THE BCS PROFESSIONAL EXAMINATION
Diploma

April 2005

EXAMINERS' REPORT

Database Systems

Question 1

1. Relational theory and relational algebra are the foundation of modern relational databases. For EACH of the following terms, describe and discuss the concept involved. You should support your answer with your own simple examples and any appropriate diagrams.

a) Union Compatibility
b) Referential Integrity
c) Entity Integrity
d) Results Set
e) Domain Integrity **(5 x 5 marks)**

Answer Pointers

There were five sub-topics, each worth 5 marks:

Union Compatibility – generally misunderstood by the students - who discussed the UNION set operator rather than the concept of union compatibility – although some students did grasp the difference. Most produced Venn diagrams and a simple UNION example. I gave some credit (1-2 marks) if the UNION work was good but only accurate answers got 4-5.

Referential Integrity – almost universally well done by all students – they all grasped the foreign key issues with good examples. This was the best of the five topics.

Entity Integrity – again, very well done by most students with the issues of non-duplication (uniqueness) of primary keys mentioned by nearly everyone. The issue of no NULLS was less well covered.

Results Set – surprisingly badly done (many students actually skipped this), although many grasped the key idea of applied logic generating different results sets depending on the SELECT/WHERE clauses.

Domain Integrity – performance was average to good here with most students covering NULL/NOT NULL and data types / domains quite well but less well covered were CHECK constraints and defaults.

Question 2

2. Using your own simple SQL code examples, discuss and explain how SQL handles the following data extraction requirements. You must clearly annotate and explain each example used and comment upon any relevant issues.

a) Presenting aggregated and summarised data **(10 marks)**
b) Searching for incomplete or uncertain data **(10 marks)**
c) Extracting data from multiple tables **(5 marks)**

Answer Pointers

This consisted of three sub-questions:

Presenting Aggregated & Summarized Data – the best of the three with nearly all students discussing MIN, MAX, COUNT, SUM and AVG. The down side was that the examples supplied, were in the main, simple SELECT statements and only a minority gave more complex GROUP BY and HAVING examples.

Searching for Incomplete or Uncertain Data – often misunderstood by many – who discussed the use of NULL in tables or the role of operators such as > or < rather than the use of wildcards and the LIKE operator (as I intended). I gave a small amount of credit to these. However, many students did grasp the meaning of the question and gave several very good LIKE/wildcard search statements using underscore (_) and %.

Extracting Data from Multiple Tables – most students correctly discussed JOINS and the better ones talked about the different flavours (INNER, OUTER etc). All students gave at least a simple join example. Interestingly, many students also introduced the idea of sub-selects – which although not directly relevant did gain one or two marks if well presented.

Examiner's Comments – Questions 1 and 2

I got the distinct impression that most students were very confident on SQL issues and could readily supply actual SQL code examples (excepting some of the slight misunderstandings mentioned above).

As regards the more abstract relational theory/algebra, the questions on referential, entity and domain integrity were obviously better answered than the union compatibility and results set topics – probably because the three integrity rules are so easily associated with writing actual SQL code (setting up foreign keys, primary keys, NOT NULL, CHECK etc).

The fact that hands-on, concrete technical skills are preferred by students is no surprise and over the two questions I was generally very well impressed by most students' performance.

Question 3

3. Highlight the problems and solutions for the management of concurrency in database systems by answering the following sub questions:

a) Explain the concepts of *transaction schedule,* and *serialisability*. **(8 marks)**

b) Describe briefly the kinds of problems that might appear when a schedule is not serializable. **(9 marks)**

c) Explain what is a locking mechanism and how it can be used to resolve these problems. **(8 marks)**

Answer Pointers

3a

The student could have a different wording but importantly is the meaning of what he is writing a possible answer is a s follows:

A ***transaction*** is a logical unit of work that transforms a database from one consistent state into another. A transaction is not necessarily just a single update statement or database operation, it is rather a sequence database "update" operations. When database transactions are executing concurrently in an interleaved fashion, the order of execution of operations from the various transaction forms what is known as **transaction Schedule** (**or History**). Serial Schedules is a schedule where all operations of each transaction are executed consecutively without any interleaved operation from other transactions. Non-serial Schedule is a schedule where the operations from a set of concurrent transactions are interleaved.

