

ADABASについて

石井 義興

(株) ビジネス・コンサルティング

ADABASは、西独SOFTWARE AG社が1971年に発表したDBMSである。ADABASが開発された背景は、理論などではなく、日常の不便さをいかに減かすせよという努力の集大成である。ADABASが西独で開発されたという点に注意しなければならない。いわゆるアメリカ臭がない。時を同じくして発表されたE.F.CODDのリレーショナル・モデルの考え々と共通点の多い事も興味深い。

ADABASは、PETER SCHNEELIという1人の技術者によって長期間検討され、1969年からこのシステムが開発された。デザイン、プログラミング、デバッグは、ほぼ1人のアイデアによって行われ、全体は5人で開発された。その内ユティリティを別にすれば、ADABAS本体は1人で全てが行われた。「虫の多い、質のよいソフトウェアを開発するためには、大勢で開発すべきではない」という彼の思想からである。

SOFTWARE AG社は、ADABASを開発・販売するために作られた会社である。これは、PETER KREISという1人の経営者の次の思想に基づいている。「本当に優れた物を作り、販売するためには、他の仕事とまざって行う事、金でのりスくは経営者が持つべきである」と。

ADABASがデザインされ開発された時侯は、いわゆるテータ・ベース論議の始まった頃である。CODASYLのSystems Committee, DBTGなどの活動と時を同じくしている。しかし、ADABASを開発した技術者達は、CODASYLやCODDの理論などには興味がなく、これらの活動について知っていた様子は無い。現在でも興味はない様だ。しかしここ重要なことは、両者とも全く独立して、お互いを知らずに行っていたが、今に共通の基盤があったということである。これは、システム開発手段をより強力にしようという事である。プログラミング言語と、ファイルのアクセス、メソッドに改善を加え、テータの独立性の保持や、リカバリーの共通化、セキュリティ保護、などの問題を解決し、システム 開発、保守の生産性の向上をはかろうというのである。現在は、ドイツで作られた中核部分にアメリカの血がまじり、言語の階層化や、処理の効率化、オンライン化などが行われている。

ADABASは元来西独のソフトウェア会社が作成したソフトウェア・パッケージであったが、それが金世界に広まり、使われる様になったのは、アメリカのソフトウェア、マーケティングの専門家、JOHN MAURELIERが、ADABASの機能の優秀性を早くから理解し、そのマーケティング活動に打ちこんだからである。現在は、北米、ヨーロッパ、日本は勿論、南米、アフリカにもユーザーがいる。

1. ADABASのねらい

ADABASは次の様な事を改善しようとしてデザインされた。

- ・ テータのタイポやリンクスの変更が容易でも、必要作ったプログラムが何となくのまどろみがないが、

- ・ レコード内に定義されているフィールドの順序が変わっても、以前作ったプログラムがそのまゝ動かないか
- ・ レコード(ファイル)に新しいフィールド(属性)を追加されて、レコード長やレイアウトが変わっても、以前作ったプログラムがそのまゝ動かないか
- ・ 値の入っていないフィールドや、値の入っていないくり返し、英数字項目のトレーディング、スペース、数字項目のリーディング、ゼロなどを何とかがテキストに入れない方法はないか。
- ・ 5桁と定義したフィールドに7桁、10桁と、より長いデータを入れる方法はないか。
- ・ インター、レコード、リレーションを自然に作る方法はないか。
- ・ 1つのファイルに多くのダイレクト、アクセス、キーを付けられないか
- ・ 1つのファイルに対して内容検索がスピーディに行えないか。
- ・ 複数回のファイルにまたがって、スピーディに内容検索を行えないか。
- ・ O/Sや、I/Cが変わっても、変更なしにできるか。
- ・ オペレーションナルなDBとマネージャナルなDBを同一に取扱えないか。
- ・ テキスト上のファイルのレコード件数を何時でもえる方法はないか。
- ・ どんな状態でダウンロードしても、それを簡単にリカバーする方法はないか。
- ・ データ、プロセス、ツニングの中で非効率が多い、ソート、マージを減らす方法はないか。
- ・ テキストのサイズが変わっても、以前作ったプログラムがそのまゝ動かないか。
- ・ デバイス、ハイパーキを容易に作れないか。
- ・ データ、ベースへのデータのローディングは簡単に行えないか。
- ・ データ構造などを考えないで済まないか。
- ・ システム全体をオプションマイズする様に、時々調節する方法はないか。

