



Power Appsで作る!

大学向け業務アプリ
自習書

著作権

このドキュメントに記載されている情報 (URL 等のインターネット Web サイトに関する情報を含む) は、将来予告なしに変更することがあります。別途記載されていない場合、このソフトウェアおよび関連するドキュメントで使用している会社、組織、製品、ドメイン名、電子メールアドレス、ロゴ、人物、場所、出来事などの名称は架空のもので、実在する名称とは一切関係ありません。お客様ご自身の責任において、適用されるすべての著作権関連法規に従ったご使用をお願いします。マイクロソフトは、このドキュメントに記載されている内容に関し、特許、特許申請、商標、著作権、またはその他の無体財産権を有する場合があります。別途マイクロソフトのライセンス契約上に明示の規定のない限り、このドキュメントはこれらの特許、商標、著作権、またはその他の無体財産権に関する権利をお客様に許諾するものではありません。

© 2021 Microsoft Corporation. All rights reserved.

Microsoft、Power Apps は、米国 Microsoft Corporation の米国およびその他の一定の国における/またはその関連会社の商標 です。その他のすべての商標は、それぞれの所有者に帰属します。

変更履歴

No.	日付	版	変更概要
1	2021年 06月 16日	1.0	初版作成

目次

目次

1. 本書の目的	5
2. 概要	6
2.1. アプリ作成の対象者	6
2.2. アプリ作成で使用するソフトウェア・サービス	6
2.3. アプリ作成で使用するサンプルデータ	6
2.4. Power Apps アプリ作成のシナリオ	7
2.5. Power Apps の画面構成	9
3. 演習 1 テーブルの作成	11
3.1. Microsoft Dataverse の概要	11
3.2. データ構造を設計する	11
3.3. テーブルを作成する	12
3.4. テーブルに列を作成するための基本操作	14
3.5. テーブルの列を作成する	23
4. 演習 2 ビューの作成	26
4.1. 作成するビューについて	26
4.2. ビューを編集するための基本操作	27
4.3. ビューを作成する	35
5. 演習 3 モデル駆動型アプリの作成	40
5.1. モデル駆動型アプリの新規作成	40
5.2. ビジネス プロセス フローの作成	54
5.3. ダッシュボードの作成	68
6. 演習 4 承認依頼の自動化	77
6.1. クラウド フローの新規作成	77
6.2. トリガーを設定する	79
6.3. 最初の条件を作成する	81
6.4. 申請しない場合の処理を作成する	85
6.5. 承認依頼処理を作成する	88
6.6. 承認結果の条件を作成する	91
6.7. 承認後の処理を作成する	92
6.8. フローをテストする	94
7. 演習 5 (オプション 1) キャンバス アプリの作成	95
7.1. 事前準備	95
7.2. キャンバス アプリの新規作成	96

7.3.	照会画面を作成する.....	102
7.4.	更新画面を作成する.....	118
8.	演習 6 (オプション 2) リマインダーフローの作成.....	128
8.1.	クラウド フローの新規作成.....	128
8.2.	スケジュールされたフローのテスト.....	133
9.	おわりに	137

1. 本書の目的

本書は、Microsoft Power Apps のシステム管理者、または、アプリの作成者が Power Apps アプリの基本的な構築を一から学習できるように簡易に記載されています。そのため、幅広く機能を網羅的に紹介したり、例外的な処理を記述するのではなく、Microsoft Power Apps の基本的なカスタマイズを中心にご理解いただいたり、メリットをご体験いただくことを目的としています。

- ※ 操作画面は本執筆時点のものであり、変更される可能性があります。考え方をご理解いただければ、画面変更があったとしても対応可能なスキルが身につくよう考慮されています。
- ※ 本コンテンツ内で紹介するアプリケーションの作成手順は大阪電気通信大学様の事例を参考とさせて頂いておりますが、全く同じ内容ではございません。予めご了承ください。

2. 概要

本書では、主に Microsoft Edge を利用します。

Power Apps がサポートされているブラウザ及びオペレーティングシステムは以下です。

ブラウザ	サポートされているバージョン
Microsoft Edge	最新の 3 つのメジャーリリース
Google Chrome	最新の 3 つのメジャーリリース
Mozilla Firefox (最新バージョン)	最新の 3 つのメジャーリリース
Apple Safari	13 以降

オペレーティングシステム	サポートされているバージョン
Windows	Windows 8.1 またはそれ以降
Mac OS	10.13 またはそれ以降
iOS	iOS 13 移行
Android	10 またはそれ以降

2.1. アプリ作成の対象者

Power Apps システム管理者

Microsoft Dataverse システム管理者

Power Apps 構築担当者

2.2. アプリ作成で使用するソフトウェア・サービス

Microsoft Dataverse

アプリ デザイナー

Microsoft Power Automate

Power Apps Studio

Microsoft Edge

Power Apps per app plan ライセンスもしくは Power Apps per user plan ライセンス

2.3. アプリ作成で使用するサンプルデータ

本書内で使用する以下のサンプルデータは、以下のリンクからダウンロード可能です。事前にダウンロードし、.zip ファイルを展開したうえでご活用ください。

https://download.microsoft.com/download/e/4/d/e4d065f4-6cfc-48ab-b7df-7b587b0a73c2/Sample_University_bizapp_withPowerApps.zip

学校基本情報サンプル.csv

受験実績情報サンプル.csv

2.4. Power Apps アプリ作成のシナリオ

ある大学では入学志願者数を増やすための活動の一つとして高校訪問を行っています。この活動を支援してより訪問活動の効果を得られるようにするシステムを開発することになりました。ソリューションの主な機能として以下を目標とします。

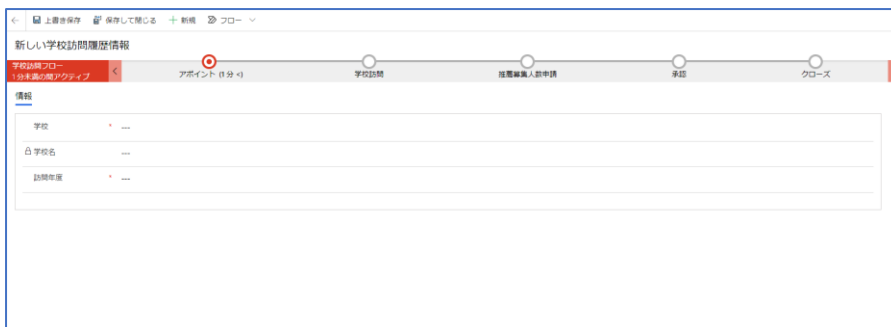
- 各高校の過去の受験人数などを多角的に分析できる
- 訪問先として抽出した高校への訪問アポイントを登録できる
- 訪問アポイントに対して訪問実績（ヒアリング内容など）を登録できる
- ヒアリング・調整の結果に応じて推薦枠の申請を行うかどうかを選択できる
- 調整した推薦枠を申請した場合、管理者に承認依頼のメールが自動送信される
- 管理者は承認依頼メールから直接承認または否認できる
- （オプション）推薦枠が承認された後、願書送付時期が近づくとリマインドメールが自動送信される

本書ではモデル駆動型アプリとして上記機能を作成します。以降の演習を終えると、次のような見目のアプリが完成します。

メイン画面（ダッシュボード：過去の入学者数推移を分析・訪問先を検討し、訪問実績を管理します）



訪問実績管理画面（学校訪問の業務フローに従って業務遂行を管理します）



また、オプションのキャンパス アプリでは以下のような画面を作成します。

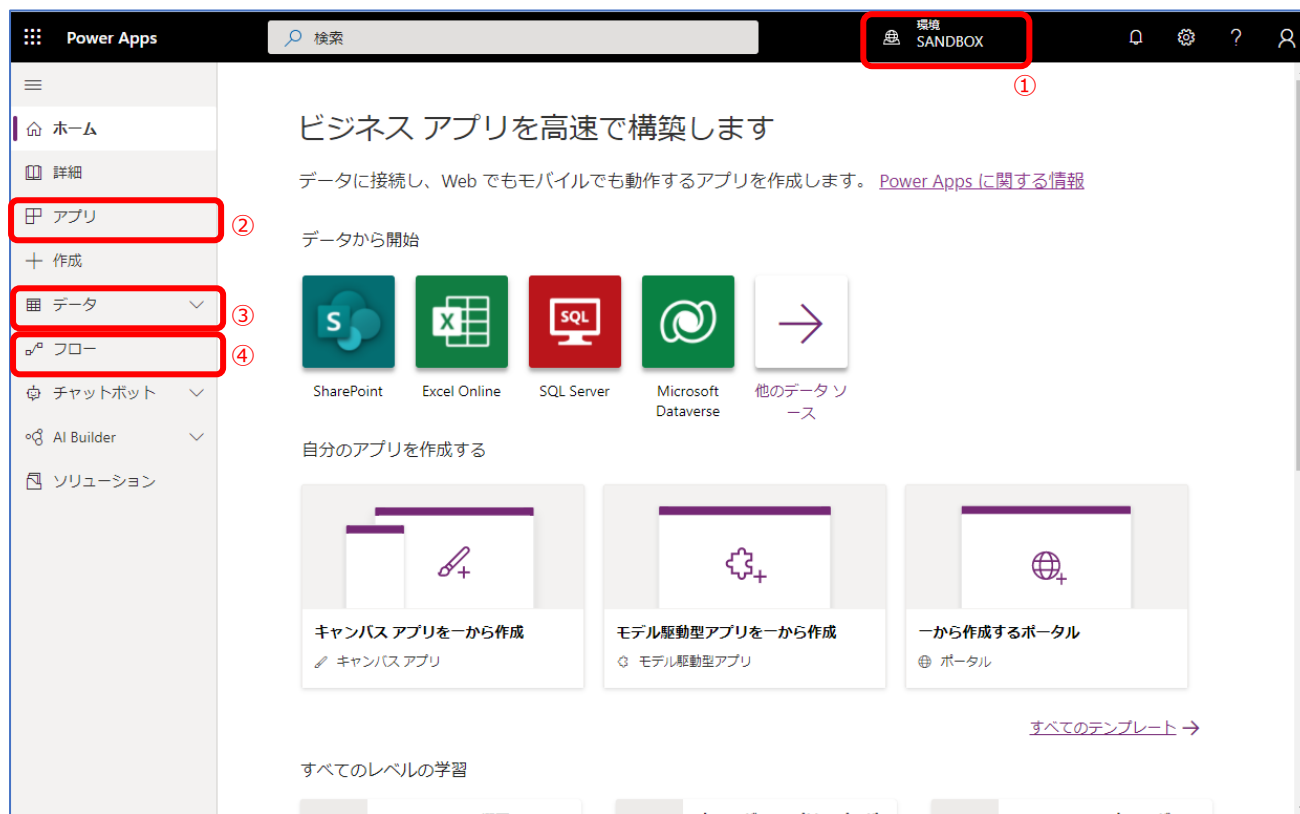
キャンパス アプリにより、タブレット端末などで訪問時にリアルタイムでヒアリング内容等を登録することができます。



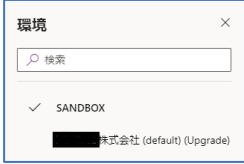

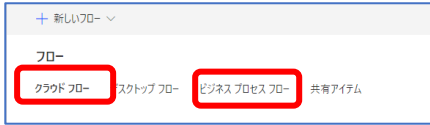
2.5. Power Apps の画面構成

作業に入る前に、Power Apps の画面構成を理解しておきましょう。

ブラウザで URL に <https://make.powerapps.com/> を入力すると以下のような画面が表示されます。



この演習で主に使用するのは以下の 4 アイテムです。

No.	名前	説明
①	環境	<p>現在の環境です。環境とは組織のビジネス データ、アプリ、チャットボット、およびフローを保存、管理、共有する場所です。さまざまな役割、セキュリティ要件や対象ユーザーを持つ、アプリを分離するコンテナとしても機能します。</p> <p>環境についての詳細は環境の概要で確認してください。環境の作成を参照して専用の新しい環境を作成することもできます。クリックすると画面右側に以下のようなパネルが表示されるので、必ず作業を始める前に作業を行うことが許可されている環境を選択してください。</p> <p>また、環境の作成が出来ない場合は、システム管理者にご確認ください。</p> 
②	アプリ	<p>アプリの新規作成、既存アプリの編集やアプリの実行をすることができます。</p> <p>作成したアプリはここに一覧として表示されます。</p>
③	データ	<p>Power Apps で扱うデータに関する操作メニューが含まれており、展開すると以下のサブメニューが表示されます。この演習で使用するのはテーブルのみです。</p> 
④	フロー	<p>フローの新規作成、既存フローの編集やその他のフロー管理をすることができます。</p> <p>作成したフローは、種類ごとにここに一覧として表示されます。</p> <p>フローの種類には以下がありますが、この演習で使用するのはクラウドフローとビジネスプロセスフローのみです。</p> 

3. 演習 1 テーブルの作成

3.1. Microsoft Dataverse の概要

Dataverse を使用するとビジネスアプリケーションが使用するデータを安全に保存し、管理することができます。

また Dataverse を使用するモデル駆動型アプリでは通常のデータベースのような複雑な手続きを必要とせず、いくつかの設定をするだけで参照したい情報にアクセスしたりデータを更新したりすることが可能です。

Dataverse についての詳細な情報は [Microsoft Dataverse とは](#)で確認することができます。

以下の手順で選択中の環境の Dataverse にアプリで使用するテーブルを作成します。

3.2. データ構造を設計する

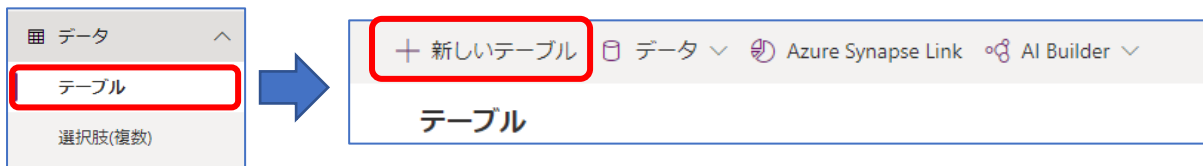
モデル駆動型アプリでは、プログラムをコーディングすることなく簡単な設定のみでアプリを開発することができますが、そのためには目的を明確に設計した Dataverse のテーブルが必要です。この演習では以下のテーブルを作成します。

No	名前	論理名	説明
1	学校基本情報	SchoolInfo	学校マスタです。 学校名、所在地、電話番号などの基本情報を管理します。 このテーブルには学校ごとに 1 件ずつのデータを登録します。
2	受験実績情報	ExaminationResults	受験者統計データです。 学校、年度ごとの志願者、受験者、合格者、入学者の数を管理します。このデータをもとに高校ごとの推移を分析して訪問先の学校を選定します。 このテーブルには学校ごと、年度ごとに 1 件ずつのデータを登録します。
3	学校訪問履歴情報	SchoolVisitedHistory	学校訪問の実績 (履歴) データです。 訪問の回ごとに訪問業務フローを管理し、また過去の訪問の内容がどのようなものであったかを参照することもできます。 学校訪問は 1 つの学校に対し年度内に何度も実施する可能性があるため、このテーブルには学校ごと、年度ごとに複数件のデータを登録することができます。 複数回の訪問には以下のようなケースが考えられます。 <ul style="list-style-type: none">・ 1 回目：ご挨拶のみ (推薦枠の申請をしない)・ 2 回目：ヒアリングに基づき推薦枠を申請 ⇒ 否認・ 3 回目：学校側と再調整した推薦枠を申請 ⇒ 承認

3.3. テーブルを作成する

テーブルを作成するには以下の手順を実施します。

- ① 左側のパネルから [テーブル] を選択し、続いて上部に表示されたメニューから [新しいテーブル] を選択します。



- ② 画面右側にテーブルの基本属性を入力するパネルが表示されますので、テーブル名を設定します。
今回は学校基本情報テーブルを作成します。

The '新しいテーブル' (New Table) form is shown. It has three input fields: '表示名*' (Display Name) with a red circle 1, '表示名の複数形*' (Display Name Plural) with a red circle 2, and '名前*' (Name) with a red circle 3. The '名前*' field contains the text 'cr4ca_'.

No.	項目	入力内容
1	表示名	学校基本情報
2	表示名の複数形	学校基本情報 (自動で設定されます)
3	名前	SchoolInfo

※注: [名前] の入力欄の前についている値はオブジェクトをユニークに管理するための接頭辞です。
接頭辞は環境によって異なった値となります。

- ③ 同じパネルの下部にプライマリ列の設定用項目が表示されています。プライマリ列は、レコードに関するメインのユーザー フレンドリなテキスト識別子 (ほとんどの場合は名前または数値) です。この列の値は、ユーザーがレコードの一覧から選択する必要がある場合に表示されます。学校基本情報ではこの列を学校コードとして設定します。

The 'プライマリ名の列' (Primary Name Column) form is shown. It has two input fields: '表示名*' (Display Name) with a red circle 4 and '名前*' (Name) with a red circle 5. The '名前*' field contains the text 'cr4ca_ Name'. There is also a checkbox with a red circle 6 labeled '添付ファイルを有効にする (メモとファイルを含む)' (Enable attachments (include memo and files)).

No.	項目	入力内容
4	表示名	学校コード
5	名前	SchoolCode
6	添付ファイルを有効にする	オフのままにする

- ④ パネル最下部の [作成] ボタンをクリックしてテーブルを作成します。

The bottom of the panel shows two buttons: '作成' (Create) and 'キャンセル' (Cancel). The '作成' button is highlighted with a red box.

⑤ テーブルが作成されました。

テーブル

テーブル ↑ ↓	名前 ↓	種類 ↓	カスタマイ... ↓	タグ ↓
学校基本情報	cr4ca_schoolinfo	カスタム	✓	カスタム

探している項目が見つからない場合は、上記のフィルターをリセットして検索範囲を広げてください。

注：標準テーブルが一覧に表示されている場合、自分が作成したテーブルを見つけにくいことがあります。このような場合は画面右上にあるフィルターで [カスタム] を選択すると見つけやすくなります。

新しいテーブル データ Azure Synapse Link AI Builder

≡ カスタム 🔍 検索

- ✓ ≡ 一覧
- ≡ グループ
- すべて
- マネージド
- ✓ カスタム
- 既定

上記 ①～⑥ と同様の手順を繰り返して次の 2 テーブルを作成します。

No.	テーブルの表示名	テーブルの名前	プライマリ列の表示名	プライマリ列の名前
1	受験実績情報	ExaminationResults	受験実績 ID	ID
2	学校訪問履歴情報	SchoolVisitedHistory	訪問 ID	VisitID

必要な 3 テーブルがすべて作成できました。

次の手順で各テーブルに列を作成します。

テーブル

テーブル ↑ ↓	名前 ↓	種類 ↓	カスタマイズ可能 ↓	タグ ↓
学校基本情報	cr4ca_schoolinfo	カスタム	✓	カスタム
学校訪問履歴情報	cr4ca_schoolvisithistory	カスタム	✓	カスタム
受験実績情報	cr4ca_examinationresults	カスタム	✓	カスタム

探している項目が見つからない場合は、上記のフィルターをリセットして検索範囲を広げてください。

3.4. テーブルに列を作成するための基本操作

列を追加する手順は、その列に格納する値のデータ型により若干の違いがあります。

ここでは前の手順で作成したテーブルごとに、手順の異なるデータ型の列をピックアップして紹介します。これらの手順を終了後、同様の手順で各テーブルに不足する列を追加していただきます。

A. 共通手順

ここでは列の定義を行う上での基本的な操作方法を説明します。以降の手順ではこれらの操作を省略しています。

- ① テーブルの一覧から、作業対象のテーブル名をクリックします。[学校基本情報] を選択してください。

テーブル ↑ ↓	名前 ↓	種類 ↓	カスタマイ... ↓	タグ ↓	
学校基本情報	...	cr4ca_schoolinfo	カスタム	✓	カスタム
学校訪問履歴情報	...	cr4ca_schoolvisithistory	カスタム	✓	カスタム
受験実績情報	...	cr4ca_examinationres...	カスタム	✓	カスタム

探している項目が見つからない場合は、上記のフィルターをリセットして検索範囲を広げてください。

- ② テーブルに定義されている列が一覧表示されます。

表示名 ↑ ↓	名前 ↓	データ型 ↓	種類 ↓	カスタ... ↓	必須 ↓	検索可能 ↓	
UTC 変換タイムゾーンコード	...	utcconversion...	整数	標準	任意		
インポートシーケンス番号	...	importsequen...	整数	標準	✓	任意	✓
ステータス	...	statuscode	選択肢	標準	✓	任意	✓
タイムゾーン規則のバージョン番号	...	timezonelev...	整数	標準	任意		
バージョン番号	...	versionnumber	大きい...	標準	任意		
レコード作成日	...	overriddencre...	日付のみ	標準	✓	任意	✓
学校基本情報	...	cr4ca_schoolin...	一意識...	標準	✓	必須	✓
作成者	...	createdby	参照	標準	✓	任意	✓
作成者(代理)	...	createdonbeh...	参照	標準	✓	任意	✓
作成日	...	createdon	日時	標準	✓	任意	✓
修正者	...	modifiedby	参照	標準	✓	任意	✓