(4 marks)

Serializability: is a non-serial schedule, which allows transaction to execute concurrently without interfering with one another. A given interleaved execution of a set of transactions is considered to be serializable if it produces the same results as ***some*** serial execution of the same transactions, i.e. as if transactions were executed one after the other.

(4 marks)

3b)
The most usual problems that can arise when a schedule is not serializable are:
The ***lost update*** **problem**: Two transactions update the same database value and one of the transactions loses the update as it commits sooner.

(3 marks)

The ***uncommitted dependency*** **problem**: A transaction accesses/update a record that has been updated by another transaction that has not been committed and which has finally been rolled back. This means that the current value of that record is not the real value since the rollback operation will restore the original value. The transaction may then manipulate this value producing wrong results.

(3 marks)

The ***inconsistent analysis*** **problem**: A transaction sees an inconsistent state of the database and therefore performs an inconsistent analysis.

(3 marks)

3c)
Locking mechanism is a technique used to control access to data. Following a transaction request to access for example a database record, the record is locked and access is denied to other transaction to prevent incorrect update.
(4 marks)
The effect of the lock is to prevent other transactions from changing the object. Once a transaction that has locked an object finishes with that object, the lock on that object is released (*unlock*) and the object is made available to other transactions. This locking mechanism resolve the above concurrency problems and ensures serializability by requiring access to various data items to be done in a mutually exclusive manner, i.e. when a datum is modified then all other transactions that need the same piece of data cannot modify it.

(4 marks)

Examiner's Comments
Most of the students have answered this question quite well especially the last sub question concerning locking mechanisms. Candidates should try and cover enough depth and be related to the number of marks on offer instead of regurgitating a lot of irrelevant textbook material as it was the case for the locking mechanism sub question. However, answers to the question on the principles of concurrency and serialisability part were very poor with many candidates producing contrived answers irrelevant to the scenario. The examiner suggests that candidates should 'read around' topics such as this and also go through real example to appreciate the difference between serial and non-serial schedule in a concurrent environment.

Question 4

4. The table (**Figure 1**) on the next page [below] represents a sample report layout for a construction company that manages several projects. Each project has its own number (P-No), name (P-Name), and employees assigned to the project. Each employee has an employee number (E-No), name (E-Name), and a job classification (Job-Class).
The company charges its client by billing the hours spent on each contract. The charge per hour (Chrg-Hr) rate is dependent on the employee position or job classification (Job-Class). The total charges (Tot-Chrg) is the product of hours billed (Hrs-Billed) and charges per hour (Chrg-Hr).

P-No	P-Name	E-No	E-Name	Job-Class	Chrg-Hr	Hrs-Billed	Tot-Chrg
1	Harricane	101	John News	Elect. Eng.	65	13	845
		102	David Senior	Comm. Tech.	60	16	960
		104	Anne Ramoras	Comm. Tech.	60	19	1,140
2	Coast	101	John News	Elect. Eng.	65	15	975
		103	June Arbough	Biol. Eng.	55	17	935
3	Satellite	104	Anne Ramoras	Comm. Tech.	60	18	1,080
		102	David Senior	Comm. Tech.	60	14	1,920

Figure 1: A Sample Report Layout

A database designer was asked to develop a database from which the information contained in the Sample Report in **Figure 1** above could be generated. For this, he/she designed the **Project Table** (in **Figure 2** below) the structure of which matches the above report formats. He/she omitted the total charge attribute because he/she thought that it could be calculated using charge per hour (Chg-Hr) and Hours billed (Hrs-Billed).

