به نام خدا

راهنماي جداول اطلاعاتي بودجه‌بان ـ ويرايش 2 ـ 14/8/1392

#  بانك

## كد بانك

## عنوان بانك

اين سامانه داراي چهار بانك اطلاعاتي است كه نام آن‌ها در اين جدول ذخيره مي‌شود و هر كدام يك كد دارند كه در جدول ركورد و جدول فرم مورد استفاده قرار مي‌گيرند. فيلدهاي «انتخاب از بانك ديگر» نيز به همين كد ارجاع مي‌دهند. مي‌توانستيم حالا كه نام بانك‌ها را مي‌دانيم و تعدادشان محدود است، از جدول استفاده نكنيم، در اين صورت امكان افزايش بانك‌ها را در آينده نداشتيم. اين احتمال وجود دارد كه در توسعه‌هاي بعدي نرم‌افزار نياز به افزايش بانك‌ها پيش آيد.

#  برنامه - سطح

## كد سطح

## عنوان سطح

## كد سطح پدر

استثناءاً در بانك برنامه‌ها نياز به سطح‌بندي داريم. سطح‌بندي اين معنا را دارد كه عملاً مدير سامانه اين اختيار را دارد كه به صورت مجازي چند بانك پله‌اي ايجاد نمايد كه هر كدام وابسته و تحت ديگري است. طبيعتاً هر كدام از اين بانك‌هاي مجازي (همان سطح‌ها) فرم‌هاي مختص به خود را خواهند داشت. اما در محاسبات مالي، ارقام بودجه از هر پله به پله بالاتر رفته و جمع خواهد شد. اين جدول هم اسامي سطح‌ها را نگهداري مي‌نمايد و هم ترتيب پله‌اي آن‌ها را با استفاده از كد سطح پدر (Parent-Child). نكته: هر ركورد از سطح پايين لزوماً بايد به يك ركورد از سطح بالاتر خود متصل باشد. پس نمي‌توان ركورد جديد به بانك برنامه اضافه كرد، مگر اين‌كه حتماً به ركوردي از سطح بالاتر وصل شده باشد

#  فرم

## كد فرم

## كد بانك

## كد سطح

## عنوان فرم

## فرم اصلي

هر تعداد فرم مي‌توان در ذيل يك بانك داشت. اكنون كه سطح‌هايي در بانك برنامه داريم ناگزير هستيم كه كد سطح را هم به فرم ضميمه نماييم. طبيعتاً اين كد معمولاً خالي خواهد بود، فقط هنگامي كه كد بانك 2 باشد (مثلاً كد بانك برنامه) ما بايد كد سطح را هم چك كنيم و سپس فرم‌ها را نمايش دهيم. بنابراين هر سطح از بانك برنامه فرم‌هاي مختص به خود را خواهد داشت. وقتي يك كاربر روي بانك كليك مي‌نمايد، ما محتواي فرم اصلي را نمايش مي‌دهيم و لينك ساير فرم‌ها را كنار دست او مي‌گذاريم، تا اگر نياز داشت كليك كرده و مشاهده نمايد. هر سطح و يا هر بانك تنها يك فرم اصلي مي‌تواند داشته باشد.

#  فيلد

## كد فيلد

## كد فرم

## عنوان فيلد

## كد نوع فيلد

## قابل تكرار

## فيلد تركيبي

هر فرم مي‌تواند هر تعداد فيلد در خود داشته باشد. نوع فيلد از طريق جدول نوع فيلد تعيين مي‌شود. اگر اين فيلد تكرارپذير باشد، يعني كاربر هنگام ورود اطلاعات قادر باشد بيش از يك اطلاع وارد آن نمايد، در همين جدول مشخص خواهد شد. اگر هم فيلد تركيبي باشد، همين‌جا ثبت مي‌شود تا در هنگام گرفتن ليست فيلدهاي فرم براي نمايش مجبور نباشيم جدول فيلد- تركيبي را چك كنيم. اگر نياز نبود مي‌توانيم اين فلگ را حذف كرده و از طريق بررسي آن جدول ديگر به تركيبي‌ها برسيم.

