

Le but de ce document est de donner une définition formelle des fonctions dont est composé le langage C2QL.

## Préambule

**Définition 1** Ici, pour simplifier, on appelle chaîne de caractères tout mot sur l'alphabet

$$\Sigma = \{a, \dots, z\} \cup \{A, \dots, Z\} \cup \{0, \dots, 9\}$$

**Définition 2** On appelle nom d'attribut toute chaîne de caractères.

**Définition 3** On appelle schéma relationnel tout ensemble de noms d'attributs.

## Définitions générales

Soit  $\mathcal{V}$  un ensemble, appelé ensemble des valeurs.

**Définition 4** On appelle relation de schéma relationnel  $\Delta$  un ensemble de fonctions de  $\Delta \cup \{id\}$  dans  $\mathcal{V}$ .

Chacun des éléments de la relation (chacune de ces fonctions) est appelé(e) ligne.

Pour chaque ligne  $l$  de la relation et chaque  $\alpha$  de  $\Delta$ ,  $l(\alpha)$  est appelé attribut de nom  $\alpha$  pour la ligne  $l$ .

L'image de  $id$  est appelée identifiant de la ligne, et elle est, au sein de chaque relation, unique pour chaque ligne.

**Définition 5** On appelle  $S$  l'ensemble des schémas relationnels possibles. Autrement dit, on pose  $S = \mathcal{P}(\Sigma^*)$ .

On appelle  $R$  l'ensemble des relations possibles,

et on introduit la fonction  $\text{sch}$  de  $R$  dans  $S$  qui à une relation associe son schéma relationnel.

## Projections et sélections

**Définition 6** Pour tout ensemble  $\delta$  de noms d'attributs, on appelle projection sur les attributs  $\delta$  la fonction suivante :

$$\begin{aligned} \pi_\delta : R &\rightarrow R \\ r &\mapsto \{l|_{(\delta \cap \text{sch}(r)) \cup \{id\}} / l \in r\} \end{aligned}$$

**Définition 7** On appelle  $L$  l'ensemble de toutes les lignes possibles.

On appelle prédicat toute fonction de  $L$  dans  $\{true, false\}$ .

On appelle domaine d'un prédicat  $p$  le plus petit ensemble  $D$  tel que :

$$\forall (l, l') \in L^2, (l|_D = l'|_D \Rightarrow p(l) = p(l'))$$

et on le note  $\text{dom}(p)$ .

**Définition 8** On appelle sélection de prédicat  $p$ , pour tout prédicat  $p$ , la fonction :

$$\begin{aligned} \sigma_p : R &\rightarrow R \\ r &\mapsto r \cap p^{-1}(\{true\}) \end{aligned}$$

## Jointure naturelle

**Définition 9** On dit qu'une paire de relations  $(r, r')$  est joignable si on a :

$$\forall l \in r \quad \exists l' \in r' \quad \forall \alpha \in \text{sch}(r) \cap \text{sch}(r'), \quad l(\alpha) = l'(\alpha)$$

Si  $(r, r')$  est une paire de relations joignables, et que  $l$  est une ligne de  $r$ , on appelle correspondants de  $l$  dans  $r'$  l'ensemble des lignes  $l'$  de la propriété précédente. On note  $\text{cor}_{r,r'}(l)$  l'ensemble de ces lignes-là.

**Définition 10** Si  $l$  et  $l'$  sont deux lignes correspondantes, on appelle concaténation de  $l$  et de  $l'$ , notée  $l.l'$  la fonction de  $\text{sch}(l) \cup \text{sch}(l') \cup \{id\}$  définie par :

$$\begin{cases} l.l'(\alpha) = l(\alpha) & \text{si } \alpha \in \text{sch}(r) \setminus \text{sch}(r') \\ l.l'(\alpha) = l'(\alpha) & \text{si } \alpha \in \text{sch}(r') \setminus \text{sch}(r) \\ l.l'(\alpha) = l(\alpha) = l'(\alpha) & \text{si } \alpha \in \text{sch}(r) \cap \text{sch}(r') \\ l.l'(id) = l(id).l'(id) \end{cases}$$

où  $l(id).l'(id)$  est compris comme la concaténation de listes (les valeurs qui ne sont pas des listes sont assimilées à des listes avec un seul élément).

