

XEAD Driver 販売管理システム

売上実績テーブル

XEAD Driver とは何か？

下山 吉洋

2013/01/22

第1章 XEAD Driver とは何か

XEAD Driver のサンプルシステムである、販売管理システムは 2008 年 5 月に発行した、「販売管理でシステムで学ぶモデリング講座」の実装環境です。

XEAD Driver は、(有) デービーコンセプト 渡辺幸三氏によって開発された無償の開発ツールです。

XEAD Driver は、2010 年 3 月にベータ版が発表され、2011 年 4 月 8 日に正式版がリリースされ、その後もバージョンアップが進められ、最新版は 2012.12.18 版 <XEAD Driver V1R1M16> です。



データベースも標準の Apache Derby のほか、PostgreSQL、MySQL にも対応しています。PostgreSQL では、データベースエンコードに UTF-8 を指定することでテーブル名、フィールド名に漢字を使用することも出来ます。

「販売管理でシステムで学ぶモデリング講座」には論理設計の部分までが記述され、物理レベルのデータモデル、コードは一切記載されていません。

XEAD Driver のサンプル仕様書である、販売管理システム、Sample.xeaf はこの本の完全な実装システムです。

XEAD Driver がリリースされたことにより、実装システムの物理構造、全コードが公開されました。

また、Java 言語で記述された XEAD Driver、編集ツールである XEAD Editor のソースコードも全て github で公開されました。

上流から下流、設計から実装までが一気通貫して完結した資料の提供は大変珍しく、生きた資料です。書籍、設計情報、仕様書、実装システム、そして全てのソースコードまで公開されたのは、世界で初めての偉業です。

渡辺幸三氏が、「業務別データベース設計のためのデータモデリング入門」を日本実業出版社から発行されたのが (2001/7/1) ですから、10 年以上も、一人で地道に開発を続けられてきたことに頭が下がります。

第2章 XEAD Modeler とは何か

システム全体を俯瞰できるツールや、設計情報が提供されています。
渡辺氏は、設計ツールである XEAD Modeler もフリー、オープンソースとして github で提供されています。

設計ツールとしての、XEAD Modeler で閲覧することができる、コンセプトウェア（設計情報、レファレンスモデル製品）も無償提供しています。

CONCEPTWARE／財務管理
CONCEPTWARE／科目履修管理
CONCEPTWARE／販売管理
CONCEPTWARE／生産管理
CONCEPTWARE／自治体

これまでは、実際に、これらの設計情報で実際に動くシステムが提供されていないので大変参考になる設計情報ですが、これを元に独自に実装するのは骨の折れる、時間の掛かる仕事です。私自身も実装に挑戦しましたが完成には至っていません。

上流工程だけでは、最終的に動かす実装環境が無ければ、食べられない「絵に書いた餅」でしかありません。

XEAD Driver の発表とともに、「CONCEPTWARE／販売管理」を設計情報とした、販売管理の実装システムとして「Sample.xeaf」が提供されました。

「絵に描いた餅」が食べられるようになったのです。

やっと動くシステムを手に入れることができたので、中身を分析してみよう。

サンプルシステムは、63 個のテーブルで構成されています。

三要素分析法でテーブルの派生、参照、親子の関係に正規化されています。

全てを説明することは、膨大すぎますので、私が注目した点を簡単に解説してみます。私的な見解ですので勘違いや誤りがありましたらご指摘いただけましたら幸いです。

サンプルシステムの販売管理で、全体の要となる、売上実績、仕入実績、受払実績の処理のうち売上実績テーブルについての解説です。



「販売管理でシステムで学ぶモデリング講座」P5 の仕入/売上/受払を押さえよう。

P6 の図3 販売管理システムの会計3要素と各取引の関係を参考にしたい。

販売管理業務では、見積、受注、出荷、発注、請求...などの伝票のインプット、アウトプットが発生する。

何らかのルールを持って効率的にインプット、アウトプットを行うためにはシステムの然るべき場所へハブが構成される必要がある。

椿正明氏のブログより、(D23 データハブによる通信)

http://mtsubaki.at.webry.info/201104/article_4.html

「DBはハブ機能をもつが、ハブ機能の実装をDB方式に限定する必要はない。

文法と用語の標準化を施した掲示板を作ればよい。」

業務システムのテーブルにも役割をもたせ、取りまとめ場所（掲示板）としてのテーブルとなり、またその結果を集計するサマリーの元ネタとなるテーブルとすれば、ハブテーブルとなりシステム全体の見通しが良くなる。

