

IFT1155 — Démonstration 4

Guillaume Poirier-Morency
guillaumepoiriermorency@gmail.com

- ▶ services
- ▶ base de donnée (concepts fondamentaux)

Services

On va migrer le code pour peupler la base de données dans un service.

Base de données relationnelle

- ▶ spécifiée par un schéma
- ▶ organise les données en tables
- ▶ gère relations entre les tables
- ▶ très complexes (beaucoup de fonctionnalités)

SQLite

- ▶ stockage dans un fichier
- ▶ transactionnel

Table

Une table est une structure de données qui représente des données (lignes) et leurs attributs (colonnes).

Le langage SQL permet de manipuler les lignes et les colonnes d'un ensemble de tables.

Table 1: Table contenant les données des utilisateurs

id	username	password
1	bob	1234
2	alice	4567
3	charlie	7890

Tuple

Un tuple est un ensemble ordonné et fini d'éléments.

```
(1, 2, 3)
```

```
("chat", "chien", 6)
```

```
(78.90, "text", 14, 0x002323)
```

```
(1) # singleton
```

Une ligne dans une table constitue un tuple.

Une requête SQL permet de manipuler ou des sélectionner des tuples.

Types

Il existe quatre types de données:

- ▶ NULL
- ▶ INTEGER: nombres entiers
- ▶ REAL: nombres à points flottants
- ▶ TEXT: données texte encodé en UTF-8
- ▶ BLOB: données binaires

Opérations

- ▶ sélection
- ▶ insertion
- ▶ mise à jour
- ▶ destruction
- ▶ manipulation de la structure (schéma)

Exemples

```
select * from users where username = "bob";
```

```
insert into users (username, password) values ("bob", "1234
```

```
update users set password = "4567" where username = "bob";
```

```
delete from users where username = "bob";
```

```
create table users (  
    username text,  
    password text  
);
```

Prédicats

Les prédicats (mot-clé `where`) permet d'indiquer la condition que les éléments d'un tuple doivent respecter afin que l'opération (`select`, `update`, `delete`) soit applicable.

- ▶ `between`
- ▶ `is`
- ▶ `and`, `or`, `not`
- ▶ `in`

```
select * from users where username in ("bob", "alice");
```

Sélection

L'opération de sélection est la plus complexe. Elle permet d'effectuer les traitements suivants:

- ▶ joindre deux tables
- ▶ grouper
- ▶ trier

Joindre

L'opération de jointure (produit cartésien) permet d'inclure les données d'une autre table dans une sélection.

L'inclusion peut être contrôlée par un prédicat (mot-clé `on`).

L'algorithme (type `inner`):

Pour chaque tuple dans la sélection et dans la jointure, inclure la concaténation si la condition est respectée.

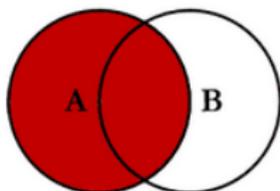
```
select * from users
  join cars on users.id = cars.owner_id;
```

Types de jointure

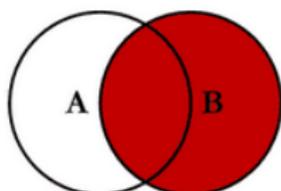
Il existe d'autres types de jointure

- ▶ left: inclu toujours les données initiale
- ▶ right: inclu toujours les données jointes
- ▶ inner: inclu les données communes (intersection)
- ▶ full: inclu toutes les données (union)
- ▶ cross: aucune condition

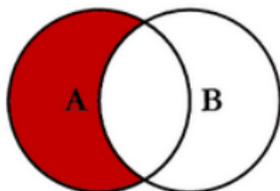
SQL JOINS



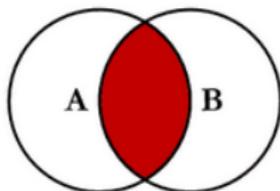
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



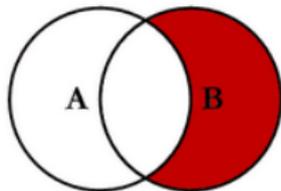
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



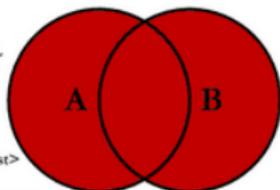
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



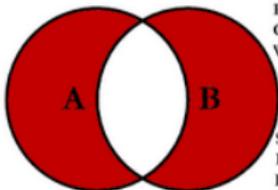
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

Figure 1: Types de jointure en SQL

Grouper

L'opération consiste à classer un ensemble de tuples selon un critère et d'appliquer le traitement sur chaque classes.

```
select sex, avg(hour_pay) from workers group by sex;
```

```
("men", 12.75),
```

```
("women", 10.85)
```

On peut filtrer le résultat d'un groupage avec le mot-clé `having`.

```
select sex, from workers  
group by sex having avg(hour_pay) > 12;
```

```
("men")
```

Tri

On peut trier selon une ou plusieurs colonnes, de manière ascendante ou descendante.

```
select * from users order by login_attempts desc;
```