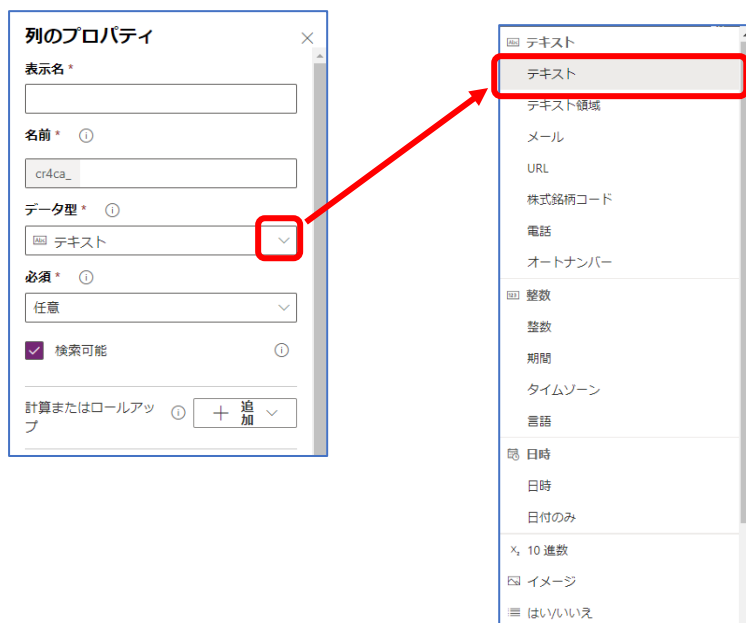
注：テーブルを作成すると自動的に追加される標準列が表示されています。

この状態では自分で作成した列が探しにくい場合があります。テーブルのフィルターと同様、画面右上方にあるフィルターで [カスタム] を選択すると探しやすくなります。

- ③ 画面上のメニューから列の追加を選択します。

+	列の追加	削除	テーブルの削除	データ ↓	Azure Synapse Link ↓	AI Builder ↓	設定	カスタム ↓	検索
---	------	----	---------	-------	----------------------	--------------	----	--------	----

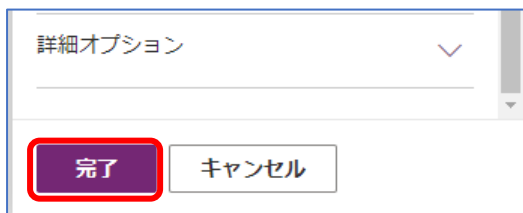
- ④ 画面右側にパネルが開きます。ここに列名やデータ型などの属性を入力します。
 [データ型] はプルダウンを展開することにより表示されるリストから選択することにより入力します。
 また、[必須] も [任意] / [推奨] / [必須] のいずれかの値をプルダウンから選択して入力します。



ここでは操作例として [学校名] 列を定義してみましょう。

No.	項目	入力内容
1	表示名	学校名
2	名前	SchoolName
3	データ型	テキスト
4	必須	必須

- ⑤ 必要な項目の入力が完了後は、パネル最下部の [完了] をクリックします。



- ⑥ 定義済みの列の属性を変更する場合は、列の一覧から [表示名] をクリックして列属性のパネルを開きます。

The image shows two panels from a software interface. The left panel, titled 'テーブル > 学校基本情報', displays a table with columns: '学校コード', 'プライマリ名の列', '...', 'cr4ca_school...', 'テキスト', 'カスタム', '必須', and '検索可能'. The '学校コード' column header is highlighted with a red box. The right panel, titled '学校コード', shows configuration options: '表示名*' (学校コード), '名前*' (cr4ca, SchoolCode), 'データ型*' (テキスト), '必須*' (必須), and a checked '検索可能' checkbox. There are also help icons (i) next to several fields.

注：[名前] などのように、変更ができない属性は上記のように入力できない状態が表示されます。

このような項目を修正する必要がある場合は、この列を一度削除してから再作成してください。

※ 既にアプリなどから参照している列は削除できません。

また、列属性の変更後も、列の追加時と同様、忘れずに [完了] をクリックしてください。

- ⑦ 列の定義が終わったら必ず画面右下の [テーブルの保存] をクリックしてください。

テーブルを保存せずに他のテーブルや別のメニューに移動すると設定内容が失われてしまいます。



B. 選択枝列

列のデータ型を [選択枝] とすると、その列は選択枝に含まれる値以外は受け付けられないため、条件分岐の判定に使用する列など、決まった選択枝から入力させたい場合に有効です。

ここでは学校基本情報テーブルの学校種別列を例として手順を説明します。

- ① 学校基本情報テーブルで新規列を追加し、以下のように属性を入力してください

学校種別

表示名 *
学校種別 ①

名前 * ①
cr4ca_ SchoolType ②

データ型 * ①
≡ 選択枝 ③

選択枝 *
▼

必須 * ①
任意 ▼

No.	項目	入力内容
1	表示名	学校種別
2	名前	SchoolType
3	データ型	選択枝

- ② 選択枝を展開し、[新しい選択枝] を選択します。選択枝の編集パネルが開いたら選択枝を登録します。

学校種別

表示名 *
学校種別

名前 *
cr4ca_ SchoolType

データ型 * ①
≡ 選択枝 ▼

選択枝 *
▼

十新しい選択

学校種別

表示名 *
学校種別

名前 *
cr4ca_ schooltype

詳細を表示

項目 (2)
公立
私立

新しい項目の追加

注：選択枝を追加するには [新しい項目の追加] をクリックし、追加された空の入力欄に入力します。

この例では [公立] と [私立] を選択枝として登録しました。

C. オートナンバー列

[オートナンバー] は新しいデータが追加されるたびにシステムが自動的にカウントアップする番号を持つデータ型で、操作者が意識することなく登録データにユニークな番号を振りたい場合などに有効です。

ここでは受験実績情報テーブルの受験実績 ID 列を例として手順を説明します。

- ① 受験実績情報テーブルで受験実績 ID 列を選択し、以下のように属性を入力してください

受験実績ID

表示名 *
受験実績ID ①

名前 * ①
cr4ca_ ID ②

データ型 * ①
オートナンバー ③

必須 * ①
必須 ④

検索可能 ①

オートナンバーの種類 * ①
文字列が先頭に付加される数 ⑤

プレフィックス
⑥

最小桁数 * ①
16 ⑦

シード値 * ①
1,000 ⑧

No.	項目	入力内容
1	表示名	受験実績 ID (入力済み)
2	名前	ID (入力済み)
3	データ型	オートナンバー
4	必須	必須
5	オートナンバーの種類	文字列が先頭に付加される数
6	プレフィックス	(空白のまま)
7	最小桁数	16
8	シード値	1,000

注 : [プレフィックス]、[最小桁数]、[シード値] は自由に設定しても以降で定義する処理に影響しません。

D. 参照列

テーブルの列としてマスターテーブルへの参照情報を定義することができます。参照列を通じてマスターテーブルが持っている列のデータを、あたかも自分の列であるように参照することが可能となります。

ここでは受験実績情報テーブルの学校列を例として手順を説明します。

- ① 受験実績情報テーブルで新規列を追加し、以下のように属性を入力してください。

列のプロパティ

表示名 *
学校 ①

名前 * ①
cr4ca_ School ②

データ型 * ①
参照 ③

関連テーブル *
学校基本情報 ④

必須 * ①
必須 ⑤

検索可能 ①

No.	項目	入力内容
1	表示名	学校
2	名前	School
3	データ型	参照
4	関連テーブル	学校基本情報
5	必須	必須

E. 計算列

自テーブルの項目や関連（参照）テーブルの項目を使って、計算結果や加工した内容を列データとして表示させることが可能です。例えば 2 つ以上の列の値の合計を表示したり、参照テーブルから名称を表示したりできます。ここでは受験実績情報テーブルの学校名列を例として手順を説明します。

- ① 受験実績情報テーブルで新規列を追加し、以下のように属性を入力後に [計算またはロールアップ] の [追加] をクリックし、表示されたリストから [計算] を選択してください。

列のプロパティ ×

表示名*
学校名 ①

名前* ①
cr4ca_ SchoolNameRef ②

データ型* ①
テキスト ③

必須* ①
任意

検索可能 ①

計算またはロールアップ ① **+ 追加** ↓

No.	項目	入力内容
1	表示名	学校名
2	名前	SchoolNameRef
3	データ型	テキスト

+ 計算

+ ロールアップ

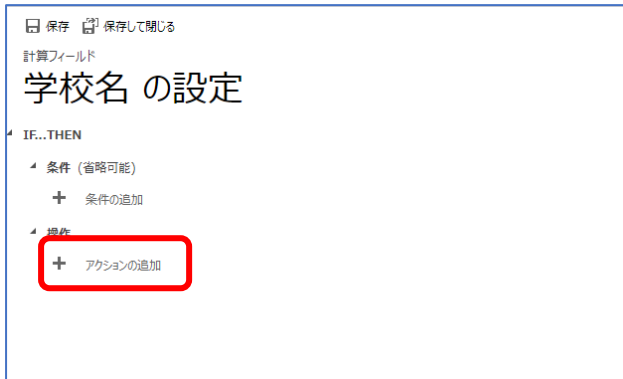
注：[計算またはロールアップ] の [追加] をクリックすると下記のメッセージが表示されますので、[保存] を選択してください。

保留中の変更

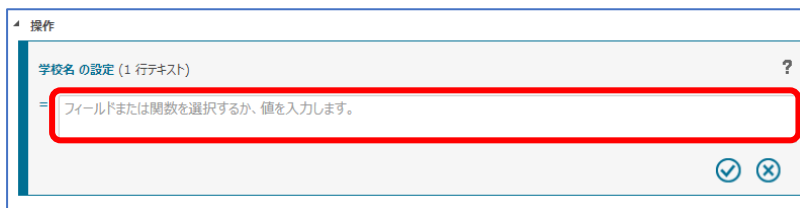
計算済み列のエディターを開くには、まず現在のテーブルを保存する必要があります。現在のテーブルを保存する準備ができている場合は、下の [保存] をクリックします。

保存 キャンセル

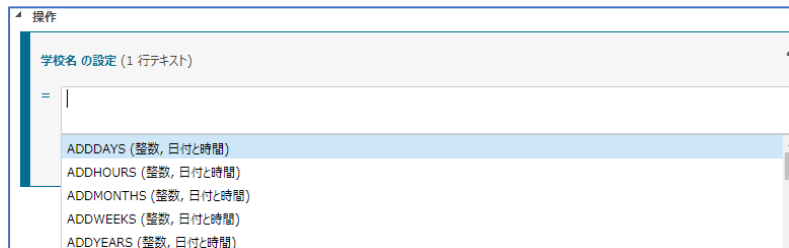
- ② 下記の新しいウィンドウが開きます。ウィンドウが開いたら [アクションの追加] をクリックしてください。



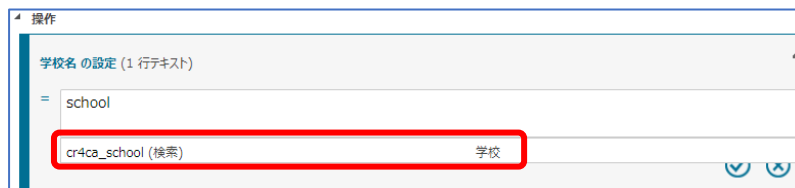
[アクションの追加] をクリックすると次のような入力フィールドが表示されます。



このフィールドをクリックして入力できる状態にすると、式の候補がリストで表示されます。

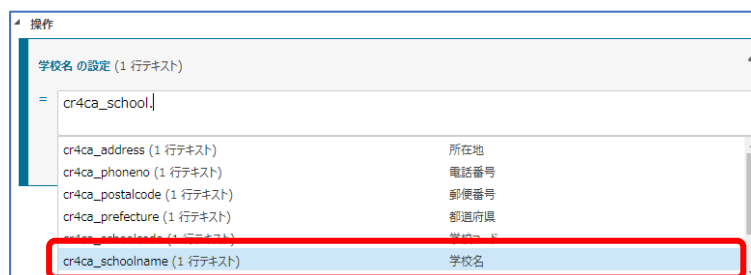


“school” と入力するとリストが絞られます。ここに表示された [学校] を選択してください。

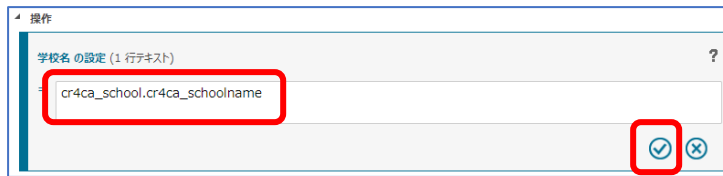


選択すると入力フィールドに [接頭辞_school] が入力されます。

この単語に続けてピリオド [.] を入力すると列名のリストが表示されますので、[学校名] を選択します。



[接頭辞_school.接頭辞_schoolname] と入力されたのを確認して、チェックアイコンをクリックします。

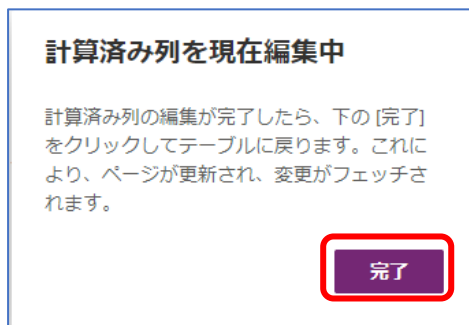


最後に、[保存して閉じる] をクリックして設定内容を確定します。



注: 別ウィンドウで作業している間、もとのウィンドウではこのようなメッセージが表示されています。

別ウィンドウの作業が終わった後は [完了] をクリックしてこのメッセージを閉じてください。



F. はい/いいえ列

「はい/いいえ」は選択肢型と同様に選択肢の値のみを受け付けるデータ型ですが、選択肢が常に 2 個である点と、選択肢の値は内部的には true/false である点が選択肢型と異なります。

ここでは学校訪問履歴情報テーブルの承認結果列を例として手順を説明します。

- ① 学校訪問履歴情報テーブルで新規列を追加し、以下のように属性を入力してください。

The screenshot shows a configuration window for a column named '承認結果'. The fields are as follows:

- 表示名*: 承認結果 (1)
- 名前*: cr4ca_ ApprovalResult (2)
- データ型*: はい/いいえ (3)
- 項目: 承認 (4) and 否認 (5)
- 既定値: 否認 (6)
- 必須*: 任意 (7)
- 検索可能: (8)

No.	項目	入力内容
1	表示名	承認結果
2	名前	ApprovalResult
3	データ型	はい/いいえ
4	はい(true)の表示	承認
5	いいえ(false)の表示	否認
6	既定値	否認
7	必須	任意

3.5. テーブルの列を作成する

ここまでの列定義手順を参考にしながら、各テーブルに残りの列を定義してください。

以下に各テーブルの列定義を示します。

注：前の手順で既に定義済みのものは背景をグレーにしてあります。

連続して作業するとテーブルの保存を忘れがちです。必ず「テーブルの保存」をクリックするよう注意してください。

【学校基本情報】

No.	表示名	名前	データ型	必須	初期値	備考
1	学校コード	SchoolCode	テキスト	○		プライマリ名の列
2	学校名	SchoolName	テキスト	○		
3	学校名カナ	SchoolNameKana	テキスト			
4	郵便番号	PostalCode	テキスト			
5	都道府県	Prefecture	テキスト			
6	所在地	Address	テキスト			
7	電話番号	PhoneNo	電話			
8	学校種別	SchoolType	選択肢			選択肢：公立、私立

【受験実績情報】

No.	表示名	名前	データ型	必須	初期値	備考
1	受験実績 ID	ID	オートナンバー	○		プライマリ名の列
2	学校	School	参照	○		関連：学校基本情報
3	年度	Year	整数	○		
4	学校名	SchoolNameRef	テキスト			計算：操作 > 学校. 学校名
5	一般志願者数	GeneralApplicants	整数			
6	一般受験者数	GeneralExaminees	整数			
7	一般合格者数	GeneralSuccessfullExaminees	整数			
8	一般入学者数	GeneralEnrollees	整数			
9	総合型志願者数	ComprehensiveApplicants	整数			
10	総合型受験者数	ComprehensiveExaminees	整数			
11	総合型合格者数	ComprehensiveSuccessfullExaminees	整数			
12	総合型入学者数	ComprehensiveEnrollees	整数			
13	推薦募集人数	AcceptingRecommenddees	整数			
14	推薦志願者数	RecommendApplicants	整数			
15	推薦受験者数	RecommendExaminees	整数			
16	推薦合格者数	RecommendSuccessfullExaminees	整数			
17	推薦入学者数	RecommendEnrollees	整数			

【学校訪問履歴情報】

No.	表示名	名前	データ型	必須	初期値	備考
1	訪問 ID	VisitID	オートナンバー	○		プライマリ名の列
2	学校	School	参照	○		関連：学校基本情報
3	訪問年度	VisitedYear	整数	○		
4	学校名	SchoolNameRef	テキスト			計算：操作 > 学校. 学校名
5	アポイント日時	AppointmentDateTime	日時			
6	アポイント担当者	Appointer	テキスト			
7	受付担当者	AcceptedBy	テキスト			
8	訪問日時	VisitedDateTime	日時			
9	訪問者	Visitor	テキスト			
10	訪問概要	VisitedSummary	テキスト領域			
11	担当者	Interviewee	テキスト			
12	担当者役割	IntervieweeRole	テキスト			
13	担当者電話番号	IntervieweePhone	電話			
14	担当者メールアドレス	IntervieweeEmail	メール			
15	卒業見込生徒数	ExpectedGraduates	整数			
16	進学希望数	ExpectedExaminees	整数			
17	進路未定者数	CourseUndecideds	整数			
18	当学志望者数	AspirantsForUs	整数			
19	申請に進む	ProceedToApplication	はい/いいえ		いいえ	
20	推薦募集人数	AcceptingRecommendees	整数			
21	申請日時	AppliedAt	日時			
22	申請者	Requester	メール			
23	承認者	Approver	メール			
24	承認結果	ApprovalResult	はい/いいえ		否認	はい：承認 いいえ：否認
25	承認/否認日時	ApprovedAt	日時			
26	願書送付要否	ApplicationRequired	はい/いいえ		送付不要	はい：送付要 いいえ：送付不要
27	願書送付予定日	ApplicationProvidingDate	日付のみ			
28	願書送付担当者	ApplicationProvider	メール			
29	願書送付日	ApplicationProvidedAt	日付のみ			
30	願書送付済み	ApplicationProvided	はい/いいえ		いいえ	
31	クローズド	Closed	はい/いいえ		いいえ	

4. 演習 2 ビューの作成

4.1. 作成するビューについて

テーブルに登録されたデータを、条件によるフィルターをかけたり並び順を変えたりして、必要な情報を見やすい状態かつ必要な列のみを表示させる機能がビューです。

モデル駆動型アプリでは参照したいデータを定義したビューを使用するため、前の演習で作成した各テーブルに対してビューを作成する必要があります。この演習では以下のビューを作成します。

※注：多くのビューは既定の（自動的に作成される）ものを変更して使用します。

No.	テーブル	ビュー	備考
1	学校基本情報	アクティブな学校基本情報	学校基本情報を一覧表示するために使用します。 (規定のビューを変更します)
2	受験実績情報	アクティブな受験実績情報	受験実績情報を一覧表示するために使用します。 (規定のビューを変更します)
3	学校訪問履歴情報	アクティブな学校訪問履歴情報	学校訪問履歴情報を一覧表示するために使用します。 (規定のビューを変更します)
4		願書送付状況	願書送付状況の確認用に推薦枠が承認されたデータを一覧表示するために使用します。 (ビューを新規作成します)
5	Option	学校訪問アポイント情報	※オプション キャンパス アプリでアポイント情報を表示するために使用します。 (ビューを新規作成します)
6		学校訪問実績情報	※オプション キャンパス アプリで訪問実績（履歴）一覧を表示するために使用します。 (ビューを新規作成します)

