



DB設計方法

どのように進めればいいのか？



Contents

- 本日の講座の目的と皆さんの思いの言語化
 - データベース、リレーショナルデータベースって何？
 - ER図は何を表しているの？
 - 概要設計の方法
 - 詳細設計は何をするの？
-
- 本日のまとめと感想

本日の講座の目的と皆さんの思いの言語化

スキルを身につけるときや覚えるときには目的意識を持つことがとても重要になってきます

下のことを言語化しましょう

- ・ いま具体的にどんなことが課題になっていますか？
- ・ それを解決するために今は、何が必要だと思っていますか？
- ・ 今日は何を（どんなスキル）持ち帰りたいですか？

データベース、リレーショナルデータベースって何？

全体像を把握しよう

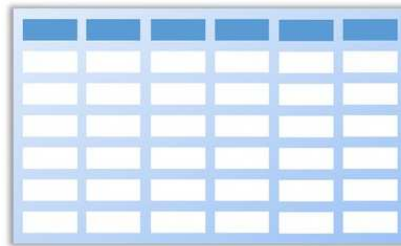
先ずは大まかな用語を知って、どんなものなのかを知りましょう

データベース(DB)



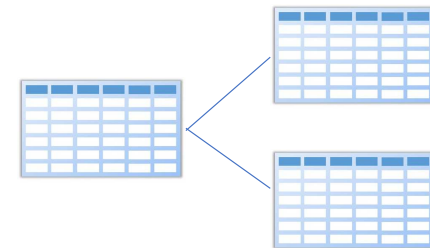
いろんなデータが詰まっているところ
テーブル(表形式になっている)

テーブル



DB内にあるマスで
列をカラム、行をレコードと言います

リレーショナル データベース(RDB)



