تعریف چند سروری و چند دیتابیسی :

ارتباط بین جدول Product از دیتابیس محصولات (DBProduct) و جدول ProducItem از دیتابیس خدمات (DBProductService ) که نه تنها از پایگاه داده مختلف می باشند بلکه به روی سرور های مختلفی قرار دارند. به این ترتیب امکان استفاده از این رابطه در ورود اطلاعات هر طرف ممکن است. همچنین امکان استفاده از این ارتباط در جستجو و نمایش اطلاعات میسر می باشد. برای مثال ستون نوع محصول که از جدول ProductType در DBProducts می باشد در لیست نمایش پیش فرض ProducItem تعریف شده است که البته به علت بین سروری بودن این ارتباط تعریف لینک سرور در پایگاه داده مورد نیاز است.

مشخصات موجودیتها :

امکان بارگذاری خودکار داده ها : جداول پایه مانند برند و نوع گارانتی

تولید خودکار لیست نمایش پیش فرض :

قانون : 1- ستونهای کلید اصلی موجودیت اضافه میشوند 2- ستونهای ساده غیر کلید اصلی که کلید فرعی نیز نیستند بر اساس 3 مرحله زیر شناسایی و اضافه میشوند. مراحل : ....

3- ستونهای روابط اضافه میشوند (شرح جزئیات)

تولید خودکار لیست جستجوی پیش فرض

قانون : 1- ستونهای کلید اصلی موجودیت اضافه میشوند 2- ستونهای ساده غیر کلید اصلی که کلید فرعی نیز نیستند بر اساس 3 مرحله زیر شناسایی و اضافه میشوند. مراحل : ....

3- ستونهای روابط اضافه میشوند (شرح جزئیات)

# امکان تعریف موجودیت ها:

موجودیتها در مدل فراداده از پایگاه داده هدف استخراج می شوند. هر موجودیت می تواند عیناً معادل یک جدول در پایگاه داده هدف باشد و یا انشعابی از یک جدول. فارق از ساختار اصلی موجودیت ها که شامل خصوصیات و روابط و .. می باشد، بسیاری از بخشهای مدل فراداده بر اساس موجودیت ها تعریف می شوند که در جای خود به آنها و ارتباطشان با موجودیتها اشاره خواهد شد. امّا خود موجودیت در مدل فراداده دارای خصوصیات زیر است.

|  |  |
| --- | --- |
| عنوان | توضیحات |
| Name |  |
| Alias |  |
| IsEnabled |  |
| IsReadonly |  |
| EntityListViewID |  |
| EntitySearchID |  |
| SearchInitially |  |
| SelectAsComboBox |  |
| IsOrginal |  |
| DeterminerColumnID |  |
| DataMenuSettingID |  |

# امکان تعریف انواع ستونها:

توضیحات ستون.....

## ستون رشته ای:

برای این نوع ستون موجودیت سیستمی StringColumnType تولید می شود (c3583c1c-dcc2-42c1-979e-aa893da6b6e7) و دارای خصوصیات زیر است.

|  |  |
| --- | --- |
| عنوان | توضیحات |
| MaxLength | طول ماکزیمم رشته برای اعتبارسنجی؛ این خصوصیت مستقیماً از پایگاه داده هدف بروزرسانی می شود (608254b4-72f0-419f-a038-ad4ff657a6ff)؛ نمونه : خصوصیت کد (Code) در موجودیت دفتر (Office) |
| MinLength | حداقل طول رشته برای اعتبارسنجی؛ این خصوصیت در برنامه مدیریت فراداده قابل مقداردهی است(bf58c975-a36d-4a12-b7a3-1cbf252eff52)؛ نمونه : خصوصیت کد (Code) در موجودیت دفتر (Office) |
| Format | فرمت مورد قبول داده برای اعتبارسنجی؛ این خصوصیت در برنامه مدیریت فراداده قابل مقداردهی است(bf58c975-a36d-4a12-b7a3-1cbf252eff52)؛ نمونه : خصوصیت آدرس ایمیل در موجودیت شخص (GenericPerson) |

بر حسب طول رشته، در تولید ظاهر ورود اطلاعات خصوصیت یا همان ColumnUISetting ، انواع زیر در نظر گرفته می شوند :

#### رشته کوتاه :

|  |  |
| --- | --- |
| عنوان | توضیحات |
| تعریف | رشته هایی برای تعریف مقادیر مثل نام و نام خانوادگی |
| تولید ظاهر | به صورت پیش فرض طول این ستونها کمتر یا مساوی 64 کاراکتر فرض می شود و ColumnUISetting با یک ستون معمولی و یک ردیف تولید می شود  ad8db1ea-07d5-481a-95fd-5c93e6562e87 |
| نمونه | نام در موجودیت شخص حقیقی (RealPerson) |

#### رشته متوسط :

|  |  |
| --- | --- |
| عنوان | توضیحات |
| تعریف | رشته هایی برای تعریف مقادیر مثل عنوان یک پرونده |
| تولید ظاهر | به صورت پیش فرض طول این ستونها بیشتر از 64 و کمتر یا مساوی 256 کاراکتر فرض می شود و ColumnUISetting با یک ستون نیم صفحه و یک ردیف تولید می شود  ad8db1ea-07d5-481a-95fd-5c93e6562e87 |
| نمونه | نام در موجودیت شخص حقوقی (LegalPerson) |

#### رشته بلند :

|  |  |
| --- | --- |
| عنوان | توضیحات |
| تعریف | رشته هایی برای تعریف مقادیر مثل آدرس |
| تولید ظاهر | به صورت پیش فرض طول این ستونها بیشتر از 256 فرض می شود و ColumnUISetting با یک ستون تمام صفحه ایجاد می گردد، اگر طول رشته کوچکتر یا مساوی 1024 بود ColumnUISetting یک ردیفی، اگر بزرگتر از 1024 بود (و نامحدود نبود) دو ردیفی و اگر طول رشته نامحدود بود 3 ColumnUISetting ردیفی می باشد.  ad8db1ea-07d5-481a-95fd-5c93e6562e87 |
| نمونه | مثال : آدرس در موجودیت GenericPersonAddress و یا توضیحات در موجودیت درخواست سرویس (ServiceRequest) |

## ستون عددی :

برای این نوع ستون موجودیت سیستمی NumericColumnType تولید می شود (e64b16d0-bfc0-46fd-9116-67b59e86132f) که دارای خصوصیات زیر است :

|  |  |
| --- | --- |
| عنوان | توضیحات |
| Precision | طول بخش صحیح عدد؛ این خصوصیت اگرچه در برنامه مدیریت فراداده قابل مقداردهی است(c99a93a8-756a-42a2-a9f2-d346c3f0299b) امّا در بروزرسانی جداول پایگاه داده هدف شناسایی شده و مقدار آن جایگزین می شود(6c99db26-bb1b-40df-ba63-1e9a8bbe5eff). |
| Scale | طول بخش اعشاری عدد؛ این خصوصیت اگرچه در برنامه مدیریت فراداده قابل مقداردهی است(c99a93a8-756a-42a2-a9f2-d346c3f0299b) امّا در بروزرسانی جداول پایگاه داده هدف شناسایی شده و مقدار آن جایگزین می شود(6c99db26-bb1b-40df-ba63-1e9a8bbe5eff). |
| MinValue | حداقل مقدار برای اعتبارسنجی؛ این خصوصیت در برنامه مدیریت فراداده قابل مقداردهی است(c99a93a8-756a-42a2-a9f2-d346c3f0299b). |
| MaxValue | حداکثر مقدار برای اعتبارسنجی؛ این خصوصیت در برنامه مدیریت فراداده قابل مقداردهی است(c99a93a8-756a-42a2-a9f2-d346c3f0299b). |
| Delimiter | جداکننده سه رقمی عدد؛ این خصوصیت در برنامه مدیریت فراداده قابل مقداردهی است(c99a93a8-756a-42a2-a9f2-d346c3f0299b) امّا به هنگام ایجاد ستون عددی، به صورت پیش فرض توسط قطعه کدی تشخیص داده می شود(9d5217f1-8a54-49a1-a1d2-0ff4dad8df1a). |

در تولید ظاهر خصوصیات عددی به صورت کلی، به صورت پیش فرض ColumnUISetting با یک ستون معمولی و یک ردیف ایجاد می گردد. ( ad8db1ea-07d5-481a-95fd-5c93e6562e87)

در تولید کنترل ورود اطلاعات اینکه خصوصیت اعشاری است یا خیر و یا اینکه جداکننده دارد یا خیر در نظر گرفته می شود. (f8981e30-c94d-44be-80e0-0ba321de6d67)

نمونه هایی از خصوصیات عددی در زیر اشاره شده است:

عدد بدون اعشار : ستون شماره شناسنامه (BirthCertificateNumber) در موجودیت شخص حقیقی (RealPerson)

عدد اعشاری: ستون نمره (UserRate) در موجودیت خلاصه سرویس (ServiceConclusion)

عدد مبلغ: ستون مبلغ (Price) در موجودیت خدمات اضافه (ServiceAdditionalItem)

## ستون تاریخ :

برای این نوع ستون موجودیت سیستمی DateColumnType تولید می شود. ستونهای تاریخ یا به صورت تاریخ میلادی در پایگاه داده تعریف می شوند(5a4645cf-0487-48d5-89bb-65590717ebfd) و یا به صورت رشته که خود می تواند دارای تاریخ میلادی و یا شمسی باشد(e0a2dd6f-ec52-4430-b15f-9e272e60f7cb). به هنگام ایجاد ستونهای جدید رشته ای، قطعه کدی تاریخ بودن ستون را پیش بینی می کند(caf85527-3025-4e25-8b24-50861ad0329b). در برنامه مدیریت فراداده نیز ستونهایی که از نوع رشته می باشند قابلیت تبدیل به نوع ستون تاریخ را دارا می باشند (559a48c0-417b-4813-8b9e-b6ad00bdd936). این ستونها مجدداً می تواند به نوع رشته بازتبدیل شوند(8351e66f-a105-44ba-8a5c-7715aa287708).

موجودیت DateColumnType دارای خصوصیات زیر است :

|  |  |
| --- | --- |
| عنوان | توضیحات |
| ShowMiladiDateInUI | نمایش تاریخ میلادی در واسط کاربری؛ مقدار این خصوصیت در درجه اول مستقیماً به روی ستون در برنامه مدیریت فرارداده قابل تعیین شدن است، در صورت عدم مقداردهی مستقیم روی ستون (Null بودن مقدار)، از تنظیمات پایگاه داده در برنامه مدیریت فراداده خوانده می شود. در آنجا هم مقدار اولیه False است بنابراین بصورت پیش فرض برای تمامی ستونهای تاریخ در واسط کاربری تاریخ شمسی نمایش داده می شود (5bd4e7ec-d66f-4476-a4e7-0c4c410b98b1). |
| DBValueIsString | آیا مقدار خصوصیت در پایگاه داده هدف رشته است؟؛ البته اینکه مقدار خصوصیت در پایگاه داده هدف رشته است یا خیر از خصوصیت OriginalTypeEnum ستونها نیز قابل برداشت است امّا اینجا نیز به صورت اختصاصی ذخیره می شود. کاربردش در کدهای واسط کاربری است (cbeb78f5-f3ac-41d8-b615-c2f50657509c) |
| DBValueIsStringMiladi | آیا مقدار که رشته است میلادی است؟؛ این خصوصیت در صورت True بودن خصوصیت DBValueIsString معنی پیدا میکند. به هنگام ایجاد ستون رشته/تاریخ در صورت وجود داده در پایگاه داده هدف توسط قطعه کدی به صورت پیش فرض تشخیص داده می شود(e0a2dd6f-ec52-4430-b15f-9e272e60f7cb). البته این خصوصیت در برنامه مدیریت فراداده نیز قابل مقداردهی است؛ به هنگام استفاده از این خصوصیت در واسط کاربری، در صورت مشخص نبودن مقدار (Null بودن مقدار)، از تنظیمات پایگاه داده در برنامه مدیریت فراداده خوانده می شود. در آنجا هم مقدار اولیه False است بنابراین بصورت پیش فرض برای تمامی ستونهای تاریخ رشته مقادیر به صورت شمسی فرض می شود (5bd4e7ec-d66f-4476-a4e7-0c4c410b98b1). دقت شود که از این خصوصیت برای ذخیره سازی و یا دریافت مقدار تاریخ در پایگاه داده هدف استفاده می شود و نمایش تاریخ در واسط کاربری وابسته به همان خصوصیت ShowMiladiDateInUI می باشد. |

مثالهایی از انواع ستون تاریخ و تنظیمات مربوطه:

* ستون از نوع تاریخ میلادی در پایگاه داده و نمایش میلادی (ShowMiladiDateInUI برابر true) : ستون تاریخ بروزرسانی(UpdateDate) در جدول مورد سرویس (ServiceItem)
* ستون از نوع تاریخ میلادی در پایگاه داده و نمایش شمسی (ShowMiladiDateInUI برابر true نباشد) : خصوصیت تاریخ تولد(BirthDate) در موجودیت شخص حقیقی (RealPerson)
* ستون از نوع رشته در پایگاه داده با مقادیر شمسی و نمایش میلادی(DBValueIsStringMiladi برابر true نباشد و ShowMiladiDateInUI برابر true باشد) : خصوصیت تاریخ (PersianDate) در موجودیت درخواست سرویس (ServiceRequest)
* ستون از نوع رشته در پایگاه داده با مقادیر میلادی و نمایش شمسی (DBValueIsStringMiladi برابر true) : خصوصیت تاریخ خودکار (AutoDate) در موجودیت بررسی درخواست سرویس (ServiceRequestReview)

## ستون زمان :

برای این نوع ستون موجودیت سیستمی TimeColumnType تولید می شود. ستونهای زمان یا از نوع زمان در پایگاه داده تعریف می شوند(5a4645cf-0487-48d5-89bb-65590717ebfd) و یا به صورت رشته که خود می تواند دارای فرمت 24 ساعته، 12 ساعته میلادی و یا شمسی باشد(e0a2dd6f-ec52-4430-b15f-9e272e60f7cb). به هنگام ایجاد ستونهای جدید رشته ای، قطعه کدی زمان بودن ستون را پیش بینی می کند(caf85527-3025-4e25-8b24-50861ad0329b). در برنامه مدیریت فراداده نیز ستونهایی که از نوع رشته می باشند قابلیت تبدیل به نوع ستون زمان را دارا می باشند (559a48c0-417b-4813-8b9e-b6ad00bdd936). این ستونها مجدداً می تواند به نوع رشته بازتبدیل شوند(8351e66f-a105-44ba-8a5c-7715aa287708).

موجودیت TimeColumnType دارای خصوصیات زیر است :

|  |  |
| --- | --- |
| عنوان | توضیحات |
| DBValueIsString | آیا مقدار خصوصیت در پایگاه داده هدف رشته است؟؛ البته اینکه مقدار خصوصیت در پایگاه داده هدف رشته است یا خیر از خصوصیت OriginalTypeEnum ستونها نیز قابل برداشت است امّا اینجا نیز به صورت اختصاصی ذخیره می شود. کاربردش در کدهای واسط کاربری است (cbeb78f5-f3ac-41d8-b615-c2f50657509c) |
| DBValueStringTimeFormat | فرمت مقدار زمان در صورتی که رشته است؛ این خصوصیت در صورت True بودن خصوصیت DBValueIsString معنی پیدا میکند و مقادیر ممکن در آن "نامشخص"، "24ساعته"، "12 ساعته میلادی" و "12 ساعته شمسی" میباشد. به هنگام ایجاد ستون رشته/زمان در صورت وجود داده در پایگاه داده هدف توسط قطعه کدی به صورت پیش فرض تشخیص داده می شود(e0a2dd6f-ec52-4430-b15f-9e272e60f7cb). البته این خصوصیت در برنامه مدیریت فراداده نیز قابل مقداردهی است. به هنگام استفاده از این خصوصیت در واسط کاربری، در صورت مشخص نبودن مقدار (Null بودن مقدار)، از تنظیمات پایگاه داده در برنامه مدیریت فراداده خوانده می شود. در صورتی که آنجا نیز تعیین نشده باشد بصورت پیش فرض مقادیر به صورت "24ساعته" فرض می شود (71b3797c-8b3e-421d-8c63-8f3baefaf4fd). دقت شود که از این خصوصیت برای ذخیره سازی و یا دریافت مقدار زمان در پایگاه داده هدف استفاده می شود و نمایش زمان در واسط کاربری همواره به صورت 24 ساعته انجام می شود. |

مثالهایی از انواع ستون زمان و تنظیمات مربوطه :

* ستون از نوع زمان در پایگاه داده : ستون زمان بروزرسانی (UpdateTime) در موجودیت مورد سرویس (ServiceItem)
* ستون از نوع رشته در پایگاه داده و مقادیر 12 ساعته شمسی (DBValueStringTimeFormat برابر با AMPMShamsi) : فیلد زمان (Time) در موجودیت درخواست سرویس (ServiceRequest)
* ستون از نوع رشته در پایگاه داده و مقادیر 24 ساعته (DBValueStringTimeFormat برابر با Hours24) : خصوصیت زمان خودکار (AutoTime) در موجودیت بررسی درخواست سرویس (ServiceRequestReview)

## ستون تاریخ و زمان :

برای این نوع ستون موجودیت سیستمی DateTimeColumnType تولید می شود. ستونهای تاریخ/زمان یا به صورت ذاتاً میلادی در پایگاه داده تعریف می شوند(5a4645cf-0487-48d5-89bb-65590717ebfd) و یا به صورت رشته که خود می تواند دارای تاریخ میلادی و یا شمسی و یا زمان 12 ساعته شمسی و یا میلادی باشد(e0a2dd6f-ec52-4430-b15f-9e272e60f7cb). به هنگام ایجاد ستونهای جدید رشته ای، قطعه کدی تاریخ/زمان بودن ستون را پیش بینی می کند(caf85527-3025-4e25-8b24-50861ad0329b). در برنامه مدیریت فراداده نیز ستونهایی که از نوع رشته می باشند قابلیت تبدیل به نوع ستون تاریخ/زمان را دارا می باشند (559a48c0-417b-4813-8b9e-b6ad00bdd936). این ستونها مجدداً می تواند به نوع رشته بازتبدیل شوند(8351e66f-a105-44ba-8a5c-7715aa287708).