#  فرم - فيلد اصلي

## كد فرم

## كد فيلد اول

## كد فيلد دوم

## كد فيلد سوّم

## پيشوند اول

## پيشوند دوم

## پيشوند سوم

تنها فرم‌هاي اصلي مي‌توانند فيلد اصلي داشته باشند و حتماً هم بايد داشته باشند. پس اگر فرم اصلي تغيير كرد، بايد فيلد اصلي جديد تعريف شود. اين جدول فيلدهاي اصلي را نگهداري مي‌نمايد. ما قبلاً در سامانه مديريت اطلاعات پويا (pdm) براي هر فرم اصلي يك فيلد اصلي از نوع عبارت در نظر گرفته بوديم كه ما را با يك مشكل مهم مواجه كرد: ناكافي بودن يك فيلد براي نشان دادن ركورد! مثلاً مي‌خواهيم به يك شخص ارجاع دهيم، از فيلد بانك ديگر، نام خانوادگي كافي نيست، نام هم كافي نيست. اما اگر بتوان هر دو را تركيب كرد مثلاً با نام پدر، ارجاع دقيق‌تر صورت مي‌گيرد. در حقيقت فيلدهاي سه‌گانه اصلي كه با هم تركيب مي‌شوند، يك شاخص براي ركورد ايجاد مي‌كنند. تعيين آن نيز بر عهده مدير سامانه است. نوع فيلد را هم ديگر به «عبارت» محدود نمي‌كنيم، شايد بخواهد مثلاً نوع مؤسسه كه از اطلاعات پايه فراخواني مي‌شود را هم داخل سه‌گانه شاخص قرار دهد. سه پيشوند كمك مي‌كنند كه تركيب سه فيلد معنادار گردد، جايي‌كه نياز باشد.

#  نوع فيلد

## كد نوع فيلد

## عنوان نوع فيلد

## تگ نوع نمايش

## نيازمند لوكاپ

انواع 13گانه‌اي كه از فيلدها پيشنهاد داده‌ايم در اين جدول وارد مي‌شود توسط برنامه‌نويس. مدير سامانه دسترسي به اين جدول براي ويرايش ندارد. اصلاً اين جدول نياز ندارد كه صفحات ورود و ويرايش داشته باشد. تيم فني اطلاعات آن را مستقيم وارد ديتابيس مي‌كند. از 13 نوع فيلد ما 8 روش نمايش فيلد داريم. اين 8 روش را با تگ‌هايي در جدول ثبت مي‌كنيم، تا هنگام ساختن عناصر HTML بتوان به سادگي ظاهر را ساخت. يعني وقتي كاربر در حال ورود اطلاعات در يكي از فرم‌هاست، فيلدهاي قابل ويرايش به يكي از اين 8 صورت ديده مي‌شوند. مثلاً ورودي يك‌خطي كوچك هم براي «عدد» كاربرد دارد، هم «نشاني الكترونيكي» و هم «نشاني سايت». اگر تيم فني تشخيص داد كه بايد اين موارد هم جدا شوند، مي‌شود اين تگ‌ها را افزايش داد. يا اگر 13 نوع فيلد واقعاً متمايز باشند، مي‌توان تگ را كنار گذاشت و مستقيماً براي هر نوع فيلد يك ساختار HTML در نظر گرفت. فيلدهايي كه بايد از «اطلاعات پايه» انتخاب شوند و يا از «بانك ديگر» مشخص خواهند شد كه نيازمند لوكاپ هستند. اين نياز نيز توسط برنامه‌نويس تعيين شده و در اين جدول ثبت مي‌گردد.