様々な顧客の要望による集計値を計算し易い、サマリーを導出し易い掲示板としてのテーブルを検討することになるでしょう。

はてなキーワード > ハブ空港より

『ハブとはハブ空港などと呼ばれる

航空路の基点となる空港の内、通常の空港よりも数多くの他空港への航空路が確保され、離着陸する航空機の機数や取り扱う旅客や貨物の量も非常に多い空港のこと。

当該空港から他の空港へと繋がる航空路の広がる様が、車輪のハブ（空港）とスポーク（航空路）のように見えることから、「ハブ空港」と呼ばれるようになった。』

テーブルは記録媒体でしかないので、実際に記録を行う処理が必要だが、処理に関連するテーブルを限定し無秩序な処理にならないようにする工夫が必要です。

そうすることで、影響の範囲が判り易い、保守し易いシステムになります。

ハブ空港を拠点に効率の良い運行計画を立てることで全体最適化を目指す。

世界中が、やたらに直行便で網の目のように飛行経路を持つことが出来ないように、業務システムも制約を持った処理を考慮することが必要です。

三要素分析法では、サブシステムという単位で、テーブルと機能を一括りにし他のサブシステムと分割する。

このサブシステムは、運行を含めた空港にあたると思って欲しい。

一般の空港と、ハブ空港のように

一般的なサブシステムとハブとなるサブシステムに分けられる。

具体的な、例を「売上実績」テーブルをもとに解説してみよう。

次の章で、XEAD Modeler で閲覧した設計情報と XEAD Editor で閲覧できる実行可能な仕様を比較してみます。

第3章 売上実績テーブル

XEAD Modeler、売上実績と関連サブシステム

テーブル名: 売上実績 ID(←主鍵): DT100 タイプ: 02 トランザクション

一次識別子: 売上No.

説明: 売上実績情報を保持するためのテーブル。売掛残高が増減するたびに追加される。いずれのフィールド値についても、ユーザがパネルから直接編集することはない。

CREATE文

```
CREATE TABLE DT100(
  NRURIAGE CHAR(7) NOT NULL COMMENT '売上No.',
  KBKADEN CHAR(2) NOT NULL COMMENT '売上品',
  FGVALID CHAR(1) NOT NULL COMMENT '区分',
  KBURIAGE CHAR(2) NOT NULL COMMENT '売上区分',
  DTURIAGE DATE NOT NULL COMMENT '売上日',
  NRTORIHAKISAKI CHAR(8) NOT NULL COMMENT '顧客No.',
  NRSYOHIN CHAR(7) COMMENT '売上商品No.',
  TXSYOHIN CHAR(30) COMMENT '商品名/摘要',
  SOLOT SMALLINT COMMENT 'ロット行',
  PRHANBAI INTEGER COMMENT '標準販売価格',
  PRURIAGE INTEGER COMMENT '売上単価',
  QTURIAGE INTEGER NOT NULL COMMENT '売上数量',
  KBURIAGEUM CHAR(5) NOT NULL COMMENT '売上単位区分',
  AMURIAGE INTEGER NOT NULL COMMENT '取扱取引番',
  AMTAX INTEGER COMMENT '税額',
  AMGENKA INTEGER COMMENT '売上原価',
  NRSEIKYU CHAR(7) COMMENT '請求No.',
  NRJUYUGOIN CHAR(6) COMMENT '担当営業No.',
  NRTORIHAKIKANRI CHAR(7) NOT NULL COMMENT '取引管理No.',
  DTCREATE TIMESTAMPT NOT NULL COMMENT '実績追加日時',
  DTNEND SMALLINT NOT NULL COMMENT '取引年度',
  DTMSEQ SMALLINT NOT NULL COMMENT '取引月序',
  AMNEBIKI INTEGER NOT NULL COMMENT '情報分類',
  AMARARI INTEGER NOT NULL COMMENT '取引職',
  CONSTRAINT DT100_PK PRIMARY KEY (NRURIAGE),
  CONSTRAINT DT100_FK1 FOREIGN KEY (NRSEIKYU) REFERENCES DT110 (NRSEIKYU),
  CONSTRAINT DT100_FK2 FOREIGN KEY (NRTORIHAKISAKI) REFERENCES DT010 (NRTORIHAKISAKI),
  CONSTRAINT DT100_FK3 FOREIGN KEY (DTNEND, DTMSEQ, NRTORIHAKISAKI) REFERENCES YT050 (DTNEND, DTMSEQ, NRTORIHAKISAKI),
  CONSTRAINT DT100_FK4 FOREIGN KEY (DTNEND, DTMSEQ, NRJUYUGOIN) REFERENCES YT060 (DTNEND, DTMSEQ, NRTANTOUJUYUGOIN),
  CONSTRAINT DT100_FK5 FOREIGN KEY (DTNEND, DTMSEQ, NRTORIHAKISAKI, NRSYOHIN) REFERENCES YT051 (DTNEND, DTMSEQ, NRTORIHAKISAKI, NRSYOHIN)
);
```

DT100 売上実績の説明に「売上実績情報を保持するためのテーブル。売掛残高が増減するたびに追加される。いずれのフィールド値についても、ユーザがパネルから直接編集することはない。」と明記されている。

