

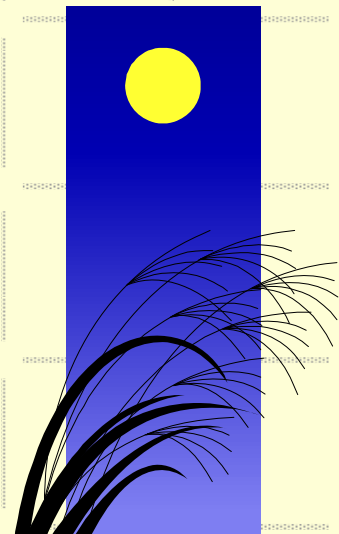
スキーマってなんだろう

2007/10/06 OSC2007Tokyo/Fall




日本LDAPユーザー会

太田俊哉





今日の内容

- LDAPのスキーマについて説明します
 - LDAPのデータ構造などについて説明します
 - やや観念的な話を中心です
- 

混沌から整理へ

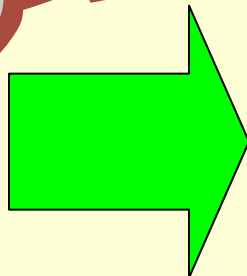
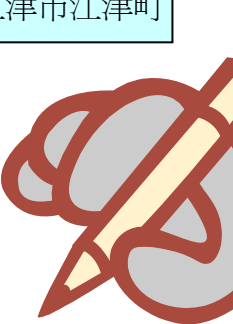
江津さん

123-4567 109-3434.

goutu@sancou.... 島根県江津市江津町

goutu-01@gmail.

enoca@anet.n..



電話番号 〒 住所 メール



枠の中に入れる

The diagram illustrates data entry into a table. A table with five columns and six rows is shown. The first row contains the following data: 氏名 (Name): 江津, 電話番号 (Phone Number): 123-4567, メール (Email): goutu@san, 〒 (Postal Code): 109-3434, 住所 (Address): 島根県江津市江津町. Callout boxes with arrows point to each of these fields. A large grey arrow on the right points to the table with the text 'この枠は?' (This frame?).

氏名	電話番号	メール	〒	住所	
江津	123-4567	goutu@san	109-3434	島根県江津市江津町	

123-4567

goutu@sancou....

109-3434.

島根県江津市江津町

goutu-01@gmail.

enoca@anet.n..

この枠は？

枠の意味


- 1つのデータにはいろいろな項目がある
- ある目的(たとえば住所録)でデータを集めるときには項目はおのずと決まる
- データの構造を決める

.....スキーマ

器と構造



LDAPのスキーマ

- いろいろな用途に合わせて、標準でいろいろなものが提供されている
 - 特別な応用例では、別途提供されている場合もある(例: Sambaとの連携)
 - もちろん自分で作ることも可能(若干ルールはある)
- 



LDAPのスキーマの例

- 電話帳が一番わかりやすい
 - 個人に関する情報(氏名、住所、電話番号、メールアドレスなどなど)
 - 個人⇒主体、個人にかかわる情報⇒属性
 - 情報は属性とデータの組
 - 属性は複数個存在、1つの属性に複数の値もありえる(ここが大事)



標準スキーマ

- あらかじめLDAPサーバに準備されているもの
- OpenLDAPだと以下の通り

ファイル	説明
core.schema	OpenLDAP core (必須)
cosine.schema	Cosine and Internet X.500 (有用)
inetorgperson.schema	InetOrgPerson (有用)
misc.schema	Assorted (実験用)
nis.schema	Network Information Services (参考)
openldap.schema	OpenLDAP Project (実験用)

枠....ではない

- データを枠の中に入れる
 - 考え方は合っているが、LDAPの枠は表ではない
 - LDAP ≠ 表形式データベース
 - データベースでよくやる正規化はしません
- データベースの一種ではあるが
 - 木構造
 - データは手繰っていく

