データベース設計

ワークの概要

- データベース設計は独学で勉強することが難しく、企業によりローカルルールの違いもあり一概に説明するのは困難ではありますが、こんなことをやっていきますよという基礎を簡単にご紹介したいと思います。

- データベース設計には大きく分けて、「論理設計」と「物理設計」がありますが、このワークでは「論理設計」について解説していきます。

※論理設計とは、システムにおいてどのような情報をデータベースに格納すべきかを検討し、その「格納すべき情報」を「どのような形で保持するか？」を決めるフェーズ。

物理設計とは、論理設計の結果を受けて、データを格納するための物理的な領域や格納方法を決めるフェーズ。

データベースとは？

- **大量の情報**を保存し、コンピュータから**効率よくアクセスできる**ように加工した**データのあつまり**。

- ECサイトやSNSなどのデータを扱うシステムのほとんどすべてにデータベースが使われているため、システムとデータベースは切っても切れない関係。

システム開発のプロセスとデータベース設計

- システム開発のプロセス
  1. 要件定義(なにをつくるか？)
  2. **設計(どうつくるか？)**
  3. 開発(実装)
  4. テスト(期待通りに動くか)

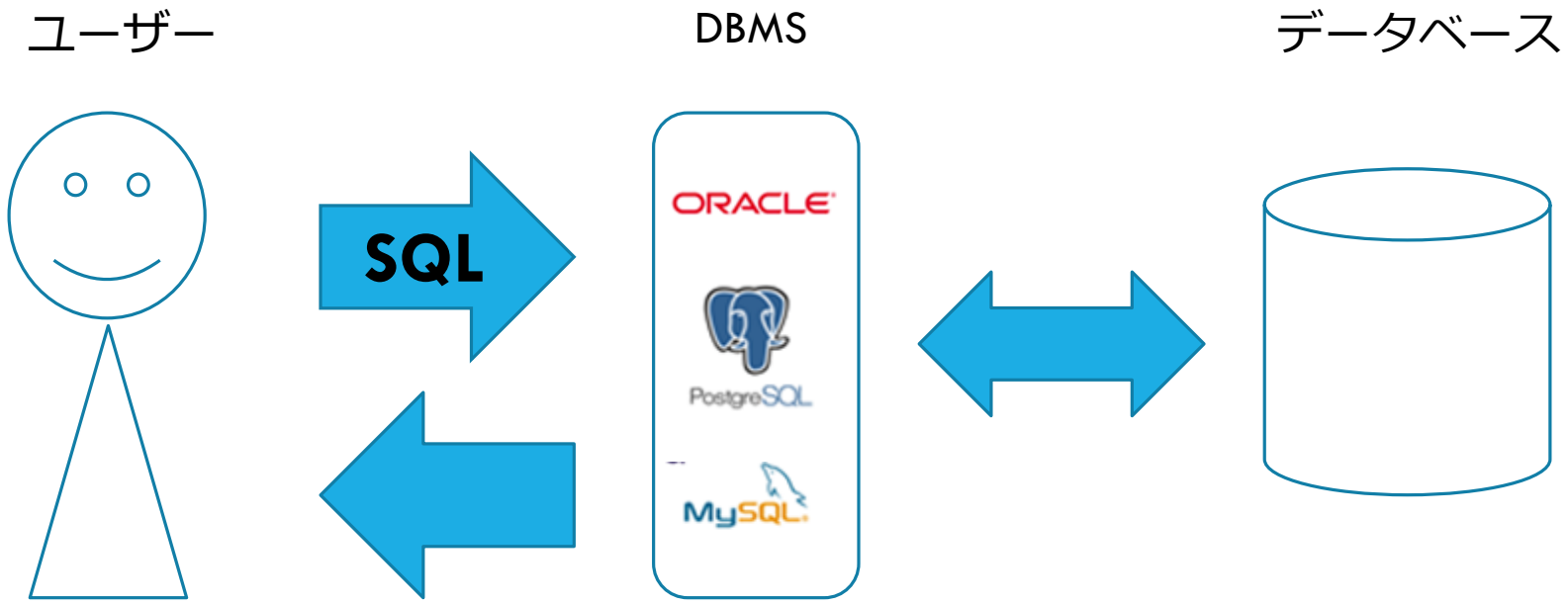
- 設計のプロセスの内訳
  - **データベース設計(データの保持について決める)**
  - アプリケーション設計(提供機能を決める)
  - インターフェース設計(使用画面などを決める)

- データベース設計とは
  - データベースに保持するデータに関する設計
  - システムの拡張性やパフォーマンスに多大な影響を与える、システム開発において極めて重要なプロセス

データベースとDBMS

- 「データベース」は「データの集まり」を指すために使われる言葉で、そのデータベースを管理するためのシステムを「DBMS(Database Management System)」と呼ぶ。
- 代表的なDBMSとして、Oracle Database、MySQL、PostgreSQLなどがある。

※SQLはDBMSに対して指示を行うための言語。



データベースの種類

- 現在、商用で利用されているデータベースには、何種類かあり、その分類はデータを保持するフォーマットを基準にして行われている。

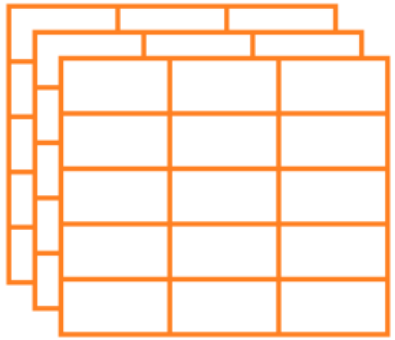
- 代表的なデータベースとして、リレーショナルデータベース、階層型データベース、ネットワーク型データベース、オブジェクト指向データベースの4種類がある。

