THE BCS PROFESSIONAL EXAMINATION
Professional Graduate Diploma

April 2006

EXAMINERS' REPORT

Advanced Database Management Systems

Question 1

1. *a)* "Domains are to relations as nouns are to sentences". (Date and Darwen 1998).

Discuss the above statement in order to establish the role played by the concept of domain in the relational model of data. **(9 marks)**

b) Discuss whether the concept of domain in the relational model is equivalent to the concept of class as defined in object-oriented terminology. **(16 marks)**

Answer Pointers

This question assesses "Relational Model Conformity" and "Object-oriented Systems".

a) A domain in relational terms is a set of values that define the values which may be permitted for an attribute. Domain sets may be finite or countably infinite e.g. a domain may contain a finite set of valid part numbers or be specified as containing values from the infinite set of integers.

A relation may be perceived as being a set of logical assertions:

Attributes	Part	Component
Domain	Part_Number	Part_Number
	P1	P2
	P1	P3

Here this relation makes two assertions:

Part P2 is a component of part P1
Part P3 is a component of part P1

Here the domains define what things can have an assertion made about them and the relation defines the assertion. This is analogous to a sentence which has a noun or nouns and makes a logical assertion about it or them.

b) In object-oriented terminology "A class is a blueprint or prototype that defines the variables and the methods common to all objects of a certain kind". If the term domain is associated with its realisation in many relational database products it is not at all obvious that it equates to the same thing as class. In such products domains are often limited to Number, Character_String, Money etc. On closer inspection, however, even these have some resemblance to a class. For example, the operations which are permitted on Number differ from those on Character_String and comparison operations will carry out different types of comparisons depending on which domain is chosen for an attribute. This is similar to the concept of class in an OO environment. There is no stipulation in the relational model that domains should be limited to these simple types or that operations should be system defined

and not user defined. There is no reason why a domain should not be user defined and stipulate a much more complex set of operations. Domains such as photograph and audio are perfectly acceptable and will support operations appropriate to the type of data they define. Domains in this sense act like types in strongly typed languages. The concept of type is effectively the same as class in a strongly typed OO language and therefore the relational concept of domain equates with class in such a language.

Examiner's Comments

Candidates attempting part a) of this question were expected to begin by giving a definition of the term domain. Disappointingly, many confused the concept with attribute or column. For full marks candidates needed to be able to appreciate that rows (tuples) in a relation can be perceived as a set of logical assertions. This perception makes sense of the quotation. Very few candidates demonstrated this insight.

Part b) of the question attracted only a few good answers. Some candidates were able to spot that a domain and a type might be related. Many, however were unable to see the link to a class and quite a few argued that since domains did not support operations that they could not be classes! The point of the question was to explore the fact that any database that offers full support for domains must have a capability of defining operations on domains. Such a system could be realised where an OO like programming language is used to define the domains.

Question 2

2. *a)* Give an example of a query which has two differing relational algebra implementations and explain why the choice of implementation may affect the time taken to execute the query. **(6 marks)**

b) When a query is optimised, it is essential that the revised query returns the same result as the original. List a number of relational algebra transformations that achieve this. **(12 marks)**

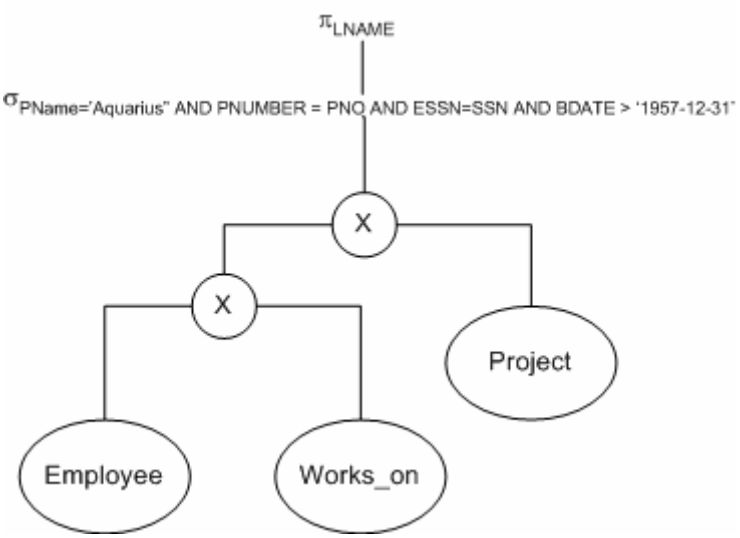
c) Compare and contrast heuristic query optimisation with cost-based query optimisation. **(7 marks)**

