

شرحی مختصر بر ابزار طراحی Power-Designer

مقدمه

در راستای توسعه نرم افزارها، ابزارهایی جهت طراحی اولیه نرم افزارها بنام Case Tool وجود آمدند. یکی از این ابزارها که در این متن به شرح آن میپردازیم نرم افزار Power-Designer میباشد.

این نرم افزار دارای یک هسته اصلی با نام Power-Designer MetaWorks، میباشد که برای اجرای سه قسمت از این نرم افزار می باشد. این سه قسمت عبارتند از:

۱ - Power-Designer DataArchitect برای طراحی دیاگرام مدل منطقی داده ها (CDM)^۱

۲ - Power-Designer AppModeler for PowerBuilder برای طراحی دیاگرام مدل فیزیکی

داده ها (PDM)^۲ از روی مدل منطقی داده ها (CDM)

۳ - Power-Designer ProcessAnalyst برای طراحی دیاگرام گردش داده ها (DFD)^۳

لازم به ذکر است که این نرم افزار Document کاملی از هر یک از دیاگرامها ارائه میدهد. در ادامه شرح هر یک از قسمتهای فوق، بطور مفصل آمده است.

فصل اول - طراحی مدل منطقی داده ها

برای طراحی هر سیستم نرم افزاری باید ابتدا موجودیتهای سیستم شناخته شده و سپس بین آنها روابط (Relation) خاصی که وجود دارد را معرفی نمود. (شکل ۱-۱)

از لحاظ Cardinality، این روابط میتوانند بصورت یک به یک، یک به چند، یا چند به چند باشند. همچنین میتوان وجود رابطه را برای هر یک از دو موجودیت مرتبط "الزامی"^۴ یا "اختیاری"^۵ در تعریف کرد.

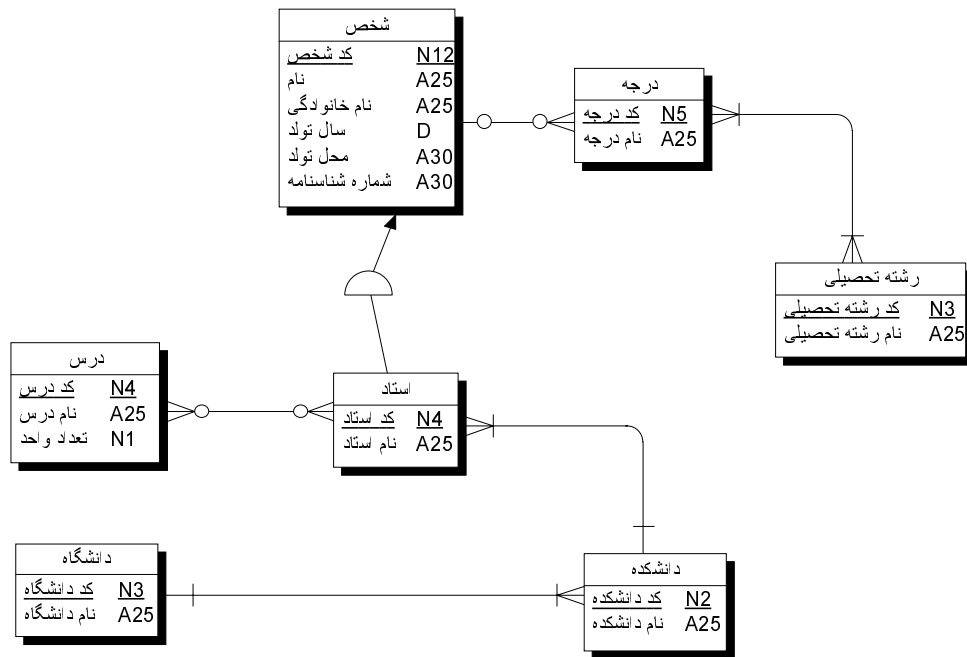
^۱ Conceptual Data Model

^۲ Physical Data Model

^۳ Data Flow Diagram

^۴ Mandatory

^۵ Optional



شکل ۱-۱: طراحی موجودیتها در DataArchitect

بصورت خلاصه و کلی میتوان عملیات قابل انجام در DataArchitect را بصورت زیر بیان نمود:

۱- تعریف صفات خاصه

از قابلیت‌هایی که برای تعریف مدل منطقی در این نرم افزار وجود دارد، تعریف صفت خاصه^۱ یا قلم داده و نوع آن و اینکه این صفت خاصه، دارای چه ویژگی‌هایی است، اعم از کلید اصلی (PrimaryKey) و غیر تهی بودن (Mandatory) صفت خاصه، میباشد.