データベースの種類

リレーショナルデータベース

関係データベースとも呼ばれ、現在広く利用されているデータベース。

特徴としては、データを人間が理解しやすい2次元の形式で管理するため、データの取り扱いが他のデータベースに比べると直感的で簡単であること。

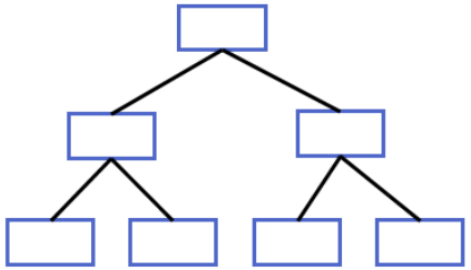


関係データベース

階層型データベース

データの階層関係に着目してデータを表現するデータベースで、データを親子関係として表現する木構造になっている。

一世代前の主流データベースであったが、現在はリレーショナルデータベースの普及に伴い、あまり一般的には使用されなくなってきている。



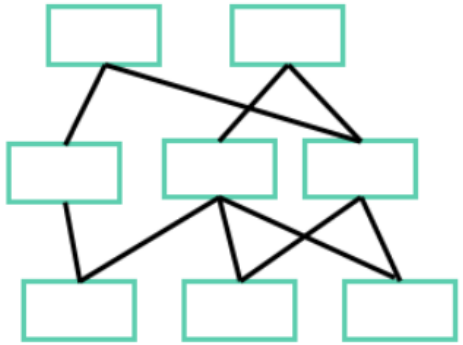
階層型データベース

データベースの種類

ネットワーク型データベース

データを網の目のような構造で表すのがネットワーク型データベース。

こちらは先ほどの階層型データベースで表現できない子レコードが複数の親レコードを持つことができる。

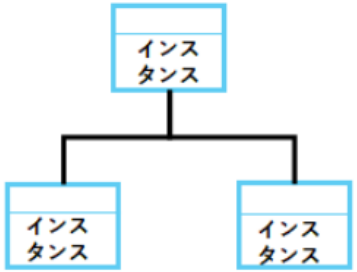


ネットワーク型データベース

オブジェクト指向データベース

JavaやC++など、オブジェクト指向言語と呼ばれるタイプの言語があり、データと操作をまとめて「オブジェクト」という単位で管理するためにそう呼ばれている。

このオブジェクトをデータベースに保存するために作られたのがオブジェクト指向データベースである。



オブジェクト指向データベース

問題1

- データベースの論理的構造を規定した論理データモデルのうち，関係データモデルの説明として適切なものはどれか。(ITパスポート H26年秋期)

ア. データとデータの処理方法を，ひとまとめにしたオブジェクトとして表現する。

イ. データ同士の関係を網の目のようにつながった状態で表現する。

ウ. データ同士の関係を木構造で表現する。

エ. データの集まりを表形式で表現する。

答え

ア. オブジェクト指向におけるクラス化の説明です。

イ. ネットワーク型データベースの説明です。

ウ. 階層型データベースの説明です。

エ. 正しい。関係データベースの説明です。

テーブルとは？

- リレーショナルデータベースでは、あらゆるデータを「テーブル」という単位で扱う。
  テーブルは、データを保存しておくための「表」にあたるもの。
- 例えば、お買い物サイトを開発する例で考えてみると、

登録した会員情報や、販売している商品情報を保存しておく必要が出てきた。

そこで、「Shopping」という名前のデータベースを作成して、そこに

・会員情報を保存しておくテーブル（名前: users）

・商品情報を保存しておくテーブル（名前: items）

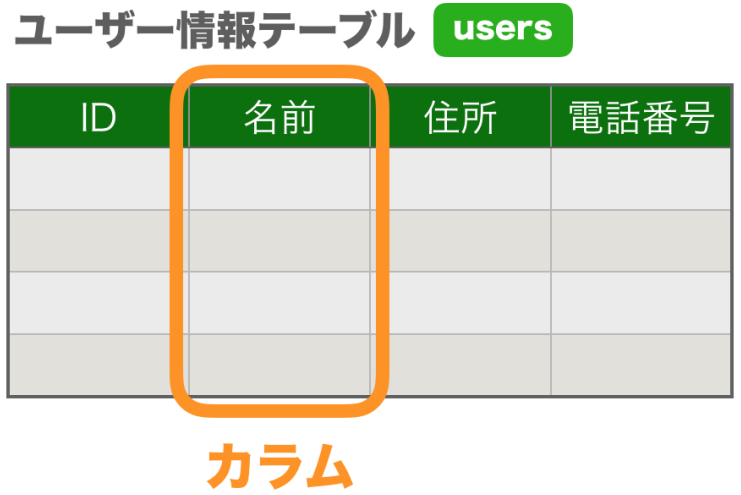
保存しておくためのテーブルを用意することにした。

右図のように、特定の種類のデータ一覧を保存することができるのがテーブルである。一つのデータベースの中には複数のテーブルを作成することができる。

テーブルの構成要素（カラム・レコード）

➢カラム

テーブルの「列」にあたるもの。Usersテーブルの中には一人一人のユーザIDや名前、住所、電話番号の項目を保存出来るようにする。それぞれの項目にあたる「列」がカラム。



➢レコード

テーブルの「行」にあたるもの。Usersテーブル名前に「山田」、「佐藤」、「鈴木」、「田中」の4人分のデータが保存されている場合、この1件分のデータにあたる「行」がレコード。

テーブルの構成要素（フィールド）

- **フィールド**

レコードの中の「入力項目」にあたるもの。 Users テーブル名前に「山田」、「佐藤」、「鈴木」、「田中」の4人分のデータが保存されている場合、一人一人のレコードの要素にあたる「入力項目が」フィールド。

ユーザー情報テーブル users

ID	名前	住所	電話番号
1	山田		
2	佐藤		
3	鈴木		
4	田中		

フィールド

テーブルの構成要素（キー）

- ある特定のデータを引き出すための鍵となる列
- キーにはいくつか種類があるが、特に重要なものとして**主キー**と**外部キー**がある

- **主キー(primary key)**

・その値を指定すれば、必ず1行のレコードを特定できるような列(の組み合わせ)

・テーブルにおいて**必ず一つ存在する**必要があり、かつ、**一つしか存在しない**

・右のテーブルにおいては社員IDによってレコードを一意に定めることができるので、社員IDが主キーとして機能していることになる

・名前と年齢はこの3行だけみると重複がないので良さそうにも見えるが、将来的にレコードが増えた場合に重複の可能性があり、主キーとして適切でない

社員

社員ID	名前	年齢
1	佐藤	20
2	鈴木	30
3	田中	40

主キー

- 主キーの定義から「テーブルに重複行は存在しない」というルールも導かれる
- 重複した行があれば、主キーによってレコードを一意に定めることができない
- 主キーはNULL(フィールドにデータが入っていない状態)を禁止する制約がある(NOT NULL制約)
- 複数列を組み合わせなければ主キーが作れないこともある。こうした複数列を組み合わせて作るキーを**複合キー**と言う。

