

راهنمای جداول اطلاعاتی بودجه بان - ویرایش ۲ - ۱۳۹۲/۸/۱۴

۱. بانک

کد بانک

عنوان بانک

این سامانه دارای چهار بانک اطلاعاتی است که نام آنها در این جدول ذخیره می شود و هر کدام یک کد دارند که در جدول رکورد و جدول فرم مورد استفاده قرار می گیرند. فیلدهای «انتخاب از بانک دیگر» نیز به همین کد ارجاع می دهند. می توانستیم حالا که نام بانکها را می دانیم و تعدادشان محدود است، از جدول استفاده نکنیم، در این صورت امکان افزایش بانکها را در آینده نداشتیم. این احتمال وجود دارد که در توسعه های بعدی نرم افزار نیاز به افزایش بانکها پیش آید.

۲. برنامه - سطح

کد سطح

عنوان سطح

کد سطح پدر

استثناً در بانک برنامه ها نیاز به سطح بندی داریم. سطح بندی این معنا را دارد که عملاً مدیر سامانه این اختیار را دارد که به صورت مجازی چند بانک پله ای ایجاد نماید که هر کدام وابسته و تحت دیگری است. طبیعتاً هر کدام از این بانکهای مجازی (همان سطحها) فرمهای مختص به خود را خواهند داشت. اما در محاسبات مالی، ارقام بودجه از هر پله به پله بالاتر رفته و جمع خواهد شد. این جدول هم اسامی سطحها را نگهداری می نماید و هم ترتیب پله ای آنها را با استفاده از کد سطح پدر (Parent-Child). نکته: هر رکورد از سطح پایین لزوماً باید به یک رکورد از سطح بالاتر خود متصل باشد. پس نمی توان رکورد جدید به بانک برنامه اضافه کرد، مگر این که حتماً به رکوردی از سطح بالاتر وصل شده باشد

۳. فرم

کد فرم

کد بانک

کد سطح

عنوان فرم

فرم اصلی

هر تعداد فرم می توان در ذیل یک بانک داشت. اکنون که سطحهایی در بانک برنامه داریم ناگزیر هستیم که کد

سطح را هم به فرم ضمیمه نماییم. طبیعتاً این کد معمولاً خالی خواهد بود، فقط هنگامی که کد بانک ۲ باشد (مثلاً کد بانک برنامه) ما باید کد سطح را هم چک کنیم و سپس فرم‌ها را نمایش دهیم. بنابراین هر سطح از بانک برنامه فرم‌های مختص به خود را خواهد داشت. وقتی یک کاربر روی بانک کلیک می‌نماید، ما محتوای فرم اصلی را نمایش می‌دهیم و لینک سایر فرم‌ها را کنار دست او می‌گذاریم، تا اگر نیاز داشت کلیک کرده و مشاهده نماید. هر سطح و یا هر بانک تنها یک فرم اصلی می‌تواند داشته باشد.

۴. فیلد

کد فیلد

کد فرم

عنوان فیلد

کد نوع فیلد

قابل تکرار

فیلد ترکیبی

هر فرم می‌تواند هر تعداد فیلد در خود داشته باشد. نوع فیلد از طریق جدول نوع فیلد تعیین می‌شود. اگر این فیلد تکرارپذیر باشد، یعنی کاربر هنگام ورود اطلاعات قادر باشد بیش از یک اطلاع وارد آن نماید، در همین جدول مشخص خواهد شد. اگر هم فیلد ترکیبی باشد، همین‌جا ثبت می‌شود تا در هنگام گرفتن لیست فیلدهای فرم برای نمایش مجبور نباشیم جدول فیلد- ترکیبی را چک کنیم. اگر نیاز نبود می‌توانیم این فلگ را حذف کرده و از طریق بررسی آن جدول دیگر به ترکیبی‌ها برسیم.