#  فيلد - تركيبي

## كد فيلد پدر

## كد فيلد پسر

مي‌توانستيم يك Parent در همان جدول فيلد بگذاريم كه نشان دهد فيلد مورد نظر زيرمجموعه كدام فيلد تركيبي است. ولي به نظر رسيد احتمالاً بسياري از فيلدها غير زيرمجموعه باشند و اين فيلد زياد خالي بماند. البته شايد آن روش براي برنامه‌نويسي راحت‌تر باشد. اين اختيار براي تيم فني كه ارتباط فيلدهاي تركيبي با زيرمجموعه‌هايشان را از طريق اين جدول برقرار نمايند و يا اگر اين روش را دشوار ديدند، به آن روش ديگر عمل كنند. در اين جدول، فيلد پدر هميشه يك فيلد تركيبي است. موشَّح: معمولاً فيلدهاي تركيبي تكرارپذير هستند، يعني اصلاً تركيب مي‌كنند كه در تكرار، اتصال فيلدها حفظ شود. تصور مي‌شود بهتر است تا فيلدي تركيبي شد، آن را تكرارپذير در نظر بگيريم.

#  فيلد - محاسباتي

## كد فيلد

اين جدول را هم مي‌شود با يك فيلد بولين در جدول فيلد جايگزين كرد. اما فقط در چند عدد از ركوردها كاربرد دارد و سرباره داده‌اي‌اش ممكن است مقبول نباشد. در هر صورت هر فيلدي كه محاسباتي تعيين گردد، يك اتفاق ساده را در پي خواهد داشت و آن چنين است: اولاً اين گزينه تنها در بانك برنامه قابل انتخاب است. ثانياً حتماً بايد نوع فيلد عددي باشد. ثالثاً بايد در پايين‌ترين سطح بانك برنامه هم باشد. مثلاً اگر سطوح ما: برنامه، طرح، پروژه،‌ باشند، فقط در پروژه. حالا چه اتفاقي مي‌افتد؟ وقتي در حال نمايش يك ركورد از سطح طرح در بانك برنامه هستيم، اگر آن ركورد متصل باشد به چند ركورد ديگر از سطح پروژه در بانك برنامه، و آن پروژه‌ها در يكي از فرم‌هاي خود فيلدي عددي داشته باشند كه محاسباتي ثبت شده باشد، حالا برنامه‌نويس همان فيلد را عيناً نامش را در فرم اصلي ركورد حاضر از سطح طرح نمايش مي‌دهد، در حالي كه واقعاً در اين فرم چنين فيلدي نيست، رقم آن را نيز از جمع كردن آن فيلد در تمامي ركوردهاي پروژه‌هاي زيرمجموعه آن طرح به دست مي‌آورد و نمايش مي‌دهد. مثلاً بودجه تمام پروژه‌ها را جمع كرده و بودجه طرح را ايجاد مي‌كند! همين اتفاق در سطح برنامه هم مي‌افتد، ولي البته با دولايه ارجاع و جمع كردن. موشَّح: ما در جدول تنظيمات چند گزينه داريم كه فيلدهاي اصلي بودجه را تعيين مي‌نمايد، از طريق كدهاي فيلدها كه مدير سامانه در بخش تنظيمات وارد مي‌كند. شايد همين بخش تنظيمات ما را از جدول فيلد-محاسباتي بي‌نياز نمايد. زيرا هر فيلدي كه مربوط به بودجه بود و قرار است هزينه‌ها روي آن‌ها انجام شود، حتماً محاسباتي هم هست. اما با توجه به اين‌كه ممكن است ما فيلدهاي محاسباتي ديگري نيز داشته باشيم كه غيربودجه‌اي باشند، خوب است به بودن اين جدول توجه بيشتري كرد. شايد امكان مفيدي باشد كه فيلدهاي غير «اصل بودجه» را هم بشود محاسباتي نمود تا در تجميع پروژه‌ها جمع شوند.