社員

社員ID	名前	年齢
1	佐藤	20
1	佐藤	20
2	田中	40

社員

会社ID	社員ID	名前	部署
1	1	佐藤	3
1	2	鈴木	2
2	1	佐藤	2

外部キー

- **2つのテーブル間の列同士の関連性を設定するもの**
- 以下の2つのテーブルでは、社員テーブルの「部署ID」列が「外部キー」にあたる

社員

社員ID	名前	部署ID
1	佐藤	1
2	佐藤	3
3	田中	2

部署

部署ID	部署名
1	営業
2	開発
3	経理

- 社員テーブルの部署IDカラムはあくまでも部署テーブルの部署IDカラムを参照しているにすぎないので、部署テーブルにない部署IDを社員テーブルに入れることはできない(例えば、社員テーブルの部署IDに4を入れることはできない)
- 存在しない部署データを社員テーブルに登録してしまうことを防止できる
- 社員テーブルに対して「**制約**」を課している(非参照整合性制約)

問題2

- リレーショナルデータベースの主キーに関する記述のうち，適切なものはどれか。

(ITパスポート平成21年秋期)

ア. 関係データベースの各表は，主キーだけで関係付けられる。

イ. 主キーとして指定した項目は，NULLを属性値としてもつことができる。

ウ. 一つの表において，主キーとして指定した項目の値に同一のものがあってもよい。

エ. 一つの表において，複数の項目を組み合わせて主キーとしてもよい。

答え

ア. 関係データベースの各表は、主キー及び外部表の主キーを参照する外部キーの両者によって関連つけられます。

イ. 主キー列の項目には、NULL(空値)であってはならないという制約(NOT NULL制約)があります。

ウ.主キー列の項目には、同じ列中の値の中で一意でなくてはならないという制約(UNIQUE制約)があります。

エ. 正しい。複数の列の値の組合せを主キーとすることがあります。(複合キー)

問題3

- 関係データベースを使い“社員”表と“部署”表を作成して社員情報を管理する。“社員”表と“部署”表に、必要に応じて設定する主キーと外部キーの適切な組合せはどれか。

ここで、社員は必ず“部署”表に存在する部署に所属するものとし、社員データの追加や更新をするときには、参照制約を利用して整合性を確保するものとする。

(ITパスポート平成25年春期)

社員

社員コード	社員名	入社年	生年月日	部署コード

部署

部署コード	部署名

	主キー	外部キー
ア	“社員”表の社員コード, “部署”表の部署コード	なし
イ	“社員”表の社員コード, “部署”表の部署コード	“社員”表の部署コード
ウ	“部署”表の部署コード	“社員”表の社員コード, “社員”表の部署コード
エ	“社員”表の部署コード	“社員”表の社員コード, “部署”表の部署コード

答え

正解は、イ

　主キーは、表内の1行を特定できる属性なので、行ごと異なる番号が振られた社員表の社員コード、部署表の部署コードになります。

　また、社員表と部署表は部署コードを通じて関係があるため、部署表の主キーを参照している社員表の部署コードが外部キーになります。

問題4

- 関係データベース"注文"表の"顧客番号"は"顧客"表の主キー"顧客番号"に対応する外部キーである。このとき、参照の整合性を損なうデータ操作はどれか。ここで、ア～エの記述におけるデータの並びは、それぞれの表の列の並びと同順とする。

(基本情報技術者平成24年秋期)

注文

伝票番号	顧客番号
0001	C005
0002	K001
0003	C005
0004	D010

顧客

顧客番号	顧客名
C005	福島
D010	千葉
K001	長野
L035	宮崎

ア. “顧客”表の行 [L035 宮崎] を削除する。

イ. “注文”表に行 [0005 D010] を追加する。

ウ. “注文”表に行 [0006 F020] を追加する。

エ. “注文”表の行 [0002 K001] を削除する。

答え

ア. "注文"表の"顧客番号"列には、"LO35"の値をもつ行がないので削除可能です。

イ. "顧客番号"が"D010"の行は、顧客表にすでに存在しているので、問題なく追加できます。

ウ. **正しい。**"顧客番号"が"F020"の行は、"顧客"表に存在していないので追加できません。このようなときには先に"顧客"表にレコードを追加してから"注文"表への追加を行う必要があります。

エ.この問題の場合、"注文"表の行の削除は特に問題になることはありません。

データベース設計のプロセス

- データベース設計には4つのプロセスがあります。

①エンティティの抽出
　どんなデータを管理するエンティティ(テーブル)が必要かを明確にする
②エンティティの定義
　エンティティ(テーブル)ごとにどんな属性(カラム)が必要かを明確にする
③正規化
　テーブルを分割することでデータの冗長性(ムダ・重複)をなくす
④ER図(やらなくてもOK)
　エンティティ(テーブル)間の関係を視覚的にわかりやすく作図する

エンティティ

- エンティティとは、**システムにおいて管理する必要があるデータ**のこと。

- エンティティ(Entity)は日本語で「実体」という意味。
  具体的には、**顧客・社員・店舗**のような物理的実態を伴うものもあれば、
  注文・予約というような概念しか存在しないものもある。

- リレーショナルデータベースでは、**テーブルの名前になるもの**という理解でも良い。

エンティティの抽出・定義

エンティティの抽出

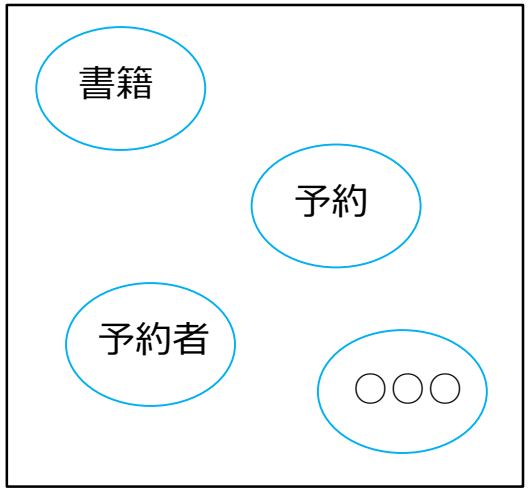
・実現したいシステムで、どんなエンティティを管理する必要があるか？を洗い出すプロセス

