

ভার্শন ব্যাবধান : ডিবেজ-৩ + হতে ডিবেজ - ৪

এরিক ডি সিদ্ধা (হবিন)

আজ হতে প্রায় চার বছর আগে যখন বোল্ড্যাও ইন্টারন্যাশনাল, হ্যাটসওয়্যার, স্ট্রাগিওন সফটওয়্যার, ডাটাইজ ইন্টারন্যাশনাল, ওরাকল কর্পোরেশন, সাইমেন্টেক সফটওয়্যার এবং সফটওয়্যার পারফরম্যান্স কোম্পানীগুলো তাদের Paradox, Professional Developer, Data ease, Oracle (SQL), Fox base+, Foxpro এবং Profile এর নিত্য নতুন ভার্সন বের করে ডাটা এন্ট্রির বাজার দখল করে নিলেছিল তখন AshtonTate এর নীরবতার অনেকই অবাক হয়েছেন। অর্থাৎ AshtonTate এর নিজস্ব শ্যাভেরেটরীতে বসে শতাধিক প্রোগ্রামার টিকিই যোগানে তাদের আবিষ্কার চালিয়ে থাকতেন। হঠাৎ করেই বাজারে দেখা দিল ডিবেজ নামাচারের চরম ভার্সনটি। ডিবেজ-৪ এর প্রথম সিরিয়ালটি প্রকাশের পরপরই এর বিক্রি বেড়ে যায় হু হু করে। কিছু দিনের ভিতরই এই বিক্রি ডিবেজের সমষ্টিগতহার মার্কেটে প্রথম স্থানে উঠে আসে। Software Digest, e Infoworld সহ অনেক নামকরা কম্পিউটার জার্নালই তাদের প্রোগ্রামার টেক-ই-ডিবেজ-৪কে PERFECT বলে ঘোষণা করে। আর এভাবেই পুরোনো বাজার নতুন করে ব্যক্তিগত কলন Ashton Tate তার 'আম্বল-৪' নিয়ে।

"Advancing the world wide standard in data management and application development" এই নীতিটি সামনে রেখে ডিবেজ-৪ এর ১.১ নং সিরিয়ালটি বের হয় ১৯৯১ সালের দিকে। আমেরিকায় প্রথমে এর মূল্য ধরা হয় ১১৯ ডলার। বাজারে এদের ১০.৫ সিরিয়াল চলাই এবং এর স্থানীয় মূল্য প্রায় ১০,০০০ টাকা।

ইথিওপেটেড ডেভেলপমেন্ট এনভায়রনমেন্ট ডিবেজ-৪ ডিবেজ-৩+ অপেক্ষা অনেক সাবলীল এবং এর ইউজার ইন্টারফেস বেশ উন্নত। আবার শ্যাভেরেজ প্রোগ্রামিং-এ ডিবেজ-৪ এর সাথে ডিবেজ-৩+ এর তুলনা করা সম্ভব নয়। পুরো প্যাকেজটি যে সকল নতুন নতুন ফীচার নিয়ে আবির্ভূত হয়েছে তার মধ্য রয়েছে একটি নতুন বাহার ইউজার ইন্টারফেস। তিন শরীরে অধিক নতুন কমাণ্ড ও ফাংশন, একটি SQL (Structured Query Language) ইন্টারফেস, একটি Templet সার্ভিসেজ, WYSIWYG মুদ্রণের জন্য উইনডোজিং এবং মেনু বিকি। টেমপ্লেট ম্যাক্রোজটি গ্রি-পাট, ফেনেল এবং স্ক্রীন ফর্ম তৈরীর জন্য ব্যবহার করা যায়। ডিবেজ-৪ এর নতুন আরো যা রয়েছে, তা হল একটি এন্ট্রিকেশন ডেভেলপার, DOCUMENT.GEN নামে একটি প্রোগ্রাম ডেভেলপার এবং FORMBROW.GEN নামে একটি ফর্ম ডেভেলপার যা প্যাকেট টেমপ্লেট সাপেক্ষে তার চাইড টেমপ্লেট স্ক্রীন ফর্মে Browse করতে দেয়।

প্রাথমিক আলোচনা এবং ভূমিকা :

