

STEP4 講義資料
POSアプリの仕様

Ver1.0

Step3の振り返りと宿題への導入

Step3の振り返りと宿題への導入 1/2

Step3ではNextjs,Flask,DB及びクラウドの学習を実施した。その後、顧客価値を考え、ビジネス検討を実施し、自らアプリケーションを構築した。

Step3全体

(3-2)

- ・ビジネスアイデアの検討
- ・Nextjs、Flask、DB構築
- ・クラウドへのデプロイ
- ・上記要素の連携

(3-1)

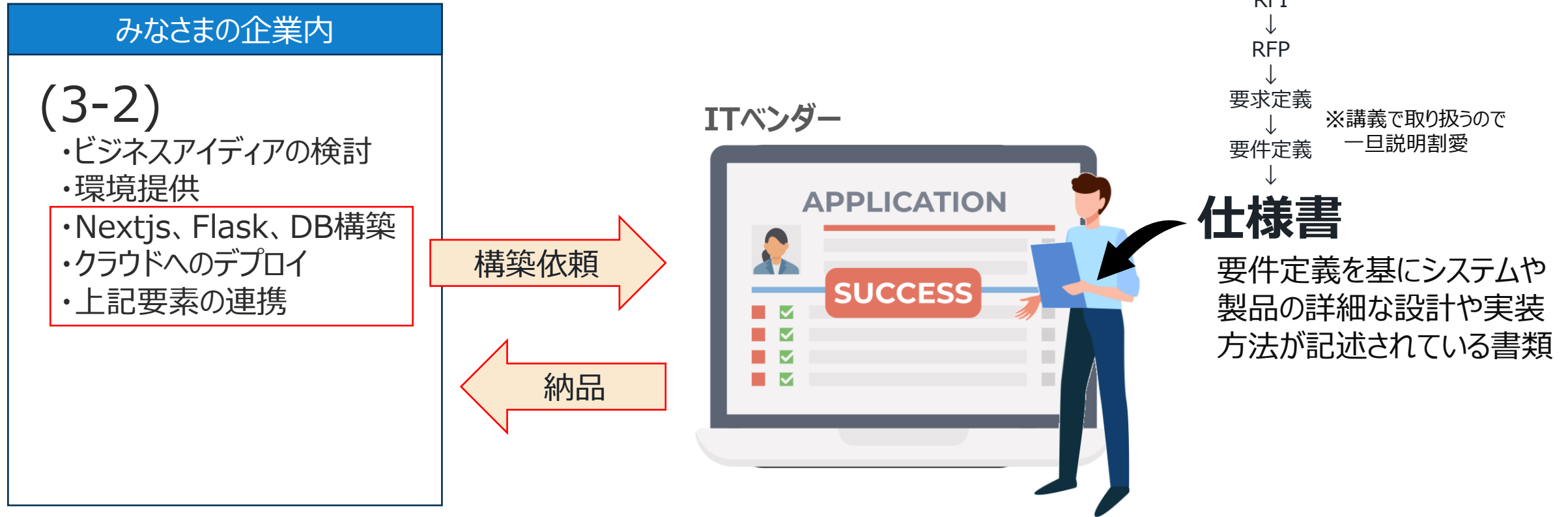
- ・Nextjs学習/構築
- ・Flask学習/構築
- ・SQLite学習/構築
- ・上記要素の連携

構築



Step3の振り返りと宿題への導入 2/2

みなさまの企業内では(内製開発もありますが)、外部ベンダーへ構築依頼することが多い。プロダクトの実装前に作成されるドキュメントが仕様書であり、プロダクトの品質を表している。その品質を向上するため、RFPを的確に記述する必要がある。



POS仕様書について

宿題：POSアプリの実装

POSアプリの仕様書から実装することで、テック面の復習やFastAPIについて学ぶだけでなく、外部ベンダーとのやり取りの解像度を高める。仕様書に即して、指定したアーキテクチャーで実装すること。

