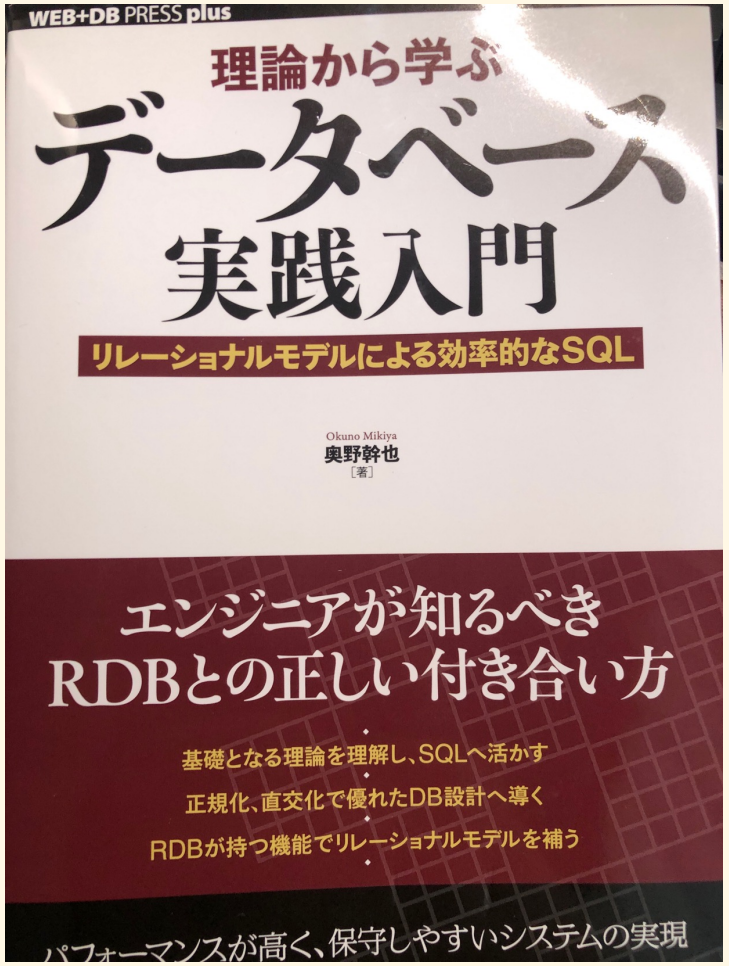
Expert GUI
データベース開発

今井ひかる

2022/12/5

概要

- データベース設計について再度学習し、FPHXパラメタなどのExpertGUIに関係があるデータベースについて再度検討した。



どんなデータベースを作るべきか

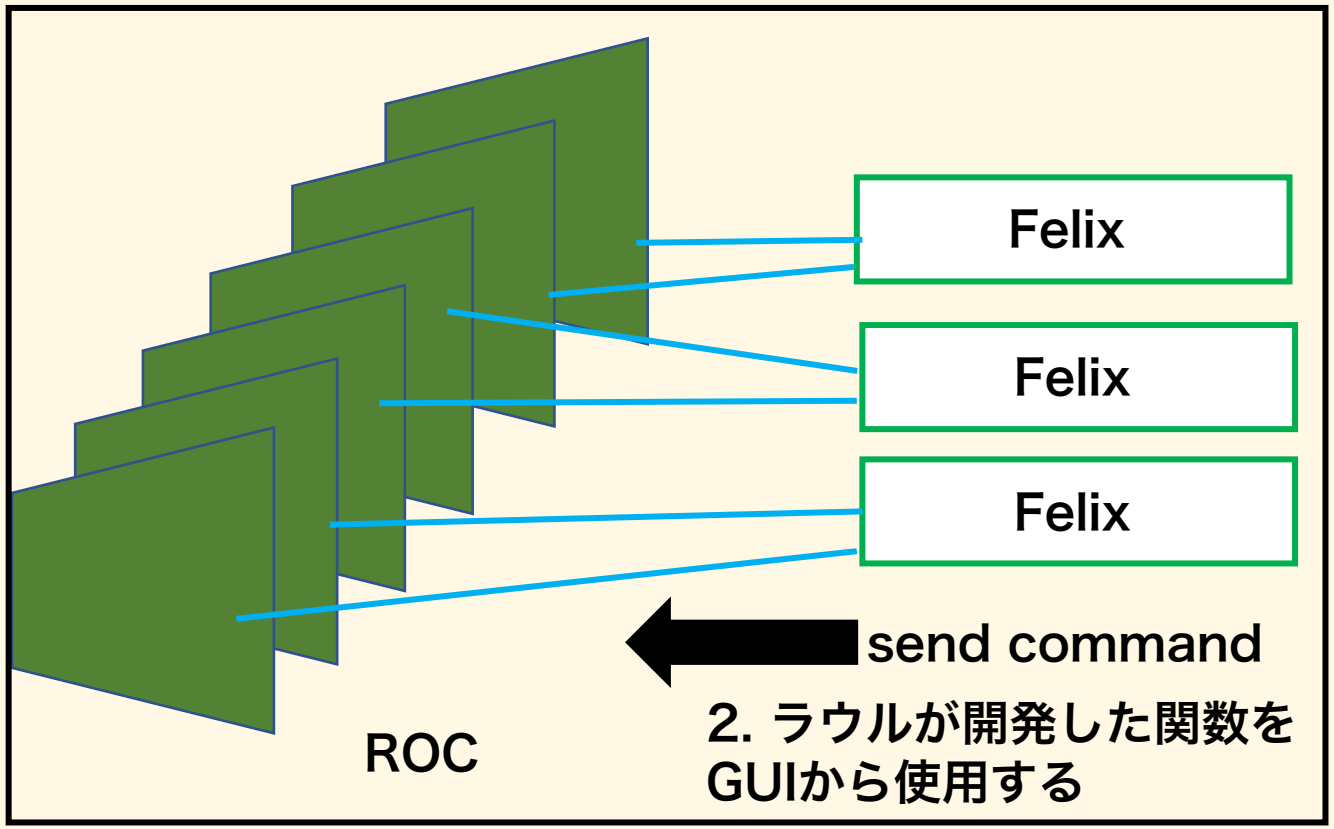
ソフトウェア（日々変化する可能性がある）

- 1. DAC値、LVDSなどの FPHX chip パラメータの値。
