BCS THE CHARTERED INSTITUTE FOR IT
BCS HIGHER EDUCATION QUALIFICATIONS
Level 6 Professional Graduate Diploma in IT

September 2013

Examiners Report

Advanced Database Management Systems

SECTION A

Overall comments

The quality of most exam scripts was pleasing. Q1 and Q2 were universally popular and it was clear most candidates knew a great deal about the technical topics under discussion – leading to some very strong marks. A small minority went completely the other way and wrote either very little or essentially just rambled and padded out a technical question with vague or irrelevant responses

General Comments (Q1) :

Nearly every candidate attempted this question and 72% of those achieved pass level marks.

Answer pointers part a :

Two-phase locking should mention the concepts of a transaction having two distinct stages: a ‘growing’ phase where it acquires all locks necessary for it to complete its tasks (and cannot release any locks) and a ‘shrinking’ phase in which it systematically releases those locks and returns the data resources as it runs down (and is not allowed to acquire any new locks). In other words, 2PL ensures that a transaction must acquire a lock on a data item before doing any work on that data and once a transaction has finished with a lock and cannot grab more locks. Serializability is the idea that parallel transactions can execute concurrently - via interleaving (using a non-serial schedule) - yet without interfering with one another – so as give a final outcome on the database the same as if those transactions had been executed in a sequential (serial) manner. 2PL thus stops two competing transactions from colliding over the same data item(s) and thus violating the ACID principles.

Answer pointers part b :

This covers problems such as the lost update problem, dirty (uncommitted) read problem, non-repeatable reads, phantom reads etc. For each, a description of the problem and a simple time-lapse example involving two transactions T1 and T2 is needed for full marks. Referral to the different types of schedule – serial and interleaved – and the impact on these problems would get bonus marks.

Answer pointers part c :

Schedule – a sequence of operations drawn from two or more concurrent transactions that preserves the order of the operations in each of the individual transactions. A schedule can be serial or interleaved.
Cascaded rollback - when transaction T1 fails and induces a rollback which in turn causes other transactions - which were dependent on the success of T1 – to likewise fail and be rolled back.
Optimistic locking – based on the assumption that inter-transaction conflict is rare so individual transactions are allowed to proceed unsynchronized and are only checked for conflicts at the end – just before commit. Useful in low data contention environments because it avoids the

overhead of locking/waiting for unlocking but inefficient if data conflicts are common as transactions will have to be repeatedly restarted.
Pessimistic locking – assumes a high degree of inter-transaction conflict and locks up all data resources ahead of access immediately – even if other transactions never try and access them. Saves re-running transactions in highly contentious environments but this ‘belt and braces’ approach can induce overhead in locking/releasing data resources that were never in conflict in the first place.
Checkpoint – A point of synchronization between the database and the transaction log file (journal) that allows easy identification of which transactions need to be re-processed (redo/undo) in the event of database recovery being needed.

Examiner Comments:

Q1 was extremely popular – with just about all candidates opting to answer it (155 answers). Part (a) was generally well answered with most candidates clearly understanding the main concepts such as the growing/shrinking phases of 2PL, the different types of schedule and the various types of locks. Most provided good diagrams. Consequently few did badly. Part (b) was again mostly well answered with most candidates listing and then discussing the issues raised in the marking scheme – lost updates, dirty reads etc. Part (c) was, surprisingly, the weakest aspect of Q1, with many candidates giving quite vague and unfocussed responses. Luckily for them it carried only 5 marks. So in summary, Q1 was a good question for most of the candidates who attempted it.

Question 2

General comments :

76% of candidates attempted this question and 62% of those achieved pass level marks.

Answer pointers part a :

Major categories are: reporting tools (end-user desktop), query tools (SQL development and QBE interfaces), application development tools (for more advanced and regular tasks), EIS tools (decision-support), OLAP tools (multi-dimensional queries) and data mining tools (identification of new, unpredicted patterns and trends).