**Définition 11** Pour  $r$  et  $r'$  deux relations joignables, on appelle jointure naturelle de  $r$  et  $r'$  la table

$$r \bowtie r' = \{l.l' / l \in r, l' \in \text{cor}_{r,r'}(l)\}$$

On utilisera aussi la notation préfixe. En effet, si on appelle  $\text{Rj}$  l'ensemble des paires de relations joignables, on vient de définir la fonction

$$\begin{array}{ccc} \bowtie: & \text{Rj} & \rightarrow \text{R} \\ & (r, r') & \mapsto r \bowtie r' \end{array}$$

## Fragmentation et défragmentation

La défragmentation est un cas particulier de jointure naturelle, le seul attribut en commun pour les deux tables est l'identifiant des lignes.

**Définition 12** Deux relations  $r$  et  $r'$  sont dites unifiables si :

$$\begin{cases} \{l(id)/l \in r\} = \{l(id)/l \in r'\} \\ \text{sch}(r) \cap \text{sch}(r') = \emptyset \end{cases}$$

On remarquera que deux relations unifiables sont également joignables.

On note  $\text{Ru}$  l'ensemble des paires de relations unifiables, qui est donc un sous-ensemble de  $\text{Rj}$ , qui lui même est un sous-ensemble de  $\text{R}^2$ .

**Définition 13** Pour tout ensemble de noms d'attributs  $\delta$  on appelle fragmentation de fragment gauche  $\delta$  l'application suivante :

$$\begin{array}{ccc} \text{frag}_\delta & \text{R} & \rightarrow \text{Ru} \\ & r & \mapsto (\{l|_{(\text{sch}(r) \cap \delta) \cup \{id\}} / l \in r\}, \{l|_{(\text{sch}(r) \setminus \delta) \cup \{id\}} / l \in r\}) \end{array}$$

**Définition 14** On appelle défragmentation la restriction de la jointure naturelle à  $\text{Ru}$ .

$$\begin{array}{ccc} \text{defrag} : & \text{Ru} & \rightarrow \text{R} \\ & (r, r') & \mapsto r \bowtie r' \end{array}$$

## Chiffrement et déchiffrement

Vu que pour l'instant on s'intéresse uniquement aux contenus des tables pour démontrer la correction sémantique des lois de composition, on ne parlera pas pour l'instant des éventuelles clefs de chiffrement et déchiffrement.

**Définition 15** On appelle chiffrement tout couple  $c$  de fonctions  $(\text{Enc}, \text{Dec})$  de  $\mathcal{V}$  dans  $\mathcal{V}$  vérifiant  $\text{Dec} \circ \text{Enc} = \text{id}$ .

Pour toute valeur  $v$  de  $\mathcal{V}$  on note  $c(v) = \text{Enc}(v)$  et  $c^{-1}(v) = \text{Dec}(v)$

**Définition 16** Pour une ligne  $l$  définie sur  $\Delta$ , pour  $\alpha$  un attribut, et pour  $c$  un chiffrement, on appelle version de  $l$  chiffrée pour  $\alpha$  avec le chiffrement  $c$  la ligne notée  $c(l)_\alpha$  définie par :

$$\begin{cases} \forall \beta \in \Delta \setminus \{\alpha\} & c(l)_\alpha(\beta) = l(\beta) \\ & c(l)_\alpha(\alpha) = c(l(\alpha)) \quad \text{si } \alpha \in \Delta \end{cases}$$