・例えば、「本の予約管理システム」に必要なエンティティとして考えられるのは、**書籍・予約・予約者**などのエンティティ(システムの詳細によって異なる)

・実現したいシステムが果たす機能の一連の流れを踏まえて抽出していく

・どんなシステムを作るか？にかなり依存するので、**要件定義と重なるプロセス**である

本の予約管理システム



エンティティの定義

・各エンティティがどのような属性(データ)を保持するか決めるプロセス

・リレーショナルデータベースでは「テーブル」という形でエンティティを保持するが、各テーブルがどのような「列」を持つのか定義するプロセスとも言える

・特に重要なのが「キー」を特定すること

予約

予約番号	予約者名	日にち
1	佐藤	2020-04-01
2	山本	2020-04-02
3	田中	2020-04-03

正規化

- データベースのデータ構造をより効率的で、重複や無駄のないシンプルな構造にするための手順
- 正規化の目的は、更新時のトラブルを起こりにくくすること

例) 開発部が中央区から西区に移転することになったので、該当する所在カラムを更新する

社員ID	名前	部署ID	部署名	所在
1	佐藤	1	開発	大阪府大阪市中央区
2	鈴木	3	営業	大阪府大阪市北区
3	田中	2	総務	大阪府大阪市北区
4	山田	1	開発	大阪府大阪市中央区

所在
大阪府大阪市西区
大阪府大阪市北区
大阪府大阪市北区
大阪府大阪市中央区

- データに冗長な部分があると、このように更新漏れなどが発生し、データに矛盾が生じてしまうというトラブルが発生してしまう

※冗長とは、データが重複していて無駄であるという意味

正規化のデメリット

- 正規化を進めれば進めるほど、小さなテーブルが増えてしまい、パフォーマンスが低下してしまう。

　テーブルが増えれば関連するテーブルを検索するたびにSQLを実行する回数が増え、速度が低下するため。

- 場合によっては正規化せずに性能を向上させる設計も必要。

問題4

- データの正規化に関する記述のうち，適切なものはどれか。

(基本情報技術者平成18年秋期)

ア. 関係データベースに特有なデータベース構築技法であり，データの信頼性と格納効率を向上させる。

イ. データの重複や矛盾を排除して，データベースの論理的なデータ構造を導き出す。

ウ. データベースの運用管理を容易にするために，レコードをできるだけ短く分割する。

エ. ファイルに格納するデータの冗長性をなくすることによって，アクセス効率を向上させる。

答え

ア. 目的はデータに矛盾や重複を生じさせてないことです

イ. 正しい

ウ. 目的は保全性のためではありません

エ. 目的はデータに矛盾や重複を生じさせてないことです

正規形

- 正規化された表のこと
- 正規形にするための主な作業は、テーブルにキーを設定し、余分なカラムを整理し、適切にテーブルを**分割**すること
- 正規形には第1～第5正規形まで存在するが、業務では第3正規形まで考えることが多い
- 第3正規形まで達成したらその時点で以降の正規形の条件も満たしていることがほとんど

第1正規形

- 「１つのフィールドには１つの値しか含まない」という原則が守られた形
- なぜ１つのフィールドに複数の値が含まれることが認められないのか？
  ⇒複数の値を許すと、主キーが各列の値を一意に定めることができなくなる
  ⇒主キーの定義に反する
- 以下のようなテーブルは許されないので、正規化する必要がある

扶養者

社員ID	名前	子供
1	佐藤	太郎 花子
2	田中	
3	鈴木	次郎 三郎

第1正規形

- フィールド内の値をひとつにする方法として行に分ける方法と列に分ける方法があるが、どちらも問題がある

行に分ける方法

・主キーを定められない

・主キーにNULL禁止

社員ID	名前	子供
1	佐藤	太郎
1	佐藤	花子
2	田中	
3	鈴木	次郎
3	鈴木	三郎

列に分ける方法

・子供列を何個用意すれば良いかわからない

・大量のNULL発生

社員ID	名前	子供１	子供２
1	佐藤	太郎	花子
2	田中		
3	鈴木	次郎	三郎

第1正規形と関数従属性

- 右のようにテーブルを分割することで、第一正規形の形とテーブルとしての定義を満たすことができる

- 正規化を扱う上で「関数従属性」という概念が重要になる
- **関数従属性とは？**

・y=f(x)というようにxの値を1つ決めればyの値が1つに決まる関係性のこと

・例えばx=5とするとy=10に定まるというような関係

⇒この関係が成立している状態を「yはxに従属する」という

⇒このとき記号では「{x}→{y}」というように表す

・言い換えるならば正規化とは「**テーブルのすべての関数従属性を満たすように整理していくこと**」である

・{主キー}→{カラム}がすべてのカラムにおいて成立している状態を目指す

社員

社員ID	名前
1	佐藤
2	田中
3	鈴木

扶養者

社員ID	子供
1	太郎
1	花子
3	次郎
3	三郎

第２正規形

- **部分関数従属が解消されていて、完全関数従属のみのテーブルになっている形**

- **部分関数従属とは？**

　複数列からなる主キーの一部の列に対して従属する列がある状態

- **完全関数従属とは？**

　主キーを構成するすべての列に対して従属性がある状態

右のテーブルの主キーは{会社ID,社員ID}なので、すべてのカラムはこのキーに従属するが、「会社名」列は「会社ID」列にのみ従属している({会社ID}→{会社名})

社員

会社ID	会社名	社員ID	名前	部署ID	部署名
1	A銀行	1	佐藤	3	経理
1	A銀行	2	鈴木	2	開発
1	A銀行	3	田中	1	営業
2	B建設	1	佐藤	2	開発
2	B建設	2	鈴木	1	営業
2	B建設	3	田中	4	人事

第２正規形

- {会社ID}→{会社名}の部分関数従属を解消すると以下の2つのテーブルになる

社員

会社ID	社員ID	名前	部署ID	部署名
1	1	佐藤	3	経理
1	2	鈴木	2	開発
1	3	田中	1	営業
2	1	佐藤	2	開発
2	2	鈴木	1	営業
2	3	田中	4	人事