موجودیت DateTimeColumnType دارای خصوصیات زیر است :

|  |  |
| --- | --- |
| عنوان | توضیحات |
| ShowMiladiDateInUI | نمایش تاریخ میلادی در واسط کاربری؛ مقدار این خصوصیت در درجه اول مستقیماً به روی ستون در برنامه مدیریت فرارداده قابل تعیین شدن است، در صورت عدم مقداردهی مستقیم روی ستون (Null بودن مقدار)، از تنظیمات پایگاه داده در برنامه مدیریت فراداده خوانده می شود. در آنجا هم مقدار اولیه False است بنابراین بصورت پیش فرض برای تمامی ستونهای تاریخ در واسط کاربری تاریخ شمسی نمایش داده می شود (113afb35-1b98-401a-82dd-4bf13b0175a4). |
| DBValueIsString | آیا مقدار خصوصیت در پایگاه داده هدف رشته است؟؛ البته اینکه مقدار خصوصیت در پایگاه داده هدف رشته است یا خیر از خصوصیت OriginalTypeEnum ستونها نیز قابل برداشت است امّا اینجا نیز به صورت اختصاصی ذخیره می شود. کاربردش در کدهای واسط کاربری است (cbeb78f5-f3ac-41d8-b615-c2f50657509c) |
| DBValueIsStringMiladi | آیا مقدار که رشته است میلادی است؟؛ این خصوصیت در صورت True بودن خصوصیت DBValueIsString معنی پیدا میکند. به هنگام ایجاد ستون رشته/تاریخ و زمان در صورت وجود داده در پایگاه داده هدف توسط قطعه کدی به صورت پیش فرض تشخیص داده می شود(e0a2dd6f-ec52-4430-b15f-9e272e60f7cb). البته این خصوصیت در برنامه مدیریت فراداده نیز قابل مقداردهی است؛ به هنگام استفاده از این خصوصیت در واسط کاربری، در صورت مشخص نبودن مقدار (Null بودن مقدار)، از تنظیمات پایگاه داده در برنامه مدیریت فراداده خوانده می شود. در آنجا هم مقدار اولیه False است بنابراین بصورت پیش فرض برای تمامی ستونهای تاریخ رشته مقادیر به صورت شمسی فرض می شود (5bd4e7ec-d66f-4476-a4e7-0c4c410b98b1). دقت شود که از این خصوصیت برای ذخیره سازی و یا دریافت مقدار تاریخ در پایگاه داده هدف استفاده می شود و نمایش تاریخ در واسط کاربری وابسته به همان خصوصیت ShowMiladiDateInUI می باشد. |
| DBValueStringTimeFormat | فرمت مقدار زمان در صورتی که رشته است؛ این خصوصیت در صورت True بودن خصوصیت DBValueIsString معنی پیدا میکند و مقادیر ممکن در آن "نامشخص"، "24ساعته"، "12 ساعته میلادی" و "12 ساعته شمسی" میباشد. به هنگام ایجاد ستون رشته/تاریخ و زمان در صورت وجود داده در پایگاه داده هدف توسط قطعه کدی به صورت پیش فرض تشخیص داده می شود(e0a2dd6f-ec52-4430-b15f-9e272e60f7cb). البته این خصوصیت در برنامه مدیریت فراداده نیز قابل مقداردهی است. به هنگام استفاده از این خصوصیت در واسط کاربری، در صورت مشخص نبودن مقدار (Null بودن مقدار)، از تنظیمات پایگاه داده در برنامه مدیریت فراداده خوانده می شود. در صورتی که آنجا نیز تعیین نشده باشد بصورت پیش فرض مقادیر به صورت "24ساعته" فرض می شود (71b3797c-8b3e-421d-8c63-8f3baefaf4fd). دقت شود که از این خصوصیت برای ذخیره سازی و یا دریافت مقدار زمان در پایگاه داده هدف استفاده می شود و نمایش زمان در واسط کاربری همواره به صورت 24 ساعته انجام می شود. |

مثالهایی از انواع ستون تاریخ/زمان و تنظیمات مربوطه :

* ستون از نوع تاریخ/زمان در پایگاه داده و نمایش میلادی (ShowMiladiDateInUI برابر true) : فیلد تاریخ/زمان (DateTime) در موجودیت درخواست قطعه (RequestProductPart)
* ستون تاریخ/زمان از نوع رشته ای با مقادیر تاریخ میلادی و زمان 12 ساعته شمسی و نمایش تاریخ شمسی (DBValueIsStringMiladi برابر true و DBValueStringTimeFormat برابر با AMPMShamsi) : فیلد تاریخ/زمان رشته (StringDateTime) در موجودیت درخواست قطعه (RequestProductPart)
* ستون تاریخ/زمان از نوع رشته ای با مقادیر تاریخ شمسی و مقدار 12 ساعته میلادی و نمایش میلادی (DBValueIsStringMiladi برابر true نباشد و DBValueStringTimeFormat برابر با AMPMMiladi باشد و ShowMiladiDateInUI برابر true باشد) : فیلد تاریخ و زمان بروزرسانی (UpdateDateTime) در موجودیت خلاصه سرویس (ServiceConclusion)

## ستون Boolean:

برای این نوع ستون موجودیت سیستمی مخصوصی تولید نمی شود. تنها نکته اینست که اگر ستون Null پذیر باشد در واسط کاربری چک باکس سه حالتی (با مقادیر true، false و null) و در غیر اینصورت چک باکس دو حالتی(با مقادیر true، false) تولید می شود (01264950-cc64-4088-90ec-c5150319d9e8).

مثال چک باکس سه حالتی : فیلد تایید شده (Confirmed) در جدول آیتم خلاصه سرویس (ServiceConclusionItem)

مثال چک باکس دو حالتی : فیلد تسویه شده (IsCleared) در جدول خلاصه سرویس (ServiceConclusion)

## ستون لیست مقادیر :

برخی ستونهای فارق از اینکه از نوع رشته و یا عدد می باشند نیاز است بازه ای از مقادیر قابل انتخاب در واسط کاربری تعریف شوند. برای این ستونها موجودیتهای اختصاصی ColumnValueRange و ColumnValueRangeDetails در برنامه مدیریت فراداده و در فرم مخصوصی توسط کاربر تعریف می شوند(5389ea8a-2644-4759-9716-1abc824a1aa2). موجودیت ColumnValueRange که به صورت یک به یک با ستونهای تعریف شده در ارتباط است و ColumnValueRangeDetails نیز بصورت چند به یک با ColumnValueRange مرتبط است. لیست مقادیر در موجودیت ColumnValueRangeDetails نگهداری می شوند که در زیر به خصوصیات آن اشاره شده است:

|  |  |
| --- | --- |
| عنوان | توضیحات |
| KeyTitle | عنوان مقدار جهت نمایش در واسط کاربری. این خصوصیت اختیاری است و در صورت عدم مقداردهی همان مقادیر موجود در Value در واسط کاربری نمایش داده می شوند. |
| Value | خصوصیت اصلی است. مقداری که در پایگاه داده هدف ذخیره و بازیابی می شود. |
| Tag | کلیدی برای دسته بندی مقادیر در صورت نیاز به فیلتر کردن لیست مقادیر. |

در برنامه مدیریت فراداده می توان لیست مقادیر را در صورت وجود به صورت پیش فرض نیز از پایگاه داده هدف استخراج کرد(4efc8298-8f06-410b-bd2f-1489ef28f581).