۲- تعریف Domain

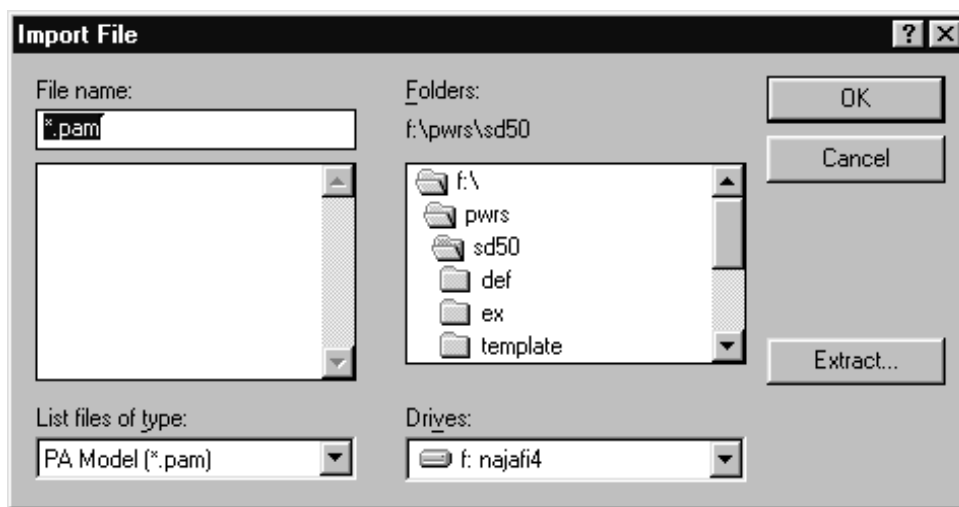
همچنین میتوان نوع داده را از نوع یک Domain خاص تعریف کرد. برای مثال در یک سیستم بانک، یک Domain با نام "Money" تعریف نماییم و آنرا از نوع "Numeric(15)" معرفی کنیم. حال هر صفت خاصه ای که مبین مقادیر پول میباشد را از نوع "Money" معرفی میکنیم. بدین ترتیب هر گاه تصمیم گرفتیم که مقادیر پول را از نوع "Numeric(20)" در نظر بگیریم، بجای اینکه بخواهیم تمام صفات خاصه "Numeric(15)" را به "Numeric(20)" تبدیل کنیم، تنها کافی است که Domain مربوطه را تصحیح کنیم، که در نتیجه هم موجب بالا رفتن سرعت ویرایش مدل داده ها، وهم باعث کاهش خطای برنامه نویس میشود.

۳ - دیکشنری داده ها

این نرم افزار دارای یک دیکشنری شامل لیست کاملی از تمام موجودیتها، صفات خاصه، Domain های تعریف شده و روابط بین موجودیتها (Relations) است. برنامه نویس میتواند با مراجعه به این دیکشنری، دید کافی از تمام اقلام تعریف شده در سیستم کسب کند.

۴ - وارد کردن صفات خاصه

میدانیم که هر موجودیت در سیستم دارای تعدادی صفات خاصه میباشد. این صفات خاصه ممکن است، قبلاً در قسمت رسم DFD بین دو پروسس ویا بین پروسس و انباره (Data Store)، رد و بدل شده باشند. در این صورت میتوان در هنگام تعریف صفات خاصه برای یک موجودیت جدید، صفات خاصه آنرا از دیاگرام گردش داده های مربوطه وارد کرد^۶. با توجه به شکل ۱-۲، این عملیات در منوی File با عنوان Power-Designer ProcessAnalyst -> Import آورده شده است.



شکل ۱-۲: وارد کردن لیست صفات خاصه از ProcessAnalyst

پس از اینکه صفات خاصه در برنامه وارد شد، میتوان با دوبار کلیک روی موجودیت مورد نظر، بوسیله گزینه Add، صفت یا صفات خاصه مربوط به موجودیت را به آن اضافه کرد.

۵ - تعریف یک یا چند زیر مدل

مدل اصلی که تقریباً تمام موجودیتها در آن قرار داده میشوند، میتواند شامل زیر مدلهایی باشد که هر یک از این زیر مدلها شامل تعدادی از کل موجودیتهای مدل اصلی میباشد.

تقسیم بندی مدل اصلی به تعدادی زیر مدل و اینکه هر زیر مدل باید شامل چه موجودیتهای و روابطی باشد، با توجه به رابطه منطقی بین موجودیتهای صورت میگیرد.

