データベース設計の流れ

# データベースができるまで

システム開発の一環としてデータベースを作成する場合、何から行えばよいでしょうか。

それを明らかにするには、まず私たちが使える材料(INPUT)と作るべきもの(OUTPUT)を明確にすることが大事です。

**INPUTで初めに行うのが、要件聴取（インタビュー）**になります。顧客から要件を聞き出すのはエンジニアにとって非常に大切な作業です。また、インタビューした要件は後からでも確認できるように一覧表にまとめておくと良いでしょう。

要件聴取で聞き出した内容を材料にして、最終的には必要な項目を持つテーブルを内部に持つデータベースを作ります。各テーブルは、CREATE TABLE文やCREATE INDEX文などの複数のDDL文を実行すればデータベースを作成することが出来ますので、**成果物(OUTPUT)は一連のDDL文**と考えてよいでしょう。

・データベース構築のINPUTとOUTPUT

顧客の要件

～～～

データベース

設計作業

必要なテーブルを

持つデータベース

INPUT

OUTPUT

|  |  |
| --- | --- |
| INPUT | 要件の一覧表(顧客から聴取したもの) |
| OUTPUT | 一連のDDL文(実行すれば必要なテーブルなどが生成されるもの) |

# 要件定義

要件定義とは、システム開発などのプロジェクトを始める前の段階で必要な機能や要求をわかりやすくまとめていく作業のことです。システム開発では、ユーザー側の要求を実現することがゴールです。その為、開発をスタートする前に、要求をもとにして実装する機能や性能などを要件定義に定め、具体的にどのように進めていくかを決めます。

# 設計工程

顧客の要件からデータベース構築するためには、以下の工程を実施することが多いです。それぞれの工程の詳細は次節以降で説明しますので、概要をイメージしておきましょう。

・データベース構築のおおまかな流れ

・DDL文

CREATE TABLE ～

CREATE TABLE ～

CREATE INDEX ～

顧客の要件

データベース設計作業

概念設計

(概念データモデル)

商品

従業員

店舗

論理設計

(論理データモデル)

商品

従業員

職種

物理設計

(物理データモデル)

employee

job

product

index

* 概念設計 (概念データモデル)

管理すべき情報はどのようなものなのかを整理します。データベースやシステムに関することは意識せず、要件に登場する情報だけを把握します。

* 論理設計 (論理データモデル)

概念設計で明らかになった各情報について、RDBを使う前提で構造を整理し詳しく具体化していきます。論理設計では、「どのようなテーブルを作り、それぞれのテーブルにどのような列を作るか」まで明らかにすればOKです。型や制約等、付随的な部分は物理設計で行います。

* 物理設計 (物理データモデル)

特定のDBMS製品（MySQLなど）を使うことを前提で、論理設計で明らかになった

各テーブルについて、その内容を詳しく具現化していきます。全てのテーブル、列について型、インデックス、制約、デフォルト値などテーブル作成に必要な全ての要素を確定させていきます。この物理設計に基づいて、CREATE TABLE文などを含む一連のDDL文を作成し、最終的にデータベース内にテーブルを作成することが出来ます。

概念設計　概念データモデル(エンティティ、属性、関連)

# 概念設計ですること

概念設計では、要件を実現するために抽象的な概念としてどのような「情報の塊」を管理しなければならないのかを明らかにします。

この情報の塊を**エンティティ(entity)**といい、通常エンティティは複数の**属性(attribute)**を持っています。さらにエンティティ同士にどのような関連があるかも、この概念設計で明らかにします。概念的な存在であるエンティティは、イメージがしにくいですが今後学習する論理設計や物理設計を経てテーブルになりますので、いわば「**テーブルの原石**」とイメージしてください。

・概念的な「もの」のイメージをつかむためのヒント

|  |  |
| --- | --- |
| エンティティ | 「テーブル」のようなもの |
| 属性 | テーブルの「列」のようなもの |
| 関連 | 「リレーションシップ」のようなもの |

たとえば、ピザ屋で勤務している従業員の管理を概念モデルで表す場合、従業員情報や店舗情報がエンティティとして考えられます。従業員エンティティは、「氏名」や「職種」という属性を持っています。また、「勤務店舗」という情報が含まれるため、従業員エンティティと店舗エンティティには関連があると言えます。

店舗エンティティ

【属性】

氏名

職種

給与

勤務店舗

【属性】

店舗名

住所

定休日

保有バイク

従業員エンティティ

従業員と店舗は

関連がある

# データモデル

データモデルとは、**現実の世界で扱っている情報をコンピュータの世界で扱うことが出来るように構造化(モデル化)**したものです。主なデータモデルには、階層型、ネットワーク型、オブジェクト指向型、**リレーショナル型**があります。