ডিবেজ-৩+ এর পরিচিত ডট প্রপার্টি, Assist ৩+ এবং প্রোগ্রামিং পরিবেশ হতে ডিবেজ-৪ এর উন্নততর পরিবেশ প্রবেশকে সেই দীর্ঘ পদক্ষেপের সাথে তুলনা করা যায় যা আপনাকে ডিবেজ-২ হতে

ডিবেজ-৩-এ যাওয়ার সময় নিতে হয়েছিল। এই প্রথম মুদ্রণ্ড তাদের জন্য যারা ডিবেজ-৩+ যা এর নিজস্ব ভার্সন সেশন যা এই ইউজার ইন্টারফেসে কাজ করে এসেছেন এবং এখন ডিবেজ-৪ এর দুইভাগ কামাইলার পরিবেশে কাজ করতে চান। অর্থাৎ প্রয়োজনীয় বিওরী এবং ডেফিনিশন ডকুমেন্টের অভাবে এতে পারছেন না। সম্পূর্ণ আলোচনার আমি আপনাদের ডিবেজ-৩+ হতে ডিবেজ-৪-এ যাওয়ার পক্ষে কিছুটা সৃণম করার চেষ্টা করব। পরে প্রয়োজনীয় যোগাযোগ দেব যাতে খুব সহজেই আপনারা ডিবেজ-৪ এর নতুন পরিবেশকে ঘরোয়া করে নিতে পারেন। পাশাপাশি ডিবেজ-৩+ এর পুরাতন এন্ট্রিকেশনসমূহ, যা আপনার কাজে তরলন, তা ডিবেজ-৪-এ স্থানান্তরের বিষয় এতে আলোচিত হবে। বিবারণিতভাবে নিচের বিষয়গুলোর উপর আলোকপাত করার চেষ্টা করব।

- ১। Indexing-এর ওজনস্বূর্ণ সম্প্রসারণ (Enhancement)
- ২। মাস্টি ফাইল রিপ্লেসন তৈরী
- ৩। নানাবিধ নতুন এবং end-user কমাণ্ড
- ৪। UDF (User Defined Function) তৈরীর পদ্ধতি।
- ৫। ইউজার ইন্টারফেস ইন্সট্রাকশন
- ৬। WYSIWYG—এনভায়রনমেন্ট তৈরীর জন্য উইন্ডোজিং, Pop-up, pull-down মেনু এবং লিষ্ট তৈরী।
- ৭। ডিবেজ-৩+ হতে ডিবেজ-৪-এ ফাইল কনভার্স
- ৮। কম্পাইলিং বোর্ড সম্পর্কিত ধারণা
- ৯। BUILD এবং DBLINK ইউটিলিটির পরিচয়
- ১০। টু-ডিভিভেনশনাল এর (Array)
- ১১। অটোমেটিক মাস্টি ইউজার ফীচার

এছাড়াও "Transaction Processing" অর্থাৎ ডিবেজ-৪ এর Random এবং Buffered ফাইল এসেসিং এবং Error হ্যান্ডেলিং বিষয়ে কিছু তাত্ত্বিক আলোচনা করা হবে। এখানে উল্লেখ যে, সকল বিষয়ের আলোচনাই মূলতঃ প্রোগ্রামারদের জন্য কন্ট্রি লিষ্ট ভাবে করা হবে। তবে সাধারণ ব্যবহারকারীদের এ আলোচনা হতে প্রয়োজনীয় জ্ঞান লাভ করতে পারেন।

হেতেই ডিবেজ-৪ এর সম্পূর্ণ বিষয় বিদ্যান নিয়ে যথেষ্ট আলোচনা এখানে সম্ভব নয়, তাই এতদাল প্রোগ্রামারদের জন্য কিছু ফীচার নাম নিতে হয়েছে। এদের মধ্যে অন্যতম TEMP-ন্যাভেগেজ এবং Dbase-SQL language. এ জন্য আমি আর্থরিকভাবে সূচনিত। তবে অন্যত্রকমে এই মুদ্রণটি বিবরণী বৃহৎ কং সম্বন্ধক ট্রান্সজিশন ইস্যুতে প্রকাশ ফালাবে বলে আমার ধারণা।

ডিবেজ INDEIGN সিস্টেম :

পুরাতন ডিবেজ-৩+ ডাটাবেজ ফাইলের (.DBF) এন্ট্রিইং এবং রেকর্ড-এর জন্য একটি সাধারণ ইনডেক্সিং স্কীম (Scheme) নিয়ে ইনকরপোরিট করাগে। এক বলা হতো OTS (One Tree System)। পাশাপাশি ইনডেক্স হব এমন