これらさらに3年、次の諸案が追加された。

- ・ 世界の稼動しているあらゆる優れたTPモニタと、インタフェースをとれないか。
- ・ ユーザレベルによって多種類の利用言語を供用できないか。
- ・ 多重処理が可能とならないか。
- ・ 権限保護、サイファリング機能が簡単に行えないか。
- ・ 非同期に稼動し、同一ファイルを更新しているプログラム群がマシンのダウン後も、同時にリカバー、リスタート出来るか。

以上のADABASの今日までのわらひである。今後は更に、次のわらひを持っている。

- ・ ADABASを、ソフトウェアではなく、ハードウェア化できないか。
- ・ 集合論的データ、プロセスツニングの方法に指向できないか。
- ・ ベーザアル、フィールドや、ベータアル、ファイルの取扱いができないか。

このわらひは、ADABASの機能を述べたことにもなる。

2. ADABASの構成

ADABAS DATA BASEの構成、ADABASのユーザ、インタフェース

ース、DBA機能について、以下述べる。

2.1 ADABAS Data Baseの構成

ADABAS Data Baseは、次の3つのデータ・セットから構成される。

ASSOCIATOR DATA SET
DATA STORAGE DATA SET
WORK DATA SET

ASSOCIATORは、スキーマ、パスワート、諸テーブル、INVERTED INDEX、INTER RECORD RELATION TABLEなどが入っているデータ・セット、ストレッジである。

DATA STORAGE DATA SETは、データ・ベースに投入されたファイルが入っているデータ・セット、ストレッジである。

ASSOCIATORとDATA STORAGEとの関係は、図書館に例えればASSOCIATORは、本のINDEX類で、DATA STORAGEは、表庫そのものと云える。

WORK DATA SETはASSOCIATOR内のINDEXに対して集合演算などを行なった際のワークとして、また、リカバーのための諸情報の貯蔵場所として用いられる。

ADABASのデータ・ベースとは、単純にファイルの集合の事である。

ASSOCIATORは、ADABASによってすべて管理され、ユーザーは、それの存在や操作に交与する必要はなく、WORKデータ・セットに因ることも同様である。

DATA STORAGEには、データが格納され、ポインターなどはいっさい入っていない。

2.2 ADABASのユーザ・インタフェース

データ・ベースを利用する人とは、いろいろの階層に分けられる。DBAについて後述にふれることにし、ここでは直接それを利用する人々に対する言語について述べる。

ADABASには、次の言語インタフェースがある。

会話型言語インタフェース

- ・ADASCRIP T
- ・ADACOM

リポート・ライタ

- ・ADAWRITER
- ・MARKIV
- ・EASYTRIEVE

ハイレベルDML

- ・ADAMINT

ローレベルDML (親言語方式)

- ・COBOL/FORTRAN/PL/I/ASMB

エンド・ユーザ・インタフェース

プログラマ・インタフェース

これらは全て、ADANUCとよばれるADABAS本体と、インタフェースを持ち、データ・ベースとのアクセスは、ADANUCによって行われる。

2.3 DBA機能

データ・ベースをスムーズに維持し、正しく、かつ最適に保持するための機能として、DBA (Data Base Administrator) 機能は重要である。

ABABASのDBA機能は強力である。それらを列挙すれば、次の通りである。

- ・ 今、各ファイルに何件のデータが投入され、それがディスク上のどの部分に、どこに何のスペースを占めているか。
- ・ 各ファイルのフォーマットはどうか。
- ・ どのファイルとどのファイルの間に関係があるか。
- ・ チェックポイントとは、何時の時点でとられているか。
- ・ あと使用されていないスペースは、どの程度残っているか。

