## INTRODUCTION

Data base techniques have been applied to specific chemical structure data in the past, mainly for use in generating graphics displays [1, 2, 3, 18]. However, these data bases have been at best conventional data bases. There appears to be no reports in the literature of the use of relational data bases [12, 13] with SQL for molecular information systems.

With a relational data base, the conceptual files are constrained to be relations, so that set theory and predicate calculus [4, 13] can be applied to them. As a result, there are major advantages to using a relational data base for holding molecular structure data. The outstanding advantage is that even with complex searches of the data base it is usually unneccessary to write a program in a procedural language, as is the case with conventional data bases and file collections. It is necessary only to specify what kind of data is to be retrieved using a non procedural [4, 13] relational data base manipulation language, of which the most common is SQL.

In this paper we develop a proposal for a relational data base for molecular information systems. The goal was a relational data base structure that could handle the structure of every conceivable molecule. We have called the resulting data base structure a two-path data base structure. Two-path relational data bases for molecular data can be used with SQL, the standard relational language for such common relational data base systems as DATABASE2

[9], INGRES [15] and ORACLE [10], in the following ways:

(a) For retrieving complete structure data for arbitrary chemical entities, down to the level of atoms.

(b) For use with graphics systems for displaying the structure of any arbitrary chemical entity.

(c) For retrieving information about the quantity or type of substructures (for example benzene or pyrimidine rings) or atoms or bonds in a chemical entity.

(d) For retrieving entities, and possibly also their structures, that contain specific quantities or types of substructure, atom or bond.

Two-path data bases provide two pathways to the structure of an entity in cases where the entity is derived from one or more substructures. The two pathways are necessary, partly for reasons of disk space conservation, but mainly because of a need for conformance with atomic occurrence number standards of the International Union of Pure and Applied Chemistry (IUPAC) [14].

## BASIC TUPLE-PER-BOND APPROACH

The basic approach to the problem of relational molecular structure data bases involves the use of a tuple (or record) to describe each

bond in a molecule. In the simplest case the bond is between two atoms, and not between two substructures.

In Figure 1 there is an example of a naive data base where each bond between atoms is described by a tuple. The data base contains information about the compounds propene and methoxyethane.

| IUPACNAME | CODE | MP | BP |
|---|---|---|---|
| PROPENE | PRN | -185 | -048 |
| METHOXYETHANE | MXH | -139 | -023 |

**CHEMICAL-ENTITY**

| CODE | EL1# | EL1 | ER1# | ER1 | BOND |
|---|---|---|---|---|---|
| PRN | 1 | C | 2 | C | 2 |
| PRN | 1 | C | 1 | H | 1 |
| PRN | 1 | C | 2 | H | 1 |
| PRN | 2 | C | 3 | H | 1 |
| PRN | 2 | C | 3 | C | 1 |
| PRN | 3 | C | 4 | H | 1 |
| PRN | 3 | C | 5 | H | 1 |
| PRN | 3 | C | 6 | H | 1 |

| | | | | | |
|---|---|---|---|---|---|
| MXH | 1 | C | 1 | O | 1 |
| MXH | 2 | C | 1 | O | 1 |
| MXH | 1 | C | 1 | H | 1 |
| MXH | 1 | C | 2 | H | 1 |
| MXH | 1 | C | 3 | H | 1 |
| MXH | 2 | C | 4 | H | 1 |
| MXH | 2 | C | 5 | H | 1 |
| MXH | 2 | C | 6 | H | 1 |

**CHEMICAL-BOND**

Figure 1

The relation CHEMICAL-ENTITY has a tuple for each chemical entity, and can have many fields giving information about the physical and chemical properties of complete chemical entity. We show only two, namely boiling and melting points, as examples.

In contrast, the relation CHEMICAL-BOND has a tuple for each bond in a chemical entity. There is thus a one-to-many (1:n) relationship [4] between CHEMICAL-ENTITY and CHEMICAL-BOND ; since a chemical entity will have many bonds, but a specific bond occurs only in one entity.

The fields EL1 and ER1 give the entities (atoms in this case) at each end of a bond (left end (L), right end (R), where right and left serve only to distinguish ends of a bond, and not the geometry of a molecule). The field BOND has the arbitrary value 1 for a single bond, 2 for a double bond, 3 for a triple bond, and

1.5 for a resonance bond, and 0.1 for a hydrogen bond. (As we shall see, there is also a need for zero and negative BOND values, to denote imaginary and broken bonds.) To denote an atom, the standard chemical symbols are used (C for carbon, H for hydrogen, and so on)

EL1# and ER1# give the occurrence numbers of atoms of the same type. The first occurrence of a carbon atom, for example, has occurrence number 1, the next occurrence number 2, and so on. The occurrence number together with the type of atom (EL1 or ER1 value) uniquely identifies an atom. For example, the double bond in propene has the structure

1C = 2C

As a result, the structure of propene and methoxyethane are as shown in Figure 2, using the data from the data base in Figure 1.