একটি সাইডওয়ে ফাইল (ডিবেজ যার এন্ট্রিটেনশন .NDX) যা ডাটাবেজ-এ রেকর্ডগুলোর হেডেওগুলির জন্য একটি পৃথক (Individual) এন্ট্রি প্যারামিটার ধারণ করে। এই এন্ট্রি প্যারামিটারগে যথাক্রমে একটি INDEX KEY EXPRESSION ও একটি ডিভিজিওনাল রেকর্ড নামের নিয়ে গঠিত হয়। রেকর্ডনামচারটি এন্ট্রিটির সাথে জড়িত ডাটাবেজ ফাইলের রেকর্ডের উপাদান (index element) নির্দেশ করে। INDEX KEY এবং ডিভিজিওনাল RECORD POINTER -এর সমন্বয়ে গঠিত সমস্ত এন্ট্রিগুলো এসেসিং আপলকোয়িক্যাল অর্ডারে সাজানো থাকে। এর ফলে ইনডেক্সিং ফাইল আপনাকে ডাটাবেজের এলোমেলো (Unsorted) ডিভিজিওনাল রেকর্ডগুলোর (যারা DBF ফাইলে অবস্থান করে) সাজানো (Sorted) অবস্থায় VIEW পেতে সাহায্য করে।

ইনডেক্স ফাইল সর্বেক্ষণের ফলে ব্যবহারকারী যা প্রোগ্রামার ডিভিজিওনাল ডাটাবেজের সার্টিং এর সমস্ত ওজাহেতে কামোদ্য হতে মুক্ত থাকতে পারেন। যেমন ধরন— ইনডেক্স-এর অনুপস্থিতিতে আপনাকে কোন ডাটাবেজ-এর ডাটা ইনপুট যা APPEND করতে হলে সমস্ত RAW ডাটা কম্পিউটারে প্রবেশ করানোর পূর্বে নিজে সার্টিংয়ে নিতে হত। অর্থাৎ ডাটাবেজের জন্য কম্পিউটারের ব্যবহার করার মূল সুবিধাই এখানে শ্রেণ পেত। অপর দিকে ইনডেক্স ফাইল তৈরী করতে একটি কমাণ্ডই যথেষ্ট সার্টিং অর্ডার এবং সর্টের জন্য এন্ড্রাপ্রেশন স্কীম CALL IDENTIFIER হিসাবে সরবরাহ করা যায় এবং যে কোন মুহুর্তে তা পাল্টানো যায়। একবার INDEX ওপেন করে নিলে ব্যক্তি যে সুবিধা পাতগে যায় তাহলে ইনডেক্সটি (যাকে INDEX TABLE ও বলা হয়)। যতক্ষণ পর্যন্ত একটিই থাকবে ততক্ষণের ভিতর যত নতুন রেকর্ড ডাটাবেজ-এ যোগ করা হবে বা ডাটাবেজ হতে ডিলিট করা হবে তা অটোমেটিকভাবেই SORTED ডাটাবেজের অংশ হয়ে যাবে। এখানে ব্যক্তি কিছু করতে হবে না।

ডিবেজ-৩+ এর অধীনে প্রত্যেক ইনডেক্স-এর জন্য একটি ফাইল প্রয়োজন পড়ে। অর্থাৎ কোন ডাটাবেজকে INDEX করতে চাইলে যে INDEX TABLE টি ডাটাবেজের SORTED রেকর্ডগুলোর একটি মেমরী প্রতিরূপিত ধারণ করবে তা INDEX CLOSE করার সময় ডিবেজ সর্বেক্ষিত হবে একটি ফাইল হিসাবে এবং তাই ফাইলটি DOS-এ File handle ব্যবহার করবে। এভাবে বিভিন্ন ফিল্ডের জন্য একাধিক INDEX TABLE ব্যবহৃত হলে প্রত্যেকটি INDEX TABLE পরস্পরকে আলাদা ফাইল তৈরী করবে এবং তাদের এন্ট্রিটেনশন হবে .NDX। তবে একত্রক সময়েও বড় অনুবিধা হচ্ছে ডিবেজ-৩+এ একটি DBF ফাইলের জন্য সর্বোচ্চ ১০টি INDEX ওপেন করা যায়। এর অর্থ হল ১ একটি ডাটাবেজের জন্য মাত্র ১০টি KEY EXPRESSION-এর ব্যবহার আপনার প্রয়োজন যা কম্পিউটার পঠিক বৈশ্বব্যহারীভাবে প্রের্য কম কাটা সীমানার মধ্যে দেবে।

