

1. 序 論

(1) 研究の背景

近年の河川管理においては、住民環境の整備や生態系保全、多自然型川づくりなど、きめ細かな対応が求められる。そのためには、河川に関する多様な分野でのデータベース整備が不可欠である。ところが地方自治体が管理するいわゆる中小河川については、いくつかの問題がある。河川の数が多いことや自治体の予算が少なく、必要なデータ項目も多様になってきただけに、河川管理を効率化する必要がありながら、調査およびメンテナンスがままならないのが現状である。

今日、コンピューター技術や観測技術の進歩、また流域の水環境を改善の社会的要請を背景にして、水理・水文分野の研究内容や解析手法は急速に発展・進化しつつある。特に、流域や河川における人為的な影響が水循環や生態系に与える影響を評価したり、適切な対策を立案する上で、信頼性の高い観測データ整備や解析手法の開発が強く求められている。しかし、観測データの質・量を向上させることや新たな解析技術の開発は容易ではなく、多くの研究グループの協力を不可欠とするような研究状況が生まれつつある。

一方、河川に関わる情報は、そのほとんどが位置を持つ空間データであることから、それを効率的に管理するGIS（地理情報システム）技術を導入することが有効である。GISを利用することにより、さらに多くの情報を自由に比較・分析したり、その結果を客観的に示すことが容易にできるようになり、今後河川環境整備に求められる総合的な評価や論理的な事業説明のための強力な手段を提供することが可能になる。GISを利用するメリットとして位置をディスプレイ上の地図で視覚的にとらえることができ、その上で地図上に表示された地点の情報を即座に見ることができるといふ点がある。また、さまざまな分野のデータを重ね合わせて表示することにより、より総合的な認識が可能になる。

このようなシステムは、従来はかなり高価なもので、しかも少数の専門家の手により運用ならびにデータ分析が行われてきた。しかし近年、需要の増大や普及分野の拡大により、そこそこ入手しやすい価格で、きちんとした機能を有するシステムも提供されるようになり、幅広い層でのGIS活用の機運が高まっている。

また最近のインターネットの普及速度には目を見張るものがある。インターネットの普及には、Webブラウザの進歩とWebページのコンテンツの充実が大いに貢献している。

Webブラウザは単にWebページのコンテンツを「静的に表示」するのではなく、さまざまなスクリプトやアプレットの活用により、Webブラウザ自身が動画表示したりユーザーの入力に応答したり「動的な表示」も行えるようになってきた。

Webページのコンテンツに関しても、たとえば製品情報を単純にhtmlファイルの中に書いておく（つまりそのファイルはそれ以上変更されない、「静的に生成」されたもの）だけでなく、製品情報をデータベース化し、そこで日々更新されるデータをもとに自動的にWebページを発信すること（つまり「動的な生成」）が当たり前のようになった。

そのインターネット技術の発展のなかで、プログラミング言語のひとつであるJavaが急速な進化・発展を遂げている。Java言語はネットワーク分散処理を駆使するエンタープライズ仕様の大規模システムから携帯電話などで動作するマイクロア

アプリケーションまで、さまざまな階層でのプログラミングに対応している。また各種関数計算も標準で実装しており、科学技術計算も可能である。さらにオブジェクト指向プログラミングをサポートしているため、各種作業の部品化や組み合わせ、そしてメンテナンスが容易になっている。

こうした特長は、野外調査、データベース構築、データ分析、技術計算、Webページの作成など、数多くの分野でJava言語を活用できることを示唆している。しかもGIS関連のプログラミングをサポートするJava言語ライブラリも市場にそろい始めている。Java言語を活用し、Web - GISを前提とした河川データベースを構築する条件は、かなり整ってきているといえる。

(2) 研究の目的と構成

本研究では、リレーショナルデータベース、GISならびにJava言語を活用して、管理が煩雑な中小河川の環境整備をより効率的効果的に支援するシステムを開発することを最終的な目的とする。そのシステムの特長には、以下のような項目を掲げたい。

GISにより多数の河川と周辺地域の情報を一元的階層的に管理・表示する。

インターネットにより作業の広域分散化と地域への公開を図る。

煩雑な多自然型改修業務を効率よく支援する。

Java言語を用いて業務を支援する各機能を部品化し、システムのメンテナンスや高機能化を容易にする。

ただし、上記のようなシステムの開発（調査によるデータ整備も含む）は、短期間に可能である訳ではない。そこで今回はシステム開発の手始めとして、以下のような事項を手がけることにする。

栃木県の中小河川をケーススタディの対象とし、現段階で収集可能な広域的データ（水系、河川、雨量観測所、流量観測所など）を電子化する。また、（全河川での調査は無理なので）1つの河川のある区間を選んで、その断面形状の測量や植物・魚類調査を実施し、そのデータも電子化する。それらの電子化データを用いて、まずはスタンドアロンな環境で、リレーショナルデータベースを構築する（第2章）。

次にそのデータベース上のデータをGIS上で活用する方法を検討する。各データに関連する位置情報を特定し、それを地図上にポイント表示できるようにする。そしてユーザーが指示したポイントの情報を表示させるようにする。また各ポイントをクリックすればリレーショナルデータベース全体を参照できるようにGISからデータベース全体にリンクできるようにもする（第3章）。

さらにデータベースをWebページ上で表示するシステムを検討する。これにはJava言語のサーバーサイドプログラミング技術を活用する。すなわちサーブレットプログラムとJSP（Java Server Pages）である。ここでは、単にデータを数値として示すだけでなく、データをもとに画像を自動生成し、それをWebページ上に表示させる方法も検討する（第4章）。

最後に、開発上の留意点をまとめるとともに、本システムの将来の発展について検討を行う（第5章）。

2 . 中小河川データベースの構築

(1) データベースの概要

河川のような複雑な対象については、多種多様なデータが相互に関連している。そこで、一つのテーブル（表のようなもの）で全てをまかなうよりも、複数のテーブルを作成して、相互の関連性（リレーション）を設定する方が都合がよい。リレーショナルデータベースでは、複数の表を組み合わせることができる。また、複数の表を組み合わせるには、それぞれの表の関係を設定する必要がある。各表に一つ特有のIDをもたせ、関係づけたい表同士に同じID項目を作成し、これらを結びつけることにより、表同士の関係が設定される。この関係をリレーションシップとよぶ。このリレーションによって、各テーブルからそれぞれ必要な項目のみを取り出して新しい表を作成することが可能になる。さらに、テーブルを表として参照するだけでなく、表中のデータ情報を見やすく表示するフォームを作成すると、作業効率が向上する。そこで今回はMicrosoft社のリレーショナルデータベースソフトACCESSを利用し、リレーショナルデータベースを構築する。

このデータベースを作成するにあたって、まずデータベースアプリケーションの構成を決める必要がある。まずテーブルをどういう項目別に分けるかを決め、次に各テーブル内に入れる内容を検討していった。さらに項目別に分けたテーブルをそれぞれ階層的に分類していくことによって関連付けを行いやすくした。またデータの検索、抽出、整理をより扱いやすいように、クエリやフォームを活用することにした。

(2) データ項目の分類

データベースのデータ項目及び入力データを表 - 1 のように、水系、河川、断面別に項目分類する。水系、河川、雨量観測所、流量観測所については栃木県の中小河川全体を対象にし、断面については姿川一河川のみを対象にする。データ項目および入力データをみてもわかるように、中小河川データベースは、データ項目のみならず、取得機関も多岐に渡る。また、その多くが電子データファイル化されておらず、新たに電子データ入力する煩雑さが伴う。

表 - 1 データ項目および入力データ

データ項目		入力データ出典
水系	水系名	栃木県土木部河川課管理河川一覧
	流域面積	栃木県土木部河川管理課河川一覧
河川	河川名	栃木県土木部河川課管理河川一覧
	水系名	栃木県土木部河川課管理河川一覧
	流域面積	栃木県土木部河川課管理河川一覧
	管理者	栃木県土木部河川課管理河川一覧
	合流河川	栃木県土木部河川課管理河川一覧
雨量観測所	観測所名	栃木県土木部河川課雨量年表
	位置座標	栃木県土木部河川課雨量観測所諸元一覧
	所在地	栃木県土木部河川課雨量年表
	関係河川	栃木県土木部河川課雨量年表
	雨量	栃木県土木部河川課雨量年表
	年月日	栃木県土木部河川課雨量年表
流量観測所	観測所名	栃木県土木部河川課流量年表
	位置座標	栃木県土木部河川課流量観測所諸元一覧
	所在地	栃木県土木部河川課流量年表
	流量	栃木県土木部河川課流量年表
	年月日	栃木県土木部河川課流量年表
断面	断面名	宇都宮土木事務所河川砂防部資料
	位置座標	宇都宮土木事務所河川砂防部資料
	断面地形	栃木県土木部河川課姿川根古屋橋上流部河川概況調査資料
	河床材料	栃木県土木部河川課姿川根古屋橋上流部河川概況調査資料
	植生状況	栃木県土木部河川課姿川根古屋橋上流部自然環境調査資料
	魚類	栃木県土木部河川課姿川根古屋橋上流部自然環境調査資料
	流下能力	栃木県土木部河川課資料

(3) テーブルの構成とリレーションの設定

中小河川でのデータベースにおいて、まずデータベースの構成を決めた。テーブルを水系、河川、断面、と分け河川から雨量観測所、流量観測所、断面から断面地形、河床材料、流下能力、植生、魚類、とそれぞれテーブル分けをし、階層的に分けて作成していく。

テーブル作成し、それぞれ関連づけさせることにより、どのテーブルからでも必要なデータを検索、抽出することができるようにする。そこで、水系基本、河川基本、雨量観測所、月雨量、日雨量、流量観測所、月流量、日流量、断面基本、断面地形、河床材料、植生、植生詳細、魚類、魚類詳細、流下能力の各テーブルに、それぞれ水系ID、河川ID、雨量観測所ID、月雨量ID、日雨量ID、流量観測所ID、月流量ID、日流量ID、断面ID、断面地形ID、河床材料ID、植生ID、植生詳細ID、魚類ID、魚類詳細ID、流下能力IDとして固有のIDを設ける。リレーションさせたいテーブル同士に同じID名の項目を作成し、そのID同士を結びつけることにより、リレーションさせるようにする。リレーションさせたテーブルは関連付けされたテーブル内のどこのテーブルからほしい項目だけを自由に組み合わせて新しく表をつくることができる。テーブル構成を図 - 1 に示す。

それぞれのテーブル内の項目は次のようにしている。

水系基本テーブル：水系ID、水系名、流域面積

河川基本テーブル：河川ID、水系ID、河川名、流域面積、管理者、合流河川

雨量観測所テーブル：雨量観測所ID、雨量観測所名、所在地、緯度、経度、河川ID

月雨量テーブル：月雨量ID、雨量観測所ID、1月、2月・・・12月

日雨量テーブル：日雨量ID、雨量観測所ID、月、1日、2日・・・31日

流量観測所テーブル：流量観測所ID、流量観測所名、所在地、緯度、経度、河川ID

月流量テーブル：月流量ID、流量観測所ID、1月、2月・・・12月

日流量テーブル：日流量ID、流量観測所ID、月、1日、2日・・・31日

断面基本テーブル：断面ID、河川ID、断面名、X、Y

断面地形テーブル：断面地形ID、Y'、Z、調査日、断面ID

河床材料テーブル：河床材料ID、Y'、河床材料、代表粒径、調査日、断面ID

流下能力テーブル：流下能力ID、開始断面ID、終了断面ID、勾配、護岸種類（左岸）護岸種類（右岸）年度、

植生テーブル：植生ID、断面ID、方形枠、優占種1、優占種2、高さ、高さ、
植生率、植生率、種数、一年草比率、調査日、Y'

植生詳細テーブル：植生詳細ID、植生名、被度、群度、年草、帰化植物、断面ID

魚類テーブル：魚類ID、開始断面ID、終了断面ID、採捕数、採捕種数、採取方法、
調査日

魚類詳細テーブル：魚類詳細ID、魚類名、採捕数

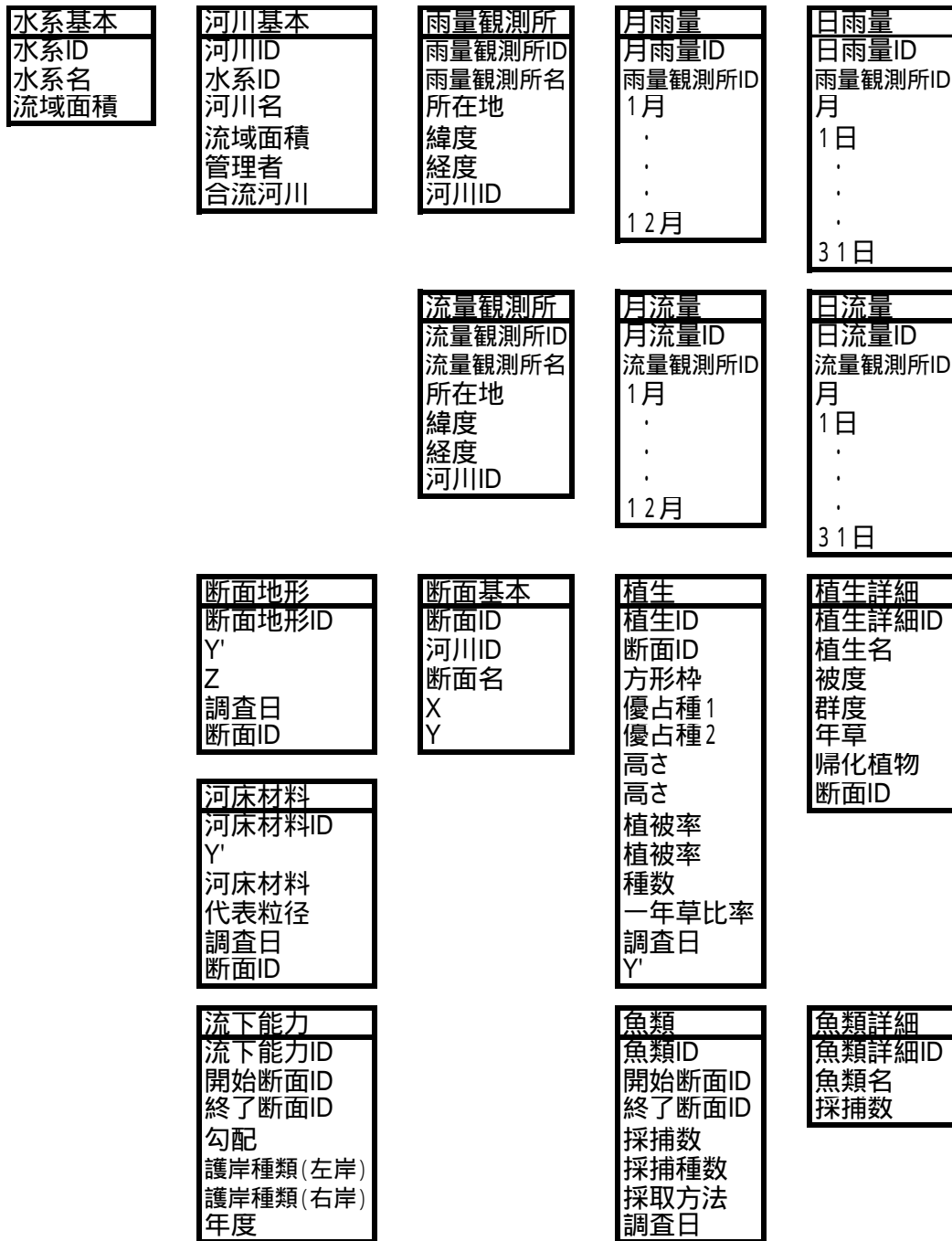


図 - 1 テーブル構成とリレーション

1) テーブルについて

テーブルとはそれぞれ一つずつが一つの表のようなものになっており、データベースでもっとも重要なデータの保存を担当するデータベースオブジェクトで、データは全てテーブルに保存される。テーブルのデータシートビュー（次頁以降参照）の外観は、表計算ソフトのワークシートに似ている。ワークシートとの大きな違いは、ワークシートが基本的にすべてのデータをひとつの表で管理するのに対し、テーブルは、情報の種類ごとに表に分けて管理する点にある。このように情報を管理することにより、クエリ（(4) クエリについて 参照）を利用して、目的に応じた表を自由に作成することができるようになる。

テーブルはレコードの集まりで、レコードとは一件分のデータのことで、「水系ID」「水系名」などのフィールドとよばれる項目で構成されている。またテーブルにはデザインビューとデータシートビューの2とおりの表示方法がある。デザインビューでは、フィールドの追加や削除、データ型の設定など、テーブルの設計を行う。データシートビューでは、データの入力や印刷など、データに関する操作を行う。

本研究では、水系基本テーブル、河川基本テーブル、雨量観測所テーブル、月雨量テーブル、日雨量テーブル、流量観測所テーブル、月流量テーブル、日流量テーブル、断面基本テーブル、断面地形テーブル、河床材料テーブル、流下能力テーブル、植生テーブル、植生詳細テーブル、魚類テーブル、魚類詳細テーブルをそれぞれ作成していった。このように階層的にテーブル分けをすることにより、テーブル間での関連付けやデータの検索、抽出をしやすい扱いやすくし、また河川環境整備についてより利用しやすい方向でデータベースを作成した。

2) 水系テーブル

水系基本テーブル

項目を水系ID、水系名、流域面積とする。水系基本テーブルを表 - 2 に示す。

表 - 2 水系基本テーブル

水系ID	水系名	流域面積
1	利根川	3728.1
2	那珂川	2100.7
3	久慈川	28.26

また、テーブル間でリレーションをしているので、項目の左をクリックすると関連させたテーブル内からその項目に該当する内容のみの表をサブテーブルとして表示することができる。水系ID2の那珂川をクリックした時のサブテーブルの一部分を表 - 3 に示す。