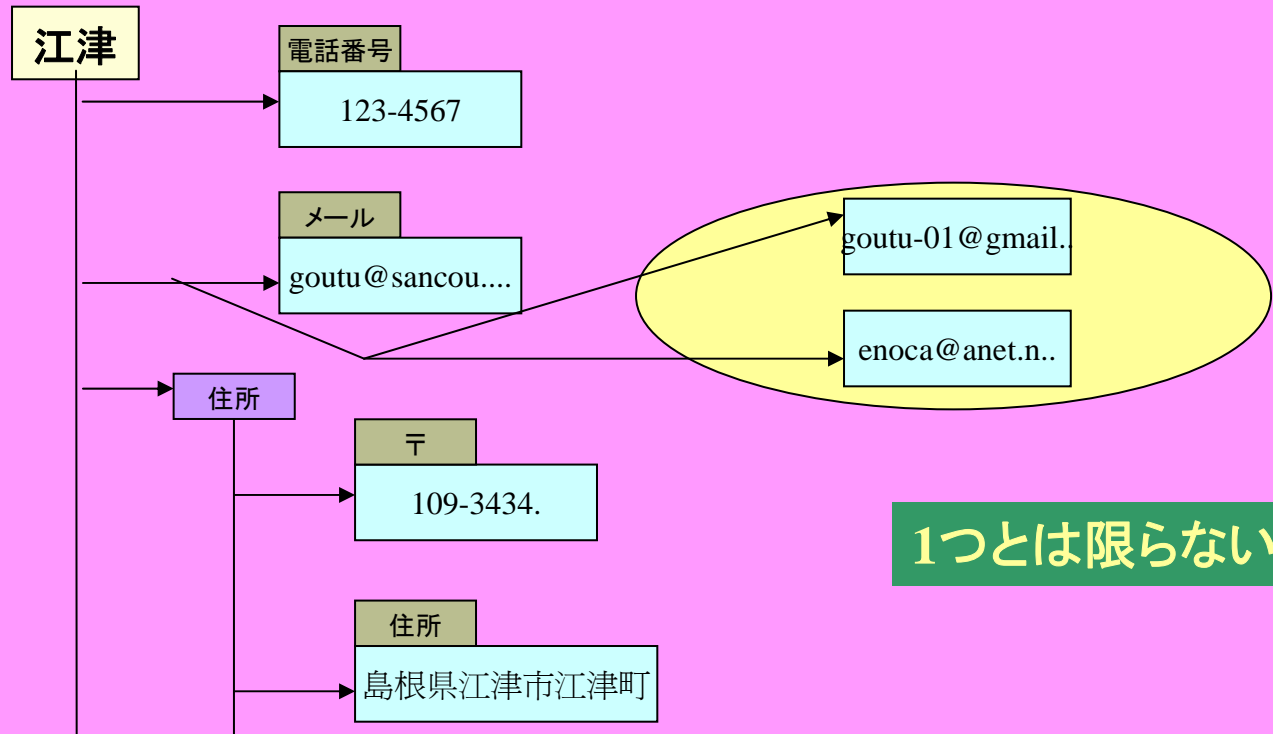
図にしてみると

氏名	電話番号	メール	〒	住所
江津	123-4567	goutu@san	109-3434	島根県江津市江津町
川平	123-4579	kawahira@s	109-3434	島根県江津市川平町
川戸	123-5432	kawato@sa	109-3434	島根県江津市桜江町
鹿賀	133-1222	shikaga@sa	109-3434	島根県江津市桜江町
明塚	142-1928	akatsuka@s	109-3434	島根県邑智郡美郷町
浜原	118-8791	hamahara@	109-3434	島根県邑智郡美郷町
沢谷	109-0122	sawatani@s	109-3434	島根県邑智郡美郷町

普通はこのような表形式

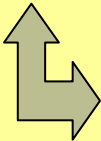
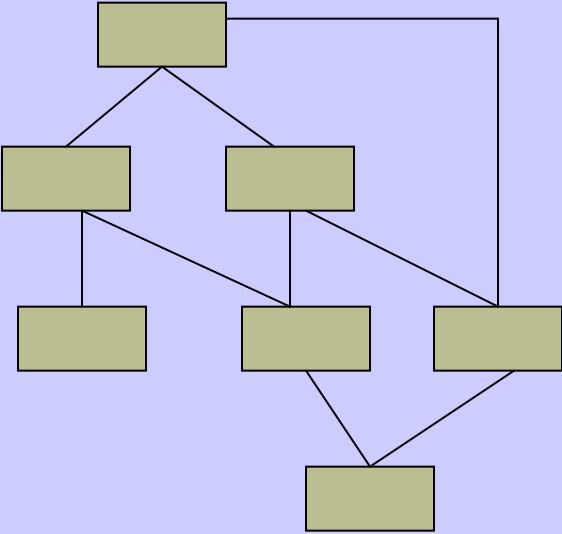
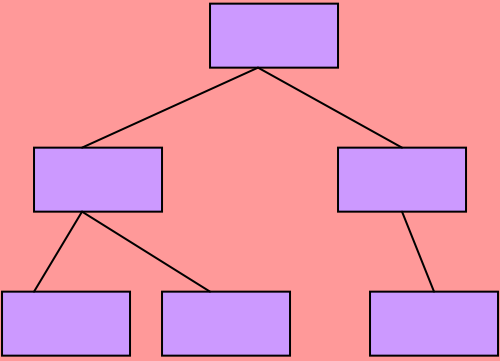
データベースとしては一般的

