OLAP / Business Intelligence pour Publik - Development #14297

Pouvoir faire des stats sur le passage par un statut particulier

14 décembre 2016 10:59 - Frédéric Péters

Statut:	Nouveau	**Début:**	14 décembre 2016
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Par exemple j'ai des statuts "Demande acceptée" et "Demande refusée" qui convergent plus tard mais j'aimerais bien pouvoir compter le nombre de demandes acceptées et le nombre de demandes refusées; ou alors pour reprendre un autre exemple à partir du workflow attaché, je voudrais connaitre la part de paiements faits en ligne.

Demandes liées:

Lié à BiJoe - Development #29914: Pouvoir déclarer des jointures nécessaires ...	**Rejeté**	**18 janvier 2019**
Lié à w.c.s. - Development #36412: api: exposer le statut waitpoint dans le s...	**Fermé**	**24 septembre 2019**

Historique

#1 - 14 décembre 2016 11:09 - Thomas Noël

(à noter que c'est déjà possible en posant un champ de traitement, ce qui est relativement pertinent par ailleurs
)

#2 - 14 décembre 2016 11:19 - Frédéric Péters

Oui mais plutôt que devoir modifier les workflows pour marquer les points potentiels de stats l'info pouvait être générée automatiquement et systématiquement par wcs-olap (c'était du moins mon idée).

#3 - 14 décembre 2016 11:23 - Thomas Noël

Oui oui, je suis juste dans une journée pinaillage.

#4 - 14 décembre 2016 11:41 - Benjamin Dauvergne

J'exporte déjà l'historique des évolutions via wcs-olap (pour montpellier qui traite ça via BO derrière), et donc les statuts historisés, mais je ne sais pas comment je pourrai faire ce qui est demandé là avec bijoe et qui demanderait à comprendre le workflow et notamment qu'une demande ne peut pas avoir été à la fois dans le statut 'demande refusée' et 'demande acceptée'. C'est la question de savoir ce qui est additif ou pas, on ne peut grouper les stats que par des domaines additifs (ou disjoints ça dépend d'où on parle), i.e si

```
nombre(demande refusée OU demande acceptée) = nombre(demande refusée) + nombre(demande acceptée)
```

Si j'affiche les décomptes regroupés par tous les statuts dans la table des évolutions on va avoir des demandes comptés n fois... On ne pourrait vraiment utiliser cette table que pour du filtrage (ex. "quels sont les demandes qui étaient à un moment en statut 'demande acceptée' ou 'demande refusée' au mois de janvier") pour l'instant tout en groupant par statut des évolutions, mais la lecture du graphique généré doit vraiment se faire en connaissant le workflow pour voir si ça a un sens.

#5 - 21 décembre 2016 14:50 - Frédéric Péters

Un bout d'information important zappé dans la description initiale c'est que le souhait (/ un souhait proche), c'est garder l'information chronologique, pour pouvoir dire "tel jour tant de passages par 'demande à valider', tel autre jour tant de passages par 'demande finalisée', etc.").

#6 - 22 décembre 2016 13:50 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Un bout d'information important zappé dans la description initiale c'est que le souhait (/ un souhait proche), c'est garder l'information chronologique, pour pouvoir dire "tel jour tant de passages par 'demande à valider', tel autre jour tant de passages par 'demande finalisée', etc.").

C'est exactement ce que permettrait le schéma actuel:
- grouper par jour et par statut des évolutions
- filtrer sur demande à valider / demande finalisée

Possible souci expliqué plus haut mais que je reformule: pour un même jour si une demande est passée par "à valider" et "finalisée", elle sera

comptée deux fois (donc si on a eu 100 demande traitées ce jour là, la somme des deux statuts fera 101, pas 100).

#7 - 22 décembre 2016 13:55 - Frédéric Péters

- Fichier Capture d'écran 2016-12-22 à 13.53.17.png ajouté

> Possible souci expliqué plus haut mais que je reformule: pour un même jour si une demande est passée par "à valider" et "finalisée", elle sera comptée deux fois (donc si on a eu 100 demande traitées ce jour là, la somme des deux statuts fera 101, pas 100).

Oui, pas de soucis, c'est exactement ça qu'il faut. Capture d'écran attachée pour donner une idée de ce que je produis à Liège.