表 - 3 水系基本テーブルとサブテーブル

水系基本テーブル						
水系ID	水系名	流域面積				
1	利根川	3728.1				
2	那珂川	2100.7				
河川ID	河川名	流域面積	管理者	合流河川		
153	那珂川	745	栃木県			
154	逆川	114.4	栃木県	那珂川		
155	鮎田川	24.9	栃木県	逆川		
156	高田川	4.6	栃木県	鮎田川		
157	塩田川	7.3	栃木県	逆川		
158	神井川	4.9	栃木県	逆川		
159	坂井川	13	栃木県	逆川		
160	菅又川	5.4	栃木県	坂井川		
161	深沢川	14.9	栃木県	逆川		
162	八反田川	14.2	栃木県	那珂川		
163	木須川	29.8	栃木県	那珂川		
164	小木須川	8.5	栃木県	木須川		
165	解石川	2.2	栃木県	那珂川		
166	荒川	427.6	栃木県	那珂川		
167	江川(鳥山)	91.3	栃木県	荒川		
168	岩川	30.1	栃木県	江川(鳥山)		
169	西江川	7.4	栃木県	江川(鳥山)		
170	長者川	8.7	栃木県	荒川		
171	内川	145.1	栃木県	荒川		
172	江川(矢板)	10.7	栃木県	内川		
173	こぶし台川		栃木県	江川(矢板)		
174	前田川		栃木県	江川(矢板)		
175	後沢川		栃木県	江川(矢板)		
176	宮川	36.3	栃木県	内川		

3) 河川テーブル

河川基本テーブル

項目を河川ID、水系ID、河川名、流域面積、管理者、合流河川にする。このうち水系IDは水系基本テーブルの水系IDとリレーションさせる。表 - 4 に河川テーブルの一部を示す。

表 - 4 河川基本テーブル

河川ID	水系ID	河川名	流域面積	管理者	合流河川
1	1	小貝川	162.2	栃木県	利根川
2	1	五行川	225.3	栃木県	小貝川
3	1	江川(真)	34.5	栃木県	五行川
4	1	行屋川	8	栃木県	五行川
5	1	野元川	59.4	栃木県	五行川
6	1	井沼川	26.3	栃木県	五行川
7	1	大沼川	7.7	栃木県	井沼川
8	1	冷子川	11.2	栃木県	井沼川
9	1	ぐみ川	11	栃木県	小貝川
10	1	百目鬼川	9.3	栃木県	小貝川
11	1	百目鬼川放水路	0	栃木県	百目鬼川
12	1	大羽川	24.2	栃木県	小貝川
13	1	小宅川	22.7	栃木県	小貝川
14	1	大川	27.5	栃木県	小貝川
15	1	桜川	17.5	栃木県	小貝川
16	1	続谷川	3.8	栃木県	小貝川
17	1	鬼怒川	1147.2	栃木県	利根川
18	1	田川	231.3	栃木県	鬼怒川
19	1	武名瀬川		栃木県	田川
20	1	釜川	5.3	栃木県	田川
21	1	御用川	19.3	栃木県	田川
22	1	釜川放水路	2.3	栃木県	田川
23	1	山田川	41.6	栃木県	田川
24	1	前川	3.5	栃木県	山田川
25	1	逆川(宇)	13.4	栃木県	田川

また、河川基本テーブルもいくつかのテーブルとのリレーションが設定されているので、水系基本テーブル同様、表中の左をクリックすれば他のテーブルからその項目に関連しているものみのテーブルをサブテーブルとして表示することができる。このとき、表 - 5 でわかるようにリレーションされているテーブルのなかから、サブテーブルとして表示させたいテーブルを選ぶことができる。これ以降にでてくるすべてのテーブルはリレーション設定されているので、サブテーブルを表示することができる。

表 - 5 河川基本テーブルとサブテーブル

河川ID		サブデータシートの挿入		合流河川
+	1			利根川
+	2			小貝川
+	3			五行川
▶	4			五行川
+	5			五行川
+	6			五行川
+	7			井沼川
+	8			井沼川
+	9			小貝川
+	10			小貝川
+	11			百日鬼川
+	12			小貝川
+	13			小貝川
+	14			小貝川
+	15			小貝川
+	16			小貝川
+	17			利根川
+	18			鬼怒川
+	19			田川
+	20			田川
+	21			田川
+	22	1 釜川放水路	2.3 栃木県	田川
+	23	1 山田川	41.6 栃木県	田川
+	24	1 前川	3.5 栃木県	山田川
+	25	1 逆川(宇)	13.4 栃木県	田川
+	26	1 寅己川	3.1 栃木県	逆川(宇)
+	27	1 赤堀川	21.9 栃木県	田川
+	28	1 弁天川	栃木県	赤堀川
+	29	1 江川(宇)	48.7 栃木県	鬼怒川

4) 雨量観測所テーブル

雨量観測所テーブル

項目を雨量観測所ID、雨量観測所名、所在地、緯度、経度、河川IDとする。
河川IDを河川基本テーブルの河川IDとリレーションさせる。雨量観測所テーブルの一部を表 - 6 に示す。

表 - 6 雨量観測所テーブル

雨量観測所ID	雨量観測所名	所在地	緯度	経度	河川ID
1	真岡	真岡市荒町1, 171 - 4	36° 26' 11"	140° 1' 5"	2
2	氏家	塩谷郡氏家町氏家2, 771	36° 40' 20"	139° 57' 35"	2
3	日光	日光市萩垣面2, 390 - 7	36° 45' 2"	139° 37' 2"	38
4	千手ヶ浜	日光市中宮祠千手2, 573 - 2	36° 44' 27"	139° 25' 25"	55
5	金精	日光市湯元	36° 48' 44"	139° 24' 15"	272
6	宇都宮	宇都宮市竹林1, 030 - 2	36° 34' 25"	139° 53' 52"	18
7	小山	小山市花垣町1 - 13 - 40	36° 21' 26"	139° 49' 45"	84
8	栃木	栃木市神田町6 - 6	36° 17' 11"	139° 44' 48"	106
9	秋山台	安蘇郡葛生町秋山	36° 30' 14"	139° 31' 30"	119
10	鹿沼	鹿沼市今宮町1, 664 - 1	36° 33' 43"	139° 44' 49"	257
11	古峰原	鹿沼市草久	36° 38' 47"	139° 32' 22"	99
12	佐野	佐野市堀米町607	36° 19' 44"	139° 35' 6"	119
13	足利	足利市伊勢町4 - 19	36° 19' 28"	139° 27' 47"	133
14	粟野	上都賀郡粟野町口粟野	36° 30' 44"	139° 40' 36"	18
15	川中島	鹿沼市草久字川中島	36° 39' 47"	139° 34' 2"	101
16	大滝	鹿沼市草久字大滝	36° 41' 25"	139° 33' 52"	101
17	松田川ダム	足利市松田町	36° 25' 26"	139° 25' 44"	139
18	西荒川ダム	塩谷郡塩谷町上寺島	36° 48' 52"	139° 48' 46"	185
19	高原	塩谷郡藤原町たて原	36° 51' 17"	139° 45' 56"	185
20	矢板	矢板市鹿島町20 - 11	36° 47' 30"	139° 54' 30"	171
21	茂木	芳賀郡茂木町茂木131	36° 31' 46"	140° 12' 4"	154
22	大田原	大田原市紫塚2 - 2, 564 - 1	36° 52' 15"	140° 1' 30"	219
23	新湯	那須郡塩原町新湯	36° 56' 51"	139° 47' 23"	215
24	馬頭	那須郡馬頭町建武1, 571	36° 44' 13"	140° 10' 42"	76
25	弓張	矢板市長井弓張	36° 51' 50"	139° 50' 2"	176

月流量テーブル

項目を月流量ID、流量観測所ID、1月から12月までの各月、年度とする。このうち流量観測所IDは流量観測所テーブル中の流量観測所IDとリレーションさせる。月流量テーブルの一部を表 - 10に示す。

表 - 10 月流量テーブル

月流量ID	流量観測所ID	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月	年度
1	1	0.68	0.71	2.25	5.32	4.52	4.68	15.75	3.48	4.78	2.68	2.5	0.08	1999
2	2	2.43	1.98	2.99	4.56	8.02	12.54	20.39	8.66	10.42	6.29	5.28	2.71	1999
3	3	0.06	0.06	0.08	0.29	0.9	0.93	0.97	1.07	0.32	0.2	0.19	0.09	1999
4	4	1.17	1.05	1.19	3.04	7.08	9.36	11.75	6.8	4.22	2.53	2.57	1.99	1999
5	5	1.38	1.32	1.27	1.85	5.51	8.26	10.36	7.41	3.9	2.72	2.63	1.86	1999
6	6	5.08	5.09	6.41	7.01	8.01	6.14	16.24	14.25	17.86	11.61	9.8	6.52	1999
7	7	0.5	0.41	0.44	1.33	3.99	4.71	5.18	5.34	2.71	1.31	0.99	0.71	1999
8	8	0.44	0.54	0.39	1	3.13	6.92	7.97	2.8	2.45	1.45	1.36	1.31	1999
9	9	3.59	4.84	4.63	7.15	3.84	3.93	21.19	23.04	9.31	4.7	2.88	1.8	1999
10	10	12.45	11.83	11.75	11.85	11.39	11.26	19.61	19.86	13.4	12.62	11.81	11.44	1999
11	11	3	2.75	2.88	3.39	3.87	3.69	4.68	4.73	4.18	3.65	3.3	2.29	1999
12	12	0.26	0.22	0.27	0.38	0.62	0.56	1.09	0.72	0.84	0.58	0.47	0.27	1999
13	13	1.13	1.05	1.23	7.02					3.66	2.87	2.61	2.74	1999
14	14	11.6	12.77	14.11	17.41	14.7	13.78	24.99	16.62	15.85	8.39	6.23	4.87	1999
15	15	0.1	0.24	0.45	1.28	1.17	1.25	7.95	0.92	1.01	0.56	0.6	0.29	1999
16	16			4.71	15.43	13.18	47.66	264.7	107	39.63	18.03	12.4	10.63	1999
17	17	0.99	0.01		11.12	11.31	28.71	182.8	45.19	19.48	9.53	6.36	2.45	1999
18	18	0.36	0.24	1.67	5.5	5.37	3.95	22.71	15.43	6.63	3.43	2.31	0.81	1999
19	19	2.79	2.25	2.3	5.11	6.66	12.42	39.36	9.08	14.27	7.14	5.66	2.91	1999
20	20	0.54	0.45	0.91	3.5	3.63	3.2	16.18	3.88	6.49	3.88	3.05	2.4	1999
21	21	1.36	0.66	1.93	4.49		16.81	50.66	27.88	13.71	4.71	3.1	1.87	1999
22	22	1.39	0.71	0.98	5.82	6.16	3.74	36.68	26.01	8.92	4.08	2.64	1.12	1999
23	23	0.82	0.76	0.76	1.53	0.74	1.29	2.52	2.63	3.05	2.48	1.38	0.66	1999
24	24	0.33	0.02		3.86	4.04	4.52	15.55	5.51	3.56	3.26	2.15	0.9	1999
25	25	0.09	0.12	0.38	1.58	1.49	1.45	13.21	9.44	1.31	1.14	0.26	0.07	1999
26	26	0.45	0.23	0.17	2.12	3.29	1.66	9.1	6.04	3.03	2.12	0.56	0.44	1999
27	27				1.69	2.38	2.43	12.83	5.82	3.74	2.87	1.07	0.33	1999
28	28	0.1	0.09	0.14	0.38	0.25	2.55	4.44	0.89	0.9	0.44	0.14	0.04	1999

5) 流量観測所テーブル

流量観測所テーブル

項目を流量観測所ID、流量観測所名、所在地、緯度、経度、河川IDにする。
このうち河川IDを河川テーブル中の河川IDとリレーションさせる。流量観測所
テーブルの一部を表 - 9 に示す。

表 - 9 流量観測所テーブル

流量観測所ID	流量観測所名	所在地	緯度	経度	河川ID
1	権現	芳賀郡益子町上山	32°26'14"	140°4'35"	1
2	妹内橋	真岡市荒町	36°26'19"	140°1'5"	2
3	桜野	塩谷郡氏家町桜野	36°40'00"	140°0'0"	2
4	監物橋	塩谷郡芳賀町西水沼	36°31'3"	140°1'45"	5
5	八ツ木	塩谷郡芳賀町芳志戸	36°35'51"	140°2'49"	2
6	明治橋	河内郡上三川町下梁	36°25'30"	139°53'34"	18
7	道城橋	河内郡上三川町上郷	36°26'30"	139°55'48"	116
8	小林	今市市小林	36°41'50"	139°50'20"	33
9	今市	今市市瀬川町	36°42'0"	139°40'40"	38
10	神橋	日光市山内	36°45'0"	139°36'2"	38
11	菖蒲ヶ浜	日光市中宮祠	36°45'4"	139°27'10"	272
12	兜橋	宇都宮市戸祭鎧塚	36°34'18"	139°52'40"	20
13	大桑	今市市栗原	36°46'3"	139°43'0"	60
14	幸橋	宇都宮市千波町	36°33'39"	139°53'46"	18
15	佐下部	今市市佐下部	36°46'8"	139°41'54"	61
16	観昇橋	小山市中央町	36°7'0"	139°46'0"	18
17	保橋	栃木市柳原町	36°20'0"	139°40'0"	18
18	天満橋	上都賀郡粟野町口粟野	36°30'50"	139°41'8"	18
19	南半田	小山市飯塚	36°18'11"	139°48'42"	85
20	淀橋	宇都宮市幕田町	36°29'50"	139°50'10"	85
21	府中橋	鹿沼市府中町	36°33'56"	139°45'28"	257
22	御幣岩橋	鹿沼市上日向町	36°33'30"	139°42'45"	99
23	倭橋	栃木市倭町	36°6'0"	139°40'0"	106
24	下皆川	下都賀郡大平町川連	36°24'0"	139°42'0"	109

月雨量テーブル

項目を月雨量ID、雨量観測所ID、1月から12月の各月、年度とする。このうち雨量観測所IDは雨量観測所テーブルの雨量観測所IDとリレーションさせる。月雨量テーブルの一部を表 - 7 に示す。

表 - 7 月雨量テーブル

月雨量ID	雨量観測所ID	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月	年度
1	1	1	35	112	138	85	205	270	95	150	77	57	6	1999
2	2	4	27	48	123	131	243	284	249	132	74	47	6	1999
3	3	7.5	21.5	138	222	111.5	266	522	245	208.5	135.5	53	4.5	1999
4	4					147	246	334	375	223	115			1999
5	5					147	247	321	362	229	111			1999
6	6	2.5	26	118	127	110	228.5	196.5	133.5	189	76	34.5	4.5	1999
7	7	0	30	107	176	88	146	191	180	152	106	44	3	1999
8	8	1	28	125	173	108	202	202	182	168	88	42	3	1999
9	9	3	17	150	181	129	269	452	279	241	146	42	2	1999
10	10	4	23	129	137	128	235	190	194	195	40	0	38	1999
11	11	9	35	199	277	156	307	392	551	250	40	0	0	1999
12	12	2	23	104	142	82	131	189	181	135	91	36	2	1999
13	13	1	12.5	99	138	67	150.5	229.5	203	161.5	79	28	1	1999
14	14	4	22	145	164	126	247	187	215	157	42	0	3	1999
15	15	6	30	187	138.5	0	264	19	664.5	272	155.5	65	3	1999
16	16	7.5	30	183	142	164	327.5	568	678.5	265.5	173	69	7.5	1999
17	17	2	10	95	143	84	196	332	221	304	107	28	1	1999
18	18	13	25	120	169	151	252	460	380	163	70	39	8	1999
19	19	2	12	78	188	101	256	523	499	221	86	48	5	1999
20	20	9	25	53	111	133	231	303	312	142	65	43	8	1999
21	21	2	32	119	166	90	225	335	145	166	62	64	9	1999
22	22	9.5	24.5	101	80	18.5	247.5	334.5	223.5	171.6	59	47	7	1999
23	23	10	21	112	284	89	203	388	458	260	134	34	4	1999
24	24	3	37	101	106	102	153	328	176	184	63	53	9	1999
25	25	0	6	89	184	143	269	601	425	171	71	18	0	1999
26	26	4	28	108	131	121	207.5	385	98	207	79	61	10	1999
27	27	8	25	136	214	158	237	600	454	181	90	31	9	1999
28	28	1	14	77	124	104	239	536	386	207	91	49	11	1999

日雨量テーブル

項目を日雨量ID,雨量観測所ID、月、1日から31日までの各日にする。このうち雨量観測所IDを雨量観測所テーブルの雨量観測所IDとリレーションさせる。日雨量テーブルの一部を表 - 8 に示す。

表 - 8 日雨量テーブル

日雨量ID	雨量観測所ID	月	1日	2日	3日	4日	5日	6日	7日	8日	9日	10日	11日
1	1	1											
2	1	2		1									7
3	1	3							7		6		
4	1	4						2	1			13	22
5	1	5				38	15	1					4
6	1	6							3	1			25
7	1	7		2	10	22						14	12
8	1	8					2					2	
9	1	9	14						1			1	
10	1	10	1		1								
11	1	11	24	1									
12	1	12		6									
13	2	1											
14	2	2		1									
15	2	3							8		2		
16	2	4						8	1			10	12
17	2	5				65	21					1	9
18	2	6							1				1
19	2	7			8	26		2				19	8
20	2	8					3					5	
21	2	9	3										
22	2	10	1										
23	2	11	13	1	1								
24	2	12		5									
25	3	1											
26	3	2											
27	3	3					1		3.5		1	2.5	
28	3	4				0.5		19.5	2.5			18	37

6) 断面テーブル

断面基本テーブル

項目を断面ID、河川ID、断面名、X、Yとする。このうち河川IDは河川基本テーブルの河川IDとリレーションさせる。断面基本テーブルの一部を表 - 12に示す。

表 - 12 断面テーブル

断面ID	河川ID	断面名	X	Y
1	85224+00		58531.532	611.0885
2	85224+25		58553.859	622.3352
3	85224+50		58576.187	633.5818
4	85224+75		58598.582	644.6899
5	85225+00		58621.775	653.9936
6	85225+25		58645.781	660.9354
7	85225+50		58670.36	665.446
8	85225+75		58695.267	667.4802
9	85226+00		58720.252	667.0177
10	85226+25		58745.067	664.0631
11	85226+50		58769.462	658.646
12	85226+75		58793.195	650.8205
13	85227+00		58816.028	640.6648
14	85227+25		58837.732	628.2803
15	85227+50		58858.093	613.7909
16	85227+75		58876.904	597.3411
17	85228+00		58894.011	579.1227
18	85228+25		58910.506	560.336
19	85228+50		58927	541.5492
20	85228+75		58943.494	522.7624
21	85229+00		58959.988	503.9757
22	85229+25		58976.482	485.1889
23	85229+50		58992.976	466.4021
24	85229+75		59009.471	447.6154
25	85230+00		59025.965	428.8286