4902102111513

カメラで読込

商品情報

商品名：ペプシコーラ

価格：140円 ×

1

削除

数量の変更

購入リスト

ペプシコーラ 140円 ×1 140円 

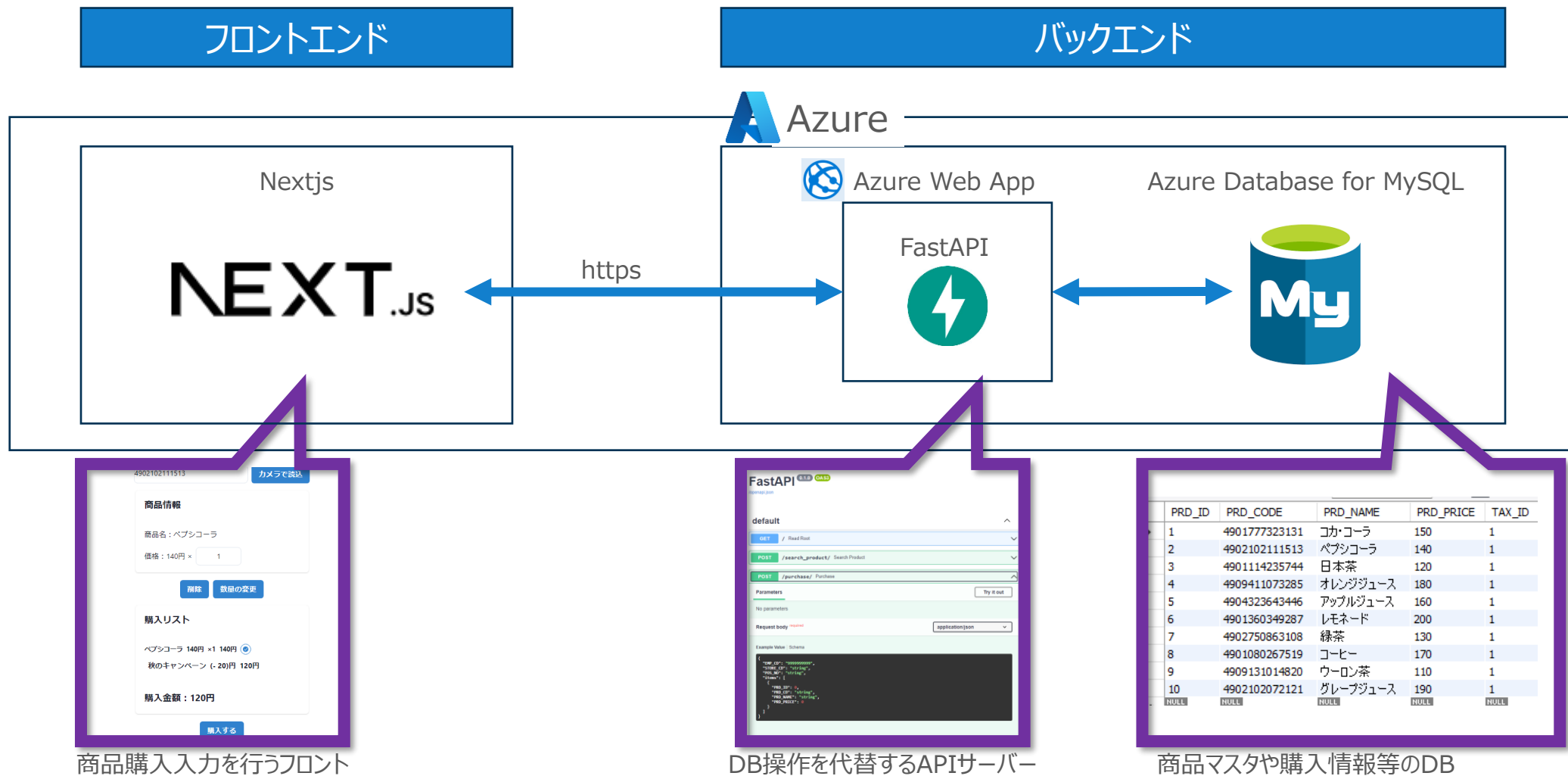
秋のキャンペーン (- 20)円 120円

購入金額：120円

購入する

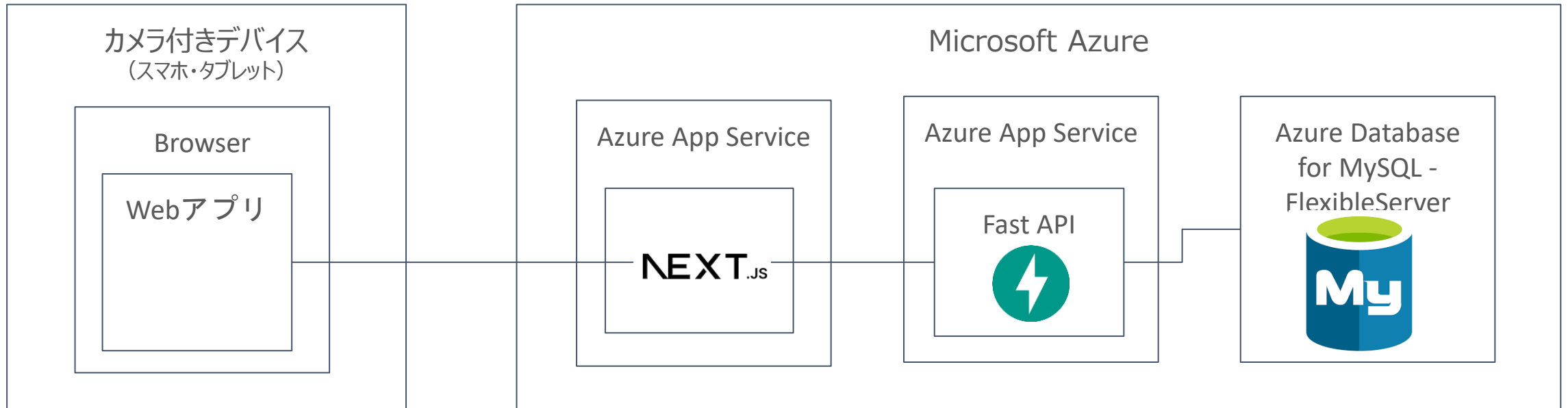
アーキテクチャ

Nextjs（フロントエンド）、MySQLとFastAPI（バックエンド）により構成されるマイクロサービスアーキテクチャでPOSアプリを再現している



基本構成（Lv1-3共通）

構成は同じだが、Lvに応じて機能面が異なる。



Lvについて

個人で作るのでLv2がベース（時間の都合に合わせて一部機能をLv1に寄せるのもOK）。

Lv1は最低限の実装。

Lv3はほぼ実用レベルのチャレンジ（覚悟して挑んでください♪）。

画面イメージ (Lv1)