P-No	P-Name	E-No	E-Name	Job-Class	Chrg-Hr	Hrs-Billed
1	Harricane	101	John News	Elect. Eng.	65	13
		102	David Senior	Comm. Tech.	60	16
		104	Anne Ramoras	Comm. Tech.	60	19
2	Coast	101	John News	Elect. Eng.	65	15
		103	June Arbough	Biol. Eng.	55	17
3	Satellite	104	Anne Ramoras	Comm. Tech.	60	18
		102	David Senior	Comm. Tech.	60	14

Figure 2: Project Table

You are asked to answer the following two questions:

a) As the Project Table in **Figure 2** (above) developed by the database designer is susceptible to update anomalies, provide ONE example of anomalies for EACH of the following:
ii) insertion
iii) deletion
iii) update anomalies **(9 marks)**

b) Using the functional dependency diagrams (fd1, fd2, etc…), describe and illustrate the process of normalisation from First Normal Form to third Normal Form for the Project Table in Figure 2 (above). In this process of normalisation, we assume that the **Project** attributes P-No, E-No and Job-Class could be used to determine the values of (P-Name), E-Name and Job-Class), and (Chrg-Hr) respectively. **(16 marks)**

Answer Pointers

4a

Examples of anomalies due to data dependency (the student may other example of anomalies): (3 marks for each)

- **Update anomalies.** Modifying the JOB_Class for employee number (E-No) 101 requires (potentially) many alterations. one for each E_No=101.
- **Addition anomalies.** Just to complete a row definition, a new employee must be assigned to a project. If the employee is not yet assigned, a phantom project will have to be created in order to complete the employee data entry.
- **Deletion anomalies.** If employee 101 quits, deletion must be made for every entry E_No=101. As such deletion is made, other vital data are lost, too.

4b

The normalisation Process:

First Normal Form:

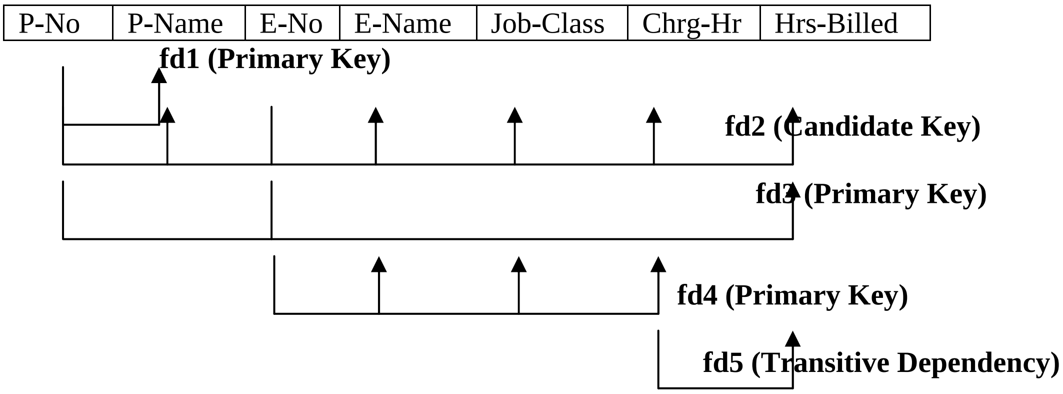
The table project contains repeating groups. To be First Normal Form, the project table must be fixed by making sure that each row defines a single record. The result of this process is shown in the Project table below:

Project

P-No	P-Name	E-No	E-Name	Job-Class	Chrg-Hr	Hrs-Billed
1	Harricane	101	John News	Elect. Eng.	65	13
1	Harricane	102	David Senior	Comm. Tech.	60	16
1	Harricane	104	Anne Ramoras	Comm. Tech.	60	19
2	Coast	101	John News	Elect. Eng.	65	15
2	Coast	103	June Arbough	Biol. Eng.	55	17
3	Satellite	104	Anne Ramoras	Comm. Tech.	60	18
3	Satellite	102	David Senior	Comm. Tech.	60	14

(3 marks)

The Dependency Diagram



Which could also be written as:

fd1: P-No --> P-Name
fd2: P-No, E-No --> P-Name, E-Name, Job-Class, Chrg-Hr, Hrs-Bill
fd3: P-No, E-No -> Hrs-Billed
fd4: E-No --> E-Name, Job-Class, Chrg-Hr
fd5: Job-Class --> Chrg-Hr