断面地形テーブル

項目を断面地形ID、Y'、Z、調査日、断面IDとする。このうち断面IDは断面基本テーブルの断面IDとリレーションさせる。ここで、Y'は左岸の堤防肩からの横断距離、またZは標高を意味する。断面地形テーブルの一部を表 - 13に示す。

表 - 13 断面地形テーブル

断面地形ID	Y'	Z	調査日	断面ID
1	0	101.558	08-Sep-00	5
2	4.55	98.941	08-Sep-00	5
3	6.7	99.191	08-Sep-00	5
4	9.3	99.646	08-Sep-00	5
5	13	97.776	08-Sep-00	5
6	13.1	97.616	08-Sep-00	5
7	15.7	97.368	08-Sep-00	5
8	33.6	97.168	08-Sep-00	5
9	33.7	97.296	08-Sep-00	5
10	38.1	99.036	08-Sep-00	5
11	39.5	99.981	08-Sep-00	5
12	40.3	98.916	08-Sep-00	5
13	44.1	100.723	08-Sep-00	5
14	0	102.664	08-Sep-00	6
15	4.7	100.674	08-Sep-00	6
16	6.3	100.656	08-Sep-00	6
17	9.3	100.559	08-Sep-00	6
18	13.9	98.764	08-Sep-00	6
19	14	98.621	08-Sep-00	6
20	17	98.226	08-Sep-00	6
21	30.4	98.139	08-Sep-00	6
22	31.8	98.334	08-Sep-00	6
23	38	99.879	08-Sep-00	6
24	40	99.819	08-Sep-00	6
25	40.8	99.814	08-Sep-00	6
26	44.9	101.579	08-Sep-00	6
27	0	102.959	08-Sep-00	7
28	4.7	101.014	08-Sep-00	7

河床材料テーブル

項目を河床材料ID、Y'、河床材料、代表粒径、断面IDとする。このうち断面IDは断面基本テーブルの断面IDとリレーションさせる。またY'は断面地形テーブル同様、左岸堤防肩からの横断距離を意味する。河床材料テーブルの一部を表 - 14に示す。

表 - 14 河床材料テーブル

河床材料ID	Y'	河床材料	代表粒径 (cm)	調査日	断面ID
1	13	玉石	10~ 20	05-Sep-00	5
2	15.7	玉石	10~ 20	05-Sep-00	5
3	33.7	ヘトコ		05-Sep-00	5
4	13.9	玉石	10~ 20	05-Sep-00	6
5	17	玉石	10~ 20	05-Sep-00	6
6	31.8	ヘトコ		05-Sep-00	6
7	15.9	玉石	10~ 20	05-Sep-00	7
8	18.9	玉石	20~ 40	05-Sep-00	7
9	30.4	ヘトコ		05-Sep-00	7
10	15.6	粗礫	5~ 10	05-Sep-00	8
11	19.6	砂		05-Sep-00	8
12	26.4	ヘトコ		05-Sep-00	8
13	12.75	玉石	10~ 20	05-Sep-00	9
14	18	玉石	20~ 40	05-Sep-00	9
15	23.5	ヘトコ		05-Sep-00	9
16	11.8	粗礫	5~ 10	05-Sep-00	10
17	17.6	玉石	10~ 20	05-Sep-00	10
18	23.7	粗礫	5~ 10	05-Sep-00	10
19	10.8	玉石	10~ 20	05-Sep-00	12
20	19.8	玉石	10~ 20	05-Sep-00	12
21	35.2	玉石	10~ 20	05-Sep-00	12
22	28.7	玉石	10~ 20	05-Sep-00	17
23	32.1	玉石	10~ 20	05-Sep-00	17
24	33.6	玉石	10~ 20	05-Sep-00	17
25	29.9	玉石	10~ 20	05-Sep-00	18
26	35.6	玉石	10~ 20	05-Sep-00	18
27	39.6	玉石	10~ 20	05-Sep-00	18

植生テーブル

項目を植生調査ID、断面ID、方形枠、優占種1、優占種2、高さ、高さ、
 植被率、植被率、種数、一年草比率、調査月日、Y'とする。このうち断面
 IDは断面基本テーブル中の断面IDとリレーションさせる。植生テーブルの一部
 を表 - 15に示す。

表 - 15 植生テーブル

植生調査ID	断面ID	方形枠	優占種1	優占種2	高さ (m)	高さ (m)	植被率 (%)
1	27	2×2	ミノバ	イヌビエ	0.6	0	5
2	27	2×2	メヒシバ		1.1	0	80
3	27	2×2	アメリカセンダングサ		1.2	0	30
4	27	2×2	メヒシバ		1.3	0	95
5	27	2×2	メヒシバ	シバ	1.5	0.1	80
6	26	2×2	オオクサキビ	メヒシバ	0.8	0	10
7	26	2×2	ヌカキビ		1.2	0	95
8	26	2×2	メヒシバ		1.3	0	95
9	26	2×2	メヒシバ		1.4	0	100
10	26	2×2	メヒシバ	シバ	1.9	0.1	60
11	19	2×2	メヒシバ		0.7	0	5
12	27	2×2	ケイヌビエ		1	0	30
13	27	2×2	メヒシバ	ヒメムカシヨモギ	1.5	0	90
14	27	2×2	メトハギ		1.5	0.5	2
15	27	2×2	オオアレチノギク		2.5	0	100
16	27	2×2	ヨモギ	シバ	2	1	30
17	26	2×2	ケイヌビエ	ヌカキビ	1	0	70
18	26	2×2	コブナグサ		1.5	0	90
19	26	2×2	ヌカキビ	ツルマメ	1	0	80
20	26	2×2	カナムグラ		2	1	5
21	26	2×2	セイタカアワダチソウ	シバ	2	0.5	20
22	19	2×2	ヒメムカシヨモギ	アキノエノコログサ	1	0	10
23		2×5	アズマネザサ		8	0	100
24		10×10	マダケ		10	0	100
25		2×5	マダケ		8	0	100
26		5×5	マダケ		8	0	100

植生詳細テーブル

項目を植生詳細ID、植生調査ID、植生名、被度、群度、年草、帰化植物とする。このうち植生調査IDは植生テーブル中の植生調査IDとリレーションさせる。

ここで、被度とは各階層に、それぞれの出現種がどれくらいの広がり葉を茂らせて生育しているかを表す。

被度 5 : 被度が調査面積の 4 分の 3 以上を占めているもの。個体数は任意。

被度 4 : 被度が調査面積の 2 分の 1 から 4 分の 3 を占めているもの。個体数は任意。

被度 3 : 被度が調査面積の 4 分の 1 から 2 分の 1 を占めているもの個体数は任意。

被度 2 : 個体数がきわめて多いか、または、被度が調査面積の 10 分の 1 から 4 分の 1 を占めているもの。

被度 1 : 個体数は多いが、被度は 20 分の 1 以下、または、個体数が少なく被度が 10 分の 1 以下のもの。

被度 + : 個体数も少なく、被度も少ないもの。

被度 r : きわめて稀に、最低被度で出現するもの。

また群度とは、調査区内やその周辺で個々の植物個体がどのような分野状態で生育しているかを示す尺度で、被度とは直接関係ない。

群度 5 : 調査区内 (樹木にあっては調査区周辺も含めて) に一面に生育していて、その葉群はだいたい連続している。

群度 4 : 1 , 2 ヶ所欠けて斑紋状に穴があいたような状態のもの。

群度 3 : 2 , 3 ヶ所に小さい斑紋状に群がって生育しているもの。

群度 2 : 2 , 3 ヶ所の小さい群をなして生育しているもの。

群度 1 : 1 , 2 ヶ所に離れて単独に生育しているもの。

植生詳細テーブルの一部を表 - 16 に示す。

表 - 16 植生詳細テーブル

植生詳細ID	植生調査ID	植生名	被度	群度	年草	帰化植物
1	1	ミソソバ	1	2	1	FALSE
2	1	イヌヒエ	1	2	1	FALSE
3	1	オヒシバ	1	1	1	FALSE
4	1	ヒメクゲ	+	1	多	FALSE
5	1	スベリヒユ	+	1	1	FALSE
6	1	チョウジタデ	+	1	1	FALSE
7	1	ヒデリコ	+	1	1	FALSE
8	1	トキンソウ	+	1	1	FALSE
9	1	ギシギシ	+	1	多	FALSE
10	1	イヌタデ	+	1	1	FALSE
11	1	イボクサ	+	1	1	FALSE
12	1	クロテンツキ	+	1	1	FALSE
13	1	ヒメヒラテンツキ	+	1	1	FALSE
14	1	コゴメガヤツリ	+	1	1	FALSE
15	1	タカサブロウ	+	1	1	FALSE
16	1	スカシタゴボウ	+	1	1~2	FALSE
17	1	アメリカアゼナ	+	1	1	TRUE
18	1	ヘビイチゴ	+	1	多	FALSE
19	1	メヒシバ	+	1	1	FALSE
20	2	メヒシバ	3	4	1	FALSE
21	2	オヒシバ	1	2	1	FALSE
22	2	アキノエノコログサ	1	2	1	FALSE
23	2	ホソアオゲイトウ	+	1	1	TRUE
24	2	コゴメガヤツリ	+	1	1	FALSE
25	2	シロザ	+	1	1	TRUE
26	2	ケアリタソウ	+	1	1	TRUE
27	2	ヒメマツバボタ	+	1	1	TRUE
28	2	コスズメガヤ	+	1	1	TRUE
29	2	ヌカキビ	+	1	1	TRUE
30	2	ハキダメギク	+	1	1	TRUE
31	2	ヨウシュヤマゴボウ	+	1	多	TRUE
32	2	スベリヒユ	+	1	1	FALSE
33	2	キンエノコロ	+	1	1	FALSE

魚類テーブル

項目を魚類調査ID、開始断面ID、終了断面ID、採捕数、採捕種数、採取方法、調査月日とする。このうち開始断面IDと終了断面IDは断面基本テーブル中の断面IDとリレーションさせる。魚類テーブルの一部を表 - 17に示す。

表 - 17 魚類テーブル

魚類調査ID	開始断面ID	終了断面ID	採捕数	採捕種数	採取方法	調査月日
1	13	31	5	1	投網	04-Aug-97
2	13	32	2	2	トラップ(ガラス釜魚キラー)	04-Aug-97
3	13	32	1	1	その他	04-Aug-97
4	13	31	14	4	投網	28-Sep-99
5	13	32	11	3	トラップ(ガラス釜魚キラー)	28-Sep-99
6	13	32	37	7	投網	24-Oct-00
7	13	32	70	9	電気ショッカー	24-Oct-00
8	13	32	36	8	電気ショッカー	04-Oct-00

魚類詳細テーブル

項目を魚類詳細ID、魚類調査ID、魚類名、採捕数とする。このうち魚類調査IDは魚類テーブル中の魚類調査IDとリレーションさせる。魚類詳細テーブルの一部を表 - 18に示す。

表 - 18 魚類詳細テーブル

魚類詳細ID	魚類調査ID	魚類名	採捕数
1	1	ウグイ	5
2	2	フナ類	1
3	2	ナマズ	1
4	3	シマドジョウ	1
5	4	カワムツ	1
6	4	ウグイ	10
7	4	カマツカ	2
8	4	コイ	1
9	5	カワムツ	7
10	5	ウグイ	1
11	5	シマドジョウ	3
12	6	カマツカ	15
13	6	オイカワ	10
14	6	フナ	4
15	6	ナマズ	1
16	6	ウグイ	4
17	6	クチボソ	1
18	6	コイ	2
19	7	タモロコ	30
20	7	オイカワ	7
21	7	ウグイ	5
22	7	ドジョウ	4
23	7	シマドジョウ	2
24	7	カワムツ	4
25	7	フナ	9
26	7	モツゴ	8
27	7	ナマズ	1
28	8	タモロコ	6

(4) クエリについて

クエリは、テーブルのフィールドを組み合わせたリ、抽出条件を設定したりすることで、テーブルから必要なデータを取り出すためのデータベースオブジェクトで、クエリ自体はデータをもたずにテーブルを参照するので、テーブルのデータを変更すると、クエリの実行結果も変わる。例として、水系基本テーブルから水系名、河川基本テーブルから河川名、河床材料テーブルから河床材料、植生テーブルから方形枠、種数、植生詳細テーブルから植生名、被度、群度を組み合わせて作成したクエリの一部を表 - 19に示す。

表 - 19 クエリ画面例

水系名	河川名	河床材料	方形枠	種数	植生名	被度	群度
利根川	姿川	玉石	2 × 2	19	ミゾソバ	1	2
利根川	姿川	玉石	2 × 2	19	ミゾソバ	1	2
利根川	姿川	玉石	2 × 2	19	ミゾソバ	1	2
利根川	姿川	玉石	2 × 2	19	イヌビエ	1	2
利根川	姿川	玉石	2 × 2	19	イヌビエ	1	2
利根川	姿川	玉石	2 × 2	19	イヌビエ	1	2
利根川	姿川	玉石	2 × 2	19	オヒシバ	1	1
利根川	姿川	玉石	2 × 2	19	オヒシバ	1	1
利根川	姿川	玉石	2 × 2	19	オヒシバ	1	1
利根川	姿川	玉石	2 × 2	19	ヒメクグ	+	1
利根川	姿川	玉石	2 × 2	19	ヒメクグ	+	1
利根川	姿川	玉石	2 × 2	19	ヒメクグ	+	1
利根川	姿川	玉石	2 × 2	19	スベリヒユ	+	1
利根川	姿川	玉石	2 × 2	19	スベリヒユ	+	1
利根川	姿川	玉石	2 × 2	19	スベリヒユ	+	1
利根川	姿川	玉石	2 × 2	19	チョウジタデ	+	1
利根川	姿川	玉石	2 × 2	19	チョウジタデ	+	1
利根川	姿川	玉石	2 × 2	19	チョウジタデ	+	1
利根川	姿川	玉石	2 × 2	19	ヒデリコ	+	1
利根川	姿川	玉石	2 × 2	19	ヒデリコ	+	1
利根川	姿川	玉石	2 × 2	19	ヒデリコ	+	1
利根川	姿川	玉石	2 × 2	19	トキンソウ	+	1
利根川	姿川	玉石	2 × 2	19	トキンソウ	+	1

(5) フォームの作成と活用

フォームは、テーブルやクエリのレコードをみやすくレイアウトし、データの入力や参照、検索、印刷を行うためのデータベースオブジェクトで、データの入力はテーブルやクエリでも行えるが、フォームでもデータ入力を行うことができる。フォームから入力したデータは、テーブルをもとにしたフォームの場合は直接テーブルに、クエリをもとにしたフォームの場合はクエリを介してテーブルの保存される。またフォームには、デザインビュー、フォームビュー、データシートビューの3通りの表示方法がある。デザインビューでは、テーブルやクエリのフィールドを配置したり、データを入力しやすくするための設定をしたりして、フォームのデザインを行う。フォームビューでは、データの入力や参照を行う。データシートビューでは、フォームに配置されたテーブルやクエリのフィールドの値を、テーブルやクエリのデータシートビューと同じようなデータシートビューで表示する。

フォームの種類には、単票形式（一画面に1レコードの内容を表示する形式のフォーム）、表形式（表の形をしたフォーム）、データシート形式（テーブルのデータシートと同様の形のフォーム）、帳票形式（単票形式に似ているが、入力欄の上に項目名が表示される）、メイン/サブフォーム（1対多の関係にある2つのテーブルを関連付けて表示する形式のフォーム）がある。

また、フォームをテーブルやクエリを利用して作成していくときに、2つ以上のテーブルやクエリを利用し、フォームの中にもう一つフォームを入れることができる。フォームの中にあるフォームのことをサブフォーム、もとのフォームのことをメインフォームとよぶ。

1) 水系フォーム

水系基本テーブルから水系名、流域面積、河川基本テーブルから河川名、流域面積、管理者、合流河川を利用し水系フォームを作成した。ここで、河川基本テーブルはサブフォームにし、データシート形式で作成。水系フォームの一部を表-20に示す。

表 - 20 水系フォーム

水系名		流域面積(km2)	
利根川	3728.1		

河川基本	河川名	流域面積	管理者	合流河川
▶	小貝川	162.2	栃木県	利根川
	五行川	225.3	栃木県	小貝川
	江川(真)	34.5	栃木県	五行川
	行屋川	8	栃木県	五行川
	野元川	59.4	栃木県	五行川
	井沼川	26.3	栃木県	五行川
	大沼川	7.7	栃木県	井沼川
	冷子川	11.2	栃木県	井沼川
	ぐみ川	11	栃木県	小貝川
	百日鬼川	9.3	栃木県	小貝川
	百日鬼川放水路	0	栃木県	百日鬼川
	大羽川	24.2	栃木県	小貝川
	小宅川	22.7	栃木県	小貝川
	大川	27.5	栃木県	小貝川
	桜川	17.5	栃木県	小貝川
	鏡谷川	3.8	栃木県	小貝川
	鬼怒川	1147.2	栃木県	利根川
	田川	231.3	栃木県	鬼怒川
	武名瀬川		栃木県	田川
	釜川	5.3	栃木県	田川
	御用川	19.3	栃木県	田川
	釜川放水路	2.3	栃木県	田川
	山田川	41.6	栃木県	田川
	前川	3.5	栃木県	山田川
	清川(空)	13.4	栃木県	田川

レコード: 1 / 175

2) 雨量観測所フォーム

雨量観測所テーブルから、雨量観測所ID、雨量観測所名、緯度、経度、所在地、河川基本テーブルから、河川名、日雨量テーブルから、月、各日を利用し雨量観測所フォームを作成した。日雨量テーブルはサブフォームにし、データシート形式で作成する。雨量観測所フォームの一部を表 - 21に示す。