همانطور که اشاره شد، توسط خصوصیت Tag می توان لیست مقادیر را در صورت نیاز با توجه به مقدار خصوصیت دیگری فیلتر نمود. در واقع در لیست وابسته تنها مواردی نمایش داده می شوند که خصوصیت Tag آنها برابر با مقدار فیلد مرجع تعیین شده باشد. برای مثال می توان به لیست مقادیر خصوصیت نقش تخصصی (EmployeeRole2) در موجودیت کارمند (Employee) اشاره کرد که با انتخاب فیلد مرجع نقش عمومی (EmployeeRole) فیلتر می شوند. مسیر و عنوان ستون فیلد مرجع در موجودیت ColumnValueRange نگهداری می شود که در زیر آمده است :

|  |  |
| --- | --- |
| عنوان | توضیحات |
| EntityRelationshipTailID | رشته رابطه ای ستون مرجع که لیست مقادیر با توجه به مقدار آن ستون فیلتر شوند. |
| TagColumnID | ستون مرجع که لیست مقادیر با توجه به مقدار آن ستون فیلتر شوند. |

در واسط کاربری برای مدیریت کردن عمل فیلتر لیست مقادیر از همان کلاسهای مدیریت رویداد واسط کاربری استفاده می شود (f1469379-7f8b-4bc7-b9b5-403d341eba76).

**چالش** : در صورتی که خصوصیتی در واسط کاربری دارای مقدار باشد و مقدار در لیست مقادیر(چه فیلتر شده و چه فیلتر نشده) موجود نباشد چه اتفاقی برای مقدار خصوصیت می افتد؟

# تعریف انواع رابطه

در اولین گام روابط به صورت خودکار از پایگاه داده هدف استخراج می شوند و سپس بعضی از انواع به صورت پیش فرض شناسایی می شوند. البته چه در فاز استخراج و چه در برنامه مدیریت فراداده امکان تعریف و تبدیل انواع رابطه به یکدیگر وجود دارد.

به صورت کلی هر رابطه به در مدل فراداده به صورت دو طرفه یا جفت تعریف می شود. دلیل آن سهولت در اختصاصی نمودن تعاریف خصوصیات هر سر رابطه در ورود اطلاعات موجودیت ها می باشد. انواع رابطه قابل تعریف در مدل فراداده رابطه یک به یک، رابطه یک به چند (وعکس آن چند به یک)، رابطه چند به چند، رابطه ارث بری و رابطه اتحاد می باشد. توضیح اینکه رابطه را می توان از هر طرف ورود اطلاعات کرد اما مهعمولا از یک طرف انجام می شود و و همچنین تقدم و تاخر داده ها و... قابل انتخاب بودن یا مستقیم بودن یا نبودن بستگی به ماهیت مستقل بودن یا نبودن هر یک از طرفین دارد.

برخی از خصوصیات تقریباً برای همه انواع رابطه یکسان می باشند که در زیر به آنها اشاره شده است:

|  |  |
| --- | --- |
| عنوان | توضیحات |
| Name |  |
| Alias |  |
| TypeEnum |  |
| IsOrginal |  |
| SearchInitially |  |
| IsOtherSideMandatory |  |
| IsOtherSideCreatable |  |
| IsOtherSideDirectlyCreatable |  |
| DeleteOption |  |
| IsNotSkippable |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

در ادامه به هر نوع رابطه به صورت اختصاصی تر پرداخته می شود.

## روابط یک به چند :

### ورود اطلاعات از طرف چند به یک:

این روابط عمدتاً به منظور افزودن خصوصیاتی به موجودیت طرف چند در قالب موجودیت دیگر (طرف یک) می باشد. اطلاعات طرف یک معمولاً از قبل موجود بوده و انتخاب می شود. در صورت نیاز به ورود اطلاعات طرف یک این ارتباط در لحظه، فرم مخصوص طرف به صورت جداگانه فراخوانی می شود. به عبارت دیگر جزئیات طرف یک به منظور ورود اطلاعات مستقیماً در طرف چند نمایش داده نمی شوند. البته حالت مطرح شده مدل پیش فرض می باشد و تمامی تنظیمات از طریق خصوصیات ذکر شده رابطه قابل اختصاصی شدن می باشند.

مثال ورود اطلاعات طرف یک از طرف چند بصورت غیر مستقیم : ورود اطلاعات موجودیت دفتر (Office) از طریق کارمند (Employee)

مثال ورود اطلاعات طرف یک از طرف چند بصورت مستقیم : ورود اطلاعات قلم کالا (ProductItem) از طریق درخواست سرویس (ServiceRequest)

### ورود اطلاعات از طرف یک به چند:

این روابط عمدتاً بر مبنای تعلق داشتن چندین مورد از موجودیت طرف چند به موجودیت طرف یک شکل می گیرد. اطلاعات طرف چند معمولاً به صورت دسته ای و وابسته به طرف یک ورود اطلاعات می شوند و بنابر ماهیت این نوع رابطه، ورود اطلاعات بصورت مستقیم همراه با موجودیت طرف یک می باشد.

مثال ورود اطلاعات طرف چند از طرف یک بصورت مستقیم: ورود اطلاعات مورد بررسی درخواست (ServiceRequestReviewItems) از طریق بررسی درخواست سرویس (ServiceRequestReview)

## روابط یک به یک :

این روابط معمولاً برای ارتباط دو موجودیت متفاوت که مکمل یکدیگر می باشند استفاده می شود. بنابر ماهیت پیاده سازی رابطه به هر حال یک طرف آن موجودیت اصلی که دارای کلید اصلی رابطه است قرار دارد و طرف دیگر موجودیت فرعی که از طریق کلید خارجی با موجودیت اصلی در ارتباط است. بنابراین در ورود اطلاعات این رابطه نیز به دو رابطه اصلی به فرعی و فرعی به اصلی تقسیم می گردد.

### ورود اطلاعات از موجودیت اصلی به فرعی :

طرف دیگر رابطه یا همان طرف فرعی بنابر مستقل بودن یا نبودن می تواند مستقیماً ورود اطلاعات شود و یا انتخاب شود.

مثال : ورود اطلاعات بررسی درخواست سرویس (ServiceRequestReview) از طریق درخواست سرویس(ServiceRequest )

### ورود اطلاعات از موجودیت فرعی به اصلی :

موجودیت طرف اصلی معمولاً از قبل موجود می باشد، بنابراین طرف اصلی در فرم ورود اطلاعات موجودیت اصلی صرفاً انتخاب می شود.

مثال : ورود اطلاعات درخواست سرویس(ServiceRequest ) از طریق فرم بررسی درخواست سرویس (ServiceRequestReview)

چالش : حفظ رابطه یک به یک.. آیا لازم است مثال دیگری سازمان و کارمند رئیس به دیتابیس اضافه شود؟

روابط ارث بری :

انواع :

ارث بری تجمیعی :

جداول Guarantee و BrandGuarantee و BrandProductTypeGuarantee و ProductGuarantee

ارث بری درون جدول ServiceItem که بر اساس نوع سرویس(ServiceTypeEnum) به دو نوع ServiceRepair و ServiceTest تقسیم میشود.

ServiceRepair سه رابطه از ServiceItem را با جداول ServiceItemPartImage ، ServiceItemRepair و RequestProductPart و همچنین یک ستون به نام RepairPartTotalPrice را اختصاصی می کند.

این رابطه ارث بری Overlap می باشد و با انتخاب نوع سرویس به مقدار 3 امکان اختصاصی نمودن رابطه به هر دو موجودیت ServiceTest و ServiceRepair وجود دارد.

جداول Office و Agency و Workshop

ارث بری چند جدولی :

جداول GenericPerson و RealPerson و LegalPerson

ارث بری جداول RealPerson و Employee

ارث بری جداول GenericPerson و Customer

روابط اتحاد :

جداول ServiceAdditionalItem و ServiceAdditionalItem و ServiceItem

امکان تعریف فرمول بروی ستون :

* + استفاده از کد تابع و ارتباط با خصوصیات موجودیت: محاسبه تعداد ساعت کل بروی ServiceItem با استفاده از کد فانکشن "محاسبه ساعات کار" و قراردادن در ستون HoursSpent
  + استفاده از توابع پایگاه داده : تعریف فرمول " محاسبه مبلغ " بروی ServiceItem که مبلغ را با ارسال دو فیلد HoursSpent و ServiceTypeEnum به StoredProcedure به نام sp\_CalculateServiceItemPrice محاسبه میکند.
  + استفاده از کد تابع به تنهایی:

1. محاسبه تاریخ روز و قرار دادن در ستون UpdateDate موجودیت ServiceItem با استفاده از کد تابع GetNow
2. محاسبه تاریخ روز و قرار دادن در ستون UpdateTime موجودیت ServiceItem با استفاده از کد تابع GetNow
   * استفاده از فرمول خطی :
3. تعریف فرمول خطی به نام " محاسبه هزينه" بروی ServiceConclusionItem که مبلغ را ServiceItem یا ServiceAdditionalItem می خواند. نتیجه این فرمول در ستون Price قرار میگیرد.
4. تعریف فرمول خطی به نام "جمع موارد" بروی ServiceConclusion که مبالغ ServiceConclusionItem مرتبط را جمع میکند و نهایتاً در ستون TotalPrice قرار میدهد.