Answer pointers part b :

Summary management – DW queries are often seeking aggregated data, not the fine detail of individual rows, particularly aggregation via specific dimensions (month, product, region etc.) so the DBMS must support pre-computed summaries and aggregates to avoid run-time computation. *Analytical functions* – many BI and DW applications want to use SQL ranking and aggregate functions, cumulative aggregates or maybe the CUBE and ROLLUP operators for OLAP analysis. *Bitmap indexes* – via compression, the support for low cardinality data and more ‘open’ type of queries rather than the usual B-tree indexes used to search for individual identifiers. *Advanced join methods* such as in-memory hash joins and partition-wise joins – very useful for the large data volumes involved in a DW. *Advanced optimizers* that can use statistics and histograms - especially on skewed data distributions – such as Oracle’s cost-based optimizer when working on star schema queries. *Function-based indexes* that can pre-index complex calculations and other expressions often used in DW applications. Sampling functions that can gather mathematical data like averages without having to read every row.

Answer pointers part c :

This will be marked holistically with bonus marks for clear diagrams of the various models but the key points to be covered are: *dimension modeling* as a specialized example of an ER Model – both based on entities and relationships, the use of a *fact table* (with composite primary key)

and a set of *dimension tables* (each with an atomic primary key) related to the fact table via foreign keys – thus producing a *star schema* (star join) model.
The better candidates should then go on to discuss issues such as the fact table is much larger than the dimension tables, that the fact table works best when the 'facts' recorded are numeric (grades, prices, ages etc.) thus allowing aggregated computations to be run leading to summarized data, that dimension tables tend to hold more descriptive data (names, addresses, identifiers etc.), the use of de-normalized data to replicate attributes across multiple dimension tables (for example, storing address or contact data in several different dimension tables) thus avoiding additional joins and enhancing query performance.

Finally, a few words on what a *snowflake schema* is (where dimensions can have their own dimensions) – caused by normalizing the original dimension table data down into two or more child dimension tables, all linked to the 'parent' dimension table via the familiar PK/FK technique. So *star schemas* use de-normalized (repeated) data and *snowflake schemas* use normalized (minimized duplication) data.

Examiner Comments:

Again, this was another extremely popular question (121 answers). Part (a) generated some good (and relatively easy) marks with just about all answers covering OLAP and data mining but with the better candidates discussing DSS and the more technical issues such as SQL. It was a good start to the question. Part (b) was generally answered poorly with much general and vague comment rather than the specific points raised in the marking scheme. Few got high scores here but then again, if the comments were sensible and intelligent, even if generic, I did reward with a few marks. Part (c) was very well answered with the vast bulk of candidates giving detailed and extensive accounts of star and snowflake schemas and the associated issues highlighted in the marking scheme. This was a strong finish to Q2.

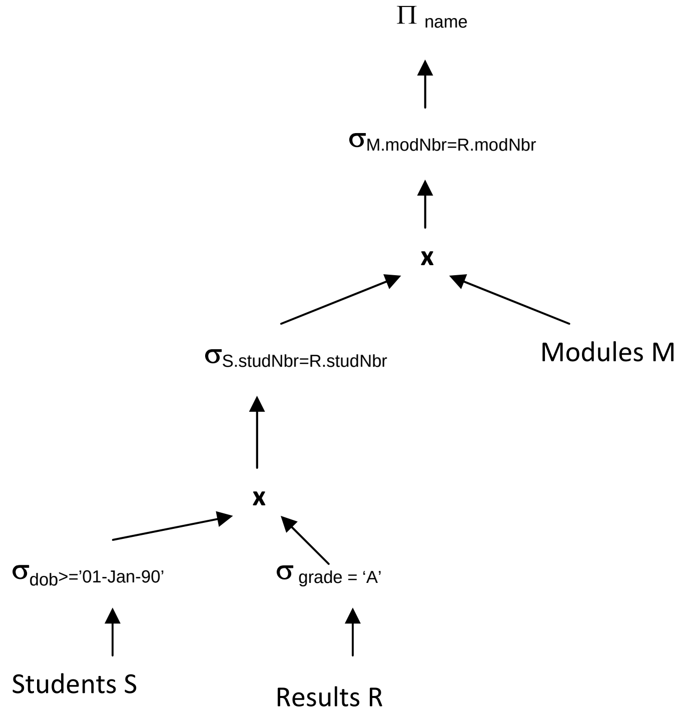
Question 3

General comments :

