the integrity of data input by the user.

(18 marks)

Specification:

A computing department in a college has a client-server computer network in which students progress through four Computer Aided Learning (CAL) software packages running on the client PCs. The college wants to monitor and record student access to the software packages in a session log. Students have access to the CAL software packages during a one-hour period each week, for a total of six weeks. A database Table called 'Logins' stores the session logs and contains the following structure:

*SessionID (the primary key),
StudentID,
SoftwarePackage,
Log-in Time,
Log-out Time,
Login Date.*

A student can only access one software package at a time but may access another package during the same one-hour period. There may be many students accessing the same package during a session, therefore the SessionID is unique for each login.

Another Table called 'Students' contains data on students, indicating who are registered users of the system:

StudentID (the primary key)
StudentName
CourseName

To log-in to the software package a student supplies their StudentID followed by their CourseName. After successfully logging in the student selects the software package they wish to use and continues until they have finished the session.

When a student has finished they must log-out and their session log is updated with the log-out time and date. Students can subsequently log back in again within the hour period and start another session or log out altogether.

Answer Pointers

- a) This part of the question should determine if candidates are aware (usually by practical experience) of the various database support software used to develop user interfaces to a database system. Candidates should realise that most of these tools need not be DBMS-vendor specific and that they are usually deployed on the client side with built in software tools to support the following functionality:
- Connection to Database set at design time
 - Data View windows showing graphically the data sources and Query Builders
 - Forms Generators based on the underlying Tables and Views that are required
- The major features imply knowledge of a particular development tool, such as Microsoft's Visual Studio, Oracle Forms or even Java Beans.
- b) This part of the question tests candidates design skills within the context of event-driven visual programming typical of database applications. Specific application/program design methods were not expected. However candidates should undertake the design in an informal but structured fashion which should be revealed in the following stages:
1. Decide on the overall Form structure (that is identify each form and decide the means of navigation).
 2. For each Form design the layout or structure of data aware components (such as fields, tables). The answer should include an example to demonstrate practical issues of Form design and an explanation of the design decisions that were made.
 3. Having designed the Form structure the 'behaviour' or functionality of each Form needs to be explained. The best way to do this is to identify control events and actions that take place following user interaction. These can include basic *Commit* events ie when a user clicks a button the parameters entered into fields are compiled into a SQL statement that for example accesses the table *Logins* and updates the database within certain contexts (ie the student has not already logged in is accessing another package at the same time).

Clearly in the short time available only superficial consideration should be given to integrity control related to the application and this is best revealed in succinct visual representations of screen shots and brief commentary where appropriate as specified in the question.

Examiner's Comments

- a) There were no problems with this part of the question. Most candidates could describe a database development software tool they had used, the most popular were those tightly associated with Windows platforms. There was scope in the question to consider integrated development packages which embrace wizards and user-focused tools.

- b) Most answers demonstrated a good understanding of the concept User Interface design issues but a number of candidates could not appreciate that a Form is simply a metaphor for a paper form and has to be related directly to an underlying database Table or View. This is why the Table structure for *Logins* was given in the question. Having the Table defined should have avoided any confusion about what UPDATE operations were required. However many candidates neglected this and often produced a design ignorant of the underlying Table structure and consequent data integrity issues. This probably stems from a lack of practical experience of developing practical solutions to real problems. The examiner was generally disappointed in the presentation of candidates' designs, in particular messy diagrams and lack of reasoned discussion. The informal design structure suggested in the answer pointer was rarely evident and the resulting presentations were mostly scrappy and hard to follow. Thus without any reasoned design commentary it was not easy to award marks to these scrappy efforts.
- c) Not very many people were able to answer this part of the question clearly. The answers were vague and confusing.
-