Answer Pointers

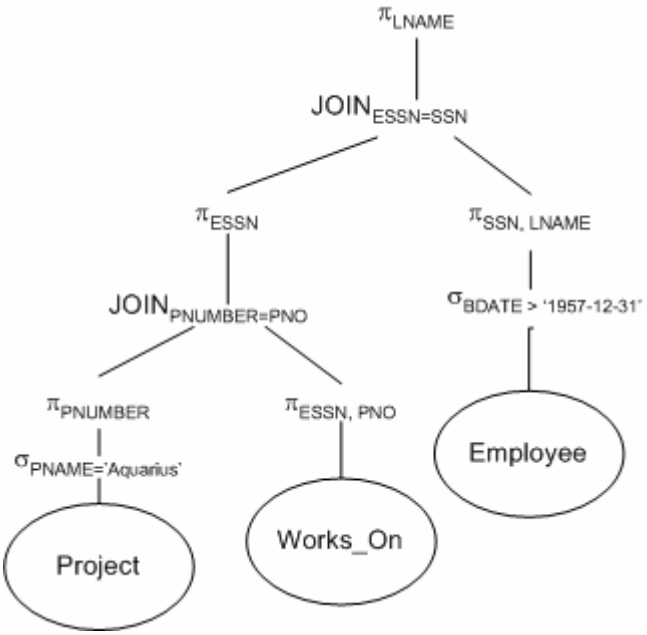
This question assesses "Query optimisation"

a) The speed at which a relational query operates can be affected by a number of factors. Some operations are just inherently slow e.g. cartesian product. Others are quite sensitive to the number of rows in a relation. For example, a join will take much longer if two large tables are joined as opposed to joining a small table with a large table. Some operations like restrict and project reduce the size of result tables whilst other operations such as cartesian product produce large results. Generally speaking carrying out operations that reduce the number of rows as early as possible will produce faster query performance.

The following RA tree



can be improved as shown:



b) Conjunctive restrict can be decomposed into individual restricts
Restrict is commutative
A sequence of projects can be amalgamated into one project
Restrict is commutative with project
Join is commutative
Restrict may be commuted with join
Project may be commuted with join
Set operations are commutative
Join, cartesian product, union and intersection are associative
Restrict may be commuted with the set operations
Project commutes with union
Some cartesian product, restrict sequences are equivalent to joins

c) Heuristic optimisation works on the basis that on average a given sequence of relational

operations runs faster than another sequence. In this method of optimisation restrict and project, which generally reduce the size of intermediate results, are performed as early as possible and cartesian products are avoided. On average this will produce a more efficient query but there will be some queries where the 'rule of thumb' is invalid. This will often happen where relations vary greatly in size. Heuristic optimisation takes no account of the current state of the database. Cost-based optimisation uses the data dictionary to estimate (rather than guess) the cost of a particular query strategy and then chooses the least cost strategy. Costs considered include secondary storage cost, temporary storage cost, computation cost, memory usage cost and in a distributed environment communication cost. Cost-based optimisation is more likely to select an optimal query strategy than heuristic optimisation, however, it requires a significant amount of extra computation. Cost-based optimisation is therefore more suited to compiled queries where the cost of optimisation can be spread across a number of executions.

Examiner's Comments

Part a) was generally well answered and the majority of candidates were able to give an example of a query where the order of execution of relational algebra operations might make a difference to query execution times. Many candidates were able to demonstrate this by considering the number of temporary table entries required in order to satisfy the query.

Part b) was largely bookwork, but was poorly answered. In general, candidates were aware that restrict is commutative with project but beyond this knew very few other legal relational algebra transformations.

In part c), only a few candidates seemed to know anything about cost-based optimisation. This made it impossible for them to contrast this type of optimisation with a heuristic approach.