リレーショナル型は、**データを表形式で管理する方法**です。**同じデータを重複して持つことが無いように、お互いの表を参照できる仕組み**（リレーションシップ）が整えられています。現在のデータベースの主流となっています。

# モデル化の手法

データベース設計で「どのようなデータを管理するか」を明確にするためにデータモデルを行いますが、モデル化の手法として**トップダウンアプローチ**と**ボトムアップアプローチ**があります。

**ボトムアップアプローチは、既存の伝票や台帳、入力画面などすでにある「もの」から現状を把握して、それを元にシステムを構築していく手法**です。

**トップダウンアプローチは最初にシステム全体のあるべき姿を把握して、それに必要な情報を収集していく手法**です。ゼロの状態からシステム全体像を作り上げる為、高度なスキルが必要になります。

授業ではトップダウンアプローチで進めていきますが、**モデル化をしやすくする為に伝票や情報表示画面を使いながら学習していきます**。

|  |
| --- |
| トップダウンアプローチ (上級者向け) |
| １．エンティティを洗い出し、リレーションシップを考えてER図を作成 |
| ２．エンティティの属性を洗い出し、主キーを決定 |
| ３．正規化を行い、多対多のリレーションを排除 |

※新規案件のシステム開発に向いている

|  |
| --- |
| ボトムアップアプローチ（中級者、初心者向け） |
| １．帳票や仕様から属性の洗い出しを行う |
| ２．主キーを見つけ、テーブルの正規化を行う |
| ３．テーブル構造からER図を作成 |

※機能改善や元となる帳票などがあるシステム開発に向いている

ER図(Entity-Relationship Diagram)

# ER図(Entity-Relationship Diagram)とは

ER図は、概念モデルおよび論理モデルを表現する手法の一つです。**世の中に実在するあらゆるものは、実体(entity)と関係(relationship)という２つの概念で表現できる**という考え方に基づいて作られた手法です。リレーショナルデータベースと相性が良く、**データベースで表現すべき対象とその関係を図式化する**ことから、リレーショナル型のデータモデルの表現にはER図を用いるのが一般的です。

# ER図の記述ルール

ER図にはIE(Information Engineering)とIDEF1Xの２つの記述方式があります。

この授業では、IE形式によるER図を学習していきます。まずは、以下の図を見てください

・IE形式によるER図の書き方



エンティティ名

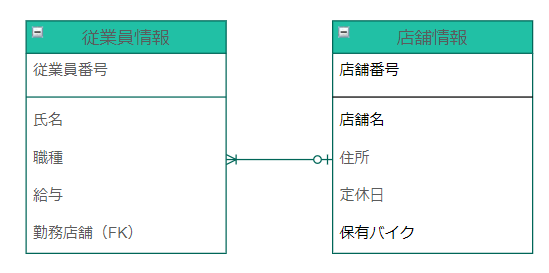
属性

主キー

外部キー

ER図に登場する**四角形はエンティティ**を表しています。そして**四角形の中には属性の一覧**が記述されます。つまり、従業員エンティティには氏名や職種などの属性が存在します。

さらに属性の一覧は、２つのグループに分けられます。**四角形の線より上には、エンティティを一意に特定する主キーとなる属性を記述**します。複数の属性で複合主キーを構成する場合、線より上に複数の属性を記述します。