画面構成と挙動は下記のとおりとする。

- ②コード入力エリア
- ①読み込みボタン
- ③名称表示エリア
- ④単価表示エリア
- ⑤購入リストへ追加ボタン
- ⑥購入品目リスト
(名称/数量/単価/単品合計)
- ⑦購入ボタン

12345678901

商品コード 読み込み

おいお茶

150円

追加

購入リスト

ソフラン x1 300円 300円
ソフラン x1 300円 300円
福島産ほうれん草 x1 188円 188円
タイガー歯ブラシ青 x1 200円 200円
四ツ谷サイダー x1 160円 160円
四ツ谷サイダー x1 160円 160円

購入

“Web画面POSアプリ”-Front概要

1. ②コード入力エリアへ商品コードをタイプし、①読み込みボタンを押下。コードを「商品マスタ」へ問合せ
2. 該当商品の名称、コード、単価を取得し画面表示
3. 該当商品がマスタにない場合は、名称表示エリアに「商品がマスタ未登録です」と表示。
4. 「追加」ボタンを押下すると、購入リストへ追加され、②③④は空欄になる。
5. 商品登録を繰り返しリストに商品を追加する作業を繰り返す。全て登録しおわたら「購入」を押下する購買結果はDBへ保存すること。
6. 購入後、ポップアップで合計金額（税込）を表示する。OKを押すとポップアップを閉じ②-④、⑥の内容をクリアする。

APIファンクション（Lv1）

APIの機能は下記の通り。

- ・機能名：商品マスタ検索
 - パラメータ：コード（商品コードのこと）
 - リターン：商品情報（商品一意キー/商品コード商品名称/商品単価）
 - 1-1 パラメータのコードに一致する商品コードの商品を1件返す。
 - 1-e1 対象が見つからなかった場合はNULL情報を返す。

- ・機能名：購入
 - パラメータ：レジ担当者コード、店舗コード、POS機ID
（商品リスト分）商品一意キー、商品コード、商品名称、商品単価
 - リターン：成否（True/False）、合計金額
 - 1-1 取引テーブルへ登録する。
取引一意キー（採番インクリメンタル 1～）、取引日時（現在日時システム日付）、
レジ担当者コード（パラメータより、空白の場合は'9999999999'）、店舗コード（'30'固定）
POS機ID（'90'固定、モバイルレジという意味）、合計金額（0）
登録後'取引一意キー'を取得し返す
 - 1-2 取引明細へ登録する。
取引一意キー（1-1の登録後の値）、取引明細一意キー（採番インクリメンタル 1～/取引ごと）、
商品一意キー（パラメータ）、商品コード（パラメータ）、商品名称（パラメータ）、
商品単価（パラメータ）
 - 1-3 合計を計算する
（繰り返し処理）商品単価を変数"V_合計金額"へ足し込んで合計金額作成。
 - 1-4 取引テーブルを更新する。
更新条件：取引一意キー（1-1の登録後の値）、更新値：合計金額（V_合計金額）
 - 1-5 合計金額をフロントへ返す。

DBイメージ (Lv1)

DBの仕様は下記の通り

商品一意キー
商品コード
商品名称
商品単価

商品マスタ

PRD_ID (PK) integer
CODE char (13)
NAME varchar (50)
PRICE integer

※CODEはCheck条件で
Uniqueになるよう制御

取引

TRD_ID (PK) integer
DATETIME timestamp
EMP_CD char(10)
STORE_CD char(5)
POS_NO char(3)
TOTAL_AMT integer

取引一意キー
取引日時
レジ担当者コード
店舗コード
POS機ID
合計金額

取引明細

TRD_ID (PK) integer
DTL_ID (PK) integer
PRD_ID integer
PRD_CODE char (13)
PRD_NAME varchar (50)
PRD_PRICE integer

取引一意キー
取引明細一意キー
商品一意キー
商品コード
商品名称
商品単価

1

商品一意キー

1

取引一意キー

1-

*

0-*

画面イメージ (Lv2)

画面構成と挙動は下記のとおりとする。

①バーコードスキャンボタン

②コード表示エリア

③名称表示エリア

④単価表示エリア

⑤購入リストへ追加ボタン

⑥購入品目リスト
(名称/数量/単価/単品合計)

⑦購入ボタン

The screenshot shows a mobile POS application interface. It features a vertical stack of elements: a blue 'スキャン (カメラ)' button at the top, followed by a code display area showing '12345678901', a name display area showing 'おーいお茶', and a price display area showing '150円'. Below these is a blue '追加' button. Underneath the '追加' button is a section titled '購入リスト' containing a list of items with their names, quantities, and prices. At the bottom is a blue '購入' button. Numbered callouts (①-⑦) point to these specific elements from the left side of the image.