Question 3

3. *a)* Explain, aided by example code, the differences between the following programming constructs that could be used to implement database integrity constraints:

- Triggers
- Check Constraints
- Stored Procedures **(9 marks)**

b) **Refer to Appendix A for this part of the question**
Examine the Business Rules in section A4 of Appendix A. Which of these Business Rules could be relatively easy to implement by programming a Trigger and which (if any) would not? Justify your answer. **(9 marks)**

c) One of your customers complains about the delivery of products which she did not order. Having investigated the matter the complaint is found to be genuine. Indeed someone reported a similar problem occurred before. It is suspected there are faults in the database software as other possible causes such as human error and a breach of security have been discounted. You are the leader of the database technical team charged with identifying possible causes of this breakdown of database integrity.

Write an account of the procedure you would follow to identify the cause of this breakdown of database integrity. Suggest possible solutions to avoid breakdown of database integrity arising in the first place.

Answer Pointers

Part a)

Trigger:

an event based activation in response to an exception that has been programmed on a Table and an action on that table. Candidates should supply example code such as

CREATE TRIGGER trg_trigger1 ON Tbl_Orders FOR INSERT AS
Exception : when x>99 THEN ROLLBACK TRANS

Candidates must also mention different types of trigger that control when the trigger is fired ie can be either BEFORE or AFTER types

Check constraint:
Part of SQL standard and embedded in the CREATE statement. Check constraints operate like triggers but do not need an activation code. In effect a check constraint will raise a db insert error. For example if a range of values being input do not match the check constraint set for a column.

CREATE TABLE tbl_orders (OrderNo, OrderDate date CHECK date < getdate(today())
the check constraint can also be set by an explicit add constraint

ALTER TABLE tbl_orders ADD CONSTRAINT chk_1 as CHECK date < getdate(today())

Stored Procedure:
A programmed exception handler that is explicitly forced (by a code block such as an exception handler) to check integrity usually during the course of another operation that has been programmed e.g.

CREATE PROC msp_AddNewOrders AS
BEGIN
Insert TBL_orders ….
Do while
RAISE EXCEPTION … go to exception handler
End while
END

Part b)
It was important to justify the decision to choose Triggers as the programming model for business rules. Also it is important to state the alternative(s) – such as application programming on the client or in middleware. Again it was expected candidates would develop an answer relevant to the scenario and its architecture (web based 3 tier Client Server) which predisposes some option of integrity checking on the client where possible using say Javascript.
Therefore going though each Business Rule, these are interpretative not definitive so reasoning must be given:
BR1: Unlikely that a trigger could code this complexity so it is best to program this in application code running on the application server (middle tier).
BR2: Yes could be built into a transaction 'purchase ticket' when the ticket table is updated checks the current transaction date with the offer expiry date.. Triggers are used where there is a need for database transaction integrity
BR3: Unlikely as membership authorisation must be validated using login screens and handled by more advanced security settings. Certainly the database will be accessed but a trigger would not be the best mechanism to use here.
BR4: No this business rule requires a lot of computation probably via a decision support system such as Analysis Services/OLAP. The business rule should never be violated as this would lead to inefficiency so the system would pre-compute the data required to uphold the business rule.
Consider these issues:
Middleware can cache data from the database and therefore could perform checks more efficiently if changes to the database are infrequent. This is still server side validation and the middleware supports more programming features (being a high level OO language such as Java/c#).

Candidates should also reveal the limitations of triggers (i.e. harder to maintain) and harder to debug with consequences of side effects (cascading and race conditions). Overall advantages would be performance and doubling up on checks that have been bypassed (i.e. caused by security failures)

Part c)

There are a couple of ways of answering this question. Good candidates will realise the DBMS already has sound built-in integrity checking and this should be mentioned – e.g. Concurrency control/ process synchronisation but the good candidates will also realise these mechanisms are not fool proof and indeed many Database Administrators (DBAs) often relax them to get better performance. Therefore a lock on a shared object could have been relaxed (read access causes wrong interpretation by a user and that user proceeds on that basis). Candidates own illustrative examples/experience gained extra marks.

Examiner's Comments
Overall
There were far too many weak/ un-substantive answers for this level. Many candidates simply recalled bookwork and did not apply their answer to the scenario and therefore offered no real practical solutions**.**

Part a)
Candidates had to demonstrate an understanding of database integrity and what constraints are relevant applied to each construct. Commented examples of code in any DB sql-2 compliant format was required but these must be clear and concise enough to show different ways of doing the same thing rather than re-invent new scenarios.

Part b)
This was an open discussion type of question and it was expected that candidates would argue for Trigger when performance is paramount or when the activation is better handled by the client.

Part c)
Another open type of question as the answer could be broad ranging so marking scheme must be flexible.