などは、何時でも知ることが出来る。

又、次の様な事が起っても、容易に、ユティリティにエントリ対処することが出来る。しかもこの際、データ・ベース上で稼働していたプログラムには、何の変更もない。

- ・ ディスクが一杯になったら拡張出来る。
- ・ データ・ベースを、ある時点で戻せる。
- ・ 新ディスク、デバイスの増設が容易である。
- ・ ダンプ・リストア
- ・ 多量データのデータ・ベースへの追加
- ・ 新フィールドの追加
- ・ 新しいファイル間リレーションの設定
- ・ キー値のフィールドのキーへの変換
- ・ ファイルの削除、キーの削除、ファイル間リレーションの削除。
- ・ データ・ベースのリオーガニゼーション

これらは全て、ユティリティにまつて行なうことが出来る。

ABABASのDBA用ユティリティは次の通り。

- | | |
|----------------|---------------------------|
| ・ LOADER | データ・ベースへのデータロード |
| ・ ADDLOAD | 多量データの追加 |
| ・ COUPLING | インタ、ファイル、リレーションシップ |
| ・ FILEMOD | 新フィールドの追加、新キーの追加など |
| ・ DDMOD | エクスポート、ストレージ、プリント、ユーティリティ |
| ・ FORMATING | データ・ベースのフォーマット |
| ・ DB REPORT | DBのステータス、レポート出力 |
| ・ DUMP/RESTORE | ダンプ・リストア |
| ・ RESTART | リスタート |
| ・ SECURITY | 機密保持 |
| ・ UNLOAD | ファイルのアンロード |

これ以外のDBA機能として重要なものに、DD/DSDがある。

ABABASのDD/DSDとしては、ABABAS DATA DICTIONARY SYSTEMがあり、市販DD/DSD用ソフトウェア・パッケージ、DATA MANAGERも、ABABASをサポートしている。

3. ABABAS コマンド

全てのABABASコマンドは、ABABASユーザリクスを通して実行される。ABABASがサポートするコマンドは以下に示すものである。

FIND, READ, UPDATE, ADD, DELETE, OPEN/CLOSE, CHECKPOINT

3.1 ADABASのFINDコマンド

FINDコマンドを使用することによって、ユーザはデータベース内の特定のレコードの論理的検索を遂行することができる。FINDコマンド内に、サークル条件あるいは使用するべき条件を指定する。

ADABASでは、1個以上のテキストリット、フィールドから部分サークル条件を構成し、複数の部分サークル条件から一つのサークル条件を作る事ができる。そのとき使用する論理演算として、次のものがある。

AND, OR, FROM-TO, BUT-NOT

ADABASは検索を実行し、その結果をリストにリスト出力する。そのリストの内容は、サークル条件、その他の条件に合致した値を持ったレコードのISN (INTERNAL SEQUENCE NUMBER) である。ISNは、ADABASが持つレコード、アイデンティファイアのことである。

ADABASでは検索の実行にあたって、データベースのデータ・ストレージ部分をアクセスしない。このことは特に注目すべきことである。ASSOCIATE CR中に、すべての必要な情報をインバータッド・リストの形で保持している。ゆえに、検索の実行に際して必要な物理的アクセスの回数は美値的に減少し、データ・ストレージ内にある実際のデータ・ポリユーラの独立性は、一層強化されたものとなっている。

ADABASには、幾つかの異なったサークルのアルゴリズムがある。ADABASは、値内のタイプと合成の度合いを基準にして、最も効率的なアルゴリズムを自動的に選択する。さらにADABASでは、アソシエータ内を通り抜けるマールレベル、インデックスを保持しており、必要とする情報の位置をすみやかに知ることができるようになっている。

次の例は、ADASCRIPTを用いたサークルの値内を示したものである。

```
FIND IN PERSONNEL-FILE ALL RECORDS  
WITH ID-NUMBER=54711.  
FIND IN APPLICANT-FILE ALL RECORDS  
WITH TRAINING=EDP AND EXPERIENCE GT5  
AND LANGUAGE=COBOL OR FORTRAN AND AGE LT 40.
```