名称	数量	単価	単品合計
ソフト	x1	300円	300円
ソフト	x1	300円	300円
福島産ほうれん草	x1	188円	188円
タイガー歯ブラシ青	x1	200円	200円
四ツ谷サイダー	x1	160円	160円
四ツ谷サイダー	x1	160円	160円

“モバイルPOSアプリ”

Frontアプリ概要

1. 「スキャン (カメラ)」をクリックするとカメラアプリが起動 (もしくはインフレームで)。
2. 利用者がバーコードを撮影
3. バーコードを読み取り→コード化
4. コードを「商品マスタ」へ問合せ
5. 該当商品の名称、コード、単価を取得し画面表示
6. 該当商品がマスタにない場合は、名称表示エリアに「商品がマスタ未登録です」と表示。
7. 「追加」ボタンを押下すると、購入リストへ追加され、②③④は空欄になる。
8. 商品登録を繰り返しリストに商品を追加する作業を繰り返す。
9. 全て登録しおわたら「購入」を押下する
10. 購買結果はDBへ保存
11. 購入後、ポップアップで合計金額 (税込) と税抜を表示する。
12. OKを押すとポップアップを閉じ②-④、⑥の内容をクリアする。

APIファンクション（Lv 2）

APIの機能は下記の通り。

- ・機能名：商品マスタ検索

- パラメータ：コード（商品コードのこと）

- リターン：商品情報（商品一意キー/商品コード商品名称/商品単価）

- 1-1 パラメータのコードに一致する商品コードの商品を1件返す。

- 1-e1 対象が見つからなかった場合はNULL情報を返す。

- ・機能名：購入

- パラメータ：レジ担当者コード、店舗コード、POS機ID

- （商品リスト分）商品一意キー、商品コード、商品名称、商品単価

- リターン：成否（True/False）、合計金額、合計金額（税抜）

- 1-1 取引テーブルへ登録する。

- 取引一意キー（採番インクリメンタル 1～）、取引日時（現在日時システム日付）、

- レジ担当者コード（パラメータより、空白の場合は'9999999999'）、店舗コード（'30'固定）

- POS機ID（'90'固定、モバイルレジという意味）、合計金額（0）、合計金額（税抜）（0）

- 登録後'取引一意キー'を取得し返す

- 1-2 取引明細へ登録する。

- 取引一意キー（1-1の登録後の値）、取引明細一意キー（採番インクリメンタル 1～/取引ごと）、

- 商品一意キー（パラメータ）、商品コード（パラメータ）、商品名称（パラメータ）、

- 商品単価（パラメータ）、消費税区分（'10'固定）

- 1-3 合計や税金額を計算する

- 税マスタより消費税区分をパラメータに税率を取得（今回は10%のみ、マスタ全般を本来は読み込む、8%も読んでおく）