- 2. チャンネルのマスク状況。

ハードウェア（月単位？で変化するもの）

- 3. ラダーとROCの関係。ラダーがどのROCのどのPortに挿さっているか。
- 4. FelixとROCの関係。ROCがどのFelixと繋がっているか。

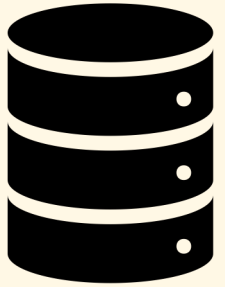
やるべきこと



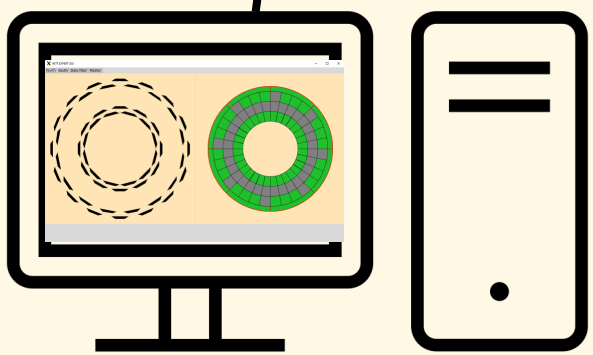
4.　外部のPCから
複数のFelixを操作する

