Database Architecture และ

Data Structure สำหรับ Natural Billing

Natural Billing สำหรับ Windows 3.x พัฒนาตั้งแต่ ปี 2536 เพื่อแก้ปัญหา การโต้เถียง อันเนื่องจาก Error 30 – 50% กรณีชุมสาย ...ไม่ส่ง Line Revert หรือ Single ยืนยัน การใช้โทรศัพท์ เพื่อลด Error ให้ เหลือ Error 5 – 10% หรือ 3 – 5% กรณี Optimized ในแต่ละ Site จึงต้องพัฒนา ด้วย AI หรือ Expert Systems

Natural ต้องวาง Data Structure ใหม่ เดิมทำงานบน Dos ถูกจำกัด RAM เพียง 640 KB ประมวลผล ทีละ Extension เมื่อทำงาน บน Windows มี RAM 4 MB จึงอยากรวมข้อมูล ทุก Extension เป็นก้อนเดียวกัน แต่ Relation DataBase ขณะนั้น ไม่รองรับข้อมูลมหาศาลบน Windows จึงพัฒนา DataBase Engine ต่อไป ยุคนั้น ปี 2536 ใน DataBase ทั่วไป ยังแบ่งข้อมูลเป็น Section ไม่ได้ แต่ Natural ต้องแบ่งข้อมูล เป็น รายเดือน เพื่อรองรับข้อมูล ...มหาศาล กับ Hard Disk ขนาด 20 GB (ปกติ Database ทั่วไป ต้อง Load Record ทั้งหมด ขึ้น RAM ถึงจะประมวลผลได้ Natural จึงพัฒนา Database Engine ในลักษณะ คล้าย ๆ Virtual Memory)

Natural พัฒนา Object-Oriented DataBase แบ่งข้อมูล แต่ละเดือน เป็น Section แล้ว ยังอ้างถึง ในแต่ละ Records ขนาด 512 Byte ในลักษณะเดียวกับ Virtual Memory ด้วย Primary Object เพื่อจัดเก็บ ข้อมูลต่าง ๆ ลงแต่ละ Sector ย่อย ๆ ขนาด 2KB ของ Virtual Memory และ File Management อย่างลงตัว โดย Primary Object มีขนาด 32 Byte ในรุ่น 1.0 – 4.0 และ ขนาด 64 Byte ในรุ่น 5.0 – 14.0 (ณ. ปัจจุบัน ตั้งแต่ 15.0 ขึ้นไป Primary Object จะมีขนาด 128 Byte เพื่อลดการ Load Records ในกรณีจำเป็นเท่านั้น)

Primary Object ของ Natural สำหรับ OODB : Object-Oriented DataBase รุ่น 1.0 สำหรับ Telephone Billing เสมือนเป็น Reference ไปยังแต่ละ Records อันประกอบด้วย Fields ต่าง ๆ ที่แสดงถึง เอกลักษณ์ ของแต่ละ Records สำหรับ OODB 2.0 รองรับ ERP / MRP ตั้งแต่ปี 2540 เพิ่ม Reference ID เพื่อความรวดเร็ว ในการประมวลผล และ OODB 3.0 รองรับ Social Enterprise ด้วย Object Chains

OODB 1.xx ของ Natural โดยการแบ่งข้อมูล เป็น รายเดือน และ ใช้เทคนิกอ้างถึง Record ต่าง ๆ ด้วย Primary Object เลียบแบบ Virtual Memory รวมถึง Native Query คล้าย ๆ Microsoft LINQ ณ. ปัจจุบัน ทำให้ OODB ของ Natural ทำงานเร็วกว่า SQL Script หรือ Relation Database ทั่วไป **มากกว่า 1,000 เท่า**

```
329     struct SortPHONE    // 128 Byte
330     {
331         nx::DateTime            Begin;
332         nx::DateTime            Ended;
333
334         nx::phone::Hexphone     Station;
335         nx::phone::Hexphone     Authorz;
336
337         nx::phone::Hexphone     Start;
338         nx::phone::Hexphone     Nexts;
339
340         nx::phone::Hexphone     Service;
341         nx::phone::Hexphone     TrunkDest;
342
343         nx::phone::DialNumber   DialPhone;
344
345         nx::phone::Hexphone     DIALACCOUNT;
346         nx::phone::Hexphone     DESTINATION;
347
348         nx::TimeSpan            Using;
349
350         unsigned int            FLAG;
351         int                     Phone;
352         int                     Error;
353         int                     Device;
354
355         double                  Amout;
```