- （繰り返し処理）商品リストの商品単価を変数"V_合計金額（税抜）"へ足し込んで合計金額（税抜）を作成。

- 併せて、商品単価×税率の結果を変数"V_合計金額"へ足し込んで合計金額作成。

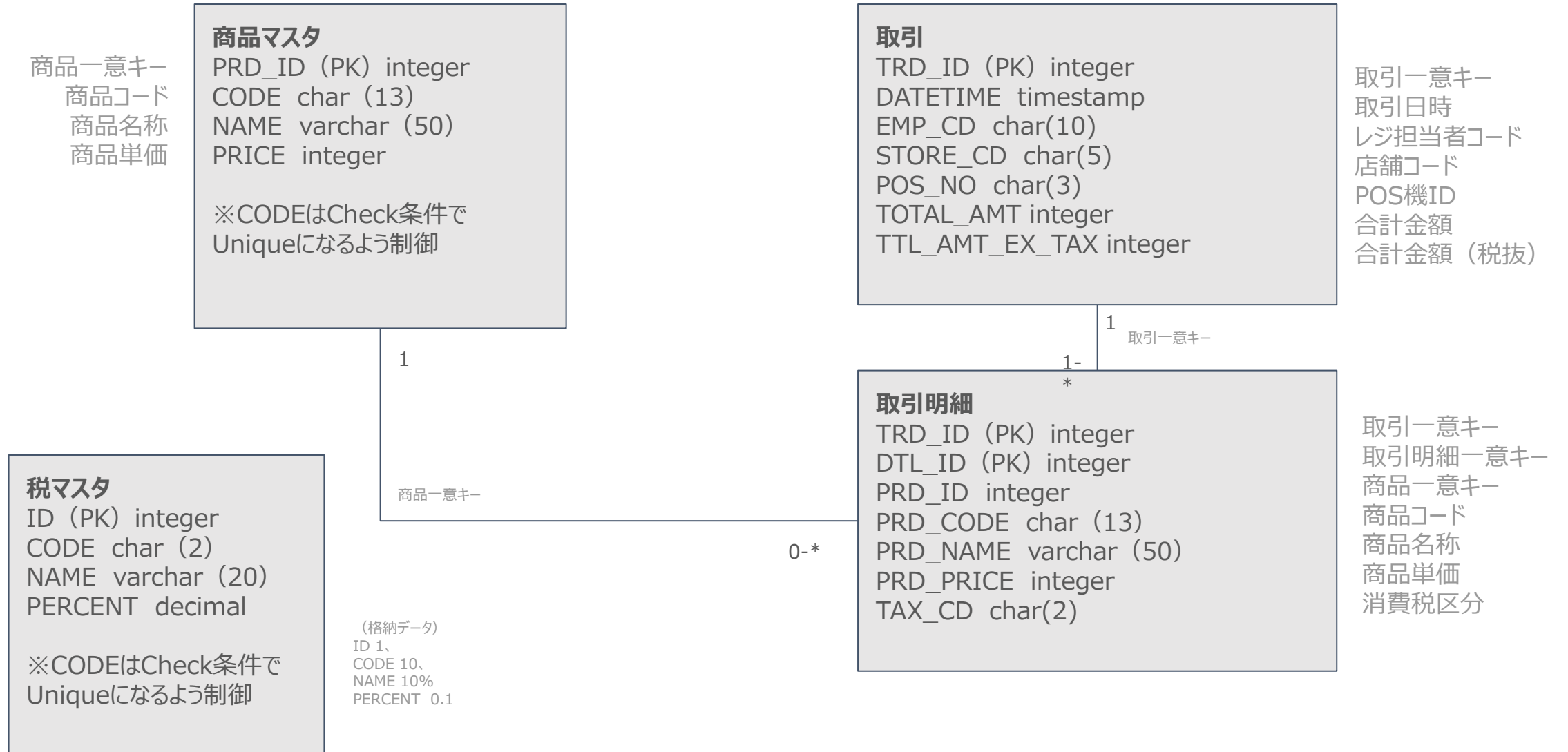
- 1-4 取引テーブルを更新する。

- 更新条件：取引一意キー（1-1の登録後の値）、更新値：合計金額（V_合計金額）、合計金額（税抜）（V_合計金額（税抜））

- 1-5 合計金額、合計金額（税抜）をフロントへ返す。

DBイメージ (Lv2)

DBの仕様は下記の通り



画面イメージ (Lv3)

画面構成と挙動は下記のとおりとする。

- ①バーコードスキャンボタン
- ②コード表示/入力エリア
- ③名称表示エリア
- ④単価表示エリア
- ⑧数量表示エリア/入力エリア
- ⑨購入リストから削除ボタン
- ⑨数量変更ボタン (更新ボタン)
- ⑥購入品目リスト
(名称/数量/単価/単品合計/販促企画)
選択中の商品は太字で表示
- ⑪編集開始ボタン
- ⑩合計表示エリア
- ⑦購入ボタン

スキャン (カメラ)

12345678901

四ツ谷サイダー

160円3個

リスト削除数量変更

購入リスト

ソフラン x2 300円 600円

福島産ほうれん草 x1 188円 188円

タイガー歯ブラシ青 x1 200円 200円

特売20円引き (-20円) 180円 180円

四ツ谷サイダー x3 160円 480円

↑↑↑↑

合計1,614 (税抜1,468) 円

購入

“FullモバイルPOSアプリ”Frontアプリ概要

■スキャン (カメラ)

スマホのカメラを利用したアプリが起動し、特定の枠内にバーコードが入ると認識撮影読み込み、リストに1件追加されたメッセージが浮き上がる。連続スキャンが可能で、メッセージ表示後同一商品であれば数量のインクリメント、別商品なら購入リストに商品が追加になる。

■購入リストの「↑」ボタンorリンク

リストから「↑」に該当する行の商品を選択をすると②③④⑧に値が表示され選択状態になる。

■リスト削除

購入リストに入れた商品 (選択状態の商品) を削除

■数量変更

購入リストに入れた商品 (選択状態の商品) の数量を変更
ただし「0個にはできない」「最大99個まで」

■購入ボタン

購入リストに入っている商品を購入する。ポップアップで合計金額 (税込) と税抜を表示する。OKを押すとポップアップを閉じ②-④、⑥、⑧、⑩の内容をクリアする。

□コード表示/入力エリア

スキャン後のコードを表示。キーボード入力も可能、確定で商品マスタを検索し、該当商品がある場合は③④を表示数量は1個とし購入リストへ追行する。購入リストにある商品の場合入力済の数量+1で表示する。確定する場合は「数量変更」を押下。