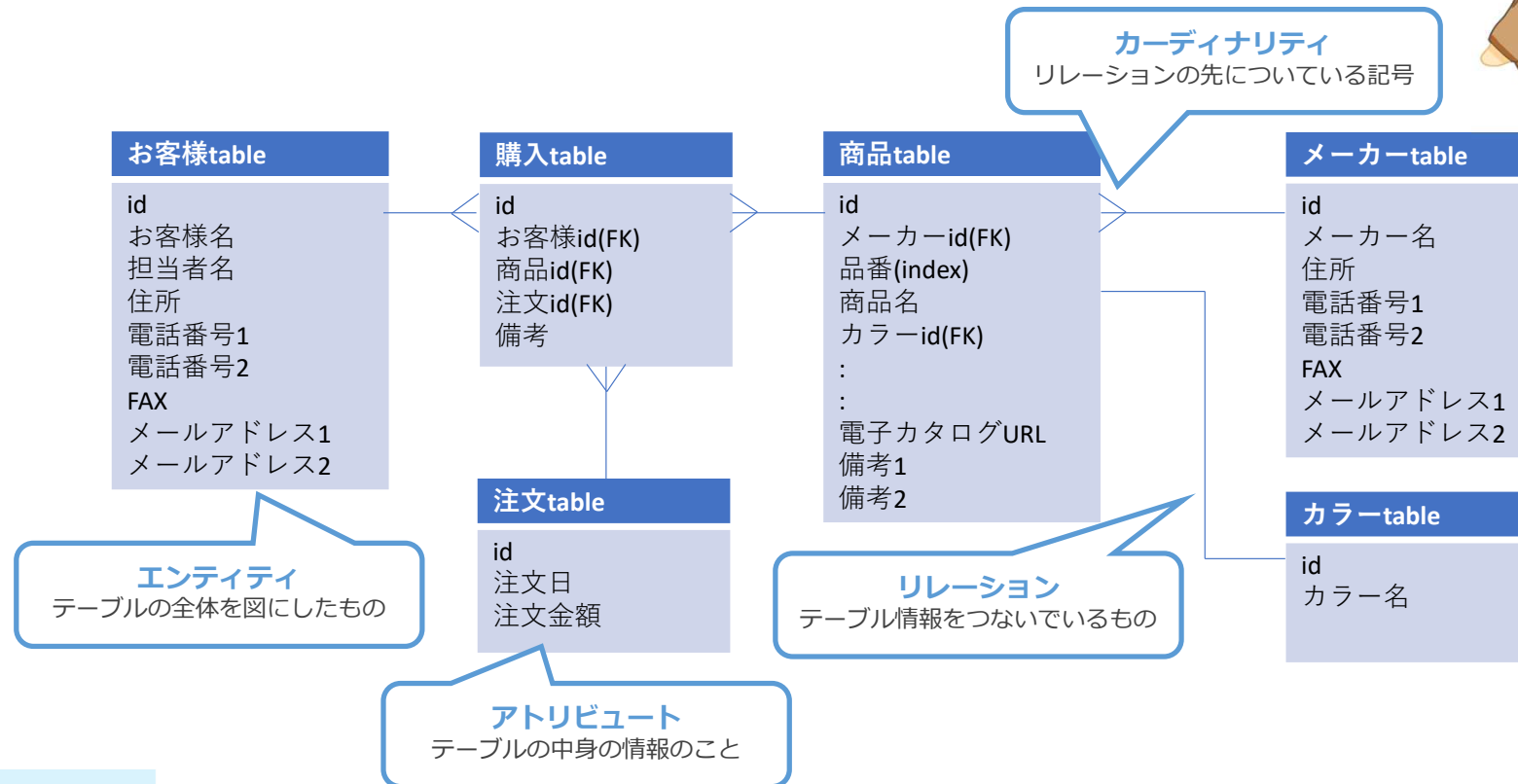
テーブル同士がお互いに関連づいてい
る仕組みのDB

ER図は何を表しているの？

先ずは何となく全体を見てみよう

複数のテーブルの関係性を確認するのに**ER図**というものを作ります
実際にどういうものなのか紐解きながら覚えていきましょう

この書き方はIE記法っていうんだね



ER図は何を表しているの？

テーブルの関係性を線と記号で見よう

ここではカーディナリティとリレーションに注目してみよう
右の2つのテーブルの関係性を先ずは言語化すると…

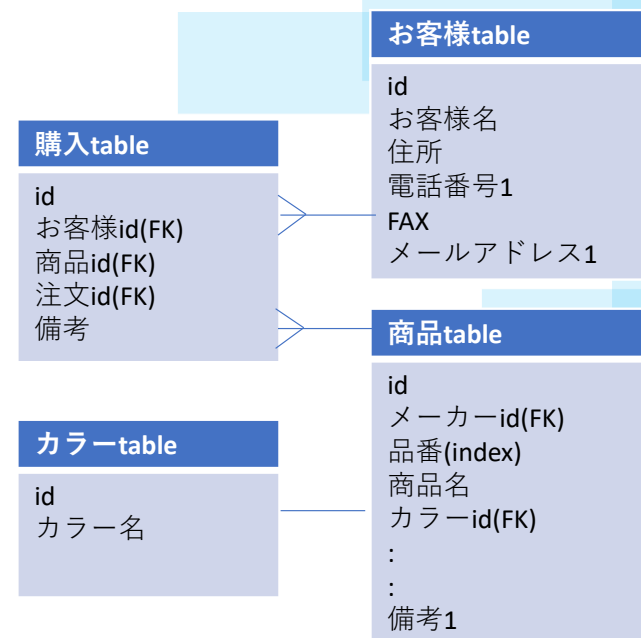
1人のお客様が複数の商品を購入する可能性がある

1つの商品は複数のお客様が購入する可能性がある

1つの商品には、複数の色から1つ選択される

こんな感じになります

関係性があるのであれば、リレーションでつなぎ、さらにカーディナリティで分類していきます



ホントはもっと細かく書くけど
設計ではここまでOK!!