会社

会社ID	会社名
1	A銀行
2	B建設

- 第2正規形は「社員」と「会社」という**異なるレベルのエンティティを、きちんとテーブルとしても分離してあげる作業**ともいえる

第３正規形

- **推移的関数従属が解消されている形**

- **推移的関数従属とは？**

　2段階の関数従属がある状態

- 社員テーブルには以下のような問題がある

「部署として存在しているが、社員がいない部署をテーブルに反映できない」

一時的な欠員のため社員がいない「総務」という部署があるかもしれないが、それが反映できない構造になっている

- この原因は**推移的関数従属**によるもの

　{部署ID}→{部署名}、{会社ID,社員ID}→{部署ID}

という2つの関数従属が成立しているため、全体としては

　{会社ID,社員ID}→{部署ID}→{部署名}

という**2段階の関数従属**が成立している

社員

会社ID	社員ID	名前	部署ID	部署名
1	1	佐藤	3	経理
1	2	鈴木	2	開発
1	3	田中	1	営業
2	1	佐藤	2	開発
2	2	鈴木	1	営業
2	3	田中	4	人事

第3正規形

- これを解消するには以下のようにテーブルを分割する

社員

会社ID	社員ID	名前	部署ID
1	1	佐藤	3
1	2	鈴木	2
1	3	田中	1
2	1	佐藤	2
2	2	鈴木	1
2	3	田中	4

会社

会社ID	会社名
1	A銀行
2	B建設

部署

部署ID	部署名
1	営業
2	開発
3	経理
4	人事
5	総務

- 第3正規形は「社員」と「部署」という**異なるレベルのエンティティを、きちんとテーブルとしても分離してあげる作業**とみれば、第2正規形と同じ意味を持っているといえる

正規化のまとめ

- 正規化とは、テーブルのすべての列が関数従属を満たすように整理することにより、データの冗長性を排除すること

- 第1正規形は**1つのフィールドには1つの値に**すること
- 第2正規形は**部分従属性の解消**
- 第3正規形は**推移的関数従属の解消**

- 正規化したテーブルはJOINによる結合で正規化前の状態に戻せる

  ⇒「無損失分解」という、正規化の逆操作は結合であるともいえる

問題5

- 次の受注データを、受注に関する情報と商品に関する情報に分割し正規化を行った結果、表はどのように分割されるでしょうか。ここで、単価は商品番号により一意に決定されるものとします。

受注番号	発注者名	商品番号	商品名	個数	単価
T0001	山田花子	M0001	鉛筆	5	100
T0002	木村太郎	M0002	消しゴム	3	50
T0003	佐藤明子	M0003	ノート	2	300

答え

- 第2正規形まで整理されているので、第3正規形にしていきます。
- {受注番号}→{商品番号}→{商品名, 単価}という推移的関数従属を解消する必要があります。

受注番号	発注者名	商品番号	商品名	個数	単価
T0001	山田花子	M0001	鉛筆	5	100
T0002	木村太郎	M0002	消しゴム	3	50
T0003	佐藤明子	M0003	ノート	2	300

答え

- 以下のように分割することで、第3正規形にすることができます。

受注

受注番号	発注者名	商品番号	個数
T0001	山田花子	M0001	5
T0002	木村太郎	M0002	3
T0003	佐藤明子	M0003	2

商品

商品番号	商品名	単価
M0001	鉛筆	100
M0002	消しゴム	50
M0003	ノート	300

問題6

学生の科目履修に関する「学生コード、学生名、科目コード、科目名、履修年度、成績」という情報を、一つの表に記録したとします。

このデータベースを整理してより効率的で使いやすいものにするために正規化を行い、幾つかの表に分割してください。

学生コード	学生名	科目コード	科目名	履修年度	成績
1001	佐藤 太郎	K01	国語	2017	優
1001	佐藤 太郎	K02	数学	2017	良
1001	佐藤 太郎	K03	英語	2017	可
1001	佐藤 太郎	K04	社会	2018	優
1002	田中 次郎	K01	国語	2019	良
1002	田中 次郎	K02	英語	2018	可
1003	鈴木 花子	K03	数学	2019	優

答え

- 既に第1正規形を満たしているので次は第2正規形にするために、部分関数従属を解消します。学生名は学生コードにのみ従属しており、科目名は科目コードにのみ従属してるので、分割する必要があります。

学生コード	学生名	科目コード	科目名	履修年度	成績
1001	佐藤 太郎	K01	国語	2017	優
1001	佐藤 太郎	K02	数学	2017	良
1001	佐藤 太郎	K03	英語	2017	可
1001	佐藤 太郎	K04	社会	2018	優
1002	田中 次郎	K01	国語	2019	良
1002	田中 次郎	K02	英語	2018	可
1003	鈴木 花子	K03	数学	2019	優