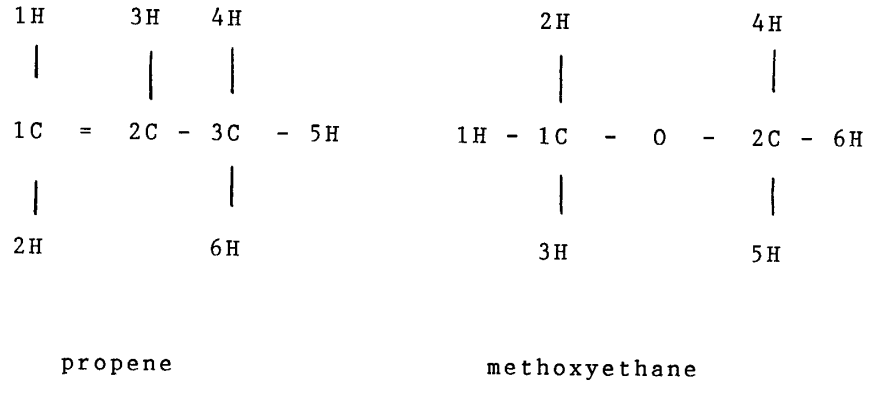


Figure 2

The occurrence numbers to the left of the atomic symbols in Figure

2 would likely be omitted in any display generated by a graphics system. The occurrence numbers are crucial for the storage of the structure of any molecule, no matter how large or complex. They are also used by IUPAC in chemical nomenclature. The two-path approach, which we present shortly, is to a considerable extent due to a need for synchronization of occurrence numbers with both data base and chemical methodology.

**Limitations of the simple tuple-per-bond approach**

The simple tuple-per-bond approach above could, in theory, hold the structure of every conceivable molecule, but is limited in two important ways. First, the same chemical group, for example a benzene or pyrimidine ring, occurs in many compounds, possibly many times within the same compound. In the data base in Figure 1 the atomic structure for benzene would have to be replicated for every molecule containing a benzene ring derivitive. Thus repetition of the atomic structure of such groups, for every compound in which they occur, is an unacceptable waste of storage space. It has to be remembered that some 2 million compounds have been described in the literature.

The other limitation is that we cannot use SQL to express queries involving these common subgroups, such as benzene rings. Thus we could not pose queries such as:

What compounds have two distinct benzene rings.

What compounds have exactly have one purine and

one pyrimidine group?

and so on. Nevertheless, with the data base in Figure 1, we can apply SQL to the molecular structure at the atomic level, even if we cannot deal with chemical groupings such as benzene rings. For example, the query:

How many carbon atoms are there in methoxyethane?

can be expressed in SQL as:

```
SELECT MAX (EL1#) FROM CHEMICAL-BOND
WHERE CODE IN (SELECT CODE FROM CHEMICAL
                WHERE IUPACNAME = 'METHOXYETHANE');
```

As an another example:

List the unsaturated (have one or more double carbon-carbon bonds) compounds.

```
SELECT IUPACNAME FROM CHEMICAL
WHERE CODE IN (SELECT CODE FROM CHEMICAL-BOND
                        WHERE EL1 = 'C' AND ER1 = 'C'
                          AND BOND = 2);
```

The data base in Figure 1 is the probably the best we can do using a single-level approach. If we want to be able to deal with sub-

structures, however, we need a multiple-level recursive approach.

**Multiple-level relational data bases**

Recursive bill-of-materials data base structures [4, 5 ] are commonly used for data bases describing objects made up of substructures, which in turn are made up of subsubstructures, and so on. Such data bases are commonly used in the assembly and maintenance of complex objects in manufacturing plants. They are thus multiple-level data bases. They are recursive or cyclic data bases because a relation in the data base participates in a many-to-many relationship with itself [4, 5], thus giving rise to multiple levels corresponding to the levels of substructure, subsubstructure, and so on, in the entities described by the data base. In theory there is no limit to the number of levels of nesting of substructures that can be employed in a bill-of-materials data base.

In order to eliminate redundancy due to the same basic chemical groupings appearing in many compounds, and to allow SQL expressions to reference basic structures (methyl groups, benzene rings, purines, amino acid residues, and so on) a recursive data base structure is needed for molecular structure data. However, we cannot use the standard bill-of-materials data base structure. The problem is that in addition to subgroupings within groupings chemical bonding between the atoms of the groupings and subgroupings must be specified at all levels. It is the need for provision for this bonding at all levels that gives rise to a rather special data base structure for molecular structure data.

At first sight it may appear that an obvious modification of the bill-of-materials data base structure will serve, where each atom in a bond is always uniquely specified in terms the structures it is within. For example, in an expanded tuple of CHEMICAL-BOND in Figure 1 to handle molecules with three levels of structure, instead of 3 C we might use 2 XYZ 5 PQR 3 C, meaning carbon occurrence 3 within PQR substructure occurrence 3 within XYZ substructure occurrence 2. However, this modification will not work, because of space wastage and IUPAC carbon atom occurrence number problems. The only total solution appears to be the two path structure.

## THE TWO-PATH SOLUTION TO THE MULTIPLE-LEVEL DATABASE PROBLEM

There are major problems with the use of conventional multiple-level recursive (bill-of-materials) data bases and modifications thereof for molecular data. If we permit levels greater than 2 we have the disadvantages:

(a) The number of fields in the CHEMICAL-BOND relation increases in proportion to the number of levels needed for the most complex compound, so that a great deal of space is wasted in the CHEMICAL-BOND relation. The number of fields increases because each tuple must describe a chemical bond, and therefore must identify two atoms at either end of the bond. Thus if we need to refer to atom 2 XYZ 3 PQR 3 C, as mentioned previously, we need to add 2 x 4 or eight additional fields to CHEMICAL-

BOND in Figure 1. But with large numbers of simpler molecules the additional fields would have null values and so a very great deal of disk space would be wasted.

(b) The occurrence numbers used with carbon atoms, in particular, will in many cases not match standard IUPAC chemical nomenclature. This should be obvious. For example, suppose we use the numbering system 1-6 for the carbon atoms of a benzene ring, and store the structure of benzene in CHEMICAL-BOND. Then, for example, if we attempt to store 2,3,6,7-tetrachloro-dibenzo-p-dioxin (the toxin dioxin) as being based on two benzene rings, the correct IUPAC carbon atom occurrence numbers cannot be extracted from the data base.

and the advantage:

(a) Compounds can always be referenced, in searches of the data base, in terms of common substructures, no matter what level of recursion is involved.

