**BCS THE CHARTERED INSTITUTE FOR IT**

**BCS HIGHER EDUCATION QUALIFICATIONS**

**BCS Level 5 Diploma in IT**

**April 2013**

**Database Systems**

## SECTION A, Question 1

### General Comments (Q1)

This question was the most popular, having been attempted by nearly 90% of the candidates, Around 60% achieved pass level marks.

(a) A dentist uses forms, such the ones shown below, to keep a record of treatments given to patients. Each patient has a number, name, and a category (for example, adult, child, pregnant…). A patient can received many treatments on the same day, but we will assume that the same treatment is not administered twice on the same day. For example, a patient cannot have two extractions on the same day.

| **Dentist Appointment Form** | | |
|---|---|---|
| **Patient Number**: P102 **Patient Name**: John Smith<br>**Patient Category Number**: 1<br>**Patient Category Description**: Adult | | |
| **Appointment Date** | **Treatment ID** | **Treatment Description** |
| 13-Aug-2011 | T05 | Root canal |
| 13-Aug-2011 | T03 | Extraction |
| 22-Nov-2011 | T03 | Extraction |

| **Dentist Appointment Form** |
|---|
| **Patient Number**: P104 **Patient Name**: Ashok Kumar<br>**Patient Category Number**: 2<br>**Patient Category Description:** Child |

| Appointment Date | Treatment ID | Treatment Description |
|---|---|---|
| 13-Aug-2011 | T05 | Root canal |
| 27-Oct-2011 | T03 | Extraction |

(i) Identify any repeating groups in the above representation and describe how they can be removed to produce a table in first normal form (1NF). Show the obtained 1NF table filled with the data shown above. Identify the key attributes of this relation.

**(6 marks)**

**Answer Pointers and Marking Scheme:** (3 marks for repeating group and keys + 3 marks for tables)

Repeating group of attributes represents the attributes which have multiple values in the form. These are: Appointment Date, Treatment ID, Treatment Description.

An obvious choice of a key attribute is PatNbr given that the data in the form is related to patients.

To produce 1NF table(s), the repeating group of attributes needs to be removed along with a copy of the key attribute. Therefore, we obtain the following two tables which are in 1NF:

Patient (PatNbr, PatName, PatCatgNbr, PatCatgDesc)

Treatments (PatNbr*, AppDate, TreatID, TreatDesc)

| PatNbr | PatName | PatCatgNbr | PatCatgDesc |
|---|---|---|---|
| P102 | John Smith | 1 | Adult |
| P104 | Ashok Kumar | 2 | Child |

| PatNbr | AppDate | TreatID | TreatDesc |
|---|---|---|---|
| P102 | 13-Aug-2011 | T05 | Root Canal |
| P102 | 13-Aug-2011 | T03 | Extraction |
| P102 | 22-Nov-2011 | T03 | Extraction |
| P104 | 13-Aug-2011 | T05 | Root Canal |
| P104 | 27-Oct-2011 | T03 | Extraction |

**Examiner's Comments:** Most students managed to identify the repeating group of attributes correctly and to normalise the table accordingly.

(ii) If left un-normalised, the table obtained in (i) could lead to some inconsistencies (anomalies). Give an example of inconsistencies that may occur in this table as a result of:

- Update operation
- Delete operation

**(4 marks)**

**Answer Pointers and Marking Scheme:** (2 marks each)

Update anomaly: If, for example, we need to change the description of treatment T05, this change needs to be applied to all instances of that description. As a result, if the description is mistyped in one of the records, then it will leave the table in an inconsistent state: we will have T05 with two or more descriptions instead of one.

Delete anomaly: if we delete the record of a patient (e.g., as a result of them leaving the surgery), we could delete (and lose) the details of certain treatments if they were exclusively given to that patient.

**Examiner's Comments:** Most students struggled with this question. It is important that students familiarise themselves with the reasons why normalisation is undertaken and the issues it addresses.

(ii) Normalise the relation obtained in (i), showing functional dependencies and how you progress from 1NF through 2NF to a set of 3NF relations.
At each stage show the primary key and any foreign keys of each relation and state assumptions that you make about any of the relationships between the columns of data.