表 - 21 雨量観測所フォーム

雨量観測所名	宇都宮	雨量観測所ID	6																																																																																																																																																																																		
所在地	宇都宮市竹林1, 030-2																																																																																																																																																																																				
河川名	田川	緯度	36° 34' 25"																																																																																																																																																																																		
		経度	139° 53' 52"																																																																																																																																																																																		
日雨量	<table border="1"> <thead> <tr> <th>月</th> <th>1日</th> <th>2日</th> <th>3日</th> <th>4日</th> <th>5日</th> <th>6日</th> <th>7日</th> <th>8日</th> <th>9日</th> <th>10日</th> <th>11日</th> <th>12日</th> </tr> </thead> <tbody> <tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td>4</td></tr> <tr><td>3</td><td></td><td></td><td></td><td></td><td></td><td></td><td>7</td><td>2</td><td>3</td><td></td><td></td><td></td></tr> <tr><td>4</td><td></td><td>05</td><td></td><td></td><td>1</td><td>65</td><td>2</td><td></td><td></td><td>22</td><td>7.5</td><td></td></tr> <tr><td>5</td><td></td><td></td><td></td><td>56.5</td><td>17</td><td></td><td></td><td></td><td></td><td></td><td>4.5</td><td></td></tr> <tr><td>6</td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td>1.5</td><td></td><td></td><td>1</td><td></td></tr> <tr><td>7</td><td>25</td><td>4</td><td>4</td><td>0.5</td><td>0.5</td><td></td><td></td><td></td><td></td><td>22</td><td>15.5</td><td>13</td></tr> <tr><td>8</td><td></td><td></td><td></td><td>8.5</td><td></td><td>0.5</td><td></td><td></td><td></td><td>1</td><td></td><td>27</td></tr> <tr><td>9</td><td>6</td><td>0.5</td><td></td><td></td><td>0.5</td><td></td><td></td><td>13.5</td><td>25.5</td><td>16.5</td><td></td><td>4</td></tr> <tr><td>10</td><td>1.5</td><td>0.5</td><td></td><td></td><td></td><td></td><td>0.5</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>11</td><td>6</td><td>1.5</td><td>0.5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>0.5</td><td>6</td></tr> <tr><td>12</td><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table>												月	1日	2日	3日	4日	5日	6日	7日	8日	9日	10日	11日	12日	1													2											3	4	3							7	2	3				4		05			1	65	2			22	7.5		5				56.5	17						4.5		6							3	1.5			1		7	25	4	4	0.5	0.5					22	15.5	13	8				8.5		0.5				1		27	9	6	0.5			0.5			13.5	25.5	16.5		4	10	1.5	0.5					0.5						11	6	1.5	0.5								0.5	6	12	4											
月	1日	2日	3日	4日	5日	6日	7日	8日	9日	10日	11日	12日																																																																																																																																																																									
1																																																																																																																																																																																					
2											3	4																																																																																																																																																																									
3							7	2	3																																																																																																																																																																												
4		05			1	65	2			22	7.5																																																																																																																																																																										
5				56.5	17						4.5																																																																																																																																																																										
6							3	1.5			1																																																																																																																																																																										
7	25	4	4	0.5	0.5					22	15.5	13																																																																																																																																																																									
8				8.5		0.5				1		27																																																																																																																																																																									
9	6	0.5			0.5			13.5	25.5	16.5		4																																																																																																																																																																									
10	1.5	0.5					0.5																																																																																																																																																																														
11	6	1.5	0.5								0.5	6																																																																																																																																																																									
12	4																																																																																																																																																																																				
*	レコード: 2 / 12																																																																																																																																																																																				

3) 流量観測所フォーム

流量観測所テーブルから、流量観測所ID、流量観測所名、緯度、経度、所在地、河川基本テーブルから、河川名、日流量テーブルから、月、各日を利用し流量観測所フォームを作成した。日流量テーブルはサブフォームにし、データシート形式で作成した。流量観測所フォームの一部を表 - 22に示す。

表 - 22 流量観測所フォーム

流量観測所名	<input type="text" value="権現"/>	流量観測所ID	<input type="text" value="1"/>
所在地	<input type="text" value="芳賀郡益子町上山"/>		
河川名	<input type="text" value="小貝川"/>	緯度	<input type="text" value="32° 26' 14"/>
日流量		経度	<input type="text" value="140° 4' 35"/>

月	1日	2日	3日	4日	5日	6日	7日	8日	9日	10日	11日
▶ 1	0.82	0.82	0.69	0.69	0.69	0.69	0.82	0.82	0.69	0.69	0.69
2	0.58	0.58	0.58	0.58	0.58	0.47	0.47	0.47	0.47	0.47	0.69
3	0.69	0.69	0.69	0.58	0.58	0.58	0.58	0.96	0.82	0.96	0.82
4	4.01	3.18	2.92	2.45	2.22	2.22	2.45	2.1	1.81	2.01	8.72
5	3.72	3.45	3.18	7.47	33.16	12	8.29	6.69	5.61	4.94	5.61
6	0.82	0.58	0.47	0.47	0.47	0.38	0.58	0.69	0.47	0.38	3.45
7	25.72	15.23	12.51	11.01	10.06	7.47	5.61	4.31	4.01	4.01	14.11
8	1.11	0.96	0.96	0.82	0.96	1.11	1.11	0.96	0.69	0.82	0.96
9	2.01	4.01	2.01	1.62	1.81	1.27	1.11	1.44	1.44	1.11	0.96
10	8.72	4.62	3.72	3.18	2.68	2.45	2.22	2.22	2.01	1.81	1.62
11	6.69	6.32	4.31	3.72	3.18	2.68	2.45	2.22	2.01	1.81	1.62
12	1.11	1.62	1.44	1.11	1.11	1.11	1.11	0.96	0.58	0.96	0.82

レコード: / 12

4) 断面フォーム

断面基本フォーム

断面基本テーブルから、断面名、X、Y、河川基本テーブルから、河川名、断面地形テーブルから、Y'、Z、調査日を利用し断面基本フォームを作成する。断面地形テーブルはサブフォームにしデータシート形式で作成する。また、Y'、Zをグラフ化した。断面基本フォームの一部を表 - 23に示す。

表 - 23 断面基本フォーム

河川名	姿川	X	58670.360	断面ID	7
断面名	225+50	Y	665.446		

断面地形

	Y'	Z	調査日
▶	0	102.959	00/09/08
	4.7	101.014	00/09/08
	6.8	101.029	00/09/08
	9.5	100.949	00/09/08
	15.9	98.909	00/09/08
	17.9	98.389	00/09/08
	18.9	98.286	00/09/08
	30.4	98.524	00/09/08

レコード: 1 / 1:

横断-標高

Y'	Zの合計
0	102.959
4.7	101.014
6.8	101.029
9.5	100.949
15.9	98.909
17.9	98.389
18.9	98.286
30.4	98.524
39.9	100.949
40.8	100.949
41.3	100.949
45.3	102.959

河床材料フォーム

断面基本テーブルから、断面名、X、Y、河川基本テーブルから、河川名、河床材料テーブルから、Y、河床材料、代表粒径、調査日を利用し河床材料フォームを作成した。河床材料テーブルはサブフォームにし、データシート形式で作成する。河床材料フォームの一部を表 - 24に示す。

表 - 24 河床材料フォーム

断面名 X
 河川名 Y

F5河床材料

	Y'	河床材料	代表粒径(cm)	調査日
▶	11.8	粗礫	5~10	00/09/05
	17.6	玉石	10~20	00/09/05
	23.7	粗礫	5~10	00/09/05
*	0			

レコード: / 3

断面名 X
 河川名 Y

F5河床材料

	Y'	河床材料	代表粒径(cm)	調査日
▶	14	砂		00/09/05
	24	玉石	10~20	00/09/05
	29.8	玉石	10~20	00/09/05
*	0			

レコード: / 3

植生フォーム

植生テーブルから、優占種 1、優占種 2、植被率 1、植被率 2、種数、一年草率、方形枠、Y、調査日、河川基本テーブルから、河川名、断面基本テーブルから、断面名、X、Y、植生詳細テーブルから、植生名、種類、群度、被度、年草、帰化植物を利用し植生フォームを作成した。植生詳細テーブルはサブフォームにしてデータシート形式で作成する。表 - 25に植生フォームの一部を示す。

表 - 25 植生フォーム

河川名	姿川	植生詳細				
断面名	230+50	植生名	被度	群度	年草	帰化植物
方形枠	2×2	メヒシバ	3	4	1	<input type="checkbox"/>
Y	12.8	オヒシバ	1	2	1	<input type="checkbox"/>
調査月日	99/09/28	アキノエノコログサ	1	2	1	<input type="checkbox"/>
優占種1	メヒシバ	ホンアオグイトウ	+	1	1	<input checked="" type="checkbox"/>
優占種2		コゴメガヤツリ	+	1	1	<input type="checkbox"/>
高さⅠ (m)	1.1	シロザ	+	1	1	<input checked="" type="checkbox"/>
高さⅡ (m)	0	ケアリタノウ	+	1	1	<input checked="" type="checkbox"/>
種数	23	ヒメマツバボタ	+	1	1	<input checked="" type="checkbox"/>
一年草比率 (%)	87	コスズメガヤ	+	1	1	<input checked="" type="checkbox"/>
植被率Ⅰ (%)	80	ヌカキビ	+	1	1	<input checked="" type="checkbox"/>
植被率Ⅱ (%)	0	ハキダメギク	+	1	1	<input checked="" type="checkbox"/>
		ヨウシュヤマゴボウ	+	1	多	<input checked="" type="checkbox"/>
		スベリヒユ	+	1	1	<input type="checkbox"/>
		キンエノコロ	+	1	1	<input type="checkbox"/>
		スカシタゴボウ	+	1	1~2	<input type="checkbox"/>
		イヌタデ	+	1	1	<input type="checkbox"/>
		カタバミ	+	1	多	<input type="checkbox"/>
		ザクロソウ	+	1	1	<input type="checkbox"/>
		エノキグサ	+	1	1	<input type="checkbox"/>
		イヌビエ	+	1	1	<input type="checkbox"/>
		ヌカキビ	+	1	1	<input type="checkbox"/>
		コツブキンエノコロ	+	1	1	<input type="checkbox"/>
		オオイヌタデ	+	1	1	<input type="checkbox"/>
		レコード	1	/ 23		

3 . GISによる中小河川データベースの活用

(1) GISについて

GIS (geographic information system : 地理情報システム) とは、地理的に分布する情報を、位置を表わす空間情報と性質を表わす属性情報を地理的に分布する情報を統合的に管理することにより、空間的な情報の統合が行える意思決定支援システムであり、多様な情報源から大量の空間的データを取り込み、地図情報を主体としたデータベースを作成し、それを効率的に蓄積、検索、交換、解析し、地図出力やレポート作成などができるように設計されたシステム。現在、わが国におけるGISの整備が遅れている現況をふまえ、GISの効率的な整備をすすめるとともに、データ整備における国、公共機関、民間企業などの重複投資を避け、その相互利用を図っていく必要がある。そして、多様なGIS利用主体のニーズを満たす共通のデータ基盤を標準化していくことが求められている。

GIS技術を河川環境整備に利用すると、メリットとして位置をディスプレイ上の地図で視覚的にとらえることができ、その上で地図上に表示された地点の情報を即座に見ることができるという点がある。また、さまざまな分野のデータを重ね合わせて表示することにより、より総合的な認識が可能になる。さらに、GISに位置情報が入力された時点で、その後どのような縮尺の地図を作成しても、その位置を正確に表示できる。座標を伴わない単なる絵としての地図とは異なり、GIS上では、位置情報を正確に表示しつつ、地図の必要な部分の拡大、縮小等が自在にできるため、位置情報の正確な把握に有効である。

本研究では、作成したデータベース内の緯度経度の把握できる地点の地図上でのポイント作成、緯度経度のわからないものではデジタイザを利用したポイント作成、それぞれポイントを作成した個所の位置情報の表示。また、1地点の情報だけでなく、検索、抽出できるようにGISと先に作成したデータベース全体をリンクさせ、どのポイントからでも、データベース全体を参照できるようにした。

GISの製品は、従来はUnixベースで動作する高価なものばかりであったが、最近ではWindows環境で動作する軽量低価格なものが普及するようになってきた。本研究では、そのなかでもMapInfoという製品を使用することにする。この製品の価格・性能はもちろんのこと、この製品のラインアップには、インターネット配信を念頭に置いたJavaライブラリもあり、将来的なシステムのグレードアップがスムーズにいくと考えたからである。

(2) データベースのGISでの活用

緯度経度がわかっている雨量観測所と流量観測所についてポイント作成を行った。次に姿川根古屋橋上流部改修部の21ポイントについて、緯度経度の情報がなかったため、デジタイザを利用して緯度経度を確認し、ポイント作成を行った。

1) 雨量観測所

緯度経度を含む雨量観測所のテーブルをエクセルにエクスポート（アクセスのデータをエクセルやワードに変換する機能）する。このエクセルからマップ上でポイントの作成を行った。このとき、座標系が度分秒なので度分秒変換機能を利用して小数度に変換させる。マップ上のポイントをクリックすると情報をみることができ、また元の表全体を表示することができる。雨量観測所のGIS画面例を図 - 2 雨量観測所GIS画面例に示す。



図 - 2 雨量観測所GIS画面例

2) 流量観測所

1) 雨量観測所と同様にGIS上のマップにポイントの作成を行う。緯度経度を含むテーブルをエクスポートでエクセルに変換し、緯度経度の度分秒表示を小数度表示に変換してポイント作成に利用する。流量観測所のGIS画面例を図 - 3 流量観測所GIS画面例に示す。



図 - 3 流量観測所GIS画面例

また、1) 雨量観測所や2) 流量観測所で作成したマップのポイントを同時に表示することができる。同時に表示したものを図 - 4 雨量観測所流量観測所GIS画面例に示す。



図 - 4 雨量観測所流量観測所GIS画面例

3) 断面位置

姿川根古屋橋上流部の改修部、NO.225からNO.231+75の範囲で25m間隔のポイントを作成した。緯度経度の情報がなかったためデジタイザを利用し、地図からポイントを読み取った。

デジタイザで地図を読むには少なくとも4点以上の基準点を必要とする。はじめに1/50000の宇都宮の地形図をデジタイザが使える大きさに縮小し、その4隅の緯度経度を基準点として入力、利用した。ポイント作成をする個所は姿川根古屋橋上流部なので、この地図では範囲面積が小さすぎポイントをとることができない。そこで根古屋橋上流部周辺から緯度経度を4点読み取る。次に姿川根古屋橋上流部の1/1500の平面図をデジタイザに利用し、先に読み取った緯度経度4点を基準点として利用した。これより、姿川根古屋橋上流部の改修部、NO.225からNO.231+75の25m間隔の合計21ポイントの緯度経度を読み取る。読み取った緯度経度を地点別にエクセルでまとめ、マップ上へのポイント作成を行った。ポイントをクリックすればそのポイントの情報をみることができ、また、もとの表全体を表示することができる。これを図 - 5 断面位置GIS画面例に示す。

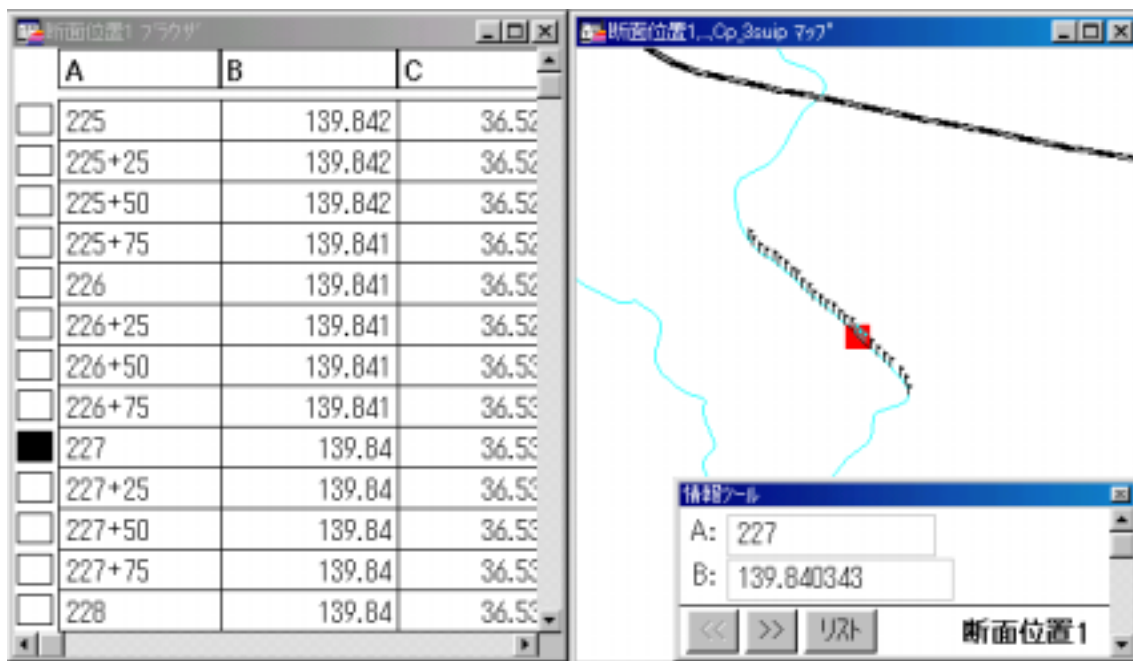


図 - 5 断面位置GIS画面例

4) 河床材料

断面位置のポイント範囲で、河床材料のポイント作成を行った。情報を知りたい個所をクリックすれば情報が表示される。河床材料情報の表を一緒に表示することができる。画面例を図 - 6 に示す。