If we restrict the data base to two levels, using the CHEMICAL-BOND relation only for structure data, we have the disadvantage:

(a) We cannot reference many chemical compounds, in searches of the data base, in terms of common substructure in cases where the substructure is more than one level down from the original compound.

and the advantages:

(a) The number of fields in the CHEMICAL-BOND relation does not depend on the complexity of the molecules, and there is little waste of space.

(b) The numbering (occurrence number) system used for carbon atoms can match IUPAC nomenclature.

The problem is to uncover the structure of a data base that has all of the advantages of the two approaches above and none of the disadvantages. The two-path structure solves this problem, since for many compounds two distinct pathways can be followed in extracting information about the structure of any chemical entity. However, before presenting the two-path approach in detail, the concept of bond cleavage tuples has to be introduced, since it is crucial to the two-path solution.

**Use of bond-cleavage tuples**

It is clearly useful, from both an information retrieval and storage-space utilization viewpoint, to use common groupings of atoms, such as the methyl or grouping, in both the CHEMICAL-ENTITY and extended CHEMICAL-BOND relations. However, a very great refinement of this technique is both possible and desirable, when it is considered that many different groupings are formed by removal of

one or more atoms from some chemical compound.

The compound benzene is a common example of this. Benzene is a 6-carbon ring, with resonance bonds between the carbon atoms, and with each of the 6 carbons bonded to a hydrogen atom. Many chemical compounds are formed by removing a single hydrogen atom and replacing it with something else. If we replace it with chlorine, we get chlorobenzene, with an hydroxyl group, we get phenol (or hydroxybenzene), with an amine group we get aniline, and so on.

Thus we could regard a benzene molecule minus a hydrogen atom as a grouping, and store its structure in the data base. However, that is not a good idea, because we can also form many compounds be replacing two hydrogens from benzene by other atoms or groupings. Replacing the 1-carbon hydrogen by chlorine, and the 3-carbon hydrogen by bromine, for example, gives 1-chloro-3-bromobenzene; if instead we use hydroxyl groups as replacements we get 1,3-dihydroxybenzene, and so on. Thus a benzene molecule minus a specific two hydrogen atoms could also be regarded as a grouping and its structure could also be stored in the data base. We can continue until we are left with the bare 6-carbon ring with no hydrogens, which is the final grouping.

Clearly, it makes more sense to store only the structure of benzene, and structure a compound based on benzene has having had one or more carbon-hydrogen bonds broken, and replaced by new bonds. Thus phenol would consist of benzene less a carbon-hydrogen bond and plus a carbon-hydroxyl bond. The cleavage of a bond in a compound to form a new compound would be noted in the data base in

the BOND field by giving it the value -1 (or -2 in the rare case of cleavage of a double bond). In the case of a double bond -1 would mean cleavage of one of the two bonds). This concept is illustrated in the two-path data base approach.

**The two-path data base structure**

The two-path data base structure is best understood by studying an example such as that in Figure 2.

| IUPACNAME | M-CODE | A-CODE | MP | BP | TYPE |
|---|---|---|---|---|---|
| BENZENE | - | BNE | - | - | COMPOUND |
| HYDROXYBENZENE | HDB | - | - | - | COMPOUND |
| 2,3,6,7-DICHLORODIBENZO-p-DIOXIN | DOX | - | - | - | COMPOUND |
| DIBENZO-p-DIOXIN | DBZ | DBZ | - | - | COMPOUND |
| HYDROXY | - | OH | - | - | GROUP |

**CHEMICAL-ENTITY**

| CODE | EL2# | EL2 | EL1# | EL1 | ER2# | ER2 | ER1# | ER1 | BOND |
|---|---|---|---|---|---|---|---|---|---|
| HDB | 1 | BNE | 1 | C | 1 | BNE | 1 | H | -1 |
| HDB | 1 | BNE | 1 | C | - | - | 1 | OH | 1 |
| DOX | 1 | DBZ | 2 | C | 1 | DBZ | 2 | H | -1 |
| DOX | 1 | DBZ | 2 | C | - | - | 1 | C1 | 1 |
| DOX | 1 | DBZ | 3 | C | 1 | DBZ | 3 | H | -1 |
| DOX | 1 | DBZ | 3 | C | - | - | 2 | C1 | 1 |
| DOX | 1 | DBZ | 6 | C | 1 | DBZ | 6 | H | -1 |
| DOX | 1 | DBZ | 6 | C | - | - | 3 | C1 | 1 |
| DOX | 1 | DBZ | 7 | C | 1 | DBZ | 7 | H | -1 |
| DOX | 1 | DBZ | 7 | C | - | - | 4 | C1 | 1 |
| DBZ | 1 | BNE | 2 | C | 1 | BNE | 2 | H | -1 |
| DBZ | 1 | BNE | 3 | C | 1 | BNE | 3 | H | -1 |
| DBZ | 1 | BNE | 2 | C | - | - | 1 | O | 1 |
| DBZ | 1 | BNE | 3 | C | - | - | 2 | O | 1 |
| DBZ | 2 | BNE | 2 | C | 2 | BNE | 2 | H | -1 |
| DBZ | 2 | BNE | 3 | C | 2 | BNE | 3 | H | -1 |
| DBZ | 2 | BNE | 2 | C | - | - | 1 | O | 1 |
| DBZ | 2 | BNE | 3 | C | - | - | 2 | O | 1 |