カーディナリティ一覧

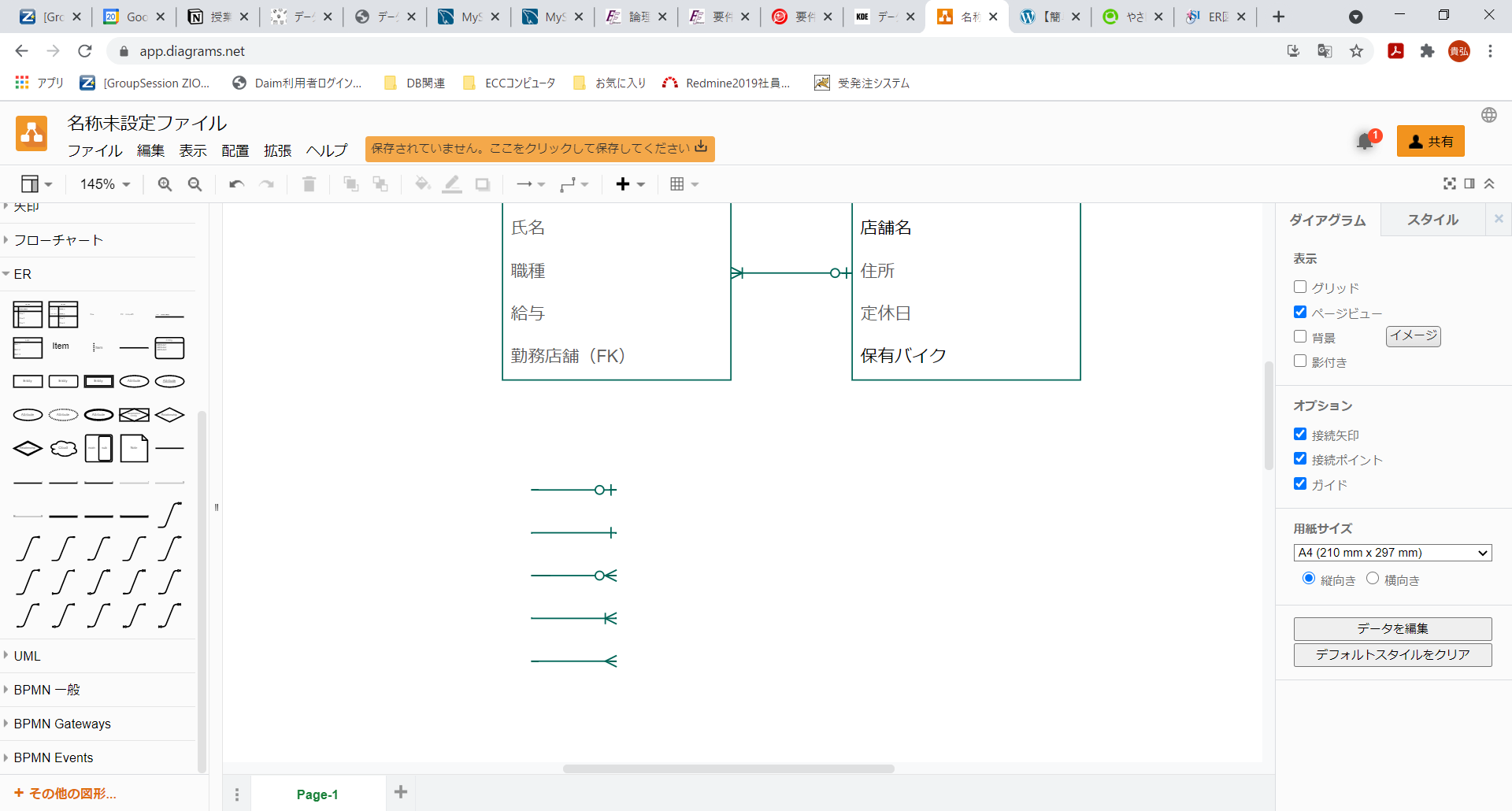
０または１

１

０または多

１または多

多(2以上)



・１人の従業員は勤務する店舗が１つある、または未配属

・１つの店舗は１人以上の複数の従業員が勤務している

複数のエンティティ間にリレーションシップがある場合には、**エンティティ同士を線でつなぎ、外部キーとなる属性には(FK)と追記**します。また、**エンティティ同士の数量的な関係を多重度やカーディナリティ**と呼びます。

ハンズオン　ER図作成ツールを利用する

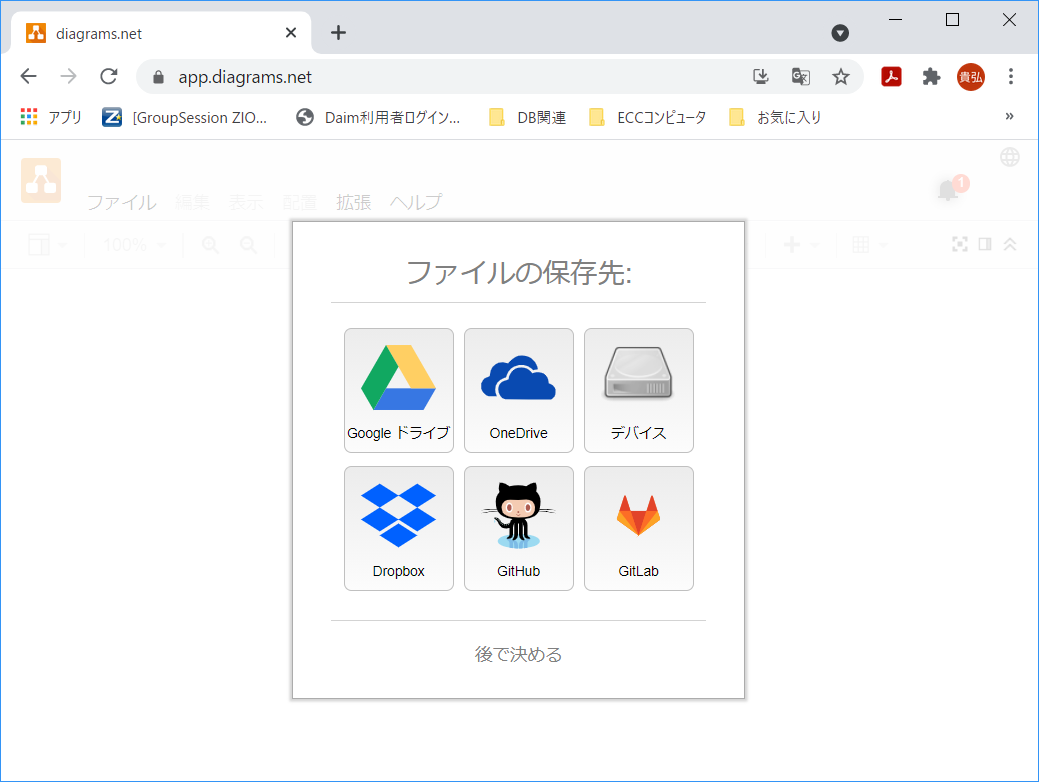
フリーで使用できる作画ツール「draw.io」を使用して、ER図の作成を行います。