5.ドキュメントを書く

データベース



3.パラメタを保存する
データベースを開発



1.フロントエンドを完成させる

データベースについて

- データベース　→ 関連する情報を集めたもの。

- リレーショナルデータベース　→ データをテーブルの集合として管理

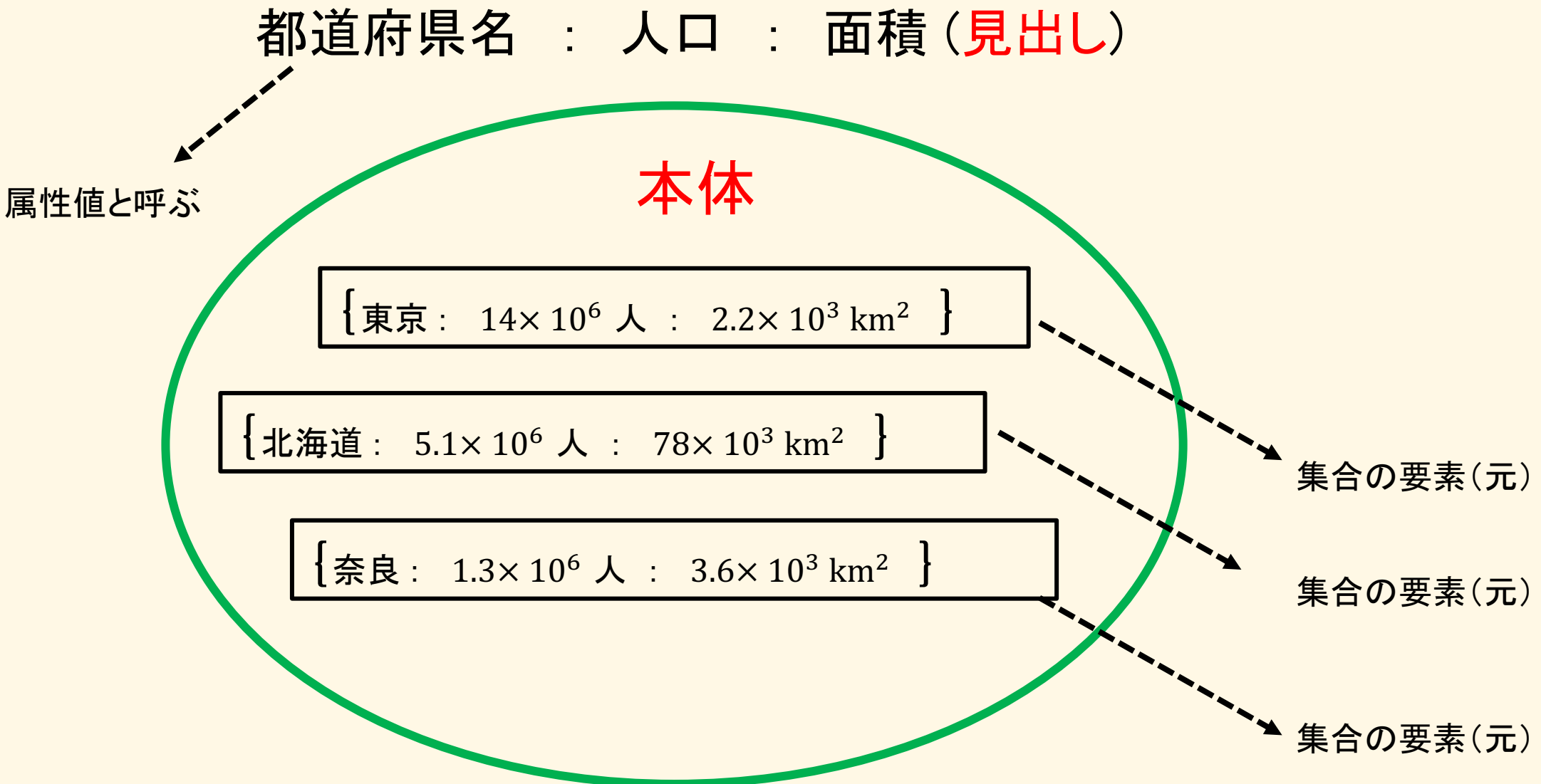
- リレーショナルデータベースを管理するシステムがある。
  これは様々な製品ある。(MySQL , PostgreSQL , Oracle)