LDAPではこちら



1つとは限らない

データベース=表という固定概念

リレーショナルデータベース	CDASYL型データベース (ネットワーク型データベース)	階層型データベース																						
<table border="1" data-bbox="79 491 533 729"><thead><tr><th>seq</th><th>id</th><th>pass</th><th>user</th></tr></thead><tbody><tr><td>100</td><td>hoge</td><td>***</td><td>hoge</td></tr><tr><td>101</td><td>foo</td><td>***</td><td>test</td></tr><tr><td>102</td><td>bar</td><td>***</td><td>gues</td></tr></tbody></table>  <table border="1" data-bbox="239 822 629 996"><thead><tr><th>seq</th><th>home</th></tr></thead><tbody><tr><td>100</td><td>/home/hoge</td></tr><tr><td>101</td><td>/home2/var</td></tr></tbody></table>	seq	id	pass	user	100	hoge	***	hoge	101	foo	***	test	102	bar	***	gues	seq	home	100	/home/hoge	101	/home2/var		
seq	id	pass	user																					
100	hoge	***	hoge																					
101	foo	***	test																					
102	bar	***	gues																					
seq	home																							
100	/home/hoge																							
101	/home2/var																							
<ul style="list-style-type: none">● 表中にデータ、表間での関係、表への演算● 一般的	<ul style="list-style-type: none">● ノードが繋がっている● 親へのリンク	<ul style="list-style-type: none">● 木構造● アクセスパスは1つ● LDAPはこれ																						