موشَّح: اگر دقت فرماييد اصلاً بانك بودجه متفاوت از بانك‌هاي ديگر، در دسترس مستقيم كاربر نيست. يعني ركوردهايش مانند ساير بانك‌ها در صفحات سامانه فهرست نمي‌شوند. تمامي ركوردهاي بانك بودجه تنها و تنها در صورتي كه به يك ركورد از بانك برنامه متصل باشند معنادار مي‌گردند. در حقيقت بودجه يكي از فيلدها يا فرم‌هاي بانك برنامه مي‌بايست مي‌بود، اما جدا كرديم و در بانكي مستقل تا مدير سامانه دستش بازتر باشد و ساده‌تر در طراحي فرم و فيلد براي نظم دادن به بودجه. ديگر ناچار نيست براي هر سطح از بانك برنامه يك فرم بودجه هم بسازد و يك ساختار واحد بودجه‌اي براي همه سطوح عمل مي‌كند.

موشَّح: حالا تنها مسأله اين است كه بانك بودجه بيشتر درگير عمليات‌هاي خودكار سامانه است، پس بايد هر جا كه نياز به عمليات خاصّي وجود دارد، كد فيلدها در بخش تنظيمات وارد شود تا برنامه به صورت خودكار بتواند آن فيلد را در حساب و كتاب خود مورد استفاده قرار دهد.

موشَّح: يك مسأله ديگر هم هست؛ وقتي مدير مافوق بودجه پروژه مادون را تقليل مي‌دهد، بايد بودجه اوليه يك‌جايي ثبت گردد. اين را بر عهده مديرسامانه مي‌توان گذاشت كه فيلدي هم براي ثبت بودجه اوليه در نظر بگيرد، به اين صورت كه وقتي نهاد بالادستي مي‌خواهد بودجه را تقليل دهد، يك نسخه از بودجه پيشنهادي در فيلدي ديگر از قبل توسط پيشنهاددهنده پروژه ثبت شده كه تغيير نمي‌كند، تا در آخر بتوان مابه‌التفاوت تقليل را در آمارها ذكر نمود.

سؤال: چطور ركوردها از سطوح مختلف با هم متصل مي‌شوند موقع ورود اطلاعات؟ بسيار ساده! وقتي يك پروژه را مي‌خواهيم ورود اطلاعات كنيم، حتماً بايد يك ركورد از سطح بالاتر انتخاب كرده و سپس ذيل آن ايجاد پروژه نماييم. زيرا روال كار چنين است كه ابتدا يك برنامه كلان داده مي‌شود و سپس نهادهاي زيرمجموعه مي‌توانند ذيل آن برنامه، پروژه‌هاي خود را پيشنهاد كنند.

#  فيلد - لوكاپ

## كد فيلد

## تگ لوكاپ

## كد منبع لوكاپ

## كد سطح لوكاپ

هر فيلدي كه نيازمند لوكاپ باشد، يك ركورد در اين جدول خواهد داشت كه تعيين مي‌نمايد از كجا بايد لوكاپ شود. اين جدول هنگام اضافه كردن فيلد به فرم‌ها توسط مدير سامانه پر مي‌شود. هر جا كه مدير سامانه فيلدي از نوع «انتخابي» استفاده كرد، بايد اين جدول براي آن به صورت خوكار پر شود. سامانه از طريق تگ لوكاپ مي‌فهمد كه كد منبع آيا به جدول اطلاعات پايه ارجاع مي‌دهد و يا به جدول بانك‌ها. در هنگام ورود اطلاعات، به سراغ منبع مربوطه رفته و ركوردهاي مورد نياز را فراخواني كرده و براي گزينش نمايش مي‌دهد. موشَّح: با توجه به اين‌كه بانك برنامه سطوح مختلف دارد، اگر تگ لوكاپ به بانك اشاره كرد و كد منبع هم بانك برنامه بود، بايد كد سطح هم اضافه شود تا معلوم گردد قرار است از كدام سطح لوكاپ صورت گيرد.