**MOLECULAR-BOND**

| CODE | EL1# | EL1 | ER1# | ER1 | BOND |
|---|---|---|---|---|---|
| BNE | 1 | C | 2 | C | 1.5 |
| BNE | 2 | C | 3 | C | 1.5 |
| BNE | 3 | C | 4 | C | 1.5 |
| BNE | 4 | C | 5 | C | 1.5 |
| BNE | 5 | C | 6 | C | 1.5 |
| BNE | 6 | C | 1 | C | 1.5 |
| BNE | 1 | C | 1 | H | 1 |
| BNE | 2 | C | 2 | H | 1 |
| BNE | 3 | C | 3 | H | 1 |
| BNE | 4 | C | 4 | H | 1 |
| BNE | 5 | C | 5 | H | 1 |
| BNE | 6 | C | 6 | H | 1 |
| DBZ | 1 | C | 2 | C | 1.5 |
| DBZ | 2 | C | 3 | C | 1.5 |
| DBZ | 3 | C | 4 | C | 1.5 |
| DBZ | 4 | C | 11 | C | 1.5 |
| DBZ | 11 | C | 12 | C | 1.5 |
| DBZ | 12 | C | 1 | C | 1.5 |
| DBZ | 1 | C | 1 | H | 1 |
| DBZ | 2 | C | 2 | H | 1 |
| DBZ | 3 | C | 3 | H | 1 |
| DBZ | 4 | C | 4 | H | 1 |
| DBZ | 11 | C | 2 | O | 1 |
| DBZ | 12 | C | 1 | O | 1 |

| | | | | | |
|---|---|---|---|---|---|
| DBZ | 5 | C | 6 | C | 1.5 |
| DBZ | 6 | C | 7 | C | 1.5 |
| DBZ | 7 | C | 8 | C | 1.5 |
| DBZ | 8 | C | 9 | C | 1.5 |
| DBZ | 9 | C | 10 | C | 1.5 |
| DBZ | 10 | C | 5 | C | 1.5 |
| DBZ | 5 | C | 5 | H | 1 |
| DBZ | 6 | C | 6 | H | 1 |
| DBZ | 7 | C | 7 | H | 1 |
| DBZ | 8 | C | 8 | H | 1 |
| DBZ | 9 | C | 1 | O | 1 |
| DBZ | 10 | C | 2 | H | 1 |
| OH | 1 | O | 1 | H | 1 |

**ATOMIC-BOND**

Figure 2

The structure in Figure 2 has to be thought about carefully. It has some quite subtle recursion features.

With complex molecules there are always two pathways to the atomic structure. One pathway allows fast descent to the IUPAC structure. The other pathway can involve many recursion iterations in descending to the atomic level, but will extract the substructures that occur in the molecule at each iteration level. The

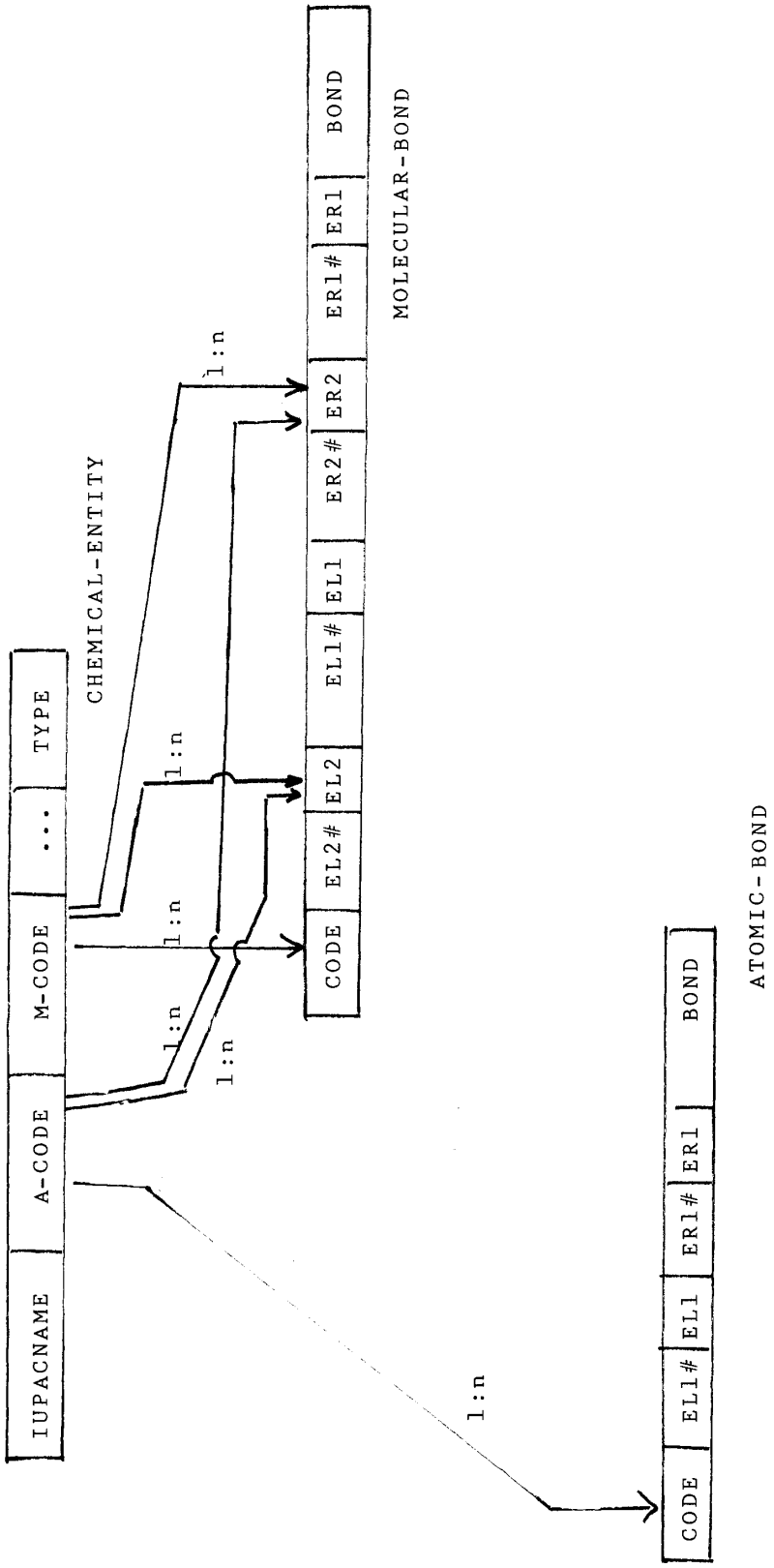


Figure 3. The general structure of a two-path data base for chemical entities.

structure diagram, showing the relationships between the relations for the data base, is in Figure 3.