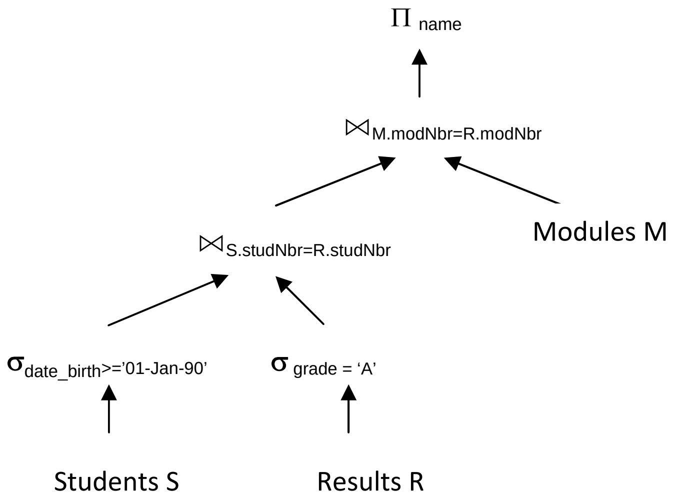
45% of candidates attempted this question and 37% of those achieved pass level marks.

Answer Pointers Part A:

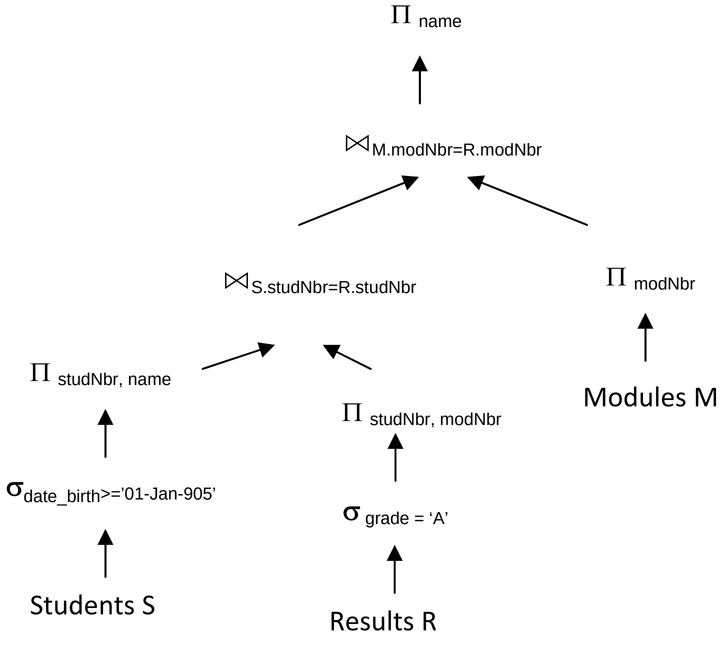
Step1: Move the Selection operations down the query tree (4 marks):



Step 2: combine the Cartesian Product with a subsequent Selection operation into a Join operation (4 marks):



Step 3: Move the Projection operations down the query tree (5 marks):



Examiner's Comments:

Very few candidates were able to produce query trees that show how selections and projections are applied to the corresponding tables.

Answer Pointers part b:

Authentication: A mechanism that determines whether a user is who he or she claims to be. It is usually implemented through a login & password mechanism. (2 marks)

Authorization: The granting of a right or privilege, which enables a user to legitimately have access to a database object. (2 marks)

Examiner's Comments:

Although most candidates managed to answer correctly, there are quite a few who mix-up the two concepts.

Answer Pointers part c :

(i) If the table is small enough to fit into the main memory then it is more efficient to read it and search it there.

(ii) A foreign key will probably appear in the join condition when joining tables, hence an index on it will speed up the join operation (for each row of the child table, we can use the index on the FK, to point directly to a matching row).

(iii) Every time a column gets updated, the corresponding index needs to reflect that update, hence creating a pressure on resources.

(iv) Indexes are sorted on the values of the column on which they are built. Hence, this provides an alternative to running a separate sorting operation after records have been retrieved.

Examiner’s Comments:

Candidates need to gain understanding of how indexes are built and why.