۵. فرم - فیلد اصلی

کد فرم

کد فیلد اول

کد فیلد دوم

کد فیلد سوم

پیشوند اول

پیشوند دوم

پیشوند سوم

تنها فرم‌های اصلی می‌توانند فیلد اصلی داشته باشند و حتماً هم باید داشته باشند. پس اگر فرم اصلی تغییر کرد، باید فیلد اصلی جدید تعریف شود. این جدول فیلدهای اصلی را نگهداری می‌نماید. ما قبلاً در سامانه مدیریت اطلاعات پویا (pdm) برای هر فرم اصلی یک فیلد اصلی از نوع عبارت در نظر گرفته بودیم که ما را با یک مشکل مهم

مواجه کرد: ناکافی بودن یک فیلد برای نشان دادن رکورد! مثلاً می‌خواهیم به یک شخص ارجاع دهیم، از فیلد بانک دیگر، نام خانوادگی کافی نیست، نام هم کافی نیست. اما اگر بتوان هر دو را ترکیب کرد مثلاً با نام پدر، ارجاع دقیق‌تر صورت می‌گیرد. در حقیقت فیلدهای سه‌گانه اصلی که با هم ترکیب می‌شوند، یک شاخص برای رکورد ایجاد می‌کنند. تعیین آن نیز بر عهده مدیر سامانه است. نوع فیلد را هم دیگر به «عبارت» محدود نمی‌کنیم، شاید بخواهد مثلاً نوع مؤسسه که از اطلاعات پایه فراخوانی می‌شود را هم داخل سه‌گانه شاخص قرار دهد. سه پیشوند کمک می‌کنند که ترکیب سه فیلد معنادار گردد، جایی که نیاز باشد.

۶. نوع فیلد

کد نوع فیلد

عنوان نوع فیلد

تگ نوع نمایش

نیازمند لوکاپ

انواع ۱۳ گانه‌ای که از فیلدها پیشنهاد داده‌ایم در این جدول وارد می‌شود توسط برنامه‌نویس. مدیر سامانه دسترسی به این جدول برای ویرایش ندارد. اصلاً این جدول نیاز ندارد که صفحات ورود و ویرایش داشته باشد. تیم فنی اطلاعات آن را مستقیم وارد دیتابیس می‌کند. از ۱۳ نوع فیلد ما ۸ روش نمایش فیلد داریم. این ۸ روش را با تگ‌هایی در جدول ثبت می‌کنیم، تا هنگام ساختن عناصر HTML بتوان به سادگی ظاهر را ساخت. یعنی وقتی کاربر در حال ورود اطلاعات در یکی از فرم‌هاست، فیلدهای قابل ویرایش به یکی از این ۸ صورت دیده می‌شوند. مثلاً ورودی یک خطی کوچک هم برای «عدد» کاربرد دارد، هم «نشانی الکترونیکی» و هم «نشانی سایت». اگر تیم فنی تشخیص داد که باید این موارد هم جدا شوند، می‌شود این تگ‌ها را افزایش داد. یا اگر ۱۳ نوع فیلد واقعاً متمایز باشند، می‌توان تگ را کنار گذاشت و مستقیماً برای هر نوع فیلد یک ساختار HTML در نظر گرفت. فیلدهایی که باید از «اطلاعات پایه» انتخاب شوند و یا از «بانک دیگر» مشخص خواهند شد که نیازمند لوکاپ هستند. این نیاز نیز توسط برنامه‌نویس تعیین شده و در این جدول ثبت می‌گردد.

۷. فیلد - ترکیبی

کد فیلد پدر

کد فیلد پسر

می‌توانستیم یک Parent در همان جدول فیلد بگذاریم که نشان دهد فیلد مورد نظر زیرمجموعه کدام فیلد ترکیبی است. ولی به نظر رسید احتمالاً بسیاری از فیلدها غیر زیرمجموعه باشند و این فیلد زیاد خالی بماند. البته شاید آن روش برای برنامه‌نویسی راحت‌تر باشد. این اختیار برای تیم فنی که ارتباط فیلدهای ترکیبی با زیرمجموعه‌هایشان را از طریق این جدول برقرار نمایند و یا اگر این روش را دشوار دیدند، به آن روش دیگر عمل کنند. در این جدول، فیلد پدر همیشه یک فیلد ترکیبی است. موشح: معمولاً فیلدهای ترکیبی تکرارپذیر هستند، یعنی اصلاً ترکیب می‌کنند که در تکرار، اتصال فیلدها حفظ شود. تصور می‌شود بهتر است تا فیلدی ترکیبی شد، آن را تکرارپذیر در نظر بگیریم.