#  اطلاعات پايه

## كد اطلاعات پايه

## عنوان اطلاعات پايه

اين جدول فقط عنوان هر دسته از اطلاعات پايه را دارد، مثلاً: وضعيت نظام وظيفه.

#  اطلاعات پايه - اطلاع

## كد اطلاع پايه

## كد اطلاعات پايه

## عنوان اطلاع پايه

ريز اطلاعات قابل گزينش در اين جدول ثبت مي‌شود كه مدير سامانه مي‌تواند وارد كرده يا ويرايش كند. مثلاً: معافيت تحصيلي، معافيت پزشكي،‌ پايان خدمت و سرباز فراري را مي‌شود ذيل وضعيت نظام وظيفه ثبت نمود كه هنگام ورود اطلاعات قابل گزينش باشند.

#  ركورد

## كد ركورد

## كد بانك

## كد سطح

هر ركورد از اطلاعات اگر بخواهد وارد سامانه شود، در هر كدام از چهار بانك: نهاد، برنامه، بودجه و جنبي، حتماً بايد ابتدا يك كد در اين جدول به آن تخصيص داده شود. اگر ركورد مزبور از بانك برنامه باشد، كد سطح را نياز دارد تا بتوان چندين ركورد پله‌اي در بانك مزبور داشت. تمامي ركوردهاي سامانه اين‌جا كد خواهند داشت و چون كدها يكسان نيستند، در حقيقت ما براي هر ركورد در بودجه‌بان يك كد منحصر به فرد داريم.

#  محتوا

## كد محتوا

## كد ركورد

## كد فيلد

## متن محتوا

هر ركورد از داده‌هاي سامانه متشكل از چندين فرم مي‌باشند كه هر فرم نيز چندين فيلد دارد. اطلاعات اين فيلدها كه در حقيقت همان اطلاعات اصلي سامانه هستند، در جدول محتوا نگهداري مي‌شوند. عدد، عبارت، لينك، آدرس فايل، حتي كدي كه اطلاع پايه را مشخص مي‌سازد و يا شماره يك ركورد از بانكي ديگر كه انتخاب شده، همه و همه در همين جدول قرار مي‌گيرند. به عبارت ديگر، تمام اطلاعات سامانه همين يك جدول است و ساير جداول فقط وظيفه و نقش تبيين ارتباط و فرمت اطلاعات اين جدول را بر عهده دارند. طبيعتاً جستجو در سامانه بسيار ساده‌تر است براي دستيابي به اطلاعات، زيرا تنها يك فيلد از يك جدول سرچ مي‌شود.

سؤال: عرض اين جدول كم است (تعداد فيلد)، اما ارتفاع آن وحشتناك خواهد بود (تعداد ركورد)، چنين كاري از سرعت نمي‌كاهد؟ پاره‌اي برنامه‌نويسان معتقدند كه هر چه تعداد جداول بيشتر باشد و تعداد ركوردهاي هر يك كمتر، دسترسي به اطلاعات سريع‌تر خواهد بود، ولي تجربه نشان داده ابزارهاي مديريت بانك‌هاي اطلاعاتي امروزين در جداول حجيم خيلي سريع و خوب عمل مي‌كنند. سامانه مديريت اطلاعات پويا چنين معماري‌اي دارد و در يك نسخه آن بالغ بر 17 هزار ركورد داراي فرم‌ها و فيلدهاي متعدد و بعضاً با اطلاعات متني حجيم وارد شده است كه از سرعت سامانه نكاسته است. در حالي كه تعداد نهادها و پروژه‌ها و بودجه‌هاي بودجه‌بان در نسخه‌هاي متعدّد آن شايد به هزار هم نرسد.