答え

- 推移的関数従属は存在しないため、以下は第3正規形を満たしています。

学生

学生コード	学生名
1001	佐藤 太郎
1002	田中 次郎
1003	鈴木 花子

科目

科目コード	科目名
K01	国語
K02	数学
K03	英語
K04	社会

履修

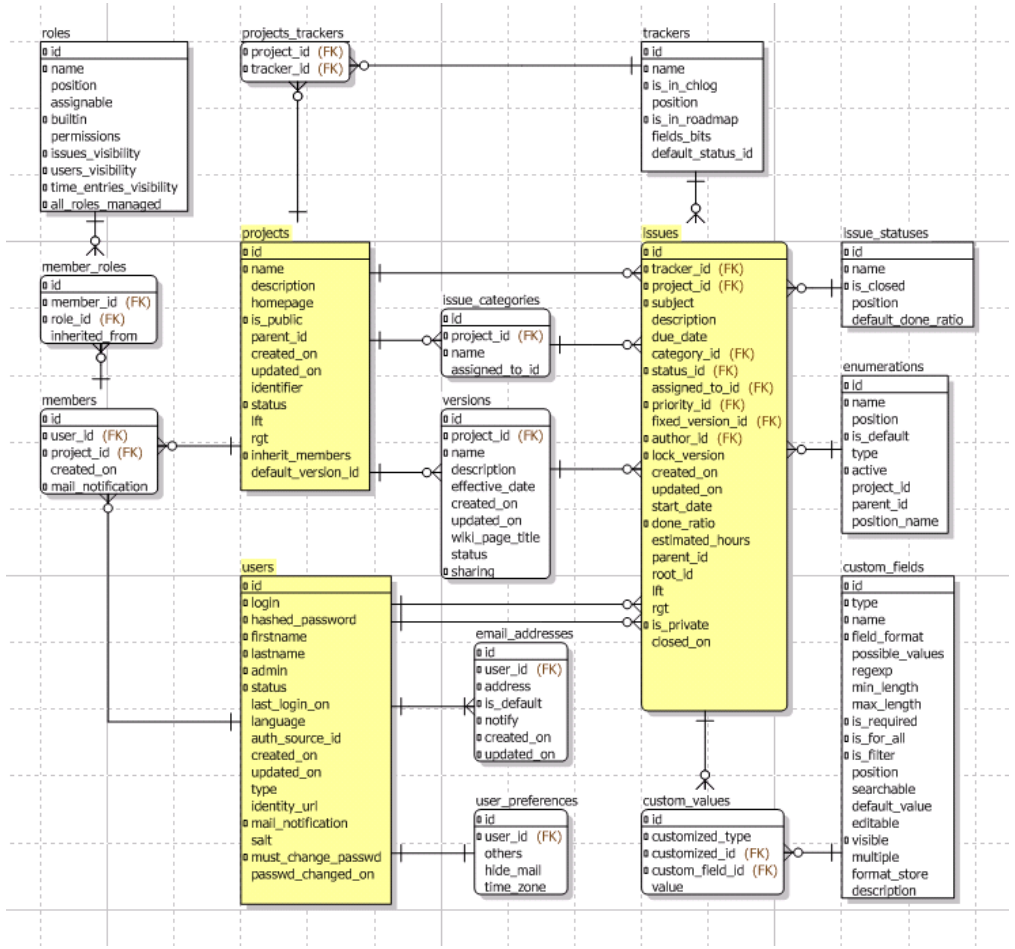
学生コード	科目コード	履修年度	成績
1001	K01	2017	優
1001	K02	2017	良
1001	K03	2017	可
1001	K04	2018	優
1002	K01	2019	良
1002	K02	2018	可
1003	K03	2019	優

ER図

- 正規化によってテーブルを整理していくと、テーブルの数が増えていきます。実際の業務システムを開発すると、何百という数のテーブルが作られることも珍しくありません。

- こうした多数のテーブルを管理するために、それぞれのテーブルがどういう意味を持っていて、**テーブル同士が互いにどういう関係にあるのか**を明示するために作る図をER図(Entity-Relationship Diagram：実態関連図)と呼びます。

- ER図にはいくつか種類が存在しますが、基本的な考え方は大きく変わるものではありません。今回は、直感的にテーブルの関連を理解しやすい「**IE表記法**」について解説します。



ER図の基本

- **エンティティ**

データのまとまり。エンティティの中に、属性情報がある。

- **アトリビュート(属性)**

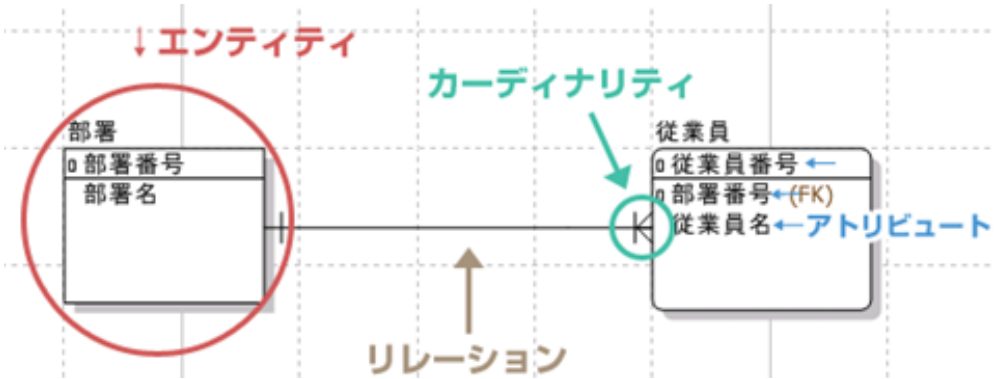
エンティティの中の属性情報。この中で、主キーや外部キーも表現する。

- **リレーション**

エンティティ同士の関係を表現する線。リレーションの詳細は記号を使って表現する。

- **カーディナリティ(多重度)**

「1対1」、「1対多」、「多対多」など、リレーションの詳細を表現する。



ER図の基本

➢リレーションの記号

・リレーションの線には、エンティティが関連する最小件数(オプショナリティ)や最大件数(カーディナリティ)を示すための記号を使用する。

記号	簡単な概要
○	0 (オプショナリティで使う)
|	1
𓃰	多 (たくさん。カーディナリティで使う)

・リレーションと上記の記号を組み合わせると下記のような記号になる

記号	最小 (オプショナリティ)	最大 (カーディナリティ)	意味
	0 (○)	1 (|)	0か1件のみ関連 (0～1)
	1 (|)	1 (|)	必ず1件のみ関連 (1～1)
	0 (○)	多 (𓃰)	0件以上関連 (0～多)
	1 (|)	多 (𓃰)	必ず1件以上関連 (1～多)

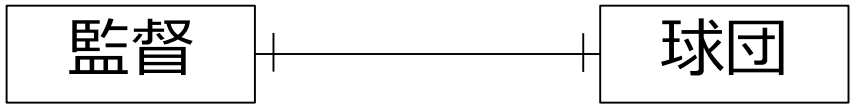
カーディナリティ(多重度)

1対1

・**監督と球団**

・一人の監督は一つの球団と契約する

・一つの球団は一人の監督と契約する



監督

監督番号	名前
81	工藤
83	原
88	矢野

球団

球団記号	球団名
H	福岡ソフトバンクホークス
G	読売ジャイアンツ
T	阪神タイガース

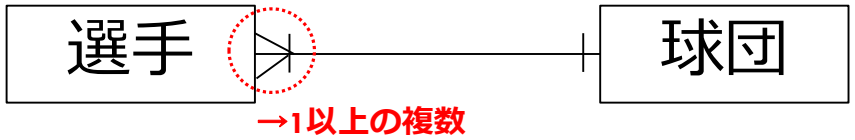
カーディナリティ(多重度)