4.2. ビューを編集するための基本操作

ビューはテーブルに付随する機能です。まず初めにビューの定義を開く手順から説明します。

- ① ビューに関する操作を行うためにはまず対象のテーブルを選択（テーブル名をクリック）します。
以下、受験実績情報を例として作業を進めます。

テーブル ↑ ↓	名前 ↓	種類 ↓	カスタマイズ可能 ↓	タグ ↓	
学校基本情報	...	cr4ca_schoolinfo	カスタム	✓	カスタム
学校訪問履歴情報	...	cr4ca_schoolvisithistory	カスタム	✓	カスタム
受験実績情報	...	cr4ca_examinationresults	カスタム	✓	カスタム

探している項目が見つからない場合は、上記のフィルターをリセットして検索範囲を広げてください。

- ② テーブル名をクリックするとテーブルに対する操作のメニューが表示されます。
その中から [ビュー] をクリックします。

表示名 ↑ ↓	名前 ↓	データ型 ↓	種類 ↓	カスタ... ↓	必須 ↓	検索可能 ↓	
一般合格者数	...	cr4ca_generalsuccessfull...	整数	カスタム	✓	任意	✓
一般志願者数	...	cr4ca_generalapplicants	整数	カスタム	✓	任意	✓
一般受験者数	...	cr4ca_generalexaminees	整数	カスタム	✓	任意	✓

- ③ 定義済みのビューが一覧表示されます。

新しくビューを作成する場合は [ビューの追加] をクリックします。

名前 ↑ ↓	ビューの種類 ↓	状態 ↓	種類 ↓
アクティブな受験実績情報	共有ビュー 既定	アクティブ	標準
アクティブな受験実績情報の簡易検索	簡易検索ビュー 既定	アクティブ	標準
受験実績情報の高度な検索ビュー	高度な検索ビュー 既定	アクティブ	標準
受験実績情報関連ビュー	関連ビュー 既定	アクティブ	標準
受験実績情報検索ダイアログ ビュー	検索ダイアログ ボックス ビュー	アクティブ	標準
非アクティブな受験実績情報	共有ビュー	アクティブ	標準

④ 次の画面が開きます。

ビューの [名前] と、必要であればビューの [説明] (用途など) を入力して作成をクリックしてください。

※この演習で受験実績情報には新しいビューは作成しません。このビューは後ほど破棄します。

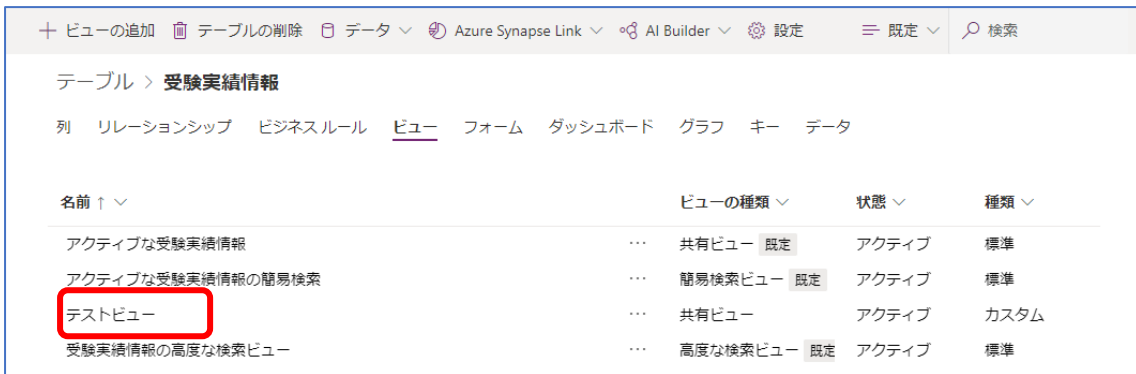


⑤ ビューの編集画面が表示されます。

ここでは編集をせず、そのまま [上書き保存] ⇒ [戻る] の順にクリックしてください。

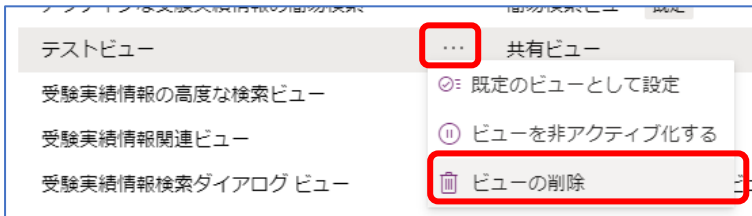


⑥ 一覧に新しいビューが追加されているのが確認できます。



Option

このビューは削除しておきましょう。(この作業はオプションです。実施しなくても演習に影響はありません)
ビューの名前の右側の [...] をクリックすると表示される [ビューの削除] をクリックします。



次のメッセージが表示されるので、[削除] をクリックしてください。

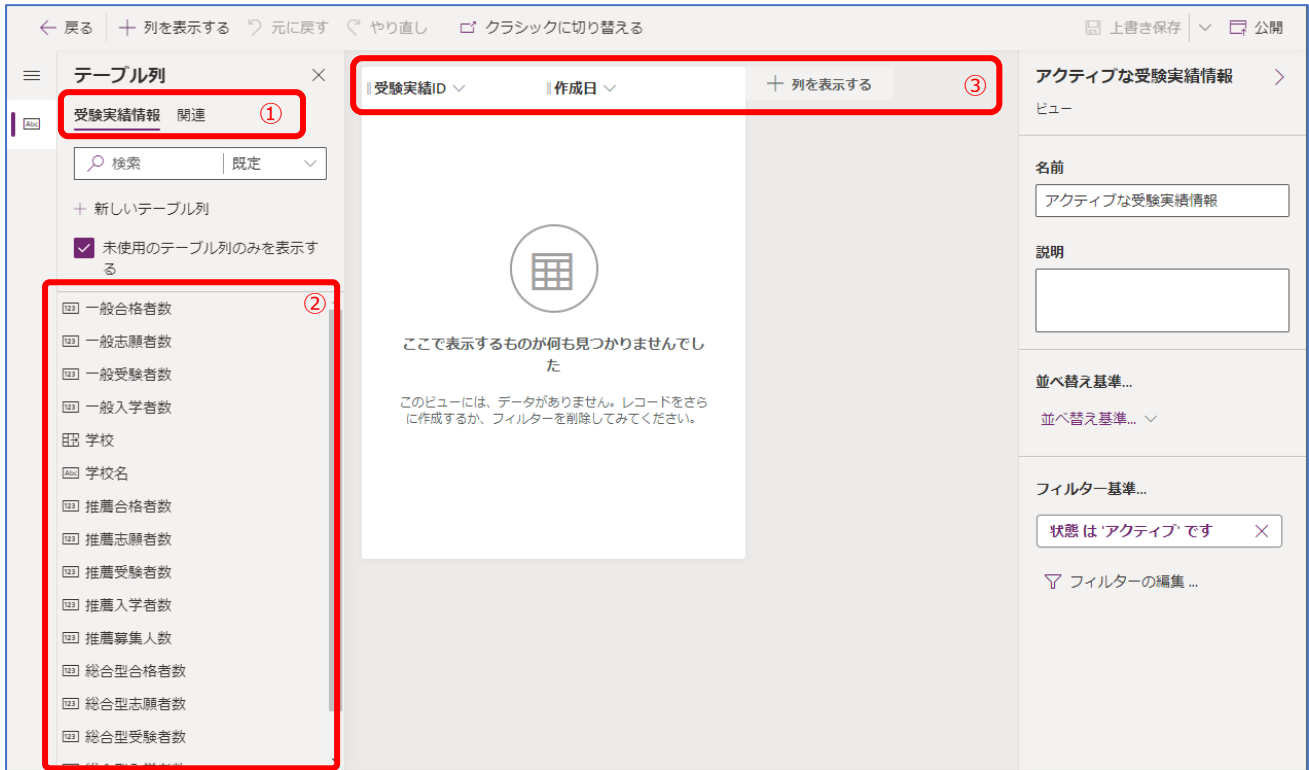


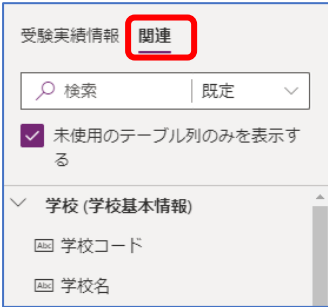
⑦ 既存のビューを編集するにはビューの [名前] をクリックします。



⑧ ビューの編集画面が表示されます。

ここでビューの編集画面の構成について整理しておきましょう。



No.	内容
①	<p>テーブルを選択します。初期表示では現在のテーブル（例では受験実績情報）が選択されています。現在のテーブルに関連するテーブルから列を参照する場合は「関連」を選択します。</p> 
②	<p>ビューに表示させる列を選択します。①で選択されたテーブルの列が一覧表示されています。この一覧から列名をクリックすることでビューに列を追加することができます。</p>
③	<p>選択済みの列が表示されます。ここで列の表示順 / 表示幅 / 並べ替え / フィルター条件の指定、さらに列の削除を行うことができます。</p>

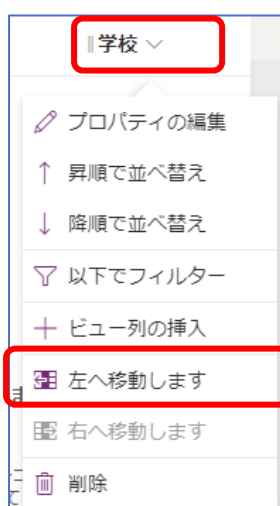
続いて、編集画面での操作を説明します。

- A. 現在のテーブルから列を追加するにはテーブル列の一覧にある項目をクリックします。
ここでは例として「学校」を追加します。

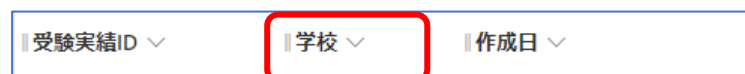


選択された列は現在ある列の右端に追加されました。

- B. 列の位置を調整するには移動したい列のメニューから「左へ（または右へ）移動します」をクリックします。
この例では学校列を左に移動します。



学校列が左に移動しました。



注：メニューを使わず、列を直接ドラッグ & ドロップしても移動が可能です。

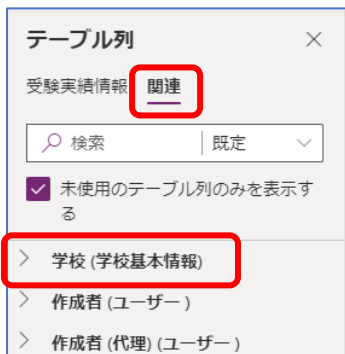
- C. 列を削除するには削除したい列のメニューから [削除] をクリックします。
この例では受験実績 ID 列を削除します。



受験実績 ID 列が削除されました。
同様に作成日列も削除してください。



- D. 関連テーブルから列を追加するにはまずテーブル列パネルの [関連] タブをクリックします。
この例では学校基本情報から学校名列をビューに追加します。



[関連] タブをクリックすることにより現在のテーブルに参照列として定義されている関連テーブルが一覧に表示されます。
[学校(学校基本情報)] をクリックして列名を展開し、[学校名] をクリックしてください。

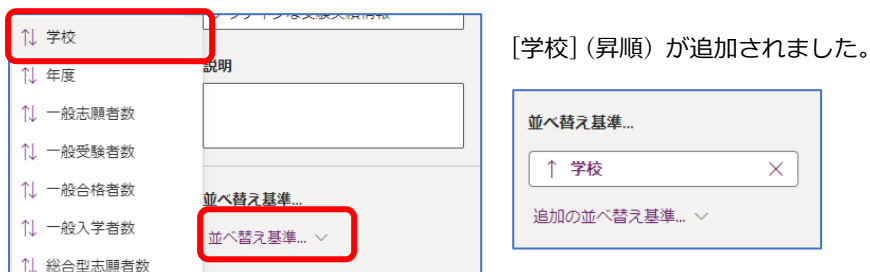


学校基本情報 (現在のテーブルでの列名は [学校]) から学校名列が追加されました。



E. データの並び順を設定するには右側パネルの「並べ替え基準」をクリックします。
ここでは「学校」(昇順)と「年度」(大きい順)を並べ替え基準に追加します。

① 「並べ替え基準」をクリックし、表示されるリストから「学校」を選択します。

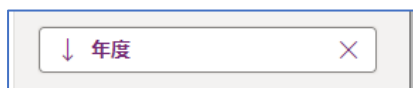


※注：図では「学校」以外の列も表示されていますが、本来この時点ではビューに追加済みの「学校」と「学校名」のみが表示されているはずで、後の手順「4.3. ビューを作成する」ですべての列を追加後は図のように表示されます。

② 「追加の並べ替え基準」をクリックし、表示されるリストから「年度」を選択します。



この時点では年度は「小さい順」なので、追加された「年度」をクリックします。



これで年度が「大きい順」となりました。このように並べ替え基準に追加した列名をクリックすることで昇順と降順を切り替えることができます。

F. データの表示条件を設定するには右側パネルの [フィルター of 編集] をクリックします。

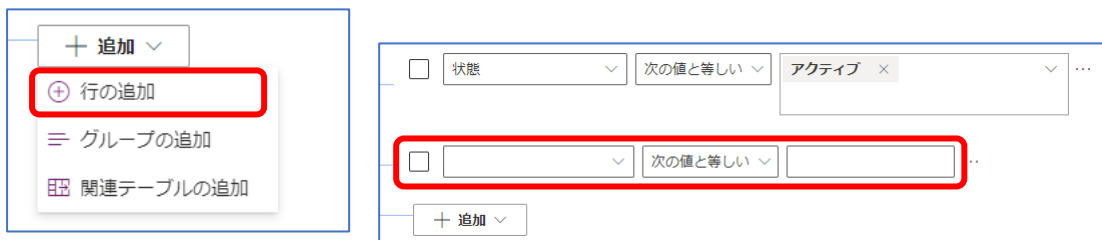
※注：ここでは操作のイメージのみを説明します。

[フィルター of 編集] をクリックすると、[フィルター of 編集] パネル (式ビルダー) が開きます

この例では既定の [アクティブな受験実績情報ビュー] を編集しているため、既に [状態がアクティブ] という条件が表示されています。



[追加] をクリックして新しい条件を追加します。例えば [行の追加] の選択ごとに条件を 1 個追加できます。



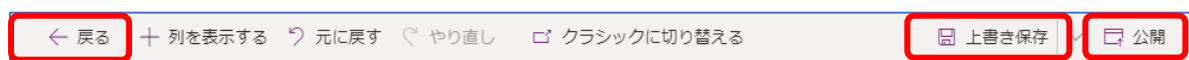
追加された条件の行で、列名、演算子、値を指定することで条件式を完成させます。

注：フィルター式の詳細については [モデル駆動型アプリのビューでフィルターを作成、編集する](#) で確認できます。

編集が完了したら [フィルター of 編集] パネル (式ビルダー) の [OK] ボタンをクリックします。

※ただしここでは何も変更しないので、[キャンセル] ボタンをクリックしてください。

G. ビューの設定が完了後、[上書き保存] ⇒ [公開] の順にクリックし、その後 [戻る] をクリックしてテーブル画面に戻ります。



4.3. ビューを作成する

ここまでの演習を参考に、必要なビューを定義しましょう。

【学校基本情報】

アクティブな学校基本情報

列定義			
No.	列名	備考	
1	学校コード		
2	学校名		
3	学校名カナ		
4	郵便番号		
5	都道府県		
6	所在地		
7	電話番号		
8	学校種別		
並び替え基準			
No.	列名	並び順	
1	学校コード	昇順	
フィルター条件			
No.	列名	演算子	値
-	(フィルター条件なし)		

【受験実績情報】

アクティブな受験実績情報

列定義		
No.	列名	備考
1	学校	
2	学校名(学校)	関連 > 学校 > 学校名
3	年度	
4	一般志願者数	
5	一般受験者数	
6	一般合格者数	
7	一般入学者数	
8	総合型志願者数	
9	総合型受験者数	

10	総合型合格者数		
11	総合型入学者数		
12	推薦募集人数		
13	推薦志願者数		
14	推薦受験者数		
15	推薦合格者数		
16	推薦入学者数		
並べ替え基準			
No.	列名	並び順	
1	学校	昇順	
2	年度	降順 (大きい順)	
フィルター条件			
No.	列名	演算子	値
-	(フィルター条件なし)		

【学校訪問履歴情報】

アクティブな学校訪問履歴情報

列定義		
No.	列名	備考
1	学校	
2	学校名(学校)	関連 > 学校 > 学校名
3	訪問年度	
4	アポイント日時	
5	訪問日時	
6	訪問者	
7	担当者	
8	推薦募集人数	
9	承認者	
10	承認/否認日時	
11	承認結果	
並べ替え基準		
No.	列名	並び順
1	訪問年度	降順 (大きい順)
2	学校	昇順
3	アポイント日時	降順 (新しい順)
4	訪問日時	降順 (古い順)

フィルター条件			
No.	列名	演算子	値
-	(フィルター条件なし)		

【学校訪問履歴情報】

願書送付状況 ※注：このビューは新規作成が必要です。

列定義		
No.	列名	備考
1	学校名(学校)	関連 > 学校 > 学校名
2	願書送付要否	
3	推薦募集人数	
4	願書送付予定日	
5	願書送付日	
6	願書送付済み	
7	願書送付担当者	

並べ替え基準

No.	列名	並び順
1	願書送付済み	昇順
2	願書送付予定日	昇順 (古い順)

フィルター条件

No.	列名	演算子	値
1	願書送付要否	次の値と等しい	送付要
2	願書送付予定日	今年 (※)	-

※注：[列名] に選択した列のデータ型により [演算子]、[値] の選択方法や内容が変化します。例えば上記の日付型の列のように、[演算子] の選択肢に [値] の意味を含むため [値] の入力が不要となるというような選択方法もあります。

Option

以下はオプションのキャンパス アプリ作成で使用するビューです。キャンパス アプリを作成しない場合はスキップしても問題ありません。

【学校訪問履歴情報】

学校訪問アポイント情報 ※注：このビューは新規作成が必要です。

列定義			
No.	列名	備考	
1	訪問 ID		
2	訪問年度		
3	学校		
4	学校名		
5	アポイント日時		
6	アポイント担当者		
7	受付担当者		
8	クローズド		
並べ替え基準			
No.	列名	並び順	
1	アポイント日時	昇順（古い順）	
フィルター条件 ※ [および] でグループ化			
No.	列名	演算子	値
1	クローズド	次の値と等しくない	はい
2	申請に進む	次の値と等しくない	はい

【学校訪問履歴情報】

学校訪問実績情報 ※注：このビューは新規作成が必要です。

列定義		
No.	列名	備考
1	訪問 ID	
2	学校	
3	学校名	
4	訪問年度	
5	訪問日時	
6	訪問者	
7	訪問概要	
8	担当者	

Option

9	担当者役割	
10	担当者電話番号	
11	担当者メールアドレス	
12	卒業見込生徒数	
13	進学希望数	
14	進路未定者数	
15	当学志望者数	
16	申請に進む	
17	推薦募集人数	
18	申請日時	
19	申請者	
20	承認者	
21	承認結果	
22	承認/否認日時	
23	願書送付要否	
24	願書送付担当者	
25	願書送付予定日	
26	願書送付日	
27	願書送付済み	
28	クローズド	

並べ替え基準

No.	列名	並び順
1	訪問日時	降順（新しい順）

フィルター条件 ※ [または] でグループ化

No.	列名	演算子	値
1	クローズド	次の値と等しい	はい
2	申請に進む	次の値と等しい	はい

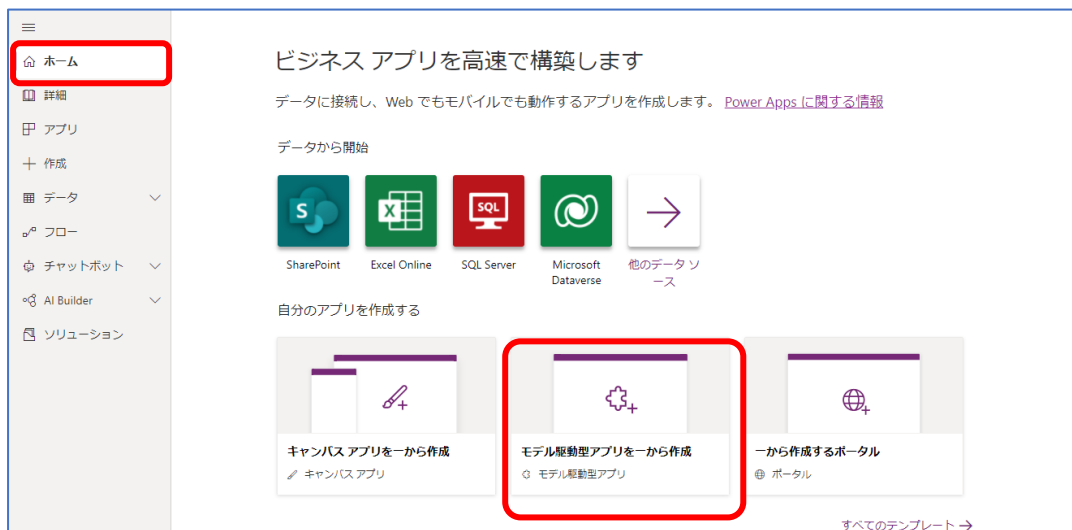
※注：[または] でグループ化する場合、[追加] ⇒ [グループの追加] で新たなグループを作成して [または] を選択します。

The screenshot shows a filter configuration window with a title 'および' (AND) and a dropdown menu set to 'または' (OR). The main area contains two filter conditions, each with a checkbox, a column name dropdown, an operator dropdown, and a value dropdown. The first condition is for 'クローズド' (Closed) with operator '次の値と等しい' (Equal to next value) and value 'はい' (Yes). The second condition is for '申請に進む' (Apply) with operator '次の値と等しい' (Equal to next value) and value 'はい' (Yes). Below the conditions are two '+ 追加' (Add) buttons.