এই সময়সূচী সমাধানের জন্য আপনি ডিবেক্স-৪ এর MULTIPLE INDEX FILING (MDX) সিস্টেমে প্রবেশ করতে পারেন। ডিবেক্স-৪ এর MDX-ফাইলিং সিস্টেম ৪গুটি পর্যন্ত আলাদা ইনডেক্স ট্রেবল তৈরী করতে দেয় এবং INDEX কে CLOSE করার সময় (set INDEX to কমা) দিয়ে অথবা CLOSE ALL কমাও দিয়ে CLOSE হতে হবে তার আসার সময় INDEX সফট্বের CLOSE হয়ে যাবে। সবচেয়ে টেকনিক একসাথে মাত্র ১টি ফাইলে সংরক্ষণ করে। এর সাহায্যে আপনি UNIQUE ইনডেক্স সম্বন্ধে মিস্র কিংবা মাচ করতে পারেন; কম্পাউন্ড ইনডেক্স এক্সপ্রেসন এবং কনভেনশন্যাস সিস্টেম সিস্টেম ইনডেক্স কী একই ফাইলের আওতাধর রেখে ব্যবহার করতে পারেন। ডিবেক্স-৪ যদিও একটি DBF ফাইলের জন্য মাত্র একটি MDX ফাইল প্রদান করতে সক্ষম তাহাপি এই একটি MDX ফাইল ৪গুটি পর্যন্ত ইনডেক্স-কী ধারণ করতে পারে বা ডিবেক্স-৩+ এর ৪গুটি MDX ফাইলের সমতুল্য। ৪গুটি ইনডেক্সও যদি আপনার জন্য যথেষ্ট না হয় তাহলে আপনি ডিবেক্স-৩+ ব্যবহার করে অতিরিক্ত ১০টি SKI (Single Key Index) ফাইল একই সময়ে মুক্ত করতে পারেন। (অর্থাৎ ডিবেক্স-৪ এর আওতাধর আপনি মাত্র ৪গুটি MDX+১০টি INDEX) = মোট ৫৭টি ইনডেক্স সুবিধা।

ডিবেক্স-৩+ এর সাথে কমপ্যুটারবিধিষ্টি রক্ষার জন্য ডিবেক্স-৪ পুরোনো MDX ফাইল পৃথকই রাখেনা, NDX এক্সপ্রেসন ইনডেক্স ফাইল তৈরীও করতে পারে। ইন্ডেক্সিং-এর মূল প্রক্রিয়া অর্থাৎ SORTING-এর পার্থক্য নির্ধারণ করা যায়, ডিবেক্স-৩+ কেবল Ascending অর্ডারে INDEX সজাত। কিন্তু ডিবেক্স-৪-এ রেকর্ডকে Ascending বা Descending দুই অর্ডারেই সর্ট করা যায়।

MDX ফাইলের কার্যপদ্ধতি:
একটি ডাটাবেজের জন্য বহন প্রথমবারের মত একটি MDX ফাইল তৈরী করা হয় তখন যদি MDX ইনডেক্সটির জন্য কোন ফাইল নাম সরবরাহ করা না হয় তবে ডিবেক্স MDX ফাইলটিকে ডাটাবেজটির নামেই তৈরী করে শুধু এক্সপ্রেসন দেয় MDX। উদাহরণ হিসাবে মনে করুন আপনার ডাটাবেজটির নাম CUSTCODE.DBF। তাহলে মাথি ইনডেক্স ফাইলটির নাম হবে CUSTCODE.MDX। এই MDX ফাইলটি একটি ইউনিক ফাইলের নাম ধারণ করবে। একে বলা হয় Production অথবা Undefined ইনডেক্স। ডিবেক্স-৪ এই ইনডেক্সটিকে অটোম্যাটিক রপন হবে এবং DBF ফাইলটির পরিবর্তনের সাথে সাথে রিফ্রেশমেন্টে updated হয়ে যাবে।