The relation CHEMICAL-ENTITY has a tuple for each chemical entity, whether molecule or chemical group or substructure, and can contain information about the physical/chemical properties of the entity, such as boiling and melting points, where relevant. The fields M-CODE and A-CODE refer to codes used for the chemical entity in the CODE fields of the relations MOLECULAR-BOND and ATOMIC-BOND respectively (and also the EL2 and ER2 fields of MOLECULAR-BOND).

A tuple in CHEMICAL-ENTITY may have either an M-CODE value referring to tuples in MOLECULAR-BOND, or an A-CODE value referring to tuples in ATOMIC-STRUCURE, or both M-CODE and A-CODE values referring to tuples in both MOLECULAR-BOND and ATOMIC-BOND. There is thus a 1:n relationship between CHEMICAL-ENTITY and ATOMIC-BOND, and five distinct 1:n relationships between CHEMICAL-ENTITY and MOLECULAR-BOND. The fact that CHEMICAL-ENTITY participates in several one-to-many (1:n) relationships with MOLECULAR-BOND means that CHEMICAL-ENTITY participates in a many-to-many relationship with itself, and thus in a recursive many-to-many relationship [4, 5].

A group of tuples in ATOMIC-BOND with the same CODE field value gives the structure, in terms of atoms only and not higher-level substructures, of the chemical entity whose code in CHEMICAL-ENTITY.A-CODE matches that CODE value. Each tuple describes a chemical bond.

A group of tuples in MOLECULAR-BOND with the same CODE

field value gives the structure, in terms of both substructures and atoms, of the chemical entity whose code in CHEMICAL-ENTITY.M-CODE matches that CODE value. Each tuple describes a chemical bond.

**Extraction of chemical structure information from a two-path data base**

To use the data base to extract structure information about any common chemical entity that does not have substructures, or which is not a derivitive, only the relations CHEMICAL-ENTITY and ATOMIC-BOND need be used. For example, in Figure 2, if we want to find out about benzene, we need only access the record for benzene in CHEMICAL-ENTITY, and the related records (code BNE) in ATOMIC-BOND. Thus the structure data is extracted in a single iteration of the data base. For example, if we wanted to know how many carbon-hydrogen bonds there are in benzene, we could execute:

```
SELECT COUNT (*) FROM ATOMIC-BOND
WHERE EL1 = 'C' AND ER1 = 'H' AND
        CODE IN (SELECT A-CODE FROM CHEMICAL-ENTITY
                WHERE IUPACNAME = 'BENZENE');
```

Conventionally, in a bond between carbon and a non carbon atom, the carbon is in the left field.

We can tell from the CHEMICAL-ENTITY tuple for benzene that it has no substructure made up of other common chemical entities, since there is no value in the M-CODE field, and a value only in

the A-CODE field. If we want a list of the names of the simple molecules in the data base, we simply execute:

```
SELECT IUPACNAME FROM CHEMICAL-ENTITY
WHERE M-CODE IS NULL.
```

In contrast, if we need the structure of the benzene derivitive hydroxybenzene (or phenol), the tuple in CHEMICAL-ENTITY for that compound has an M-CODE value, and no A-CODE value. This tells us that the structure is given in MOLECULAR-BOND and is derived from one or more substructures. If we use this M-CODE value (HDB) to access MOLECULAR-BOND we find that the matching tuples indicate that this compound is formed by cleaving a hydrogen bond in benzene and replacing the hydrogen atom with the hydroxy group (-OH). The structure of benzene (and the hydroxy group, if required), and the relevant carbon numbers used in MOLECULAR-BOND can be found by referring to ATOMIC-BOND.

These two cases are simple and obvious. The cases of dioxin (2,3,6,7-tetrachlorodibenzo-p-dioxin) and dibenzo-p-dioxin illustrate how the data base handles complex molecules. Dioxin is derived from dibenzo-p-dioxin by chlorination. But many other substances are derived from dibenzo-p-dioxin. At the same time dibenzo-p-dioxin is derived from two benzene molecules.

If we look up dibenzo-p-dioxin in CHEMICAL-ENTITY, we see that it has field values in both M-CODE and A-CODE. This means that we can get at its structure in two ways. We can use the A-CODE value ('DBZ') and ATOMIC-BOND to get the structure in terms of

atoms only (not in terms of substructures) with the IUPAC numbering code in the carbon occurrence number fields (EL1# and ER1#). Thus the ATOMIC-BOND tuples with CODE value 'DBZ' give each atomic bond in dibenzo-p-dioxin.

However, if we use just CHEMICAL-ENTITY and ATOMIC-BOND we could not ask for information about substructures, for example, the number of benzene rings that dibenzo-p-dioxin contained. To do that we need to use the M-CODE value 'DBZ' and reference the tuples in MOLECULAR-BOND with code 'DBZ'. These tuples give the structure in terms of benzene, whose atomic structure is found in ATOMIC-BOND. The result of using MOLECULAR-BOND tuples with lower level references to ATOMIC-BOND tuples to get the structure of dibenzo-p-dioxin will be correct both in terms of substructures and atoms.