- SQL → リレーショナルデータベースを操作するための言語。

リレーションとは？

リレーションは見出しと本体のペアで構成される。

本体は、属性値の組（タプル）を要素とする集合である。



リレーションとは？

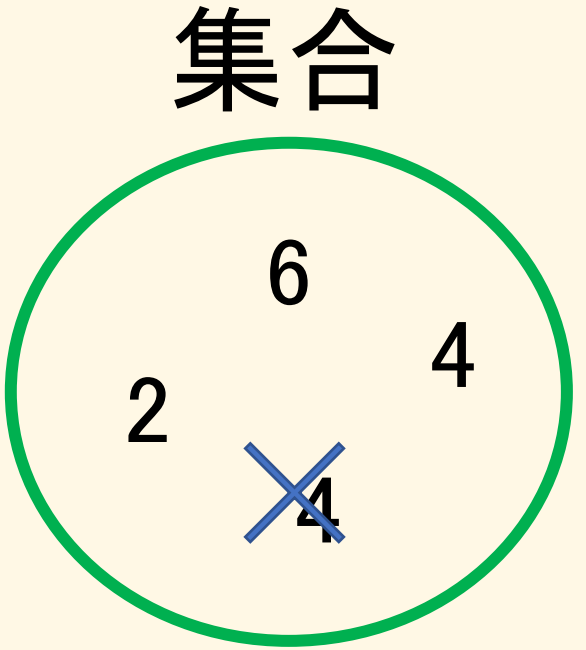
リレーションは見出しと本体のペアで構成される。

本体は、属性値の組（タプル）を要素とする集合である。

集合の注意点

1. 重複があってはならない。

2. 要素間に順番は無い。配列とは違う。



int a[3] = { 2 , 6, 4 };

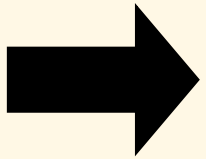
リレーションの設計～正規化と直交化～

- リレーションの見出しは適当に作ってよいか？
  → 適当に作ると重複を誘発してしまう。
  → 重複が発生すると矛盾が生じる！
  → 意味のないDB

このテーブルにある今井さんの年齢はどっちが正しいの？

名前	所属機関	年齢
今井	立教	23
今井	理研	23

update



名前	所属機関	年齢
今井	立教	23
今井	理研	24

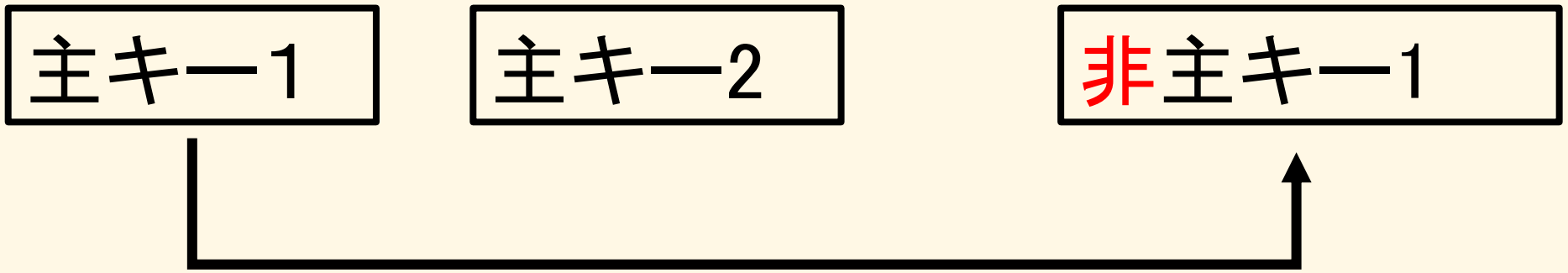
- 重複を発生させない、矛盾を起こさないために、リレーションを正規化する必要がある。

- 正規化には手順がある。それ通りに正規化していく。
- 直交化は、リレーション同士(テーブル同士)が重複しないようにすること。

リレーションの正規化

1. プライマリーキーと非プライマリーキーの関係に着目する。

   プライマリーキーが決まった時に非プライマリーキーが決定するかどうか考える。もしあった場合はリレーションを分ける。



例

名前（主キー）	所属機関（主キー）	年齢（非主キー）
今井	立教	23
今井	理研	23

名前（主キー）	所属機関（主キー）
今井	立教
今井	理研

名前（主キー）	年齢（非主キー）
今井	23

その時点では、名前（今井）が決まれば、年齢も決まります。
この関係を　名前　→　年齢　と表記することがある。

データベース再考1

主　主　主

ladder position	direction	chip id	DAC0	DAC1	LVDS
B0L000	North	1	20	25	3
B0L000	North	2	20	25	4