برای مثال یک سیستم جامع دانشگاه شامل زیر سیستم های آموزش، امور دانشجویی، امور تغذیه، انبار، انتشارات، کتابخانه و ... را در نظر بگیرید. هر یک از این زیر سیستمها دارای یک سری موجودیتهای مخصوص به خود و یک سری موجودیتهای عمومی (مانند "اشخاص"، "پرسنل" و ...) میباشد. هر یک از این زیر سیستم ها میتواند دارای یک زیر مدل شامل موجودیتهای و روابط خصوصی آن زیر سیستم و موجودیتهای و روابط کلی مربوط (مشترک در بین دو یا بیش از دو زیر سیستم) باشند.

با استفاده از منوی Dictionary/Submodel/New میتوان یک زیرمدل جدید تعریف نمود. وقتی زیر مدلی را ایجاد کردیم، شروع به قرار دادن موجودیتهای و روابط مربوط به آن مینماییم. در صورتی که موجودیت یا رابطه مد نظر قبلاً ایجاد شده باشد، دیگر احتیاجی به تعریف دوباره نداریم، بلکه با استفاده از منوی Dictionary/Submodel/Add-Remove Object میتوان یک شیء را در زیر مدل قرار داد یا از آن حذف کرد. (البته این حذف بصورت منطقی است و شیء مورد نظر را بکلی حذف نمیکند، بلکه تنها آن را از زیر مدل بر میدارد).

۶ - بروز رسانی (Update) نمایش گرافیکی

این دستور برای زیر مدلهای بکار میرود و از منوی Dictionary/Submodel/Update Graphics اجرا میگردد. بدینوسیله میتوان تمام روابطی را که بین دو موجودیت نمایان روی زیر مدل، وجود دارد را نشان داد. برای مثال اگر دو رابطه $r1$ و $r2$ بین دو موجودیت $e1$ و $e2$ وجود داشته باشد، و روی صفحه زیر مدل تنها $e1$ و $e2$ را داشته باشیم و بخواهیم ببینیم که چه روابط دیگری بین $e1$ و $e2$ وجود دارد، این دستور را بکار میبریم.

۷ - پنهان سازی و آشکار سازی اشیاء

در یک زیر مدل در صورتی که تعداد موجودیتهای و روابط خیلی زیاد و پیچیده شود، گاهی اوقات نیاز است که بعضی موجودیتهای یا روابط را بصورت موقت پنهان کنیم. بدین ترتیب اگر شیء مورد نظر یک موجودیت باشد، آن موجودیت به همراه تمام روابط متصل به آن از دید مخفی میگردد، و اگر شیء مزبور یک رابطه باشد، فقط همان رابطه از دید مخفی میشود. البته ناگفته نماند که مخفی نمودن یک شیء از روی صفحه به معنی حذف آن شیء از مدل یا زیرمدل مزبور نیست.

۸ - کنترل خطا در مدل

میتوان برای اطمینان از صحت طراحی انجام گرفته، با استفاده از دستور Check Model، ایرادات کلی مدل را پیدا نموده و در صدد رفع این ایرادات برآمد.

این خطاها شامل خطاهای موجود در Data Items یا صفات خاصه، در Entities یا موجودیتها و Relations یا روابط بین موجودیتها میباشد.

برای مثال در صورتی که یک موجودیت فاقد صفات خاصه باشد:

Error: Entity Name has no attribute

گاهی نیز این خطاها به صورت هشدار یا Warning داده میشود:

Warning: Entity Name has no Identifier

۹ - تولید مدل فیزیکی

این ابزار قابلیت تبدیل مدل منطقی به مدل فیزیکی را داراست. بدین صورت که هر موجودیت به یک جدول (Table) و هر رابطه "یک به چند" به یک کلید خارجی در جدول مرتبط با ارتباط چند تایی تبدیل میشود. روابط "چند به چند" نیز به یک جدول جداگانه تبدیل میشود و که کلید اصلی آن مجموعی از دو کلید اصلی در دو جدول مرتبط خواهد بود. این عمل را میتوان با استفاده از دستور Generate Physical Model انجام داد. البته باید توجه داشت که قبل از تولید مدل فیزیکی عمل Check Model را برای کنترل خطاها انجام دهیم.

۱۱ - ادغام دو یا چند زیر مدل

با استفاده از دستور Utilities / Merge میتوان یک مدل منطقی دیگر را به مدل منطقی فعلی اضافه نمود. این عمل در مواردی که بخواهیم دو زیر مدل از یک سیستم را باهم یکی کرده و به عنوان یک زیر مدل واحد معرفی نماییم بکار میرود.