#8 - 18 janvier 2019 19:45 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#9 - 18 janvier 2019 20:16 - Benjamin Dauvergne

- Lié à Development #29914: Pouvoir déclarer des jointures nécessaires pour une mesure ajouté

#10 - 18 janvier 2019 23:09 - Benjamin Dauvergne

- Fichier 0002-feeder-store-delay-since-receipt_time-in-evolution-t.patch ajouté

- Fichier 0005-make-statistics-on-evolutions-fixes-14297.patch ajouté

- Fichier 0001-tests-fix-indentation.patch ajouté

- Fichier 0003-feeder-add-timetamp-version-of-receipt_time.patch ajouté

- Fichier 0004-add-measure-on-max-delay-before-status-14297.patch ajouté

- Patch proposed changé de Non à Oui

Work in progress.

#11 - 08 mars 2019 00:26 - Benjamin Dauvergne

- Fichier 0004-make-statistics-on-evolutions-fixes-14297.patch ajouté

- Fichier 0003-add-measure-on-max-delay-before-status-14297.patch ajouté

- Fichier 0002-feeder-add-timetamp-version-of-receipt_time.patch ajouté

- Fichier 0001-feeder-store-delay-since-receipt_time-in-evolution-t.patch ajouté

- Statut changé de Nouveau à Solution proposée

Rebasé.

#12 - 08 mars 2019 00:27 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#13 - 24 septembre 2019 11:53 - Benjamin Dauvergne

En gros l'idée c'est de stocker le délai entre le début d'un formulaire et n'importe quelle évolution, ensuite ça peut-être utiliser par #29914 pour faire une mesure sur une colonne de la table evolution (la colonne délai) en filtrant/groupant par statut ; mais la jointure doit être faite adéquatement, comme on veut qu'un délai par statut par exemple faudra garder que la première/dernière évolution pour chaque formdata (et ça c'est un boulot pour #29914 de voir comment on peut faire, donner des exemples de SQL pour cela).

Pour les simples décompte ça va marcher tout seul, modulor la remarque plus haut que j'ai faite en #14297#note-6:

> Possible souci expliqué plus haut mais que je reformule: pour un même jour si une demande est passée par "à valider" et "finalisée", elle sera comptée deux fois (donc si on a eu 100 demande traitées ce jour là, la somme des deux statuts fera 101, pas 100).

#14 - 24 septembre 2019 15:28 - Benjamin Dauvergne

- Fichier 0002-tox.ini-use-environment-tag-to-fix-python-interprete.patch ajouté

- Fichier 0007-tests-wip-add-tests-with-bijoe.patch ajouté

- Fichier 0003-feeder-store-delay-since-receipt_time-in-evolution-t.patch ajouté

- Fichier 0005-add-measure-on-max-delay-before-status-14297.patch ajouté

- Fichier 0001-tests-add-support-for-pg_virtualenv.patch ajouté

- Fichier 0006-make-statistics-on-evolutions-fixes-14297.patch ajouté

- Fichier 0004-feeder-add-timetamp-version-of-receipt_time.patch ajouté

- Statut changé de En cours à Solution proposée

rebasé.

#15 - 24 septembre 2019 15:50 - Benjamin Dauvergne

Pour le calcul du délai unique par statut pour un formdata je propose d'ajouter les deux vues suivantes pour chaque table d'évolution :

- délai au premier passage

```
# view evolution_<slug>_first_time
SELECT distinct on (formdata_id, status_id) * FROM evolution ORDER BY formdata_id, status_id, time
```

- délai au dernier passage

```
# view evolution_<slug>_last_time
SELECT distinct on (formdata_id, status_id) * FROM evolution ORDER BY formdata_id, status_id, time DESC
```

ou alors changer complètement le chargement de la table d'evolution pour ne garder que les evolutions si le statut n'a pas encore été vu (uniquement premier passage).

#16 - 24 septembre 2019 15:53 - Frédéric Péters

Tant qu'à être là-dessus, c'est peut-être important de noter dès maintenant que souvent on ne sera pas intéressé par le temps entre le premier statut et un autre, parce que le premier statut sera un truc genre "attente de paiement", que la vraie question portera sur le temps de traitement par l'agent, qui commence plus loin.

Du coup en stat parallèle, par statut, avoir le temps du statut vers le premier statut final rencontré. ?

#17 - 24 septembre 2019 16:25 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Tant qu'à être là-dessus, c'est peut-être important de noter dès maintenant que souvent on ne sera pas intéressé par le temps entre le premier statut et un autre, parce que le premier statut sera un truc genre "attente de paiement", que la vraie question portera sur le temps de traitement par l'agent, qui commence plus loin.
>
> Du coup en stat parallèle, par statut, avoir le temps du statut vers le premier statut final rencontré. ?