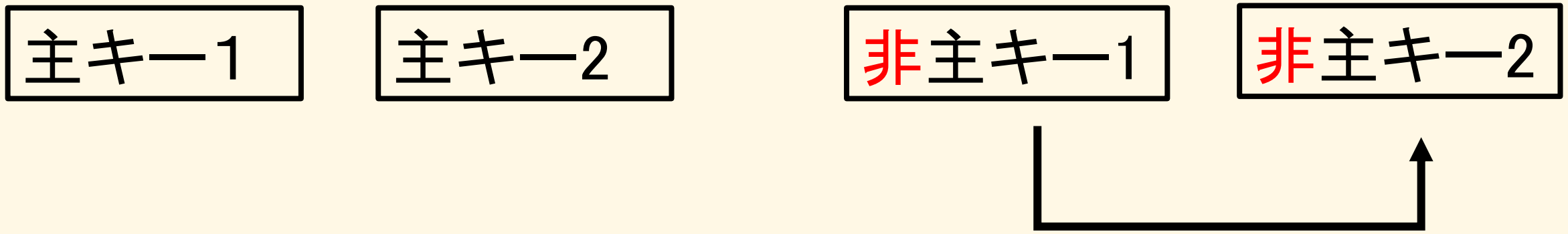
主　主　主

ladder position	direction	chip id	mask status
B0L000	North	1	11111111111111111111111
B0L000	North	2	01111111111111111111111

リレーションの正規化

2.非プライマリーキーの関係同士に着目する。

ある非プライマリーキーが決定したとき、別の非プライマリーキーが決定するかどうか考える。もしあった場合はリレーションを分ける。

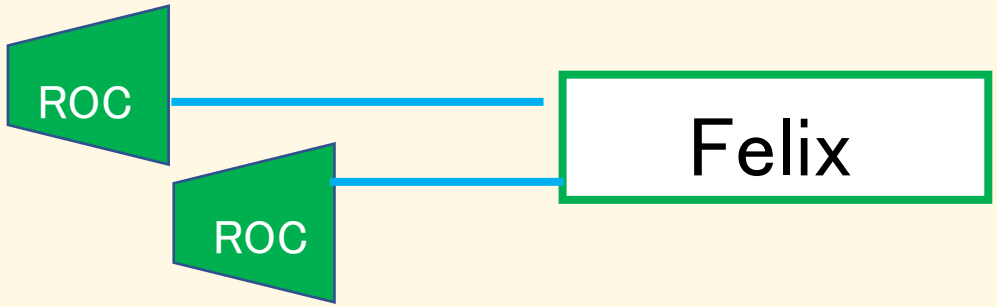


データベース再考2

主　主

ladder name	direction	ROC	column	port (module)	Felix
B0L000	N	NE0	A	1	Felix1
B0L001	N	NE0	B	2	Felix1

非プライマリキー同士で、ROC → Felix　という関係がある。



ladder name	direction	ROC	column	port (module)	Felix
B0L000	N	NE0	A	1	Fe l
B0L001	N	NE0	B	2	F lix

ROC	Felix
NE0	Felix1
NE1	Felix1

テーブルを分けて、外部キーで紐付けする。

履歴（時間）データの扱い方

- データには、ある時間、日時と合わせて管理したい場合が当然ある。
  → 更新時間、購入時間、返却締め切り（未来）

- 実は履歴（時間）データとリレーショナルデータベースは相性が悪い。

集合(リレーション)の注意点

1. 重複があってはならない。
2. 要素間に順番は無い。配列とは違う。

集合…?