However a structure generated this way can not be used for obtaining the structure of substances derived from dibenzo-p-dioxin, such as dioxin, since the the carbon-atom numbering scheme will not be the IUPAC one. Nevertheless, this method of getting at the substructure will enable queries about substructure to be handled. For example, suppose we want the number of benzene rings in dibenzo-p-dioxin. We would execute:

```
SELECT MAX (EL2#) FROM MOLECULAR-BOND
WHERE EL2 = 'BNE' AND
    CODE IN (SELECT M-CODE FROM CHEMICAL-ENTITY
            WHERE IUPACNAME = 'DIBENZO-p-DIOXIN');
```

Now suppose we need the structure of dioxin, which is a

derivitive of dibenzo-p-dioxin. The tuple in CHEMICAL-ENTITY gives only an M-CODE value ('DOX'), indicating that the substance is derived from other substances, portrayed in the matching records of MOLECULAR-BOND. These records show that dioxin is formed from dibenzo-p-dioxin by replacement of hydrogen atoms on four carbon atoms by chlorine atoms. The carbon atom occurrence numbers match the IUPAC numbers (2, 3, 6, 7) and in the MOLECULAR-BOND tuples these carbon atoms are shown as belonging to the dibenzo-p-dioxin molecule. To get at the structure of dibenzo-p-dioxin, we have two pathways

(a) via BNE MOLECULAR-BOND tuples that give the structure in terms of benzene rings but in the end without correct IUPAC carbon atom occurrence numbers, or

(b) via BNE ATOMIC-BOND tuples that give the structure only in terms of constituent atoms but with correct IUPAC carbon atom occurrence numbers.

Obviously we take route (b). It is for this reason that we say that the data base has a two-path structure. The correct IUPAC structure for dioxin is shown in Figure 4.

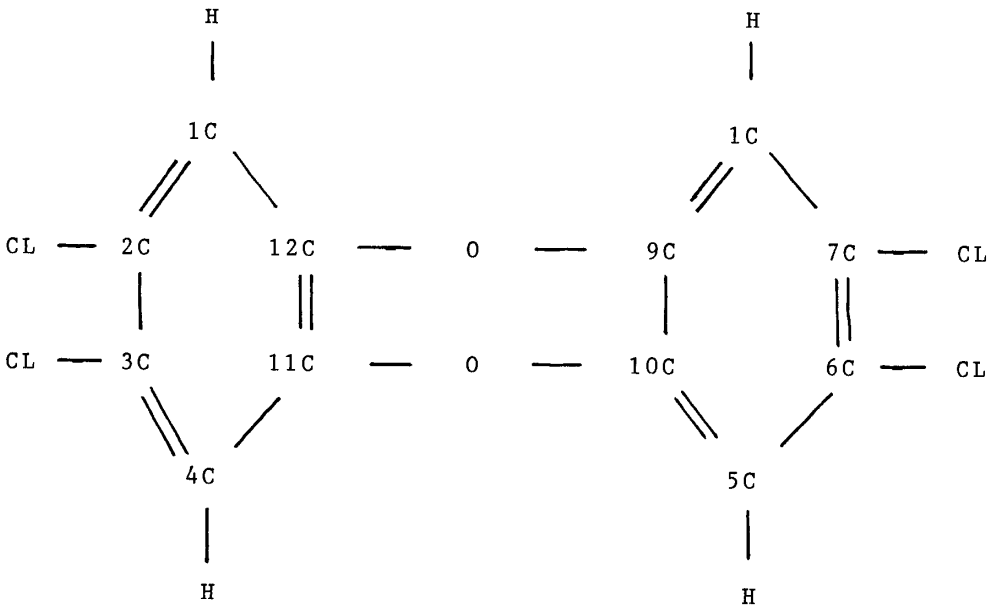


Figure 4

It is useful to be able to refer to the two paths. Suppose we call the two paths the A-path (direct pathway to atomic structure with IUPAC carbon atom occurrence numbers) and the M-path (pathway to molecular substructures).

Some rules are needed for correct use of the data base. Suppose we have an unknown compound XYZ, whose structure we wish to determine. We use the following general algorithm:

(a) Access the record for XYZ in CHEMICAL-ENTITY.

(b) If there is an A-CODE value and no M-CODE value, the compound has no substructure but may be a derivitive of (occur within) compounds. Use the reference to ATOMIC-BOND to

get the atomic structure with correct IUPAC carbon atom occurence numbers, that is, follow the A-path.

(c) If there is an M-CODE value the compound has substructure, and if there is an A-CODE in addition the compound is used as a derivitive of other compounds (at a higher level). Follow the A-path using A-CODE to get the atomic structure of XYZ with corect IUPAC carbon atom occurrence numbers. Follow the M-path down through multiple levels, where possible, only to determine constituent substructures within XYZ (including atoms).

(d) If there is an M-CODE value and no A-CODE value, the compound has substructure but is not itself a derivitive of any other compound. Follow the M-path to the next level down and then use the A-path to obtain the final IUPAC atomic structure for XYZ. Continuation on the M-path down through multiple levels, where possible, will give the substructures within XYZ.

**Two-path data base for proteins and enzymes**

The data base structure in Figure 2 appears to be capable of holding the structure of all known chemical compounds in conformance with IUPAC numbering rules. It is interesting to consider how structure information about protein molecules would be held in a two-path relational data base. Because of their variety, importannce and complexity, data base techniques are particularly relevant to protein and enzyme molecular information systems [2, 3,

6, 11, 16].