De même, on définit la version de  $l$  déchiffrée pour  $\alpha$  avec le chiffrement  $c$ , notée  $c^{-1}(l)_\alpha$ , par :

$$\begin{cases} \forall \beta \in \Delta \setminus \{\alpha\} & c^{-1}(l)_\alpha(\beta) = l(\beta) \\ & c^{-1}(l)_\alpha(\alpha) = c^{-1}(l(\alpha)) \quad \text{si } \alpha \in \Delta \end{cases}$$

**Définition 17** Pour  $\alpha$  un nom d'attribut et  $c$  un chiffrement, on appelle fonction de chiffrement de  $\alpha$  par  $c$  la fonction

$$\begin{aligned} \text{crypt}_{\alpha, c} : \quad \mathbf{R} &\rightarrow \mathbf{R} \\ r &\mapsto \{c(l)_\alpha / l \in r\} \end{aligned}$$

De même, on appelle fonction de déchiffrement de  $\alpha$  par  $c$  la fonction

$$\begin{aligned} \text{decrypt}_{\alpha, c} : \quad \mathbf{R} &\rightarrow \mathbf{R} \\ r &\mapsto \{c^{-1}(l)_\alpha / l \in r\} \end{aligned}$$

## Agrégation

**Définition 18** Pour  $\delta$  un ensemble de noms d'attributs, on appelle nom de groupe pour  $\delta$  toute application  $n$  définie de  $\delta$  dans  $\mathcal{V}$ .

On remarque que tout nom de groupe est une ligne.

$\delta$  est appelé domaine du nom de groupe  $n$ , et noté  $\text{dom}(n)$ .

De plus, pour  $r$  une relation, on définit l'ensemble des noms de groupe de  $r$  pour  $\delta$  :

$$r_\delta = \{l|_\delta / l \in r\}$$

**Définition 19** Pour  $r$  une relation et  $n$  un groupe, on appelle groupe de  $r$  pour le nom  $n$  l'ensemble des éléments de  $r$  coïncidant avec  $n$  sur  $\text{sch}(r) \cap \text{dom}(n)$ . On le note  $r_n$ .

Autrement dit :

$$r_n = \{l \in r / l|_{\text{sch}(r) \cap \text{dom}(n)} = n|_{\text{sch}(r) \cap \text{dom}(n)}\}$$

De plus, on appelle identifiants du groupe  $r_n$  l'ensemble des identifiants des lignes du groupe.

On note  $\text{IDs}(r_n)$  cet ensemble.

Autrement dit :

$$\text{IDs}(r_n) = \{l(\text{id}) / l \in r_n\}$$

**Définition 20** On dira qu'une application  $f$  est plus petite qu'une application  $g$  si  $f$  est une restriction de  $g$ .

On dira qu'un nom de groupe  $n_0$  est minimal pour une relation  $r$  donnée si c'est une plus petite application  $n$  pour laquelle le groupe de  $r$  pour  $n$  vaut  $r_{n_0}$ .

**Définition 21** Pour  $r$  une relation,  $n$  un nom de groupe, et  $\alpha$  un attribut de  $(\text{sch}(r) \setminus \text{dom}(r)) \cup \{id\}$ , on appelle valeurs du groupe  $r_n$  pour l'attribut  $\alpha$  la fonction

$$\begin{aligned} r_n(\alpha) : \quad \text{IDs}(r_n) &\rightarrow \mathcal{V} \\ l(id) &\mapsto l(\alpha) \end{aligned}$$

**Remarque :** Souvent, on supposera que l'ensemble des identifiants possible est totalement ordonné et on s'en servira pour considérer des fonctions définies sur un ensemble d'identifiants (par exemple les  $r_n(\alpha)$  définis ci-dessus) comme des listes.