Primary Object ประกอบด้วย Fields หลัก ๆ ในแต่ละ Records ของ Telephone Billing เป็น Reference หรือ เอกลักษณ์ อ้างถึงแต่ละ Records แต่เพื่อให้ Users ทั่วไป เข้าใจได้ง่าย ๆ จึงเรียกว่า Index เพราะ Native Query ตั้งแต่ปี 2536 มีลักษณะ LINQ ในปัจจุบัน แต่เน้นประมวลผล ด้วย **Binary Search**

ในแต่ละ Records ของ Telephone Billing จะมีขนาด 512 Byte

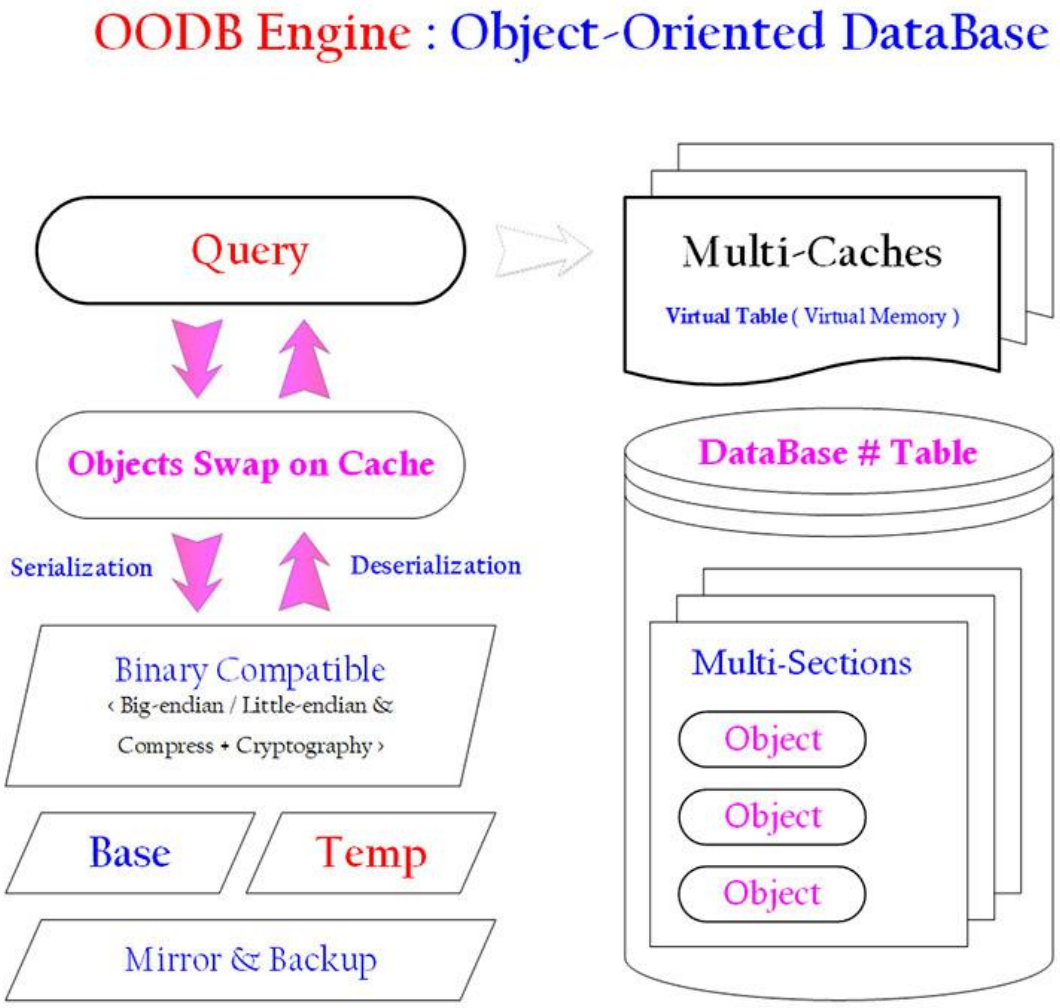
```
struct DataPHONE   // 512 Byte
{
    long long               START;
    long long               STYLE;

    nx::phone::DialNumber   DialService;        // DialService
    nx::phone::DialNumber   DialPhone;          // DialTerminal

    long long               FLAG;
    nx::phone::Hexphone     Audit;              // Sequence of PABX

    nx::phone::Hexphone     Operator;           // new **
    nx::phone::Hexphone     Authorz;            // Account : Charge Number
    nx::phone::Hexphone     Station;            // Caller  : Original Number
    nx::phone::Hexphone     Start;
    nx::phone::Hexphone     Nexts;
    nx::phone::Hexphone     TrunkDest;          // TrunkOut, TrunkIn
    nx::phone::Hexphone     TrunkCall;          // Network

    nx::phone::Hexphone     DIALACCOUNT;

    nx::phone::Hexphone     Service;            // DESTINATION
    nx::phone::Hexphone     DESTINATION;

    nx::phone::Hexphone     AuthorzCode;        // Authorize Code
    nx::phone::Hexphone     ServiceCode;        // Account Coce

    nx::DateTime            Begin;
    nx::DateTime            Ended;

    int                     Phone;
    int                     Error;

    int                     Security;
    int                     Device;             // delete

    long long               QueryNumber;        // delete
    long long               CallCategory;       // delete
    long long               PartyCategory;      // delete
    long long               OperatorCategory;   // delete
```