۸. فیلد - محاسباتی

این جدول را هم می‌شود با یک فیلد بولین در جدول فیلد جایگزین کرد. اما فقط در چند عدد از رکوردها کاربرد دارد و سرباره داده‌ای‌اش ممکن است مقبول نباشد. در هر صورت هر فیلدی که محاسباتی تعیین گردد، یک اتفاق ساده را در پی خواهد داشت و آن چنین است: اولاً این گزینه تنها در بانک برنامه قابل انتخاب است. ثانیاً حتماً باید نوع فیلد عددی باشد. ثالثاً باید در پایین‌ترین سطح بانک برنامه هم باشد. مثلاً اگر سطوح ما: برنامه، طرح، پروژه، باشند، فقط در پروژه. حالا چه اتفاقی می‌افتد؟ وقتی در حال نمایش یک رکورد از سطح طرح در بانک برنامه هستیم، اگر آن رکورد متصل باشد به چند رکورد دیگر از سطح پروژه در بانک برنامه، و آن پروژه‌ها در یکی از فرم‌های خود فیلدی عددی داشته باشند که محاسباتی ثبت شده باشد، حالا برنامه‌نویس همان فیلد را عیناً نامش را در فرم اصلی رکورد حاضر از سطح طرح نمایش می‌دهد، در حالی که واقعاً در این فرم چنین فیلدی نیست، رقم آن را نیز از جمع کردن آن فیلد در تمامی رکوردهای پروژه‌های زیرمجموعه آن طرح به دست می‌آورد و نمایش می‌دهد. مثلاً بودجه تمام پروژه‌ها را جمع کرده و بودجه طرح را ایجاد می‌کند! همین اتفاق در سطح برنامه هم می‌افتد، ولی البته با دولایه ارجاع و جمع کردن. موشخ: ما در جدول تنظیمات چند گزینه داریم که فیلدهای اصلی بودجه را تعیین می‌نماید، از طریق کدهای فیلدها که مدیر سامانه در بخش تنظیمات وارد می‌کند. شاید همین بخش تنظیمات ما را از جدول فیلد-محاسباتی بی‌نیاز نماید. زیرا هر فیلدی که مربوط به بودجه بود و قرار است هزینه‌ها روی آن‌ها انجام شود، حتماً محاسباتی هم هست. اما با توجه به این که ممکن است ما فیلدهای محاسباتی دیگری نیز داشته باشیم که غیربودجه‌ای باشند، خوب است به بودن این جدول توجه بیشتری کرد. شاید امکان مفیدی باشد که فیلدهای غیر «اصل بودجه» را هم بشود محاسباتی نمود تا در تجمیع پروژه‌ها جمع شوند.

موشخ: اگر دقت فرمایید اصلاً بانک بودجه متفاوت از بانک‌های دیگر، در دسترس مستقیم کاربر نیست. یعنی رکوردهایش مانند سایر بانک‌ها در صفحات سامانه فهرست نمی‌شوند. تمامی رکوردهای بانک بودجه تنها و تنها در صورتی که به یک رکورد از بانک برنامه متصل باشند معنادار می‌گردند. در حقیقت بودجه یکی از فیلدها یا فرم‌های بانک برنامه می‌بایست می‌بود، اما جدا کردیم و در بانکی مستقل تا مدیر سامانه دستش بازتر باشد و ساده‌تر در طراحی فرم و فیلد برای نظم دادن به بودجه. دیگر ناچار نیست برای هر سطح از بانک برنامه یک فرم بودجه هم بسازد و یک ساختار واحد بودجه‌ای برای همه سطوح عمل می‌کند.

موشخ: حالا تنها مسأله این است که بانک بودجه بیشتر درگیر عملیات‌های خودکار سامانه است، پس باید هر جا که نیاز به عملیات خاصی وجود دارد، کد فیلدها در بخش تنظیمات وارد شود تا برنامه به صورت خودکار بتواند آن فیلد را در حساب و کتاب خود مورد استفاده قرار دهد.

موشخ: یک مسأله دیگر هم هست؛ وقتی مدیر مافوق بودجه پروژه مادون را تقلیل می‌دهد، باید بودجه اولیه یک‌جایی ثبت گردد. این را بر عهده مدیر سامانه می‌توان گذاشت که فیلدی هم برای ثبت بودجه اولیه در نظر بگیرد، به این صورت که وقتی نهاد بالادستی می‌خواهد بودجه را تقلیل دهد، یک نسخه از بودجه پیشنهادی در فیلدی دیگر از قبل توسط پیشنهاددهنده پروژه ثبت شده که تغییر نمی‌کند، تا در آخر بتوان مابه‌التفاوت تقلیل را در آمارها ذکر نمود.