**(10 marks)**

**Answer Pointers and Marking Scheme:** (5 marks for each of 2NF and 3NF)

1NF tables:

Patient (PatNbr, PatName, PatCatgNbr, PatCatgDesc)

Treatments (PatNbr*, AppDate, TreatID, TreatDesc)

2NF: Partial dependencies: TreatId -> TreatDesc

Patient (PatNbr, PatName, PatCatgNbr, PatCatgDesc)

Treatment (TreatD, TreatDesc)

Appointment (PatNbr*, AppDate, TreatID*)

3NF: Transitive dependencies: PatCatgNbr -> PatCatgDesc

Patient (PatNbr, PatName, PatCatgNbr*)

PatientCategory (PatCatgNbr, PatCatgDesc)

Treatment (TreatD, TreatDesc)

Appointment (PatNbr*, AppDate*, TreatID*)

**Examiner's Comments:** Most students have had a reasonable attempt at this question. But many struggled to work out where to place the foreign keys as well as the composite primary key.

(b) Describe the approach taken to the handling of data in the early file-based systems. Discuss the disadvantages of this approach.

**(5 marks)**

**Answer Pointers and Marking Scheme:** (2 marks for description + 3 marks for disadvantages)

A file-based system is a collection of application programs that perform services for the end-users, such as the production of reports. Each program defines and manages its own data. This is usually associated with a sector of activity or department within an organization (e.g., Human Resources, Sales, Marketing…).

Disadvantages include:

- Separation and isolation of data (difficult to access data that should be available)
- Duplication of data (because of the decentralized approach taken by each department).
- Data dependence (physical structure of data files defined in the application code).
- Incompatible file formats (dependent on the application programming language)
- Fixed and limited number of queries available (application developers need to write them)

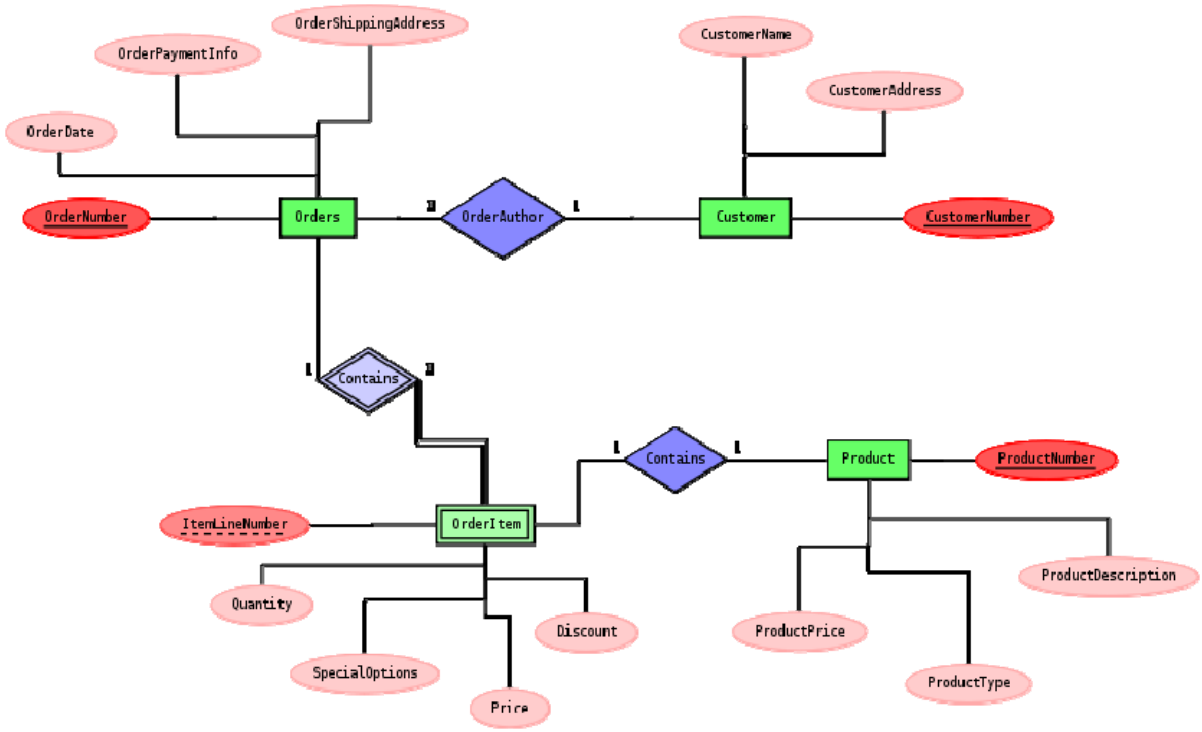
**Examiner's Comments:** Most students confused file-based systems with manual systems based on paper files. Students should refer to database textbooks where a discussion of file-based systems is usually offered in the introduction chapter.