XEAD Driver、売上実績と関連サブシステム

XEAD Editor 1.1 - [C:\Documents and Settings\User\Application Data\Xead\Sample\Sales\Orosi.xead] - 卸売業向け販売管理システム 1.0.23

ファイル(F) 編集(E) 検索(S) ツール(T) ヘルプ(H)

D 売掛残高データ管理 - DT100 売上実績

卸売業向け販売管理システム

メニュー一覧

アプリケーション

A 総勘定マスター管理

B 商品マスター管理

C 買掛残高データ管理

D 売掛残高データ管理

テーブル一覧

DT010 顧客

DT011 出荷先

DT020 振替バンク

DT100 売上実績

DT110 請求指示

DT120 受領明細

機能一覧

E 在庫管理データ管理

F 発注・入荷データ管理

G 受注・出荷データ管理

テーブル一覧

GT010 受注見出し

GT011 受注明細

GT012 出荷先明細

GT013 出荷売上明細

GT020 出荷見出し

GT030 見積見出し

GT031 見積明細

機能一覧

H 一般仕入取引データ管理

I 一般売上取引データ管理

テーブル一覧

JT010 売上指替

JT020 特殊売上

機能一覧

K 一般在庫取引データ管理

L 在庫指替データ管理

Y 統計データ管理

テーブル一覧

YT010 年次取引ISUM

YT020 月次取引ISUM

YT030 商品別月次取引ISUM

YT031 商品在庫別月次取引ISUM

YT040 仕入先別月次取引ISUM

YT041 仕入商品別月次取引ISUM

YT050 顧客別月次取引ISUM

YT051 顧客商品別月次取引ISUM

YT060 営業担当別月次取引ISUM

機能一覧

Z システム制御データ管理

テーブルID DT100 テーブル名 売上実績 モジュールID *ID モジュールチェック

一次識別子 売上No

範囲KEY項目 *None

削除操作 *Delete 有効なWhere *None

説明 売上実績情報を保持するためのテーブル。売掛残高が増減するたびにテーブルスクリプトによって追加される。いずれのフィールド値についても、ユーザがパネルから直接編集することはない。

フィールド KEY 結合テーブル スクリプト 使途要素 データ

NO.	ID	フィールド名	データタイプ	拡張データタイプ	説明
1	NRURIAGE	売上No	CHAR(7)	自動採番値(NRURIAGE)	
2	KBAKADEN	赤黒	CHAR(2)	区分値(KBAKADEN)	有効区分。赤黒処理分は無効化される。請求簿の処理
3	FVALID	区分	CHAR(1)	区分値(FVALID)	
4	KBURIAGE	売上区分	CHAR(2)	区分値(KBURIAGE)	
5	DTURIAGE	売上日	DATE		
6	NRTORIHIKISAKI	顧客No	CHAR(8)		
7	NRSYOHIN	売上商品No	CHAR(7)	NULL可	
8	TKSYOHIN	商品名/摘要	CHAR(30)	漢字, NULL可	振替取引では振替摘要が設定される
9	SLOT	ロット行番	SMALLINT(4)	NULL可	
10	PRHANBAI	標準販売価格	INTEGER(9)	NULL可	売上計上時に商品マスターからコピーされる。これと標準単価との差にもとづく金額が値引額とみなされる
11	PRURIAGE	売上単価	INTEGER(9)	NULL可	
12	QTURIAGE	売上数量	INTEGER(9)	NULL可	
13	KBURIAGEUM	売上単位区分	CHAR(5)	区分値(KBURIAGEUM)	
14	AMURIAGE	振替売上額	INTEGER(9)	NULL可	
15	AMTAK	振替	INTEGER(9)	NULL可	
16	AMGENKA	売上原価	INTEGER(9)	NULL可	
17	NRSEIKYU	請求No	CHAR(7)	NULL可	
18	NRJYUGYOIN	担当営業No	CHAR(5)	NULL可	
19	NRTORIHIKIKANRI	振替1管理No	CHAR(7)		

フィールドID NRURIAGE 物理フィールド データタイプ CHAR 桁数 7

フィールド名 売上No

拡張データタイプ なし 漢字 半角カナ 自動採番値 NRURIAGE 区分値

U R L 郵便番号 干渉動住所 電話番号 プール値

年月 年度 月序 画像ファイル リスト値

説明

NO.	使途要素	タイプ	利用形式
1	DT100 売上実績	TABLE	一次識別子
2	DT100 売上実績	TABLE	インデックス
3	DT100 売上実績	TABLE	スクリプト「変更不可設定」中の実数
4	DF011 顧客と関連情報の確認	XF300	明細テーブルの識別子
5	DF011 顧客と関連情報の確認	XF300	明細データの一覧順序
6	DF100 売上実績データの一覧	XF100	一次テーブルの識別子
7	DF100 売上実績データの一覧	XF100	明細フィールド
8	DF101 売上実績データの表示	XF200	一次テーブルの識別子
9	DF101 売上実績データの表示	XF200	明細フィールド

Z システム制御データ... ZT081 月次為替レ... ZT080 年次為替レ... ZT070 休日テーブル ZT060 消費税率 ZT051 セッション明細 ZT050 セッション ZT040 ユーザ定義区...