۱۲ - سلیقه ها Preferences

میتوان یک سری از انتخاب ها را برای این ابزار تنظیم نمود. در زیر لیستی از این سلیقه آمده است:

۱. Auto Adjust to Text: ابعاد موجودیت در نمایش گرافیکی آن، با توجه به متن آمده در درون آن تنظیم میشود.
۲. Scale: ابعاد کل نمایش گرافیکی نسبت به صفحه را مشخص میکند، برای مثال اگر Scale = %200 باشد، کل شکل گرافیکی مدل، ۲ برابر خواهد شد. البته این بزرگ شدن با ZOOM متفاوت است، زیرا نمایش صفحه در عمل چاپ نیز دو برابر خواهد شد.
۳. نحوه نمایش موجودیتها: برای مثال آیا کد شناسایی موجودیتها هم در آن بنویسد یا خیر و یا اینکه آیا نوع داده ها را را نمایش دهد یا خیر؟

۴. حداکثر طول نام اشیاء (NAME) و اینکه به حروف بزرگ و کوچک حساس باشد یا خیر را میتوان معین کرد. همچنین حروف مجاز برای نام اشیاء و حروف غیر مجاز برای نام اشیاء را میتوان مشخص نمود.
۵. حداکثر طول کد اشیاء (CODE) و اینکه به حروف بزرگ و کوچک حساس باشد یا خیر را میتوان معین کرد. همچنین حروف مجاز برای کد اشیاء و حروف غیر مجاز برای کد اشیاء را میتوان مشخص نمود.
۶. رنگ و فونت را نیز میتوان بطور کلی برای موجودیتها، روابط، وراثت، وارث و موروث ها تعیین نمود.

۱۳ - تولید مستندات طراحی

از قابلیت‌های ویژه این ابزار، تولید مستندات طراحی میباشد. این مستندات (Documents) قابل رویت در Microsoft Word میباشد و فرمت آنها بصورت RTF میباشد. این گزارش تصویر دقیقی از طراحی منطقی داده‌ها، موجودیتها و رابطه بین آنها ارائه میکند. این عمل با استفاده از دستور Create Report انجام میشود.

مواردی که میتواند در این گزارش آورده شود عبارتند از:

۱. فهرست
۲. گراف مدل اصلی
۳. گراف هر یک از زیر مدلها
۴. لیستی از Domain ها با شرح کامل خصوصیات آن
- ویژگیهای هر یک و یک لیست مرجع (Reference List) که مشخص میکند آن Domain در چه جاهایی بکار رفته است.
۵. لیستی از تمام صفات خاصه (Data Items)
- ویژگیهای هر یک و یک لیست مرجع (Reference List) که مشخص میکند آن قلم داده در چه جاهایی بکار رفته است.
۶. لیستی از تمام موجودیتها (Entities)
- ویژگیهای هر یک و لیستی از اقلام داده‌های تعریف شده برای آن
۷. لیستی از تمام روابط تعریف شده (Relations) و ویژگیهای هر یک
۸. لیستی از تمام مراجع وراثت (Inheritance)
- ویژگیهای هر یک و لیستی از فرزندان آنها (Childs)