سؤال: چطور رکوردها از سطوح مختلف با هم متصل می‌شوند موقع ورود اطلاعات؟ بسیار ساده! وقتی یک پروژه را می‌خواهیم ورود اطلاعات کنیم، حتماً باید یک رکورد از سطح بالاتر انتخاب کرده و سپس ذیل آن ایجاد پروژه نماییم. زیرا روال کار چنین است که ابتدا یک برنامه کلان داده می‌شود و سپس نهادهای زیرمجموعه می‌توانند ذیل آن برنامه، پروژه‌های خود را پیشنهاد کنند.

کد فیلد

تگ لوکاپ

کد منبع لوکاپ

کد سطح لوکاپ

هر فیلدی که نیازمند لوکاپ باشد، یک رکورد در این جدول خواهد داشت که تعیین می‌نماید از کجا باید لوکاپ شود. این جدول هنگام اضافه کردن فیلد به فرم‌ها توسط مدیر سامانه پر می‌شود. هر جا که مدیر سامانه فیلدی از نوع «انتخابی» استفاده کرد، باید این جدول برای آن به صورت خوکار پر شود. سامانه از طریق تگ لوکاپ می‌فهمد که کد منبع آیا به جدول اطلاعات پایه ارجاع می‌دهد و یا به جدول بانک‌ها. در هنگام ورود اطلاعات، به سراغ منبع مربوطه رفته و رکوردهای مورد نیاز را فراخوانی کرده و برای گزینش نمایش می‌دهد. موشج: با توجه به این که بانک برنامه سطوح مختلف دارد، اگر تگ لوکاپ به بانک اشاره کرد و کد منبع هم بانک برنامه بود، باید کد سطح هم اضافه شود تا معلوم گردد قرار است از کدام سطح لوکاپ صورت گیرد.

۱۰. اطلاعات پایه

کد اطلاعات پایه

عنوان اطلاعات پایه

این جدول فقط عنوان هر دسته از اطلاعات پایه را دارد، مثلاً: وضعیت نظام وظیفه.

۱۱. اطلاعات پایه - اطلاع

کد اطلاع پایه

کد اطلاعات پایه

عنوان اطلاع پایه

ریز اطلاعات قابل گزینش در این جدول ثبت می‌شود که مدیر سامانه می‌تواند وارد کرده یا ویرایش کند. مثلاً: معافیت تحصیلی، معافیت پزشکی، پایان خدمت و سرباز فراری را می‌شود ذیل وضعیت نظام وظیفه ثبت نمود که هنگام ورود اطلاعات قابل گزینش باشند.

۱۲. رکورد

کد رکورد

کد بانک

کد سطح

هر رکورد از اطلاعات اگر بخواهد وارد سامانه شود، در هر کدام از چهار بانک: نهاد، برنامه، بودجه و جنبی، حتماً باید ابتدا یک کد در این جدول به آن تخصیص داده شود. اگر رکورد مزبور از بانک برنامه باشد، کد سطح را نیاز دارد

تا بتوان چندین رکورد پله‌ای در بانک مزبور داشت. تمامی رکوردهای سامانه این‌جا کد خواهند داشت و چون کدها یکسان نیستند، در حقیقت ما برای هر رکورد در بودجه‌بان یک کد منحصر به فرد داریم.

۱۳. محتوا

کد محتوا

کد رکورد

کد فیلد

متن محتوا

هر رکورد از داده‌های سامانه متشکل از چندین فرم می‌باشند که هر فرم نیز چندین فیلد دارد. اطلاعات این فیلدها که در حقیقت همان اطلاعات اصلی سامانه هستند، در جدول محتوا نگهداری می‌شوند. عدد، عبارت، لینک، آدرس، فایل، حتی کدی که اطلاع پایه را مشخص می‌سازد و یا شماره یک رکورد از بانکی دیگر که انتخاب شده، همه و همه در همین جدول قرار می‌گیرند. به عبارت دیگر، تمام اطلاعات سامانه همین یک جدول است و سایر جداول فقط وظیفه و نقش تبیین ارتباط و فرمت اطلاعات این جدول را بر عهده دارند. طبیعتاً جستجو در سامانه بسیار ساده‌تر است برای دستیابی به اطلاعات، زیرا تنها یک فیلد از یک جدول سرچ می‌شود.