何がほしい？

●情報だ

●LDAPは検索が中心

- 更新頻度は少ない
- 早く検索できることが大事⇒情報が早く欲しい
- 更新直後に新しいデータが検索できる必要もない

●ではLDAPのデータ構造は？

- スキーマの中身と言い換えてもよいかも



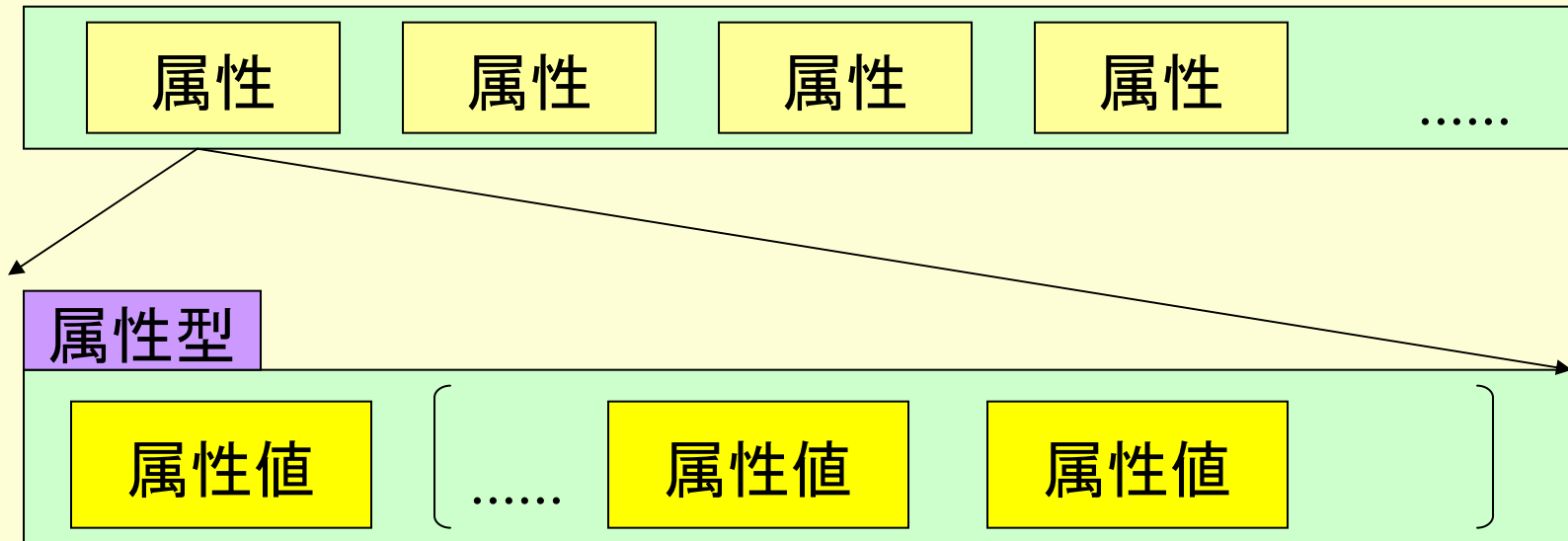


では、中身を見ていきましょう

- いくつかのキーワード
 - エントリ
 - 属性
 - 属性値
 - ディレクトリ属性ツリー(DIT)