DT100 売上実績の説明に「売上実績情報を保持するためのテーブル。売掛残高が増減するたびにテーブルスクリプトによって追加される。いずれのフィールド値についても、ユーザがパネルから直接編集することはない。」

具体的な、処理として「売掛残高が増減するたびにテーブルスクリプト」を確認する。「DT100 売上実績」という掲示板に書き込んでくるとは誰か調べてみよう。右側ペインの使途要素タブをクリックすると掲示板を参照する機能がー覧される。

説明 売上実績情報を保持するためのテーブル。売掛残高が増減するたびにテーブル値についても、ユーザがパネルから直接編集することはない。

フィールド KEY 結合テーブル スクリプト 使途要素 データ

NO.	使途要素	タイプ	利用形式
1	DF011 顧客と関連情報の確認	XF300	明細テーブル
2	DF100 売上実績データの一覧	XF100	一次テーブル
3	DF101 売上実績データの表示	XF200	一次テーブル
4	DF111 請求エントリ	XF110	一次テーブル
5	DF113 請求指示と関連情報の確認	XF300	明細テーブル
6	DF115B 請求書の印刷	XF390	明細テーブル
7	DF116 売上実績の確認/請求指示取消	XF200	一次テーブル
8	DF122 受領明細と関連情報の確認	XF300	明細テーブル
9	DF130 売上実績の一覧	XF100	一次テーブル
10	GF044 出荷売上明細別取引実績の検討	XF300	明細テーブル
11	JF012 売上振替別売上実績の検討	XF300	明細テーブル
12	JF023 特殊売上別取引実績の検討	XF300	明細テーブル
13	YF051 顧客別取引内訳の検討	XF300	明細テーブル
14	YF052 顧客商品別取引内訳の検討	XF300	明細テーブル
15	YF061 営業担当別取引内訳の検討	XF300	明細テーブル

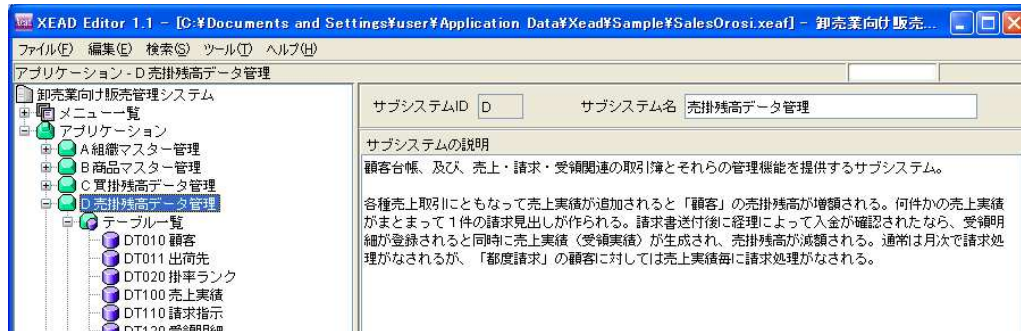
使途要素の一覧に DF011 のように、2 桁目に F と表示されているのは、機能です。テーブルスクリプトとは機能へ関連付けられたテーブルのスクリプトが働きます。

タイプ XF300、XF100、XF200 などのタイプは、スクリプトを直接記述することはありません。XF000 だけが、スクリプトを記述することが出来ますがこの一覧にはリストアップされていません。