(3 marks)

2nd Normal Form

Based on the functional dependency fd1, fd2, fd3 and fd4 mentioned above, the original Project table will be split into three tables **Project, Employee and Assign**:

Project

P-No	P-Name

Employee

E-No	E-Name	Job-Class	Chrg-Hr

Assign

P-No	**E-No**	Hrs-Billed

(5 marks)

3rd Normal Form

As a result of the transitive dependency (fd5), the table Employee is split again into **Employee** and **Job** tables:

Employee

E-No	E-Name	Job-Class

Job

Job-Class	Chrg-Hr

Although the Job-Class attribute determines the Chrg-Hr, it is better introduce (optional) Job-Code and the job table will look like:

Job

Job-Code	Job-Class	Chrg-Hr

(5 marks)

The final normalised tables are as follows:

Project

P-No	P-Name

Employee

E-No	E-Name	Job-Class

Assign

P-No	**E-No**	Hrs-Billed

Job

Job-Code	Job-Class	Chrg-Hr

Examiners' Comments

A majority of students have attempted this question and most of them have gained good marks when answering this question. However, there are still two main problems where the students have to focus on: functional dependency and the difference between 2nd and 3rd normal forms. They need more reading and practice on functional dependency especially differentiate between partial and transitive dependency. This will help making clear the difference between 2nd and 3rd normal formal.

Question 5

5. **Please refer to Appendix A on the back page for this question.**

Produce a logical data model for the SWIS order processing system.

Your logical data model must include the following:

a) A description of the different user VIEWS that are supported. **(5 marks)**
b) An Entity Relationship (ER) model diagram. **(10 marks)**
c) The tables required including the primary and foreign keys and some sample attributes. **(6 marks)**
d) A justification of your modelling decisions and any assumptions you needed to make. **(4 marks)**

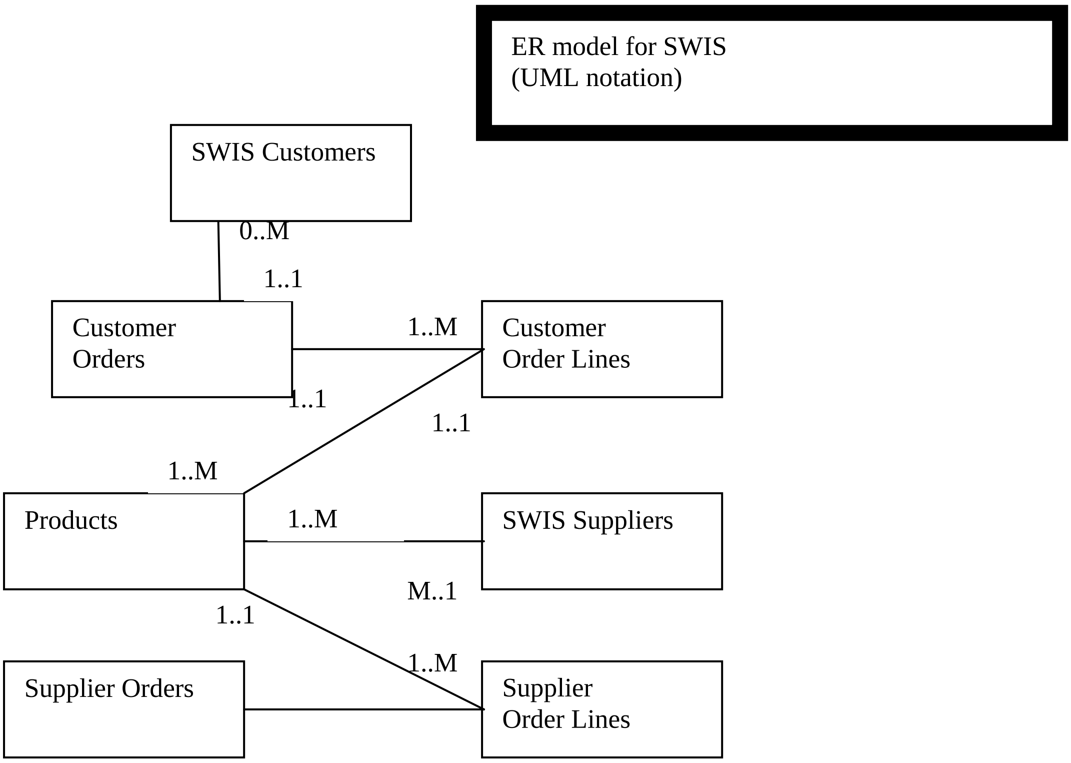
Answer Pointers
(Reference must be made to Appendix A)