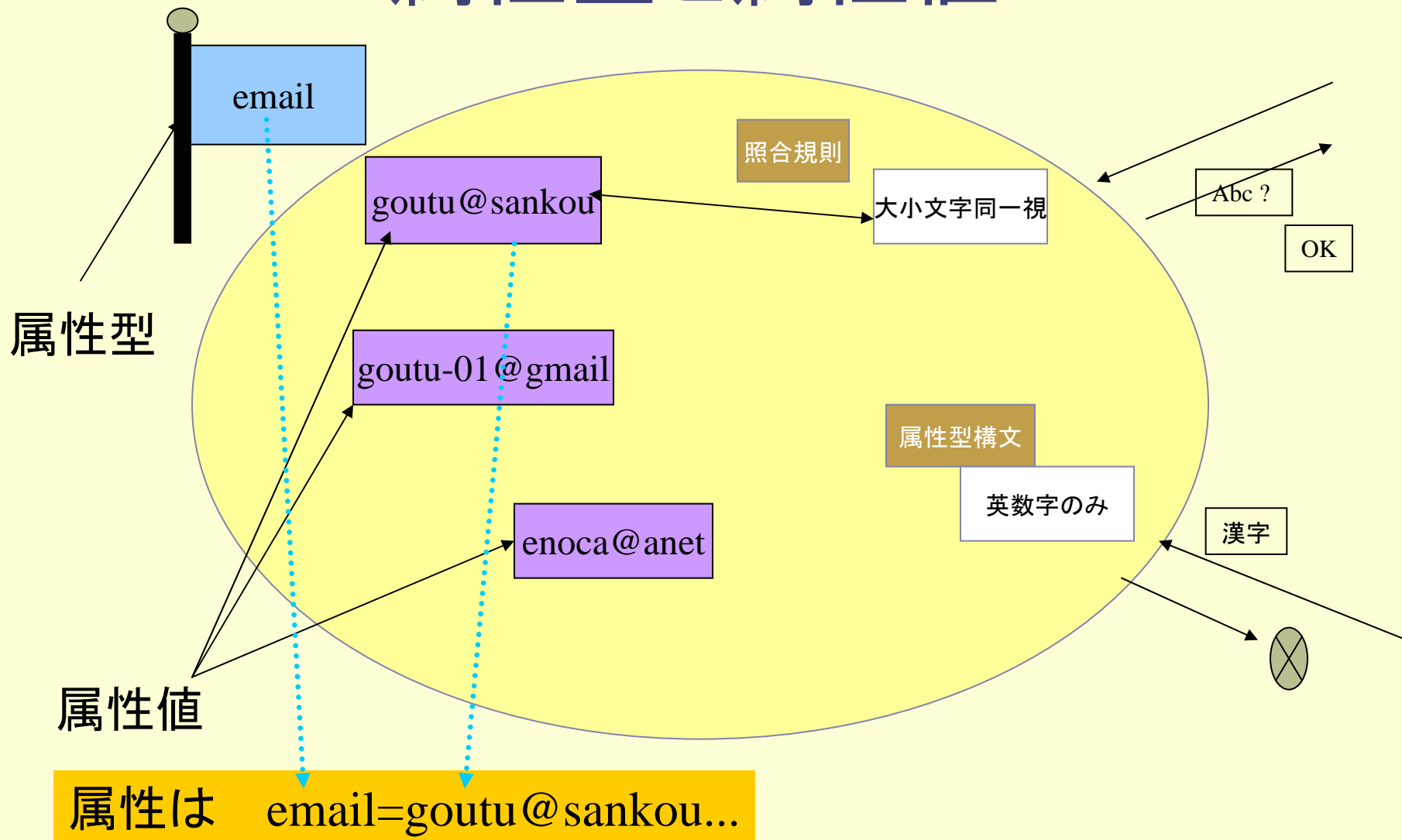


エントリー



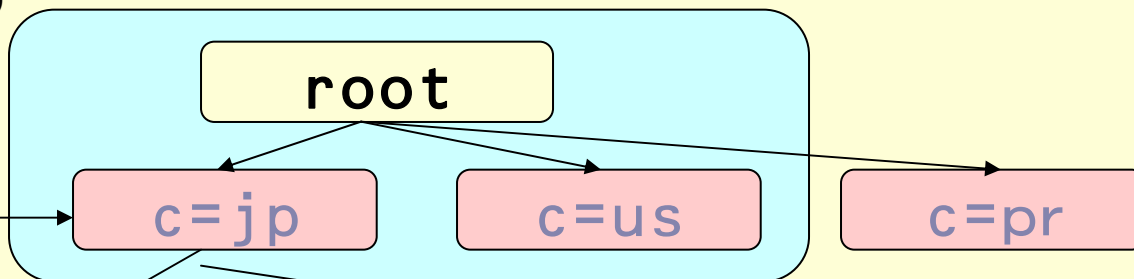
1つのエントリー⇒実世界でのあるもの(オブジェクト)
データベースの表だったらある1行
エントリーには複数の属性がある
住所、電話番号、メールアドレスなど
属性＝属性型と属性値で構成
属性型＝属性の名前
1つの属性には複数の属性値があってもよい

属性型と属性値



DIT(Directory Information Tree)

上位エントリ



直接上位エントリ

注目しているエントリ

o=日本株式会社

o=亜細亜株式会社

直接下位エントリ

ou=本社

ou=東京支店

ou=九州支店

下位エントリ

ou=庶務部

ou=営業部

ou=技術部

cn=江津

cn=川平

識別名: cn=江津,ou=庶務部,ou=本社,o=日本株式会社,c=jp

DIT:エントリ間の階層構造

識別名と相対識別名

- 相対識別名 (RDN: Relative Distinguished Name)
 - エントリの中で、一意になるようにつけられる名前
 - 例: cn=江津
- 識別名 (DN: Distinguished Name)
 - あるディレクトリ情報ツリーの中で、そのエントリを一意に表わすための名前
 - あるエントリのRDNを、ルートエントリまでの間にあるエントリのRDNを下位から上位に向かって連結したもの
 - 例: cn=江津,ou=庶務部,ou=本社,o=日本株式会社,c=jp




エントリーと相対識別名

● 相対識別名

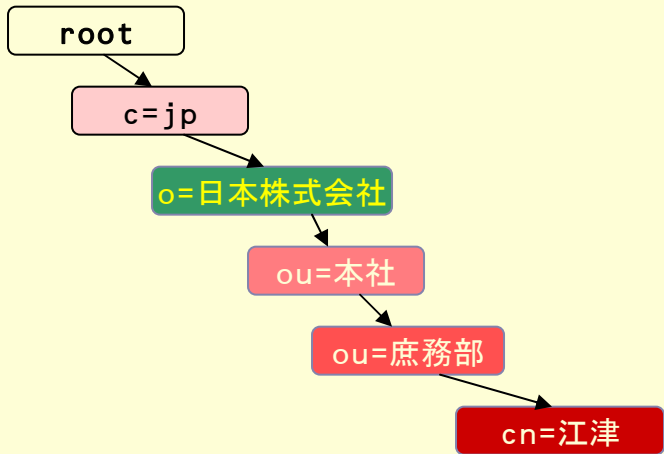
- ou=庶務部

● エントリー

- 属性の集合⇒属性:属性値の集合
 - 属性:ou
 - 属性値:庶務部
 - 相対識別名と同じ
 - エントリーの名前(識別名)は、ある属性:属性値と同じ
- 