فصل دوم – طراحی مدل فیزیکی داده ها

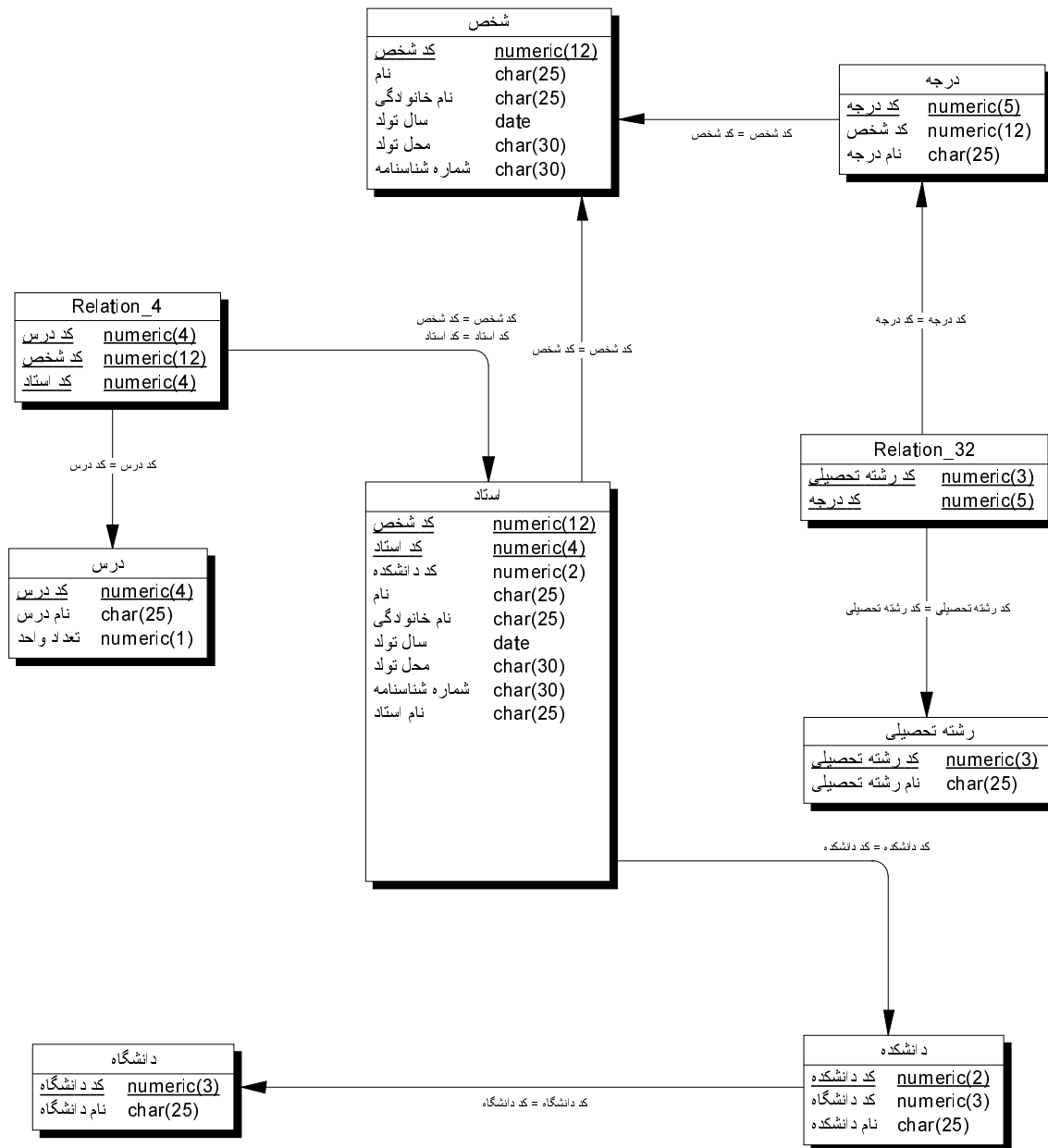
همانطور که در بخش قبل دیدیم مدل فیزیکی از روی مدل منطقی داده ها بدست میآید. همچنین این مدل با توجه به پایگاه داده های مورد نظر برای پیاده سازی، ساخته میشود.

این پایگاه های داده در نسخه 5.1.0 این نرم افزار به شرح ذیل میباشد:

1. ALLBASE /SQL F.0
2. ANSI Level 2
3. AS/400
4. CA Clipper 5.01
5. CA OpenIngres
6. CA-DB
7. DATACOM
8. DB2
9. DB2 Version 2
10. dBASE 5.0 for windows
11. dBASE IV (SQL)
12. Empress
13. Foxpro 2.5
14. Foxpro for windows 2.5
15. INFORMIX 4GL 5.0
16. INFORMIX SQL 5.0 , 6.0
17. INFORMIX SQL 7.1
18. INGRES Version 5
19. INGRES Version 6.4
20. InterBase 4.0
21. MAGIC II V5.0
22. Microsoft Access 1.0
23. Microsoft Access 1.1
24. Microsoft Access 2.0
25. Microsoft Access 95
26. Microsoft SQL Server 4.x
27. Microsoft SQL Server 6.0
28. Microsoft Visual Foxpro 3.0
29. NonStop SQL
30. NS-DBR
31. ODBC
32. OMNIS Version 3.0
33. ORACLE Version 5.0
34. ORACLE Version 6.0
35. ORACLE Version 7.x
36. OS/2 Database Manager
37. PARADOX 5.0 for windows
38. PARADOX for DOS (PAL)
39. PARADOX for windows (OPAL)

40. Progress
41. Q+E 3.0
42. R:BASE Version 3.1
43. R:BASE Version 4.5
44. R:BASE Version 5.5
45. RDB 4.0
46. RDB 4.2
47. RDB 5.0
48. RDB 6.0
49. SQL/Data System
50. SQLBase 5.0
51. SQLBase 6.0
52. SuperBase IV
53. Supra
54. Sybase 4.x
55. Sybase SQL Anywhere
56. Sybase System 10
57. Sybase System 11
58. Teradata
59. Unify 2000 V2.1
60. UniVerse 7.0
61. Visual Basic Enterprise 4.0
62. Visual Basic Professional 3.0
63. Visual dBASE 5.0
64. Watcom SQL 3.0
65. Watcom SQL 4.0
66. XDB 3.0