横断距離	標高	河床材料	代表粒径
<input type="checkbox"/> 0	101.55	玉石	10~20
<input type="checkbox"/> 4.55	98.941	玉石	10~20
<input type="checkbox"/> 6.7	99.191	玉石	10~20
<input type="checkbox"/> 9.3	99.646	玉石	10~20
<input type="checkbox"/> 13	97.776	玉石	10~20
<input type="checkbox"/> 13.1	97.616	玉石	10~20
<input type="checkbox"/> 15.7	97.368	玉石	10~20
<input type="checkbox"/> 33.6	97.168	玉石	10~20
<input type="checkbox"/> 33.7	97.296	玉石	10~20
<input type="checkbox"/> 38.1	99.036	玉石	10~20
<input type="checkbox"/> 39.5	99.981	玉石	10~20
<input type="checkbox"/> 40.3	98.916	玉石	10~20

情報パネル
断面名: 226+75
横断距離: 46.7
リセット 断面_クエリ1

図 - 6 河床材料画面例

5) 植生

断面位置で行ったポイント範囲で、植生のポイント作成を行った。情報を知りたい個所をクリックすれば情報を参照することができる。また、植生の情報の表を表示することができる。図 - 7 に画面例を示す。

断面名	優占種 1	優占種 2
<input type="checkbox"/> 228+50	メヒシバ	
<input type="checkbox"/> 230+50	ケイヌビエ	
<input type="checkbox"/> 230+50	メヒシバ	ヒメムカシ
<input type="checkbox"/> 230+50	メドハギ	
<input type="checkbox"/> 230+50	オオアレチノギク	
<input type="checkbox"/> 230+50	ヨモギ	シバ
<input type="checkbox"/> 230+25	ケイヌビエ	ヌカキビ
<input type="checkbox"/> 230+25	コブナグサ	
<input type="checkbox"/> 230+25	ヌカキビ	ツルマメ
<input type="checkbox"/> 230+25	カナムグラ	
<input checked="" type="checkbox"/> 230+25	セイタカアワダチソ	シバ
<input type="checkbox"/> 228+50	ヒメムカシヨモギ	アキノエノ

情報パネル
断面名: 230+25
優占種 1: セイタカアワダ
リセット 断面_クエリ2

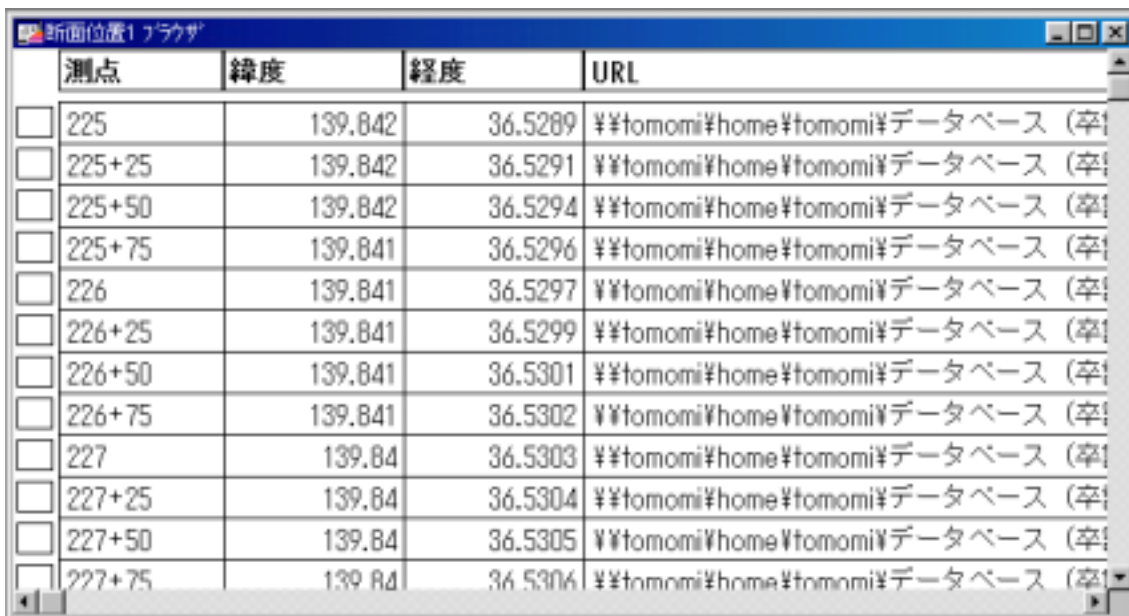
図 - 7 植生画面例

(3) リレーショナルデータベースのGISでの利用

(2)で作成したポイントを利用し、先に作成したリレーショナルデータベースにリンクさせた。GISでアクティブオブジェクトとHotLinkツールを使用するとマップのオブジェクトとラベルから、ファイルやURLなどを直接起動することができる。アクティブオブジェクトはWeb上の情報や別のアプリケーションの文書をマップから呼び出す表示機能です。

アクティブオブジェクトの作成

マップにアクティブオブジェクトを作成するにはリンク先情報を収めるためのフィールドをテーブルに追加する必要がある。アクティブオブジェクトを作成するテーブルを開く（ツールバーのウィンドウ中からブラウザウィンドウを表示、を選択し使用したいものを開く）。リンクテキストを含むために十分な長さの文字フィールドを追加する（ツールバーのテーブル中からテーブル管理を選択しテーブル定義をクリックして使用するものを選びフィールドの追加で文字数を設定し、新しいフィールドを追加する）。追加したフィールドに関連付けたいリンク先の文字列を各レコードに入力する。図 - 8 にレコードを入力した画面例を示す。



測点	緯度	経度	URL
<input type="checkbox"/> 225	139.842	36.5289	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 225+25	139.842	36.5291	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 225+50	139.842	36.5294	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 225+75	139.841	36.5296	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 226	139.841	36.5297	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 226+25	139.841	36.5299	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 226+50	139.841	36.5301	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 226+75	139.841	36.5302	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 227	139.84	36.5303	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 227+25	139.84	36.5304	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 227+50	139.84	36.5305	¥¥tomomi¥home¥tomomi¥データベース (卒!
<input type="checkbox"/> 227+75	139.84	36.5306	¥¥tomomi¥home¥tomomi¥データベース (卒!

図 - 8 アクティブオブジェクト画面例

HotLinkオプションの設定

アクティブオブジェクトを起動するHotLinkを指定するには、アクティブオブジェクトを含むテーブルをマップウィンドウに表示、レイヤ管理ダイアログ中のHotLinkを設定する。このときリンク先ファイル名を記入し、HotLinkツールでクリックしたときに起動する対象を設定する。HotLinkツールでリレーショナルデータベースを起動させた画面例を図 - 9 と図 - 10に示す。

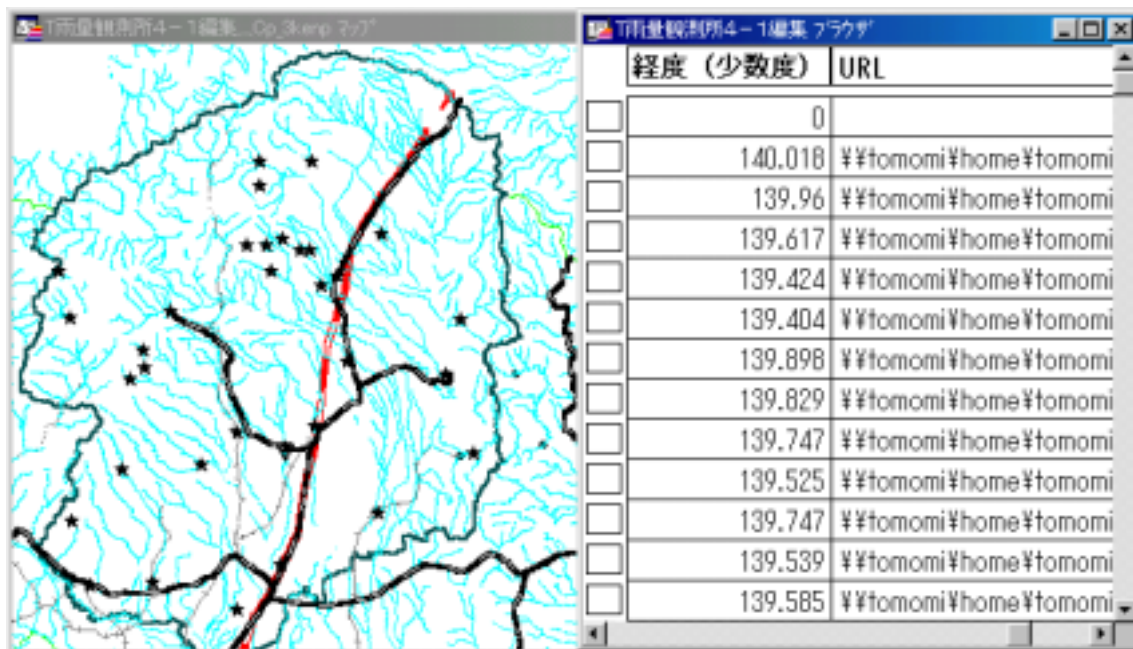
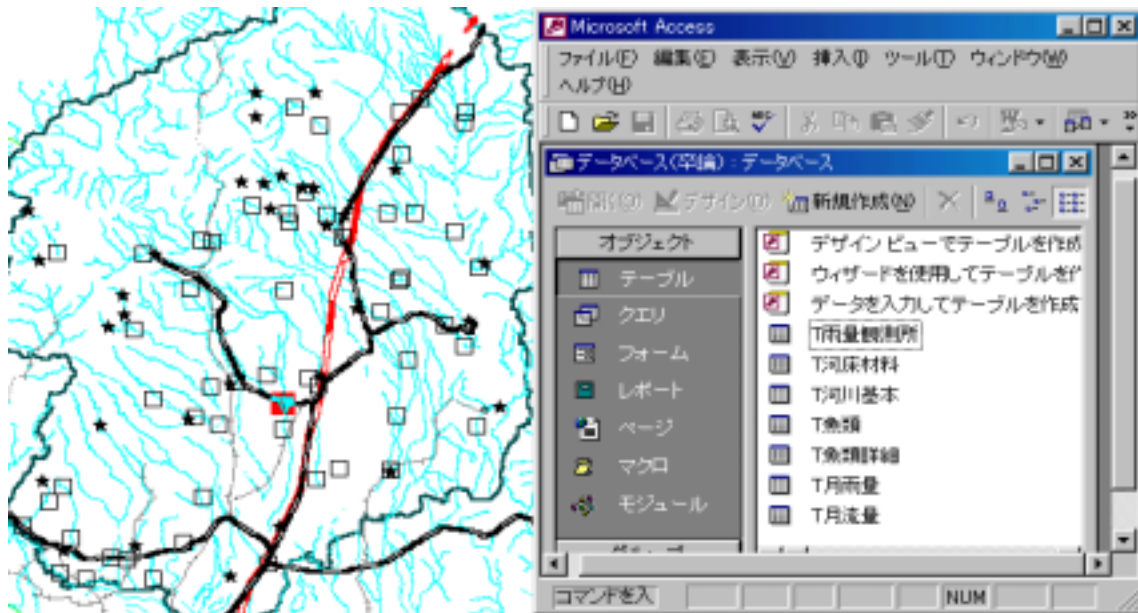


図 - 9 HotLink起動雨量観測所画面例とブラウザ画面例

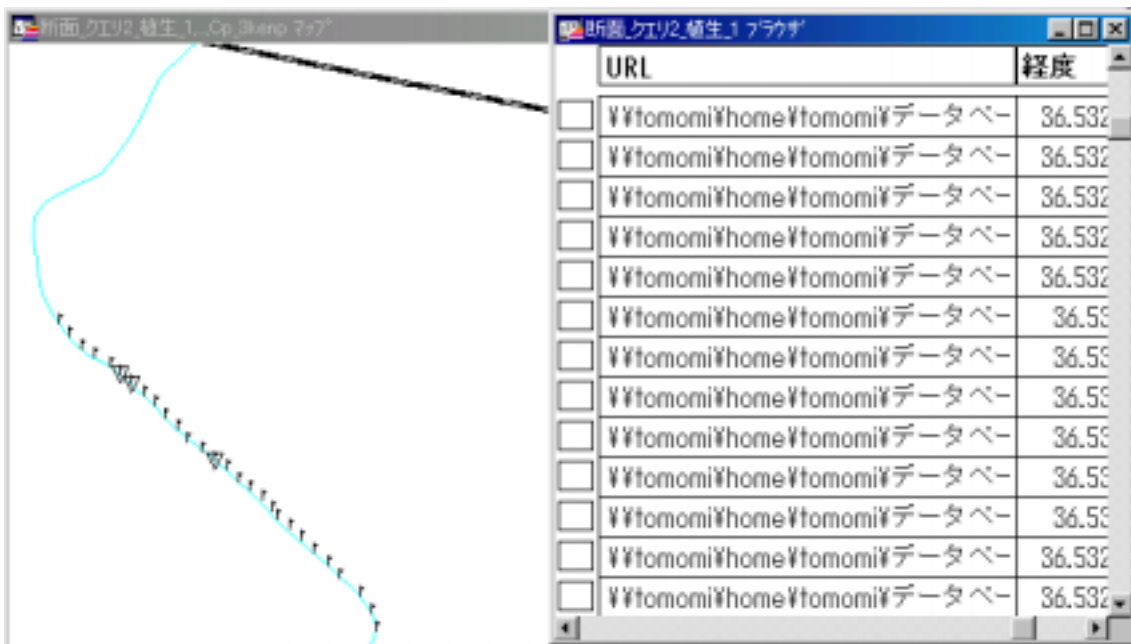
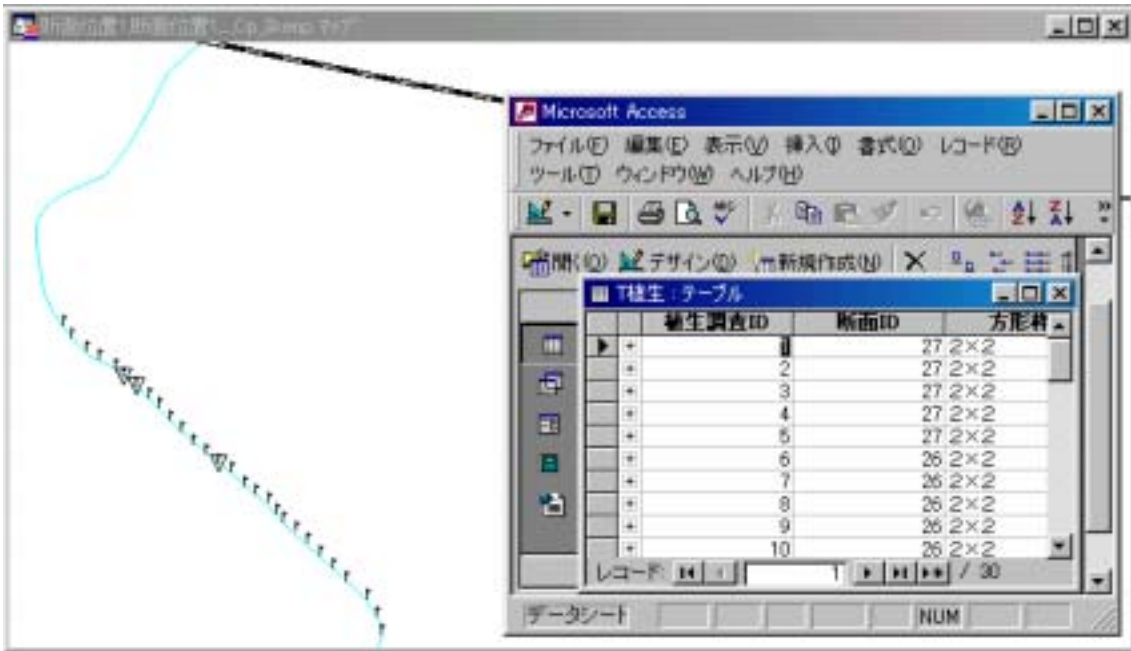


図 - 10 HotLink植生画面例とブラウザ画面例

4 . Java言語を用いたWebデータベースの構築

(1) Webデータベースの目的と構成

1) Webデータベースの目的

本章では、第2章で開発したデータベースをWebデータベースとして発展させる。これは、サーバーに主たるシステムを構築し、それをLANあるいはインターネットなどのネットワークでつながれたコンピュータからWebブラウザを介して利用するものである。本章で開発するシステムによって、以下の事項が可能になる。

第2章と第3章で開発してきたシステムはスタンドアロン環境であり、システムがインストールしてあるコンピュータに対峙しなければ、システムを利用できない。Webデータベースとすることで、いつでもどこでも誰でもシステム利用することが可能になる。

このことは、このシステムを利用するのが、河川管理者のオフィスのみでなく、一般市民でもよいことを意味する。したがって、地域にデータを公開することで、河川の状況を住民に理解してもらい、あるいは総合学習に役立てるなど、地域貢献の一翼を担うことができる。

また、河川管理者側にとっても有効なのは、管理事務所が散在していても共通した業務を行えることである。しかも日常業務に密接したデータ項目をWebデータベースに配置することによって、業務をこなしながら自然とデータベースが充実するようになる。たとえば、災害時の被災箇所の測量データならびに復旧工事での断面データが生成されたときに、それを単に事務所資料としてしまわずに、Webデータベースに入力すれば、災害復旧ごとにデータが拡充し、それを他の事務所で、いつでも閲覧し、参考にすることができる。

ユーザー側（クライアント）が用意するソフトウェアは基本的にはWebブラウザだけでよい。したがって、各コンピュータに特別なソフトウェアを配布・インストールする必要はない。当然、システムの修正や変更の際も、サーバー側だけの対応で済ますことができる。

本章で開発するシステムは、Java言語を用いて開発する。したがって、単にデータの表示やメンテナンスだけでなく、クライアントの指示に従ってのデータ分析や、技術計算を実行するプログラムコードも容易に実装することが可能である。このことは、Webデータベース（あるいはWeb-GIS）が単に、現有するデータの表示だけでなく、クライアントとのメッセージのやり取りや、設計支援システムへ発展させる可能性を示している。本章では、そのほんの可能性を検証してみることにする。

2) Webデータベースのシステム構成

本章で開発するシステムは、以下のような構成とする。

データベースは第2章で構築したMS-Access2000システムのデータを、MySQLデータベース移植する。MySQLはフリーのデータベースサーバソフトウェアで、動作の高速性や安定性で有名で、世界的にも各所で利用されている。この移植作業を通して、万が一の場合には、MS-Accessを介してMySQLに接続し、データのメンテナンスを行えるようにした。