**SECTION A, Question 2**

**General Comments (Q2)**

This question was very popular. It covered the topic of data modelling on the syllabus. Performance was generally good with a wide range of marks achieved.

Refer to the following ER model which represents an order processing application.




a) Using the ER model above give an example of EACH of the following modelling concepts

i) Entity Type 1 mark
ii) Entity occurence 1 mark
iii) Key Attribute 1 mark
iv) Relationship Type 1 mark
v) Cardinality 1 mark

b) Using a defined ER modelling notation, explain how you would extend the ER model to represent :-
   i) An optional participation constraint in a relationship.
   ii) A mandatory participation constraint in a relationship.

6 marks

c) Discuss the validity of the restrictions imposed by the relationship between Orders and Products if this model was applied in a practical order processing application. Provide example data to assist in your answer.

3 marks

d) Using examples derived from the ER model explain the difference between :-
   i) Strong and Weak Entity Types
   ii) Binary and Ternary Relationship

6 marks

e) Why is it good practice to resolve M:N relationships in an ER model? Give an example from the ER model above where a M:N relationship has been resolved.

5 marks

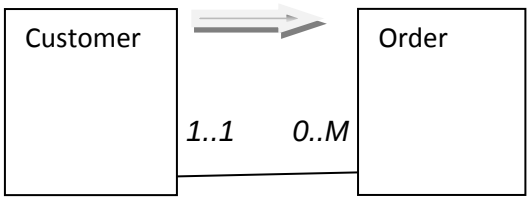
**ANSWER POINTERS part a :** shown next to the question sub part (i-v)

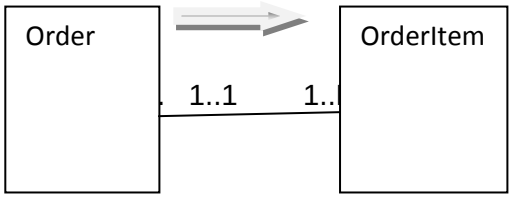
i) Entity Type 1 mark **ANS Orders**
ii) Entity occurence 1 mark **ANS 345, JCSmith, Whitehall London WC1,**
iii) Key Attribute 1 mark **ANS OrderNumber**
iv) Relationship Type 1 mark **ANS OrderAuthor**
v) Cardinality 1 mark ANS **OrderLine must belong to ONE Order**

**Examiner's Comments part a :**
Very few candidates succeeded in getting part (ii) correct.

**Answer Pointer part b :**

UML notation




The extension is made to the degree of the relationship defining whether say an order MUST (obligatory) have a Customer, where as a Customer may not (NON Obligatory) make Orders

**Examiner's Comments :**
Most candidates choose a Yourdon notation and understood the simple extension needed to express this constraint. No problems apart from some candidates using a different ER model for their answer than the one supplied … only one or two few marks were lost as a consequence. The question implies … the ER model supplied.

**Answer Pointer part c :**
Strong entity Types always have a key attribute (eg orders) where as weak entity types do not have key attributes of their own and are dependent on a strong entity type eg (orderline). Binary relationships are evident in the ER model eg Customers and Orders and involve 2 entity types, where as ternary involve 3 entity types an example might be Orders having a relationship with products, supplier and customers.

**Examiner's Comments :**
It was surprising how many candidates did not know the difference between binary and ternary relationships or could construct meaningful examples using the ER model supplied.

**Answer Pointer part d :**
It means that an orderline (eg give an example of a onlineshopping cart) can only contain one item per order. This could be valid in some very special circumstances but normally the restrictions are invalid for realistic order processing systems.

**Examiner's Comments :**
Surprising to find that very few candidates attempted this part.

**Answer Pointer part e :**
At the logical level Many to many relationships need to be 'resolved' before mapping to relations. Only 1:M and 1:1 relationships can be mapped from an ER model to relations. An example would be between Orders and Products where an OrderItem Entity Type is created with 2 x 1:M relationships created as a result. The transformation is quite common and in this instance provides more meaning to the ER model.