Part a)
The following distinct user VIEWS are required:

- Customers View
- SWIS View of a processing a Customer order
- SWIS View of processing a Supplier order including supply, holding (warehouse) and delivery
- Supplier View of processing a SWIS supplier order

Examiner's Comments
This question was intended to 'lead' candidates into the ER modelling (part b) that follows and allow them to think about functional requirements. Many candidates misunderstood the concept of a VIEW as a SQL View programming context. The lesson here for candidates is to read the whole question first and then the context of VIEW should be interpreted correctly. Perhaps it was also the case that some candidates had little practical experience of data analysis and the value of user views.

Answer Pointers Part b)
ER model



Examiner's Comments
Largely disappointed with the answers produced, this revealed many flaws in understanding data modelling. There are key things that go wrong in answering this type of question the most important is candidates not reading the scenario thoroughly enough. Also accuracy is important, for instance in the designation of relationship/participation constraints. Many candidates could do more than simply recall 'stock' modelling solutions found in most textbooks (in this case an order processing system).

Many candidates could not identify the critical difference here – SWIS is ‘multi-processing system operating on the different views found throughout the system and not the single order processing system found in most text books. Only the more able candidates realised a model similar to the above was necessary and that supplier and customer types of order must be associated so that the warehouse knows which products a customer has ordered following receipt of a supplier order.

Answer Pointers Part c) and part d) – to include commentary on modelling

Tables required are indicative and largely depend on part b) as they are mapped from ER model.

A customer order requires 5 tables customers can browse the supplier catalog prior to purchasing

CustOrders (CorderID, custID ,order date, payment, status)
SWISCustomers(CustID, name, delivery address, reg_status) (for each customer)
CustorderLine(CorderID, productID, dispatch date,WarehouseRxdate, qtyordered, qtydispatched, lowest quoted price, actual price)
Products(ProductID, desc) (products ordered

The customer will only know about the lowest quoted price and when the delivery date is for individual products. But SWIS will need to link the supplier order that links to the customer order (for re-assembling later when goods are supplied)

SupplierCustomerorders(SorderID,CorderID, date created)

2. A Supplier Orders needs 4 tables one of these is a SupplierProduct table to determine which supplier supplies which product
SupplierProducts (SuppID, productID, date agreement, discount)

A supplierOrder table is need to assemble the customer orders in supplier orders

SOrder(SorderID, SuppID, orderdate, payment, status)
Suppliers(SuppID, name, address, service agreement)
SorderLine(SorderID,SProductCode, qty, discount, del date)

Suppliers will send invoices to SWIS for a SWIS order and similarly for SWIS to customers.

Examiner’s Comments

Candidates who lost marks on part b) also lost marks when tackling parts c) and d). In the main poor modelling decisions cascaded through each part of this question. However there was an opportunity to defend/argue about the modelling decisions and revisit the data model again. Some candidates actually did this and gained marks as a result. Some candidates who constructed an inappropriate data model also recovered some marks by correctly interpreting the mapping of ER models to tables.

Question 6

6. **Please refer to Appendix A on the next page for this question.**
SWIS and its suppliers are investigating the use of web technologies to allow remote access to their databases. One functional requirement will be to allow SWIS read access to a supplier’s database so that it can check the availability of products that a customer has ordered.