OODB 1.x สำหรับ Telephone Billing แต่ละ Records เป็น C++ Structure มีขนาดแน่นอน

ในขณะ OODB 2.x สำหรับ ERP / MRP เป็น Dynamic Object ขนาดของแต่ละ Record ไม่แน่นอน แต่จะจัดเก็บข้อมูล ...ลง Storage มีขนาดแน่นอน (โดยแต่ละ Table สามารถกำหนด ได้อิสระ เป็น 1 – 8 KB)

```
nx::DateTime            Create;             // Formula by Extension
nx::DateTime            Update;             // Formula by Trunk

nx::phone::Services     ADDRESS;            // Exchange ID
nx::phone::Services     NETWORK;            // Exchange Network

long long               SerialID;           // <= Speed & Security
long long               RecordID;           // <= Speed & Security

nx::phone::Hexphone     NewEDIT;
nx::phone::Hexphone     OldEDIT;

nx::DateTime            NewTIME;
nx::DateTime            OldTIME;

nx::TimeSpan            Ring;               // Duration
nx::TimeSpan            Tranf;              // Duration
nx::TimeSpan            Using;              // Duration
nx::TimeSpan            Second;             // Duration

nx::phone::Operator     FORMULA_EXT;

nx::phone::Operator     OPERAT_TRNK;
nx::phone::Location     LOCATE_TRNK;
nx::phone::Operator     FORMULA_TRN;

nx::phone::Operator     OPERAT_DIAL;
nx::phone::Location     LOCATE_DIAL;
nx::phone::Operator     FORMULA_MAP;

nx::phone::Operator     OPERAT_END;
nx::phone::Location     LOCATE_END;
nx::phone::Operator     FORMULA_END;

wchar_t                 COMMENT[ 32 ];

double                  Prev;
double                  Serv;
double                  Rate;
double                  Amout;
```