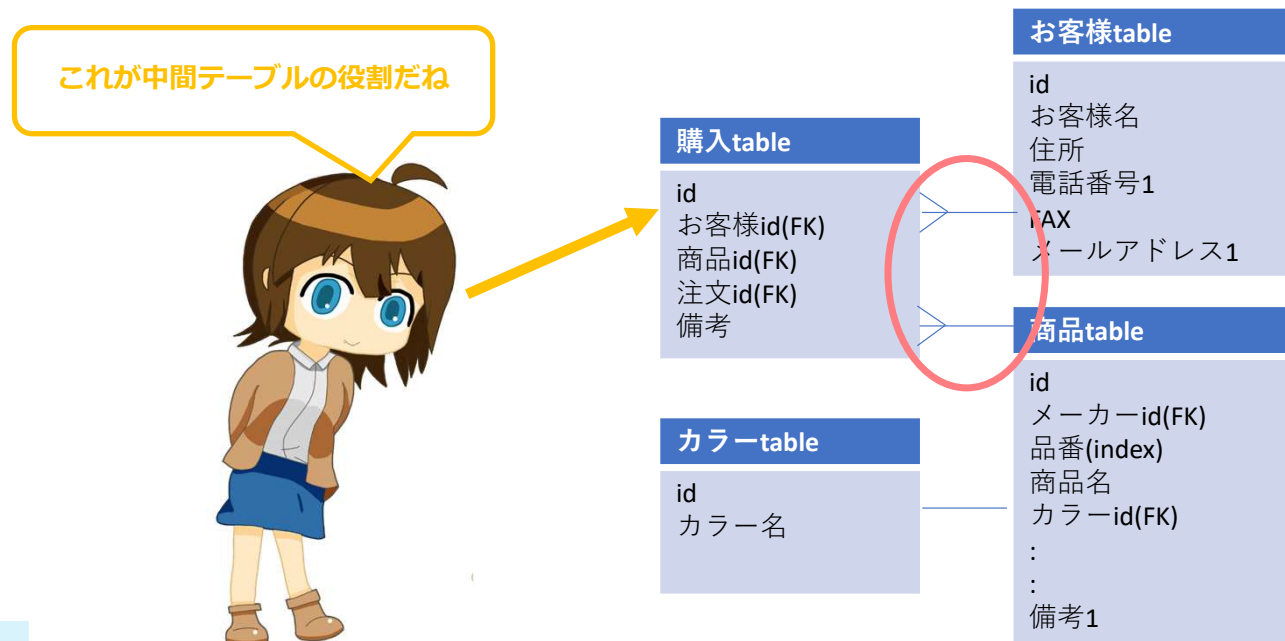
ER図は何を表しているの？

気になる部分があったのでは？

テーブルの中で、複数のお客様が複数の商品を購入する場合があるので、この場合は多対多ではないの？と思ったのではないですか？

はい、その通りです

この場合は、中間テーブルというものを入れて1対多の関係性に直して表現しましょう



概要設計の方法

機能から考えてみよう

作りたいシステムの機能を書き出してみるとどんなテーブルが必要かが見えてきます

■ 商品検索画面

■ お客様登録

■ 購入履歴

■ 商品ページ

商品table

id
メーカーid(FK)
品番(index)
商品名
カラーid(FK)
:
:
電子カタログURL
備考1
備考2

メーカーtable

id
メーカー名
住所
電話番号1
電話番号2
FAX
メールアドレス1
メールアドレス2

どんなテーブルが必要かな…

お客様table

id
お客様名
担当者名
住所
電話番号1
電話番号2
FAX
メールアドレス1
メールアドレス2

カラーtable

id
カラー名

注文table

id
注文日
注文金額

購入table

id
お客様id(FK)
商品id(FK)
注文id(FK)
備考



詳細設計は何をするの？

テーブルの内容を考えてどんどん分けていこう

実際にどうやってテーブルを分けていくかは、実際にテーブルを作って切り分けていくわかりやすいです

管理id	お客様名	会社名	電話番号	購入商品	カラー	購入商品	カラー	購入商品	カラー	備考
1	Aさん	A社	000	リンゴ	赤	塩	白	きゅうり	緑	いろいろ
2	Bさん	B社	111	砂糖	白					
3	Cさん	C社	222	リンゴ	赤	牛乳	白			なんか
4	Aさん	A社	000	レモン	黄色					

横の繰り返しを無くそう
これが**第一正規化**というよ

管理id	お客様名	会社名	電話番号	購入商品	カラー	備考
1	Aさん	A社	000	リンゴ	赤	いろいろ
1	Aさん	A社	000	塩	白	いろいろ
1	Aさん	A社	000	きゅうり	緑	いろいろ
2	Bさん	B社	111	砂糖	白	
3	Cさん	C社	222	リンゴ	赤	なんか
3	Cさん	C社	222	牛乳	白	なんか
4	Aさん	A社	000	レモン	黄色	



