Technical Writing Examples

Plus A Few Tips

What is wrong?
How to rewrite?

Before

The volume of information has been rapidly increasing in the past few decades. While computer technology has played a significant role in encouraging the information growth, the latter has also had a great impact on the evolution of computer technology in processing data through the years. Historically, many different kinds of databases have been developed to handle information, including the early hierarchical and network models, the relational model, as well as the latest object-oriented and deductive databases. However, no matter how much these databases have improved, they still have their deficiencies. Much information is in textual format. This unstructured style of data, in contrast to the old structured record format data, cannot be managed properly by the traditional database models. Furthermore, since so much information is available, storage and indexing are not the only problems. We need to ensure that relevant information can be obtained upon querying the database.

After

Much information is textual. This unstructured data cannot be managed properly by traditional database models. Furthermore, storage and indexing are not the only problems. We need to ensure that relevant information can be obtained upon querying.

Before

As each value is passed to the server, the “heart” of the system, it is checked to see whether it is in the appropriate range.

After

Each value passed to the server is checked to see whether it is in the appropriate range.

Before

Tree structures can be utilized for dynamic storage of terms.

After

Terms can be stored in dynamic tree structures.

Before

The following theorem can now be proved.

After

We can now prove the following theorem.

Before

The authentication process and update activities are the responsibility of the database manager module.

After

Authentication and updating are the database manager’s responsibility.

Before

This language has more “power” than the functional form.

After

This language allows simpler expression of queries than does the functional form.

Before

In SIMD, the same instructions are applied simultaneously to multiple data sets, whereas in MIMD different data sets are processed with different instructions.

After

In SIMD, multiple data sets are processed simultaneously by the same instructions, whereas in MIMD multiple data sets are processed simultaneously by different instructions.

Before

A static model is appropriate because each item is written once and read often.

After

A static model is appropriate because each item is only written once but is read often.

Before

Type inference derives information about programs.

After

Type inference proves that programs obey the language typing rules.

Before

We are planning to consider possible options for extending our results.

After

We are considering how to extend our results.

Before

$p \leftarrow q_1 \wedge \cdots \wedge q_n$ is a conditional dependency.

After

The dependency $p \leftarrow q_1 \wedge \cdots \wedge q_n$ is conditional.

Before

For each x_i, $1 \leq i \leq n$, x_i is positive.

After

Each x_i, where $1 \leq i \leq n$, is positive.

Before

(Within pseudocode on strings …)

For $1 \leq i \leq |s|$

1. Set $c \leftarrow s[i]$
2. Set $A_c \leftarrow A_c + 1$

After

For each character *c* in string *s*, increment A_c.

Other tips

Beware of analogies

Writing a program is like building a model with connector blocks.

Open strong

This paper does not describe a general algorithm for transactions.

General-purpose transaction algorithms guarantee freedom from deadlock but can be inefficient. In this paper we describe a new transaction algorithm that is particularly efficient for a special case, the class of linear queries.

We describe a new transaction algorithm that is particularly efficient for linear queries. While general-purpose transaction algorithms guarantee freedom from deadlock, they can be inefficient. So, we focus on a common special case.

Be consistent

Style, terminology, variable names, ordering of list items, numbering 1-2-3 vs. a-b-c, fonts, indentation, …

Credits

- Most examples taken/adapted from “Writing for Computer Science”, by Justin Zobel (2nd ed.), 2004
- Some taken/adapted from a document by the CMU SCS Research Documents Group, 1989.