1対多

・**選手と球団**

・一人の選手は一つの球団に所属する

・一つの球団には複数の選手が所属する



選手

選手番号	名前
9	柳田
6	今宮
22	藤川

球団

球団記号	球団名
H	福岡ソフトバンクホークス
T	阪神タイガース

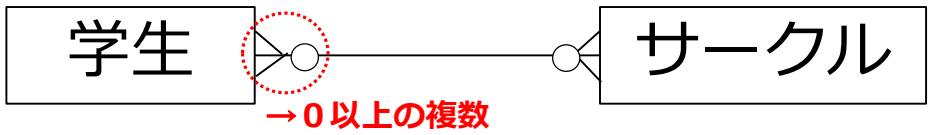
カーディナリティ(多重度)

多対多

・**学生とサークル**

・一人の学生は複数のサークルに所属する

・一つのサークルには複数の学生が所属する



学生

学生番号	名前
1	柳田
2	今宮
3	藤川
4	和田
5	内川

サークル

サークル番号	サークル名
1	野球
2	バレーボール
3	軽音楽
4	ダンス
5	卓球

ER図の作成

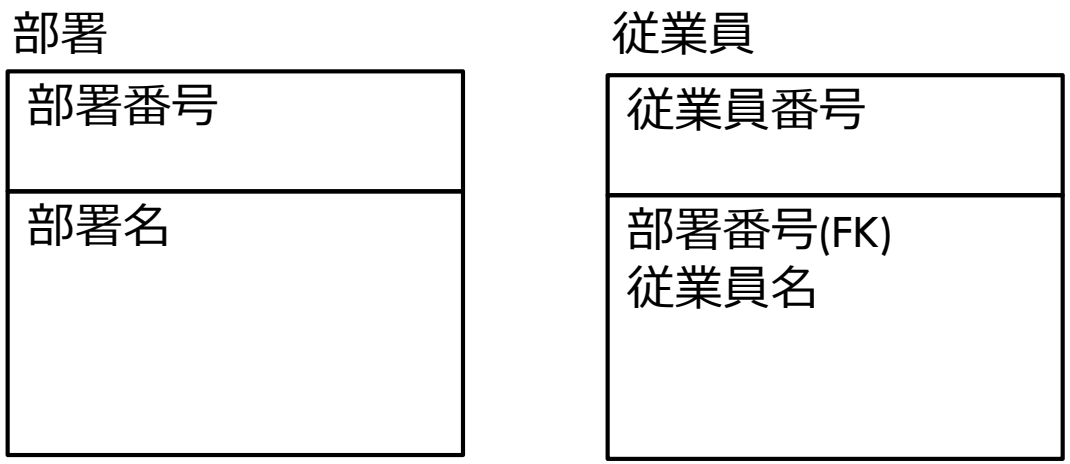
- 例として、部署テーブルと社員テーブルのER図を作成します。ER図の作成は、エンティティを作る → 依存関係を調べる → リレーションをはるの順番で行います。

- エンティティを作る

「部署」と「社員」のエンティティを作成します。

四角の中を横線で区切って、上のスペースに主キーを、下のスペースに非キーを記述します。

他のテーブルの主キーを参照する外部キー(foreign key)については、略称の「FK」を記述します。



ER図の作成

➢依存関係を調べる

部署テーブルと社員テーブルの関係を考えます。

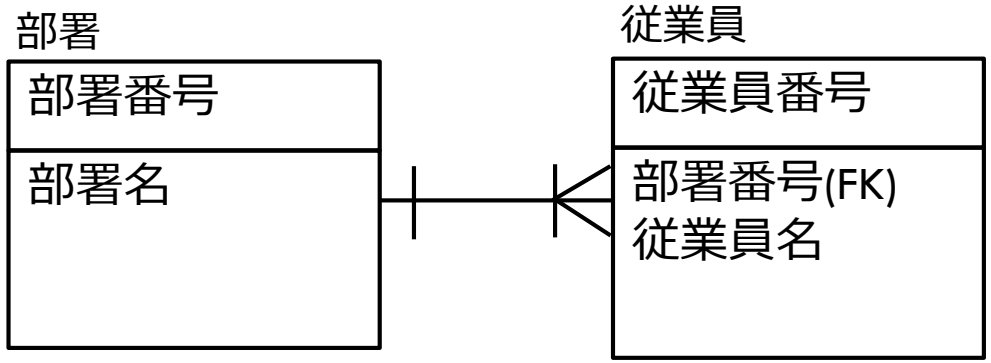
・一つの部署には1人以上の複数の社員がいる。

・社員はどこか一つの部署に所属している。

このことから、部署テーブルと社員テーブルの関係は「１対多」であること考えます。

➢リレーションをはる

二つのエンティティにリレーションをはります。部署テーブルと社員テーブルの関係は「１対多(1対１以上の複数)」の関係になるので、図のようになります。



問題6

- E-R図の説明と、その応用例として、適切なものはどれか。

(ITパスポート試験平成２４年度春期試験)

ア. 作業順序や作業時間を表した図であり、例えば、システム開発の日程管理をするのに用いられる。

イ. 実体同士の関連を表した図であり、例えば、関係データベースの表同士の関連を表すのに用いられる。

ウ. 順次、選択、繰返し構造を組み合わせて表した図であり、例えば、プログラムの流れを記述するのに用いられる。

エ. 状態の遷移や条件を記載した図であり、例えば、通信プロトコルの仕様を記述するのに用いられる。

答え

ア. アローダイアグラム(作業工程を表現した図)の説明です。

イ.正しい。

ウ. フローチャート(プログラムの流れを設計するための図)の説明です。

エ. 状態遷移図(状態が遷移していく様子を表現する図)の説明です。

問題7

- お弁当屋さんの注文管理システムのデータベースには、「顧客」、「商品」、「注文」、「注文商品内訳」の4つのテーブルがあります。この4つのテーブルのER図をIE表記法で作成してください。それぞれぞれテーブルの関係は以下の通りです。
  - 1人の顧客は1つ以上の商品を複数の注文を行うことができる
  - 1つの商品は1つ以上の複数注文できることができる
  - 注文商品内訳は注文データをもとに、注文した商品とその数量を管理している