SECTION B

Question 4

General comments:

This was an extremely unpopular question with only 16% of candidates attempting it. Of those, 56% achieved pass level marks.

CODE SAMPLE

```
<%
response.ContentType = "text/xml"
set conn=Server.CreateObject("ADODB.Connection")
conn.provider="Microsoft.Jet.OLEDB.4.0;"
conn.open server.mappath("/db/database.mdb")

sql="select fname,lname from tblGuestBook"
set rs=Conn.Execute(sql)

response.write("<?xml version='1.0' encoding='ISO-8859-1'?>")
response.write("<guestbook>")
while (not rs.EOF)
response.write("<guest>")
response.write("<fname>" & rs("fname") & "</fname>")
response.write("<lname>" & rs("lname") & "</lname>")
response.write("</guest>")
rs.MoveNext()
wend

rs.close()
conn.close()
response.write("</guestbook>")
%>
<guestbook><guest><fname>Terje</fname><lname>Beck</lname></guest>
```

Answer pointers part a :

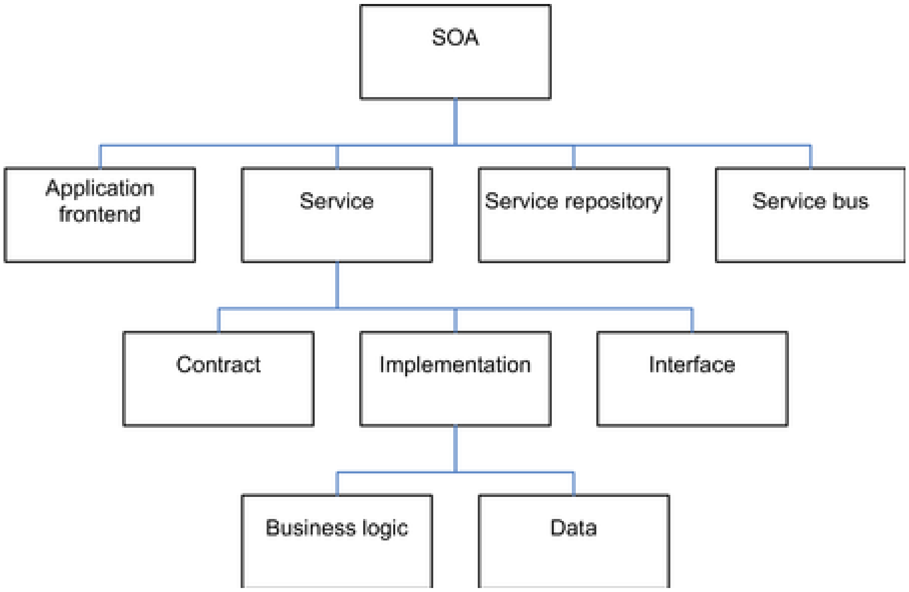
The reason why a SOA has emerged is because of the large and disparate information systems that prevail over the internet particularly in large organisations that have large and diverse IS even within the same functional area. SOA overcomes this problem to a certain extent by allow simultaneous use and easy mutual data exchange between programs of different vendors without additional programming or making changes to the services. These services are also reusable, resulting in lower development and maintenance costs and providing more value once the service is developed and tested. Having reusable services readily available also results in quicker time to market. SOA generally provides a way for consumers of services, such as web-based applications, to be aware of available SOA-based services. The best known are REST, JavaRMI and SOAP/XML web services.

Examiner's comments :

Most candidates clearly had no idea of the 'cloud' and concentrated on the older technology of SOAP and even web services tied to PHP and ASP. It was important to focus on the role of SOA in general as a mechanism for non-intelligent data transfer that otherwise would be difficult or impossible.

Answer pointers part b :

It is best to include a diagram that shows where a database sits in this architecture. eg



The Data layer is supported by a DBMS and it is clear that databases play a strategic role by the very nature of services that support modern distributed systems of large organisations. In effect ,the service provider/enabler decides what service implementation is required to be called to access a piece of data. Data will be passed usually in XML according to rules that match the database schema i.e. DTD or XSLT written in the implementation code. As can be seen the business logic which could include database constraints can be viewed as independent from the database.