draw.ioは、ブラウザ上で動作し会員登録不要で無料で使用できる作画ツールです。

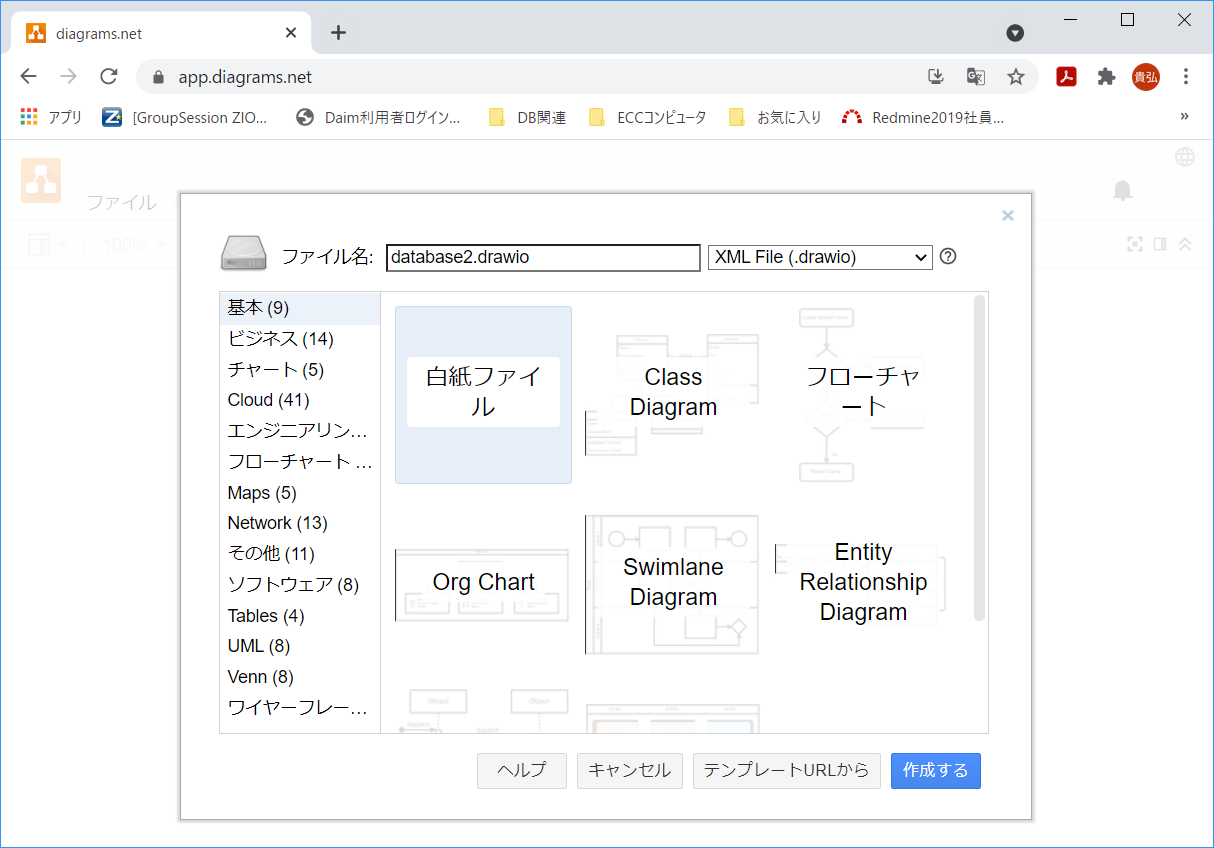
１．draw.ioのサイトに移動する。

https://app.diagrams.net/

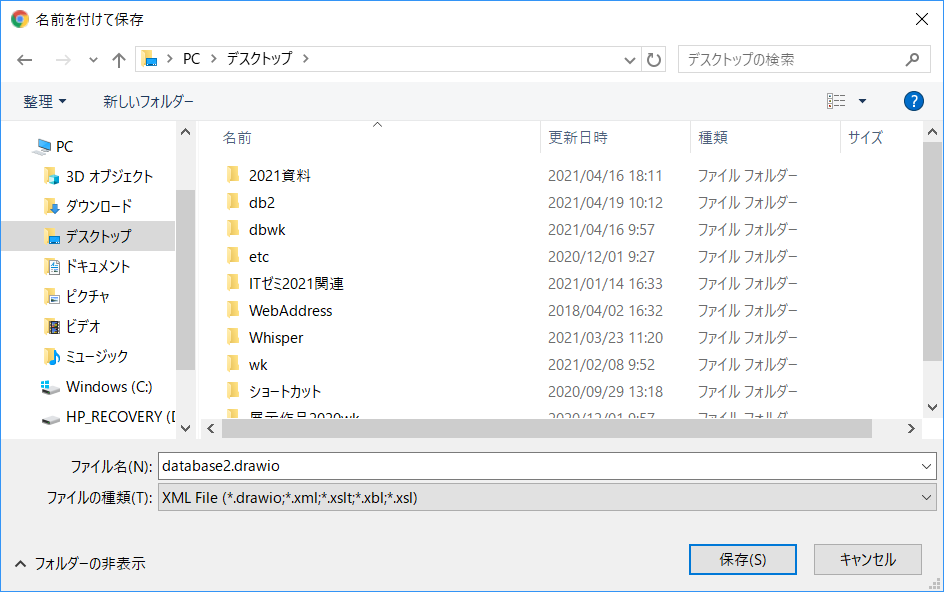
２．ファイルの保存先からデバイスを選択する



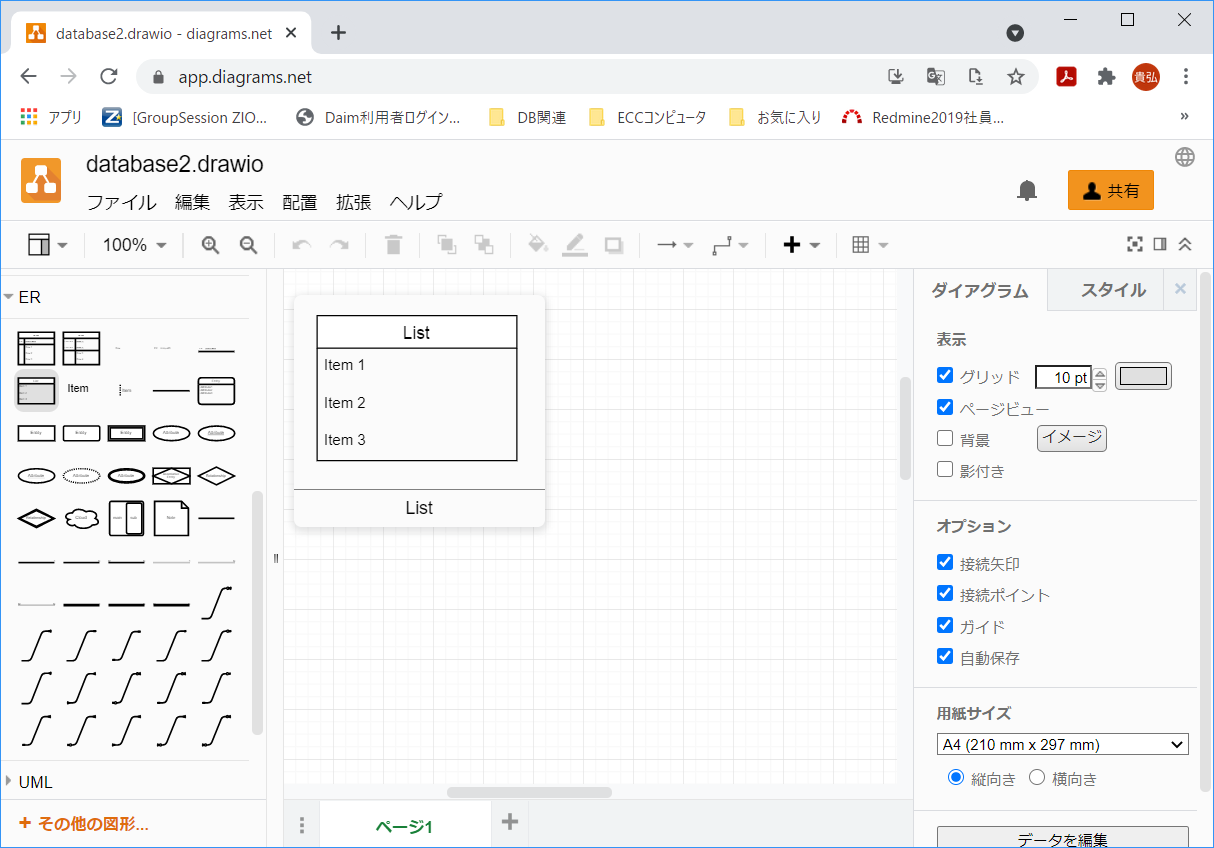
３．白紙ファイルを選択して作成する



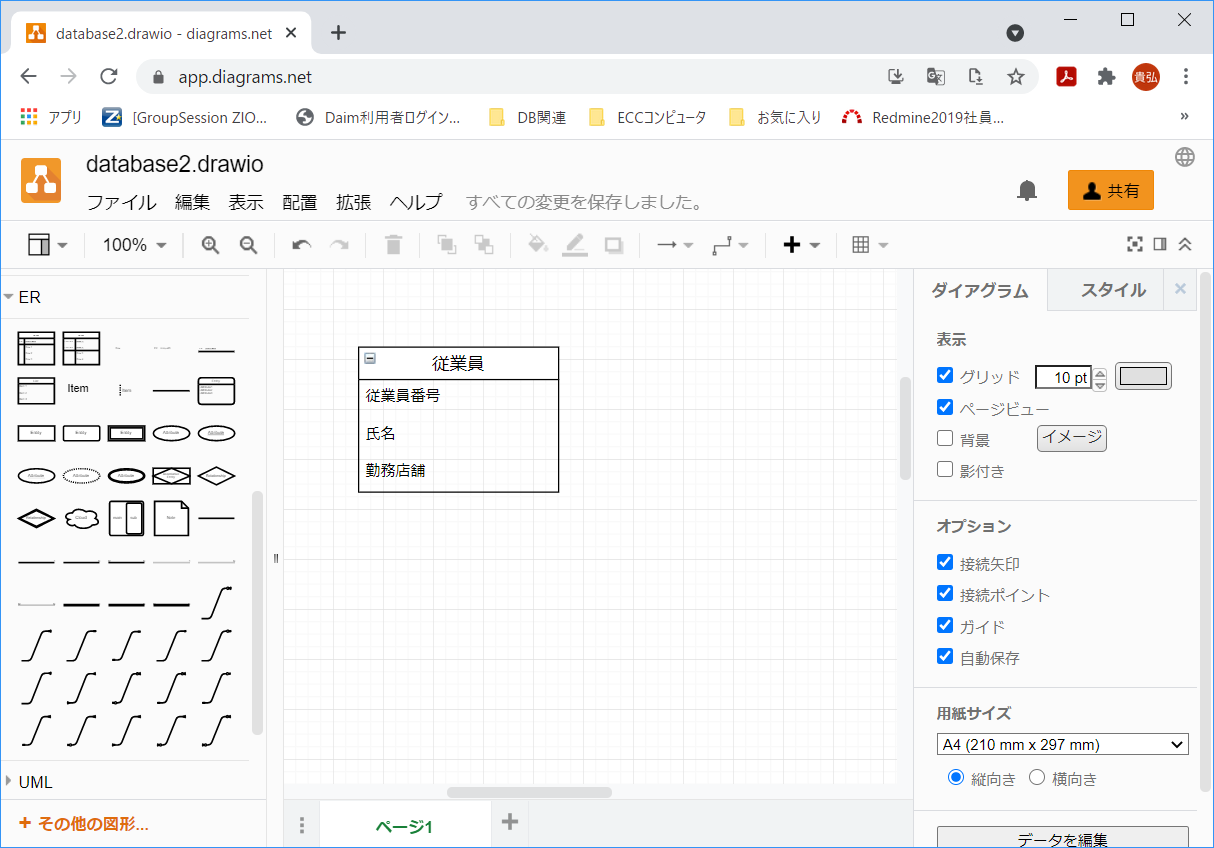
４．ファイル名を「database2」に設定して保存



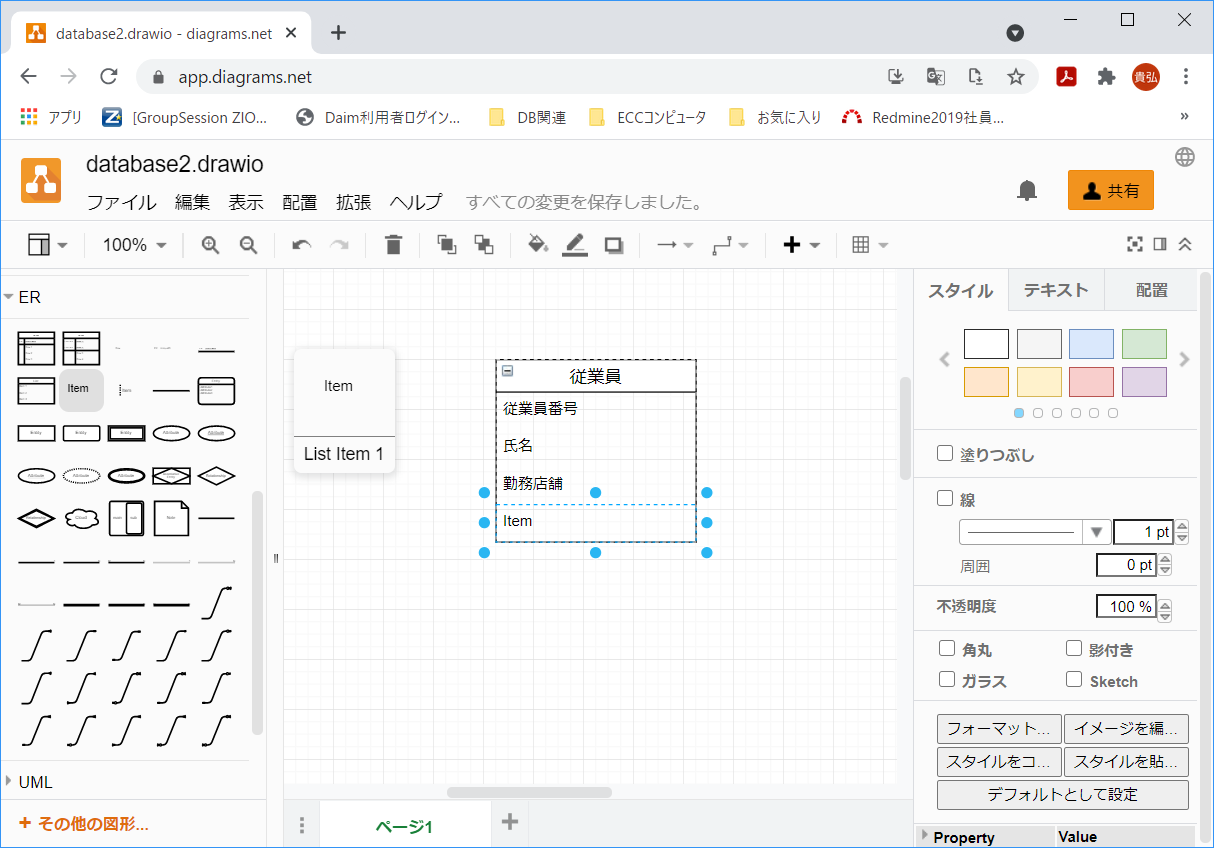
４．左メニューからERタグをクリックして、リストを配置



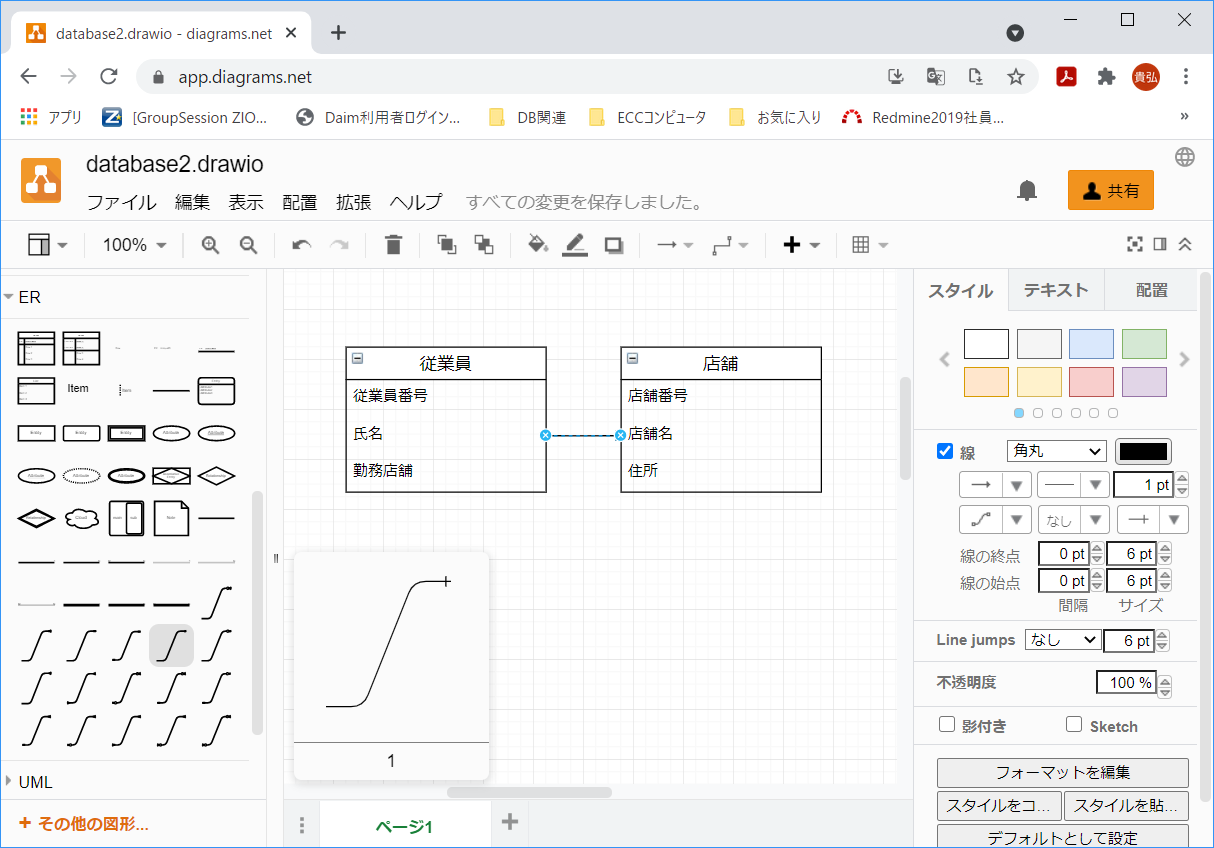