FINDコマンドには、
FIND Sorted と
FIND Coupled.
および集合演算コマンドの
三つの変形がある。

FIND Sorted コマンドが通常のFINDコマンドと異なる点は、リストによって指定された順序にソートされた集合 (ISNのリスト) を返すことである。ユーザは、大分類へ小分類の基準として1〜3個のキーを指定することができる。ソート条件として指定するこれらのキーを、必ずしもサークル条件にする必要はない。

```
FIND IN PERSONNEL-FILE ALL RECORDS  
WITH DEPT-NUMBER = 740  
AND SORT BY SECTION-NUMBER, LAST-NAME, FIRST-NAME.
```

FIND Coupled
コマンドは、ポインタを
たいてい命令に対応する。

FIND Coupled コマンドを用いる前には、ユティリティによってファイルは論理的に結合してなければならぬ。これをファイルのカップリングと呼ぶ。ひとたびファイル間の関係が設定されれば、異なったファイル中のキーは一つの検索の一部として使用できる。

カップリング手順を解りやすくするために、2本のファイルがあるとしよう。1本のファイルには、住所に関する情報が入っており (Resident-File)、もう1本のファイルには所有権に関する情報が入っている (Property-File)

とする。

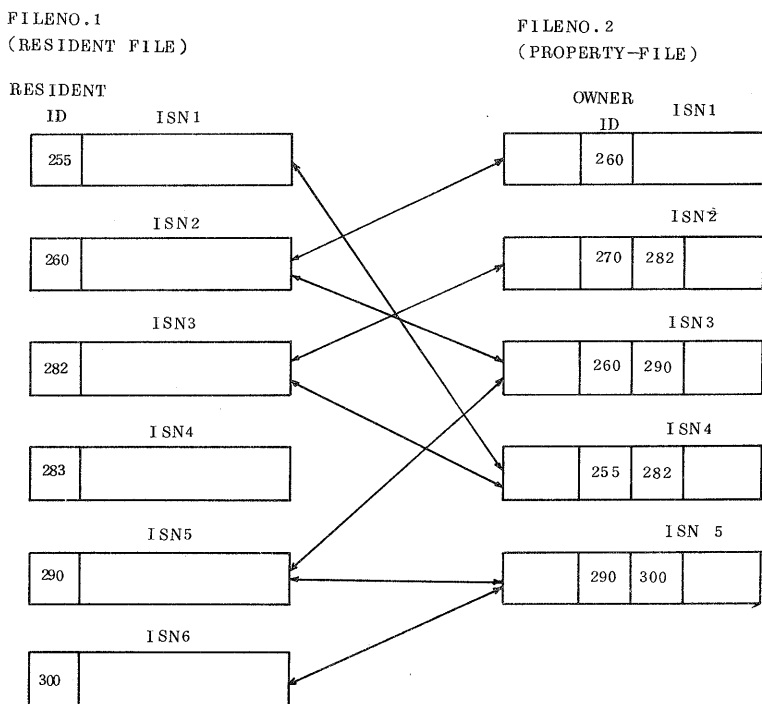
Resident-File の各レコードには、居住者を一意的に識別するための Resident-ID というキー・フィールドがある。

Property-File の各レコードには、特定の土地の所有者を記録した Owner-ID というキー・フィールドがある。

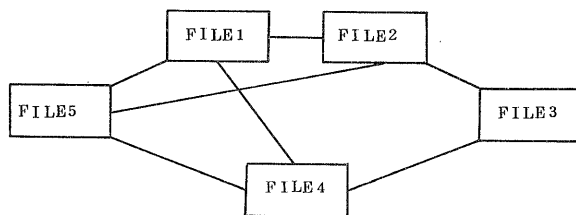
いずれのファイルは、Resident-ID と Owner-ID という共通の内容を持つキー・フィールドを介して所有権という関係で論理的に結合されているとする。

1人の居住者は、何筆かの土地を所有している可能性があり、1筆の土地には複数の所有者がいる可能性 (Owner-ID はマルチプル フィールドであるかもしれない) があるので、カプリングの処理によって Resident-File の1個のレコードは、Property-File の複数個のレコードと結びつくかもしれない。その逆も起りうる。