#  ركورد - سطح

## كد ركورد پدر

## كد ركورد پدر

نهادها به صورت طبقاتي ثبت مي‌شوند. يعني نهادهاي رده اصلي شايد بيش از 30 عدد نباشند، اما هر يك نهادهاي ديگري زيرمجموعه خود دارند. كار بدين صورت است كه اگر مدير سامانه يا نهاد بودجه‌گذار يك ركورد به بانك نهاد وارد نمايند، حتماً Parent خواهد بود. اما اين نهادهاي رده يك وقتي بخواهند وارد سامانه شده و نهاد ايجاد كنند، هر تعداد نهاد كه ورود اطلاعات نمايند، خودشان پدر آن بوده و آن نهاد پسرشان محسوب مي‌گردد. نهادهاي پسر نيز خود مي‌توانند پدر باشند. اين جدول اطلاعات سطوح نهادها را ثبت مي‌كند. موشَّح: هيچ كجاي سامانه ابزاري وجود ندارد كه كاربران بتوانند ركوردها را سطح‌بندي و طبقه‌بندي كنند. اين عمليات به صورت خودكار انجام مي‌شود،‌ وقتي كه يك نهاد مشغول ثبت ركوردهاي نهادهاي زيرمجموعه خود است. در اين حالت كد نهاد جاري به عنوان پدر براي نهادهاي جديد ثبت مي‌شود.

#  ركورد - تأييد

## كد تأييد

## كد ركورد مورد تأييد

## كد فرم مورد تأييد

## كد ركورد نهاد تأييدكننده

## تأييد شده

## توضيح تأييدكننده

## تاريخ تأييد

عمليات تأييد به صورت اجمالي در سه نقطه از سامانه به انجام مي‌رسد. نخست جايي‌كه پروژه پيشنهاد مي‌شود، هر نهاد مافوق آن را تأييد مي‌كند تا بودجه‌گذار. بودجه‌گذار علاوه بر تأييد ركوردهاي بانك برنامه، ركوردهاي بانك بودجه را نيز بايد تأييد نمايد، همان ركوردي از بانك بودجه كه به ركورد پروژه از بانك برنامه متصل است. سومين تأييد نيز مربوط به گزارشات است. گزارش يكي از فرم‌هاي پروژه در بانك برنامه است. وقتي مدير پروژه آن را تكميل نمود، ابتدا ناظر تأييد مي‌كند، سپس نهاد مافوق و در نهايت نهاد بودجه‌گذار. ما از طريق جدول ركورد- نمايش اجباري مي‌توانيم به صورت خودكار هر نهاد را مجبور نماييم فرمي را مدام مشاهده كند كه بايد تأييد يا رد نمايد. اگر يك ركورد تأييد نشود، اگر تاريخ تأييد بخورد و همچنان تأييد نشده باقي‌مانده باشد، معناي آن رد شدن بوده و بازگشت به كارتابل نهاد قبلي. موشَّح: وقتي فرم «اطلاعات اصلي» پروژه را نمايش مي‌دهيم براي كسي كه خودش ايجاد كننده آن ركورد بوده، يعني درست وقتي كه پيشنهاددهنده پروژه دارد به فرم اصلي بانك برنامه در سطح پروژه نگاه مي‌كند، يك كليد به ذيل فرم اضافه مي‌نماييم: «ارسال پيشنهاد پروژه». با فشردن اين دگمه يك ركورد جديد به جدول «ركورد-نمايش اجباري» اضافه مي‌شود كه مدير بالادستي را ناگزير مي‌كند اين فرم را در كارتابل خود مشاهده كند. مادامي‌كه اين ركورد وجود دارد، دگمه «ارسال پيشنهاد پروژه» غيرفعال مي‌ماند. در نهايتِ اين سناريو، وقتي مدير بالادستي دگمه «تأييد» را زد، اصلاً دگمه «ارسال پيشنهاد پروژه» ديگر نمايش داده نمي‌شود، زيرا اين پروژه تصويب شده است و بايد اجرايي شود. اما اگر «رد و اصلاح» زد، يعني بازگشت داد تا اصلاح شود، دوباره دگمه «ارسال پيشنهاد پروژه» فعّال مي‌شود. (چه رد و چه تأييد، نتيجه در جدول «ركورد-تأييد» ثبت مي‌شود) اكنون پيشنهاددهنده پروژه مي‌تواند فرم پروژه را ويرايش نموده و هر وقت آماده شده دوباره دگمه «ارسال پيشنهاد پروژه» را بزند. مي‌تواند يك دگمه «صرف‌نظر از پروژه» هم داشته باشد كه يا ركورد پروژه را حذف كند و يا از دور فقط خارج نمايد.