**Examiner's Comments :**
A good number of candidates answered this part correctly.

**SECTION A, Question 3**

**General Comments (Q3)**
Generally candidates performed well in answering parts a),b) and c). Very few candidates produced a satisfactory answer to part d).

The following tables define the relationship between STREETS and JUNCTIONS that intersect STREETS

```
STREETS
Street name
---------------------
S1     Grange Road
S2     High Rise
S3     Morton Ave
S4     Muston Lane
S5     Borrowby Court
S6     Fell Road
```

JUNCTIONS

```
loc1 loc2
---- ----
S1   S3
S1   S4
S1   S5
S2   S3
S2   S5
```

a) Show with the aid of a sketch of the intersection of streets and junctions.

**5 marks**

b) Show your working, determine what output is produced by each of the following SQL queries.

**6 marks**

c) Explain the overall function of each query.

**6 marks**

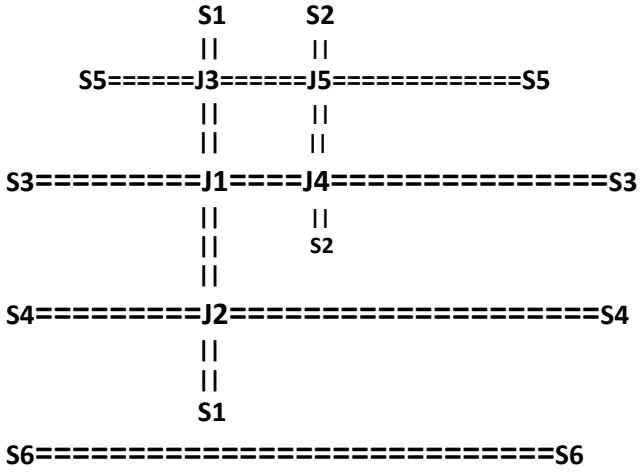
```
Query1:
SELECT DISTINCT Name
FROM Streets x, Junctions y
WHERE x.street = y.loc2

Query2:
SELECT Name FROM Streets x
WHERE x.street NOT IN
 (SELECT y.loc2 FROM Junctions y)
```

d) Using the above sample data, give an example of an SQL query that would utilise the GROUP BY and HAVING operators.

**8 marks**

**Answer Pointer part a :**



**Examiner's Comments :**
There were a small number of candidates who did not understand this part producing a wide variety of answers. Otherwise this was a part question that candidates in the majority gained an easy 5 marks.

**Answer Pointer part b :**

Query 1

Result set

Name
---
Borrowby Court
Morton Ave
Muston Lane

List streets that end in a junction with any other street.

+ working is important in case the of error in output.

Query2:
which streets DO NOT have junctions with other streets
that end in a junction

Result set

Name
---
Grange Road
High Rise
Fell Road

**Answer Pointer part c :**

Query 1 Which streets have junctions with other streets

JOIN clause is used to match those streets that exist in the Junctions table. DISTINCT eliminates duplicates

Query 2: Which streets do not have junctions with other streets

A subquery is used with the inner part selecting only those streets matching junctions that exist with streetids in the street table then the outer part looks for those non matching tuples with the streets table again. Understanding of sub query is an alternative way of expressing a JOIN

**Examiner's Comments part b and c :**
Candidates tended to merge the answer to these parts by combining their working out with an explanation of each query. This was accepted in this instance and both parts were marked accordingly. See the general comments at the beginning that relate to this part.

**Answer Pointer part d :**
An example query should reveal an aggregate function (COUNT) that can be used with GROUP BY followed by a condition (HAVING) count for each street how many junctions there are:-

```
SELECT COUNT(x.street) as NoJunctions , x.name
FROM Junctions y , Streets x
where y.loc1 = x.street
OR y.loc2 = x.street
GROUP BY x.name
HAVING COUNT(x.street) > 1
```

Result set

```
NoJunctions name
----------- ---------------
2           Borrowby Court
3           Grange Road
2           High Rise
2           Morton Ave
```

**Examiner's Comments :**
Many candidates did not understand that a GROUP BY involves grouping a resultset in order to compute some total using an aggregate function such as COUNT, SUM AVG etc. Some candidates confused ORDER BY and GROUP BY and many candidates clearly had not studied these SQL clauses and practised using them.