ADABAS FILE COUPLING



従って、一般的には次の様な構造も考えられる。



ファイルのカウプリングを済ませた後は、両方のファイル中のキー、フィールドからサーチ条件を作成することができる。

```
FIND IN RESIDENT-FILE ALL RECORDS WITH  
AGE GT 40 AND SALARY GT 20000  
AND COUPLED WITH PROPERTY-FILE WITH  
PROPERTY-SIZE GT 150.
```

始めの二つの条件(年齢と俸給)は Resident-File の中のキー、フ

ィールドであり、Property size は、Property-File 中のキー、フィールドである。

上の例では、Resident-File がコマンドの始めに述べられており、それゆえ、この検索では主ファイルとみなされる。ADABAS は、始めの二つの条件を満たす Resident-File 中のレコードで、かつ、それに結びついた Property-File 中のレコードから着目条件を満たしているレコードを見い出し、それらのレコードの集合 (ISN) をリストにこて返す。

主ファイルと二次的ファイルの関係を覚えていなら、それは単に検索コマンドの中のサーチ条件の順序を逆転するだけよい。カウプリングが両方向的属性を持っているので、新たなカウプリング関係の設定を必要としない。

```
FIND IN PROPERTY-FILE ALL RECORDS WITH  
PROPERTY-SIZE GT 150  
AND COUPLED WITH RESIDENT-FILE WITH  
AGE GT 40 AND SALARY GT 20000
```

この例のファイルで、例えは所与の検索別のキーを用いて、論理的に

異ったカウプリングをとることをできる。その場合は、論理的に完全に異ったカウプリング関係となる。

集合演算コマンド (88 コマンド) は、すでに得られた2つの集合に対して、AND、OR 等の集合演算を行ない、新たな集合を得るためのコマンドである。

3.2 READ コマンド

READ コマンドには、5つのタイプがある。

```
READ RANDOM  
READ LOGICAL SEQUENTIAL  
READ PHYSICAL SEQUENTIAL  
READ DESCRIPTOR VALUE  
READ FIELD DESCRIPTION TABLE
```

前の3つのコマンドはデータベースからレコードを読み込むためのものであり、後の2つはアソシエータのアクセスにのみ使用される。

READ RANDOM コマンドは、データストレージから1個のレコードを読み込む。

READ LOGICAL SEQUENTIAL コマンドは、1つのファイル全体の読み込みや、ディスクリアク、フィールドの値が一定範囲にあるレコードの読み込みで使用される。このコマンドを発行する際に、前処理としてのリストを必要としない。それは、アソシエータ内に ISN について分類済みのリストが存在し、それが一定の順序を決定する際に使用されるからである。

READ PHYSICAL SEQUENTIAL コマンドは、物理的順序に従って2つのファイルを読み込むときに使用する。レコードの読み込まれる順序は、レコードがデータ・ストレージ内で現在物理的に位置している順序とまったく同じである。この読み方は SAM に対応する。

ADABAS では、データ・ストレージ内の一つの物理的ブロックに対して、圧縮された論理レコードをストアするので、ファイルを逐次処理していくために必

要な物理的アクセスの回数は小さく減少する。また、一度の物理的アクセスで多くの論理レコードを読み込むことになるので、コールドレジを小さく効率的に使用することになる。

READ DESCRIPTOR VALUE コマンドは、任意のディスクリポジション(キー・フィールド)に関するデータ・ベースの状態を調べするために使用する。ユーザには、元のキー・フィールドの値と共に、その登録数を通す。このコマンドは、アソシエータだけをアクセスする。

READ FIELD DESCRIPTION TABLE コマンドは、データベース中の任意のファイルに対する、フィールド・ディスクリポジション・エントリを読み込むときに使用する。

各フィールドに対して、フィールド名、標準長、標準フォーマット、フィールド属性等を得ることができる。

3.3 UPDATE コマンド

ユーザは、UPDATE コマンドを用いて、レコード中の任意のフィールドの値を修正することができる。新しいフィールド値が必要とするフィールド長は旧値のフィールド長より長くても短くてもよい。そのため柔軟性のある更新機能によって、レコード中のフィールドの拡張は、非常に容易にできる。この現データの設計や、生成のやり直しを必要としない。