سؤال: عرض این جدول کم است (تعداد فیلد)، اما ارتفاع آن وحشتناک خواهد بود (تعداد رکورد)، چنین کاری از سرعت نمی‌کاهد؟ پاره‌ای برنامه‌نویسان معتقدند که هر چه تعداد جداول بیشتر باشد و تعداد رکوردهای هر یک کمتر، دسترسی به اطلاعات سریع‌تر خواهد بود، ولی تجربه نشان داده ابزارهای مدیریت بانک‌های اطلاعاتی امروزی در جداول حجیم خیلی سریع و خوب عمل می‌کنند. سامانه مدیریت اطلاعات پویا چنین معماری‌ای دارد و در یک نسخه آن بالغ بر ۱۷ هزار رکورد دارای فرم‌ها و فیلدهای متعدد و بعضاً با اطلاعات متنی حجیم وارد شده است که از سرعت سامانه نکاسته است. در حالی که تعداد نهادها و پروژه‌ها و بودجه‌های بودجه‌بان در نسخه‌های متعدّد آن شاید به هزار هم نرسد.

۱۴. رکورد - سطح

کد رکورد پدر

کد رکورد پدر

نهادها به صورت طبقاتی ثبت می‌شوند. یعنی نهادهای رده اصلی شاید بیش از ۳۰ عدد نباشند، اما هر یک نهادهای دیگری زیرمجموعه خود دارند. کار بدین صورت است که اگر مدیر سامانه یا نهاد بودجه‌گذار یک رکورد به بانک نهاد وارد نمایند، حتماً Parent خواهد بود. اما این نهادهای رده یک وقتی بخواهند وارد سامانه شده و نهاد ایجاد کنند، هر تعداد نهاد که ورود اطلاعات نمایند، خودشان پدر آن بوده و آن نهاد پسرشان محسوب می‌گردد. نهادهای پسر نیز خود می‌توانند پدر باشند. این جدول اطلاعات سطوح نهادها را ثبت می‌کند. موشح: هیچ کجای سامانه ابزاری وجود ندارد که کاربران بتوانند رکوردها را سطح‌بندی و طبقه‌بندی کنند. این عملیات به صورت خودکار انجام می‌شود، وقتی که یک نهاد مشغول ثبت رکوردهای نهادهای زیرمجموعه خود است. در این حالت کد نهاد جاری به عنوان پدر برای نهادهای جدید ثبت می‌شود.