Question 4

4. *a)* Explain the characteristics that differentiate a Data Warehouse from a Database System. **(7 marks)**

b) Explain why a Data Warehouse is *unlikely* to contain data that is in 3rd Normal Form. **(4 marks)**

c) **Refer to Appendix A for this part of the question**

i) Describe an application area relevant to SWIFT that would benefit SWIFT from the support of a Data Warehouse and the associated analysis tools. Include in your answer the type and nature of the business data and the analysis tools that would be used in a Data Warehouse. **(10 marks)**

ii) Explain how transaction data added to the SWIFT database could be made available for use in a Data Warehouse. **(4 marks)**

Answer Pointers
Part a)
A Data Warehouse (DW) is a large de-normalised repository of data used for analysis and decision support. Often the application type is known as OLAP (On-line analytical processing) where as a DBMS is primarily designed for OLTP type data (transactional) and is required to perform high volume updates and fast queries in real time. A DW is more consolidated to form OLAP analysis of operational data collected over many years. A DW is more like a product of a DBMS application development and not always easy to integrate at run time. A DW has a different strategic importance and is geared towards building decision support systems and a knowledge-based applications architecture and environment that supports both everyday tactical decision making and long-term business strategizing. The Data Warehouse environment positions a business to utilize an enterprise-wide data store to link information from diverse sources and make the information accessible for a variety of user purposes, most notably, strategic analysis.

Part b)
Mainly because the data is aggregated into cubes that are cross joined by dimensions such as time, price and geography for example. This means the cube with its precomputed data can quickly be searched and analysed by 'drilling' down or through the data as you would with using an pivot table/cross tab query on a spreadsheet (but on much larger scale). Similarly pivot tables express hierarchical data sets.
Until the advent of Data Warehouses, enterprise databases were expected to serve multiple purposes, including online transaction processing, batch processing, reporting, and analytical processing. In most cases, the primary focus of computing resources was on satisfying operational needs and requirements. Information reporting and analysis needs were secondary considerations. As the use of PCs, relational databases, 3NF normalised data and end-user computing grew and changed the complexion of information processing, more and more business users demanded that their needs for information be addressed. Data Warehousing has evolved to meet those needs without disrupting operational processing.

Part c)
Any application perhaps but must be business centred and OLAP enabled for pattern matching and trend analysis involving price sensitivity, time seasonal etc.

All data in Data Warehouse is accurate at some moment in time, providing an historical perspective. This differs from the operational environment in which data is intended to be accurate at the moment of access. The data in the Data Warehouse is in effect, a series of snapshots. Once the data is loaded into the enterprise data store and data marts, it cannot be updated. It is refreshed on a periodic basis, as determined by the business need. The operational data store, if included in the Warehouse architecture, may be updated.
SWIFT would use DW for analysing trends in travel plans and may integrate other data such as individual data collected on customers. This is what a loyalty or membership scheme is used for. The tools used depend on vendor of the DW but for example SQLServer includes a tool set called Analysis services and contains various wizards that assist users build the structure and maintain the content of a DW. Further tools include extended SQL to handle hierarchical data and structured data based on 'cubes', Depending on the application DW could be assisted by data mining but this was not essential to mention.

Transaction data is incorporated into a DW using tools such as Data transformation services which filter and clean source data so that it is consistent and maybe summarised to the correct form. Transaction data would need to be transformed into the new structure and regularly maintained.

Examiner's Comments
Part a)
It was pleasing to see most candidates were familiar with the concept of a DW and associated tools. However some candidates seemed to confuse the DW idea with data mining.
Part b)
Some very unconvincing answers – perhaps examples would be better than the long-winded attempts that were received.

Part c)
Open question but the scenario has a lot of opportunities for imaginative answers which only the best candidates could muster. The key to a good mark was an understanding of what a data warehouse could be used for and an idea of the scale and scope of the SWIFT application. Too much emphasis on data mining seemed to be present perhaps to disguise a lack of knowledge of more important associated DW concepts. Again avoid writing just on bookwork – apply ideas and concepts.

One or two first class solutions are worthy of mentioning:
- An example of the integration process (of OLTP with OLAP)

- Treatment of generic codes such as price codes.
- Within SWIFT various applications may represent price codes in different ways:
Price is always represented in a consistent way, regardless of the many ways by which it may be encoded and stored in the source data. Appreciation of this and similar well thought out strategies were present in the best answers to this part.