フィールドの更新に際して必要なデータ・フォーマットの交換は、すべてADABASが行なう。このときユーザは、更新すべきフィールドだけを指定すればよい。他のすべてのフィールドは、もとのままである。更新によるインバーテッド・リストへの影響は、ADABASが自動的に処理する。

3.4 ADD RECORD コマンド

ADD コマンドは、既存のファイルに新しいレコードを追加するとき使用する。

ユーザは、新レコードを生成するとき、元のすべてのフィールドに値を指定しなくてもよい。現在設定すべき値のみを与えればよい。値の存在しないフィールドは、ADABASによって空フィールドであることが記録され、元のフィールドは圧縮される。レコードを追加すると、関係するインバーテッド・リストを更新することが必要になる。ADABASは、これをすべて自動的に処理する。

もしもユーザが、相当量のレコードを1回のランで既存のファイルに追加したいのであれば、ADD RECORD コマンドの代わりに、ファイルをロードする速さでレコードを追加する機能を持つADABASユーティリティを使用するとよい。

3.5 DELETE コマンド

DELETE コマンドは、ファイルから1個の論理レコードを削除する機能を持つ。レコードの削除の結果、空きスペースとなったアソシエータ・ストレージ領域が直ちに再使用可能な状態になる。この結果、無駄な空きスペースが生ずることがないので、相当量の外部記憶域を節約できる。

3.6 OPEN/CLOSE コマント

OPENとCLOSEの両コマンドは、アプリケーションのデータベースに対する論理的な開始と終了を示すために使用する。ADABASのデータ検索保護の機能を使用しているとき、OPEN コマンドは、必要な検索保護の情報を含

んではないかならない。

3.7 CHECK POINT コマンド

ユーザは、CHECK POINT コマンドを用いて、データベースの更新プログラムの実行中に論理的4エック・ポイントを取ることが出来る。

これらの4エック・ポイントでの状態は、ADABASデータ保護ファイルに書き込まれ、その後、ユーザが4エック・ポイントでの状態を回復して再実行するとともに使用する。

さらに4エック・ポイントの同期機能もあり、ユーザ・プログラムが、バックモードや、オンライン・モードの一方あるいは両方が同時に実行されるとともに使用できる。

4. ADABAS ユーティリティ

ADABAS ユーティリティとは、特定のデータベース機能を遂行するために特に設計された一連のルーチン群のことである。もっとも重要なものとして、以下の機能がある。

- ・ データベースへの新ファイルのロード機能
- ・ データベースからファイルを削除する機能
- ・ ファイルのバックアップ機能とその解除機能
- ・ ファイル中に新しいフィールドを定義する機能
- ・ あるフィールドをキー・フィールドに交換する機能とそれを解消する機能
- ・ ファイルの論理的リオーガリング機能
- ・ データベースのセーブ、リストアの機能
- ・ 特定の4エック・ポイントの状態に回復する機能
- ・ データ構造情報を生成し、それを附帯する機能
- ・ データベースの物理的拡張機能
- ・ データベースの状態を報告する機能

以上のことに付いて、以下に説明する。

4.1 ファイルの生成と削除

4.1.1 ファイルの生成

ファイルをADABASデータベースにロードするためには、3つの段階を踏む必要がある。その3つの段階とは、

- (1) ロードするファイル中のフィールドの論理的構造を定義する。
- (2) 生データの編集と圧縮
- (3) 圧縮されたデータを、データベースのデータ・ストレージ部分にロードし、ファイル中のすべてのキー・フィールドに対して、アソシエータ内に、インバーテクトリストを生成する。

ファイル定義は、データ定義言語を用いて容易にできる。データの編集と圧縮、データベースのアソシエータデータ・ストレージの両方のロードは、ADABAS L0ADER ルーチンで行なう。