* محاسبه فرمول بصورت پیش فرض :

محاسبه تاریخ روز شمسی و قرار دادن در ستون PersianDate موجودیت ServiceRequest با استفاده فرمول خطی. همچنین خصوصیت CalculateFormulaAsDefault این ستون باید مقدار True قرار داده شود.

محاسبه شناسه پست سازمانی و شناسه دفتر در در خواست سرویس بصورت پیش فرض

امکان استفاده از StoredProcedure :

امکان استفاده از DB Function :

امکان تعریف نماهای مختلف داده ListView:

لیست نمایش داده برای خلاصه سرویس

امکان تعریف نماهای جستجوی داده : SearchView:

لیست نمایش داده برای خلاصه سرویس

انواع :

به حالت Value/Text : فیلد EmployeeRole در جدول Employee ، در این حالت مقادیر از ستون EmployeeRole به حالت عددی خوانده میشوند اما در فرم مقادیر Text معادل نمایش داده میشوند.

به حالت Value تنها: فیلد WorkshopLevel در جدول Office، در این حالت لیست مقادیر از ستون WorkshopLevel عیناً خوانده و نمایش داده میشوند. حال میتواند این ستون عددی یا رشته ای باشد.

انواع و کاربردها

امکان استفاده از فیلد محاسباتی پایگاه داده : فیلد Name در موجودیت GenericPerson

امکان تعریف وضعیتها:

مقداردهی پیش فرض خصوصیات: خصوصیت کد در سازمان : که اگر isAgency مقدار true داشته باشد مقدار کد برابر 10 و اگر isWorkshop تیک بخورد مقدار پیش فرض کد 20 می شود.

فقط خواندنی سازی موجودیت : موجودیت شهر/استان که اگر نوع داده لود شده کشور باشد فرم کلاً فقط خواندنی می شود.( خود کمبو نوع نمی شود. بررسی شود)

غیرفعال سازی خصوصیت: مخفی کردن خصوصیت WorkshopLevel در واحد اداری وقتی تیک آژانس انتخاب می شود.

فقط خواندنی سازی خصوصیت: خصوصیت نوع در موارد خلاصه سرویس که اگر IsConfimed برابر با true باشد این خصوصیت فقط خواندنی می شود

غیرفعال سازی رابطه: مخفی کردن رابطه پدر در شهر/استان وقتی نوع کشور انتخاب می شود.

فقط خواندنی سازی رابطه: در موارد خلاصه سرویس اگر خصوصیت IsConfimed انتخاب شود روابط ServiceItem و ServiceAdditionalItem فقط خواندنی می شوند

بازه داده برای خصوصیت : فیلتر نقش 2 کارمند بر اساس نقش 1 در فرم کارمند

امکان تعریف BackednAction :

امکان اعمال کد تابع عمومی قبل از لود شدن داده موجود :

عمومی : کد تابع EditPersianDateMonthDay که تاریخهای شمسی که ماه و یا روز تک رقمی دارند را به ماه و روز دو رقمی اصلاح می کند.

موجودیت اختصاصی: کد تابع "اصلاح کد ملی" به روی موجودیت شخص حقیقی که در صورت نبودن دو صفر اول کد ملی به آنها دو صفر را اضافه می کند.

امکان اعمال کد تابع عمومی قبل از بروزرسانی شدن داده:

اختصاصی : کد تابع حذف عبارت "شرکت" از نام اشخاص حقوقی (در حال حاضر در صورتی که این فیلد تغییری کرده باشد اعمال می شود، بررسی شود که منطقی است یا خیر)

کد تابع عمومی : کد تابع عمومی "اعتبارسنجی درخواست" که پیغام خطای کاربر غیر مجاز می دهد در صورتی که شناسه درخواست دهنده برابر با عدد 666 باشد

امکان اعمال کد تابع عمومی قبل از حذف داده:

اختصاصی : کد تابع "اعتبارسنجی حذف درخواست سرویس" که به هنگام حذف درخواست سرویس اگر خصوصیت userId با شناسه کاربر جاری یکسان نباشد خطا می دهد

کد تابع عمومی : کد تابع عمومی "اعتبارسنجی درخواست حذف" که پیغام خطای کاربر غیر مجاز می دهد در صورتی که شناسه درخواست دهنده برابر با عدد 666 باشد.

امکان اعمال کد تابع عمومی بعد از بروزرسانی داده :

اختصاصی : تابع دیتابیس sp\_CalculateCustomerValue که ارزش مشتری را با گرفتن پارامتر شناسه از ServiceConclusion محاسبه می کند

امکان اعمال کد تابع عمومی بعد از حذف داده :

اختصاصی : تابع دیتابیس sp\_CalculateCustomerValueByServiceRequestID که ارزش مشتری را با گرفتن پارامتر شناسه درخواست از ServiceConclusion محاسبه می کند. به این علت با بالایی تفاوت دارد زیرا بعد از حذف دیگر شناسه اصلی وجود ندارد که از تابع بالا استفاده شود.

امکان فیلتر کردن روابط :

فیلتر رابطه نوع سرویس در موجودیت مورد بررسی درخواست (ServiceRequestReviewItems) که بر اساس درخواست سرویس فیلتر می شود (انواع درخواست سرویس تعریف شده در درخواست سرویس مرتبط)

فیلتر رابطه ServiceItemRepair با ServiceActionType که بر اساس نوع سرویس از ServiceItem و نوع محصول از درخواست سرویس فیلتر می شود.

امکان تعریف رابطه : ایجاد رابطه یک به چند بین Product از دیتابیس DBProducts و ProductItem از دیتابیس DBProductService

امکان تعریف انواع اعتبارسنجی ها :

امکان فقط خواندنی نمودن موجودیتها :

امکان تعریف انواع اعتبارسنجی به روی ستونها :

امکان تعریف لیست مقادیر :

لیست مقادیر نقش کارمند در فرم کارمند

امکان فقط خواندنی نمودن ستونها : فقط خواندنی نمودن خصوصیت PersianDate در ServiceRequest

امکان ورود اطلاعات اجباری ستونها : اجباری نمودن ستون Name در شخص حقوقی

اعتبارسنجی ستونها بر حسب نوع:

ستونهای رشته ای :

امکان تعریف حداکثر طول رشته: فیلد کد در جدول Office

امکان تعریف حداقل طول رشته: فیلد کد در جدول Office

امکان تعریف فرمت: با تعریف فرمت میتوان نسبت به اعتبارسنجی ستونهای رشته ای به هنگام ثبت فرم اقدام کرد.

مثال : فیلد EmailAddress در جدول شخص عمومی (باید فرمت با مقدار ^(?!\.)(""([^""\r\\]|\\[""\r\\])\*""|([-a-z0-9!#$%&'\*+/=?^\_`{|}~]|(?<!\.)\.)\*)(?<!\.)@[a-z0-9][\w\.-]\*[a-z0-9]\.[a-z][a-z\.]\*[a-z]$ برای این ستون ست شود)

ستونهای عددی :

امکان تعریف تعداد اعشار و عدد صحیح : فیلد UserRate در خلاصه سرویس که یک رقم صحیح و یک رقم اعشار دارد

امکان تعریف بیشینه و کمینه : فیلد UserRate در خلاصه سرویس که مقدار بیشینه آن 5 و کمینه 1 می باشد

ستون تاریخ :

امکان بزگتر کو چکتر تاریخ روز و ...

اعتبارسنجی روابط :

فقط خواندنی نمودن رابطه : ؟؟

اجباری نمودن رابطه : اجباری نمودن رابطه با شهرها در GenericPersonAddress

اعتبارسنجی روابط بر حسب نوع:

روابط ارث بری :

ارث بری Disjoint باشد : رابطه ارث بری بین GenericPerson و LegalPerson و RealPerson

ارث بری Disjoint نباشد : ؟؟؟

ارث بری TolatParticipation باشد : رابطه ارث بری بین GenericPerson و LegalPerson و RealPerson

ارث بری TolatParticipation نباشد : رابطه ارث بری بین GenericPerson و Customer

روابط اتحاد :

اتحاد TolatParticipation باشد : ؟؟؟

اتحاد TolatParticipation نباشد : رابطه ارث بری بین ServiceConclusionItem و ServiceItem و ServiceAdditionalItem

اعتبارسنجی اختصاصی موجودیتها :

امکان دو نوع اعتبار سنجی وجود دارد :

1. اعتبارسنجی اختصاصی بر اساس فرمول اعتبارسنجی : فرمول اعتبارسنجی برای درخواست سرویس که با استفاده از کد تابع اعتبارسنجی درخواست سرویس اجازه ثبت درخواست برای کالا با عنوان برند "Hisenese" را نمی دهد.
2. کد تابع قبل از بروز رسانی : به " امکان اعمال کد تابع عمومی قبل از بروزرسانی شدن داده" مراجعه شود.