#  ركورد - بسته

## كد ركورد

## كد فرم

## كد فيلد

## تاريخ بسته شدن

بعضي فيلدهاي بعضي ركوردها و يا بعضي فرم‌هاي بعضي ركوردها در بعضي مراحل كار تبديل به «فقط خواندني» مي‌شوند. اين جدول چنين اطلاعاتي را نگهداري مي‌نمايد. مثلاً وقتي نهاد بودجه‌گذار يك پروژه را تصويب كرد (ركورد جدول ركورد- تأييد آن ايجاد شد و تأييد بود) بلافاصله بايد فرم اصلي آن ركورد از بانك برنامه قفل شود، بسته شود و فقط امكان مشاهده آن براي نهاد پيشنهاددهنده پروژه وجود داشته باشد. ما اجازه ويرايش را عملاً با اين كار از آن نهاد مي‌گيريم. طبيعتاً اين‏گونه تغييرات زنجيره‌اي در ديتابيس را حتماً بايد از طريق ترانزاكشن انجام دهيم تا اگر يك بانك ويرايش شد و در ويرايش ساير بانك‌ها مشكل پديد آمد، رول‌بك شود خودبه‌خود. در غير اين صورت ديتابيس ناهماهنگ و به هم ريخته خواهد شد، مثلاً تصويبي صورت گرفته ولي فرم آن بسته نشده!

#  ركورد - نمايش اجباري

## كد ركورد نهاد اجباركننده

## كد ركورد نهاد اجبارشونده

## كد ركورد اجباري

## كد فرم اجباري

## ديده شده

## تاريخ ثبت اجبار

## تاريخ اعمال اجبار

اولاً اين‏كه اصلاً اين جدول كارش نمايش اجباري يك فرم يك ركورد خاصّ از يك بانك است. نمايش اجباري دقيقاً كار همان كارتابل را انجام مي‌دهد، كارتابلي كه هر كاربر هنگام ورود به سامانه در اختيار خواهد داشت. به دو گونه اين جدول پر مي‌شود؛ به صورت خودكار و به صورت دستي. وجه خودكار آن فراوان است در مواردي كه گردش كار ساخته مي‌شود. وقتي مثلاً يك نهاد پروژه‌اي را پيشنهاد مي‌دهد، يك ركورد اجبار در اين جدول ثبت مي‏‌شود كه نهاد مافوق را ملزم مي‌كند تا اين ركورد جديد را ببيند. اين ركورد در كارتابل او خواهد بود تا فلگ ديده‌شده علامت بخورد. سپس به آرشيو كارتابل منتقل مي‌گردد. دستي اين‌چنين است كه مثلاً نهاد بودجه‌گذار وقتي بخواهد توجه يك نهادي را به يك قانون مثلاً از بانك جنبي جلب نمايد، اين كار را اين‌چنين انجام مي‌دهد. موشَّح: پس در بانك اطلاعات جنبي جلوي نام هر فرم يك دگمه قرار داده مي‌شود، اگر كليك شود و يك يا چند تا از نهادهاي پايين‌دستي انتخاب گردد، آن فرم براي آن نهاد(ها) اجباراً ديده خواهد شد. مي‌توان تاريخ نمايش را هم آينده در نظر گرفت با ثبت «تاريخ اعمال اجبار». در اين صورت فرم مزبور در تاريخ ذكر شده داخل كارتابل مخاطب مي‌رود.