A chain of amino acid residues is a polypeptide, and a protein is a group of one or more (usually intertwined) long polypeptide chains occurring naturally [7]. Note that if we have three amino acid residues (A, B, C) in a chain A-B-C, the chain can be formed in two ways, namely H-A-B-C-OH, or HO-A-B-C-H, since a given sequence of residues can begin with either an amino-group or a carboxyl group. Thus the sequence of amino acids in itself is not enough to uniquely determine the nature of the polypeptide.

In addition to the polypeptide bonds that form the chain, in a protein there can be (a) cross-link bonds between side chains and (b) hydrogen bonds between residues resulting in helix formation. In a protein, there are usually a few of these cross-link bonds between side chains of different amino acids. For example, in the protein bovine ribonuclease, with 124 amino acid residues in a single chain, in addition, residue 40 is further bonded (cross-linked) to residue 95 via the side chains, and some other residues are also cross-linked. Furthermore, parts of the sequence of a chain can coil into a helix (alpha-helix) [7, 8], and for this to happen the hydrogen on the N of residue n will hydrogen bond to the oxygen on residue n + 4.

With a two-path data base the structure of a protein in terms of amino acid residues would be held in MOLECULAR-BOND. Each tuple would describe a link between two amino acid residues of a chain. There could be several chains. In addition there would be a tuple for the H-N bond at one end of a chain and a C-OH bond at the

other. This is illustrated in Figure 5, which shows the 7-residue polypepide serine-glysine-serine-cysteine-alanine-serine-serine (CODE 'PPD'). The data base also shows a hydrogen bond between residue 2 (glycine) and residue 6 (serine).

| CODE | EL2# | EL2 | EL1# | EL1 | ER2# | ER2 | ER1# | ER1 | BOND |
|---|---|---|---|---|---|---|---|---|---|
| PPD | 1 | SER | 1 | N | - | - | 2 | H | 1 |
| PPD | 1 | SER | 1 | C | 1 | GLY | 1 | N | 1 |
| PPD | 1 | GLY | 1 | C | 2 | SER | 1 | N | 1 |
| PPD | 2 | SER | 1 | C | 1 | CYS | 1 | N | 1 |
| PPD | 1 | CYS | 1 | C | 1 | ALA | 1 | N | 1 |
| PPD | 1 | ALA | 1 | C | 3 | SER | 1 | N | 1 |
| PPD | 3 | SER | 1 | C | 4 | SER | 1 | N | 1 |
| PPD | 4 | SER | 1 | C | - | - | 1 | OH | 1 |
| PPD | 1 | GLY | 1 | O | 3 | SER | 2 | H | 0.1 |

**MOLECULAR-BOND**

Figure 5

The atomic structure of each amino acid residue of a protein would be stored in ATOMIC-BOND. There would also be tuples in MOLECULAR-BOND for any additional (cross-link) bonds between the amino-acid residues of the chain.

If the polypeptide chain has n residues, it takes n + 1 tuples in MOLECULAR-BOND to describe it if there are no side-chain

cross-links or hydrogen bonds between residues. Each cross-link or hydrogen bond would need an additional tuple. Thus PPD, with 7 residues and one hydrogen bond requires (7 + 1) + 1, or 9 tuples. It is therefore clear the structure of any protein, no matter how complex, can be stored in atomic detail, using the two-path data base structure in Figure 2.

[Only an amino acid residue would be stored in ATOMIC-BOND for purposes of storing protein structure. Nevertheless, in a universal chemical data base the structure of an amino acid would also be in MOLECULAR-BOND, probably as two tuples, recorded as a residue bonded to an H atom and bonded to an OH group. The structure of a residue in terms of further substructures, especially where the side chain was complex, might also be in MOLECULAR-BOND.]

In the case of an enzyme, the structure of the coenzyme would typically be in MOLECULAR-BOND in several ways. Atomic details of amino acid residues and other constituent compounds (of the coenzyme) would be in ATOMIC-BOND.

**Isomers**

Consider fumaric and maleic acid. These compounds are isomers (geometrical isomers). They have the same chemical structure as far as the existence of specific chemical bonds is concerned, but yet are different because of the directions of the bonds in space. There are other types of isomers, such as optical isomers, involving right-handedness and left-handedness (as in L-alanine and D-alanine) [14]. As presented so far, the two-path data base in Fig-

ure 3 could not distinguish fumaric and malic acid, nor L-alanine and D-alanine.

The problem of isomers can be solved as far as a two-path data base is concerned, by employing the concept of a zero strength (or imaginary) bond. We simply add some additional tuples to the data base. The additional tuples all have BOND value 0, indicating an imaginary bond between atoms (or groups) that places the atoms, or projections of atoms onto a plane, in an imaginary ring structure. The order in the ring serves to distinguish the isomers. The ATOMIC-BOND relation in Figure 6 uses imaginary isomer rings to distinguish fumaric and maleic acid. (For the sake of brevity, we have treated -COOH as an atom; strictly, in ATOMIC-BOND only atoms may be used.)

| CODE | EL1# | EL1 | ER1# | ER1 | BOND |
|---|---|---|---|---|---|
| FUM | 1 | C | 1 | COOH | 1 |
| FUM | 1 | C | 1 | H | 1 |
| FUM | 2 | C | 2 | COOH | 1 |
| FUM | 2 | C | 2 | H | 1 |
| FUM | 1 | C | 2 | C | 2 |
| FUM | 1 | COOH | 2 | H | 0 |
| FUM | 2 | H | 2 | COOH | 0 |
| FUM | 2 | COOH | 1 | H | 0 |
| FUM | 1 | H | 1 | COOH | 0 |

| | | | | | |
|---|---|---|---|---|---|
| MAL | 1 | C | 1 | COOH | 1 |
| MAL | 1 | C | 1 | H | 1 |
| MAL | 2 | C | 2 | COOH | 1 |
| MAL | 2 | C | 2 | H | 1 |
| MAL | 1 | C | 2 | C | 2 |
| MAL | 1 | H | 2 | H | 0 |
| MAL | 2 | H | 2 | COOH | 0 |
| MAL | 2 | COOH | 1 | COOH | 0 |
| MAL | 1 | COOH | 1 | H | 0 |