顧客

顧客番号	顧客名
1	未来のかたち本校
2	未来のかたち2校
3	株式会社かたち

商品

商品番号	商品名	値段
1	からあげ弁当	500円
2	ハンバーグ弁当	650円
3	のり弁当	400円
4	ミックスフライ弁当	750円
5	とんかつ弁当	800円

問題7

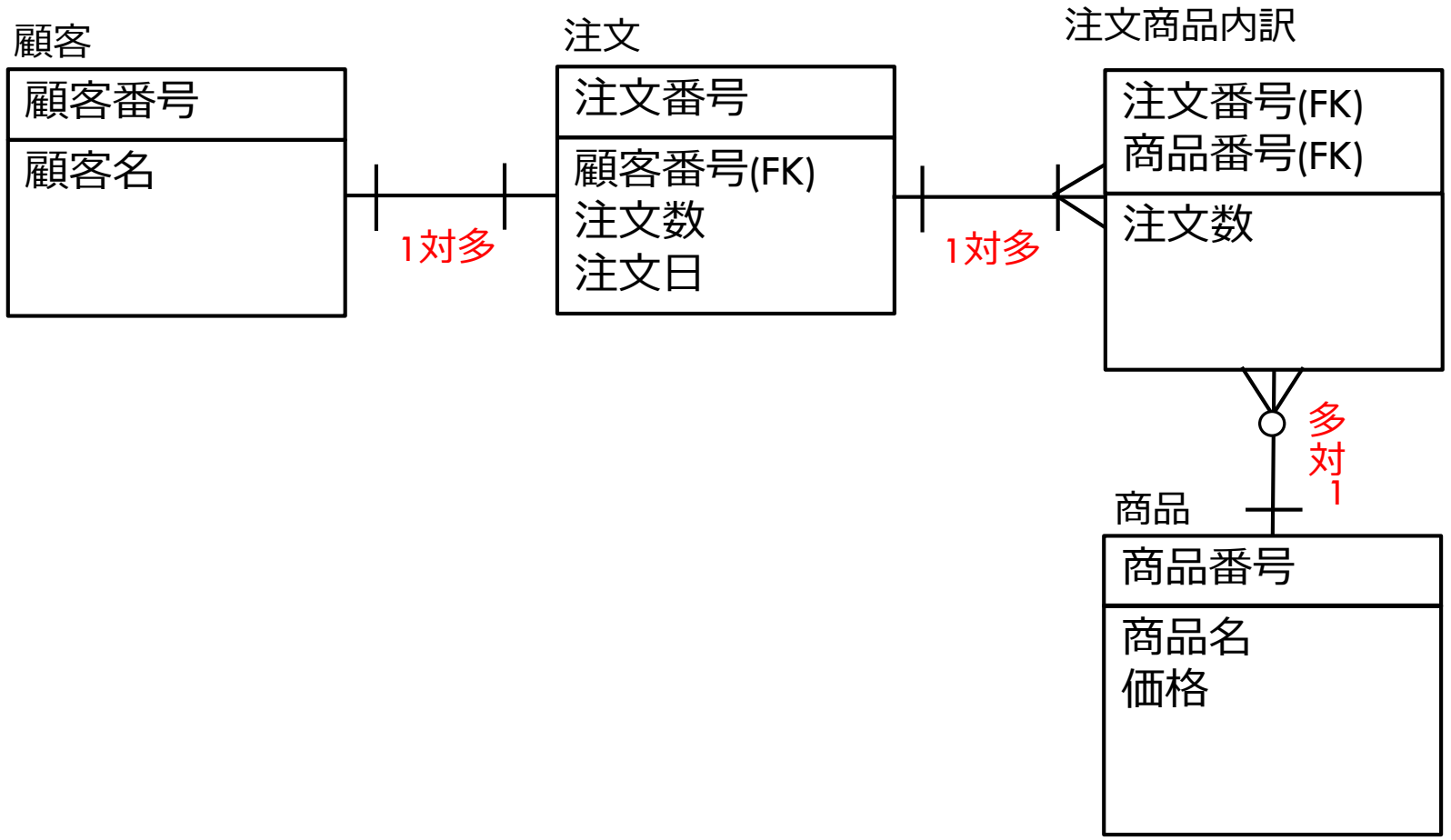
注文

注文番号	顧客番号	注文日
1	1	2020-04-06
2	2	2020-04-06
3	3	2020-04-06
4	1	2020-04-07
5	2	2020-04-08

注文商品内訳

注文番号	商品番号	注文数
1	1	5
1	2	10
2	4	20
3	3	5
4	1	20
5	4	10
5	1	20

答え


顧客
顧客番号
顧客名
1対多
注文
注文番号
顧客番号(FK)
注文数
注文日
1対多
注文商品内訳
注文番号(FK)
商品番号(FK)
注文数
多
対
1
商品
商品番号
商品名
価格

答え

顧客

顧客番号	顧客名
1	未来のかたち本校
2	未来のかたち2校
3	株式会社かたち

注文

注文番号	顧客番号	注文日
1	1	2020-04-06
2	2	2020-04-06
3	3	2020-04-06
4	1	2020-04-07
5	2	2020-04-08

答え

注文

注文番号	顧客番号	注文日
1	1	2020-04-06
2	2	2020-04-06
3	3	2020-04-06
4	1	2020-04-07
5	2	2020-04-08

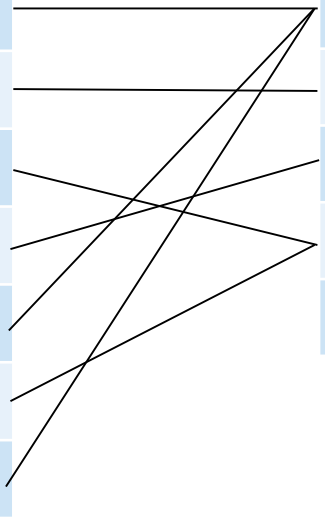
注文商品内訳

注文番号	商品番号	注文数
1	1	5
1	2	10
2	4	20
3	3	5
4	1	20
5	4	10
5	1	20

答え

注文商品内訳

注文番号	商品番号	注文数
1	1	5
1	2	10
2	4	20
3	3	5
4	1	20
5	4	10
5	1	20



商品

商品番号	商品名	値段
1	からあげ弁当	500円
2	ハンバーグ弁当	650円
3	のり弁当	400円
4	ミックスフライ弁当	750円
5	とんかつ弁当	800円