ডিবেক্স-৪-এর ট্যাক ফাইলের প্রোগ্রামটি ডিভাইসে এ কাজ করে তা জানতে হলে আমাদের MDX ফাইলের বাইট সেভেলে পদার্থ করতে হবে। DBF ফাইলের ২৮নং বাইটটি ডাটাবেজটির INDEX STATUS-একে Flag বাইটও বলা হয়। এই status বাইটটি যদি ০১ (হেক্স)-এ সেট করা থাকে তবে বুঝতে হবে ফাইলটির জন্য একটি Production Index রয়েছে এবং ডিবেক্স-৪-এ মাঠ স্ক্রীনি তখন এ ফাইলটি বুঝে নেয়। অপর দিকে বাইটটি যদি

হেক্স ০০ তে সেট করা থাকে তবে বুঝা যাবে DBF ফাইলটির কোন Production Index নেই।

পূর্বেই বলা হয়েছে যে, ডিবেক্স-৩+এ প্রত্যেকটি ইনডেক্স কী এর জন্য আলাদা MDX ফাইল তৈরী হবে। ডিবেক্স-৪-এ ৪গুটি আলাদা কী নিয়েও একটি MDX ফাইল তৈরী হবে অর্থাৎ এ ক্ষেত্রে ৪গুটি আলাদা ফাইল তৈরী হওয়ার কথা ছিল। এমন এই ইনডেক্সটির ৪গুটি ফাইল লোকের কল্প একটু জটিল মনে হতে পারে। প্রকৃতপক্ষে মূল সমাধান বেশ সহজ। MDX ফাইলে অর্ধগুটি প্রত্যেকটি INDEX TABLE-কে নির্দিষ্টভাবে চিহ্নিত করার জন্য আলাদা আলাদা Tag থাকে। এই Tagগুলো নির্দিষ্ট INDEX এর জন্য ACCESS সরবরাহ করে। Tagগুলো প্রোগ্রামারর Tag কমাও দিয়ে ডিফাইন করে দেন। নিচের উদাহরণটি দেখা করুন।

```
USE CUSTCODE
INDEX ON CUST_ID TAG CUST_ID
INDEX ON house+ street + city tag
address
```

INDEX ON state + Zip.TAG location
এই উদাহরণটিতে CUSTCODE.DBF ফাইলটির জন্য বইসিস্টেমের CUSTCODE.MDX নামে একটি মাঠি ইনডেক্স তৈরী হবে। CUSTCODE.MDX ফাইলে ডিফাইন ইনডেক্স থাকবে এবং প্রত্যেকটি INDEX-এর জন্য আলাদা ACCESS-TAG (Key Expression) থাকবে। Tagগুলো হল CUST-ID, address এবং location। এই Tagগুলো ব্যবহার করে ব্যবহারকারী অথবা তার প্রিন্টপেশন প্রোগ্রাম ডিবেক্স-৪কে বলে দিবে কখন কোন ইনডেক্স ব্যবহার করতে হবে। যে INDEXটি বর্তমানে (currently) প্রদেয় রয়েছে থাকে বলা হয় PRIMARY KEY এবং বাকি আলাদা INDEX-কে বলা হয় SECONDARY KEYS। ডিবেক্স-৪ ব্যবহারকারী কোন মাঠিফাইল কী নিশ্চিত করে না। ব্যবহারকারীকে অথবা প্রিন্টপেশনকে তা চিহ্ন করে দিতে হয়।

MDX ও অ্যানালা ইনডেক্স সিস্টেম নিয়ে কিছু কথা:

বর্তমানে ডাটা ম্যানিপুলেশন ও ফিডিং জগতে B-tree (Binary-Tree) সিস্টেমে প্রোগ্রামিং ব্যবহারের বিদিক পড়িয়েছে। ১৯৭৫ সালে Wilrh: N-এর "Algorithms + Data structure = Programs" পুস্তকের উপর ভিত্তি করে যে INDEX STRUCTURE তৈরী করা হয় তা ছিল বেশ কম্প্লেক্স এবং একে MULTIPLE লে আউট ব্যবহার করা হয় বিয়ার একটি INDEX ARRAYতে ক্যানেরী সার্চিং এলিমে পুরো ফাইল স্ক্রু সর্ট করা যায়। Rick Spence, Herbert schild, EDWARD MENDEL SON-এর মত বিদ্যাত প্রোগ্রামাররাও এই প্রোগ্রামিংয়ের পক্ষে সারা গেল। এলিক হতে Ashton-tate তার ডিবেক্স ম্যানুয়ালের কোন সিরিজই B-tree System ব্যবহার করেনি। ডিবেক্স-৩+ পর্যন্ত ব্যবহৃত হয়েছে সিস্টেম কী ইনডেক্স সিস্টেম এবং এবার ডিবেক্স-৪ এর ব্যবহৃত হচ্ছে মাল্টিপল ইনডেক্সিং; স্বভাবতই বিঘ্নটি কিছুটা পাপালামে মনে হতে পারে। কিন্তু প্রকৃতপক্ষে MDX ফাইলের ব্যবহারের কলে 1/0^৩ সীড স্ক্রু কন বড়ক