□合計表示エリア

購入リストの内容が更新されるタイミングで金額が更新される。

APIファンクション（Lv3）

APIの機能は下記の通り。

- ・機能名：商品マスタ検索

- パラメータ：コード（商品コードのこと）

- リターン：商品情報（商品一意キー/商品コード商品名称/商品単価）

- 1-1 パラメータのコードに一致する商品コードの商品を1件返す。

- 1-e1 対象が見つからなかった場合はNULL情報を返す。

- ・機能名：購入

- パラメータ：レジ担当者コード、店舗コード、POS機ID

- （商品リスト分）商品一意キー、商品コード、商品名称、商品単価

- リターン：成否（True/False）、合計金額、合計金額（税抜）

- 1-1 取引テーブルへ登録する。

- 取引一意キー（採番インクリメンタル 1～）、取引日時（現在日時システム日付）、

- レジ担当者コード（パラメータより、空白の場合は'9999999999'）、店舗コード（'30'固定）

- POS機ID（'90'固定、モバイルレジという意味）、合計金額（0）、合計金額（税抜）（0）

- 登録後'取引一意キー'を取得し返す

- 1-2 取引明細へ登録する。

- 取引一意キー（1-1の登録後の値）、取引明細一意キー（採番インクリメンタル 1～/取引ごと）、

- 商品一意キー（パラメータ）、商品コード（パラメータ）、商品名称（パラメータ）、

- 商品単価（パラメータ）、消費税区分（'10'固定）

- 1-3 合計や税金額を計算する

- 税マスタより消費税区分をパラメータに税率を取得（今回は10%のみ、マスタ全般を本来は読み込む、8%も読んでおく）

- （繰り返し処理）商品リストの商品単価を変数"V_合計金額（税抜）"へ足し込んで合計金額（税抜）を作成。

- 併せて、商品単価×税率の結果を変数"V_合計金額"へ足し込んで合計金額作成。

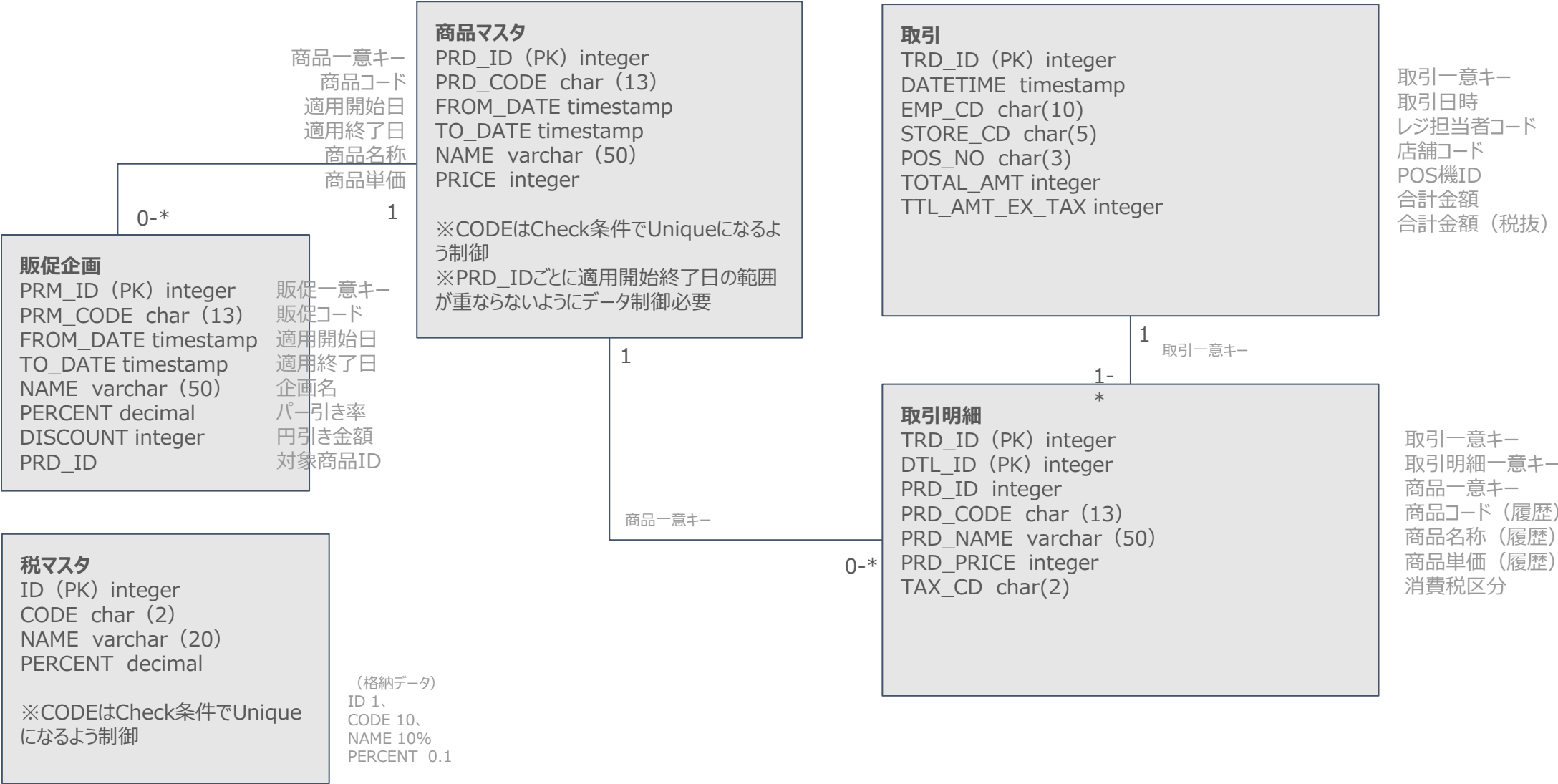
- 1-4 取引テーブルを更新する。

- 更新条件：取引一意キー（1-1の登録後の値）、更新値：合計金額（V_合計金額）、合計金額（税抜）（V_合計金額（税抜））

- 1-5 合計金額、合計金額（税抜）をフロントへ返す。

DBイメージ (Lv3)

DBの仕様は下記の通り



参考：POS構築の企業側のシナリオ

下記がPOSの構築を依頼してきた企業内部のシナリオである。仕様書の読み解きの際に、一助とすること。



ポップアップ専用Webサイト（簡易POSレジ機能含む）をリリースすることになった

背景

- ポップアップストアやテナントを作りあらたな顧客体験を新規事業部で提供する事となった。
- UI・UXについての詳細はこれから詰めていくことになるが、このポップアップストアですでに決まっていることがいくつかある。