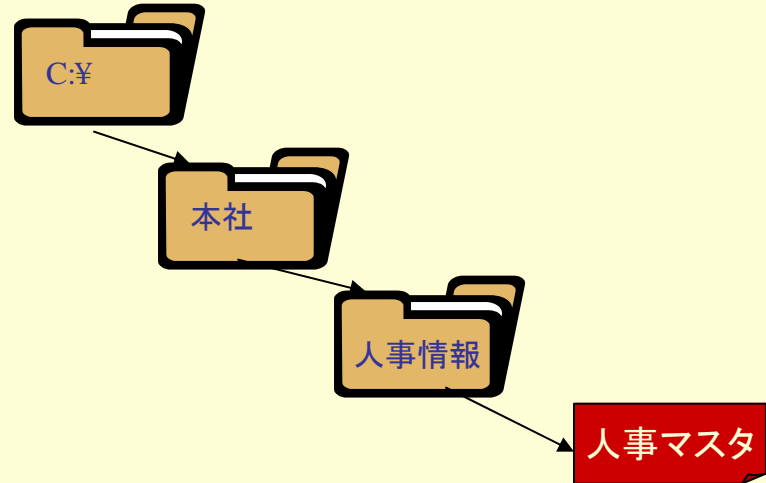
比べてみよう

LDAPのDIT



- ・各エントリ毎に属性型と属性値が存在
- ・エントリそのものが値を持つ
- ・エントリ≠器
- ・ある特定の情報⇒識別名
- ・特定のエントリ⇒相対識別名

ファイルシステムのディレクトリ



- ・ディレクトリそのものは値を持たない
- ・ディレクトリは器
- ・ある特定の情報⇒フルパス



エントリの属性は任意？

- 一定のルールがあります
- オブジェクトクラス
 - エントリの構造を規定
 - 構造型オブジェクトクラス(structural objectClass)
 - MUST属性
 - MAY属性
 - 補助オブジェクトクラス(auxiliary objectClass)
 - topは例外
 - その正体は OID(Object ID)