そのデータベースとクライアントを結びつける運用システムは、いわゆるMVC構造とする。その意味と実装方法を、表 - 26に示す。

表 - 26 MVC構造と今回の実装方法

区分	意味	実装方法
Model (モデル)	データベースと直接データをやり取りしたり、さまざまな複雑な処理(データ分析や技術計算)を扱う。	Java Beans
View (ビュー)	データをクライアントに表示する部分。今回は、Webブラウザへの表示コード。	JSP (Java Server Pages)
Controller (コントローラ)	クライアントからの要求をもとに、Model部分を動作させ、その結果をもとにView部分を組み立てる。	Java Http Servlet

システムをこのように役割分担させることによって、それぞれを開発するスタッフを分けて、平行してシステムを開発することが可能になるし、メンテナンスも容易になる。

今回は運用システムの開発に当たって使用したソフトウェアは、Java2 SDK1.4.0とJakartaプロジェクトのTomcat4.0.3付属のServlet用ライブラリ、ACMEのJavaAPIライブラリ、そしてシェアウェアのエディターPeggyPro4.0.3(アカデミック利用によりフリーライセンスを取得)である。

サーバーの仕様は表 - 27の通りである。

表 - 27 サーバーの仕様

メーカー	Faith
CPU	AthlonMP800+
メモリ	512MB
ハードディスク	40GB (UltraATA/100)
OS	Microsoft Windows XP
Webアプリケーションサーバー	Jakartaプロジェクト Tomcat4.0.3

以上の構成でわかるとおり、OS以外はすべてフリーのものを使用している。これで、開発にかかる費用を最小限に抑えることができる（このことについての問題点は後述する）。また使用しているサーバーソフトやJavaシステムは、Windows環境だけでなくUnix環境でも動作させることが可能なので、基本的には可搬性は高いものといえる。当然ながらUnix環境についても、フリーで十分な機能を有するものが配布されているので、それを活用すればシステムの運用コストも大幅に抑えることが可能だろう。

(2) Webデータベースのページ構成

Webデータベースでは、Webページの記述言語であるHTMLの機能を利用することができる、なかでもリンク機能は、さまざまなデータ項目（正確にはそれを表示するページ）を結びつけてクライアントに表示することができる。そのために上から下へといった階層的なメニュー構成はない。ただし、今回はその代わりに、データ構造自身が階層的（たとえば水系 河川 断面 断面形状のように）になっている。それをよりどころにして、どのようなWebページを作成していけばよいか検討した。その結果を図 - 11に示す。

HOME	タイトルページ
事務所リスト	管理事務所のリストを表示する。
事務所詳細情報	管理事務所の名称、管理機関、所在、管理する河川のリスト
河川検索	河川検索画面を表示する。河川名、管理者、水系から検索する。
河川リスト	河川の検索結果をリスト表示する。
河川情報表示	指定された河川の名称、管理事務所、流域面積、雨量・流量観測所一覧、河川断面のデータの有無など、
河川新規登録	河川データの新規登録
河川情報更新	河川データの情報更新
河川情報削除	河川データの削除
河川断面リスト	位置情報のある（何らかの調査がなされた）断面のリスト
断面形状リスト	断面測量データが存在している断面のリスト
断面形状表示	指定された断面の断面測量データの数値表示および断面形状のグラフ表示。

図 - 11 Webページの構成

事務所リスト・河川リストと同様に雨量・流量観測所リストも作成したり、それぞれの観測所の詳細情報や各年月日のデータの数値表示やグラフ表示も可能である。また断面形状と同様に、植生や魚類の調査データも表示することは可能である。本研究では、時間の都合でそれらを割愛したが、本研究の成果をもってすれば、それらを付加するのは実に容易である。

(3) Java Beansの作成

1) Java Beansの基本構造

(2)、2)に記したように、本章のシステムではMVC構造をとることにし、そのModel部分にはJava Beansを用いることにする。Java Beansとは、以下の要件を備えるJava言語のクラスのことである。

デフォルトコンストラクタすなわち引数を持たないコンストラクタをもつ。
プロパティを参照・設定するメソッド(アクセッサメソッド)をもつ。そのめそつどの名称は、参照メソッドは「get 」、設定メソッドは「set 」とする(はプロパティの名称)。
Serializableインタフェースを実装する。

ただし、 については、必ずしもアクセッサメソッドに使われている名称の変数が定義されている必要はなく、最終的に「get 」や「set 」で何らかの作業ができればよい。むしろプロパティとは、Java Beansを利用する側から見て参照や設定ができる属性であって、Java Beans内部での実装方法は問われない。

また についても、ネットワークを介してJava Beansをやり取りしないなら、特に必要はない。

2) 本システムでのJava Beansの種類と仕様

本システムでは、Java Beansを主としてデータベースとのデータのやり取りに活用する。本システムに收容されているデータは大きく分けて、河川関連、管理者関連、雨量観測所関連、流量観測所関連、の4グループであるが、(2)で記したように、本章では、 と に重点を置き、 のデータも多少交えてシステムを開発することになっている。これらのグループごとに、対象とする作業ごとにJava Beansを定義する必要がある。以下に今回のシステムで使用したJava Beansのクラスとその仕様を説明する。

河川関連のJava Beans

クラス名	WsListBean
概要	水系データのリストを取得する。
親クラス	なし
メソッド	
void clear()	全プロパティのクリア
void doQuery()	リスト検索の実行
int getCount()	データベースに登録されている水系の個数を返す。
int getWsId(int i)	i番目のデータの水系IDを返す。
String getWsName(int i)	i番目のデータの水系名称を返す。

クラス名	RiverSortBean
概要	河川リストを検索する際の並べ替え項目。
親クラス	なし
メソッド	
void clear()	全プロパティのクリア
int getCount()	データベースに登録されている並べ替え項目の個数を返す。
String getField(int i)	i番目の並べ替え項目のフィールド名を返す。
String getName(int i)	i番目の並べ替え項目のWebページでの表示名を返す。

クラス名	RiverListBean
概要	検索条件に合う河川データのリストを取得する。
親クラス	なし
メソッド	
void clear()	全プロパティのクリア
void doQuery()	リスト検索の実行
int getCount()	検索条件に合う河川データの個数を返す。
int getRvId(int i)	i番目の河川の河川IDを返す。
String getRvName(int i)	i番目の河川の河川名称を返す。
String getWsName(int i)	i番目の河川の水系名称を返す。
String getMgName(int i)	i番目の河川の管理者名称を返す。
String getSort(int i)	i番目の河川のデータを返す
String getSortField()	(河川検索画面で並べ替え項目に) 指定された項目名を返す。 (関連するどれかのテーブルのフィールド名でなくてはならない)
void setSortField(String str)	河川リストの並べ替え項目をセットする。(関連するどれかのテーブルのフィールド名でなくてはならない)
void setSortName(String str)	並べ替え項目のデータをどんなフィールド名にするかセットする。(見分けのための名称なので何でもよい)
void setWhere(String str)	河川リストの検索条件をセットする。引数にはSQL構文のWHERE節(「WHERE」は含まない)をセットする。
String getWhere()	河川リストの検索条件を返す。具体的には、SQL構文のWHERE節(「WHERE」は含まない)を返す。
String getSQL()	河川リストの検索に用いたSQL文の全体を返す。

クラス名	RiverInfoBean
概要	一河川の詳細データの操作
親クラス	なし
メソッド	
void clear()	: 全プロパティのクリア
void doQuery()	: 検索の実行
void doInsert()	: 新規登録の実行
void doUpdate()	: 情報更新の実行
void doDelete()	: 情報削除の実行
void setRvId(int i)、void setRvId(String str)	: 対象河川の河川IDをセットする。
int getRvId()	: 対象河川の河川IDを返す。
void setWslId(int i)、void setWslId(String str)	: 対象河川の水系IDをセットする。
int getWslId()	: 対象河川の水系IDを返す。
void setMglId(int i)、void setMglId(String str)	: 対象河川の管理者IDをセットする。
int getMglId()	: 対象河川の管理者IDを返す。
void setOfId(int i)、void setOfId(String str)	: 対象河川の事務所IDをセットする。
int getOfId()	: 対象河川の手続きIDを返す。
void setRvName(String str)	: 対象河川の河川名称をセットする。
String getRvName()	: 対象河川の河川名称を返す。
void setMgName(String str)	: 対象河川の管理者名称をセットする。
String getMgName()	: 対象河川管理者名称を返す。
void setOfName(String str)	: 対象河川の管理事務所名称をセットする。
String getOfName()	: 対象河川管理事務所名称を返す。
void setRvArea(double d)、void setRvArea(String str)	: 対象河川の流域面積をセットする。
double getRvArea ()	: 対象河川の流域面積を返す。
void setRvOrigin(String str)	: 対象河川の源流名称をセットする。
String getRvOrigin ()	: 対象河川の源流名称を返す。
void setRvTerm(String str)	: 対象河川の末端名称をセットする。
String getRvTerm ()	: 対象河川の末端名称を返す。
void setRvMemo(String str)	: 対象河川の備考事項をセットする。
String getRvMemo ()	: 対象河川の備考事項を返す。
String getSQL()	: 河川データの操作に用いたSQL文の全体を返す。
protected int setInt(String str)	: Stringオブジェクトを整数値に直す。
protected double setDouble (String str)	: Stringオブジェクトを実数値に直す。
protected String setString (String str)	: Stringオブジェクトのnullチェック
protected String myError(String str Exception exc)	: エラー発生時にメッセージstrとExceptionオブジェクトexcを与えて、エラーメッセージを表示する。

クラス名	SectionListBean
概要	指定された河川の（何らかのデータを有する）断面のリスト
親クラス	RiverInfoBean
メソッド	
void clear()	: 全プロパティのクリア
void doQuery()	: setRvIdメソッドでセットした河川IDで検索の実行
int getCount()	: 断面データの個数を返す。
int getScId(int i)	: i番目の断面IDを返す。
String getScName(int i)	: i番目の断面名称を返す。
double getScExt(int i)	: i番目の断面の延長距離を返す。
double getScLocX(int i)	: i番目の断面位置の横座標を返す。
double getScLocY(int i)	: i番目の断面位置の縦座標を返す。
String getScMemo (int l)	: i番目の断面の備考事項を返す。
String getSQL()	: この検索に用いたSQL文の全体を返す。
その他、RiverInfoBeanに含まれるpublicなメソッド	

クラス名	SectionInfoBean
概要	指定された河川の（何らかのデータを有する）断面の個別情報
親クラス	RiverInfoBean
メソッド	
void clear()	: 全プロパティのクリア
void doQuery()	: setRvId、setScIdメソッドでセットしたIDで検索の実行
void setScId(int i)、void setScId(String str)	: 対象断面の断面IDをセットする。
int getScId()	: 対象断面の断面IDを返す。
String getScName()	: 断面名称を返す。
double getScExt()	: 断面の延長距離を返す。
double getScLocX()	: 断面位置の横座標を返す。
double getScLocY()	: 断面位置の縦座標を返す。
String getScMemo ()	: 断面の備考事項を返す。
String getSQL()	: この検索に用いたSQL文の全体を返す。
その他、RiverInfoBeanに含まれるpublicなメソッド	

クラス名	PtShapeScListBean
概要	指定された河川の断面形状のデータを有する断面のリスト
親クラス	RiverInfoBean
メソッド	
void clear()	: 全プロパティのクリア
void doQuery()	: setRvIdメソッドでセットした河川IDで検索の実行
int getCount()	: 断面データの個数を返す。
int getScId(int i)	: i番目の断面IDを返す。
String getScName(int i)	: i番目の断面名称を返す。
double getScExt(int i)	: i番目の断面の延長距離を返す。
int getPtCount(int i)	: i番目の断面での断面測量点数を返す。
String getSQL()	: この検索に用いたSQL文の全体を返す。
その他、RiverInfoBeanに含まれるpublicなメソッド	

クラス名	PtShapeListBean
概要	指定された河川および断面の断面測量データの操作
親クラス	RiverInfoBean
メソッド	
void clear()	: 全プロパティのクリア
void doQuery()	: setRvId、setScIdメソッドでセットした河川IDで検索の実行
int getCount()	: 測量データの個数を返す。
int getPtId(int i)	: i番目の測量データのIDを返す。
double getPtYeta(int i)	: i番目の測量点の左岸基準点からの距離を返す。
double getPtZ(int i)	: i番目の測量点の標高データを返す。
Dtate getPtDate(int i)	: i番目の測量点の調査年月日を返す。
String getSQL()	: この検索に用いたSQL文の全体を返す。
double getMaxZ()	: この断面の最高標高値を返す。
double getMinZ()	: この断面の最低標高値を返す。
double getMaxDepth()	: この断面の取りうる最大水深を返す。
double getTotalWidth()	: この断面の河道幅を返す。
void setRough(double d)、void setRough (String str)	: この断面の粗度係数をセットする。
double getRough ()	: この断面の粗度係数を返す。
void setSlope(double d)、void setSlope (String str)	: の断面の河床勾配をセットする。
double getSlope ()	: この断面の河床勾配を返す。
void setWaterLevel(double d)、void setWaterLevel (String str)	: の断面に水位をセットする。
double getWaterLevel ()	: この断面に設定された水位を返す。
void setWaterDepth(double d)、void setWaterDepth (String str)	: の断面に水深をセットする。
double getWaterDepth ()	: この断面に設定された水深を返す。
その他、RiverInfoBeanに含まれるpublicなメソッド	

クラス名	DrawSpecBean
概要	グラフィック表示する際の描画仕様。
親クラス	なし
メソッド	
void clear()	: 全プロパティのクリア
void setMargine(int i)、void setMargine (String str)	: 描画対象コンポーネントのなかで実際に描画する領域の余白を指定する (ピクセル値)。
int getMargine()	: 描画する領域の余白 (ピクセル値) を返す。
void setLineThick(float f)、void setLineThick (String str)	: 描画する線の太さを指定する。
float getLineThick ()	: 描画する線の太さを返す。
void setAspectRatio(double f)、void setAspectRatio(String str)	: 鉛直方向にどれだけ拡大して描画するか指定する。
double getAspectRatio ()	: 鉛直方向の描画倍率を返す。
Srping getName(int i)	: i番目の並べ替え項目のWebページでの表示名を返す。

管理者関連のJava Beans

クラス名	MgListBean
概要	管理者のリストを取得する。
親クラス	なし
メソッド	
	void clear() : 全プロパティのクリア
	void doQuery() : リスト検索の実行
	int getCount() : データベースに登録されている管理者数を返す。
	int getMgId(int i) : i番目のデータの管理者IDを返す。
	String getMgName(int i) : i番目のデータの管理者名称を返す。

クラス名	OfficeListBean
概要	管理事務所のリストを取得する。
親クラス	なし
メソッド	
	void clear() : 全プロパティのクリア
	void doQuery() : リスト検索の実行
	int getCount() : 事務所の個数を返す。
	int getOfId(int i) : i番目の事務所IDを返す。
	String getOfName(int i) : i番目の事務所名称を返す。
	String getMgName(int i) : i番目の事務所の管理者名称を返す。
	void setWhere(String str) : 河川リストの検索条件をセットする。引数にはSQL構文のWHERE節 (「WHERE」は含まない) をセットする。
	String getWhere() : 河川リストの検索条件を返す。具体的には、SQL構文のWHERE節 (「WHERE」は含まない) を返す。
	String getSQL() : 河川リストの検索に用いたSQL文の全体を返す。

クラス名	OfficeInfoBean
概要	—管理事務所の個別情報の操作。
親クラス	なし
メソッド	
void clear()	: 全プロパティのクリア
void doQuery()	: getOfIdでセットした事務所の検索の実行
void doInsert()	: 新規登録の実行
void doUpdate()	: 情報更新の実行
void doDelete()	: 情報削除の実行
void setOfId(int i)、void setOfId(String str)	: 事務所IDをセットする。
int getOfId()	: 事務所IDを返す。
void setMgId(int i)、void setMgId(String str)	: 管理者IDをセットする。
int getMgId()	: 管理者IDを返す。
void setOfName(String str)	: 事務所名称をセットする。
String getOfName(String str)	: 事務所名称を返す。
void setMgName(String str)	: 管理者名称をセットする。
String getMgName(String str)	: 管理者名称を返す。
void setOfAddress(String str)	: 事務所所在地をセットする。
String getOfAddress (String str)	: 事務所所在地を返す。
void setOfMemo(String str)	: 備考データをセットする。
String getOfMemo (String str)	: 備考データを返す。
void setWhere(String str)	: 河川リストの検索条件をセットする。引数にはSQL構文のWHERE節 (「WHERE」は含まない) をセットする。
String getWhere()	: 河川リストの検索条件を返す。具体的には、SQL構文のWHERE節 (「WHERE」は含まない) を返す。
String getSQL()	: 河川リストの検索に用いたSQL文の全体を返す。

雨量観測所関連のJava Beans

クラス名	StationListBean
概要	検索条件に合う観測所のリストを取得する。
親クラス	なし
メソッド	
void clear()	全プロパティのクリア
int getCount()	検索条件に合う観測所の個数を返す。
int getStId(int i)	i番目の観測所IDを返す。
int getRvId(int i)	i番目の観測所が属する河川のIDを返す。
String getStName(int i)	i番目の観測所名称を返す。
String getRvName(int i)	i番目の観測所が属する河川の名称を返す。
void setWhere(String str)	観測所リストの検索条件をセットする。引数にはSQL構文のWHERE節(「WHERE」は含まない)をセットする。
String getWhere()	観測所リストの検索条件を返す。具体的には、SQL構文のWHERE節(「WHERE」は含まない)を返す。
void setTable(String str)	観測所リストの検索に用いるテーブル名をセットする。引数strに用いるのは、つぎの2つのどれか。 StationListBean.TABLE_RAIN : 雨量観測所の検索 StationListBean.TABLE_FLOW : 雨量観測所の検索
String getTable()	観測所リストの検索に用いるテーブル名を返す。
String getSQL()	観測所リストの検索に用いたSQL文の全体を返す。