**ATOMIC-BOND**

Figure 6

The isomer-ring technique, with zero (strength) bonds, is simple and effective, and can be applied to all isomeric compounds, without it being necessary to clutter the data base with angle and vector data. The very existence of a zero strength bond can also be used to denote that a compound is an isomer. Thus the SQL expression:

```
SELECT IUPACNAME FROM CHEMICAL-ENTITY
WHERE A-CODE IN (SELECT CODE
                 FROM ATOMIC-BOND
                 WHERE BOND = 0);
```

gives a list of all isomers in the data base. The isomer ring technique does not appear to have been proposed hitherto in the chemi-

cal literature. (The reader is invited to extract the structures of fumaric and maleic acid from the data base and observe the distinction.)

**Three dimensional structure of molecules**

The two path data base structure proposed so far can hold the structure of the most complex molecules, but only in terms of constituent groupings and atoms. We have not so far discussed recording of data in the data base about the structure of molecules in 3-dimensional space [7, 8]. Three-dimensional structure is important in applications involving the chemical activity of molecules, which depends on surface structure and the electrical potentials (Van der Walls potentials) at the surface [11, 18]. The surface structures of enzymes is an important area of on-going research.

It would clearly be possible to extend the two path data base so that a vector (for length and direction) relative to a standard reference frame be recorded with every bond. This non-predictive approach would enable the three dimensional structure of any molecule to be generated. An alternative, more predictive, approach is to include the data on the bond angles possible with each type of bond for each type of atom in a separate relation of the two-path data base. Extension of the two-path approach to include 3-dimensional structure data is the subject of a separate paper.

**SUMMARY**

A relational data base structure, called the two path structure, has been proposed for holding the structure of all chemical entities. The conventional bill-of-materials recursive data base structure for complex objects cannot be applied to a data base for chemical entities because of restrictions involving the need to specify the bonding between high level substructures in terms of bonds between atoms, the lowest level entities in the structure. There are further restrictions of design freedom because conventional IUPAC carbon atom occurrence numbers must be obtainable from the data base. The two-path data base solves the restriction problems by providing two pathways to the structure of complex chemical entities. One pathway allows a descent to the atomic structure in terms of correct IUPAC carbon atom occurrence numbers in just one or two iterations of the structure. The other pathway allows many iterations of descent in terms of substructures, for purposes of referencing substructures at any level in retrievals. A two-path data base is recursive in nature.

The data base appears to be capable of faithfully recording the structure of all chemical entities, including proteins and enzymes. The two-path approach is based on a tuple per chemical bond. By using negative bond strengths, a substructure can be recorded as a molecule minus some bonds plus some other bonds. Information needed to distinguish chemical isomers can be included quite easily, by using bonds of zero strength that form imaginary rings. This has been called the isomer-ring technique.

**REFERENCES**

1. Adamson, G.W., Bird, J.M, Palmer, G., Warr, W.A., 1986. In-house chemical data bases of Imperial Chemical Industries, Journal of Molecular Graphics 4(3), 165-177.

2. Allen F.H., et al, 1979. The Cambridge Crystallographic Data Centre: Computer-based search, retrieval, analysis and display of information, Acta Cryst., B35, 2331-2339.

3. Bernstein, F.C., et al, 1977. The protein databank: A computer-based archival file for macromolecular structures, Journal of Molecular Biology, Vol. 112, 535-542.

4. Bradley, 1982. File & Database Techniques, Holt, Rinehart & Winston, New York.

5. Bradley, J., Recursive relationships and natural quantifier set theoretic expression techniques, Computer Journal, in press.

6. Bryant, S.H., Sternberg, M.J.E.,1987. Comparison of protein structural profiles by interactive computer graphics, Journal of Molecular Graphics, 5(1), 4-7.

7. Clothia, C., 1984. Principles that determine the structure of proteins, Ann. Rev. Biochemistry, vol. 53, 537-572.

8. Fletterick, R.J., Schroer, T., and Matela, R.J., 1985. Molecular structure: Macromolecules in Three-Dimensions, Blackwell Scientific Publications.

9. IBM Corp., 1984. IBM DATABASE2 General Information, Form GC26-4073, IBM Corp., Endicott, New York.

10. ORACLE Corp., 1984. ORACLE SQL/UFI Reference Guide, Menlo Park, California.

11. Jakes, S.E., Willet, P., 1986. Pharmacophoric pattern matching in files of 3-dimensional chemical structures: Selection of inter-atomic distance screens, Journal of Molecular Graphics, 4(1), 12-20.

12. Kim, Won, 1979. Relational database systems, ACM Computing Surveys, 11, pages 185-211.

13. Maier, D. 1983. Theory of Relational Databases, Computer Science Press, Potomac, Maryland.

14. Pine, S. 1987. Organic Chemistry, 5th Ed., McGraw-Hill, New York, Chapter 3.

15. Relational Technology Inc., 1984. INGRES Reference Manual, Berkeley, Calif.

16. Ricketts, D., 1987. Perq: interactive molecular modelling systems, Journal of Molecular Graphics, 5(2), 63-70.

17. Stryer, L., 1988, Biochemistry, W.H. Freeman & Co., New York.

18. Taylor, R ., 1986. The Cambridge Structural Database in molecular graphics: Techniques for the rapid identification of conformal minima, Journal of Molecular Graphics, 4(2), 123-131.