決まっている事

- モバイルでQRコードを読み込むとテクワンのポップアップ専用Webサイトへ遷移する動線
- 固定端末も設置し、常にポップアップ専用Webサイトを表示し、実物の商品を目で確認しつつWeb上で商品のより詳細な情報を確認することができる。
- 商品情報は常に更新される（原料などの情報は頻繁に変わらないが、販売価格、キャンペーン、販促企画、リアルタイム口コミ、SNSアナリシスなど）ので、商品マスタは既存システムの基幹系の商品マスタを連携して利用する。（最低1時間に1回同期すること）
- ポップアップストアは数店舗展開予定。各店舗で独自商品や地産の商品など取り扱うことを容認している。
- システム開発の内製化を進めることが決まっており社内での方針により自社での内製開発をゼロにはできない。一方でERPやパッケージ製品も導入しているため、SIerやパートナー企業への発注もゼロにすることができないことは容認されている。
- セキュリティ事故や情報漏洩は顧客からの信頼を著しく低下させるため細心の注意を払う事が全社に徹底して言い渡されている。またIT部のガイドに準ずる必要があることが通達されている。

IT部と会話

- 先輩から教わったアプリ開発方法とは違う手法がいくつも存在しており過去実績が多数存在する模様。
- 新事業の可能性は感じている様だが、既存事業のITに対しての影響のことを懸念しているようだ。
- 安全性、可用性、リソース、連携方式、サービスレベルなど心配事は多く出てくる。
- ビジネスとテックでの意思疎通が必須なのは間違いない。しかし、同期の齋藤でも相互理解度がこのレベルなのでプロジェクトが進まない気がする。丸投げしたら進まないのは目に見えているようなものだ。最低限は双方歩み寄りねばならない。
- 商品の仕様や共通事項などはビジネスサイドよりもIT部の方が詳しい。仕様化するのに必要だったらしい。
- 要するに慎重にやるべきだということのようだ。IT部の責任であるセキュリティとガバナンスがポップアップ専用Webサイトでも担保されることが理解できないと協力できない模様。（協力したくないわけではない。）

POSアプリの実装に当たって

POSアプリの作成に当たって、以下の通りお願いします。

宿題のルール

○コーディング、デプロイについて

- 基本的に個人での実装を行ってください。
 - ー詰まってしまった、中身がよくわからないと言うときに受講生同士&TA&講師へのヘルプはもちろんOKです。

○デプロイ環境について

- Azureの環境を提供します。こちらへ記載しますのでご確認をお願いいたします。 [リンク](#)
 - ーApp Service、MySQLは運営から提供いたします(Step3と同様)。そちらを使用ください。
 - ーDBはクラスごとに割り振ります。TAさんが管理者です。DBは共有、テーブルは個別で作ってください。商品マスターデータは作成ください。

○納品について

- 納品者(つまり発注側)はTAさんです。構築が完了したら、TAさんにスレッドで報告してください。
 - ーTAさんが動作の検収をいたします。
- 2/13に全員提出済みです
- 次週、次々週のHRで途中経過を報告しましょう。
- 最終発表に直接絡むものではないですが、提出は必須です！

DB操作に使えるツール：A5M2

ER図の作図、DDL（データベース定義言語）の自動作成、DBのグラフィカルな操作、SQL問い合わせ等のDBにまつわる一通りのことができてしまうフリーのツール

[to english page](#)

A5:SQL Mk-2

SQL Development Suite

Copyright© 1997 松原正和

高機能かつ軽量の汎用SQLクライアント / ER図作成ソフト / フリーソフト
 (読み：エーファイブ・エスクーエル・マークツー)
 for Oracle Database, Microsoft SQL Server, IBM DB2, PostgreSQL, MySQL etc...


[オンラインヘルプ](#)


[ベータ版公開！](#)


[掲示板](#)


[BLOG](#)


[寄付の募集](#)

 **Download !!**
 A5:SQL Mk-2
 from Vector

Vectorからダウンロード
Version 2.18.2 (x64, x86)

 **Download !!**
 A5:SQL Mk-2
 from Microsoft Store

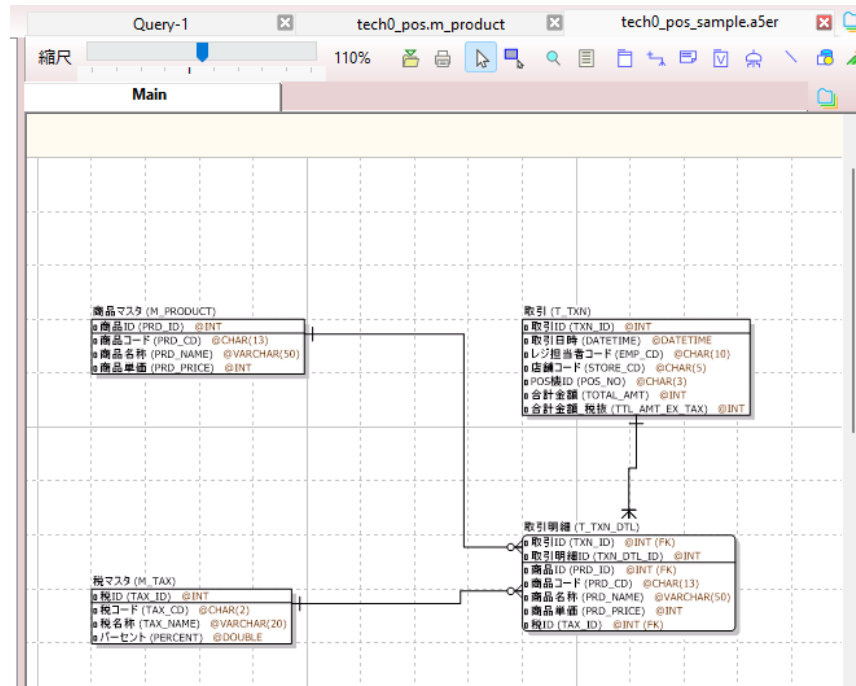
Microsoft Storeからダウンロード
Windows 10, 11
Version 2.18.3 (x64)