５．エンティティと属性を入力する



６．属性を追加する時はList Itemをエンティティにドラッグ＆ドロップする



７．リレーションを選択してエンティティに接続、パラメータは右メニューで変更する



# エンティティを導きだす方法

顧客からの要件から、「どのようなエンティティが必要か」を導きだすのは、非常に高度な作業です。曖昧な要件に基づいてデータベースの利用イメージを頭の中に広げて、そこに登場する情報を見つけ出さなければならないからです。そこで、要件からエンティティを導きだすヒントを次に紹介します。

【ステップ１】候補となる用語を洗い出す

・要件の中から「名詞」を抜き出す。

・要件が実現されている姿を仮定して、

そこに登場する「人」「物」「事実」「行為」などの用語を書き出す。

【ステップ２】不要な用語を捨てる

・他の用語の具体例でしかないものは排除する。

・計算や集計をすれば算出可能な値は排除する。

【ステップ３】関連がありそうなものをまとめる

・同じ用語に関連するものを集める。

【ステップ４】エンティティ名と属性名に分ける

・ステップ３でまとめたグループの中で「～の～」という日本語が成り立つ場合、

前者がエンティティ名に、後者がその属性名になる。

ハンズオン　商品情報のER図を作成する

以下の、商品管理を行う要件定義から商品情報のER図を作成してみましょう。

【商品管理の要件定義】

・ピザ屋で取り扱っている「マルゲリータ」や「チキンナゲット」などの商品を管理したい。

・商品には、「ピザ」や「ドリンク」などのカテゴリーがある。

・商品は「チーズ」や「トマト」などの複数の原材料から作られる。

・原材料にこだわりがあり、特定の産地の物を使用している。

・原材料の原価と商品の販売価格から利益を知りたい。

・原材料は複数の商品から使用される。

１．候補となる用語を洗い出す。

|  |  |  |  |
| --- | --- | --- | --- |