たとえば人を表わす時には

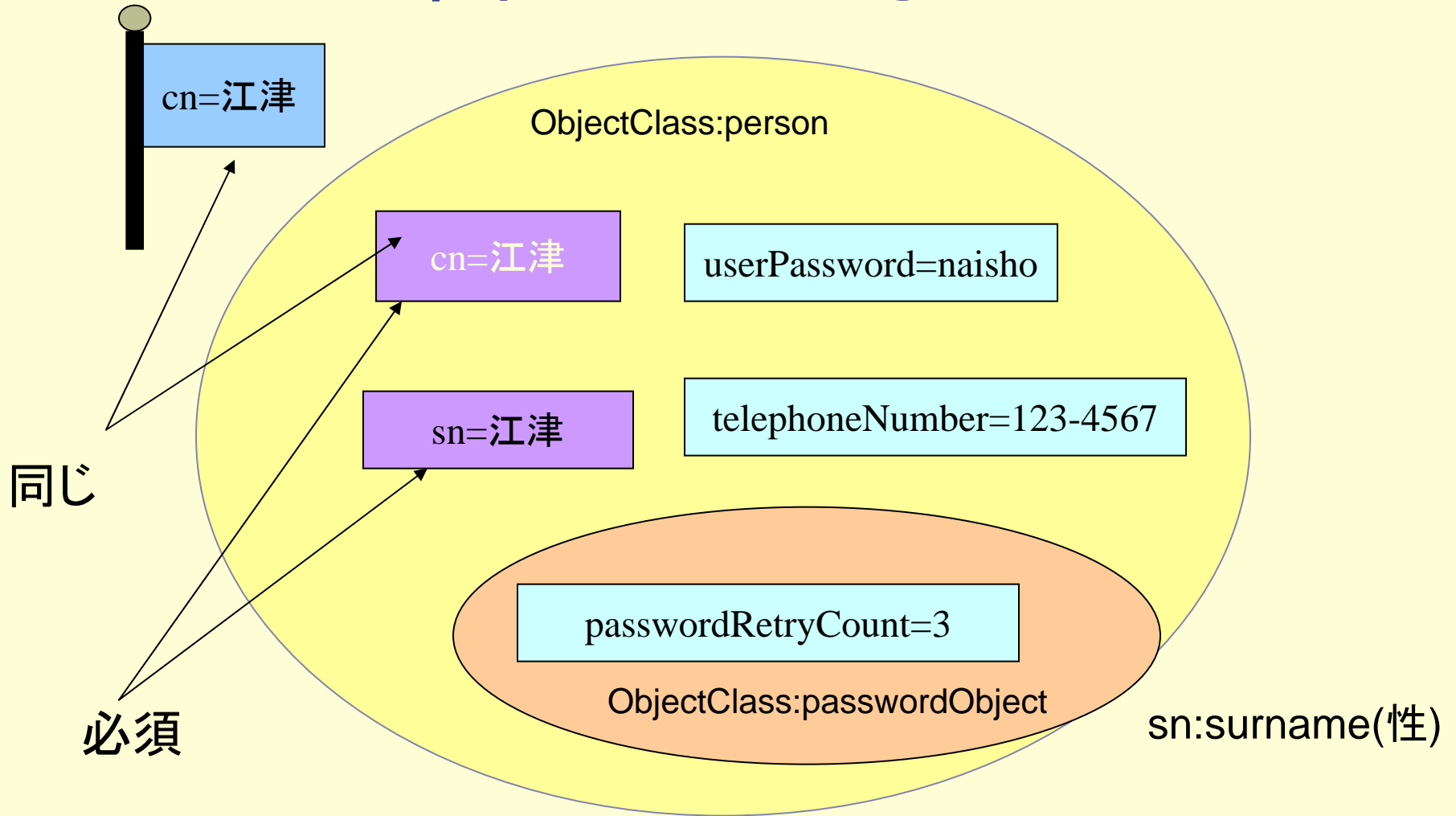
● オブジェクトクラス `person`

- MUST属性
 - `cn`、`sn`
- MAY属性
 - `userPassword`、`telephoneNumber`、`seeAlso`、`description`
- 補助オブジェクトクラス
 - `passwordObject`
 - `passwordExpirationTime`、`passwordExpWarned`、`passwordRetryCount`、`retryCountResetTime` など

● 階層構造を取る


- 上から下へ、属性が引き継がれる

図にしてみると






照合規則

- 属性の比較照合の時に、どのような照合条件を満たすとするかを規定
 - 同値性 (EQUALITY)
 - caseExactMatch、caseIgnoreMatchなど
 - 順序性 (ORDERING)
 - caseExactOrderingMatch、caseIgnoreOrderingMatchなど
 - 部分文字列性 (SUBSTRING)
 - caseExactSubstringsMatch、caseIgnoreSubstringsMatchなど
 - 検索時に使います
- 



属性構文

- ある属性値が持つことのできる値の定義
 - その定義に合っていないと値が入らない
 - 入力値チェックみたいなもの
 - 例
 - INTEGER
 - Numeric String
 - など
- 




標準定義での例

- 属性型name と cn(OpenLDAPのドキュメントより)

```
attributeType ( 2.5.4.41 NAME 'name'  
  DESC 'name(s) associated with the object'  
  EQUALITY caseIgnoreMatch  
  SUBSTR caseIgnoreSubstringsMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )  
attributeType ( 2.5.4.3 NAME ( 'cn' 'commonName' )  
  DESC 'common name(s) associated with the object'  
  SUP name )
```





定義方法の実際

- 使用目的に応じたディレクトリ構造の設計
 - 標準スキーマの利用
 - データの準備
 - LDAP用データに変換
- 




LDIF

- LDAP Interchange Format(RFC2849)
 - LDAPの設定情報と内容を保存、交換するためのテキスト形式
 - LDAPサーバ間で互換
 - 一括登録などに使用
- 