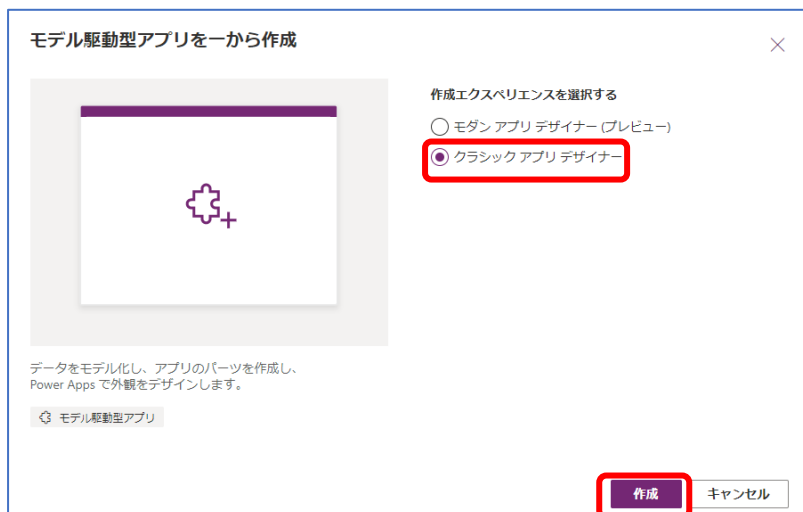
5. 演習 3 モデル駆動型アプリの作成

5.1. モデル駆動型アプリの新規作成

1. アプリを新規作成するには、[ホーム] から [モデル駆動型アプリを一から作成] をクリックします。



次の画面が表示されるので、[クラシック アプリ デザイナー] が選択されていることを確認の上 [作成] をクリックしてください。



新しいブラウザー（タブ）に次の画面が表示されますので、必要事項を入力します。

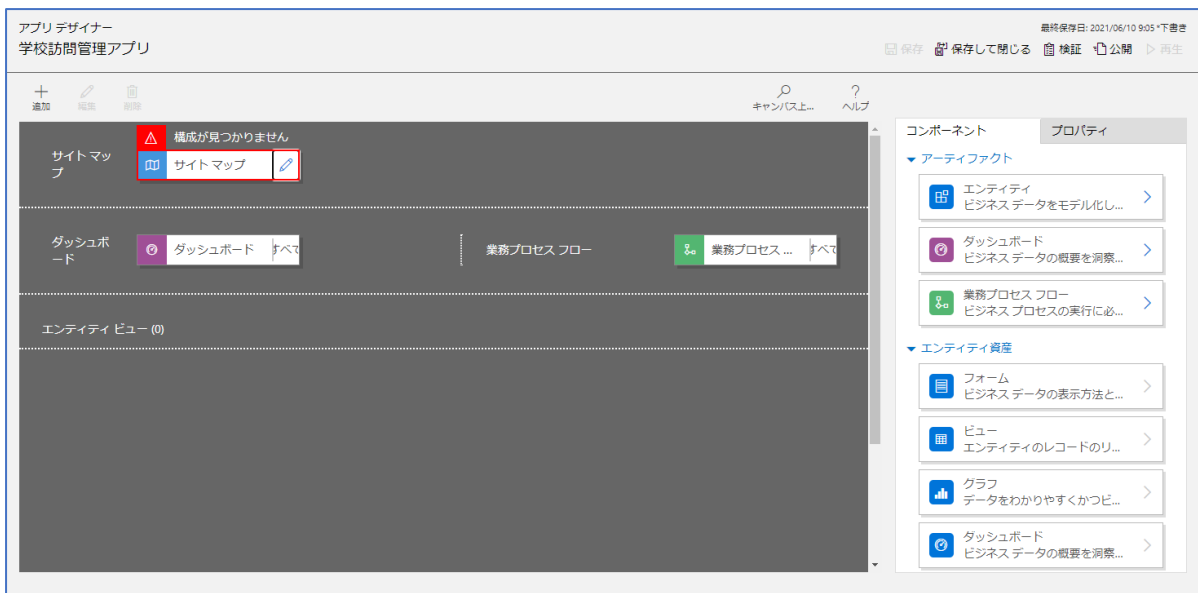
The screenshot shows the '新しいアプリを作成' (Create New App) interface in Power Apps. The '名前' (Name) field contains '学校訪問管理アプリ' (1). The '一意の名前' (Unique Name) field contains 'SchoolVisitManagerApp' (2). The '完了' (Completed) button is highlighted with a red box. The 'アプリタイトル' (App Title) is '学校訪問管理アプリ'. The 'アイコン' (Icon) section has '既定のイメージの使用' (Use default image) checked. The '統一インターフェイスのU...' (Unified interface U...) section has a URL: 'https://dynamics.com/Apps/unique/cr4ca_SchoolVisit'.

No.	項目	入力
1	名前	学校訪問管理アプリ
2	一意の名前	SchoolVisitManagerApp ※ 接頭辞はそのままにして、接頭辞の後に続けて入力してください。

※注：その他の項目は任意ですが、ここでは初期表示のままとしてください。

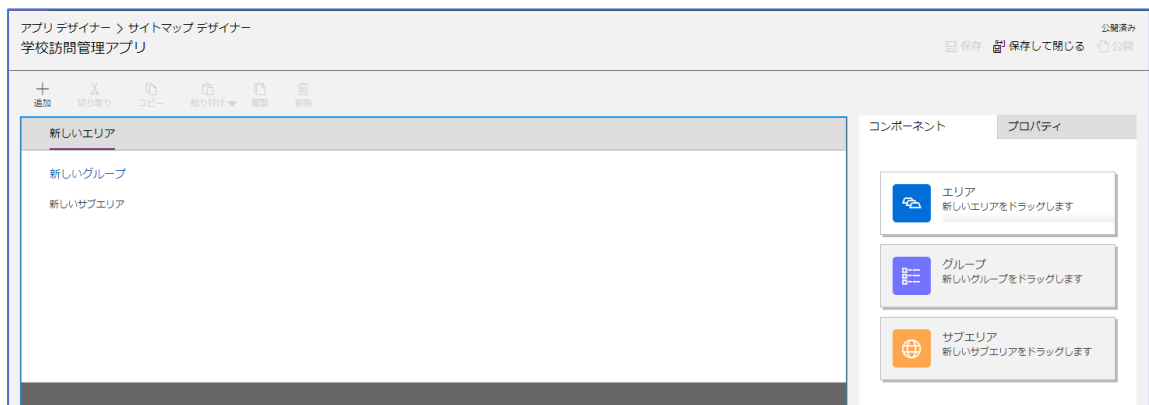
入力できたら [完了] をクリックしてください。

2. アプリ デザイナーが表示されます。

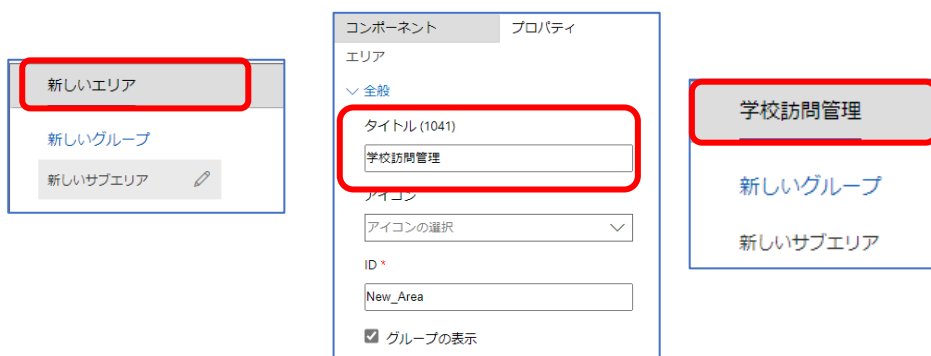


3. はじめに、サイトマップを定義します。

- ① [サイトマップ] の鉛筆アイコンをクリックして、サイトマップ デザイナーを開きます。



- ② [新しいエリア] をクリックすると、画面右側にあるパネルで [プロパティ] タブが表示されます。
[タイトル] に "学校訪問管理" と入力してください。



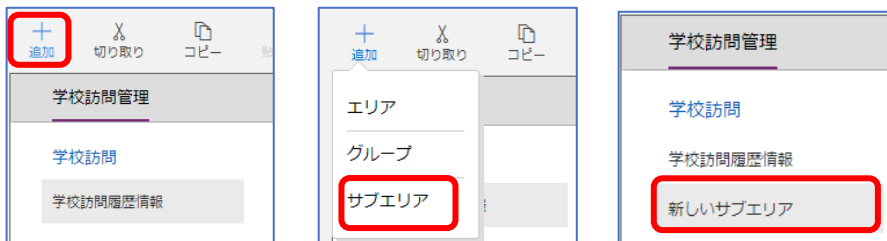
- ③ 同様に [新しいグループ] をクリックしてタイトルを“学校訪問”としてください。



- ④ [新しいサブエリア] をクリックし、画面右側パネルのプロパティタブで以下のように入力してください。

No.	項目	入力
1	種類	エンティティ
2	エンティティ	学校訪問履歴情報

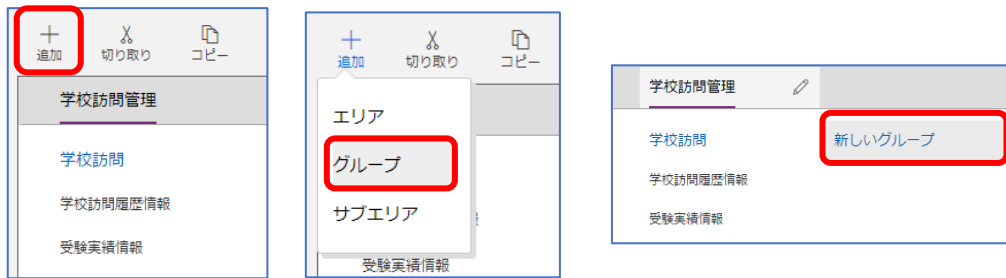
- ⑤ [追加] をクリックし、リストから [サブエリア] を選択してください。



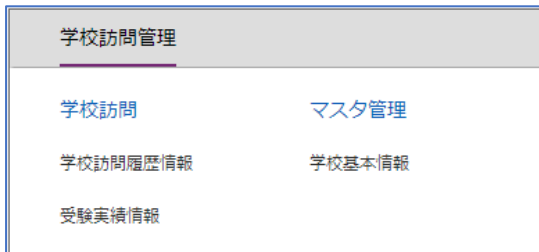
- ⑥ 追加された [新しいサブエリア] をクリックし、プロパティに以下のように入力してください。

No.	項目	入力
1	種類	エンティティ
2	エンティティ	受験実績情報

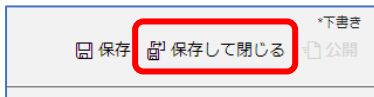
- ⑦ 次は [学校訪問管理] をクリック後、[追加] から [グループ] を選択してください。



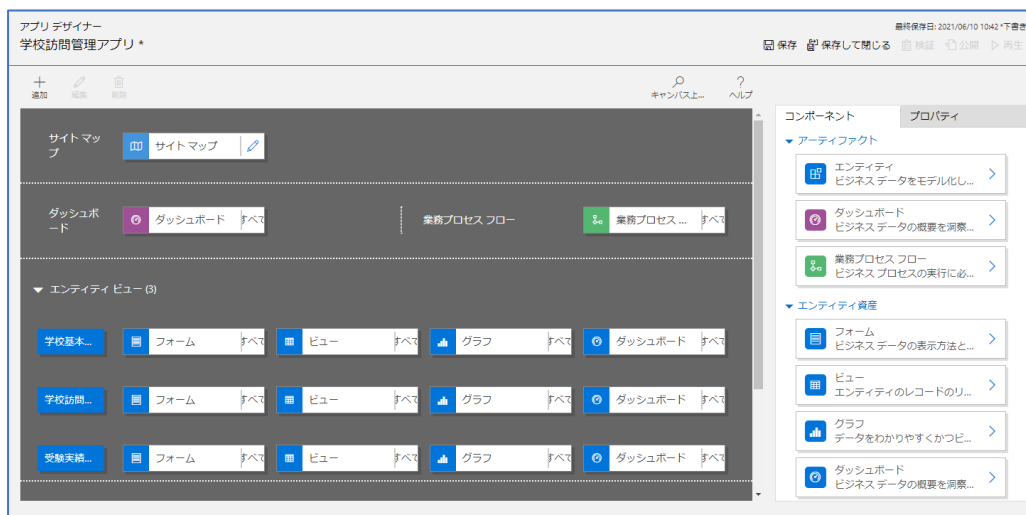
- ⑧ 前の手順の容量で [新しいグループ] のタイトルを “マスタ管理” に変更後、サブエリア [学校基本情報] を追加してください。完成形はこのようになります。



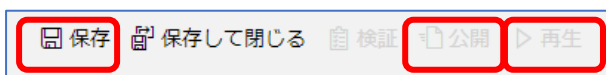
- ⑨ [保存して閉じる] をクリックしてサイトマップ デザイナーを終了します。



4. アプリ デザイナーに戻ったら現状の設定でアプリをテストします。



- ① [保存] ⇒ [公開] ⇒ [再生] の順にクリックし、アプリを起動します。

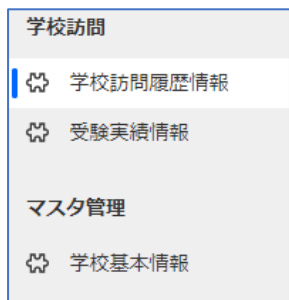


注：[保存] をクリックすると [公開]、[公開] をクリックすると [再生] が順に有効化されます。

アプリが起動されました。



アプリ画面の左ペインに、先ほど設定したサイトマップが表示されているのが確認できます。



※注：サイトマップ デザイナーで設定した [エリア] は [グループ] を取りまとめる器のような概念です。左ペインの下部でエリアを選択すると、その中に定義した内容にサイトマップが切り替わります。

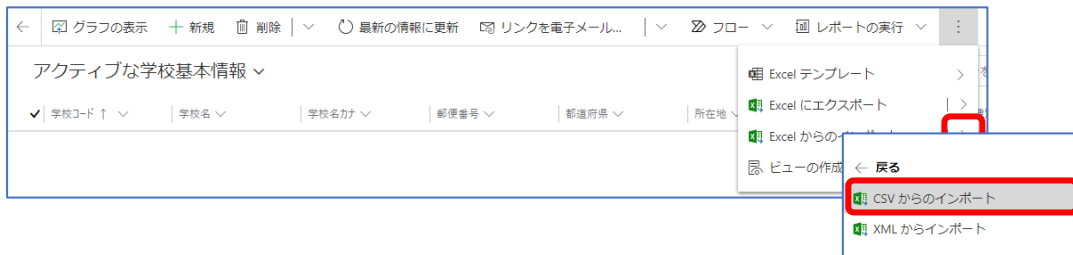
ただし、このアプリのようにエリアが一つしかない場合はエリア選択が表示されません。

② データをインポートして表示してみましょう。

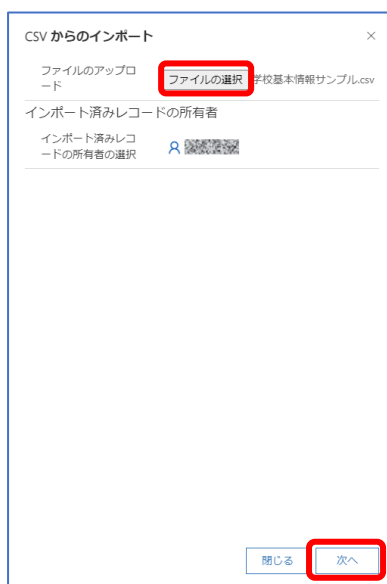
サイトマップから [学校基本情報] を選択してください。



次に、画面上部のメニューから [Excel からのインポート] ⇒ [CSV からのインポート] を選択してください。



[CSV からのインポート] が開いたら [ファイルの選択] をクリックして [学校基本情報サンプル.csv] を選択し、[次へ] をクリックしてください。



次に表示されたパネルで [マッピングの確認] をクリックしてください。

さらに次の画面で [インポート完了] をクリックします。

CSVからのインポート

① 学校基本情報サンプル.csv をインポートしようとして... 942 バイト

区切り文字の設定

データ区切り文字

二重引用符 ("")

フィールド区切り文字

コンマ (,)

先頭行を見出しとして使用する

重複データの設定

重複の許可 いいえ

戻る **マッピングの確認**

CSVからのインポート

① 学校基本情報サンプル.csv をインポートしようとして... 942 バイト

ソースファイルの各列を Dynamics 365 のエンティティ フィールドにマップします。

データ マップの名前

マップ属性 すべて表示

プライマリ フィールド

学校名 * 学校名

学校コード * 学校コード

オプションフィールド

ソース フィールド Dynamics 365 エンティティ フィールド

(変更しないでください) マップなし

(変更しないでください) マップなし

(変更しないでください) マップなし

学校名カナ 学校名カナ

学校種別 学校種別 (OptionSet)

戻る **インポート完了**

※注：基本的にウィザードが自動的に適切なテーブル列をマップしますが、ソースフィールド (CSV の項目) に対して適切なテーブル列がマップされていることを確認してください。テーブルに存在しないフィールドや計算列については [マップなし] で問題ありません。

次のメッセージが表示されますがそのまま [確認] をクリックしてください。

データを送信

1 つ以上のマップされていないフィールドがウィザードで検出されました。マップされていないフィールドを無視して続行しますか?