অর্ধেক ডাটা নিয়ে বিশ্লেষণাল বেগিমে কাজ করার সুবিধা অনেকগুলি ফিডিং পেয়েছে বলা যায়। এটি এমন একটি উচ্চ সফটওয়্যার নির্দেশনা থাকে ভিত্তি করে ডিফাইন্ড ডাটাবেজ এবং প্রক্রেট ম্যানুয়ালগুলো সীমিত ডাটাবেজ প্রেসিগ্নি সমস্ত তাদের অসুবিধা কাটিয়ে উঠতে পারবে। পরিষ্করে ডিবেক্স-৪-এর MDX ফাইল নিয়ে আমি ব্যক্তিগতভাবে কিছু অভিজ্ঞতা করতে চাই। যা হল MDX ফাইলের ব্যবহার খুব সতর্কতার সাথে ম্যানুয়াল চাইতে। কেননা ডিবেক্স-৪ এর অনেক কমপ্লেক্স এক্সপ্রেসন আছে। তাই, তবে পরিষ্কার করে এটি বুঝা যায়। MDX ফাইলের ব্যবহার আপনার কেবলকে স্ট্রীমলাইন করে নিতে সাহায্য করবে এবং ডিবেক্স-৩+ অপেক্ষা একটি উন্নততর ও সুবিধাজনক ইনডেক্স কীম উপকার দেবে।

মাঠি ফাইল বিশ্লেষণ:
বিঘ্নটি বুঝার আগে "ফাইল বিশ্লেষণ" সম্পর্কে কিছুটা ধারণা পণ্ডায় থাক। ব্যবহার জীবনে এমন হতে পারে যে, আপনাকে দুটো ডাটা সেটের (ডাটাবেজ) উপর একই সময়ে কাজ করতে হবে। ডাটাবেজ দুটি যদি একটি অপারেটর সাহায্য সম্পর্কিত হয় তবে তাদের মাঝে তত্ত্ব আলাদা প্রদান করা যেতে পারে। আবার জটিলতা, স্পীড অথবা রাইডেরী রাইর্ ডাটাবেজগুলোকে পৃথকভাবেও ব্যবহার করা যাবে।