LDIFの形式

- テキストファイル
 - エントリ記述の集合体
 - dn: hogehoge....で始まる
 - エントリ記述を列挙
 - コマンドも書ける
- 

LDIFの例

```
dn: dc=my-domain,dc=com
```

```
objectClass: dcObject
```

```
objectClass: organization
```

```
dc: my-domain
```

```
o: my-domain
```

```
dn: ou=People,dc=my-domain,dc=com
```

```
objectClass: organizationalUnit
```

```
ou: People
```

```
dn: uid=ldapuser,ou=People,dc=my-domain,dc=com
```

```
objectClass: account
```

```
objectClass: posixAccount
```

```
uid: ldapuser
```

```
cn: ldapuser
```

```
userPassword: ldapuser
```

```
loginShell: /bin/bash
```


```
uidNumber: 1000
```

```
gidNumber: 1000
```

```
homeDirectory: /home/ldapuser
```




直接スキーマを操作する場合

- 基本的にはありえない
 - 手で、動的に修正すると、痕跡が残らない
 - CSVなどで元データを作り登録するのが基本
 - ⇒LDIFファイル
 - 状況を見ることはありえる
- 



閲覧ツール

- 商用LDAP製品にはたいていある
 - 差別化のためについている
 - フリーのもの
 - OpenLDAPにはない(!)
 - フリーでいくつかある
 - 特定の用途に特化したもの
 - LAM(Ldap Account Manager)
- 

phpldapadminでの例(1)

The screenshot displays the phpldapadmin interface for a Local LDAP Server. The left sidebar contains navigation icons and the text "Local LDAP Server" with a clock icon. Below the icons are labels: スキーマ (Schema), 検索 (Search), 再描画 (Refresh), 情報 (Info), インポート (Import), エクスポート (Export), and ログアウト (Logout). The login status is "Logged in as: uid=ldapuser,ou=People". The domain is "dc=my-domain,dc=com (1)".

The main content area is titled "サーバーのスキーマ Local LDAP Server". It includes navigation links: "ObjectClass 一覧 | 属性タイプ | 文法一覧 | 一致ルール". Below this is a search field for "objectClass に移動:" with a dropdown menu showing "- all -" and a "Go" button.

The selected object class is "account", with the following details:

- OID: 0.9.2342.19200300.100.4.5
- 種類: structural
- 継承元: top
- 派生先: (なし)

必須属性	オプション属性
<ul style="list-style-type: none">• userid	<ul style="list-style-type: none">• description• host• localityName• organizationName• organizationalUnitName• seeAlso

phpldapadminでの例(2)

Local LDAP Server

スキーマ 検索 再描画 情報 インポート エクスポート ログアウト

Logged in as: uid=ldapuser,ou=People

dc=my-domain,dc=com (1)

dc=my-domain

サーバー: Local LDAP Server 関連名: dc=my-domain,dc=com

- Refresh
- このエントリをコピー
- このエントリを削除
- ヒント: 属性を削除するにはテキストフィールドを空にして保存をクリックします。
- 別のエントリと比較
- 新規属性を追加
- サブツリーをエクスポート
- ヒント: 属性のスキーマを閲覧するには、属性名をクリックします。
- エクスポート
- 内部属性を表示
- 名称変更
- 子エントリ作成
- ひとつの子を閲覧

dc

my-domain
(名称変更)

o

my-domain
(値追加)

objectClass

dcObject

phpldapadminの例(3)

Local LDAP Server



スキーマ 検索 再描画 情報 インポート エクスポート ログアウト

Logged in as: uid=ldapuser,ou=People

dc=my-domain,dc=com (1)

高度な検索フォーム

(簡易検索フォーム | 事前定義検索)

サーバー Local LDAP Server

ベース DN dc=my-domain,dc=com



検索スコープ ひとつ (基準の下の1レベル)

検索フィルター objectClass=*

属性表示 cn, sn, uid, postalAddress, telephoneNumbe

Order by

検索

Entries found: 1 (0.42 秒)

[結果エクスポート] [書式: list table]

ベース DN: dc=my-domain,dc=com

ou=People


dn ou=People,dc=my-domain,dc=com

objectClass organizationalUnit

ou People



LDAPのスキーマとは

- 木構造ではあるが、ファイルシステムとは違う
 - DNとかRDNとかの独特な概念
 - 属性、それに附属する属性構文など規則が多数
 - 定義するのは大変
 - あらかじめできているのをうまく利用
- 



情報を共有しましょう

- LDAPの情報はそれほど多くない
 - Sambaよりは圧倒的に少ない
 - 個人で使うものではないので
 - LDAPユーザー会
 - MLで活動中です
 - 日経Linuxにも連載中
 - 是非ご参加を

Fin