امکان تعریف دکمه های Command :

امکان تعریف DataView و GridView :

جستجوی عمومی نمای گرافیکی داده (DataView) : منوی عمومی جستجو و نمایش گرافیکی داده که امکان انتخاب نوع موجودیت و سپس جستجوی پیش فرض را بروی آن موجودیت می دهد.

جستجوی عمومی نمای جدولی داده (GridView) : منوی عمومی جستجو و نمایش جدولی داده که امکان انتخاب نوع موجودیت و سپس جستجوی پیش فرض را بروی آن موجودیت می دهد.

امکان تعریف نمای داده موجودیت در منوی برنامه : برای هر موجودیت میتوان نمای داده آن را (با تنظیمات منوی داده پیش فرض) در منو تعریف نمود. نمونه : نمای داده مشتری در زیر منوی نمای داده.

امکان تعریف گرید داده موجودیت در منوی برنامه : برای هر موجودیت میتوان گرید داده آن را (با تنظیمات منوی داده پیش فرض) در منو تعریف نمود. نمونه : گرید داده مشتری در زیر منوی نمای داده.

امکان تعریف گزارش نمای داده : به بخش گزارشات مراجعه شود.

امکان تعریف گزارش گرید داده : به بخش گزارشات مراجعه شود.

تنظیمات منوی داده :

امکان تعریف رشته روابط گرید داده یا نمای داده و اختصاصی سازی منوی داده آنها : برای موجودیت مشتری نمای داده رشته رابطه با خلاصه درخواست سرویس (از طریق درخواست سرویس و سپس خلاصه سرویس) تعریف شده است. تنظیمات منوی خلاصه سرویس نیز برای این رابطه تعیین شده است که خود دارای گرید داده رابطه یا موارد خلاصه سرویس می باشد.

امکان تعریف رشته روابط گزارشات : برای موجودیت مشتری گزارشاتی بر اساس رشته رابطه با خلاصه درخواست سرویس (از طریق درخواست سرویس و سپس خلاصه سرویس) تعریف شده است. رابطه با گزارش لیست صورتحساب، گزارش دایره ای صورتحساب و گزارش نمای داده صورتحساب

دسترسی به منوی داده : چه در نمای داده، گرید داده و یا قلم اطلاعاتی داده چه در فرم ورود اطلاعات و چه کارتابل و .. می توان به منوی داده دسترسی داشت. سپس تنظیمات داده اختصاصی و یا پیش فرض به روی منوی داده اعمال می شود.

امکان استفاده از View :

نمای view\_ServieRequests در سامانه بروزرسانی شده است و گزارش چارت ساعات سرویس هر برند برای این نما تعریف گردیده است. همچنین ارتباطی بین این نما و ServiceRequest تعریف شده و از خصوصیت HoursSpent این نما در ستونهای نمایش پیش فرض ServiceRequest استفاده شده است. همچنین تنظیمات منوی داده برای این نما به تنظیمات منوی داده درخواست سرویس ارجاع شده است.

امکان تعریف انواع گزارشات:

انواع گزارشات :

گزارشات خارجی مستقیم به روی قلم داده :

گزارشات جستجو محور :

امکان تعریف جستجوی پیش فرض : ساختار SearchRepository

برای این دسته از گزارشات علاوه بر خصوصیات و قابلیتهای اختصاصی هر نوع گزارش، امکان تعریف جستجوهای پیش فرض نیز به منظور فیلتر نمودن بازه داده ها مقدور است.

گزارش نمای داده : گزارش نمای داده صورتحساب در بخش گزارشات. در این گزارش از جستجوی پیش فرض به نام "جستجوی صورتحساب از ابتدای سال" استفاده شده است. همچنین تنظیمات منو همان "تنظیمات منوی صورتحساب و جزئیات" می باشد.

گزارش گرید داده : گزارش گرید داده صورتحساب در بخش گزارشات. در این گزارش از جستجوی پیش فرض به نام "جستجوی صورتحساب از ابتدای سال" استفاده شده است. همچنین تنظیمات منو همان "تنظیمات منوی صورتحساب و جزئیات" می باشد.

گزارش چاپی لیستی ساده : گزارش چاپی لیست صورتحساب ساده در بخش گزارشات. در این گزارش از جستجوی پیش فرض به نام "جستجوی صورتحساب از ابتدای سال" استفاده شده است. همچنین لیست نمایش به نام "لیست پیش فرض خلاصه سرویس" می باشد.

گزارش چاپی لیستی با گروهبندی : در بخش گزارشات به نام "گزارش چاپی لیست صورتحساب گروهبندی" می باشد. در این گزارش گروهبندی بر اساس ستون نمایشی عنوان برند می باشد.

گزارش چاپی لیستی با زیر گزارش : در بخش گزارشات به نام "گزارش چاپی لیست صورتحساب زیر گزارش" می باشد. زیر گزارش مرتبط نیز همان موارد صورتحساب به نام "گزارش چاپی لیست موارد صورتحساب" است که ارتباط گزارش و زیر گزارش از طریق ستون “ID” از سمت گزارش و ستون " ServiceConclusionID" از سمت زیر گزارش می باشد.

گزارش چارت ستونی : گزارش با عنوان "گزارش چارت ستونی صورتحساب" که دسته بندی اول آن بر اساس نام برند و دسته بندی دوم بر اساس نامه دفتر می باشد. به این ترتیب بر اساس این دو دسته نمودار ستونی تعداد صورتحسابها نمایش داده می شوند.

گزارش چارت دایره ای : گزارش با عنوان "گزارش چارت دایره ای صورتحساب" که دسته بندی آن بر اساس نام برند می باشد که بر اساس آن نمودار دایره ای تعداد صورتحسابها نمایش داده می شوند.

گزارش چارت خطی : گزارش با عنوان "گزارش چارت خطی صورتحساب" که دسته بندی اول آن بر اساس نام برند و دسته بندی دوم بر اساس نامه دفتر می باشد. به این ترتیب بر اساس این دو دسته نمودار خطی تعداد صورتحسابها نمایش داده می شوند.

گزارش چارت راداری : گزارش با عنوان "گزارش چارت راداری صورتحساب" که دسته بندی اول آن بر اساس نام برند و دسته بندی دوم بر اساس نامه دفتر می باشد. به این ترتیب بر اساس این دو دسته نمودار راداری تعداد صورتحسابها نمایش داده می شوند.

گزارش کراس تب : گزارش با عنوان "گزارش کراس تب صورتحساب" که ستونهای آن بر اساس نام برند و مدل محصول و ردیفهای آن بر اساس نامه دفتر می باشد. به این ترتیب بر اساس این ستونها و ردیفها جدول کراس تب تعداد صورتحسابها نمایش داده می شوند.

گزارشات جستجویی خارجی : با ثبت تنظیمات گزارش خارجی برای هر موجودیت، یک جدول با پیشوند "xr\_" در پایگاه داده استقرار موجودیت ایجاد می شود. ستونهای این جدول کلیدهای اصلی موجودیت و یک ستون به نام ReportKey از نوع int می باشد.

بعد از هر بار جستجوی داده ها در فرم گزارش، کلیدهای اصلی داده ها به همراه یک شناسه ایجاد شده ReportKey در این جدول قرار می گیرند و سپس این ReportKey به عنوان پارامتر به آدرس تنظیم شده گزارش ارسال می شود. حال کد پشت گزارش میتواند برای بدست آوردن کلیدهای رکوردهای جستجو شده از این ReportKey استفاده نموده و داده ها را فیلتر نماید. برای مثال گزارشی به آدرس "<http://dolatkiam/ReportServer/Pages/ReportViewer.aspx?%2fRequestConclusionsByReportKey>" در ریپورتینگ سرویس برای صورتحسابها تعریف شده است. این گزارش از یک SP استفاده می کند که جدول ServiceConclusion را با xr\_ServiceConclusion جوین زده و از پارامتر ReportKey برای فیلتر کردن داده ها استفاده می کند. SP استفاده شده به شکل زیر می باشد.