確認 キャンセル

次のパネルで再度 [完了] をクリックしてください。

CSVからのインポート

データがインポート用に送信されました

進行状況の追跡 **完了**

アプリ画面の [最新の情報に更新] をクリックしてください。

インポートに時間がかかっている場合は何度か繰り返し [最新の情報に更新] をクリックしてください。



学校コード	学校名	学校名カナ	郵便番号	都道府県	所在地	電話番号	学校種別
00000001	なんば高校	ナンバコウコウ	590-0400	大阪府	泉南郡熊取町	072345xxxx	私立
00000002	竜王高校	リュウオウコウ...	520-2500	滋賀県	蒲生郡竜王町	074832xxxx	公立
00000003	天龍寺高校	テンリュウジコ...	616-8300	京都府	京都市右京区嵯峨	075881xxxx	私立
00000004	まほろば高校	マホロバコウコウ	630-8100	奈良県	奈良市芝辻町	074222xxxx	公立
00000005	三宮高校	サンノミヤコウ...	650-0000	兵庫県	神戸市中央区波...	074222xxxx	私立
00000006	手塚高校	テマリコウコウ	640-8500	和歌山県	和歌山市七番丁	073456xxxx	公立

インポートが成功すると上のようなデータが表示されます。

同じ要領で、[受験実績情報] に [受験実績情報サンプル.csv] をインポートしてください。

※注：マッピングの確認でソースフィールドに対して適切なテーブル列がマップされていることを確認してください。[受験実績 ID] (オートナンバー) と [学校名(学校)] (計算) 及びテーブルに存在しないフィールドは [マップなし] で問題ありません。



学校	学校名(学校)	年度	一般志願者数	一般受験者数	一般合格者数	一般入学者数	総合志願者数	総合受験者数	総合合格者数	総合入学者数
00000001	なんば高校	2,021	38	35	20	18	10	10	10	10
00000001	なんば高校	2,020	40	40	35	30	11	10	9	9
00000001	なんば高校	2,019	35	35	30	28	10	10	10	10
00000002	竜王高校	2,021	20	20	15	12	7	7	7	7
00000002	竜王高校	2,020	18	18	13	12	8	8	7	7
00000002	竜王高校	2,019	15	15	10	10	9	8	7	7
00000003	天龍寺高校	2,021	5	5	2	2	4	4	2	1
00000003	天龍寺高校	2,020	10	10	8	8	3	2	2	1
00000003	天龍寺高校	2,019	20	20	14	13	2	2	2	2

③ インポートしたデータを編集してみます。

[アクティブな受験実績情報] で 1 番上の行を選択し、[編集] をクリックしてください。



学校	学校名(学校)	年度	一般志願者数	一般受験者数	一般合格者数	一般入学者数	総合志願者数	総合受験者数	総合合格者数	総合入学者数
00000001	なんば高校	2,021	38	35	20	18	10	10	10	10
00000001	なんば高校	2,020	40	40	35	30	11	10	9	9
00000001	なんば高校	2,019	35	35	30	28	10	10	10	10
00000002	竜王高校	2,021	20	20	15	12	7	7	7	7

フォームが表示されますが、デフォルトのフォームではテーブル列が不足していることが確認できます。次のステップで各テーブルのフォームを作成します。

現在のブラウザタブを閉じてアプリ デザイナーに戻ってください。



00000000000001002
受験実績情報

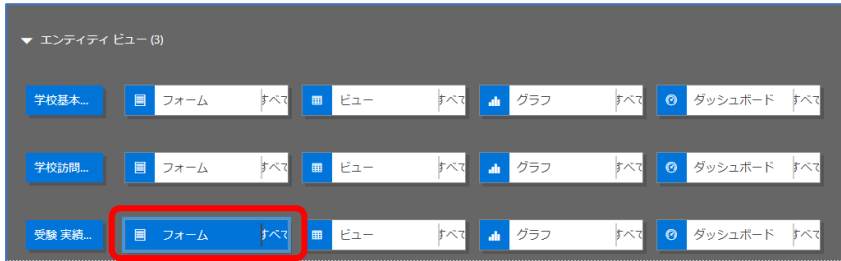
全般 関連

受験実績ID * 00000000000001002

所有者 * 

5. [エンティティビュー] に表示されている各テーブルの [フォーム] からフォームの登録作業を行います。

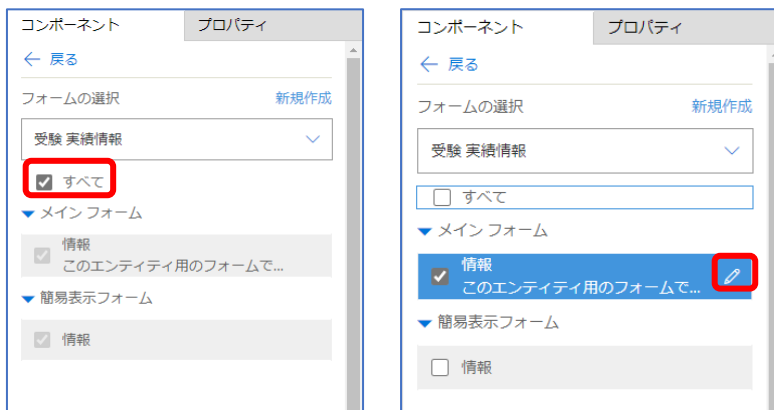
① [受験実績情報] の [フォーム] をクリックしてください。



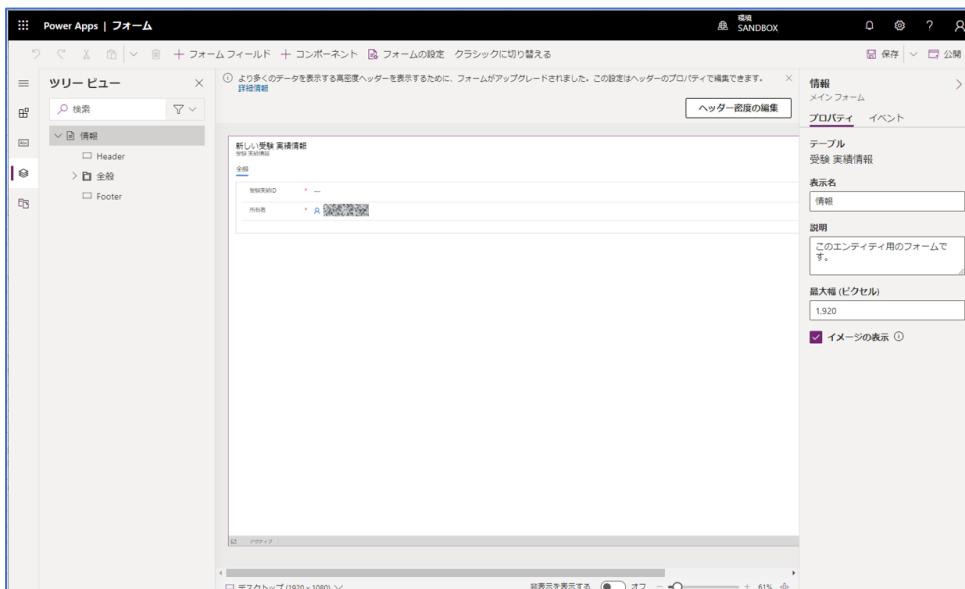
右側にあるパネルの [コンポーネント] タブに受験実績情報テーブルのフォームが表示されます。

[すべて] のチェックボックスをオフにしてください。

オフにすると [メインフォーム] の [情報] がチェックされた状態となります。この中の編集アイコンをクリックしてください。※編集アイコンはマウスを近づけると表示されます。



新しいウィンドウでフォーム編集画面が表示されます。

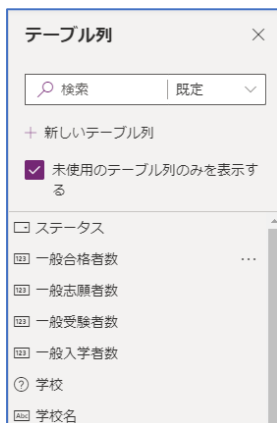


- ② 現在表示されている [受験実績 ID] と [所有者] は人が入力する項目ではないので除外します。
ただし、これらは必須の項目なので削除できません。項目を選択すると右パネルの [プロパティ] タブに [非表示] のチェックボックスが表示されるのでチェックをオンにします。

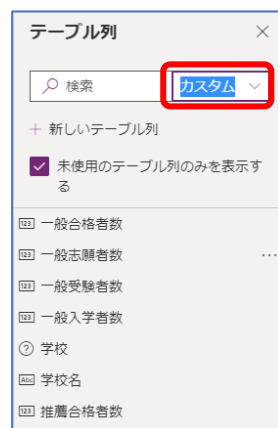


2 項目とも非表示に設定したら次に進みます。

- ③ 画面上部の [フォームフィールド] をクリックしてください。
画面左に [テーブル列] のパネルが表示されます。ここに一覧表示されている項目名をクリックすることにより、フォーム上に表示する入力項目を指定していきます。



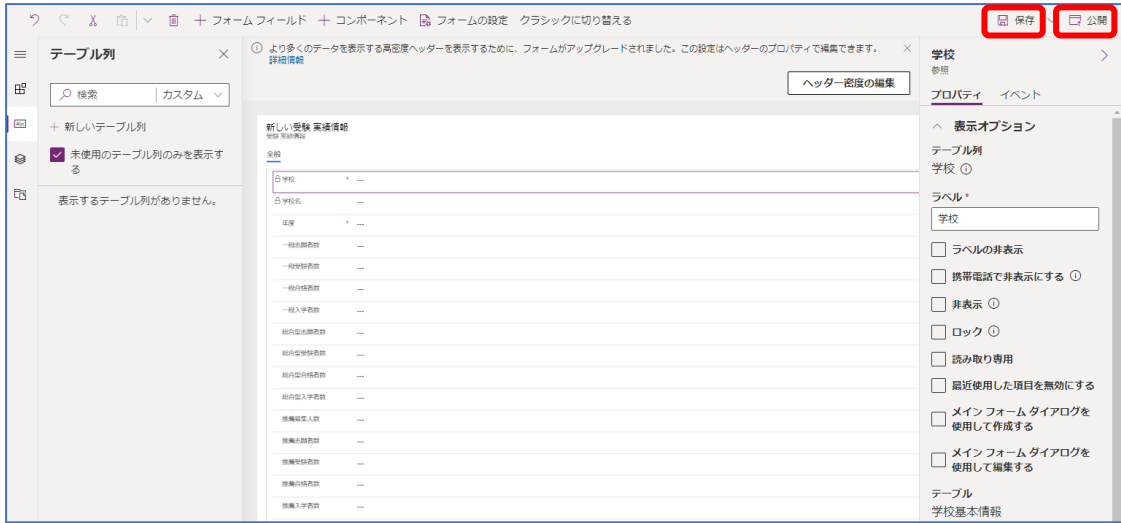
リストの項目を絞り込むため、[カスタム] のフィルターを設定します。



各項目をクリックしてフォームに追加してください。

※注：フォームに追加された項目はドラッグ&ドロップすることで表示順を変更することが可能です。

- ④ すべての項目を追加したら、[保存] ⇒ [公開] の順にクリックし、最後に画面を閉じてください。



- ⑤ ここまでの手順と同じ要領で他のテーブルにもフォームを登録してください。

各フォームの項目とその表示順は下表に従って設定してください。

No.	学校基本情報	受験実績情報※前の手順で定義済み	学校訪問履歴情報
1	学校コード	学校	学校
2	学校名	学校名	学校名
3	学校名カナ	年度	訪問年度
4	郵便番号	一般志願者数	—
5	都道府県	一般受験者数	—
6	所在地	一般合格者数	—
7	電話番号	一般入学者数	—
8	学校種別	総合型志願者数	—
9	—	総合型受験者数	—
10	—	総合型合格者数	—
11	—	総合型入学者数	—
12	—	推薦募集人数	—
13	—	推薦志願者数	—
14	—	推薦受験者数	—
15	—	推薦合格者数	—
16	—	推薦入学者数	—

- ⑥ アプリ デザイナーで [保存] ⇒ [公開] 後、再度 [再生] してフォームのテストをしてください。
 [編集] でフォームを開いて確認後は、[←] または [保存して閉じる] でビューの画面に戻ってください。



下記はフォームの表示例です。

【学校基本情報】

学校コード	* 00000001
学校名	* なんぼ高校
学校名カナ	ナンボコウコウ
郵便番号	590-0400
都道府県	大阪府
所在地	泉南郡熊取町
電話番号	072345xxxx
学校種別	私立

【受験実績情報】

学校	* 00000001
学校名	なんぼ高校
年度	* 2,021
一般志願者数	38
一般受験者数	35
一般合格者数	20
一般入学者数	18
総合型志願者数	10
総合型受験者数	10
総合型合格者数	10
総合型入学者数	10
推薦募集人数	10
推薦志願者数	10
推薦受験者数	10
推薦合格者数	8
推薦入学者数	8

【学校訪問履歴情報】

このテーブルにはまだデータがないので、[新規] をクリックしてフォームを表示します。



学校	*	---
学校名		---
訪問年度	*	---

ここで、[学校] の検索アイコンをクリックしてみます。

学校	学校の検索
学校名	学校基本情報
訪問年度	00000001 2021/06/10 13:42
	00000002 2021/06/10 13:42
	00000003 2021/06/10 13:42
	00000004 2021/06/10 13:42
	+ 学校基本情報の新規作成

リストには学校コードと何らかの日時が表示されていますが、これは既定で検索用ビューがリスト表示に使用されているためです。[ビューの変更] で学校名が表示されるビューに切り替えてみましょう。

	+ 学校基本情報の新規作成	ビューの変更
学校	学校の検索	ビューの変更
学校名	学校基本情報検索ダイアログビュー (既定)	
訪問年度	アクティブな学校基本情報	
	非アクティブな学校基本情報	
	<input type="checkbox"/> 自分のレコードのみ	

学校の検索で学校名が表示されるようになりました。

学校	学校の検索
学校名	学校基本情報
訪問年度	アクティブな学校基本情報
	00000001 なんば高校
	00000002 竜王高校
	00000003 天龍寺高校
	00000004

学校訪問履歴情報は業務フローを管理するテーブルであり、業務フローのステップごとに登録すべき項目が異なるため、フォームとしては最低限の項目のみを設定しました。

次の演習では、業務フローのステップに応じて入力項目をコントロールする仕組みを、ビジネス プロセス フローを使用して実現する手順を説明します。

5.2. ビジネス プロセス フローの作成

1. フローを新規作成するには、メニューから [フロー] を選択します。

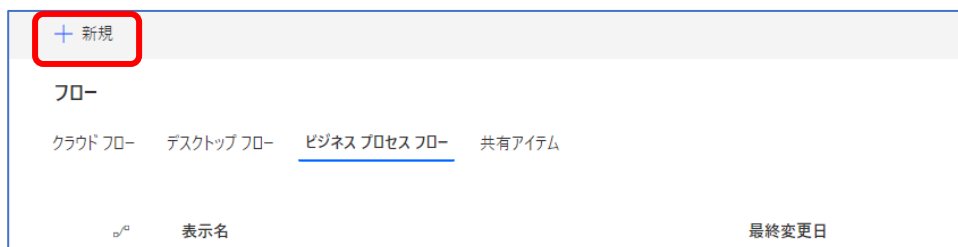


[フロー] を開くと次のような画面が表示されます。

この演習ではビジネス プロセス フローを作成するので、[ビジネス プロセス フロー] を選択してください。



[新規] をクリックします。



次の画面が表示されるので、必要事項を入力してください。入力後 [作成] をクリックします。

ビジネス プロセス フローの作成 ×



1 人または複数のエンドユーザーが定義されたプロセスに従って、複数のステップから成るタスクを完了することができます。

例:

- 新しい従業員のオンボードとトレーニングを行う
- 潜在顧客をナーチャリングし、営業案件を完結させる

フロー名 *

 ①

名前 *

 ②

テーブルを選択する * ①

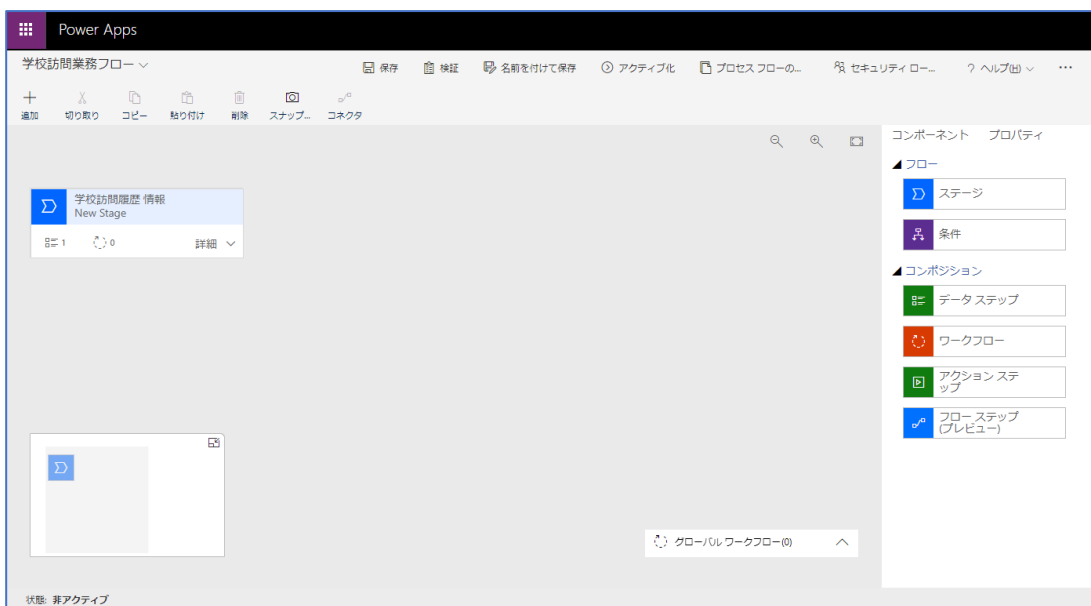
 ×

学校訪問履歴 情報 ③

作成
キャンセル

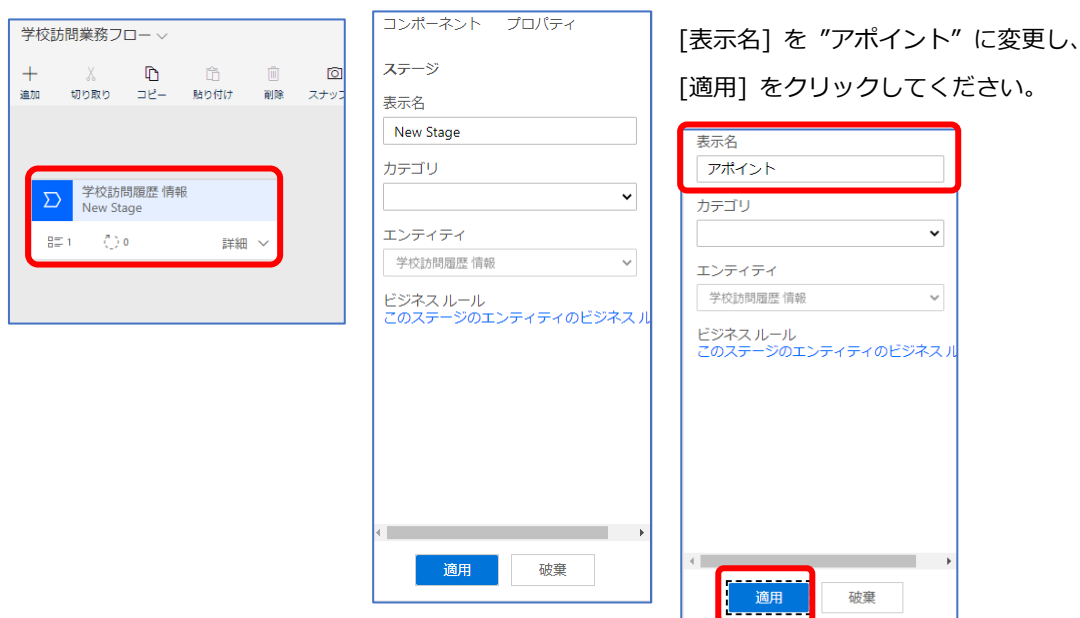
No.	項目	入力
1	フロー名	学校訪問業務フロー
2	名前	SchoolVisitBusinessFlow
3	テーブルを選択する	“学校訪問履歴” と入力するとその下にテーブル名が表示されるので、クリックしてチェックをオンにします。

ブラウザの新しいタブでフローが開きます。



2. 最初のステージを編集します。

[New Stage] をクリックしてください。画面右側のパネルに選択したステージのプロパティが表示されます。



ステージの名前が変更されました。



3. [アポイント] ステージのデータステップを定義します。

ステージ内の [詳細] をクリックして展開し、[データステップ#1] を選択してください。

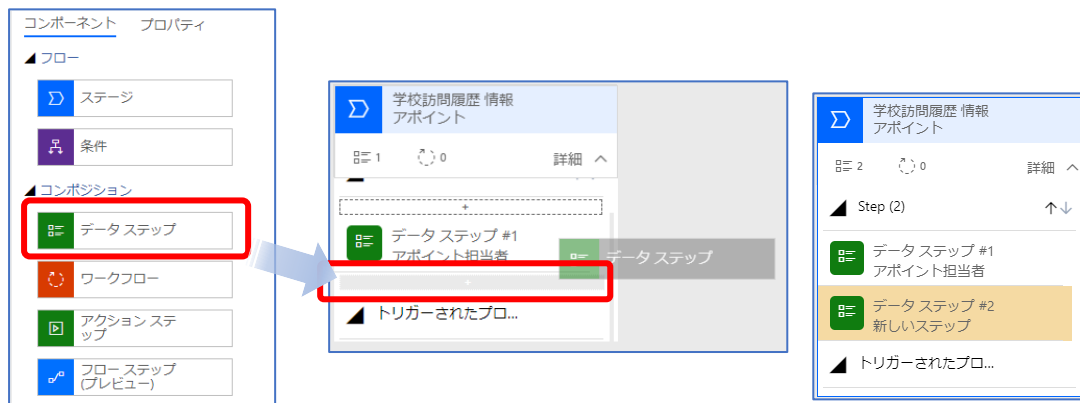
画面右側のパネルに、選択したデータステップのプロパティが表示されるので、表のように入力して [適用] をクリックしてください。

No.	項目	入力
1	ステップ名	アポイント担当者 (※)
2	データフィールド	アポイント担当者
3	必須	必須 (オン)

※注：最初のステップのみ手入力が必要ですが、2 ステップ目以降は自動的に設定されます。

画面右側のパネルで [コンポーネント] タブを選択してください。

利用可能なコンポーネントが表示されるので、[コンポジション] 内の [データステップ] をドラッグして [アポイント] ステージのデータステップ #1 [アポイント] の下にドロップします。