1. 売掛残高が増減するたびに働くテーブルスクリプト

売掛残高が増減するたびにテーブルスクリプトの具体的な例
キーワードから、アタリをつけましょう。

XEAD Driver で、D 売掛残高データ管理を選択し、サブシステムの説明を見ると。

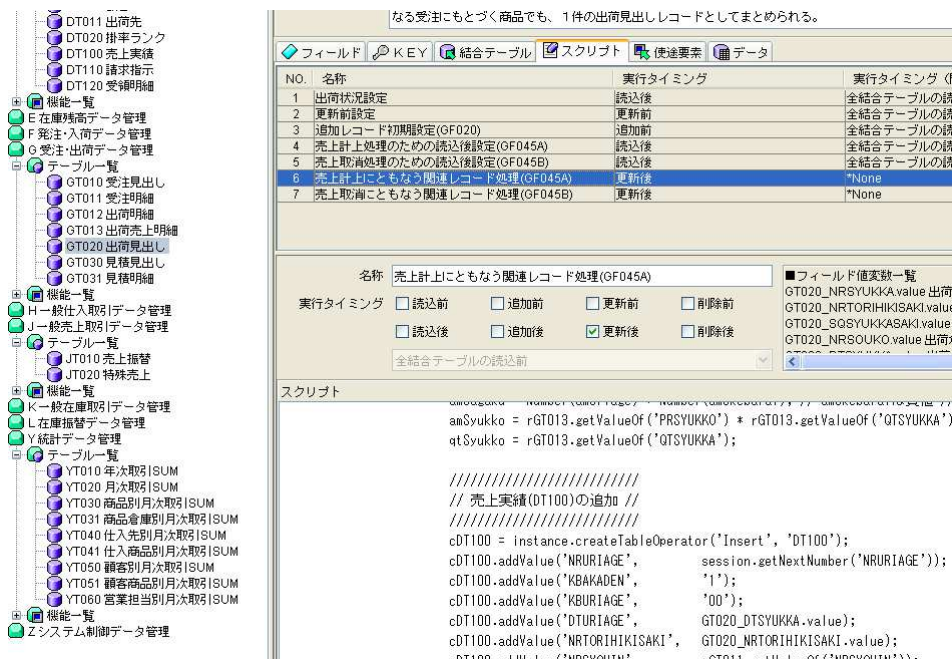


「各種売上取引にともなって売上実績が追加されると「顧客」の売掛残高が増額される。」と書いてあります。

このシステムでは、売上傳票の発行はなく、出荷伝票が発行されたときに売上実績が計上されます。

GT020 出荷見出しを選択し、スクリプトタブを選択し、6 番目にある「売上計上に伴う関連レコード処理 (GF045A)」の内容を確認します。

スクリプトの表示部をスクロールすると「売上実績 (DT100) の追加」のコメントが見られます。コードの詳細をご確認ください。



7 番目、「売上取消にともなう関連レコード処理 (GF045B)」

Table ID: GT020, Table Name: 出荷見出し

一次識別子: 出荷No

範囲KEY項目: *None

削除操作: *Delete, 有効行Where: *None

説明: 顧客（出荷先）への出荷のまとまりを管理するためのテーブル。同一出荷先へ同一のとなる受注にもとづく商品でも、1件の出荷見出しレコードとしてまとめられる。

NO.	名称	実行タイミング
1	出荷状況設定	読込後
2	更新前設定	更新前
3	追加レコード初期設定 (GF020)	追加前
4	売上計上処理のための読込後設定 (GF045A)	読込後
5	売上取消処理のための読込後設定 (GF045B)	読込後
6	売上計上にもとづく関連レコード処理 (GF045A)	更新後
7	売上取消にもとづく関連レコード処理 (GF045B)	更新後

名称: 売上取消にともなう関連レコード処理 (GF045B)

実行タイミング: ☐ 読込前 ☐ 追加前 ☐ 更新前 ☐ 削除前 ☐ 読込後 ☐ 追加後 ☒ 更新後 ☐ 削除後

全結合テーブルの読込前

スクリプト

```

var amUriage = 0; var amTax = 0; var amUriageGoukei = 0; var amTaxGoukei = 0; var
var amUkebarai = 0; var qtUkebarai = 0; var amSagaku = 0; var qtUriage = 0; var
var qtSyukko = 0; var amSyukko = 0; var qtSyukkoUpdated = 0; var amSyukkoUpdatec
var errorMessage = ''; var count = 0; var dtNend = 0; var dtMSeq = 0;

////////////////////////////////////
// 出荷売上取消処理 (GF045B) によって更新された場合の処理 //
////////////////////////////////////
if (instance.functionID == 'GF045B') {

    //////////////////////////////////////
    // 関連する顧客マスター (DT010) の検索 //
    //////////////////////////////////////
    rDT010 = instance.createTableOperator('Select', 'DT010');
    rDT010.execute('SELECT * FROM DT010 WHERE DT010.CUSTOMERID = ' + instance.getTableOperator().getPrimaryKey() + ' ');
    
```

GT020 売上見出しのレコードが更新された場合の関連レコードに対するスクリプトが記述されています。

スクリプトはテーブル側に記述されています。起動したいスクリプトを機能側から指定できるように、「if (instance.functionID == 'GF045B') {」のように呼び出し側の機能を特定できるようなオブジェクトが用意されています。

6 番目、7 番目のように、機能別に 2 本のスクリプトに別れていますが、実行はタイミングがどちらも「更新後」ですから、スクリプトは 1 本で、動作するスクリプトを、次のような構成にしても同じです。

```

if (instance.functionID == 'GF045A') {
    //売上計上に伴う関連レコード処理
}
if (instance.functionID == 'GF045B') {
    //売上取消にともなう関連レコード処理
}

```

スクリプトはテーブル自身が決められたタイミングで処理することになり、パネル等の機能は、ユーザインタフェースに限定されます。

第4章 売上実績テーブル関連動作のまとめ

1. 出荷見出しテーブル

機能'GF045A'や'GF045B'が呼び出され出荷見出しテーブルが更新されると、更新後のタイミングで、関連するテーブルの処理がなされ、掲示板機能である、売上実績テーブルも処理される。