প্রোগ্রামারর মনে দুটো ডাটাবেজ-এ একটি বা ততোধিক কলাম যুক্ত নিয়ে কাজ করতে চান তখন তারা ডাটাবেজ দুটোকে একসাথে লিঙ্ক করে নেন। একটি উদাহরণ লক্ষ্য করুন। PROJECT.DBF ও SUPPLIER.DBF নামে আপনার দুটো ডাটাবেজ রয়েছে। PROJECT.DBF ফাইলের কলামের মধ্যে হচ্ছে S_NO, P_NO, J_NO এবং QTY এবং SUPPLIER.DBF ফাইলের কিছু ফিল্ড হচ্ছে S_NO, SNAME, STATUS এবং CUSTC। এরপর নামকরণ আবার PROJECT ফাইলের সমস্ত রেকর্ডকে সাগ্রাইয়ারদের (SNAME)সহ LIST করতে চাই। PROJECT ফাইলের ফিল্ডগুলো সার্চ করে আমরা সাগ্রাইয়ারদের নামে কোন রেকর্ড পাইনা। কিন্তু আমরা এটি সহজেই নিজে করতে পারি SUPPLIER ডাটাবেজ-এর কার্ডগুলো সার্চ করে। কেননা সেখানে সাগ্রাইয়ারদের নাম নির্দেশক SNAME ফিল্ডটি রয়েছে। ডাটাবেজ প্রোগ্রামিং-এ যদি "ফাইল বিশ্লেষণ" পদ্ধতি নং থাকত তবে আপনার এ কাজের জন্য আপনি সরাসরি LIST ব্যবহার করতে পারতেন না। বিচার আপনার যা করতে হত তা হ'ল, একটি স্ট্রেট প্রোগ্রাম রটিন দিয়ে ডাটা প্রক্রেট PROJECT ফাইলের সমস্ত রেকর্ড লিষ্ট করতে এবং আরেকটি রটিন দিয়ে SUPPLIER ফাইল USE করে তা হতে SNAME ফিল্ডটি LIST করে নিতেন। ডিবেক্স-এ SET RELATION TO কমাওটি দিয়ে এক্সপ্রেসন থাকে হতে আপনি মুক্ত হতে পারেন। পাশাপাশি সবগুলো ফিল্ডই (SUPPLIER ডাটাবেজের ফিল্ডসহ) এমনভাবে উপস্থাপন করতে পারেন যাতে ডাটাবেজ দুটির মাঝে কোন পার্থক্য সূচিত না হয়। নিচের উদাহরণের সাহায্যে বিঘ্নটি যাচা করা যাক।

```
SELECT O
USE Supplier
INDEX ON S_No to s_no
SELECT O
USE project
SET RELATION TO s_no INTO
:supplier
```

LIST s_no, supplier → sname : p_no.
_l_no. qly

এখানে SET RELATION কমান্ডটি করেছি ডাটাবেজের (PROJECT-সম্পর্কে USE করা হয়েছে বিক্রয় এটি কাস্টমি) সাথে INTO - রূপ ডাটাবেজ (SUPPLIER) এর সাথে সম্পর্ক (relationship) ডিফাইন করে। ডিবেজ-এ যে ডাটাবেজ হতে RELATION সেট করা হয় তাকে বলে Parent এবং যার সাথে করা হয় তাকে বলে Child. উল্লেখিত উদাহরণ PROJECT হচ্ছে প্রজেক্ট এবং SUPPLIER হচ্ছে চাইল্ড। পুর্বেই বলেছি যে, দুটি ডাটাবেজের সাথে রিলেশন একটি কমান্ড কিস্তের উপর নির্ভর করে করতে হয়। এই কমান্ড কিস্তিকে বলে Master Key, এই ক্ষেত্রে s_no হচ্ছে মাস্টার-কী। লক্ষ্য করলে দেখবেন s_no দুটি ডাটা বেজেই রয়েছে। এখানে একটা বিষয় উল্লেখ করা প্রয়োজন-দুটো ডাটাবেজের জন্য যে দুটি কমান্ড কিস্তি Master Key হিসাবে ব্যবহৃত হবে তাদের কিস্তি-টাইপ (ফ্যাক্টোর, নিউমেরিক, লজিক্যাল- ইত্যাদি) এবং কিস্তি-উইডথ (Width) একই হতে হবে।

রিলেশনশীপ স্থাপন করার সময় চাইল্ড ডাটাবেজটিকে অবশ্যই INDEX করে দিতে হবে। অবশ্য এখানে শুধু s_no এর জন্য INDEX তৈরি করতেই। কেননা s_no-ই হচ্ছে মাস্টার-কী কিস্তি। তবে এটা প্রয়োজন হবেন, যদি ডাটাবেজটি পুর্বেই ইনডেক্স করা থাকে এবং ঐ ইনডেক্সটি Up to Date হয়।

ডিবেজ-এর পুর্বকর্তী জার্নালসেভারে ব্যবহারকারীর এই ধাপ Relation এজারেস বা কন্ট্রোলের ক্ষেত্রে সীমাবদ্ধতা ছিল। এর অর্থ তখন আপনি একটি

প্রজেক্ট ফাইলের জন্য মাত্র একটি রিসোর্সেট ডাটাবেজ তৈরী করতে পারতেন। ডিবেজ-৪-এ আপনি বহু সংখ্যক ফাইল রিলেশন তৈরী করতে পারেন, যেখানে PARENT ডাটাবেজটির সাথে একাধিক CHILD ডাটাবেজ সংযুক্ত থাকবে। একটি উদাহরণ দেখুন-
SELECT 1 && open work area
USE Customers &&
.open parent