Est-ce qu'il ne faudrait pour se faire exporter le statut "waitpoint" d'un "statut" dans le schéma ?

Et donc ça donnerait delay_from_first_waitpoint / delay_to_first_endpoint :

- construire evolution en gardant qu'une ligne par statut (la première fois qu'on y arrive)
- pour chaque ligne calculer le délai depuis le premier waitpoint (delay_from_first_waitpoint) et jusqu'au premier statut terminal (delay_to_first_endpoint)

#18 - 24 septembre 2019 16:45 - Frédéric Péters

> > Du coup en stat parallèle, par statut, avoir le temps du statut vers le premier statut final rencontré. ?
>
> Est-ce qu'il ne faudrait pour se faire exporter le statut "waitpoint" d'un "statut" dans le schéma ?

Peut-être qu'on peut en effet uniquement avoir des infos sur les (temps de) trajets entre certains statuts. (au total donc début, waitpoints, endpoints)

> Et donc ça donnerait delay_from_first_waitpoint / delay_to_first_endpoint :

Parce que l'attente du paiement de l'usager c'est un waitpoint, à mon avis first_waitpoint ça ne tient pas.

#19 - 24 septembre 2019 17:09 - Benjamin Dauvergne

- Lié à Development #36412: api: exposer le statut waitpoint dans le schéma du workflow ajouté

#20 - 25 novembre 2019 10:24 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#21 - 21 janvier 2022 13:01 - Benjamin Dauvergne

- Statut changé de En cours à Nouveau

- Assigné à Benjamin Dauvergne supprimé

Je me dés-assigne sans fermer, je note qu'on est jamais arrivé à bien définir de façon automatique les temps à mesurer, le mieux qu'on sache faire c'est du premier waitpoint (mais à quelques millisecondes prêt on devrait arriver au premier waitpoint juste après receipt_time, ça n'apporte pas grand chose) au premier endpoint. Il serait vraiment nécessaire d'avoir une notion d'étapes un peu linéaire dans les statuts (savoir quand on rentre dans la zone de traitement et quand on en sort, typiquement sur une démarche où le paiement doit avoir lieu avant le traitement, on ne veut rien décompter avant le paiement).

Fichiers

Capture du 2016-12-14 10-58-45.png	32,2 ko	14 décembre 2016	Frédéric Péters
Capture d'écran 2016-12-22 à 13.53.17.png	385 ko	22 décembre 2016	Frédéric Péters
0002-feeder-store-delay-since-receipt_time-in-evolution-t.patch	6,22 ko	18 janvier 2019	Benjamin Dauvergne
0005-make-statistics-on-evolutions-fixes-14297.patch	4,02 ko	18 janvier 2019	Benjamin Dauvergne
0001-tests-fix-indentation.patch	1,7 ko	18 janvier 2019	Benjamin Dauvergne
0003-feeder-add-timetamp-version-of-receipt_time.patch	2,13 ko	18 janvier 2019	Benjamin Dauvergne
0004-add-measure-on-max-delay-before-status-14297.patch	5,4 ko	18 janvier 2019	Benjamin Dauvergne
0004-make-statistics-on-evolutions-fixes-14297.patch	4,02 ko	07 mars 2019	Benjamin Dauvergne
0003-add-measure-on-max-delay-before-status-14297.patch	5,58 ko	07 mars 2019	Benjamin Dauvergne
0002-feeder-add-timetamp-version-of-receipt_time.patch	2,24 ko	07 mars 2019	Benjamin Dauvergne
0001-feeder-store-delay-since-receipt_time-in-evolution-t.patch	6,27 ko	07 mars 2019	Benjamin Dauvergne
0002-tox.ini-use-environment-tag-to-fix-python-interprete.patch	736 octets	24 septembre 2019	Benjamin Dauvergne
0007-tests-wip-add-tests-with-bijoe.patch	4,99 ko	24 septembre 2019	Benjamin Dauvergne
0003-feeder-store-delay-since-receipt_time-in-evolution-t.patch	6,28 ko	24 septembre 2019	Benjamin Dauvergne
0005-add-measure-on-max-delay-before-status-14297.patch	5,58 ko	24 septembre 2019	Benjamin Dauvergne
0001-tests-add-support-for-pg_virtualenv.patch	3,13 ko	24 septembre 2019	Benjamin Dauvergne
0006-make-statistics-on-evolutions-fixes-14297.patch	4,02 ko	24 septembre 2019	Benjamin Dauvergne
0004-feeder-add-timetamp-version-of-receipt_time.patch	2,24 ko	24 septembre 2019	Benjamin Dauvergne