(4) サーブレットプログラムの作成

サーブレットプログラムには、httpプロトコルでのクライアントとのやり取りに対応するために、以下のパッケージをインポートする必要がある；

- ・ java.io.*
- ・ javax.servlet.*
- ・ javax.servlet.http.*

また、さらに次のクラスを継承しておかなくてはならない：

HttpServlet

そのうえで、親クラスにある次のメソッドをオーバーライドすることで、サーブレットプログラムを記述していくことになる；

- ・ public void doGet(HttpServletRequest request,
 HttpServletResponse response)
- ・ public void doPost(HttpServletRequest request,
 HttpServletResponse response)

サーブレットプログラムは、ひとつのWebアプリケーションに1つだけでもよいし、いくつかに分割してもよい。いずれにしても、それぞれのサーブレットでよく使うメソッドは汎用化して1つのクラスにまとめておけば、いろいろと便利である。今回開発するシステムでは、実際の運用サーブレットは1つのクラスにしているが、後々のためにも、まずは汎用なサーブレット用クラスを作成しておき、それを継承して、具体的な処理を記述していくことにする。

1) 汎用サーブレットクラスの作成

今回のシステムを開発するプロセスの当初では、河川情報や事務所情報などの対象ごとにサーブレットを作成していた。その経験で、いくつかのメソッドは常に用いられていたもので、それをまとめておけば、今後のサーブレットの作成も容易になると考えた。リスト - 1 に、現時点での汎用サーブレットクラスのソースコードを示す。

リスト - 1 汎用サーブレットMyHttpServlet.java

```
// MyHttpServlet
// Servlet用の親クラス
// 以下の処理をサポートする。
// ・メッセージページの作成
// ・エラー表示ページの作成
// ・日本語文字コード変換
// ・クライアントからのリクエストに対応する処理は、
//protected void myTask(HttpServletRequest request, HttpServletResponse response)
// をオーバーライドして記述する。
//

package common; // パッケージの宣言

import java.util.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyHttpServlet extends HttpServlet{

    // doGetの処理
    public void doGet(HttpServletRequest request, HttpServletResponse response){
        myTask(request, response);
    }

    // doPostの処理
    public void doPost(HttpServletRequest request, HttpServletResponse response){
        myTask(request, response);
    }

    // doGetもdoPostもここで処理される。
    // 具体的な処理は、このメソッドをオーバーライドして記述する。
    protected void myTask(HttpServletRequest request, HttpServletResponse
response){
        showMessage("SvHome", response);
    }

    // メッセージ表示用ページの作成
    protected void showMessage(String myMes, HttpServletResponse response){
        try{
            response.setContentType("text/html; charset=Shift_JIS");
            PrintWriter pw=response.getWriter();
            pw.println(this + " : " + myMes);
        }catch(Exception ex){showError("showMessage:"+myMes, ex, response);}
    }

    // エラー表示用ページの作成
```

```

protected void showError(
    String myStr, Exception ex, HttpServletResponse response){
    try{
        response.setContentType("text/html; charset=Shift_JIS");
        PrintWriter pw=response.getWriter();
        pw.println("エラー in " + this + " : " + myStr + "<br>");
        pw.println(ex + "<br>");
        ex.printStackTrace(pw);
        pw.println("「戻る」ボタンで戻ってください");
    }catch(Exception exx){}
}

// 日本語文字コードの変換
protected String jpEncode(String myStr) throws UnsupportedEncodingException{
    if (myStr==null){
        return null;
    }else{
        return (new String(myStr.getBytes("ISO-8859-1"),"JISAutoDetect"));
    }
}

}////////// End of Class //////////

```

2) 運用サーブレットの作成

今回開発するシステムの具体的な運用サーブレットプログラムは、以下のような方針で記述することにする。

サーブレットで一般的によく用いられる処理は、汎用クラスMyHttpServletクラスの中に書き、それをMyHttpServletクラスを継承することで、利用する。

クライアントの要求内容は、リクエストヘッダの「action」パラメータを参照することで判断する。「action」パラメータの値ごとに、処理内容を振り分ける。

「action」の中に細分化された処理を求める場合もある。これもリクエストヘッダ上のあるキーに値がセットされているかどうかで判断する。主たるキー名称は以下の通りである。

キー名称	意味
show	検索条件を設定して検索結果をWebページに表示する。
renew	先と同じ検索条件での検索結果をWebページに表示する。
execute	表示したページに関連した操作を実行する。
cancel	表示したページに関連した操作は実行しない。

河川の断面形状は、それぞれグラフ化したものを画像ファイルとして保存しておくのではなく、呼び出された際に画像ファイルのデータを自動的に生成し、これをクライアントに送るものとする。これにはACMEのGIFエンコーダ用クラスを利用することにする。


```

        if("RiverSearch".equals(action)){
            myUrl = "/river/RiverSearch.jsp";

//////////
////////// 河川を検索して検索結果をリスト表示
        }else if("RiverList".equals(action)){
            // 河川検索用Beanの取得
            RiverListBean myRvList =
                (RiverListBean) (mySession.getAttribute("myRvList"));
            if (myRvList==null){myRvList=new RiverListBean();}
            // ソート項目用Beanの取得
            RiverSortBean myRvSort=
                (RiverSortBean)(mySession.getAttribute("myRvSort"));
            if (myRvSort==null){myRvSort=new RiverSortBean();}

            if(request.getParameter("renew")==null){
                // 「最新情報」の指定がなければ、検索条件を設定し直す。
                myRvList.clear();

                String myRvName =
                    jpEncode(request.getParameter("RvName"));
                int myWsId;
                try{myWsId
Integer.parseInt(request.getParameter("WsId"));
                }catch(Exception e){myWsId=0;}
                int myMgId;
                try{myMgId
Integer.parseInt(request.getParameter("MgId"));
                }catch(Exception e){myMgId=0;}

                String myWhere="";
                if(!myRvName.equals("")){
                    myWhere=myWhere+
                        "and ( River.Name like '%" +myRvName+"%' ) ";}
                if(myWsId != 0){
                    myWhere = myWhere +
                        "and ( System.WsId = " + myWsId + " ) ";}
                if(myMgId != 0){
                    myWhere = myWhere +
                        "and ( Manager.MgId = " + myMgId + " ) ";}

                myRvList.setWhere(myWhere.substring(4));}

            int mySort= Integer.parseInt(request.getParameter("Sort"));
            if(mySort != 0){
                myRvList.setSortField(myRvSort.getField(mySort));
                myRvList.setSortName(myRvSort.getName(mySort));
            }
        }
}

```

```

// 河川の検索を実行
myRvList.doQuery();

// セッションに検索結果リストをセットする。
mySession.setAttribute("myRvList",myRvList);
// 表示用JSPのURL名をセット
myUrl="/river/RiverList.jsp";

////////////////////////////////////
//////// 指定された一河川の詳細情報を表示
}else if("RiverInfo".equals(action)){
// 河川詳細情報用のBeanの取得
RiverInfoBean myRvInfo=
(RiverInfoBean)(mySession.getAttribute("myRvInfo"));
if (myRvInfo==null){myRvInfo=new RiverInfoBean();}
// 雨量観測所リスト用のBeanの取得
StationListBean myRainList=
(StationListBean)(mySession.getAttribute("myRainList"));
if (myRainList==null){myRainList=new StationListBean();}

if(request.getParameter("renew")==null){
// 「最新情報」の指定がなければ、検索条件を設定し直す。
myRvInfo.setRvId(request.getParameter("RvId"));
myRainList.clear();
myRainList.setTable(StationListBean.TABLE_RAIN);
myRainList.setWhere(
StationListBean.TABLE_RAIN+
".RvId="+myRvInfo.getRvId());
}
// 検索の実行
myRvInfo.doQuery();
myRainList.doQuery();
// 検索結果をセッションにセット
mySession.setAttribute("myRvInfo",myRvInfo);
mySession.setAttribute("myRainList",myRainList);
// 表示用JSPのURL名をセット
myUrl = "/river/RiverInfo.jsp";

////////////////////////////////////
//////// 河川情報の新規登録
}else if("RiverInsert".equals(action)){
// 新規登録用ページの表示
if(request.getParameter("show")!=null){
myUrl = "/river/RiverInsert.jsp";

// 新規登録の実施
}else if(request.getParameter("execute")!=null){
// 河川詳細情報用のBeanの取得
RiverInfoBean myRvInfo=
(RiverInfoBean)(mySession.getAttribute("myRvInfo"));

```

```

if (myRvInfo==null){myRvInfo=new RiverInfoBean();}
myRvInfo.clear();//情報のクリア
//新規登録用ページに入力された内容をBeanにセット
myRvInfo.setRvName(
    jpEncode(request.getParameter("RvName")));
myRvInfo.setWsId(request.getParameter("WsId"));
myRvInfo.setOfId(request.getParameter("OfId"));
myRvInfo.setRvArea(request.getParameter("RvArea"));
myRvInfo.setRvOrigin(
    jpEncode(request.getParameter("RvOrigin")));
myRvInfo.setRvTerm(
    jpEncode(request.getParameter("RvTerm")));
myRvInfo.setRvMemo(
    jpEncode(request.getParameter("RvMemo")));

//Beanの新規登録メソッドの実施(戻り値はエラー状況)
String myInsert=myRvInfo.doInsert();
if(myInsert.equals("")){
    mySession.setAttribute("myRvInfo",myRvInfo);
    myUrl="/river/RiverInfo.jsp";

}else{
    showMessage("MainServlet:RiverInsert.execute:"+
        myInsert,response);
    return;
}

//新規登録のキャンセル
}else if(request.getParameter("cancel")!=null){
    myUrl=myName+"?action=RiverList&renew=true";

}

```

////////////////////////////////////

///// 河川情報の更新

```

}else if("RiverUpdate".equals(action)){

```

```

//更新用ページの表示

```

```

if(request.getParameter("show")!=null){

```

```

//河川詳細情報用のBeanの取得

```

```

RiverInfoBean myRvInfo=

```

```

(RiverInfoBean)(mySession.getAttribute("myRvInfo"));

```

```

if (myRvInfo==null){myRvInfo=new RiverInfoBean();}

```

```

//河川詳細情報の検索

```

```

myRvInfo.setRvId(request.getParameter("RvId"));

```

```

myRvInfo.doQuery();

```

```

//表示用JSPのURL名をセット

```

```

myUrl = "/river/RiverUpdate.jsp";

```

```

//情報更新の実施

```

```

}else if(request.getParameter("execute")!=null){
    // 河川詳細情報用のBeanの取得
    RiverInfoBean myRvInfo=
        (RiverInfoBean)(mySession.getAttribute("myRvInfo"));
    if (myRvInfo==null){myRvInfo=new RiverInfoBean();}
    myRvInfo.clear();

    // 更新用ページに入力された内容をBeanにセット
    myRvInfo.setRvName(
        jpEncode(request.getParameter("RvName")));
    myRvInfo.setWsId(request.getParameter("WsId"));
    myRvInfo.setOfId(request.getParameter("OfId"));
    myRvInfo.setRvArea(request.getParameter("RvArea"));
    myRvInfo.setRvOrigin(
        jpEncode(request.getParameter("RvOrigin")));
    myRvInfo.setRvTerm(
        jpEncode(request.getParameter("RvTerm")));
    myRvInfo.setRvMemo(
        jpEncode(request.getParameter("RvMemo")));

    // Beanの更新メソッドの実施（戻り値はエラー状況）
    String myUpdate=myRvInfo.doUpdate();
    if(myUpdate.equals("")){
        mySession.setAttribute("myRvInfo",myRvInfo);
        myUrl= myName+"?action=RiverInfo&RvId="+
            myRvInfo.getRvId();

    }else{
        showMessage("MainServlet:RiverUpdate.execute:"+
            myUpdate, response);
        return;
    }

    // 情報更新のキャンセル
}else if(request.getParameter("cancel")!=null){
    myUrl="/river/RiverInfo.jsp";

}

////////////////////////////////////
///// 管理事務所リストの表示
}else if("OfficeList".equals(action)||"OfficeSearch".equals(action)){

    // 事務所リスト用のBeanの取得
    OfficeListBean myOfList=
        (OfficeListBean)(mySession.getAttribute("myOfList"));
    if (myOfList==null){myOfList=new OfficeListBean();}

    // 事務所の検索
    myOfList.clear();

```

```

myOfList.setWhere("Office.Name <> ''");
myOfList.doQuery();
// 検索リストをセッションにセット
mySession.setAttribute("myOfList",myOfList);
myUrl="/office/OfficeList.jsp";

////////////////////////////////////
///// 管理事務所情報の詳細表示
}else if("OfficeInfo".equals(action)){
// 事務所詳細情報用のBeanの取得
OfficeInfoBean myOfInfo=
    (OfficeInfoBean)(mySession.getAttribute("myOfInfo"));
if (myOfInfo==null){myOfInfo=new OfficeInfoBean();}
// 河川リスト用のBeanの取得
RiverListBean myRvList=
    (RiverListBean)(mySession.getAttribute("myRvList"));
if (myRvList==null){myRvList=new RiverListBean();}

if(request.getParameter("renew")==null){
// 「最新情報」指定がなければ検索条件を設定し直す。
myOfInfo.setOfId(request.getParameter("OfId"));
myRvList.clear();
myRvList.setWhere("River.OfId = "+
    request.getParameter("OfId"));
}
// 事務所詳細情報の検索
myOfInfo.doQuery();
// 事務所が管理する河川の検索
myRvList.doQuery();
// 検索結果をセッションにセット
mySession.setAttribute("myOfInfo",myOfInfo);
mySession.setAttribute("myRvList",myRvList);

myUrl = "/office/OfficeInfo.jsp";

////////////////////////////////////
///// 河川断面リストの表示
}else if("SectionList".equals(action)){
SectionListBean myScList=
    (SectionListBean)(mySession.getAttribute("myScList"));
if (myScList==null){myScList=new SectionListBean();}

if(request.getParameter("renew")==null){
myScList.clear();
myScList.setRvId(request.getParameter("RvId"));
}
myScList.doQuery();

mySession.setAttribute("myScList",myScList);

```

```

        myUrl="/section/SectionList.jsp";

//////////
////////// 河川断面詳細情報の表示
        }else if("SectionInfo".equals(action)){
            SectionInfoBean myScInfo=
                (SectionInfoBean)(mySession.getAttribute("myScInfo"));
            if (myScInfo==null){myScInfo=new SectionInfoBean();}

            if(request.getParameter("renew")==null){
                myScInfo.clear();
                myScInfo.setRvld(request.getParameter("Rvld"));
                myScInfo.setScld(request.getParameter("Scld"));
            }
            myScInfo.doQuery();

            mySession.setAttribute("myScInfo",myScInfo);
            myUrl="/section/SectionInfo.jsp";

//////////
////////// 断面形状データを有する断面のリスト表示
        }else if("PtShapeScList".equals(action)){
            PtShapeScListBean myPSScList=
                (PtShapeScListBean)(mySession.getAttribute("myPSScList"));
            if (myPSScList==null){myPSScList=new PtShapeScListBean();}

            if(request.getParameter("renew")==null){
                myPSScList.clear();
                myPSScList.setRvld(request.getParameter("Rvld"));
            }
            myPSScList.doQuery();

            mySession.setAttribute("myPSScList",myPSScList);
            myUrl="/shape/PtShapeScList.jsp";

//////////
////////// 断面形状データのリスト表示
        }else if("PtShapeList".equals(action)){
            if(request.getParameter("redraw")==null){
                PtShapeListBean myPtList=
                    (PtShapeListBean)(mySession.getAttribute("myPtList"));
                if (myPtList==null){myPtList=new PtShapeListBean();}

                if(request.getParameter("renew")==null){
                    myPtList.clear();
                    myPtList.setRvld(request.getParameter("Rvld"));
                    myPtList.setScld(request.getParameter("Scld"));
                }

                String myList=myPtList.doQuery();

```

```

        if(myList.equals("")){
            mySession.setAttribute("myPtList",myPtList);
        }else{
            showMessage("MainServlet:PtshapeList:"+
                myList,response);
            return;
        }
    }

    DrawSpecBean myDraw=
        (DrawSpecBean)(mySession.getAttribute("myDraw"));
    if (myDraw==null){myDraw=new DrawSpecBean();}
    if(request.getParameter("ARatio")!=null){
        myDraw.setAspectRatio(request.getParameter("ARatio"));}
    if(request.getParameter("LThick")!=null){
        myDraw.setLineThick(request.getParameter("LThick"));}
    if(request.getParameter("Margine")!=null){
        myDraw.setMargine(request.getParameter("Margine"));}
    mySession.setAttribute("myDraw",myDraw);

    myUrl = "/shape/PtShapeList.jsp";

```

```

////////////////////////////////////

```

```

///// 断面形状の描画と出力

```

```

    }else if("PtShapeDraw".equals(action)){
        // 断面形状データリスト用のBeanの取得
        PtShapeListBean myPtList=
            (PtShapeListBean)(mySession.getAttribute("myPtList"));
        if (myPtList==null){
            showMessage("MainServlet:PtshapeDraw:No Section Queried.
                ",response);
            return;
        }
    }

```

```

// 断面特性の取得

```

```

double dblTotalWidth=myPtList.getTotalWidth();
double dblMinZ=myPtList.getMinZ();
double dblMaxZ=myPtList.getMaxZ();
double dblMaxDepth=myPtList.getMaxDepth();

```

```

// 断面形状描画仕様用のBeanの取得

```

```

DrawSpecBean myDraw=
    (DrawSpecBean)(mySession.getAttribute("myDraw"));
if (myDraw==null){myDraw=new DrawSpecBean();}
// 断面形状描画仕様の設定
float LThick=myDraw.getLineThick(); //線の太さ
double dblARatio=myDraw.getAspectRatio();
int Margine=myDraw.getMargine();

```

```

//ユーザー座標枠の設定

```

```

double WX1,WY1,WX2,WY2,DeIWX,DeIWy;