様々な伝票が見出し、明細の親子関係の関係を持って実装され、レコードに登録してあるタイミングで、関連するテーブルをスクリプトで更新する。

テーブルにイベントを持たせることで、イベントの発生単位で責任分界を委譲できる。

フォーム側にイベントが発生し、テーブルを操作する方法は関連する全てのテーブルに対する処理を、フォーム側で記述することになり、テーブル単位での更新順序や、テーブルの CRUD、Before/After どのタイミングで実行するか複雑な記述が必要になる。

また、仕様変更が発生すると全てのテストをしなければならず、影響の範囲が見積もりにくい。

XEAD Driver のようにスクリプトは、テーブル側にあるのが正解でしょう。

テーブルは静的で、フォーム側にスクリプトを持つ方法は、最初のボタンの掛け違いと考えられるような気がします。

第2章で、「テーブルは記録媒体でしかないので、実際に記録を行う処理が必要だが、処理に関連するテーブルを限定し無秩序な処理にならないようにする工夫が必要です。」と書いたのですが、XEAD Driver はこの問題点を見事に解決しています。

2. バッチ処理

テーブル側のスクリプトだけでは、処理しきれない全体にわたる集計処理や月次の更新処理は、XF000 の機能を利用し自由に記述することが出来る。

発生単位のスクリプトは、発生側で責任を持ち処理が終わっているのならば、バッチ側の処理は掲示板となるテーブルを単純に集計するだけの簡単な内容になるでしょう。

3. ハブのお話

最初に空港の例を挙げたが、売上実績テーブルは売上取引のハブ、仕入実績、受払実績も各取引のハブとして重要な意味を持ちます。

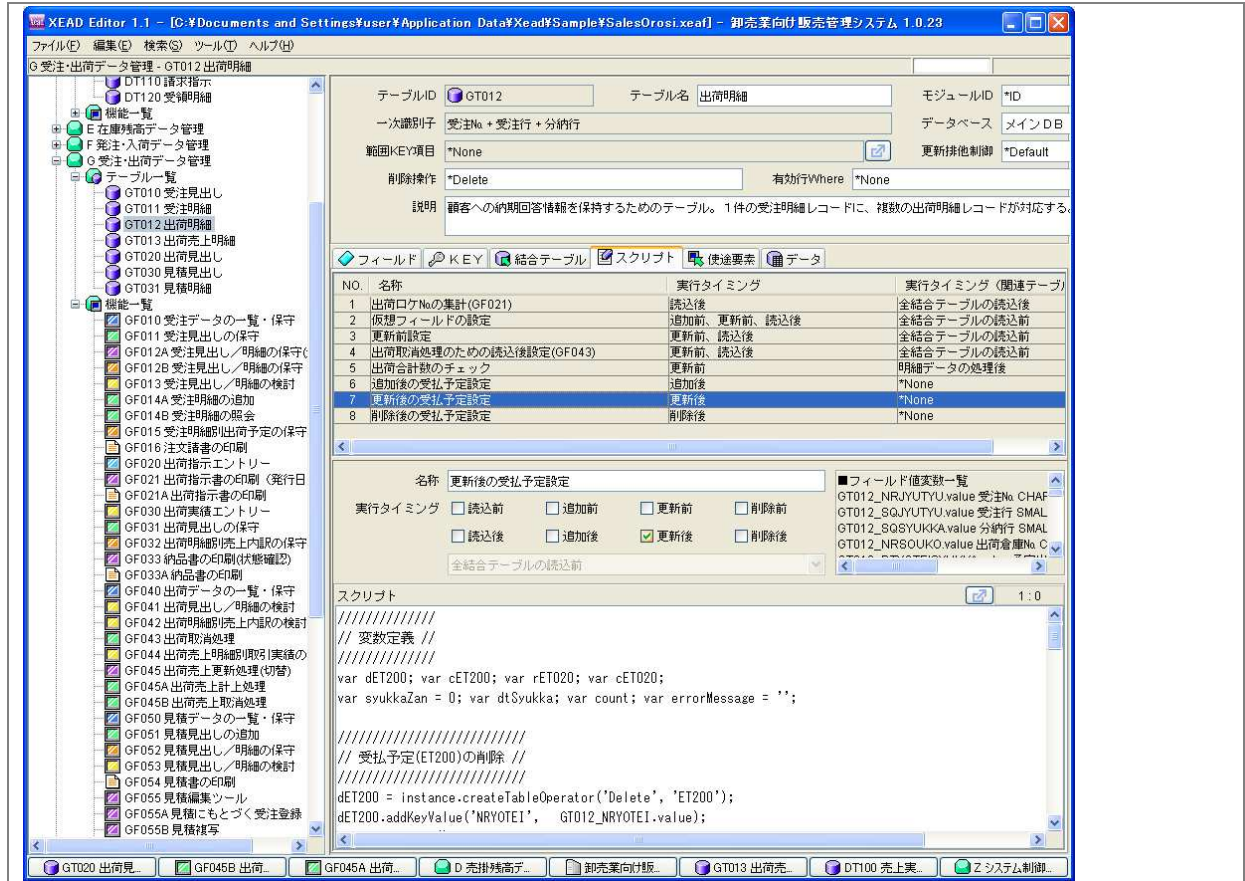
一般空港の売上見出しから、スクリプトという飛行機が情報をハブ空港へ運びます。

業務全般的には、売上実績、仕入実績、受払実績のハブ空港から、集約のサマリーテーブルへ集計処理が行われます。

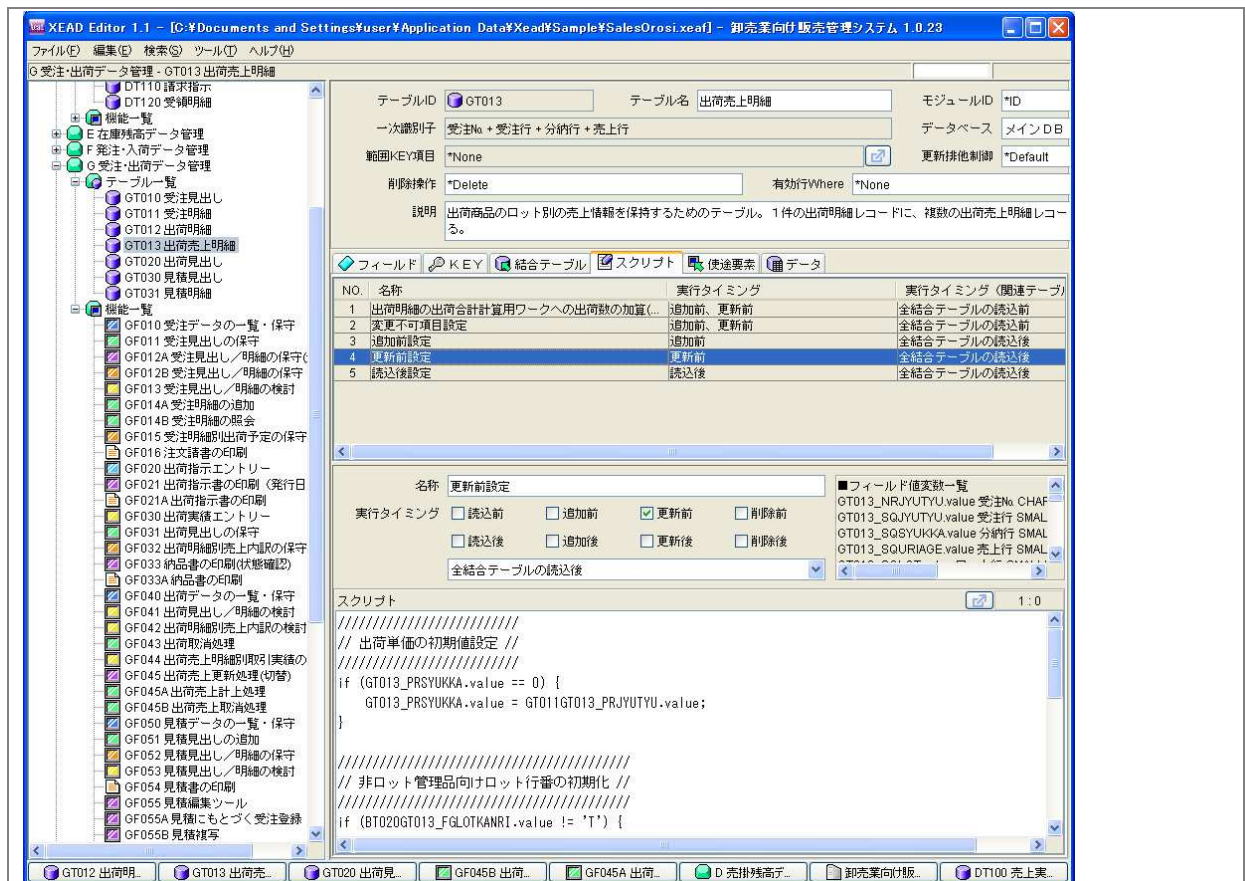
スクリプトとテーブルを組み合わせたサブシステムが実質的なハブ機能を持つ場合もあり、空港が、テーブルとサブシステムの定義が曖昧だがなんとなくイメージをつかんでいただければ幸いです。