بودجه‌بان يك سامانه پيام‌رساني نيز لازم دارد كه ما جدولي براي آن در نظر نگرفتيم. شايد يك يا دو جدول نياز داشته باشد كه تيم فني خود مي‌توانند آن را طراحي نمايند. مي‌شود هم با ترفندهايي از همين جدول براي پيام‌رساني استفاده كرد كه وارد جزئيّات آن نمي‌شويم. تنها نكته خاصّ در سرويس پيام‌رساني بودجه‌بان اين است كه مقصد پيام تنها دو حالت مي‌تواند داشته باشد: نهاد فرادست يا نهاد فرودست. بودجه‌گذار تنها مي‌تواند براي نهادهاي رده يك و اصلي پيام ارسال نمايد. آن‌‌ها هم براي بودجه‌‌گذار و هم براي نهادهاي زيردست خود.

#  كاربر

## كد كاربر

## كد ركورد نهاد

## نام كاربري

## رمز عبور هش شده

## غيرفعال بودن

## تگ نقش در سامانه

فقط سه نقش در سامانه داريم كه دو تاي از آن‏ها ركورد مشخصي در بانك نهاد دارند. تگ نقش تعيين‌كننده نقش كاربر بوده و دسترسي‌هاي او را طبق پيش‌فرضي كه در فلوچارت بودجه‌بان نمايش داده شده مديريت مي‌كند. مدير سامانه هم كه هيچ ركوردي ندارد و فقط يك ركورد در همين جدول است. كاربرهاي بودجه‌گذار و نهادهاي رده اول توسط مدير سامانه قابل تعريف هستند. بودجه‌گذار مي‌تواند نهادهاي رده اول را كاربر نمايد. هر نهاد عملگر نيز نهادهاي زيرمجموعه خود را نام كاربري مي‌دهد. موشَّح: براي تبديل يك نهاد به يك كاربر، كافيست در صفحه «ايجاد كاربر» نام آن نهاد از فهرست انتخاب شود و نام كاربري و رمز عبور اوليه ثبت گردد. از آن لحظه آن نهاد يك كاربر سامانه محسوب مي‌گردد. اين وظيفه براي نهادهاي رده يك بر عهده مدير سامانه و يا نهاد بودجه‌گذار است و براي بودجه‌گذار تنها بر عهده مدير سامانه. ساير نهادهاي زيرمجموعه نيز توسط نهاد بالادستي خود كاربر مي‌شوند. دسترسي به ريست كردن رمز نيز به همين ترتيب مي‌باشد.

#  لاگ - ورود

## كد كاربر

## تاريخ ورود

## نشاني آي‌پي

## رشته ايجنت مرورگر

هر ورودي به سامانه در اين جدول لاگ مي‌شود. مي‌توان و بايد هم جداول ديگري داشت كه ورودهاي غيرمجاز، يا تلاش براي هك كردن را ثبت كند. SQL اينجكشن را هم وقتي تشخيص داديم، خوب است در يك جدول ديگر لاگ كنيم، تا مدير سامانه با ديدن اين اطلاعات از وضعيت سامانه مطلع شود. خوب است به جاي استفاده از كپچا، تعداد تلاش‌هاي ناموفق براي ورود محدود شود. مثلاً اگر 5 بار رمز را غلط زد، آن IP براي ده دقيقه بسته شود، يا آن نام كاربري. تشخيص و تصميم بر عهده تيم فني.

با آرزوي موفقيت براي تمامي اعضاي تيم فني سامانه بودجه‌بان، متن فوق تنها يك پيشنهاد اوليه است و خوب است اصلاحات و ويرايش خود را همين‌جا ثبت نماييد. ضمن اين‌‌كه هميشه برنامه‌نويسان در هنگام كار با نيازهاي جديدي مواجه مي‌شوند كه بايستي به صورت معقول به بانك اضافه كنند.

سيدمهدي موسوي موشَّح

10 آبان 1392