**SECTION B, Question 4**

**General Comments (Q4)**

This question was popular, having been attempted by nearly 70% of the candidates, Around 55% achieved pass level marks.

*(a)* *With reference to a sample relation of your own choosing,* explain and discuss the following relational model terminology, including its function and any related concepts as well as supplying the 'everyday' term for each formal term. A good illustrative diagram is strongly suggested.

- Tuple
- Attribute
- Domain
- Degree
- Cardinality
- Primary Key
- Foreign Key

Each item is equally weighted.

(15 Marks)

**Answer Pointers and Marking Scheme:** Three marks per item with full marks only for covering all the following points. Tuple means row. Each row is uniquely identified by a primary key. Attribute means column. Each column must have a unique name within that table (relation) and can be isolated across tables by qualifying the column name with the

table name (such as student.name). The domain specifies what are acceptable data values within a column (and by implication the acceptable operations on that column's data). The degree of a table is the width or number of columns and is part of the union compatibility criteria (the other one being that like-for-like columns in other results sets share comparable domains). The cardinality of a table is simply the depth or number of rows in the table. Good diagram gets bonus marks.

**Examiners' Comments:** Strong responses were found on: Tuples, Attributes, Primary and Foreign Keys – with text-book answers and all associated concepts clearly explained. The answers regarding Domains, Degree and Cardinality were not so much badly done but more unexpected responses. There is, as one student rightly pointed out, some ambiguity over the use of these terms and they are often (correctly) used in settings outside of the scope of this question – such as on ER diagrams. If the comments were sensible they were treated sympathetically and marks awarded, even if not precisely pertinent to the relational model. The only obvious area of confusion regarded 'Domain' with quite a few students confusing this with 'Attribute' but on the whole this was a well answered question.

(b) *Using your own simple examples*, explain how the UNION, INTERSECT and MINUS (DIFFERENCE) operations work and what limitations the concept of UNION COMPATIBILITY places upon the sets being processed. Suitable diagrams will gain bonus marks.

(10 Marks)

**Answer Pointers and Marking Scheme:** This question will be marked holistically with extra marks awarded for clear examples and diagrams but the key points are as follows. UNION as the combination of all sets, INTERSECT as the 'overlap' of sets and MINUS as what is in one set but not another. The role of duplicate removal from the final results set. Suitable notation and clear use of Venn diagrams to highlight the universal set and the above three operators. A clear definition of *union compatibility* - stating that in order to apply the set operators all component sets must have the same degree (number of columns) and that comparable columns in each set must be of the same data type (compare like with like). It should also be made clear that UNION (an operator to be applied) is not the same as UNION COMPATIBILITY (a condition to be satisfied).

**Examiners' Comments:** Again, some text-book answers to the three set operations showing deep understanding. Most students supplied clear Venn diagrams and/or sample relations. Good answers. The only weakness was on 'union compatibility' where, with a few exceptions, it was clear that this concept was not well understood.

## SECTION B Question 5

### General Comments (Q5)

This question was attempted by 46% of the candidates. Around 52% achieved pass level marks

(a) Describe the various interfaces, tools and techniques that a user (technical or otherwise) may employ when interacting with a database.

(10 Marks)

**Answer Pointers and Marking Scheme:** This is quite an open type of sub-question that sets form-based interfaces into context. The better students will include issues such as (non-technical) end-user interfaces like web-based (three-tier) interfaces, non-web (two-tier) GUIs such as forms & reports through to those interfaces designed for technical users like developers and DBAs such as forms/report generators and other software development environments – both graphical and command-line (for example Oracle's APEX and SQL*Plus respectively). It also includes third party interfaces for developers such as TOAD and specialized applications like Oracle's Enterprise Manager for DBAs.

**Examiners' Comments :** This was widely interpreted (well beyond the scope of the model answer) with answers covering a dizzying range of issues – some valid, others not. Marks awarded for sensible comments. Very few students gave the type of response expected but a very small number latched onto what was expected and gave detailed accounts of the issues highlighted in the marking scheme. Not so much badly answered but differently answered.

(b) Explain what the term *data validation* means. *Using your own examples*, describe the various data validation techniques that may be embedded into a forms-based interface to a database.