۱۵. رکورد - تأیید

کد تأیید

کد رکورد مورد تأیید

کد فرم مورد تأیید

کد رکورد نهاد تأییدکننده

تأیید شده

توضیح تأییدکننده

تاریخ تأیید

عملیات تأیید به صورت اجمالی در سه نقطه از سامانه به انجام می‌رسد. نخست جایی که پروژه پیشنهاد می‌شود، هر نهاد مافوق آن را تأیید می‌کند تا بودجه‌گذار. بودجه‌گذار علاوه بر تأیید رکوردهای بانک برنامه، رکوردهای بانک بودجه را نیز باید تأیید نماید، همان رکوردی از بانک بودجه که به رکورد پروژه از بانک برنامه متصل است. سومین تأیید نیز مربوط به گزارشات است. گزارش یکی از فرم‌های پروژه در بانک برنامه است. وقتی مدیر پروژه آن را تکمیل نمود، ابتدا ناظر تأیید می‌کند، سپس نهاد مافوق و در نهایت نهاد بودجه‌گذار. ما از طریق جدول رکورد-نمایش اجباری می‌توانیم به صورت خودکار هر نهاد را مجبور نماییم فرمی را مدام مشاهده کند که باید تأیید یا رد نماید. اگر یک رکورد تأیید نشود، اگر تاریخ تأیید بخورد و همچنان تأیید نشده باقی‌مانده باشد، معنای آن رد شدن بوده و بازگشت به کارتابل نهاد قبلی. موشح: وقتی فرم «اطلاعات اصلی» پروژه را نمایش می‌دهیم برای کسی که خودش ایجاد کننده آن رکورد بوده، یعنی درست وقتی که پیشنهاددهنده پروژه دارد به فرم اصلی بانک برنامه در سطح پروژه نگاه می‌کند، یک کلید به ذیل فرم اضافه می‌نماییم: «ارسال پیشنهاد پروژه». با فشردن این دکمه یک رکورد جدید به جدول «رکورد-نمایش اجباری» اضافه می‌شود که مدیر بالادستی را ناگزیر می‌کند این فرم را در کارتابل خود مشاهده کند. مادامی که این رکورد وجود دارد، دکمه «ارسال پیشنهاد پروژه» غیرفعال می‌ماند. در نهایت این سناریو، وقتی مدیر بالادستی دکمه «تأیید» را زد، اصلاً دکمه «ارسال پیشنهاد پروژه» دیگر نمایش داده نمی‌شود، زیرا این پروژه تصویب شده است و باید اجرایی شود. اما اگر «رد و اصلاح» زد، یعنی بازگشت داد تا اصلاح شود، دوباره دکمه «ارسال پیشنهاد پروژه» فعال می‌شود. (چه رد و چه تأیید، نتیجه در جدول «رکورد-تأیید» ثبت می‌شود) اکنون پیشنهاددهنده پروژه می‌تواند فرم پروژه را ویرایش نموده و هر وقت آماده شده دوباره دکمه «ارسال پیشنهاد پروژه» را بزند. می‌تواند یک دکمه «صرف‌نظر از پروژه» هم داشته باشد که یا رکورد پروژه را حذف کند و یا از دور فقط خارج نماید.

۱۶. رکورد - بسته

کد رکورد

کد فرم

کد فیلد

تاریخ بسته شدن



بعضی فیلدهای بعضی رکوردها و یا بعضی فرم‌های بعضی رکوردها در بعضی مراحل کار تبدیل به «فقط خواندنی» می‌شوند. این جدول چنین اطلاعاتی را نگهداری می‌نماید. مثلاً وقتی نهاد بودجه‌گذار یک پروژه را تصویب کرد (رکورد جدول رکورد- تأیید آن ایجاد شد و تأیید بود) بلافاصله باید فرم اصلی آن رکورد از بانک برنامه قفل شود، بسته شود و فقط امکان مشاهده آن برای نهاد پیشنهاددهنده پروژه وجود داشته باشد. ما اجازه ویرایش را عملاً با این کار از آن نهاد می‌گیریم. طبیعتاً این‌گونه تغییرات زنجیره‌ای در دیتابیس را حتماً باید از طریق ترانزاکشن انجام دهیم تا اگر یک بانک ویرایش شد و در ویرایش سایر بانک‌ها مشکل پدید آمد، رول‌بک شود خودبه‌خود. در غیر این صورت دیتابیس ناهماهنگ و به هم ریخته خواهد شد، مثلاً تصویری صورت گرفته ولی فرم آن بسته نشده!

۱۷. رکورد - نمایش اجباری

کد رکورد نهاد اجبار کننده

کد رکورد نهاد اجبار شونده

کد رکورد اجباری

کد فرم اجباری

دیده شده

تاریخ ثبت اجبار

تاریخ اعمال اجبار

اولاً این‌که اصلاً این جدول کارش نمایش اجباری یک فرم یک رکورد خاص از یک بانک است. نمایش اجباری دقیقاً کار همان کارتابل را انجام می‌دهد، کارتابلی که هر کاربر هنگام ورود به سامانه در اختیار خواهد داشت. به دو گونه این جدول پر می‌شود؛ به صورت خودکار و به صورت دستی. وجه خودکار آن فراوان است در مواردی که گردش کار ساخته می‌شود. وقتی مثلاً یک نهاد پروژه‌ای را پیشنهاد می‌دهد، یک رکورد اجبار در این جدول ثبت می‌شود که نهاد مافوق را ملزم می‌کند تا این رکورد جدید را ببیند. این رکورد در کارتابل او خواهد بود تا فلگ دیده‌شده علامت بخورد. سپس به آرشیو کارتابل منتقل می‌گردد. دستی این‌چنین است که مثلاً نهاد بودجه‌گذار وقتی بخواهد توجه یک نهادی را به یک قانون مثلاً از بانک جنبی جلب نماید، این کار را این‌چنین انجام می‌دهد. موشح: پس در بانک اطلاعات جنبی جلوی نام هر فرم یک دگمه قرار داده می‌شود، اگر کلیک شود و یک یا چند تا از نهادهای پایین دستی انتخاب گردد، آن فرم برای آن نهاد(ها) اجباراً دیده خواهد شد. می‌توان تاریخ نمایش را هم آینده در نظر گرفت با ثبت «تاریخ اعمال اجبار». در این صورت فرم مزبور در تاریخ ذکر شده داخل کارتابل مخاطب می‌رود.