การออกแบบ Natural Billing เน้นประสพการณ์ จากการออกแบบ VLSI / Computer Architecture กลายมาเป็น OODB 1.x ในขณะ OODB 2.0 และ OODB 3.0 ได้รับ อิธิพล จาก Jini Network Technology แต่ทั้งหมดอยู่บนพื้นฐาน Object-Oriented Database Engine โครงสร้าง และ เป้าหมาย เดียวกัน



OODB 1.x ของ Natural พัฒนา Client / Service Protocol บน TCP/IP ของตนเอง ตั้งแต่ปี 2560 ในปี 2567 ได้พัฒนาให้ รองรับ ทั้ง Data Redundancy, Real-Time Redundancy & Load Balancing ตามลำดับ สำหรับการเป็น Backup / Mirror ระหว่างกัน หรือ Co-operative Computing ระหว่างกัน

โครงสร้าง Database File สำหรับ Telephone Billing

SortPHONE มีชนิดข้อมูลเป็น C/C++ Struct ทำหน้าที่เป็น Primary Object มีลักษณะ File เป็น

C:\Natural\Phone\SORT-YYYYMM.INDEX { YYYY คือ ปี และ MM คือ เดือน }

DataPHONE เป็น C/C++ Struct ทำหน้าที่เป็น Record ของ DataBase มีลักษณะ File เป็น

C:\Natural\Phone\DATA-YYYYMM.PHONE และ

C:\Natural\Backup\BACK-YYYYMM.PHONE { YYYY คือ ปี และ MM คือ เดือน }

กฎการตรวจสอบข้อมูล **ห้ามซ้ำกัน** สิ่งสำคัญ ในการรวมตัวเป็น Primary Object คือ Field Name ทั้ง Station คือ Extension ผู้ใช้โทรศัพท์, Begin / Ended คือ ช่วงเวลาการใช้โทรศัพท์ ตั้งแต่ เริ่มโทร จนถึง วางสาย DialPhone คือ Dial Number และ TrunkDest คือ สายนอกการโทรไปยังปลายทาง ทั้งหมดรวมกัน ห้ามซ้ำกัน ทุก Fields ใน SortPHONE ถือเป็น Fields สำคัญ เสมือนเป็น เอกลักษณ์ และ ทั้งหมดมีอยู่ใน DataPHONE

Station คือ Extension ผู้ใช้โทรศัพท์ และ Authorize อาจเป็น Virtual Extension ผู้ถูกคิดค่าบริการ หรือ ผู้อนุญาติ การใช้โทรศัพท์ รวมถึง Start / Nexts คือ Extension เกี่ยวกับ Call Forward / Call Transfer ทั้งหมดจะอ้าง หรือ เชื่อมโยงไปถึง DataFRONT ซึ่งเป็น C/C++ Struct เกี่ยบกับ Extension

TrunkDest / TrunkCall อ้าง DataFRONT เกี่ยวกับ Trunk หรือ Channel ใน Communication

DIALACCOUNT / DESTINATION อ้าง DataFRONT สำหรับ DialGroup ของ Dial Number

DataPHONE มีชนิดข้อมูลเป็น C/C++ Struct ทำหน้าที่เก็บข้อมูลของ Department, Extension, Trunk และ DialGroup (Department คือ Group ของ Extension, Trunk และ DialGroup)

```
struct   DataFRONT // NEW == 512 Byte OLD == 128 Byte
{
    long long               START;
    long long               STYLE;

    nx::phone::Hexphone     NUMBERS;
    nx::phone::Hexphone     STATION;

    nx::phone::Hexphone     AUTHORZ;

    int                     LEVEL;
    int                     ORGANIZ;
    int                     SECTION;
    int                     DIALNUM;

    int                     RATETEL[ 8 ];

    long long               CONDITN;
    long long               STATUSE;

    nx::DateTime            ORIGINAL;
    nx::DateTime            DETERMIN;

    nx::DateTime            BOOK_INP;
    nx::DateTime            CHCK_INP;
    nx::DateTime            CHCK_OUT;
    nx::DateTime            CHCK_INV;

    long long               PASSWORD;
    nx::DateTime            PASSTIME;
    long long               FUCTIONS;
    long long               SECURITY;

    nx::phone::Hexphone     NEW_EDIT;
    nx::phone::Hexphone     OLD_EDIT;

    nx::DateTime            NEW_TIME;
    nx::DateTime            OLD_TIME;

    unsigned int            EXCHANGE;
```