Question 5

5. Refer to Appendix A for this question

a) Describe the distributed information system technology known as *'Web Services'*. Explain what Web Services facilities would be required and explain how they could be configured to support the following functionality:-

SWIFT on behalf of its customers is investigating the use of web technologies to allow remote access to all of the airline operator databases. One functional requirement will allow SWIFT to have read access to airline operators databases so that they can check the availability and price of seats on any flight.

(12 marks)

b) Describe the characteristics of a distributed database technology that uses data replication. Explain how a data replication approach used by SWIFT would differ from the Web Services approach you described in your answer to part *a)* above.

(13 marks]

Answer Pointers

Part a)

Web services are platform-neutral Web-based enterprise applications that use open, XML-based standards and transport protocols to exchange data with calling clients. Various major vendors have their own flavour supporting major vendors products typical Microsoft (.NET) or Sun (the Java 2 Platform) and IBM. These vendors supply APIs and tools that are needed to create and deploy interoperable Web services and clients. Candidates should therefore be familiar with at least ONE vendor's web-services development tools to answer this question.
For example relevant to SWIFT
SWIFT would use Web services and clients to pass parameter data to the method calls supported by an Airline DB (needs to enabled with remote connection and distribution of message calls) and process the data returned. For document-oriented web services such as SWIFT invoices it would be possible to send XML marked up documents containing the service data back and forth. No low-level programming is needed. The following list describes typical Web services/XML APIs.

API for XML Processing (e.g. JAXP) supports the processing of XML documents using Document Object Model (DOM), API for XML Parsing (SAX), and XML Style sheet Language Transformation (XSLT). The JAXP API enables applications to parse and transform XML documents independently of a particular XML processing implementation.

The Java API for XML Registries (JAXR) lets you access business and general-purpose registries over the Web. JAXR supports the ebXML Registry/Repository standards and the UDDI specifications.

The Java API for XML-based RPC (JAX-RPC) uses the SOAP standard and HTTP so client programs can make XML-based remote procedure calls (RPCs) over the Internet. JAX-RPC also supports WSDL so you can import and export WSDL documents. With JAX-RPC and a WSDL, you can easily interoperate with clients and services running on Java-based or non-Java-based platforms such as .NET.

The SOAP with Attachments API for Java (SAAJ) enables you to produce and consume messages conforming to the SOAP 1.1 specification and SOAP with Attachments note.

Part b)
Large corporations have databases that are physically distributed and have targeted operational computers serving critical applications which also demand physical separation of core data sources. There are critical needs in the industry for distributed systems that seamlessly update and accurately synchronize replicas in real-time and that provide fail-safe recovery during times of major security breaches or physical attack. – This is managed by REPLICATION services built into a DBMS and managed in the same way as a centralised database.
Some technical discussion of relevant Web Service API's is expected relevant to the discourse which uses web services to see inside remote databases (airline operator) and to exchange documents. SWIFT would therefore see REPLICATION as a way of scaling out there DB operations and also autonomising there departments so that they can hold data that is operational correct (e.g. booking at centre X – data held locally and distinct from bookings at centre Y). However SWIFT managers/directors want to see the bigger picture so the replication of data is 'merged' to form a unified physically separate but logically centralised data store.
The differences (replication/web services) are not so distinct as in practice both web services and replication need to work together. Web services usually see 'distributed transactions' with data still held centrally (logically) in fact web services discovers the distributed data (sing SAX/WDDI). Instead Replication has to know what it is connecting to and its configuration. So Replication is tightly coupled whereas Web services is loosely coupled but beyond that both can interoperate together due to the nature of web transport protocols and techniques such as HTTP and XML.

Examiner's Comments
Part a)
Excellent candidates were expected to describe standards and relate these standards to vendor web services API's. Weaker candidates seemed to recourse to book work and this will be reflected in the lower marks awarded. However only about 10% of candidates actually knew what a web service architecture was. This was very surprising given that the technology is quite recent it is still based on a core element of the syllabus – CORBA and RMI (in other words distributed object oriented information systems). Many candidates thought web services represented ASP or JSP type applications with the client-server model, those candidates lost a lot of time for little reward that went down this path.
The most worrying aspect was the inability to draw a simply information system architecture for SWIFT with some indication of protocols and message exchange. A diagram would have helped explain web services if the technical know-how was weak.