| 「名詞」、「人」、「物」、「事実」、「行為」などの用語を書き出す | | | |
| ピザ屋 | マルゲリータ | チキンナゲット | ピザ |
| ドリンク | カテゴリー | 原材料 | 産地 |
| 販売価格 | 利益 | チーズ | トマト |

２．不要な用語を捨てる

|  |  |  |
| --- | --- | --- |
| 他の用語の具体例でしかないもの | マルゲリータ | チキンナゲット |
| ピザ | ドリンク |
| チーズ | トマト |
| 計算や集計をすれば算出可能なもの | 利益 |
| 商品管理とは無関係のもの | ピザ屋 |

※具体例のみしか無い場合は名詞に変換する。例）「チーズ」「トマト」=>原材料名

３．関連がありそうなものをあつめる

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| グループ１ | 商品 | 商品名 | カテゴリー | 販売価格 |
| グループ２ | 原材料 | 原材料名 | 産地 | 原価 |

４．エンティティ名と属性名に分ける

|  |  |  |  |
| --- | --- | --- | --- |
| エンティティ | 属性 | | |
| 商品 | 商品名 | カテゴリー | 販売価格 |
| 原材料 | 原材料名 | 産地 | 原価 |

５．draw.ioで商品情報のER図を作成する。

https://app.diagrams.net/

※主キーの設定や多対多の分割は、次項の論理設計で行います。

現段階でのER図を作成してください。

概念設計はかなりあいまいで難しい作業です。データベースの設計や他の人が行った設計の結果を見たり繰り返すうちに、エンティティが浮かぶようになります。

また、今回はトップダウンアプローチで概論設計を行いましたが、**慣れないうちはイメージしやすい出力結果画面を先に考えて、そこから必要な項目を洗い出すボトムアップアプローチで設計**にチャレンジしてみましょう。