どちらからインストールすればよいかは[こちら](#)を参考にしてください。

<https://a5m2.mmatsubara.com/>

DB操作に使えるツール：A5M2

お絵描き感覚でER図を作成すると、相当するテーブルを作るためのSQLクエリが自動作成される。またテーブル定義書なども自動出力可能



作図ツールで
ER図を作成

ワンクリック

ワンクリック

```

1 Project Name : tech0-pos-sample
2 Date/Time   : 2023/08/20 23:02:12
3 Author      : wtdsl
4 DBMS Type   : MySQL
5 Application  : A5M2-M-2
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

```

相当するクエリ
を自動作成

A	B	C	D	E	F	G
1	エンティティ情報					
2	システム名	Tech0-POS-sample	作成者	saib3		
3	サブシステム名		作成日	2023/08/20		
4	スキーマ名		更新日			
5	論理エンティティ名	商品マスタ	タグ			
6	物理エンティティ名	M_PRODUCT				
7	備考					
8						
9						
10						
11						
12	カラム情報					
13	No.	論理名	データ型	Not Null	デフォルト	備考
14	1	商品ID	INT	Yes (PK)		
15	2	商品コード	CHAR(13)	Yes (FK)		
16	3	商品名称	SVARCHAR(50)	Yes		
17	4	商品単価	INT	Yes		
18						
19	インデックス情報					
20	No.	インデックス名	カラムリスト	ユニーク		備考
21						
22	リレーションシップ情報(FK)					
23	No.	制約名	カラムリスト	参照元エンティティ名	参照先カラムリスト	
24						
25	リレーションシップ情報(PK)					
26	No.	制約名	カラムリスト	参照元エンティティ名	参照先カラムリスト	
27	1		PRD_ID	T_TXN_DTL	PRD_ID	
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						

テーブル定義
書を自動作成

DB操作に使えるツール：A5M2

また特定のDBへ接続し、テーブルのレコードを覗く、SQLのクエリを書いて飛ばすことなどの操作も可能

The screenshot displays the A5SQL Mk-2 x64 Version 2.18.4 interface. The left sidebar shows a tree view of the database structure, including the 'tech0pos_sample' database and its tables: 'm_product' (商品マスタ), 'm_tax' (税マスタ), 't_txn' (取引), and 't_txn_dtl' (取引明細). The main window shows a query editor with the following SQL query:

```
SELECT * FROM tech0_pos.m_product WHERE PRD_PRICE>100
```

The query results are displayed in a table with the following columns: PRD_ID, PRD_CD, PRD_NAME, and PRD_PRICE. The results show 11 records of products with prices greater than 100.

PRD_ID	PRD_CD	PRD_NAME	PRD_PRICE
1	4514603263213	三ツ矢サイダー PET500ml	160
3	4514603217216	三ツ矢サイダー PET1.5L	370
4	4514603347814	三ツ矢サイダー PET430ml	134
5	4514603306514	三ツ矢サイダー PET300ml	130
6	4514603217513	三ツ矢サイダー 缶500ml	140
7	4514603217612	三ツ矢サイダー 缶250mlN*20本入り	111
8	4514603217612	三ツ矢サイダー 缶250mlN*30本入り	111
10	4514603217414	三ツ矢サイダー ボトル缶300ml	140
11	4514603431414	三ツ矢サイダーゼロストロングPET500ml	160

The status bar at the bottom indicates that the result set is not editable and that the content is reflected in the database immediately. The SQL editor shows the query: 'SELECT * FROM tech0_pos.m_product WHERE PRD_PRICE>100'.

スマホカメラでのバーコード読み取り機能を開発する場合の注意

スマホカメラにWebアプリからアクセスする場合、基本的にhttpsのセキュア通信上でしかアクセスできません
Nextjsでは、self-signed certificates機能を使うことでhttps通信をlocal上で使用することができる

self-signed certificatesを使う手順

1

package.json内"scripts" -> "dev"の"next dev"を"next dev --experimental-https"に修正する

2

※上記でnpm run devをするとエラーが出る場合は、一度.nextフォルダを削除

3

PCとスマホを同一Wifiにつなぎ、PCのローカルIPに接続
※<https://192.168.1.10:3000/pos2>

4

※iphoneはSafariではなくchromeであればいける

5

セキュリティアラート画面が出るので気にせずページへ接続する

6

カメラへのアクセスを許可するかとポップアップが出る→許可