追加した [データステップ#2] を選択し、プロパティを入力して [適用] をクリックしてください。

No.	項目	入力
1	ステップ名	受付担当者 (※)
2	データフィールド	受付担当者
3	必須	必須 (オン)

※注：ステップ名を変更せず、データフィールドを先に設定してみてください。

ステップ名には自動的にデータフィールドに入力した値が設定されます。

同様に [データステップ#3] を追加し、以下のようにプロパティを設定してください。

No.	項目	入力
1	ステップ名	アポイント日時
2	データフィールド	アポイント日時
3	必須	必須 (オン)

これで [アポイント] ステージの設定は完了しました。

4. フローの全体像を定義します。

以降で追加するステージへのデータステップ定義は最後にまとめて行っていただきます。

ステージを追加するには、[コンポーネント] タブから [フロー] 内の [ステージ] をドラッグ&ドロップします。

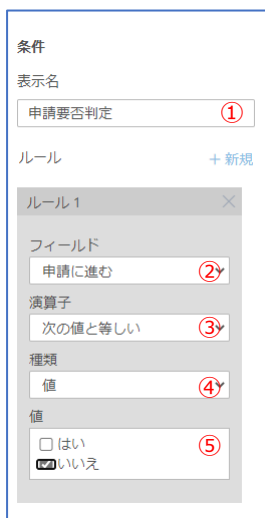


このステージの表示名を“学校訪問”としてください。

[学校訪問] ステージの次は条件分岐を追加します。追加方法はステージの時と同様です。



条件のプロパティは以下のように設定してください。

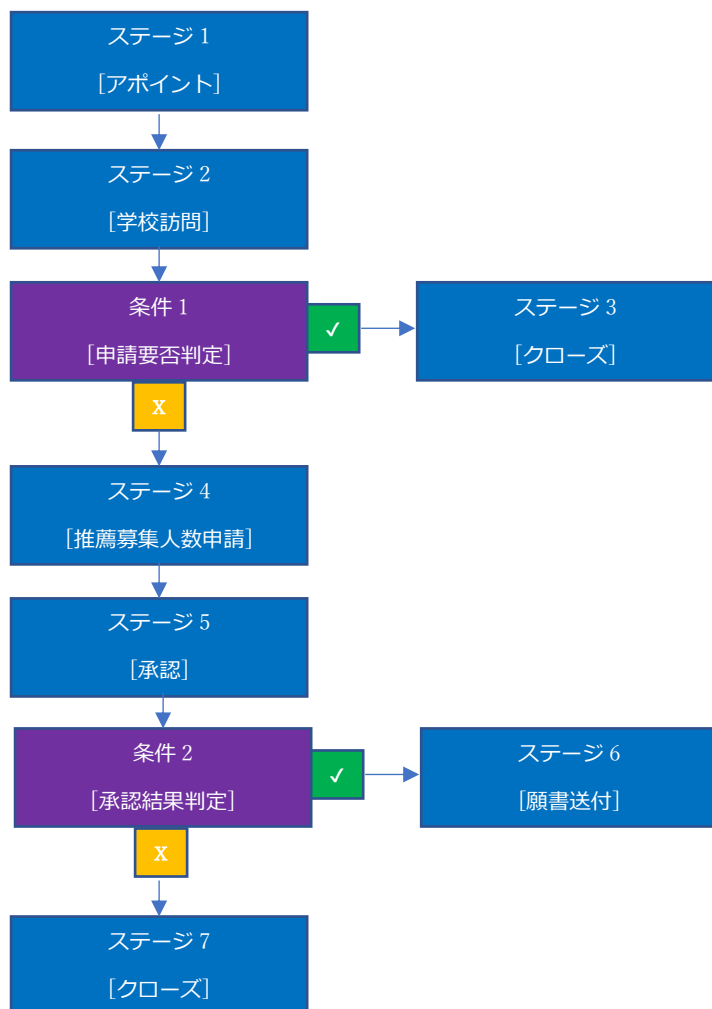


No.	項目	入力
1	表示名	申請要否判定
ルール 1		
No.	項目	入力
2	フィールド	申請に進む
3	演算子	次の値と等しい
4	種類	値
5	値	いいえ

※注：複数のルールが必要な場合は [新規] をクリックして追加可能です。

条件から接続するのは、✓ 側 (右) がルールに合致した場合のステージ、X 側 (下) が合致しない場合のステージです。[申請要否判定] は ✓ 側に [クローズ]、X 側に [推薦募集人数申請] ステージをそれぞれ追加してください。

この要領で、フロー全体の構成を次の図のとおり設定してください。



5. 各ステージ、条件の内容を入力してフローを完成させます。

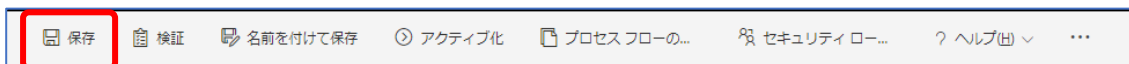
下表に従って必要なステップや条件プロパティを設定してフローを完成させてください。

※注：連続して作業するとプロパティ設定の [適用] をクリックし忘れがちです。 [適用] をクリックしないと入力した内容が反映されませんので注意してください。

ステージ	表示名	No.	ステップ	データフィールド (ステップ名)	必須
ステージ 1	アポイント	1	データステップ#1	アポイント担当者	✓
		2	データステップ#2	受付担当者	✓
		3	データステップ#3	アポイント日時	✓
ステージ 2	学校訪問	1	データステップ#1	訪問者	✓
		2	データステップ#2	訪問日時	✓
		3	データステップ#3	担当者	✓
		4	データステップ#4	担当者役割	
		5	データステップ#5	担当者電話番号	
		6	データステップ#6	担当者メールアドレス	
		7	データステップ#7	卒業見込生徒数	
		8	データステップ#8	進学希望数	
		9	データステップ#9	進路未定者数	
		10	データステップ#10	当学志望者数	
		11	データステップ#11	訪問概要	✓
		12	データステップ#12	申請に進む	
ステージ 3	クローズ	1	データステップ#1	クローズド	
ステージ 4	推薦募集人数申請	1	データステップ#1	推薦募集人数	✓
		2	データステップ#2	申請日時	✓
		3	データステップ#3	申請者	✓
		4	データステップ#4	承認者	✓
ステージ 5	承認	1	データステップ#1	承認結果	
		2	データステップ#2	承認/否認日時	
		3	データステップ#3	願書送付要否	
		4	データステップ#4	願書送付予定日	
		5	データステップ#5	願書送付担当者	
ステージ 6	願書送付	1	データステップ#1	願書送付日	✓
		2	データステップ#2	願書送付済み	✓
		3	データステップ#3	クローズド	✓
ステージ 7	クローズ	1	データステップ#1	クローズド	

条件	表示名	ルール	No.	項目	入力内容
条件 1	申請要否判定	ルール 1	1	フィールド	申請に進む
			2	演算子	次の値と等しい
			3	種類	値
			4	値	いいえ
条件 2	承認結果判定	ルール 1	1	フィールド	承認結果
			2	演算子	次の値と等しい
			3	種類	値
			4	値	承認

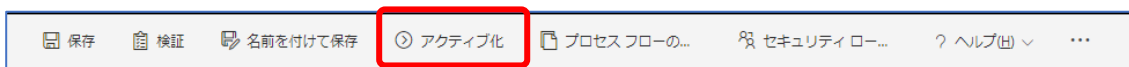
すべての入力が完了したら、[保存] をクリックしてください。



次に [検証] をクリックしてください。「検証が成功しました。」というメッセージが表示されれば OK です。



最後に [アクティブ化] をクリックしてください。



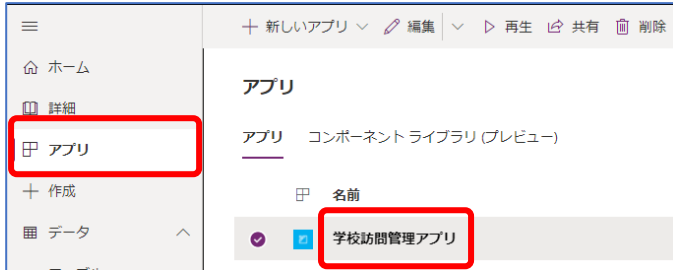
次のメッセージが表示されるので、[アクティブ化] をクリックしてください。



アクティブ化には数分かかる可能性があります。アクティブ化が終わったらブラウザで現在のタブを閉じてください。

6. 完成したフローをテストします。

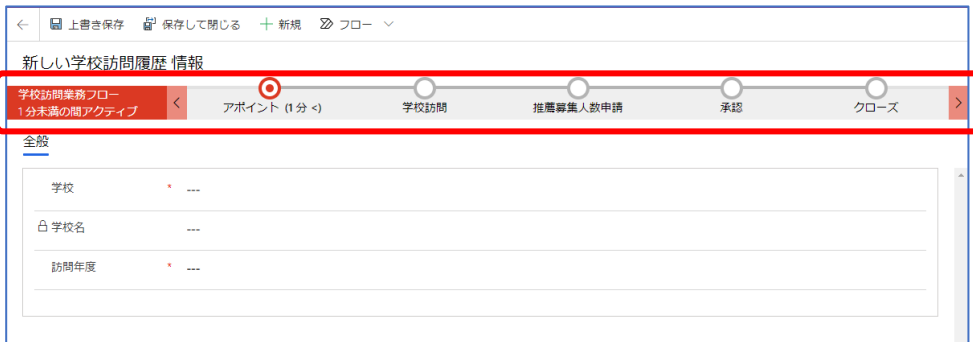
Power Apps のメニューから [アプリ] を選択し、表示されたアプリの一覧から [学校訪問管理アプリ] をクリックしてください。この操作でアプリがブラウザーの新しいタブで起動します。



[学校訪問履歴情報] が表示されていることを確認の上、[新規] をクリックします。



フォームが表示されます。以前と異なり、フローのステップが表示されていることが確認できます。



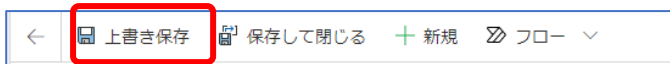
フローのテストをするためにはまず新しいデータを登録する必要があります。

図のように [学校] と [訪問年度] を入力してください。

学校 *	00000001
学校名	---
訪問年度 *	2021

データはまだ登録されていないため、計算フィールドの [学校名] が表示されていません。

登録するには [上書き保存] をクリックします。



学校	* 00000001
学校名	なんば高校
訪問年度	* 2,021

データが登録され、学校名が表示されました。

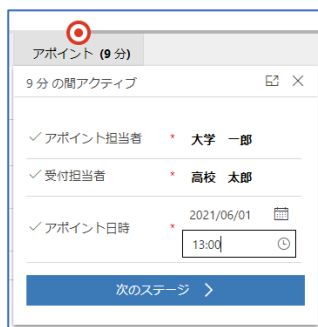
運用上は訪問実績がクローズ (申請をしない / 申請が否認された / 願書送付完了) するまで、このデータに対して情報を入力することを想定しています。

それでは、各ケースを順にテストしましょう。

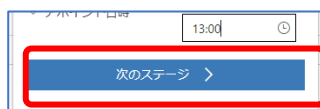
① 初回訪問 (ご挨拶のみ ; 申請は行わない)

- a. [アポイント] ステージをクリックして展開し、データを入力してください。

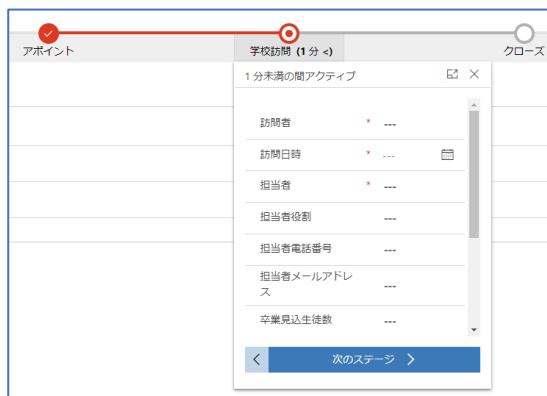
入力例)



- b. [次のステージ] をクリックしてください。



[アポイント] のステージが完了し、次の [学校訪問] ステージが開きます。

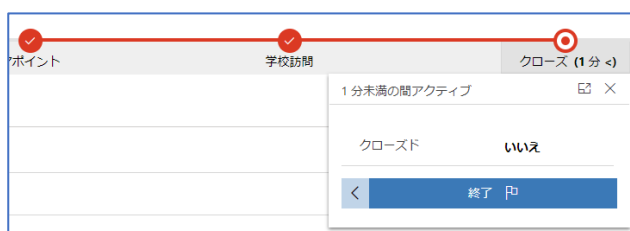


[学校訪問] ステージに入力して、[次のステージ] をクリックしてください。

入力例)

No.	項目	入力 (例)
1	訪問者	大学 一郎
2	訪問日時	(アポイント日時と同じ)
3	担当者	高校 太郎
4	担当者役割	進路指導
5	担当者電話番号	(任意)
6	担当者メールアドレス	(任意)
7	卒業見込生徒数	(未入力)
8	進学希望数	(未入力)
9	進路未定者数	(未入力)
10	当学志望者数	(未入力)
11	訪問概要	ご挨拶のみ
12	申請に進む	いいえ

このルートでは最後のステージとなる [クローズ] に進みます。



[クローズド] は運用のための項目ですが、手動で [はい] に変更して [終了] をクリックしてください。



c. 図のような状態となることを確認の上、[保存して閉じる] をクリックしてください。

これで最も短いフローの確認ができました。



② 2 回目の訪問（進路指導の状況について面談、推薦募集人数を申請するも否認）

a. 初めの手順の要領で新規データを登録し、[アポイント] ステージを入力してください。

入力後は [次のステージ] をクリックしてください。

入力例)

No.	項目	入力 (例)
1	アポイント担当者	大学 一郎
2	受付担当者	高校 太郎
3	アポイント日時	(任意 ; 1 回目より後の日付)

b. [学校訪問] ステージに入力例を参考に入力してください。

※注 : [申請に進む] は必ず [はい] と入力してください。

入力例)

No.	項目	入力 (例)
1	訪問者	大学 一郎
2	訪問日時	(アポイント日時と同じ)
3	担当者	高校 太郎
4	担当者役割	進路指導
5	担当者電話番号	(任意)
6	担当者メールアドレス	(任意)
7	卒業見込生徒数	150
8	進学希望数	90
9	進路未定者数	10
10	当学志望者数	30
11	訪問概要	当学への進学に力をいれて おられるので、推薦枠を多 く確保したい。
12	申請に進む	はい

※注 : モデル駆動型アプリでは定期的にデータが自動保存されます。

入力中にデータ保存が実行されると、入力中の項目が未入力状態に戻ってしまう
ことがあります。このような場合は再度入力してください。

入力が終わると、フローが図のようになっていることを確認してください。

これは [申請に進む] を [はい] としたことから [申請要否判定] の条件が機能していることを意味します。



c. [次のステージ] をクリックして [推薦募集人数申請] に進みます。

その後、入力例を参考に入力後、再度次のステージ (承認) に進めてください。

入力例)

No.	項目	入力 (例)
1	推薦募集人数	30
2	申請日時	(任意)
3	申請者	(ご自身のメールアドレス)
4	承認者	(ご自身のメールアドレス)

d. [承認] ステージの入力を行ってください。

入力例)

No.	項目	入力 (例)
1	承認結果	否認
2	承認/否認日時	(任意)
3	願書送付要否	送付不要
4	願書送付予定日	(未入力)
5	願書送付担当者	(未入力)

e. [次のステージ] をクリックして、[クローズ] ステージに進むことを確認してください。

その後、[クローズ] ステージで [クローズド] を [はい] に変更して終了し、[保存して閉じる] をクリックしてこのテストを終了してください。

③ 3 回目の訪問 (進路指導の先生と再度面談、推薦募集人数を調整の上申請、承認される)

a. 2 回目と同じ要領で、新規データを作成して [承認] ステージまで進めてください。

[承認] ステージに、入力例を参考に入力してください。

入力例)

No.	項目	入力 (例)
1	承認結果	承認
2	承認/否認日時	(任意)
3	願書送付要否	送付要
4	願書送付予定日	(任意)
5	願書送付担当者	(ご自身のメールアドレス)

入力後 [次のステージ] をクリックして、[願書送付] ステージに進むことを確認してください。



これですべての条件分岐がテストできました。現在のステージには何も入力せず [保存して閉じる] でテストを終了してください。

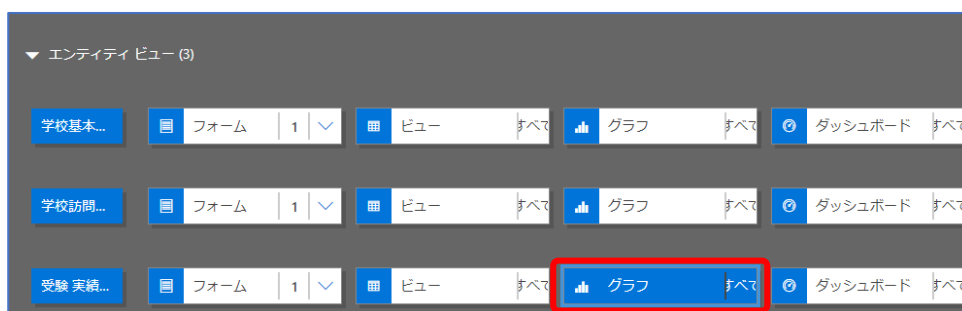
5.3. ダッシュボードの作成

1. ダッシュボードに表示するグラフを作成します。

- ① [学校訪問管理アプリ] をアプリ デザイナーで開きます。アプリの一覧で [学校訪問管理アプリ] の [...] をクリックし、表示されるメニューから [編集] を選択します。



- ② アプリ デザイナーが開いたら [エンティティビュー] にある [受験実績情報] の [グラフ] を選択してください。

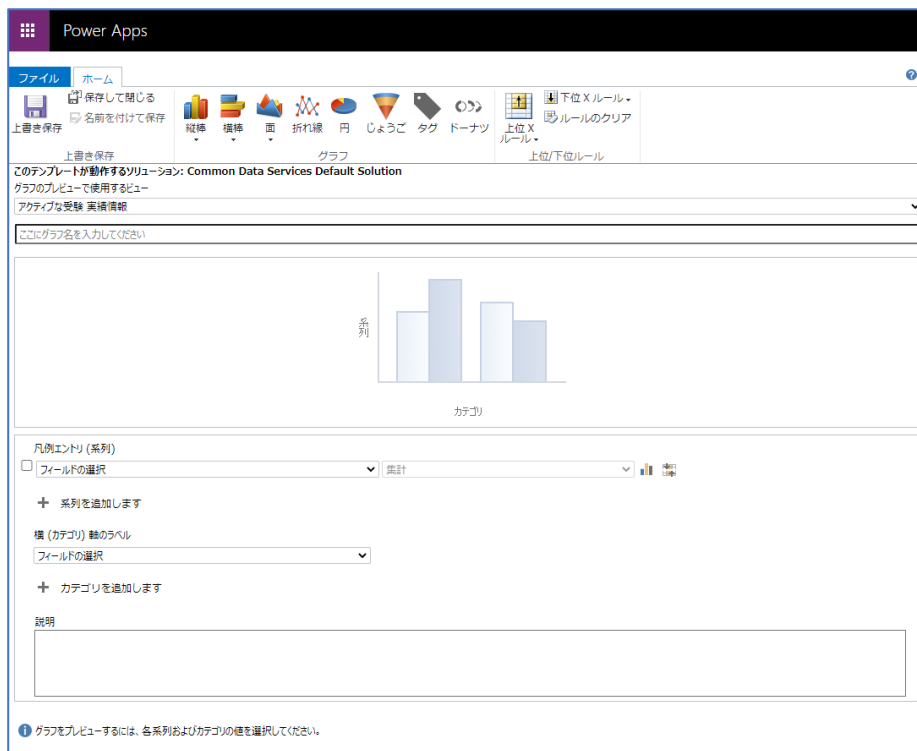


この時点で画面右側のパネルは以下の状態になっています。

ここで [新規作成] をクリックしてください。

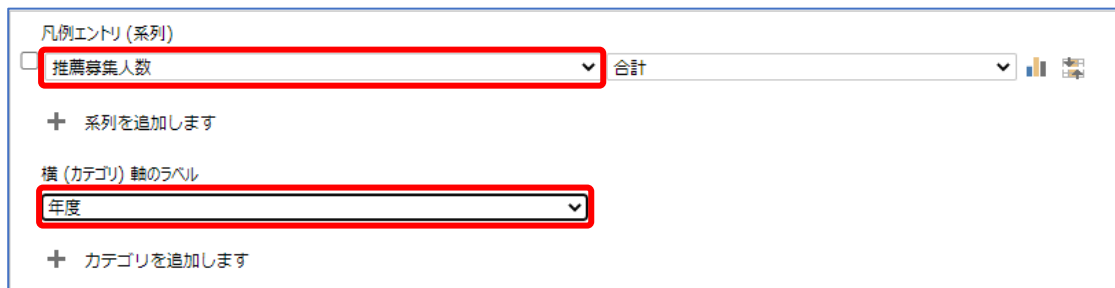


③ 新しいウィンドウでグラフ作成画面が表示されました。



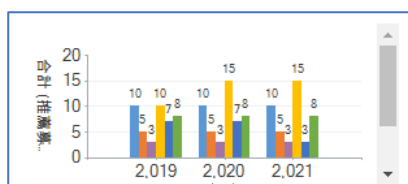
④ 年度ごとに各学校の推薦募集人数を比較できるグラフを作成します。

[凡例エントリ(系列)] に [推薦募集人数]、[横(カテゴリ)軸のラベル] に [年度] を選択してください。



[カテゴリを追加します] をクリックして横軸ラベルの項目を追加し、そこに [学校名(学校)] を選択します。

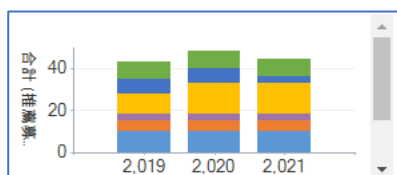
- ⑤ この時点ではグラフのプレビューが図のようになっています。
この形式では学校の数が増えると読みづらくなるので、グラフの形式を変更します。