برای مثال شکل ۱-۲ مدل فیزیکی داده ها را برای شکل ۱-۱ که در بخش قبل دیده شد را نشان میدهد:



شکل ۱-۲: مدل فیزیکی در AppModeler for PowerBuilder

در این ابزار که AppModeler for PowerBuilder نام دارد، تقریباً تمام عملیات قابل انجام در Data Architect، قابل انجام است و ما در اینجا از توضیح دوباره آنها صرف نظر نموده و به شرح سایر عملیات که مختص به آن است میپردازیم:

۱ - دیکشنری داده ها

این ابزار دارای یک دیکشنری شامل لیست کاملی از تمام جداول، ستونها یا Columns، Domain های تعریف شده، شاخصها یا Indexes و لیستی از ارجاعات (References) است.

۲ - تعریف دید (Views)

در اینجا میتوان دیدهای مختلفی را از جداول موجود در مدل فیزیکی داده ها تعریف نمود. برای هر دید، چندین ویژگی تعریف میشود. این ویژگیها عبارتند از:

- لیست جداول موجود در دید
- Query تولید کننده این دید
- SQL تولید شده از Query و یا SQL نوشته شده بوسیله کاربر

۳ - تعریف Trigger

برای هر موجودیت بطور جداگانه میتوان شش نوع Trigger تعریف کرد:

۱ - قبل از Insert

۲ - بعد از Insert

۳ - قبل از Update

۴ - بعد از Update

۵ - قبل از Delete

۶ - بعد از Delete

البته خود Power-Designer بطور پیش فرض برای تمام موجودیتها این Trigger ها را در نظر گرفته است. میتوان با دستور Trigger Template الگوی تولید Trigger را برای تمام موجودیتها داد و با توجه به این الگوی کلی، Trigger های موجودیتها را تولید نمود.

۴ - دوباره سازی Index ها

در مدل فیزیکی داده ها هم میتوان ستونهای جدول را تغییر داده و آنها را کم و زیاد کرد و حتی خواص مربوط به ستونها را (یعنی Primary, Foreign, Mandatory) عوض کرد. میبینید که با عوض کردن خواص یک ستون ممکن است تغییری کلی در شاخص های جدول ایجاد شود، بدین ترتیب نیاز است تا شاخص ها از نو ایجاد گردند. این فرمان با اجرای دستور Rebuild Indexes قابل انجام خواهد بود.

۵ - دوباره سازی Reference ها

در مدل منطقی داده ها هر موجودیت با موجودیت دیگر بوسیله یک رابطه یا Relation اتصال می یافت، ولی در مدل فیزیکی داده ها این کار بوسیله یک Reference صورت میگیرد. حال در صورتی که ستونی که مربوط به یک Reference است، (ستون کلید خارجی) را حذف نماییم احتیاج به دوباره سازی Reference ها با استفاده از دستور Rebuild References داریم.

۶- ساختن مدل منطقی داده ها

میتوان از روی مدل فیزیکی داده ها مدل منطقی داده های آن را تولید نمود. این عمل با استفاده از دستور Generate Conceptual Model قابل انجام است.

۷- کنترل خطا

این عمل برای جلوگیری از خطاهای احتمالی طراح در حین طراحی صورت میگیرد و اصطلاحاً بنام Check Model شناخته میشود.

این خطاها میتواند خطاهای مربوط به جداول، Reference ها و شاخص ها باشد.

۸- تغییر پایگاه داده مربوط به مدل فیزیکی

در صورتی که مدل فیزیکی منطبق بر پایگاه داده X تبیین شده باشد و بخواهیم آنرا بر طبق پایگاه داده Y تبیین نماییم، میتوان از دستور Change Target Database استفاده کرد.

۹- تولید یک فایل Script

با توجه به پایگاه داده انتخاب شده برای مدل فیزیکی، یک فایل Script میتوان تولید نمود که در آن پایگاه داده میتوان استفاده کرد تا بتوان فایلهای حقیقی پایگاه را استخراج کرد. برای مثال اگر پایگاه داده های انتخابی Sybase SQL Anywhere باشد، یک Script با پسوند sql تولید میشود و سپس میتوان با طی مراحل زیر به نتیجه مطلوب رسید:

۱- ISQL یا مفسر SQL را اجرا کنید

۲- فایل crebas.sql را باز کنید

۳- Script را اجرا کنید

مثالی دیگر: اگر پایگاه داده های انتخابی Visual Basic Professional 3.0 باشد، یک Script با پسوند bas و یک پروژه با پسوند mak تولید میشود. حال مراحل ذیل را طی مینماییم:

۱- Visual Basic Professional 3.0 را اجرا کنید

۲- یک Access Database جدید را بوسیله Data Manager در منوی Window بسازید

۳- پروژه crebas.mak را باز کنید

۴- برنامه را اجرا کنید (با دستور Start از منوی Run)

۵- نام فایل Database با پسوند MDB را وارد کنید و OK را بزنید

۶- جداول را با استفاده از Data Manager در منوی Window رؤیت نمایید

در هر صورت بنا به پایگاه داده انتخابی، ممکن است خروجی S-Designer برای آن متفاوت باشد و نحوه بکار بردن آن هم مختلف باشد. Power-Designer پس از تولید فایل Script، نحوه بکار بردن آنرا توضیح میدهد.

این کار توسط اجرای دستور Generate Database و انتخاب دکمه Generate Script صورت میپذیرد.

۱۰ - تولید پایگاه داده

در صورتی که در سیستم ODBC Data Source نصب شده باشد و امکان Connect شدن به آن را داشته باشیم، میتوان پایگاه داده خود را در یکی از Data Source های موجود یا یک Data Source جدید ایجاد نمود.

این کار توسط اجرای دستور Generate Database و انتخاب دکمه Generate Database صورت میپذیرد.

۱۲ - تغییر پایگاه داده

در صورتی که قبلاً یک پایگاه داده را ایجاد نموده باشیم و حال در مدل فیزیکی تغییراتی را اعمال کرده باشیم، برای انعکاس دادن این تغییرات در پایگاه داده میتوان از دستور Modify Database استفاده کرد.

برای تغییر در پایگاه داده ابتدا باید مدل فیزیکی یا PDM آنرا تبدیل به مدل بایگانی شده یا در اصطلاح Archived PDM نمود که با پسوند APM ایجاد میشود. در غیر اینصورت امکان اعمال تغییرات در پایگاه داده ممکن نخواهد بود.

۱۳ - محاسبه اندازه پایگاه داده

در صورتی که تعداد رکورد های هر یک از جداول بصورت تخمینی توسط کاربر مشخص شده باشد، میتوان اندازه پایگاه داده را معلوم نمود. این کار توسط دستور Compute Database Size صورت میپذیرد.

۱۴ - تولیداتی برای محیط برنامه نویسی Power Builder

محیط برنامه نویسی Power Builder توسط شرکت Sybase تولید شده است و در آن میتوان بصورت Visual یک نرم افزار را پیاده سازی نمود.

ابزار Power-Designer نیز محصولی از شرکت Sybase میباشد که قادر است یک سری خروجی های مفیدی را برای محیط Power Builder تولید کند.

تمام این امکانات در منوی 4GL فراهم شده است و میتوان به موارد ذیل اشاره نمود:

1 - Generate Power Builder Application

تولید برنامه برای Power Builder

2 - Generate Power Builder Query

تولید دید (View) برای Power Builder

3 - Generate Power Builder Attributes

تعریف Attribute های لازم در Power Builder

4 - Reverse Power Builder Attributes

معکوس سازی Attribute های Power Builder

فصل سوم – طراحی دیاگرام گردش داده ها

بخش سوم از طراحی با این ابزار مربوط است به طراحی دیاگرام مربوط به توابع سیستم و داده هایی که این توابع با هم رد و بدل میکنند. این بخش با نام ProcessAnalyst میباشد. این بخش نیز کماکان ویژگیهای کلی که در بالا به آن اشاره شد را در بر دارد، در اینجا تنها به ویژگیهای خاص مربوط به این قسمت میپردازیم:

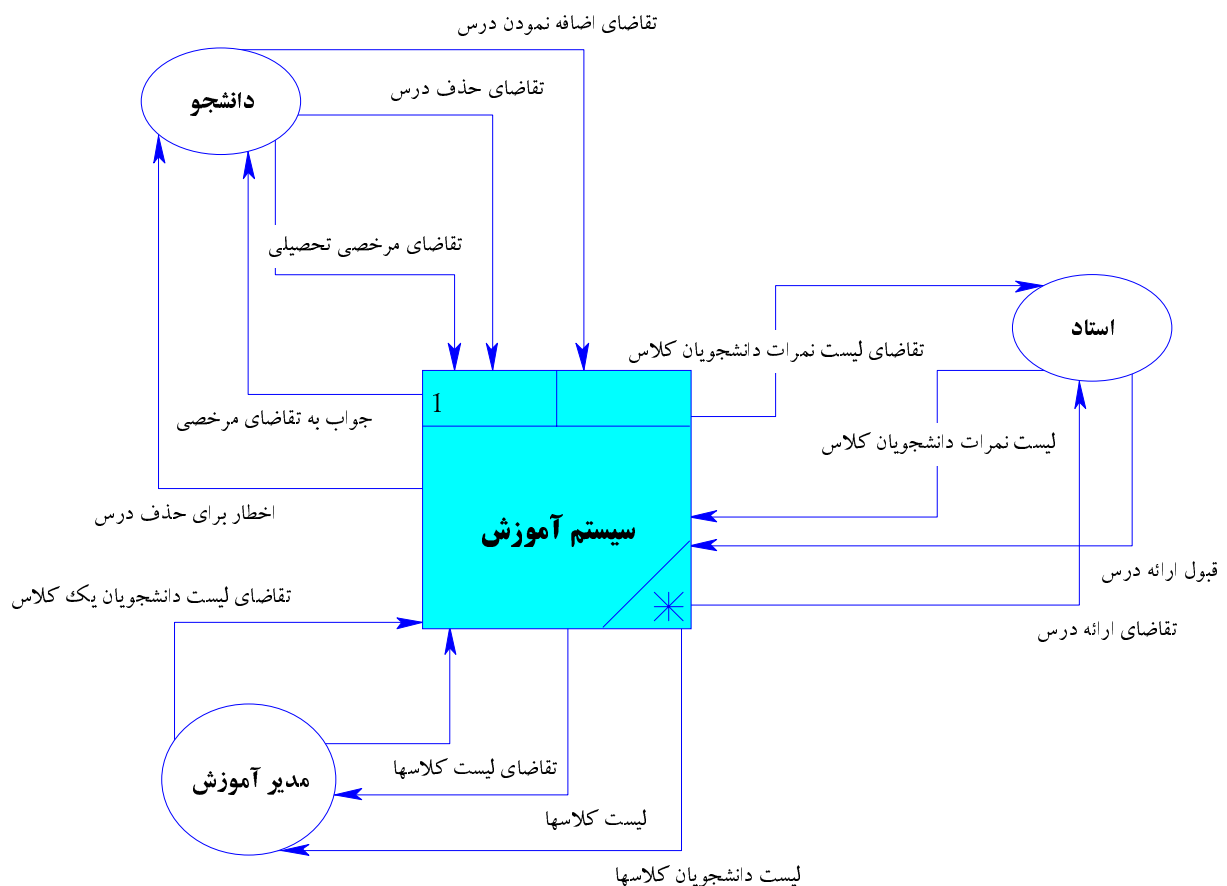
۱- روش های طراحی

میتوان برای طراحی در این ابزار از چهار روش زیر استفاده کرد:

OMT Functional Model

- Yourdon / DeMarco
- Gene & Sarson
- SSADM

برای مثال یکی از دیاگرامهای تهیه شده توسط روش SSADM در شکل ۱-۳ آورده شده است.



شکل ۱-۳: Context Diagram مربوط به سیستم آموزش در ProcessAnalyst

۲ - دیکشنری داده ها

این بخش دارای یک دیکشنری مشتمل بر موارد ذیل میباشد:

- Processes
- Data Stores
- External Entities
- Data Flows
- Domains
- Data Items

که در هر قسمت تمام ویژگیهای مربوط به آن شیء آمده و قابل تغییر و اضافه یا حذف نمودن میباشد.

۳ - چند سطحی بودن دیاگرام

درون هر یک از Process ها میتوان یک دیاگرام دیگر در نظر گرفت و این عمل را به دفعات میتوان تکرار نمود به طوری که هر چه جلوتر میرویم، توابع کوچکتر و واضح تر میشوند. با هر بار شکسته شدن یک پروسس یک سطح به جلو میرویم و یک Level به آن اضافه میشود. این عمل تا رسیدن به یک پروسس جزئی ادامه می یابد.

۴ - کنترل خطا ها

برای اطمینان از صحت طراحی صورت گرفته میتوان عمل Check Model را برای دیاگرام موجود انجام داد. این خطاها میتواند خطاهای مربوط به اجزای زیر باشد:

- Processes
- Data Stores
- External Entities
- Data Flows
- Domains
- Data Items

۵ - نسبت دادن Data Item های مربوط به هر Flow

برای هر Flow رسم شده بین دو پروسس یا Data Store میتوان Data Item ها یا مقادیر داده ای را که رد و بدل میشود را بیان نمود.

۶ - اتوماتیک بودن ایجاد Data Item های یک Data Store

در صورتی که یک Flow یا جریان داده از یک Data Store خارج و یا به آن داخل گردد، ابزار Power-Designer به طور اتوماتیک داده های مربوط به آن Flow را درون Data Store مربوطه قرار میدهد.