Examiner's comments :

Again scant knowledge was in evidence and as a result, a range of attempts that related to web site development using old middleware/database connection technology such as CGI were submitted. Some of the best answers were explained most succinctly by a diagram,

Answer pointers part c :

The code is ASP but contains sufficient standard database middleware that is common to equivalent languages such as JDBC. For example code that defines the data provider (e.g. MSAccess) and its data file location; how the server is prepared at run time so that read access to database is defined in a string (SQL select statement); how the database provider is associated with the connection string; how the successive “response.writes” define the hierarchy of tags i.e. <title> <guestbook>.
The complete XML document is generated by an iterative process as the dataset returned from the database is accessed via a cursor that points at successive rows from which the instances of the row (i.e. guest, firstname, lastname are written to the XML using the encoding defined.
So, XML schema is decoded as

```
<guestbook>
  <guest>
    <fname>
    <lname>
  </guest>
</guestbook>
```

Returning for example tagged data values

```
<guest><”a123”>
<fname><”john”></fname>
<sname><”adams”></sname>
<fname><”bill”></fname>
<sname><”franks”></sname>
</guest>
```

as data

Examiner’s comments:

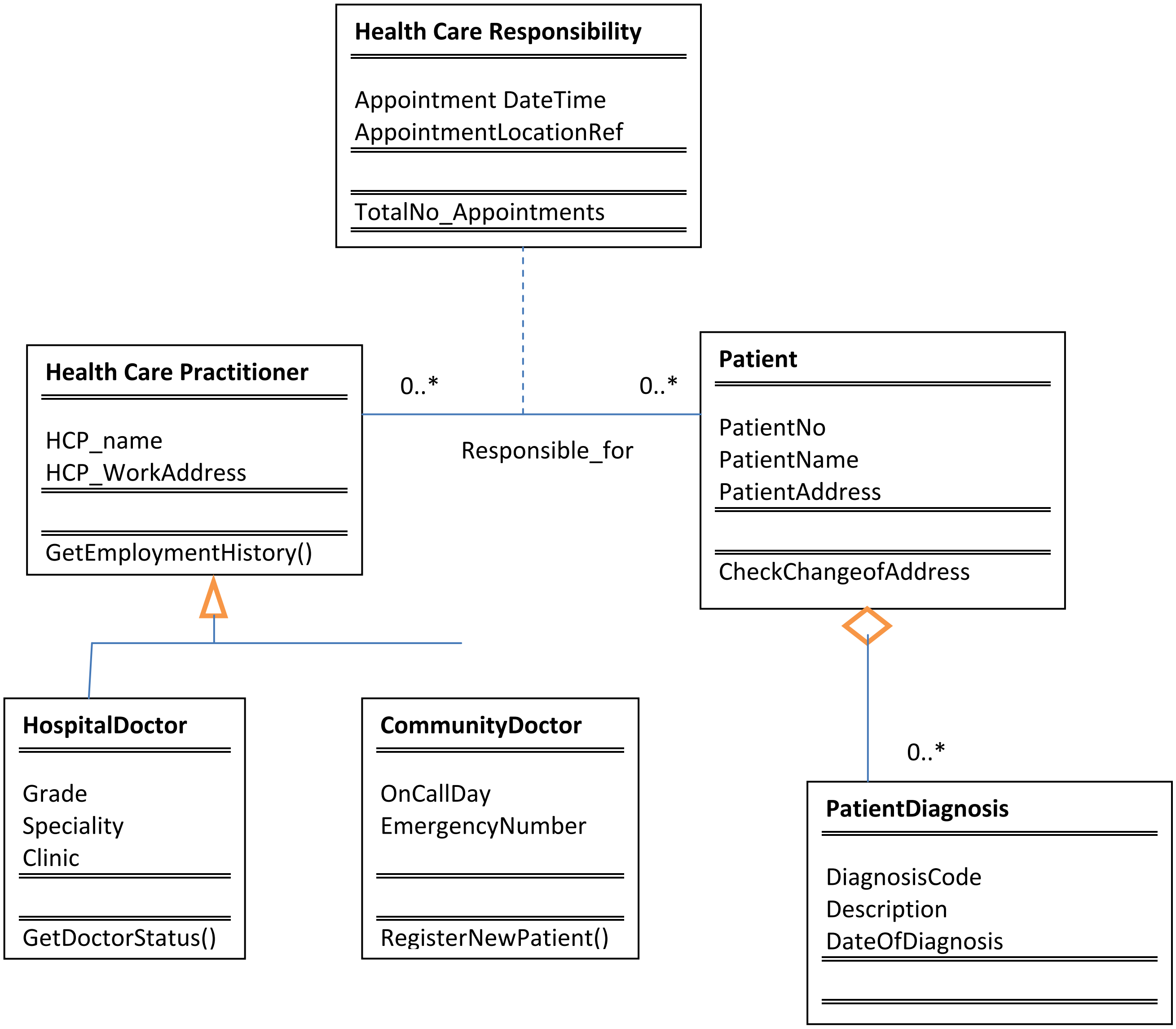
Most candidates used the sample code rather than their own but even so there was apparent a lack of confidence in coding. The main part of the XML generation was rarely backed up with an example of the XML DTD/schema instantiated with data. This could have been stated in the question but at this level it was expected candidates to produce this as part of the answer.