USE [DBProductService]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[Report\_ServiceRequestConclusions] Script Date: 7/27/2021 7:00:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[Report\_ServiceRequestConclusions]

-- Add the parameters for the stored procedure here

@ReportKey bigint

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

--EXEC sp\_CreateReportTable 'Report\_City',@ReportKey

SELECT \* from Accounting.ServiceConclusion

inner join dbo.xr\_ServiceConclusion as filter on Accounting.ServiceConclusion.ID=filter.id where filter.ReportKey=@ReportKey

END

بنابر این برای تعریف گزارشات خارجی بهتر است 3 مرحله زیر طی شود:

1. گزارش خارجی بصورت کلی و بدون استفاده از جدول واسط xr طراحی شود.
2. تنظیمات گزارش خارجی در برنامه معرفی شود. به این ترتیب جدول واسط xr نیز ساخته می شود.
3. گزارش خارجی اصلاح شود و در Query پشت آن جدول xr و پارامتر ReportKey نیز به منظور فیلتر داده ها اعمال شوند.

گزارشات مستقیم :

بروی مشتری گزارش مستقیمی تعریف شده که با استفاده از پارامترهای تعریف شده در تنظیمات آن، گزارش خارجی را به آدرس <http://dolatkiam/ReportServer/Pages/ReportViewer.aspx?%2fRequestConclusionsByCustomerID> فراخوانی می کند. این گزارش هم در منوی برنامه و هم در تنظیمات منوی مشتری قابل دسترسی است.

امکان تعریف DataLink :

تعریف گزارش لینک داده بنام گزارش لینک ریپورت که بین Region از دیتابیس خدمات و BrandProductType از دیتابیس محصولات که بر اساس رشته رابطه شهر/استان (Region )، آدرس شخص، شخص، مشتری، درخواست سرویس، قلم کالا، محصول و برند/نوع محصول (BrandProductType) می باشد. در این رشته رابطه برای موجودیتهای مشتری و درخواست سرویس از تنظیمات منوی اختصاصی استفاده شده است.

امکان تعریف گراف داده :

تعریف گزارش گراف داده بنام" گزارش گراف" که از مشتری شروع شده و رابطه با درخواست سرویس و صورتحساب شرویس را نمایش می دهد. در این رشته رابطه برای موجودیتهای مشتری و درخواست سرویس از تنظیمات منوی اختصاصی استفاده شده است.

امکان آرشیو :

امکان تعریف پوشه اختصاصی برای نوع موجودیت : پوشه ضمانتنامه برای درخواست سرویس

امکان تعریف پوشه های عمومی: تعریف پوشه مدارک عمومی

امکان تعریف تگ های اختصاصی برای نوع موجودیت : تگ مدرک اصلی برای درخواست سرویس

امکان تعریف پوشه های عمومی: تعریف تگ اصل و کپی به صورت عمومی

امکان تعریف نمایش آرشیو داده مرتبط : تعریف رابطه آرشیو بررسی درخواست سرویس به روی موجودیت درخواست سرویس و بالعکس

نمایش خودکار آرشیو برای داده های مرتبط : این خصوصیت برای بررسی درخواست سرویس فعال است و موارد مرتبط بصورت خودکار نمایش داده می شوند.

دسترسی آرشیو : کاربر کوثری اجازه افزودن یا حذف موارد آرشیو را روی درخواست سرویس ندارد اما روی بررسی درخواست سرویس دارد.

امکان نامه ها:

امکان تعریف نمایش نامه های داده مرتبط : تعریف رابطه نامه بررسی درخواست سرویس به روی موجودیت درخواست سرویس و بالعکس

نمایش خودکار نامه ها برای داده های مرتبط : این خصوصیت برای بررسی درخواست سرویس فعال است و موارد مرتبط بصورت خودکار نمایش داده می شوند.

تعریف نوع نامه عمومی : نوع نامه "نوع نامشخص" به صورت عمومی

تعریف نوع نامه اختصاصی : نوع نامه "درخواست سرویس" برای موجودیت درخواست سرویس

دسترسی نامه : کاربر کوثری اجازه افزودن یا حذف موارد نامه را روی درخواست سرویس ندارد اما روی بررسی درخواست سرویس دارد.

انواع ایجاد نامه :

نامه داخلی با انتخاب فایل : ایجاد نامه با عنوان "نامه داخلی موجود " برای اولین داده درخواست سرویس، فایل مرتبط با این نامه از آدرس D:\Files\ServiceRequestInternalLetter.docx بارگذاری می شود.

نامه داخلی تولید فایل با فیلدهای ساده : ایجاد نامه با عنوان " نامه توليد شده نمونه" برای اولین داده درخواست سرویس، قالب مرتبط با این نامه از آدرس D:\Files\ ServiceRequestLetterTemplate.docx بارگذاری می شود.

نامه داخلی تولید فایل با فیلدهای رابطه ای : : ایجاد نامه با عنوان "نامه توليد شده پيچيده" برای اولین داده بررسی درخواست سرویس، قالب مرتبط با این نامه از آدرس D:\Files\ ServiceRequestReviewLetterTemplate.docx بارگذاری می شود. در این قالب در فیلد رابطه ای Rel\_SrvReqType\_Start تمامی انواع درخواست مربوط به درخواست سرویس مرتبط با بررسی درخواست سرویس نمایش داده می شود. سپس در فیلد رابطه ای Rel\_SrvReqRevItem\_Start موارد خود بررسی درخواست سرویس نمایش داده می شود. درون این رابطه از رابطه دیگری که ارتباط موارد را با نوع درخواست تعیین میکند به نام Rel\_ReqType\_Start استفاده شده است.

ایجاد نامه با استفاده از منبع خارجی : در تنظیمات نامه کد منبع خارجی مشخص می شود و به این ترتیب می توان نامه با استفاده از منبع خارجی ایجاد کرد. برای اولین داده درخواست سرویس تولید می شود.

تولید و تکرار رابطه در جدول را هم داریم، بعداً تست شود.

تنظیمات نامه :

کد فانکشن قبل از لود نامه : فانکشن BeforeLoad از MyTestImplLibrary

کد فانکشن قبل از ذخیره نامه : فانکشن BeforeSave از MyTestImplLibrary

کد فانکشن بعد از ذخیره نامه : فانکشن AfterSave از MyTestImplLibrary

کد فانکشن دریافت اطلاعات نامه از منبع خارجی : فانکشن ExternalCode از MyTestImplLibrary

کد فانکشن ارسال نامه داخلی به خارجی : فانکشن ConvertToExternal از MyTestImplLibrary

دسترسی داده مستقیم:

دسترسی داده در دو نوع انجام می شود. دسترسی دریافت/نمایش اطلاعات و دسترسی فقط خواندنی نمودن اطلاعات.

برای موجودیت درخواست هزینه دسترسی داده در دو حالت تعریف شده است. اینکه کاربر راهبر کل باشد و یا شناسه دفتر برابر با شناسه سازمان خارجی کاربر باشد.

دسترسی اصلاح داده: برای موجودیت درخواست هزینه دسترسی اصلاح داده در دو حالت تعریف شده است. اینکه کاربر راهبر کل باشد و یا شناسه پست سازمانی درخواست برابر با شناسه پست سازمانی کاربر باشد.(یا به عبارتی خود کاربر درخواست را ایجاد کرده باشد)

دسترسی داده غیر مستقیم: برای موجودیت خلاصه سرویس دسترسی داده از دخواست سرویس مربوطه خوانده و اعمال می شود.

لاگ داده :

تعریف جریان کار:

امکان تعریف وضعیت ها در یک جریان کار : برای جریان کار "درخواست سرويس" 7 وضعیت نمونه تعریف شده است. امکان تعریف فرمول برای هر وضعیت : برای هر وضعیت فرمول هایی می توان تعریف نمود که در صورت تخقق فرمول برای داده مرتبط امکان ایجاد چنین وضعیتی در جریان کار فراهم باشد.

امکان تعریف فعالیت برای هر وضعیت : برای هر وضعیت می توان لیست فعالیتهایی تعریف نمود که در صورت محقق شدن وضعیت آن فعالیتهای انجام شوند.

|  |  |  |
| --- | --- | --- |
| عنوان وضعیت | نوع وضعیت | فرمول |
| ثبت درخواست | وضعیت شروع | HasReview برقرار نباشد |
| بررسي درخواست | وضعیت در جریان |  |
| در حال سرويس مي باشد | وضعیت در جریان |  |
| در حال صدور صورتحساب مي باشد | وضعیت در جریان |  |
| صورتحساب تسويه شد و اتمام کار | وضعیت پایان | فرمول Cleared برقرار باشد |
| درخواست لغو شد و اتمام کار | وضعیت پایان |  |
| قابل تعمير نمي باشد و اتمام | وضعیت پایان |  |

امکان تعریف تبدیل های بین وضعیت و اقدامات و فعالیتهای مرتبط : برای هر انتقال بین وضعیت مجموعه ای از مفاهیم را میتوان تعریف نمود که در زیر به آنها اشاره شده است.