Describe THREE different configurations of a distributed information system that would support the business activities of SWIS and in particular, the functional requirement referenced above.

Your answer should be assisted by the inclusion of diagrams and sketches to explain concepts and any technical details. **(25 marks)**

Appendix A: SWIS (Simply a Web Information Service) for use in answering Question 5 and 6.

SWIS is a successful E-Commerce (web-based) business that customers use to purchase products at discounted prices over the internet.

SWIS makes money when it purchases products in large quantities following many customer orders for the same product. SWIS has negotiated deals with suppliers in which the higher the quantity of products purchased then the greater the discount and the greater the saving on the supplier's quoted price. These savings are passed proportionately onto the customer who will pay less than the price the supplier would quote an individual customer. SWIS provides a value-added service that will find the cheapest price, availability and order dates of products from a range of suppliers. Therefore at the time of ordering the customer will only know the lowest quoted supplier's price they will pay for a particular product. It is only when the products are ready for dispatch that a customer will know whether there is a discount on a supplier's quoted price. Another value-added service that SWIS provides is a comparative product review of similar products. SWIS requires customers to be registered before they can order products and/or use a value-added service.

There is a range of suppliers that SWIS deals with and each supplier allows SWIS read-only access to their product data, for example to determine stock levels and purchase price.

There are many customers who can make many orders. An order can contain many different products (called a shopping cart). The products on an order can be sourced from different suppliers.

The most complex type of order that SWIS needs to process is a multiple customer, multiple product and multiple supplier order. For example SWIS creates individual supplier orders from individual customer orders involving many products that are sourced from many suppliers. SWIS owns a Warehouse that holds the products sent from suppliers prior to dispatch/delivery to customers.

Answer Pointers
Reference must be made to Appendix A

Part a)

There were 3 possible configurations and it is appropriate to relate these to the needs of SWIS and not produce text book answers of generic systems.

- Web database logically distributed tightly coupled scaled out web servers– possibly database centrally organised but run a cluster of servers (known as a dependent partitioned database).
- Web services (loosely coupled). The database is accessed at different physical locations, for example the need to interact with a local client database (eg supplier database) and the main server database (SWIS orders) of SWIS.
- Replication. Here data is copied across sites but data access and the replication process is highly controlled. Replication control is provided by built in database software and services providing transaction management and recovery which web services does not offer at the database level.

The most simple and familiar being a dynamic web-database architecture (3 tier) with client server interaction which most candidates described as they probably had direct experience of using/developing.

5 marks were available for each configuration with 3 marks for identifying pros and cons for each + 1mark for allocation to other areas as appropriate.

Part b)
Most candidates were expected to be able to describe the nature of this type of distributed system which would involve technologies such as ASP server pages, PHP etc and present queries via dynamically generated data from a database via a web form for example. A better solution is to automate the transfer of data between databases running on different web sites using a technology called XML Web Services. It was expected that candidates were able to show a basic understanding of this technology (eg ideas behind SOAP protocol and XML) the fundamental concept of message passing and connectivity allowing web services to communicate via database parameter passing (SQL query, table name , service name) etc.

For SWIS it would be appropriate for a distributed replication service to run on every database server with data held locally and replicated on a master server. This configuration would the less practical as the web server solution as there is no need to distribute the database unless there is some need to improve performance, the critical performance issue is allowing customers fast access to supplier databases in a secure manner and therefore distributing the database would only exacerbate the problem.

Examiner’s Comments parts a) and b)
The above answer pointers should reveal the appropriate amount of detail expected at this level. It also reveals the idea of categorising different systems. Although there are different ways of categorising distributed systems it is good practice to do this and marks are rewarded as a result. This indicates clear thinking and problem solving skills. However candidates must back these up with assumptions and examples from SWIS. Again only the best candidates related to this aspiration just mentioned. Most candidates simply reproduced ‘stock’ answers from text books just hoping they related to SWIS. Unfortunately only the more recent textbooks have information on web services and therefore many candidates could not report on this exciting new development.