(10 Marks)

**Answer Pointers and Marking Scheme:** The definition should be along the lines of 'ensuring that only clean, correct, accurate, well-formatted data is accepted into the database'. It should be about data integrity, not confused with database security (although validating a user ID and password is fine here). Typical examples will include format masks (for example, dates of birth or telephone numbers have valid structure), range checks (for example, financial fields are > 0.00), membership validity (for example, a supplier name is a valid entry in the database), cross-field consistency checks (so if the user enters 'Mr' in one field, the Sex field must be 'M' etc.), presence checks (to ensure no missed fields) etc. Clear examples needed for best marks.

**Examiners' Comments :** Much better answers here – the students obviously grasped the core issues and gave strong responses. As expected, there was cross-over with part (c) and such answers were rewarded. A well answered question.

(c) Describe the *form components* that may be used to implement these data validation techniques.

(5 Marks)

**Answer Pointers and Marking Scheme:** Form components would be drop-down lists to ensure only pre-validated entries can be chosen, radio buttons to ensure only a single (valid) option is selected, double-entry of key fields like passwords to rule out mistyping, automatic totaling of numerical data, on-form calendars where users can click on a given date, labels at side of each field with an example, on-form help button, highlighting which fields are mandatory via an asterisk etc.

**Examiners' Comments :** Again, this was an almost universally well answered question with many students getting the full five marks.

**SECTION B Question 6**

**General Comments (Q6)**

This question was attempted by 62% of the candidates. Around 62% achieved pass level marks

(a) Explain what is meant by a transaction and why it is an important unit of operation in a DBMS?

**(4 marks)**

**Answer Pointers and Marking Scheme:**
**Transaction**: A logical unit of work that transforms the database from one consistent state to another. It is also the unit of concurrency and recovery control.

**Examiner's Comments:** Most students managed to answer this question.

(b) The consistency and reliability aspects of transactions are due to their "ACID" properties. Discuss each of these properties and how they relate to the concurrency control and recovery mechanisms in a database.

**(12 marks)**

**Answer Pointers and Marking Scheme:** (3 marks each)
*Atomicity* The 'all or nothing' property. It is the responsibility of the recovery subsystem of the DBMS to ensure atomicity.
*Consistency* A transaction must transform the database from one consistent state to another consistent state. It is the responsibility of both the DBMS and the application developers to ensure consistency.
*Isolation* Transactions execute independently of one another. In other words, the partial effects of incomplete transactions should not be visible to other transactions. It is the responsibility of the concurrency control subsystem to ensure isolation.
*Durability* The effects of a successfully completed (committed) transaction are permanently recorded in the database and must not be lost because of a subsequent failure. It is the responsibility of the recovery subsystem to ensure durability.

**Examiner's Comments:** Most students were able to describe these properties well. Some forgot the second part of the question which is to relate these properties to concurrency and recovery management.

(c) Describe, using an example, one type of problem that can occur in a multi-user environment when concurrent access to the database is allowed.

**(5 marks)**

**Answer Pointers and Marking Scheme:** (2 marks for description + 3 marks for example)
Describe, using an example, a problem such as Lost update problem, the uncommitted dependency problem, or the inconsistent analysis problem.

**Examiner's Comments:** Again, this was a straightforward question which most of the students attempted well.

(d) Suppose we have a table called employees that, initially, has 120 records. How many rows will be in the table after executing the following commands? **Justify** your answer. Hint: the answer is one of the following: 123, 124, 0, or 1.

```
INSERT INTO employees (employee_id) VALUES (140);
SAVEPOINT emp140;
INSERT INTO employees (employee_id) VALUES (141);
SAVEPOINT emp141;
INSERT INTO employees (employee_id) VALUES (142);
SAVEPOINT emp142;
TRUNCATE TABLE employees;
INSERT INTO employees (employee_id) VALUES (143);
ROLLBACK;
```

**(4 marks)**

**Answer Pointers and Marking Scheme:** (1 mark for correct number of records + 3 marks for justification)

There will be 0 records in the employees table. The TRUNCATE statement is DDL and performs an implicit commit. After the TRUNCATE statement, there are 0 rows in the table. The one row that was inserted was removed when the ROLLBACK statement was executed.

Allocate partial marks (2) to those who answered 123 saying that the rollback will go to the last save point. No marks are allocated if an answer (number of records) is given without explanation.

**Examiner's Comments:** Not many students seem to be familiar with the effect of the TRUNCATE statement. Also, many confuse a savepoint with a commit.