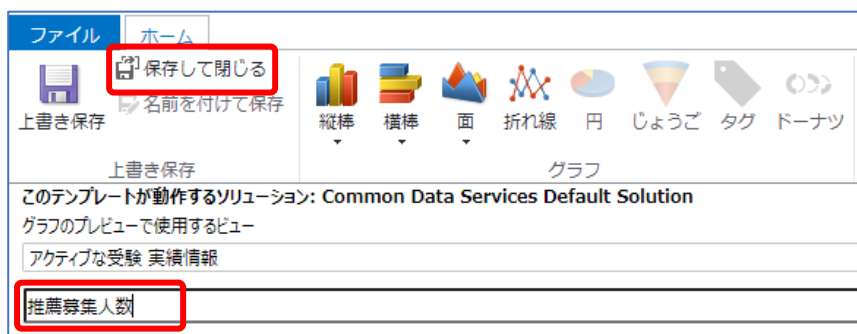
リボンにある、[縦棒] を展開し、[積み上げ縦棒] を選択します。



プレビューを確認します。



- ⑥ グラフに名前を付けます。ここでは [推薦募集人数] とします。
名前の入力ができたら [保存して閉じる] をクリックして終了してください。



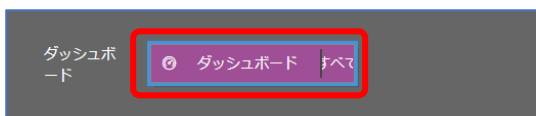
⑦ 同様の手順でグラフをすべて作成してください。作成済みのグラフは背景色をグレーにしています。

それぞれ、各種入学者数の推移を見ることによって訪問を強化すべき学校を発見するためのグラフです。

No.	グラフの名前	凡例エントリ	横軸ラベル 1	横軸ラベル 2	グラフの種類
1	推薦募集人数	推薦募集人数	年度	学校名(学校)	積み上げ縦棒
2	一般入学者数推移	一般入学者数	年度	学校名(学校)	折れ線
3	総合型入学者推移	総合型入学者数	年度	学校名(学校)	折れ線
4	推薦入学者推移	推薦入学者数	年度	学校名(学校)	折れ線

2. ダッシュボードを作成します。

① アプリ デザイナーの [ダッシュボード] を選択します。



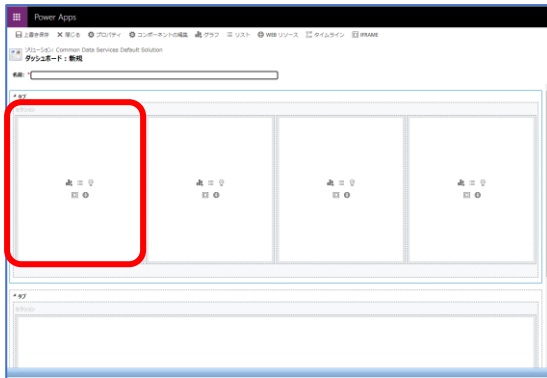
② 右側のパネルで [新規作成] をクリックするとリストが表示されるので、[クラシック ダッシュボード] を選択してください。



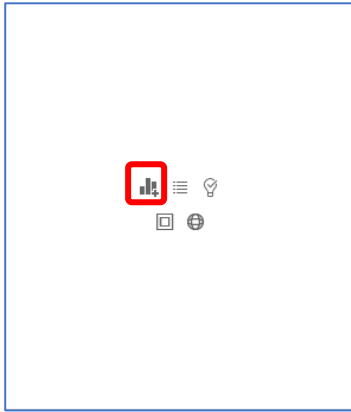
③ 次の画面でレイアウトを選択します。ここでは [4列 概要ダッシュボード] を選択してください。選択後、[作成] をクリックしてください。



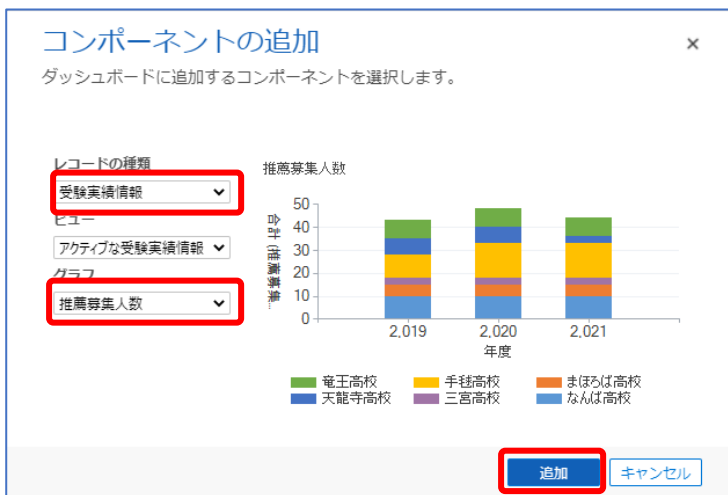
- ④ 新しいウィンドウでダッシュボード作成画面が表示されました。
一つ目のグラフの枠にグラフを割り当てます。



- ⑤ [グラフの挿入] をクリックしてください。



- ⑥ 次の画面が表示されます。

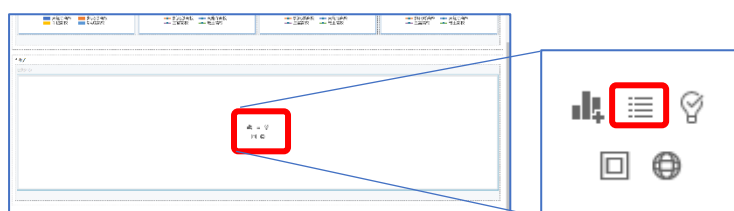


[レコードの種類] に [受験実績情報] を選択すると、先ほど作成したグラフが選択できるようになります。[グラフ] に [推薦募集人数] を選択して [追加] をクリックしてください。

⑦ 一つ目のグラフが追加できました。残りの 3 つを追加してください。



⑧ 次に、ビューを追加します。画面下部のエリアで [リストの挿入] をクリックしてください。



⑨ 次の画面が表示されます。

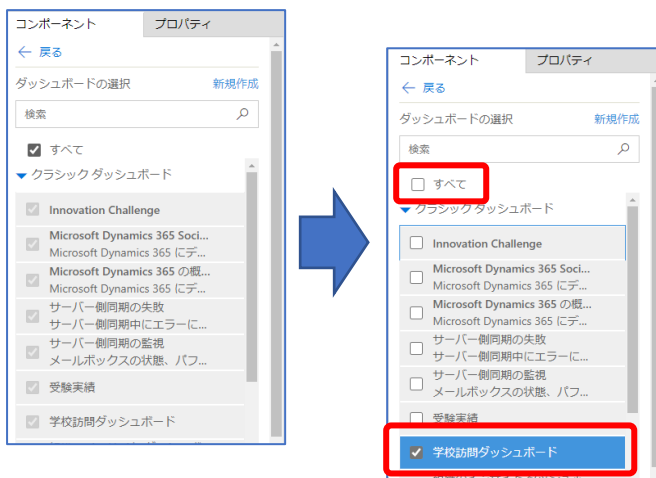
The screenshot shows a dialog box titled 'コンポーネントの追加' (Add Component) with a close button (X). The text says 'ダッシュボードに追加するコンポーネントを選択します。' (Select a component to add to the dashboard). There is a 'レコードの種類' (Record Type) dropdown menu with '学校訪問履歴情報' (School Visit History Information) selected and highlighted with a red box. Below it is a 'ビュー' (View) dropdown menu with 'アクティブな学校訪問履歴!' (Active School Visit History!) selected. At the bottom right, there are two buttons: '追加' (Add) and 'キャンセル' (Cancel), with '追加' highlighted in a red box.

[レコードの種類] に [学校訪問履歴情報] を選択して [追加] をクリックしてください。

- ⑩ ダッシュボードに名前を付けます。ここでは [学校訪問ダッシュボード] とします。
名前を入力したら [上書き保存] ⇒ [閉じる] の順にクリックしてください。



- ⑪ アプリ デザイナーで、[すべて] のチェックを外し、今作成したダッシュボードのみをチェックします。



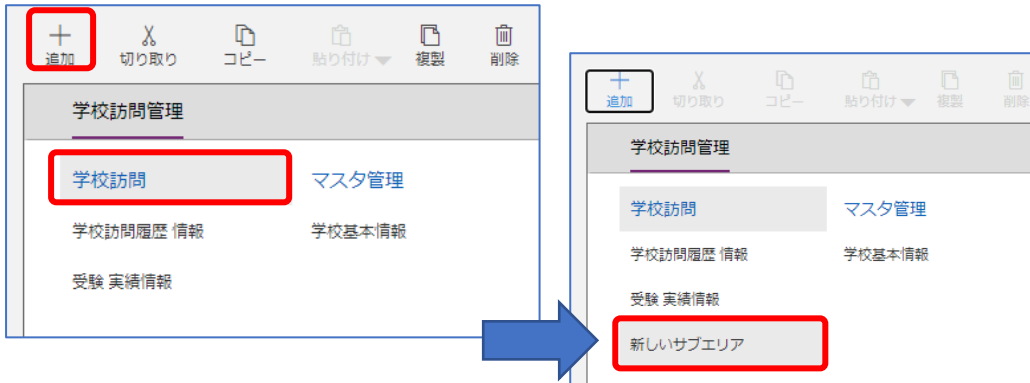
ここまでの作業内容を一旦保存してください。

⑫ サイトマップにダッシュボードを追加します。

アプリ デザイナーの [サイトマップ] の編集アイコンをクリックしてサイトマップ エディターを開きます。



サイトマップ デザイナーが表示されたらグループ [学校訪問] の下に新しいサブエリアを追加します。

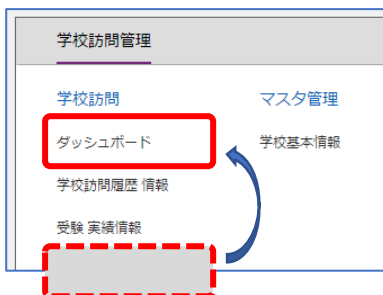


新しいサブエリアのプロパティを以下のように設定します。

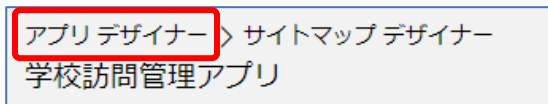
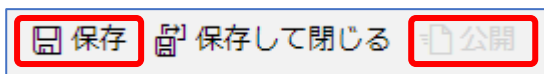
種類	ダッシュボード ①
エンティティ	エンティティの選択
URL *	/workplace/home_dashboards.aspx
既定のダッシュボード	学校訪問ダッシュボード ②

No.	項目	入力
1	種類	ダッシュボード
2	既定のダッシュボード	学校訪問ダッシュボード

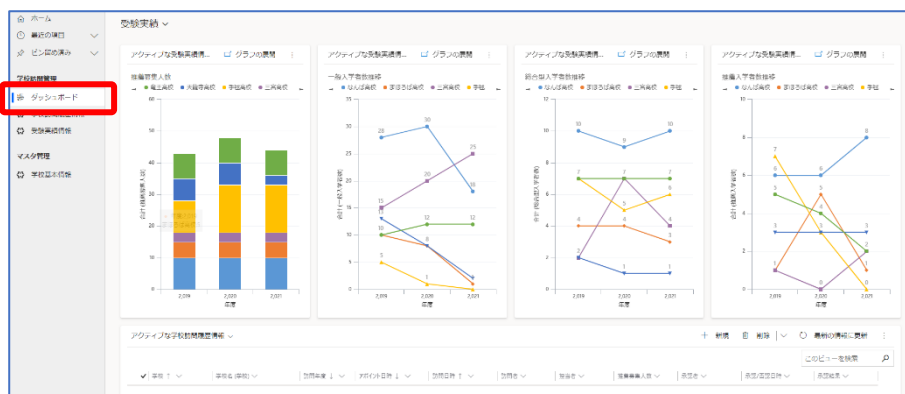
[ダッシュボード] をドラッグして、[学校訪問] グループの最上段に移動してください。



[保存] ⇒ [公開] をクリックした後、 [アプリ デザイナー] をクリックしてサイトマップ デザイナーを終了します。



続けて、アプリ デザイナーで [公開] ⇒ [再生] を順にクリックしてください。



サイトマップにダッシュボードが追加され、定義したグラフが表示されることが確認できました。

以上でモデル駆動型アプリは一応の完成をみることができました。

このままモデル駆動型アプリですべての入力を行うという運用も可能ですが、Power Platform の機能を組み合わせることでより使いやすいアプリとすることを検討しましょう。

承認依頼を自動的にメールで送信し、承認者がメールから直接承認できるようにすることによって使い勝手が飛躍的に向上します。次の演習では Power Automate のクラウドフローを使って、承認依頼の自動化を行います。

6. 演習 4 承認依頼の自動化

6.1. クラウド フローの新規作成

1. クラウド フローを新規作成するには Power Apps のメニューから [フロー] を選択し、[クラウド フロー] の [新しいフロー] を選択します。

※注：この演習は Power Apps を中心に行っているため Power Apps のメニューを使用していますが、当然、Power Automate のメニューからもフローの作成、編集を行うことができます。



[新しいフロー] を選択するとフローの種類がリストで表示されるので、[自動化したクラウドフロー] を選択してください。



2. 次の画面が表示されます。

[フロー名] に今から作成するフローの名前を入力します。今回は [学校訪問承認フロー] としてください。
[フローのトリガーを選択してください] では、フローが開始するきっかけとなるイベントをハンドリングするコンポーネントを選択します。今回は [学校訪問履歴情報] テーブルの更新をきっかけとして開始するフローを作成するので、それに対応したコンポーネントを選択します。

検索テキストボックスに “変更” と入力してください。コンポーネント一覧に Microsoft Dataverse の [行が追加、変更、または削除された場合] が表示されるので、それを選択します。



選択後、[作成] をクリックしてください。

次のような画面に遷移します。



画面の中ほどに、いくつかの入力項目をもつ箱があります。これがクラウド フローの処理を構成するコンポーネントです。クラウド フローではこのようなコンポーネントを組み合わせることで処理を実装していきます。

6.2. トリガーを設定する

1. このフローは、学校訪問履歴情報を更新する業務フローから、申請にかかわる項目が更新されたことをトリガーとします。つまり“申請をする/しない”だけでなく、申請する内容や承認者の指定が変わったとしても改めて承認依頼をする必要があると考えます。

最初に表示されているトリガー [行が追加、変更、または削除された場合] のプロパティは以下のような内容となっています。※ [詳細オプションを表示する] をクリックして展開した状態です。

- ① 種類の変更：このアプリでは学校訪問履歴情報のデータを削除することは想定していませんので、[作成または更新] をトリガーとします。
- ② テーブル名：[学校訪問履歴情報] を選択してください。
- ③ スコープ：Organization を選択してください。
- ④ 列フィルター：承認依頼をすべきかどうかを判定するために使用する項目をコンマ区切りで入力します。ただし、これまでのように日本語で定義した列名は使用できません。テーブルの列を一覧で見たときに [名前] として表示される名前を使用します。

表示名 ↑ ↓	名前 ↓
アポイント担当者	cr4ca_appointer
アポイント日時	cr4ca_appointmentdatetime
クローズド	cr4ca_closed
学校	cr4ca_school
学校名	cr4ca_schoolnameref

指定する項目は、[訪問日時], [申請に進む], [推薦募集人数], [申請者], [申請日時], [承認者] です。


- ⑤ 行フィルター：自動保存で思わぬ状態をトリガーとして動いてしまわないよう、最低限保証したい状態を OData スタイルで表現します。項目名は④と同じく [名前] を使用します。

条件は

[訪問日時] ≠ null and ([クローズド] = false or [申請に進む] = true and [クローズド] = true)
です。

- ⑥ 延期期限：入力しないでください。
- ⑦ 実行するユーザー：トリガーユーザーとしてください。

2. 入力したイメージは図のようになります。

 行が追加、変更、または削除された場合 ...