On définit la longueur de telles listes comme le cardinal de leur ensemble de départ. Par exemple, la longueur de  $r_n(\alpha)$  est  $|r_n(\alpha)| = |\text{IDs}(r_n)|$

**Définition 22** Pour  $r$  une relation, et  $n$  un nom de groupe, on appelle ligne de groupe de  $r$  pour  $n$  la ligne notée  $\text{lg}_{r,n}$  définie sur  $\text{sch}(r) \cup \{id\}$  par :

$$\begin{cases} \text{lg}_{r,n}(\alpha) = n(\alpha) & \text{si } \alpha \in \text{sch}(r) \cap \text{dom}(n) \\ \text{lg}_{r,n}(\alpha) = r_n(\alpha) & \text{si } \alpha \in (\text{sch}(r) \setminus \text{dom}(n)) \cup \{id\} \end{cases}$$

**Définition 23** Pour  $\delta$  un ensemble de noms d'attributs, on appelle fonction d'agrégation pour les attributs  $\delta$  la fonction suivante :

$$\begin{aligned} \text{group}_\delta : \quad \mathbf{R} &\rightarrow \mathbf{R} \\ r &\mapsto \{\text{lg}_{r,n} / n \in r_\delta\} \end{aligned}$$

## Réduction

La plupart du temps, les agrégations sont faites pour pouvoir faire une réduction ensuite.

On suppose que les identifiants des lignes peuvent être totalement ordonnés et donc que les fonctions définies sur des ensembles d'identifiants peuvent être vues comme des listes.

Pour toute liste  $l$  on notera  $\text{hd}(l)$  le premier élément de la liste, et  $\text{tl}(l)$  le reste de la liste.

Dans les définitions qui suivent,  $f$  est une fonction de  $\mathcal{V}^2$  dans  $\mathcal{V}$  et  $z$  est un élément de  $\mathcal{V}$ .

**Définition 24** On appelle réduction d'une liste  $t$  par la fonction  $f$  avec l'élément neutre  $z$  la valeur  $\text{red}_{f,z}(t)$  définie par induction sur la liste par :

$$\begin{cases} \text{red}_{f,z}(\emptyset) = z \\ \text{red}_{f,z}(t) = \text{red}_{f,f(z,\text{hd}(t))}(\text{tl}(t)) \end{cases}$$

Si une valeur  $v$  de  $\mathcal{V}$  n'est pas une liste, on la considère alors comme une liste à un seul élément et on pose donc  $\text{red}_{f,z}(v) = f(z, v)$ .

**Définition 25** Pour  $l$  une ligne définie sur  $\delta$ , et  $\alpha$  un nom d'attribut, on appelle réduction de l'attribut  $\alpha$  dans la ligne  $l$  par la fonction  $f$  avec l'élément neutre  $z$  la ligne  $\text{red}_{\alpha,f,z,l}$  définie sur  $\delta$  par :

$$\begin{cases} \text{red}_{\alpha,f,z,l}(\alpha) = \text{red}_{f,z}(l(\alpha)) \\ \text{red}_{\alpha,f,z,l}(\beta) = l(\beta) & \text{si } \beta \neq \alpha \end{cases}$$

**Définition 26** On appelle fonction de réduction de l'attribut  $\alpha$  par la fonction  $f$  avec l'élément neutre  $z$  la fonction suivante :

$$\begin{aligned} \text{fold}_{\alpha,f,z} : \quad \mathbf{R} &\rightarrow \mathbf{R} \\ r &\mapsto \{\text{red}_{\alpha,f,z,l} / l \in r\} \end{aligned}$$

## Opérations ensemblistes : union, différence, fragmentation horizontale

**Définition 27** On appelle union ou défragmentation horizontale de deux tables  $r$  et  $r'$  ayant le même schéma relationnel la table  $r \cup r'$ , aussi notée  $\text{hdefrag}(r, r')$ .

**Définition 28** On appelle différence ensembliste de deux tables  $r$  et  $r'$  ayant le même schéma relationnel la table  $r \setminus r'$ .

**Définition 29** Pour  $p$  un prédicat, on appelle fragmentation horizontale de critère  $p$  la fonction

$$\begin{array}{ll} \text{hfrag} : & \mathbf{R} \rightarrow \mathbf{R}^2 \\ r & \mapsto (\{l \in r/p(l)\}, \{l \in r, \neg p(l)\}) \end{array}$$