USE orders order_id IN 2 &&
open child 1 in work area 2
USE Contacts order con_id IN 3
..... && open child 1 in work area 3
USE calls order.call_id in 4
&& open child 2 in work area 4
SET RELATION OT order_id INTO
orders
con_id INTO contacts, call_id INTO
calls&& link all childs to parent
GO TOP && move record
pointer

উদাহরণটির মাধ্যমে আমরা বলতে পারি যে, CUSTOMER ডাটাবেজ তিনটি অংশের ডাটাবেজের সাথে লিঙ্ক হয়েছে। একটি কাটিমারসের কন্ট্রোল সেকেন্ড (ORDER), কাটিমারসের সাথে যোগাযোগ কর সেকেন্ড তথ্যের জন্য (CONTACTS) বিস্তারিত এবং একটি মেইল কল লগ, যা কাটিমারসের সাথে যোগাযোগ করার প্রেক্ষাপটে (CALLS)। Master Key কিস্তি হিসাবে এখানে CUST_id কে ধরা হয়েছে। Cust_id হচ্ছে কাটিমারসের ID নাম্বার (যা তিনটি ডাটাবেজেই আছে বলে কল্পনা করা হয়েছে)।

ডিবেজের পুরোনো জার্নালের আরেকটি বিপজবলক অনুবিধার কথা এখানে কথা প্রয়োজন। ডিবেজ-৩-এ আপনি যদি দুইটি ডাটাবেজকে দুইটি ভিন্ন WORK AREA (Alias)তে রাখেন কাসেন এবং দুইটি ডাটাবেজের জন্য দুইটি .NDX ফাইল ব্যবহার করেন এবং যদি দুটো INDEX-এর জন্য একই নাম ব্যবহার করেন তবে প্রকৃতপক্ষে একটি NDEX তৈরী হবে। অর্থাৎ একটি work area অন্য work area দ্বারা গভার রাইট হয়ে যাবে। ডিবেজ-৪-এ এই সমস্যা দেখা দেবে না। কেননা Production INDEX-এ ভিন্ন ভিন্ন Tag ব্যবহার করে ভিন্ন ভিন্ন index table তৈরী করা যাবে। আবার প্রত্যেক work area-তে একই Tag নামসমূহ বার বার ব্যবহার করা যাবে।

ডাটাবেজ প্রোগ্রামিং-এ ডিবেজ-৩-নং, ডিবেজ-৪ এবং অন্যান্য অনেক ডাটাবেজ ম্যানুয়ালেই এতদপ্রকারী সীমাবদ্ধতার মধ্যে ডাটাবেজ রিলেশনের সীমারাইটি সবচেয়ে কম বোধগম্য এবং অনেকেরই বিষয়টি যত্নসহকারে এড়িয়ে যান। অথচ ডাটা কোয়ারীর জন্য যতগুলো পদ্ধতি রয়েছে তাদের মধ্যে এই পদ্ধতিই সর্বশ্রেষ্ঠ পদ্ধতি। এ বিষয়ে পঠকর্মের ছোট একটি উপদেশ দিতে চাই। POWER PROGRAMMING-এর বার্থে সবারই উজ্জিত ইনডেক্স ও রিলেশনশীপ ডাটাবেজ স্যারগেট-এর পদ্ধতি আরম্ভ করা। প্রোগ্রামে এই দুটি পদ্ধতির ব্যবহারই আপনার প্রোগ্রামের কমপ্লিটনেস এবং এফেসিভেনেসি আনবে। শুরুতে হয়ত একই অনুবিধা হবে। তবে একবার পরলে টিকই সক্ষম হবেন।

WE REPAIR / UPGRADE HARD-DISKS 386/286/XTs/ATs FLOPPY DRIVES, MONITORS, KEYBOARDS, PRINTERS, SMPS, UPS & FAX



DETOSEARCH

House # 6, Road-27 (Old), Dhanmondi, Dhaka-1209, Phone-817214.
Mipur-10-B, Avenue-1, Plot-3, Dhaka-1221, Phone-802458

DETOSEARCH Provides full component level repair facilities for systems and peripherals. Original spares and components are available for all leading brands. Our expert engineers diagnose and repair quickly, efficiently and inexpensively.

Offering Exclusive Facility for Annual Maintenance Contract, LAN Installation/Support & Consultancy.

THE MAINTENANCE PEOPLE