詳細設計は何をするの？

テーブルの内容を考えてどんどん分けていこう

縦の繰り返しが出てきてしまったので、違う部分を切り出していこう

この切り出し作業を**正規化**といいます

管理id	お客様名	会社名	電話番号	購入商品	カラー	備考
1	Aさん	A社	000	リンゴ	赤	いろいろ
1	Aさん	A社	000	塩	白	いろいろ
1	Aさん	A社	000	きゅうり	緑	いろいろ
2	Bさん	B社	111	砂糖	白	
3	Cさん	C社	222	リンゴ	赤	なんか
3	Cさん	C社	222	牛乳	白	なんか
4	Aさん	A社	000	レモン	黄色	

切り出してみただ
まだまだ分けられるな…
これが**第二正規化**と**第三正規化**か…



管理id	お客様名	会社名	電話番号	備考
1	Aさん	A社	000	いろいろ
2	Bさん	B社	111	
3	Cさん	C社	222	なんか
4	Aさん	A社	000	

管理idで紐づくんだね
この紐づきidを
主キー(プライマリキー)というよ



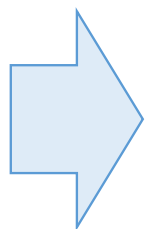
管理id	購入商品	カラー
1	リンゴ	赤
1	塩	白
1	きゅうり	緑
2	砂糖	白
3	リンゴ	赤
3	牛乳	白
4	レモン	黄色

詳細設計は何をするの？

テーブルをエンティティにしてみよう

切り出し作業(正規化)が終わったらエンティティに落とし込んど、**id**、**生成日**、**更新日**のアトリビュートを追加しよう

id	お客様名	会社名	電話番号
1	Aさん	A社	000
2	Bさん	B社	111
3	Cさん	C社	222



お客様table
id お客様名 会社名 電話番号 生成日 更新日

個別のテーブルのidは単純にidと付けてね



購入table

id
お客様id(FK)
商品id(FK)
注文id(FK)
備考

左のような中間テーブルには(FK)という文字がついています
これは『**外部キーですよ**』という意味です

簡単に言うと、テーブル同士が紐づくためのものです

個別のテーブルはidという名前ですが、こちらには**テーブル名とidを組み合わせた名前**でアトリビュートを書いてください

詳細設計は何をするの？

条件を付けていこう

いろんな条件をアトリビュートにつけていきましょう

index

データを検索しやすくするために付けるものです
*プライマリキー、外部キーにつけなくて大丈夫です

Not null

データが入っていないことを許さないという条件です
入っていても入っていなくてもいいような情報には付けません
例：備考など

ユニーク

データの内容が重複しないで、ユニーク(独自)であることを条件にします
これがないと同じ名称が被って、SQLが狂います

外部キー制約

外部キー先に必ずテーブルがあることを保証する
ユーザー登録の場合は、論理削除を採用しておいてください
(ユーザーがいなくなる可能性があるため)

このテーブルで考えるとどれが何だろう



商品table

id
メーカーid(FK)
品番(index)
商品名
カラーid(FK)
:
:
電子カタログURL
備考1
備考2

詳細設計は何をするの？

データ型とテーブルやカラムの名前の付け方

最後にデータのデータ型(入力形式)を決めて、テーブルの名前をアルファベットで表しましょう

名称付けは英語が多いですが、短縮した言葉を使うことも多々あります

名称付けの際には予約語を使わないように注意してください

下記のサイトが参考になります

<https://qiita.com/okazy/items/9dd6552fc678077e27cd>

データ型の例

型名	
INTEGER	整数(8バイト)
SMALLINT	整数(4バイト)
DOUBLE PRECISION	倍精度浮動小数 点数
CHARACTER	固定長文文字
VARCHAR	可変長文文字
DATE	日付

都度調べて
使えばいいやって感覚ね



Field-UP



上田 敬介 (うえだ けいすけ)

〒114-0011 東京都北区昭和町1-3-18 マリモハイツ101号室

- エネルギー管理士, 米国NLPプラクティショナー
- 健康予防管理専門士, 消防設備士 他

TEL:080-4955-4941

省エネ事業URL: <http://www.field-up.work/>

心理・健康予防事業URL: <http://www.field-up.work/mental/>

内装事業URL: <http://www.field-up.work/officeinterior/>