Station แทน ID No. หรือ Circuit ID ในบาง PABX และ Number แทน Extension Number

```
unsigned int            EXCHANGE;

int                     UP_WAKE;
nx::DateTime            UP_DATE;

int                     Amountfee;
int                     AmountDay;

nx::DateTime            AmountStart;
nx::DateTime            AmountUpdate;

double                  AmountTotal;

double                  AmountService[ 8 ];

nx::DateTime            PrepayDebit;
nx::DateTime            PrepayStart;
nx::DateTime            PrepayUpdate;

double                  PhoneService;
double                  PrepayCycle;
double                  PrepayTotal;

double                  PrepayService[ 8 ];

int                     FREE_FRONT[ MaxFreeFRONT ];

double                  Discount[ 8 ];

wchar_t                 COMMENTS[ MaxTCHAR ];

nx::phone::Hexphone     EXTENSION[ 32 ];

wchar_t                 NICKNAME [ MaxTCHAR ];
wchar_t                 EMAILNAME[ MaxTCHAR ];
wchar_t                 WORKGROUP[ MaxTCHAR ];

long long               WebPhone;

nx::DateTime            WebBegin;
nx::DateTime            WebEnded;

nx::phone::Hexadecimal  WebSecurity;
```

ทุก Front Interface ทั้ง Department, Extension, Trunk และ DialGroup ห้ามมี Station ซ้ำกัน

```
nx::phone::Hexadecimal  WebSecurity;

nx::DateTime            WebAccess;
nx::DateTime            WebLogout;

wchar_t                 FULLNAME[ 2 ][ MaxTCHAR ];

wchar_t                 USERNAME[ MaxTCHAR ];

inline int compare( DataFRONT &p_buffer )
{
        if ( STATION < p_buffer.STATION )
        {
            return -1;
        }

        else if ( STATION > p_buffer.STATION )
        {
            return  1;
        }

        return 0;
}

inline bool operator == ( DataFRONT &p_buffer )
{
        if ( STATION == p_buffer.STATION )
        {
            return true;
        }

        return false;
}

inline bool operator != ( DataFRONT &p_buffer )
{
        if ( STATION != p_buffer.STATION )
        {
            return true;
        }

        return false;
}
```

โครงสร้าง Database File สำหรับ Front Interface

การทำ Report หรือ แสดงข้อมูล ต่าง ๆ ของ Telephone Billing ต้องเริ่มต้นด้วย Front Interface นั้นคือ การอ้างอิงผ่าน Department, Extension, Trunk & DialGroup ซึ่งพัฒนาสืบทอดมาตั้งแต่ปี 2536 โดยจัดเก็บข้อมูล ใน DataFRONT มีชนิดข้อมูลเป็น C/C++ Struct ขนาด 512 Byte ตั้งแต่ปี 2540 รุ่น 5.0

Department ของ Extension, Trunk และ DialGroup จะถูกเก็บลง File

C:\Natural\Bin\EXGROUP.Natural และ C:\Natural\Bin\EXGROUP.Backup

Extension สำหรับ Real Extension และ Virtual Extension จะเก็บลง File

C:\Natural\Bin\EXEXTEN.Natural และ C:\Natural\Bin\EXEXTEN.Backup

Trunk สำหรับ Trunk Caller และ Trunk Destination เพื่อเป็น Channel Communication

C:\Natural\Bin\EXTRUNK.Natural และ C:\Natural\Bin\EXTRUNK.Backup

DialGroup สำหรับ Dial Number ในการจัดกลุ่มเบอร์ปลายทาง เช่น เบอร์ลูกค้า, เบอร์ตัวแทน ฯลฯ

C:\Natural\Bin\EXDIALG.Natural และ C:\Natural\Bin\EXDIALG.Backup