2022/12/01 AM9:00

2022/12/01 AM11:00

2022/12/04 PM 7:00

- 履歴（時間）データを扱う正しい、正解な方法は無い。しかし、うまくやりすごす方法はある。

避けたいデータベースデザイン

start-time	Ladder position	direction	direction	DAC0
2022/11/30　9:00	B0L000	N	1	15
2022/11/30　9:00	B0L000	N	2	15
2022/12/02　14:00	B0L000	N	1	20

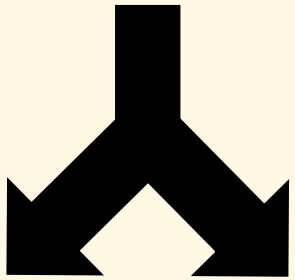
2022/12/02に北側にあるB0L000ラダー、chip1のDAC0の値を15→20に変更した状況

- このテーブルには過去の状態と現在の状態が暗黙的に混ざっている。
- つまり、1つ1つの要素の意味が異なっている。（均一では無い。）
- これは避けるべき

おすすめ？の履歴データベース

- １番簡潔に解決する方法は、現在と過去の情報を分けてテーブルに保存すること。

start-time	Ladder position	chip_id	direction	DAC0
2022/11/30 9:00	B0L000	1	N	15
2022/11/30 9:00	B0L000	2	N	15
2022/12/02 14:00	B0L000	1	N	20



start-time	Ladder position	chip_id	direction	DAC0
2022/12/02 14:00	B0L000	1	N	20
2022/12/02 9:00	B0L000	2	N	15

present

start-time	Ladder position	chip_id	direction	DAC0
2022/11/30 9:00	B0L000	1	N	15

history

おすすめ？の履歴データベース

start-time	Ladder position	chip_id	direction	DAC0
2022/12/02 9:00	B0L000	1	N	15
2022/12/02 9:00	B0L000	2	N	15
⋮				
2022/12/02 9:00	B1L116	25	S	15
2022/12/02 9:00	B1L116	26	S	15

present

start-time	Ladder position	chip_id	direction	DAC0

history

おすすめ？の履歴データベース

start-time	Ladder position	chip_id	direction	DAC0
2022/12/02 9:00	B0L000	1	N	15
2022/12/02 9:00	B0L000	2	N	15
⋮				
2022/12/02 9:00	B1L116	25	S	15
2022/12/02 9:00	B1L116	26	S	15

present

start-time	Ladder position	chip_id	direction	DAC0
2022/12/02 9:00	B0L000	2	N	15

history

おすすめ？の履歴データベース

start-time	Ladder position	chip_id	direction	DAC0
2022/12/02 9:00	B0L000	1	N	15
2022/12/10 10:00	B0L000	2	N	20
⋮				
2022/12/02 9:00	B1L116	25	S	15
2022/12/02 9:00	B1L116	26	S	15

present

start-time	Ladder position	chip_id	direction	DAC0
2022/12/02 9:00	B0L000	2	N	15

history

presentのテーブルを修正する場合は、まず初めにhistoryにタプルを移動してから、presentをUPDATEする。

これら一連の作業をまとめて実行する、トランザクションを実装するか…?

まとめ

- 履歴データなしでは、正しく正規化が行われているか検討した。

- リレーショナルデータベースは、履歴データを扱いにくい。しかし、現在と過去でテーブルを分けることで見やすくすることが可能。

他の活動報告

PCとFelixボードを遠隔で操作するためにRPCという技術を使う。

RPCを行うためのライブラリの1つgRPCを使ってみた、学習した。

```
riken_intt@DESKTOP-5QU8C9R:/mnt/c/Users/RIKEN_INTT/Desktop/Hi
E1213 15:48:22.937369900     438 socket_utils_common_posix.cc
rrno:92, created_time:"2022-12-13T15:48:22.9371869+09:00"}
server started
Hello!, I am from Hikaru PC
Fitting by pol0
 ---Fitting result---
mean (server side) --> 631.4122293950709
mean error (server side) --> 4.835869945081584
reduced chi2 (server side) --> 1.2257617820416864
Info in <TCanvas::Print>: pdf file fit.pdf has been created
```

```
[hikaru@hikaru helloworld]$ python3 riken_intt_client.py
reply_communication_status: " GOOD !!"

 Mean = 631.4122314453125  +-  4.835869789123535
reduced chi2 =  1.2257617712020874
```

理研でも宇宙線測定を行なっている。(DAC scan)

DAC値、回数、時間を指定すると自動で繰り返し宇宙線を測定するGUI機能開発。
データを取得すると、DSE除去→クラスタリングも自動で行うように。
宇宙線測定の様子が、webでみられるオンラインモニターの開発