بودجه‌بان یک سامانه پیام‌رسانی نیز لازم دارد که ما جدولی برای آن در نظر نگرفتیم. شاید یک یا دو جدول نیاز داشته باشد که تیم فنی خود می‌توانند آن را طراحی نمایند. می‌شود هم با ترفندهایی از همین جدول برای پیام‌رسانی استفاده کرد که وارد جزئیات آن نمی‌شویم. تنها نکته خاص در سرویس پیام‌رسانی بودجه‌بان این است که مقصد پیام تنها دو حالت می‌تواند داشته باشد: نهاد فرادست یا نهاد فرودست. بودجه‌گذار تنها می‌تواند برای نهادهای رده یک و اصلی پیام ارسال نماید. آن‌ها هم برای بودجه‌گذار و هم برای نهادهای زیردست خود.

۱۸. کاربر

کد کاربر

کد رکورد نهاد

نام کاربری

رمز عبور هش شده

غیر فعال بودن

تگ نقش در سامانه

فقط سه نقش در سامانه داریم که دو تای از آنها رکورد مشخصی در بانک نهاد دارند. تگ نقش تعیین کننده نقش کاربر بوده و دسترسی های او را طبق پیش فرضی که در فلوچارت بودجه بان نمایش داده شده مدیریت می کند. مدیر سامانه هم که هیچ رکوردی ندارد و فقط یک رکورد در همین جدول است. کاربرهای بودجه گذار و نهادهای رده اول توسط مدیر سامانه قابل تعریف هستند. بودجه گذار می تواند نهادهای رده اول را کاربر نماید. هر نهاد عملگر نیز نهادهای زیرمجموعه خود را نام کاربری می دهد. موشح: برای تبدیل یک نهاد به یک کاربر، کافیست در صفحه «ایجاد کاربر» نام آن نهاد از فهرست انتخاب شود و نام کاربری و رمز عبور اولیه ثبت گردد. از آن لحظه آن نهاد یک کاربر سامانه محسوب می گردد. این وظیفه برای نهادهای رده یک بر عهده مدیر سامانه و یا نهاد بودجه گذار است و برای بودجه گذار تنها بر عهده مدیر سامانه. سایر نهادهای زیرمجموعه نیز توسط نهاد بالادستی خود کاربر می شوند. دسترسی به ریست کردن رمز نیز به همین ترتیب می باشد.

۱۹. لاگ - ورود

کد کاربر

تاریخ ورود

نشانی آی بی

رشته ایجنت مرورگر

هر ورودی به سامانه در این جدول لاگ می شود. می توان و باید هم جداول دیگری داشت که ورودهای غیرمجاز، یا تلاش برای هک کردن را ثبت کند. SQL اینجکشن را هم وقتی تشخیص دادیم، خوب است در یک جدول دیگر لاگ کنیم، تا مدیر سامانه با دیدن این اطلاعات از وضعیت سامانه مطلع شود. خوب است به جای استفاده از کپیچا، تعداد تلاش های ناموفق برای ورود محدود شود. مثلاً اگر ۵ بار رمز را غلط زد، آن IP برای ده دقیقه بسته شود، یا آن نام کاربری. تشخیص و تصمیم بر عهده تیم فنی.

با آرزوی موفقیت برای تمامی اعضای تیم فنی سامانه بودجه بان، متن فوق تنها یک پیشنهاد اولیه است و خوب است اصلاحات و ویرایش خود را همین جا ثبت نمایید. ضمن این که همیشه برنامه نویسان در هنگام کار با نیازهای جدیدی مواجه می شوند که بایستی به صورت معقول به بانک اضافه کنند.

سیدمهدی موسوی موشح
۱۰ آبان ۱۳۹۲