اقدامات : اقدامات مجموعه وظایفی هستند که در صورت انجام شدن عملیات انتقال وضعیت انجام می شود. هر اقدام خصوصیات زیر را دارد :

کاربران هدف : مجموع ای از نقشها که توانایی انجام آن اقدام را در جریان کار محوله دارند.

فرمول : فرمولهایی که در صورت محقق بودن یا نبودن امکان اعتبارسنجی انجام شده بودن اقدام را می دهد.

گروه فرمها : مجموعه ای قابل تعریف از فرمها که در محدوده کل جریان کار تعریف می شود و با متصل کردن آنها به هر اقدام امکان ورود اطلاعات آن فرمها را همراه با اقدامات متناسب در کارتابل به کاربر می دهد.

امکان انجام اقدام توسط چند کاربر : به هنگام ارجاع کار می توان همزمان چند کاربر هدف را جهت انجام اقدام مشخص نمود.

فعالیتای هر انتقال : برای هر انتقال می توان لیست فعالیتهایی تعریف نمود که در صورت محقق شدن آن انتقال وضعیت فعالیتهای انجام شوند.

امکان تعریف وضعیتهای وتبدیلها و اقدامات مرتبط به صورت نموداری : تمامی وضعیتها، تبدیلها و اقدامات مرتبط را می توان به صورت نموداری تعریف نمود.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| از وضعیت | به وضعیت | نام تبدیل | اقدامات مرتبط | فرمول اقدامات | نقشهای اقدامات | گروه فرمها |
| ثبت درخواست | بررسي درخواست | ارسال به بررسي درخواست | تایید ثبت درخواست |  | راهبر اداره  راهبر کل  اپراتور ثبت درخواست | ثبت درخواست شامل فرم ثبت درخواست |
| بررسي درخواست | در حال سرويس مي باشد | ارسال به تعميرات | تایید ارسال به تعميرات | HasReview برقرار باشد | راهبر اداره  راهبر کل  بررسی کننده درخواست | بررسی درخواست شامل فرم بررسی درخواست |
| در حال سرويس مي باشد | در حال صدور صورتحساب مي باشد | ارسال به صورتحساب | تاييد و ارسال به صورتحساب |  | راهبر اداره  راهبر کل  تعمیرکار | ثبت تعميرات شامل فرم مورد سرویس |
| در حال صدور صورتحساب مي باشد | صورتحساب تسويه شد و اتمام کار | ارسال به تسويه و اتمام | اتمام جريان کار |  | راهبر اداره  راهبر کل  مسئول مالی سرویس | ثبت اطلاعات صورتحساب شامل فرم خلاصه سرویس |
| بررسي درخواست | ثبت درخواست | اطلاعات ثبتي ناقص مي باشد | بازگشت به مرحله ثبت سرويس |  | راهبر اداره  راهبر کل  بررسی کننده درخواست | بررسی درخواست شامل فرم بررسی درخواست |
| در حال سرويس مي باشد | بررسي درخواست | قابل سرويس نمي باشد | بازگشت به بررسي سرويس |  | راهبر اداره  راهبر کل  تعمیرکار |  |
| بررسي درخواست | درخواست لغو شد و اتمام کار | درخواست قابل اجرا نمي باشد | درخواست قابل اجرا نمي باشد |  | راهبر اداره  راهبر کل  بررسی کننده درخواست |  |

کارتابل و تعریف جریان کار و ارجاع جریان کار:

توضیحات بعدا اضافه شود

تعریف کاربران و سازمانها :

امکان تعریف نوع سازمان : نوع سازمان تعمیرگاه، دفتر خدمات

امکان تعریف سازمانها : تعریف مرکز ونک و تعریف تعمیرگاه تجریش

امکان تعریف نوع نقش : نقش اپراتور ثبت درخواست، تعمیرکار، بررسی کننده درخواست، راهبر اداره، مسئول مالی سرویس و راهبر کل

امکان تعریف ارتباط بین نوع سازمان و نوع نقش :

نقشهای نوع سازمان تعمیرگاه شامل : تعمیرکار و راهبر اداره

تقشهای دفتر خدمات شامل: اپراتور ثبت درخواست، بررسی کننده درخواست، راهبر اداره، مسئول مالی سرویس ، راهبر اداره و راهبر کل

امکان تعریف پست سازمانی و کاربران مرتبط :

|  |  |  |
| --- | --- | --- |
| نوع نقش و سازمان | نام پست | کاربر مرتبط |
| نقش تعمیرکار در سازمان تعمیرگاه تجریش | تعمیرکار 1 | Rezayi |
| نقش تعمیرکار در سازمان تعمیرگاه تجریش | تعمیرکار 2 | Zahed |
| نقش تعمیرکار در سازمان تعمیرگاه تجریش | تعمیرکار 3 | Pirazad |
| نقش راهبر اداره در سازمان تعمیرگاه تجریش | راهبر 1 | Rezayi |
| نقش اپراتور ثبت درخواست در مرکز ونک | اپراتور 1 | Dolatkhah |
| نقش اپراتور ثبت درخواست در مرکز ونک | اپراتور 2 | Movaseghi |
| نقش بررسی کننده درخواست در مرکز ونک | بررسی کننده سرویس 1 | Kosari |
| نقش راهبر اداره در مرکز ونک | راهبر 1 | Afshari |
| نقش مسئول مالی سرویس در مرکز ونک | مسئول مالی سرویس 1 | Karimi |
| نقش راهبر کل در مرکز ونک | راهبر کل 1 | Dolatkia |

امکان تعریف دسترسی ها :

امکان تعریف دسترسی برای کاربران در سطوح زیر :

دسترسی نوع سازمان، دسترسی نوع نقش، دسترسی ارتباط نقش و نوع سازمان، دسترسی سازمان و دسترسی پست سازمانی

امکان تعریف دسترسی به روی منابع زیر :

دیتابیس، شمای دیتابیس، موجودیت، رابطه، ستون، گزارش، دکمه دستوری، زیر سیستم

امکان تعریف انواع دسترسی : عدم دسترسی، فقط خواندنی، اصلاح، اصلاح و حذف

درخت منابع و نوع دسترسی.

بنابراین دسترسی به منابع را میتوان از بین هر کدام از سطوح کاربری و نوع منبع ذکر شده انتخاب کرد. سعی می شود در مثالهای زیر هر کدام از سطوح و یا منابع ذکر شده بکار گرفته شوند :

دسترسی نوع سازمان :

نوع سازمان تعمیرگاه به کل دیتابیس دسترسی فقط خواندنی، نمایش آرشیو و نمایش نامه دارد.

نوع سازمان دفتر خدمات به کل دیتابیس دسترسی فقط خواندنی، نمایش آرشیو و نمایش نامه دارد.

دسترسی نوع نقش :

نقش تعمیرکار به شمای تعمیرگاه دسترسی اصلاح، حذف و اصلاح آرشیو و اصلاح نامه دارد.

نقش اپراتور ثبت درخواست: به موجودیتهای درخواست سرویس، انواع درخواست سرویس، مشتری، شخص، شخص حقیقی، شخص حقوقی ،آدرس و محصول دسترسی اصلاح ، حذف و اصلاح آرشیو و اصلاح نامه دارد.

نقش بررسی کننده درخواست سرویس به موجودیتهای بررسی درخواست سرویس و مورد درخواست سرویس دسترسی اصلاح ، حذف و اصلاح آرشیو و اصلاح نامه دارد.

نقش مسئول مالی به شمای Accounting دسترسی اصلاح ، حذف و اصلاح آرشیو و اصلاح نامه دارد.

نقش راهبر اداره به شمای Common دسترسی اصلاح ، حذف و اصلاح آرشیو و اصلاح نامه دارد. این نقش به موجودیت شهر/استان (که در شمای Common می باشد) دسترسی فقط خواندنی دارد.

نقش راهبر کل به کل پایگاه داده دسترسی اصلاح و حذف دارد.

دسترسی نوع نقش نوع سازمان :

نقش راهبر اداره در نوع سازمان تعمیرگاه به موجودیتهای نوع سرویس و قلم کالا دسترسی فقط خواندنی دارد. همچنین نقش راهبر اداره در نوع سازمان تعمیرگاه به ستون کدمشتری در جدول مشتری دسترسی ندارد.

دسترسی سازمان :

به نظرم مفهوم دسترسی سازمان حذف شود