もしもユーザが特別のデータ編集を望むならば、編集、圧縮を行なうルーチンの実行中に、OWN・コード(ユーザ・エグゼキューット)を介してそれを行なうことができる。

4.1.2 ファイルの削除

ファイルの削除は、データベース中のファイルを、データベースから論理的に

削除することによって行なう。このファイルが使用していたデータストレージとアソシエータの領域は、これ以降、再使用可能である。

4.2 ファイルのカッポリングとアンカッポリング

4.2.1 ファイルのカッポリング

ユーザデータベース内の2つのファイル間に論理的関係を設定するときには、カッポリング処理を行なうADABAS 2ユーティリティを使用する。

ユーザは、カッポリングするファイルと、カッポリング条件とによって使用される共通のキー・フィールドを指定する。

カッポリング処理の結果として、アソシエータ内に2つの内部インバーテッド・リスト（各ファイルに対して1つずつ）が生成される。

4.2.2 ファイルのアンカッポリング

2つのファイル間のカッポリング関係を解消するときには、アンカッポリングユーティリティを実行する。

4.3 新しいフィールドの定義

ADABAS 2ユーティリティの中には、既存のファイル内に新しいフィールドを定義する機能がある。

この機能の便利さを、以前のResident-Fileの例を用いて具体的に示そう。

ファイルのロード後に郵便番号実行計画が与えられる。そのためには、各所有権レコードに、郵便番号というフィールドを設定する必要がある。

ADABASのフィールド定義機能を用いると、この新しいフィールドは、データの再構成とセットで設定できる。

郵便番号フィールドの値自体は、ADABASのUPDATE コマンドによって各レコードに与えられる。

4.4 キー・フィールドの宣言と解除

4.4.1 キー・フィールドの宣言

ADABAS 2ユーティリティの中には、既存のフィールドとキー・フィールドに格納できる機能が存在する。この2ユーティリティは、アソシエータで新たに必要となった情報を自動的に追加する。この2ユーティリティの実行後は、そのフィールドを、既存のキー・フィールドとまったく同様に使用できる。

4.4.2 キー・フィールドの解除

キー・フィールドを非キー・フィールドに格納できる機能が、ADABAS 2ユーティリティ中に存在する。そのフィールドがキー状態にあることに関するアソシエータ内のすべてのエントリは、削除される。これらのエントリが属していた領域は、必要に応じて再使用される。

4.5 ファイルのリオーガリング

ADABAS 2ユーティリティを用いて、ファイルのデータ・ストレージ内の位置を物理的に変えることができる。これは、データ・レコードの物理的順序と論理的順序をより良く一致させるために行う。これによってそのファイルの論理的逐次処理に要する時間の短縮が可能となる。

この手続きでは、そのファイルに対するインバーテッド・インデックスは、なおも維持される。そのファイルに対するアドレス・コンバータ（ISNとデイスクアドレスの対応テーブル）

データベースの他の部分には、この手続きによって影響されない。

4.6 データベース保護

ADABAS ユーティリティの中には、データベース保護に用いられる機能がある。

- ・ 磁気テープ上に、データベース全体又は個別ファイルのバックアップ用のコピーを生成する機能
- ・ 上のバックアップ用テープを用いて、データベース全体又はその一部をリストアップする機能
- ・ 特定のチェックポイントの前後で、ファイル又はファイル群に行った論理的更新を、再度実行したりリストアップしたりする機能

4.7 データベース機密保護

ADABAS ユーティリティの中には、機密保護に必要な種々のパスワードやキーカールを生成し維持する機能がある。

4.8 データベースの拡張

アプリケーションとデータ、ストレージに物理的ストレージを新たに割り当て、データベース全体の大きさを拡張する必要があるとき、ADABAS ユーティリティの一つがそれを行う。

この機能があるので、データベースを必要に応じて徐々に拡張していくことができる。データベースの再ロードを行なう必要もない。

4.9 データベースのステータス レポート

ADABAS ユーティリティの1つによって、データベースの使用状態を定期的に出力することができる。

このデータベースの使用状態レポートの情報は、次の通りである。

- ・ アプリケーションとデータ ストレージのための未使用領域
- ・ チェックポイント / リスタート情報
- ・ ファイル カットオフ情報

データベース ステータス レポートは、データベース全体を随時得るもの (DBA) によって、非常に便利なものである。