Question 5

General comments:

Overall this was a popular question with about 63% attempting it and 57% of those achieving pass level marks. The range of marks was acceptable and it was felt to be a fair question on a well covered topic on the syllabus.

Fig A1 UML Class Diagram



Answer pointers part a :

The 3 relationships are i) instance level relationship, ii) generalisation and iii) aggregation. Between HCP and patient is an instance relationship expressed as a relationship HCR. HospitalDoctor and CommunityDoctor are specialisations of HCP which is the generalised Class.
patientDiagnosis is part of the Patient. It is not a dependent relationship as some patients may be registered but have no current illnesses.

Examiner's comments :

A reasonable attempt but many candidates mixed up generalisation with whole-part.

Answer pointers part b :

Composition is a 'stronger' from of Aggregation represented by blacked out diamond. Meaning where there is a wish to model a strong dependency in the whole-part relationship then this means the part Class cannot exist without the whole/owning Class.
Composition usually has a strong life cycle dependency between instances of the container class and instances of the contained class. If the container is destroyed, normally every instance that it contains is destroyed as well.

Examiner's comments :

Generally poor knowledge and awareness of different OO modelling techniques

Answer pointers part c :

The best way of modelling this is to add a composition relationship to the PatientDiagnosis Class so that the strong dependency between treatment and diagnosis is established, i.e. a treatment is dependent on the diagnosis as it could be specific instance recorded for each patient rather than generally dependent.

Examiner's comments :

A variety of solutions would have been better to formally ask for a statement of assumptions. Most candidates suggested the expected solution.

Answer pointers part d :

HCP same as Class PK
ResponsibleFor(PatientNo,HCPName PK + all the attributes of HCR…)
Patient - same as Class with PK PatientNo
FK patientNo and HCPname is ResponsibleFor to above tables
PatientDiagnosis(DiagnosisCodePK, PatientNoPK, dateDiagnosisPK patientNo and DiagnosisCode both FK
Diagnosis(DiagnosisCodePK Description
HospitalDoctor(HCPNamePK, attributes of Same Class
CommunityDoctor(HCPName , attributes of CommunityDoctor

Examiner's comments :

It was surprising to find a poor range of attempts as this part should be quite easy to gain marks. This part was to show the results it might have helped candidates if part d) was done first to facilitate the process and get candidates thinking through the derivation of tables from a pseudo UML Class model.

Answer pointers part e :

The Classes are more or less the same as Entity Types in an ER model hence these are mapped to Entity/Base Tables in a Relational Model.
PK are equivalent to unique identifiers that may not be shown but must be reinstated in the Relational model. Same applies to FK which are always absent.
Methods are ignored but these could be used to check the relational model via the normal means of functional analysis. Many to Many such as ReposnsibleFor are broken down to an imaginary linking relationship table which this model actually shows as a Class(HCR) to reveal attributes carried over to the HCR relationship.
Inheritance(Generalisation) provide a means of modelling 1:1 relationships. Such that FK are posted to the specialised table and then the specialised Class can be modelled accordingly.