第5章 出荷明細、出荷売上明細のスク립ト

1. XEAD Editor 出荷明細のスク립ト確認

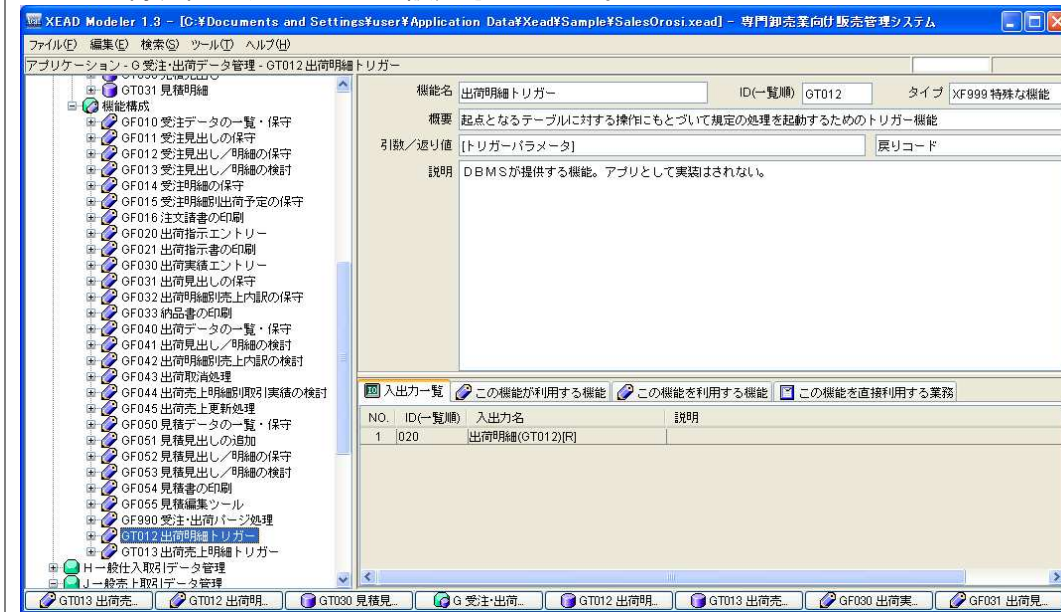


2. XEAD Editor 出荷売上明細のスク립ト確認



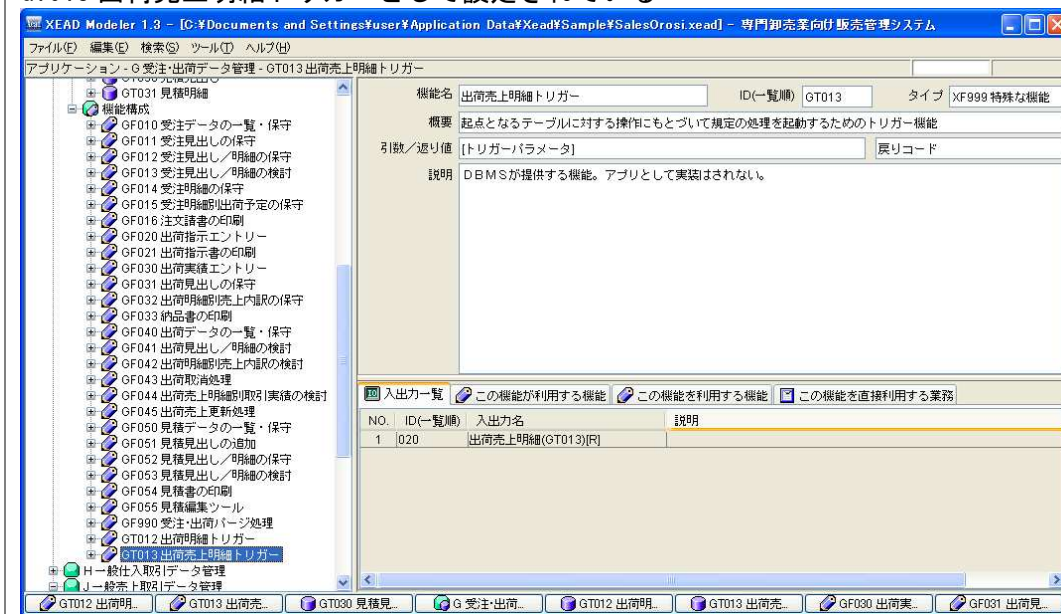
3. XEAD Modeler 出荷明細のスク립ト確認

GT012 出荷明細トリガーとして設定されている。



4. XEAD Modeler 出荷売上明細のスク립ト確認

GT013 出荷売上明細トリガーとして設定されている



5. コンセプトウェアの実装環境

XEAD Modeler で閲覧した販売管理システムでは、
GT012 出荷明細トリガー、GT013 出荷売上明細トリガー
説明の欄に「DBMS が提供する機能。アプリとして実装はされない。」と記述されている。

渡辺氏が、販売管理システムをトリガを使用した DBMS システムから XEAD Driver へ移植してきた様子が伺えます。

プラットフォームに依存しない XEAD Driver での販売管理システムを想定し、Rhino を利用した JavaScript を採用した苦勞の足跡が伺えます。

私にとって、XEAD Driver の公開なしに、コンセプトウェアから自力で販売管理システムを実装するのは実際、困難なことでした。公開に感謝します。