```

```

WX1=0.;
WY1=dblMinZ;
WX2=dblTotalWidth;
WY2=dblMaxZ;
DeIWX=WX2-WX1;
DeIWY=WY2-WY1;

//画像座標枠の設定
int viewWidth=1000;
int viewHeight=(int) (viewWidth*(DeIWY/DeIWX)*dblARatio);
int VX1,VY1,VX2,VY2,DeIVX,DeIVY;
VX1=Margine;
VY1=Margine;
VX2 = viewWidth-Margine;
VY2 = viewHeight-Margine;
DeIVX=VX2-VX1;
DeIVY=VY2-VY1;

//中間座標枠の設定
double SX, SY;
if (DeIVX < DeIVY){
    SX = 1; SY = SX * DeIVY / DeIVX;
} else {
    SY = 1; SX = SY * DeIVX / DeIVY;
}
double SX1,SY1,SX2,SY2,DeISX,DeISY;
SX1 = -LThick/2;
SY1 = -LThick/2;
SX2 = SX + LThick/2;
SY2 = SY + LThick/2;
DeISX = SX2 - SX1;
DeISY = SY2 - SY1;

//グラフィックコンテキストの取得
Frame f =new Frame();
f.addNotify();
Image myImage=f.createImage(viewWidth,viewHeight);

Graphics2D myGrp = (Graphics2D) myImage.getGraphics();

//ユーザー座標系から中間座標系への座標変換
AffineTransform myTransform = new
    AffineTransform(
        SX/DeIWX, 0., 0., SY/DeIWY,
        -SX/DeIWX*WX1, -SY/DeIWY*WY1);
//中間座標系からグラフィック座標系への座標変換
myGrp.setTransform(new AffineTransform(
    DeIVX/DeISX, 0., 0., -DeIVY/DeISY,
    (VX1*SX2-VX2*SX1)/DeISX, (VY2*SY2-VY1*SY1)/DeISY));

```



```

//縦軸と横軸
myGrp.setStroke(new BasicStroke(LThick));
myGrp.setColor(Color.blue);
myGrp.draw(new Line2D.Double(0,0,0,SY));
myGrp.draw(new Line2D.Double(0,0,SX,0));

//断面形状グラフ
myGrp.setStroke(new BasicStroke(2*LThick));
myGrp.setColor(Color.red);

int Cnt=myPtList.getCount();
    Point2D.Double pntSrc = new Point2D.Double
        (myPtList.getPtYeta(0),myPtList.getPtZ(0));
    myTransform.transform(pntSrc, pntSrc);
for(int i=0; i<Cnt; i++){
    Point2D.Double pntDst = new Point2D.Double
        (myPtList.getPtYeta(i),myPtList.getPtZ(i));
    myTransform.transform(pntDst, pntDst);
    myGrp.draw(new Line2D.Double(pntSrc, pntDst));
    pntSrc.setLocation(pntDst);

}

//GIF形式で出力
response.setContentType("image/gif");
OutputStream myOut=response.getOutputStream();
try{
    new GifEncoder(myImage,myOut).encode();
}catch(IOException e){
    showMessage("MainServlet:PtShapeDraw:"+e, response);
}
myOut.flush();
return;

}

// 応答ページへの振り分け
ServletContext sc = getServletContext();
sc.getRequestDispatcher(myUrl).forward(request, response);

}catch(Exception ex){showError("MainServlet", ex, response);}
}
}

```

(5) JSPの作成

JSP (Java Server Pages) はいわばHTMLファイルの中に、Java言語を埋め込んで動的なWebページを生成させる仕組みである。動的にするための様々なデータはJava Beansを介して取得することにする。以下にWebページの説明と表示例を示す。

1) HOME

HOMEページの表示例を図 - 12に示す。このページから事務所リストあるいは河川検索ページにジャンプする。

河川データベース+α on Web

今回の工事



[河川検索](#)

[事務所検索](#)

図 - 12 WebデータベースHOME

2) 事務所リスト

事務所リストページの表示例を図 - 13に示す。データの取得にはOfficeListBeanを用いている。事務所名称の各個には、その詳細情報ページへのリンクが張られている。

事務所一覧	
河川検索	HOME
最新情報	新規登録
10件のデータがあります。	
管理機関	事務所名称
宇都宮市	C事務所
宇都宮市	F事務所
宇都宮市	I事務所
国土交通省	A事務所
国土交通省	D事務所
国土交通省	G事務所
国土交通省	J事務所
栃木県	B事務所
栃木県	E事務所
栃木県	H事務所

図 - 13 管理事務所リスト表示例

3) 事務所詳細情報

事務所詳細情報ページの表示例を図 - 14に示す。事務所情報の取得にはOfficeInfoBeanを、事務所が管理する河川リストはRiverListBeanを用いている。その河川名称の各個には、その詳細情報ページへのリンクが張られている。

事務所詳細情報

[河川検索](#) [事務所検索](#) [HOME](#)

[最新情報](#) [新規登録](#)
[情報更新](#) [削除](#)

事務所名称	I事務所
管理機関	宇都宮市
事務所住所	栃木県宇都宮市陽東7-1-2
備考	宇都宮大学工学部のとなり

30件の河川を管理しています。

河川名称
梓川
滑川
亀久川
苦戸川
荒金沢川
高野川
黒川
今尾頭川

図 - 14 管理事務所詳細情報表示例

4) 河川検索

河川検索条件の設定ページの表示例を図 - 15に示す。河川名、水系名、管理者名を指定して検索させることができる。水系名、管理者名、並べ替え項目はそれぞれ、WsListBean、MgListBean、RvSortBeanからデータを取得して、ドロップダウンリストから選択するようにしている。

河川データベース+α on Web

[事務所検索](#)

[HOME](#)

河川を検索して一覧表を表示します。

河川名称	<input type="text"/>	を含む
水系名称	利根川	▼
管理機関	栃木県	▼
並べ替え		▼
	水系 管理機関 事務所名 流域面積 源流 末端	Go!

図 - 15 河川検索条件設定ページ

5) 河川リスト

河川検索の結果をリスト表示する。その表示例を図 - 16に示す。リスト中の河川名称の各個には、その詳細情報ページへのリンクが張られている。またこのページから河川検索ページや事務所リストページにジャンプすることもできる。



検索条件に合う河川は、175件あります。

No.	河川名称	水系	管理機関
1	シャジ沢川	利根川	宇都宮市
2	ネベ沢川	利根川	宇都宮市
3	ハタ沢川	利根川	宇都宮市
4	悪至沢川	利根川	宇都宮市
5	稲ヶ沢川	利根川	宇都宮市
6	逆川(藤)	利根川	宇都宮市
7	久蔵川	利根川	宇都宮市
8	江戸川	利根川	宇都宮市
9	黒川	利根川	宇都宮市
10	三河沢川	利根川	宇都宮市
11	姿川	利根川	宇都宮市
12	思川	利根川	宇都宮市
13	小百川	利根川	宇都宮市

図 - 16 河川検索結果の表示例

6) 河川詳細情報

指定された河川の詳細情報を表示する。その表示例を図 - 17に示す。リスト中の河川名称の各個には、その詳細情報ページへのリンクが張られている。

新規登録ボタン、削除ボタン、更新ボタンをクリックすることで、それぞれの操作にかかわるページにジャンプする。断面データがある場合には、断面リスト表次ページへジャンプするためのボタンも表示される。

またこのページから河川検索ページや事務所リストページにジャンプすることもできる。

河川詳細情報			
河川検索	事務所検索	HOME	
最新情報	新規登録	情報更新	削除
河川名称	姿川		
水系	利根川		
管理機関	宇都宮市		
管理事務所	○事務所		
流域面積	206.6		
源流			
終端	思川		
備考			
断面位置情報	155断面の位置情報があります。		一覧表示
雨量観測所	この河川の流域内にはありません。		

図 - 17 河川詳細情報の表示例

7) 河川情報登録、更新、削除

それぞれの表示例を図 - 18、19、20に示す。データの取得・操作には RiverInfoBeanが用いられている。

河川情報の登録

[河川検索](#) [事務所検索](#) [HOME](#)

新しいデータを入力してください

河川名称	<input type="text"/>	
水系	<input type="text" value="▼"/>	
管轄事務所	<input type="text" value="▼"/>	
流域面積	<input type="text"/>	
源流	<input type="text"/>	
終端	<input type="text"/>	
備考	<input type="text"/>	

図 - 18 河川情報の新規登録画面

河川情報の更新

[河川検索](#) [事務所検索](#) [HOME](#)

データを更新してください

河川名称	姿川
水系	利根川
管轄事務所	宇都宮市〇事務所
流域面積	206.6
源流	
終端	思川
備考	

図 - 19 河川情報の更新画面の表示例

河川情報の削除

[河川検索](#) [事務所検索](#) [HOME](#)

このデータを削除してよいですか

河川名称	姿川
水系	利根川
管轄事務所	〇事務所
流域面積	206.6
源流	
終端	思川
備考	

図 - 20 河川情報の削除画面の表示例

8) 河川断面リスト

指定された河川の断面リストを表示する。その表示例を図 - 21に示す。データの取得にはSev tionListBeanが用いられている。リスト中の断面名称の各個には、その詳細情報ページへのリンクが張られている。また断面形状や植生データが存在する断面のリストを表示させるためのボタンも配置されている。

断面一覧

河川検索
事務所検索
HOME

河川名称	姿川
水系	利根川
管理機関	宇都宮市
管理事務所	〇事務所

最新情報

新規登録

断面形状モデル一覧

断面形状データ一覧

断面植生データ一覧

この河川の断面位置情報は、155件あります。

No.	断面名称	延長距離	位置座標X	位置座標Y	備考
1	224+00	0.0	58531.532	611.0885	
2	224+25	25.0	58553.8594	622.3352	
3	224+50	50.0	58576.1868	633.5818	
4	224+75	75.0	58598.562	644.6899	
5	225+00	100.0	58621.7751	653.9936	
6	225+25	125.0	58645.7812	660.9354	
7	225+50	150.0	58670.3603	665.446	
8	225+75	175.0	58695.267	667.4902	
9	226+00	200.0	58720.2523	667.0177	
10	226+25	225.0	58745.0666	664.0631	
11	226+50	250.0	58769.4619	658.646	
12	226+75	275.0	58793.1946	650.8205	
13	227+00	300.0	58816.0275	640.6648	
14	227+25	325.0	58837.7324	628.2803	

図 - 21 断面リスト画面の表示例

9) 断面詳細情報

指定された河川の指定された断面の詳細なデータを表示する。その表示例を図 - 22に示す。データの取得にはSectionInfoBeanを用いている。この断面について閲覧可能なデータ（たとえば断面測量データ、植生調査データ）は、下方にメニューボタンとして表示されているので、これをクリックすればよい。河川や管理事務所の詳細情報へもジャンプできるようになっている。

河川断面詳細情報

[河川検索](#) [事務所検索](#) [HOME](#)

姿川（利根川水系）

Go!

宇都宮市 C事務所

Go!

断面名称	225+25
延長距離	125.0
X座標	58645.7812
Y座標	660.9354
備考	

断面一覧表示

断面形状モデル

断面形状データ

断面植生データ

各点植生データ

図 - 22 断面詳細情報の表示例

10) 断面形状リスト

指定された河川において断面測量データが格納されている断面リストを表示する。その表示例を図 - 23に示す。データの取得にはPtShapeScListBeanが用いられている。

リスト中の断面名称の各個には、その断面形状を実際に示すページへのリンクが張られている。

またこのページから河川検索ページや事務所リストページにジャンプすることもできる。

断面形状データ一覧

河川検索
事務所検索
HOME

河川名称	姿川
水系	利根川
管理機関	宇都宮市
管理事務所	○事務所

最新情報
新規登録

断面形状モデル一覧
断面植生データ一覧

この河川の断面形状データは、28断面について、あります。

No.	断面名称	延長距離	測量点数
1	225+00	100.0	13
2	225+25	125.0	13
3	225+50	150.0	12
4	225+75	175.0	13
5	226+00	200.0	13
6	226+25	225.0	13
7	226+50	250.0	13
8	226+75	275.0	11
9	227+00	300.0	13
10	227+25	325.0	13
11	227+50	350.0	13
12	227+75	375.0	12
13	228+00	400.0	10
14	228+25	425.0	10
15	228+50	450.0	10
16	228+75	475.0	10
17	229+00	500.0	9

図 - 23 断面形状リストの表示例

11) 断面形状データ

指定された河川および断面での断面測量データを表示する。その表示例を図 - 24に示す。データの取得はPtShapeListBeanを用いている。またこのページでは、サブレットで自動生成された断面形状グラフも表示される。その図の鉛直方向倍率や線の太さ、マージンなどはユーザー側で変更することができる。



図 - 24 断面形状データの表示例

5 . 総 括

(1) 結論

本研究では、中小河川のデータベースを整備するに当たり、リレーショナルデータベース (RDMS)、地理情報システム (GIS)、Webデータベース (WebDB) の利用について、基礎的な検討を加えた。その結果得られた知見は以下の通りである。

栃木県の中小河川全体を対象にした水系、河川、雨量観測所、流量観測所と姿川一河川のみを対象にした断面でのケーススタディを通じて、河川管理の支援を目的とした中小河川データベースの構築の方向性を示した。データベースを作成するのにテーブル構成を決め、テーブルを階層的に分けて作成したことにより、河川環境整備について、より利用しやすいデータベースを作成することができた。

中小河川データベースは、表 - 2 に示したようにデータ項目のみならず取得機関も多岐に渡る。また、その多くが電子データファイル化されておらず新たにデータ入力する煩雑さが伴う。今後、取得されたデータは当然、電子化するとともに、過去のデータも、ことあるごとに (履歴調査などの名目で) 電子化を進めていくべきである。

GISを利用することにより、観測地点等の位置情報がGISに入力された時点で、その後どのような縮尺の地図を作成しても、その位置を正確に表示できる。座標を伴わない単なる絵としての地図とは異なり、GISのモニター画面上では、観測地点等の位置情報を正確に表示しつつ、地図の必要な部分の拡大、縮小等が自在にできるため、位置情報の正確な把握に有効である。

GISをデータベース全体とリンクさせることにより、GIS上でデータベースを活用することができた。このことにより、多数のデータをGIS上で参照できるようになり、お互いに無関係であったデータであっても、地理的に近い位置にあれば、何らかの関連性を認識しながら河川環境整備に貢献できるという可能性を示すことができた。

データベースを、WebDBとして発展させることで、インターネットに接続できれば (当然Webブラウザも必要) いつでもどこでもデータを参照できる。その一例を示すことができた。

特に今回のWebDBシステムはJava言語で開発している。システム全体をMVC (モデル、ビュー、コントローラ) 構成とし、モデル部分ではさまざまな機能をJava Beansとして切り分け、コントローラにはJava Servletを用いてクライアントからの要求に対応してJava Beansを呼び出し、それをビュー部分のJSP (Java Server Pages) で表示する。このように役割分担をはっきりさせることで、効率的なシステムの開発とメンテナンスが可能であることが認識できた。

また、単に数値情報をそのまま表示するだけではなく、Java言語のさまざまな機能ならびに各所で開発されているプログラムを用いて、複雑な計算やグラフ表示が可能であることがわかった。これによって、WebDBが単なるデータを見せるだけのものではなく、そのデータを下にユーザーが分析を重ねたり、対話的に設計を支援するシステムを構築することが、それほど困難ではないことがわかった。

(2) 今後の課題

本研究は栃木県の水系、河川、雨量観測所、流量観測所、姿川一河川の断面データのみで行ったが、取り扱っていないデータはまだ、多様にある。データ項目のみならずデータ取得機関も多岐に渡る。数多くの部署でのデータを統合し、メンテナンスを行っていくことが大切である。そのため今後は、現在の河川環境整備に関わる行政機関業務や地域住民活動の流れを見直し、それらの業務や活動に直結した負荷の少ない効率的なデータベースを構築していかななくてはならない。

また、GISでのデータベースの活用においては、今回はデータベースをエクセルに変換してからのポイント作成と、ポイント作成した個所についてクリックするとアクセスのデータベース全体にリンクさせるようにしたが、今後はクリックしたポイントに関する情報を直接表示させるようにしたい。データベースの構築にあたって、GIS上で作業をしデータベースを参照できるようにするのがよいのか、データベース上でGISを参照できるようにしたほうがよいのかも検討していきたい。

WebDBに関しては、今後、さまざまなデータをもとに地域住民にわかりやすく表示したり、多自然型河道整備計画を支援したり、その可能性をさらに検討し実証していく必要がある。本研究を通じて、ユーザーにわかりやすいインターフェースとして、計算結果を図示したり、高度な計算をJava Beansとして切り分けたりすることは、もはや容易に可能であることが判明した。これもJava言語がオブジェクト指向プログラミングをサポートし、インターネット技術に密着しているために可能なことであろう。

ただし、Webアプリケーションの開発にともなうデバッグ作業には、かなりの苦痛が伴うことも実感として判明した。今回は特に、基本的なAPIとエディタで構築したプログラミング環境で開発を進めたので、なおさらであった。現時点では市販品として、さまざまなJavaアプリケーション開発環境 (Java IDE) が出回っているので、それを有効活用すべきであろう。

また、フリーなものとしてもStrutsに代表されるJakartaプロジェクトによるWebアプリケーション構築のためのフレームワークなど、さまざまなツールが開発されてきている。こうしたツールを有効活用することによって、生産性が向上していくものと思われる。

【参考文献】

- 1) 安陪和雄、和田一斗、佐藤一幸、寺川陽、浦野隆：GISとインターネットの連携による流域水環境データベースの構築、第23回土木情報システム講演集、pp43 - 46、1998.
- 2) 河原能久：水理・水文分野におけるデータベースの重要性と問題点、河川・水文・水資源情報のデータベース化と共同利用に関する研究、土木学会学術振興基金による助成「特別枠」研究成果報告書、pp77 - 79、2000.
- 3) 超図解ACCESS2000forWindows 基礎編、エクスメディア、1996.
- 4) 高羽実：はじめてのACCESS2000、秀和システム、1998.
- 5) 美しい山河を守る災害復旧基本方針、社団法人 全国防災協会、2000.
- 6) 谷尻かおり：ACCESS2000徹底入門、技術評論社、1998.
- 7) まちと水辺に豊かな自然を、財団法人 リバーフロント整備センター、1994.
- 8) MapInfo Professional ユーザーズガイド
- 9) MapInfo Professionalリファレンス
- 10) Maty Hall (岩屋宏 訳): コア・サーブレット&JSP、ソフトバンクパブリッシング、2002.