Part b)
Generally much better knowledge of replication but not very deep knowledge, particularly about synchronisation of data replicas.
Only the best candidates could see the close link between these two technologies and that they can both be used cooperatively even though the question hints at a disjunction.

Appendix A: SWIFT (Saving With Internet Flight Travel)
SWIFT is an agency that sells discounted seats on flights on behalf of some low cost airlines. These discounted seats are called 'Offers' and are only available for purchase by registered customers over the internet. There are a number of low cost airlines that SWIFT is an agency for and sometimes more than one airline can offer seats with the same start and destination. Each seat offered for sale contains some information that is supplied by the airline operator:-

- The Ticket Number that identifies the seat on a particular flight.
- The start and destination of the city/airport for the journey undertaken.
- The date and time of the flight/journey and the check-in time.

Some information is derivable and depends on the number of confirmed customer bookings/sales of seats, this information is held on the SWIFT database:-

- The baggage allowance. The baggage allowance for passengers is reduced if the aircraft is carrying cargo as well as passengers. .
- The price of the seat. The price starts at a low discounted price and increases as the bookings fill up. The earlier the seat is purchased then the cheaper the price that a customer pays. The current price is the price quoted for customers wishing to purchase seats. This may also be the final price, particularly if customers book a seat near the departure date of a flight.

It is the responsibility of SWIFT to determine the price of a seat, they have to be competitive, however they must cover the operating costs of the flights. SWIFT have negotiated a contract whereby they get 12% of the profit that an airline makes each year from the passenger flights it is commissioned to sell seats.

Figure A1 (below) shows a sample ticket purchases and contains data that was captured today (assume the date is 3rd December 2005). **Figure A2** shows the customer details.

Figure A1: TBL_Tickets

TicketNo	CustID	FlightCode	Quantity	PurchDate	Current/Final Price	Price Paid	FlightDate/Time
1962	343371	BG_8971	1	12/Nov/05	231.99	219.56	01/Dec/05 12:23
1963	343371	GTX_281	2	12/Nov/05	211.67	211.59	14/Nov/05 16:34
1964	343371	TL121_281	1	12/Nov/05	2040.72	833.59	04/Dec 05 00:56
1965	034933	GTX_281	15	12/Nov/05	299.69	209.59	16/Nov/05 04:22
1966	984311	PD0045	5	12/Nov/05	102.67	99.59	16/Nov/05 06:09
1967	984311	JDYE_6	10	10/Nov/05	251.99	222.37	30/Nov/05 05:55
1968	343371	TL121_281	1	02/Dec/05	2040.72	2040.72	04/Dec/05 00:56
1969	953534	GTX_281	1	03/Dec/05	21.59	NULL	NULL

Figure A2: TBL_Customers

CustomerID	Member	CustomerName	Address1	Address2
343371	Y	Andrews	123 Abel Ave	London
034933	Y	Ling Wing	6 Princes Street	Hong Kong
984311	N	Hutton	564 Holly Road	Manchester
953534	N	Rivers	80 Grange Way	Glasgow

Figure A3 shows flights that SWIFT has advertised on its web site.

Figure A3 TBL_Flights

OfferNo	AirlineID	FlightCode	Operating Cost	Seating Capacity	Flight Time	FlightDate
1004	B&G	BG_8971	20119.56	102	12:23	01/Dec/05
1005	B&G	GTX_281	19012.59	261	16:34	14/Nov/05
1006	JH Price	TL121_2	2033.59	100	00:56	17/Nov/05
1007	B&G	GTX_281	19012.59	260	04:22	16/Nov/05
1006	B&G	PD0045	21201.59	261	06:09	17/Nov/05
1008	JH Price	JDYE_6	22008.37	90	05:55	30/Nov/05
1009	B&G	JDYE_6	22008.37	100	11:23	30/Nov/05

BUSINESS RULES

BR1: When seats for the flights that depart on the same day to the same destination are sold out then the offer period terminates immediately.
BR2: Seats for all flights that are subject to the offer can only be purchased from 6 months prior to departure of the flight and up to 5 hours prior to departure.
BR3: Only customers who have completed an on-line registration form can purchase seats offered for sale from the SWIFT web site.
BR4: Most flights also carry cargo such as goods for export and the carrying of this cargo may reduce the seats that are offered for sale and the baggage allowance of passengers.