*種類の変更	作成または更新 ▼
*テーブル名	学校訪問履歴情報 ▼
*スコープ	Organization ▼
列フィルター	cr4ca_visiteddatetime,cr4ca_proceedtoapplication,cr4ca_acceptingrecommend ees,cr4ca_requester,cr4ca_appliedat,cr4ca_approver
行フィルター	cr4ca_visiteddatetime ne null and (cr4ca_closed eq false or cr4ca_proceedtoapplication eq true and cr4ca_closed eq true)
延期期限	トリガーの評価を遅延させる時間を入力します (例: 2020-01-01T10:10:00Z)
実行するユーザー	トリガー ユーザー ▼

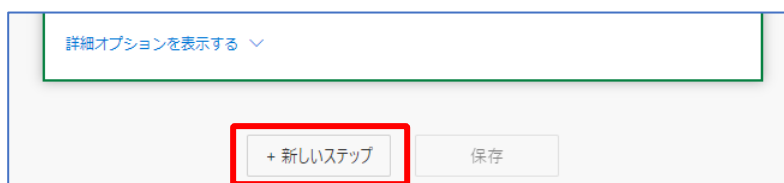
[詳細オプションを表示しない](#) ^

6.3. 最初の条件を作成する

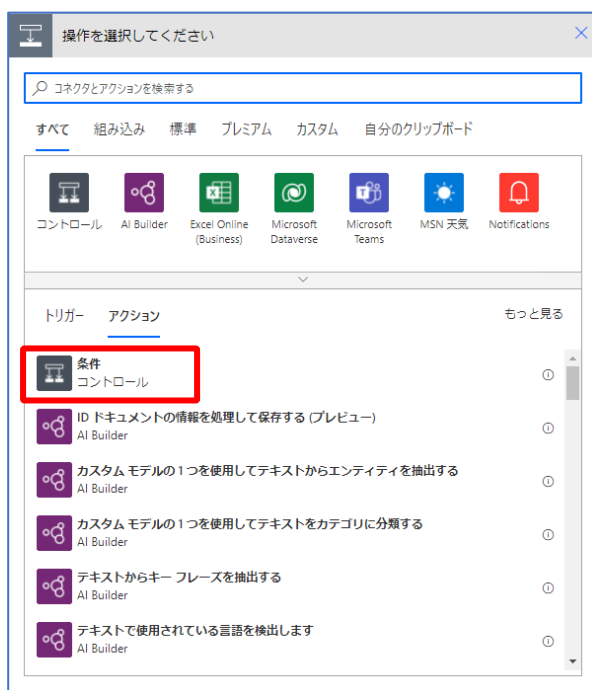
1. 入力されたトリガーが申請に進もうとしているのかどうかを判定する条件を定義します。

① まず、条件のコンポーネントを追加します。

トリガーコンポーネントの下にある [新しいステップ] をクリックしてください。



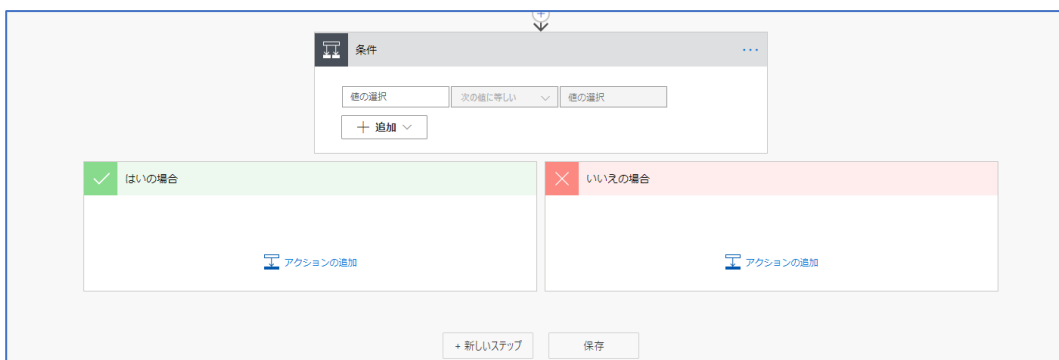
次のような画面が現れます。



[条件] を選択してください。

※注： 見つけにくい場合は検索テキストボックスに "条件" と入力してみてください。

図のようなコンポーネントが追加されます。



② 条件を記入します。

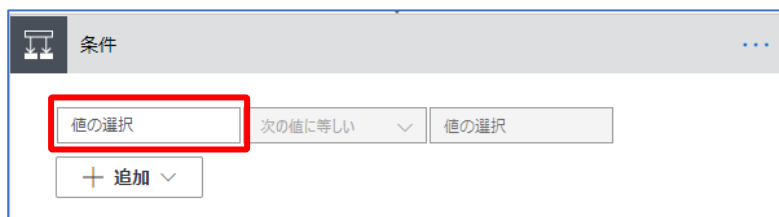
ここでの条件は、

“申請に進もうとしている” and “承認者が指定されている” and
“申請者が指定されている” and “推薦募集人数が指定されている”

という内容を指定します。

合致する場合は承認依頼を行います、合致しない場合はこの訪問実績をクローズします。

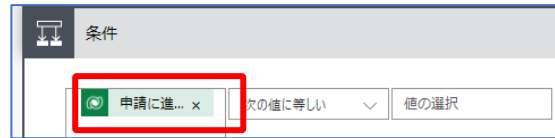
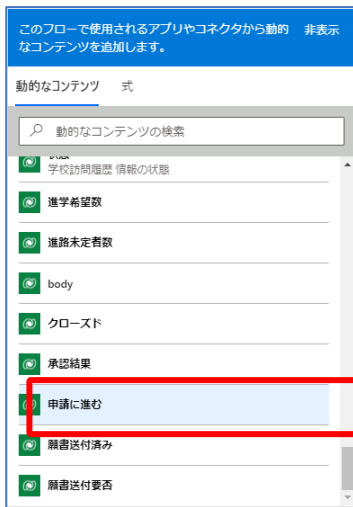
左辺の [値の選択] をクリックしてください。



動的なコンテンツの画面がポップアップします。



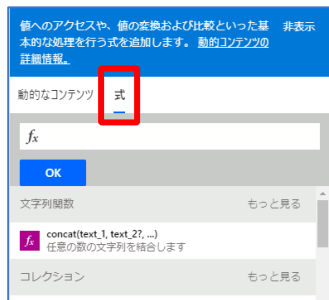
この中からまず、[申請に進む] を選択してください。



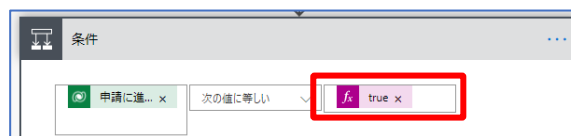
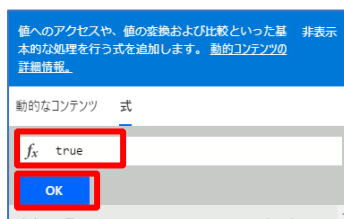
条件の左辺に設定されました。

次に、右側の [値の選択] をクリックします。

ここでは [true] という値を指定したいので、[動的なコンテンツ] の隣にある [式] をクリックしてください。

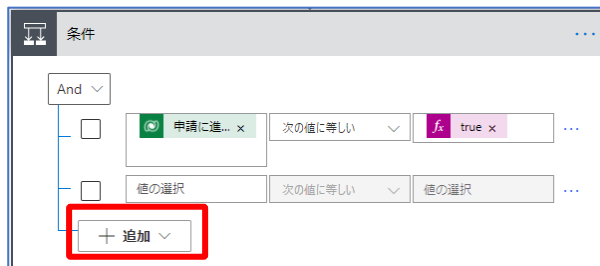


[式] に true と入力し、[OK] をクリックしてください。



右辺に [true] が設定されました。

[追加] をクリックすることで条件の行を追加することができます。



③ 残りの条件を設定してください。

No.	左辺の値	演算子	右辺の値
1	申請に進む	次の値に等しい	true
2	承認者	次の値に等しくない	null
3	申請者	次の値に等しくない	null
4	推薦募集人数	次の値より大きい	0

完成形は下図のようになります。

The screenshot shows a configuration window titled '条件' (Conditions). At the top left, there is a tree view icon and the title '条件'. Below the title, a dropdown menu is set to 'And'. There are four conditions listed, each with a checkbox on the left, a field for the left value, a dropdown for the operator, and a field for the right value. The conditions are:

- 1. 申請に進... x 次の値に等しい true x ...
- 2. 承認者 x 次の値に等しくない null x ...
- 3. 申請者 x 次の値に等しくない null x ...
- 4. 推薦募集... x 次の値より大きい 0 x ...

At the bottom, there is a '+ 追加' (Add) button.

6.4. 申請しない場合の処理を作成する

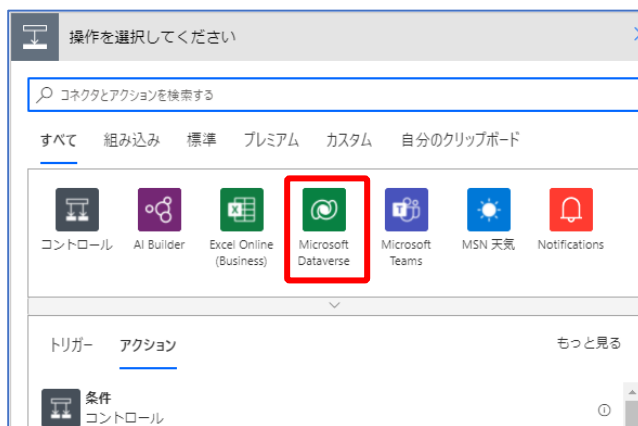
1. トリガーとなった更新で“申請に進まない”(=クローズ) 場合の処理を作成します。

この処理では学校訪問ステージで申請に進まない選択をした訪問履歴データをクローズ状態に更新します。

- ① [いいえの場合] 中にある [アクションの追加] をクリックしてください。



[操作を選択してください] の中で、Microsoft Dataverse を選択してください。

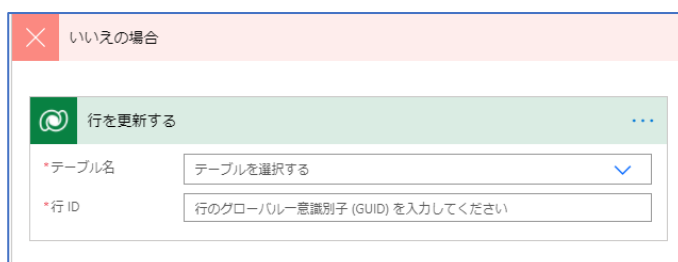


さらに、検索テキストボックスに“行を更新”と入力します。

表示された [行を更新する] を選択してください。

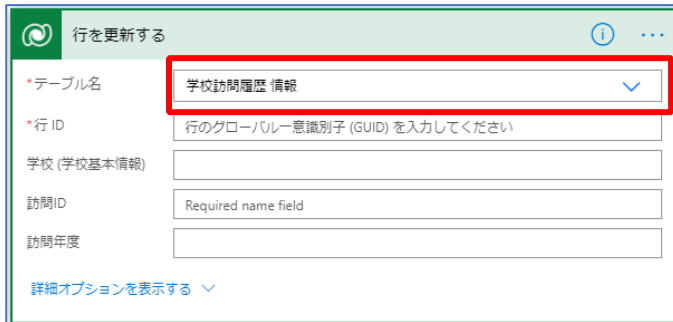


[行を更新する] のコンポーネントが表示されます。



② [行を更新する] のプロパティを設定します。

まず、テーブル名に [学校訪問履歴情報] を選択してください。



行を更新する

*テーブル名 学校訪問履歴 情報

*行 ID 行のグローバル一意識別子 (GUID) を入力してください

学校 (学校基本情報)

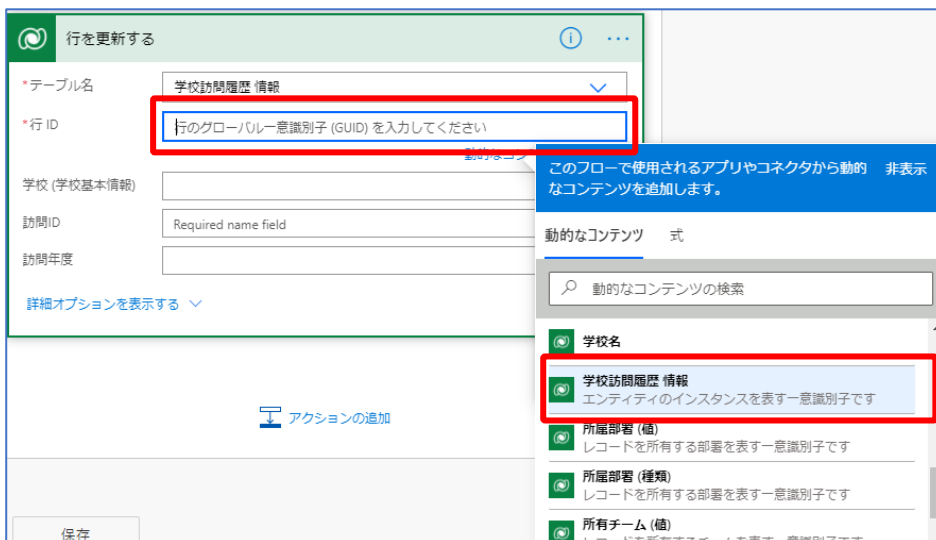
訪問ID Required name field

訪問年度

詳細オプションを表示する

テーブル名を指定すると詳細が入力可能になりました。

続いて [行 ID] をクリックして動的なコンテンツを選びます。ここでは [学校訪問履歴情報] という列を選択してください。これは自動的に定義される列で、このデータをユニークに識別するものです。



行を更新する

*テーブル名 学校訪問履歴 情報

*行 ID 行のグローバル一意識別子 (GUID) を入力してください

学校 (学校基本情報)

訪問ID Required name field

訪問年度

詳細オプションを表示する

このフローで使用されるアプリやコネクタから動的 非表示
なコンテンツを追加します。

動的なコンテンツ 式

動的なコンテンツの検索

- 学校名
- 学校訪問履歴 情報
エンティティのインスタンスを表す一意識別子です
- 所属部署 (値)
レコードを所有する部署を表す一意識別子です
- 所属部署 (種類)
レコードを所有する部署を表す一意識別子です
- 所有チーム (値)
レコードを所有するチームを表す一意識別子です

アクションの追加

保存

③ 更新する列とその内容を設定します。

[詳細オプションを表示する] をクリックしてすべての列を表示してください。

下表のとおり、データを設定してください。

※注：上の画像ではすべての列名が表示されていません。スクロールして列名を探してください。

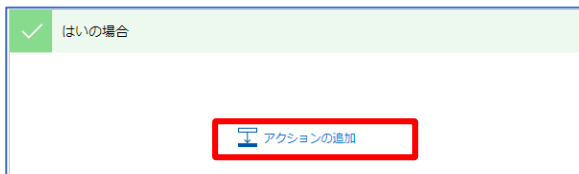
No.	列名	値
1	クローズド	はい
2	承認/否認日時	null
3	承認結果	null
4	願書送付予定日	null
5	願書送付担当者	null
6	願書送付要否	null

※注：[承認結果] や [願書送付要否] のような選択型項目に対して [null] を直接入力することはできません。選択肢の下に表示される [カスタム値] を選択して入力してください。

6.5. 承認依頼処理を作成する

1. トリガーとなった更新で“申請に進む”場合の処理を作成します。
この処理は承認依頼を送信して、承認者の回答を待つようにします。

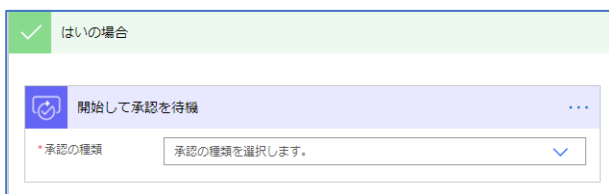
- ① [はいの場合] 中にある [アクションの追加] をクリックしてください。



[操作を選択してください] の中で検索テキストボックスに "開始して承認を待機" と入力し、表示された [開始して承認を待機] を選択してください。



アクションが追加されました。



② 承認アクションのプロパティを設定します。

承認の種類に [承認/拒否・最初に応答] を選択してください。

これにより設定するプロパティの種類が決定され、入力できる状態となります。

*承認の種類	承認/拒否・最初に応答
*タイトル	承認のタイトルを作成します
*担当者	セミコロン (;) で区切られたメール アドレス
詳細	サポートされているマークダウン (https://aka.ms/approvaldetails を参照)
アイテム リンク	承認するアイテムにリンクを追加します
アイテム リンクの説明	アイテムへのリンクについて説明します

詳細オプションを表示する ▾

タイトルに "推薦募集人数承認依頼:" と入力してください。

*承認の種類	承認/拒否・最初に応答
*タイトル	推薦募集人数承認依頼:
*担当者	セミコロン (;) で区切られたメール アドレス
詳細	サポートされているマークダウン (https://aka.ms/approvaldetails を参照)
アイテム リンク	承認するアイテムにリンクを追加します
アイテム リンクの説明	アイテムへのリンクについて説明します

詳細オプションを表示する ▾

カーソルはそのまま、[動的なコンテンツの追加] で [学校名] を選択して追記します。

このフローで使用されるアプリやコネクタから動的 非表示
なコンテンツを追加します。

動的なコンテンツ 式

動的なコンテンツの検索

- 受付担当者
- 学校名

入力結果は図のようになります。

*タイトル	推薦募集人数承認依頼: 学校名
-------	-----------------

動的なコンテンツの追加 +

上記の要領で残りのプロパティを図のように入力してください。

開始して承認を待機

*承認の種類: 承認/拒否 - 最初に応答

*タイトル: 推薦募集人数承認依頼: 学校名 x

*担当者: 承認者 x ;

詳細

学校名 x の推薦募集人数について承認をお願いいたします。

推薦募集人数: 推薦募集人数 x

面談の結果

- 卒業見込: 卒業見込生徒数 x
- 進学希望: 進学希望数 x
- 当学志望: 当学志望者数 x
- 所感等: 訪問概要 x

アイテムリンク: 承認するアイテムにリンクを追加します

アイテムリンクの説明: アイテムへのリンクについて説明します

詳細オプションを表示する

さらに、[詳細オプションを表示する] をクリックし、[要求元] を設定します。

・当学志望: 当学志望者数 x

・所感等: 訪問概要 x

アイテムリンク: 承認するアイテムにリンクを追加します

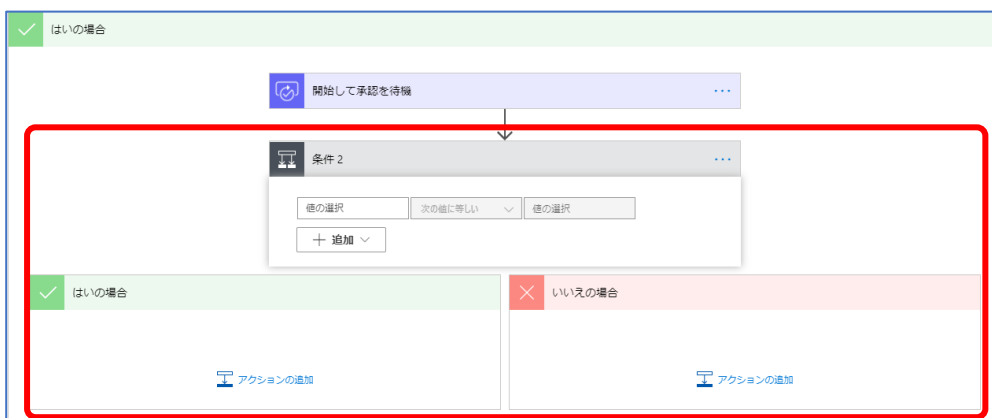
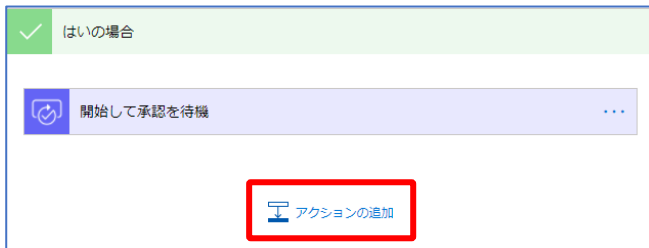
アイテムリンクの説明: アイテムへのリンクについて説明します

要求元: 申請者 x ;

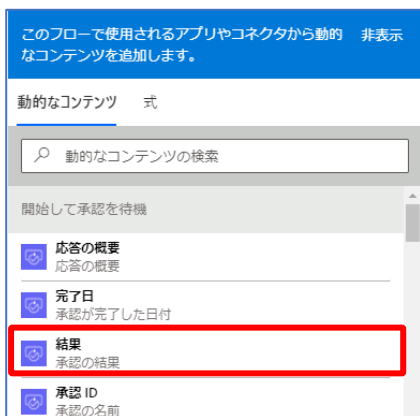
動的なコンテンツの追加 +

6.6. 承認結果の条件を作成する

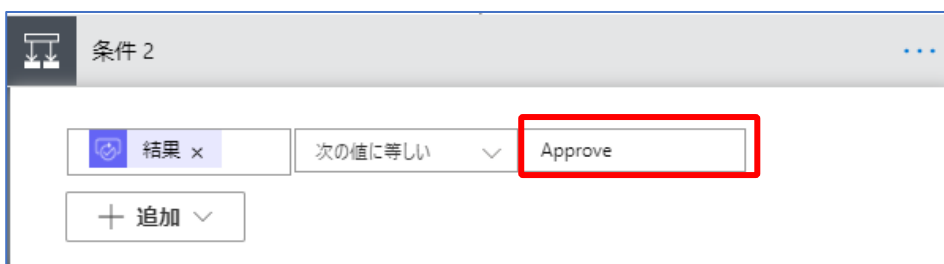
- 承認依頼の結果、承認された場合と否認された場合で処理を振り分けるための条件を作成します。
 - [開始して承認を待機] の下の [アクションの追加] をクリックして [条件] を追加してください。



- 条件式の左側の [値の選択] に [結果] を入力します。



- 演算子は [次の値に等しい] のままとし、右側の [値の選択] に "Approve" と入力してください。

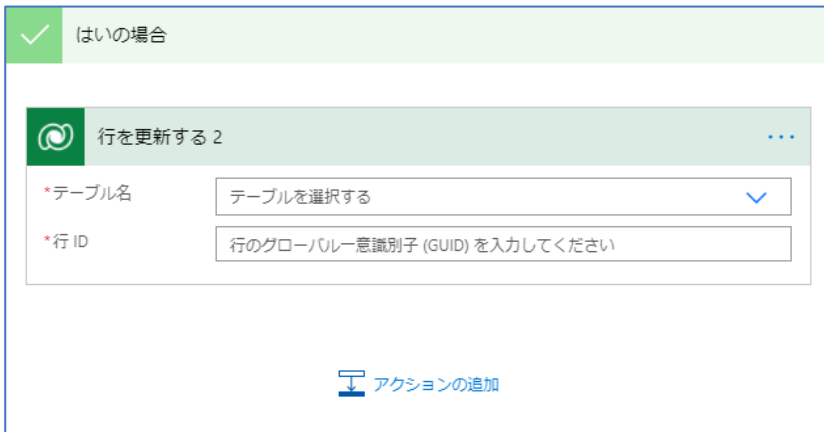


6.7. 承認後の処理を作成する

- 承認結果を受けての条件分岐を作成したので、承認時と否認時それぞれの処理を作成します。
承認時、否認時とも、前の手順で使用した [行を更新する] を使用します。

① 承認時の処理を作成します。

承認結果の条件分岐で、[はいの場合] に [行を更新する] アクションを追加してください。



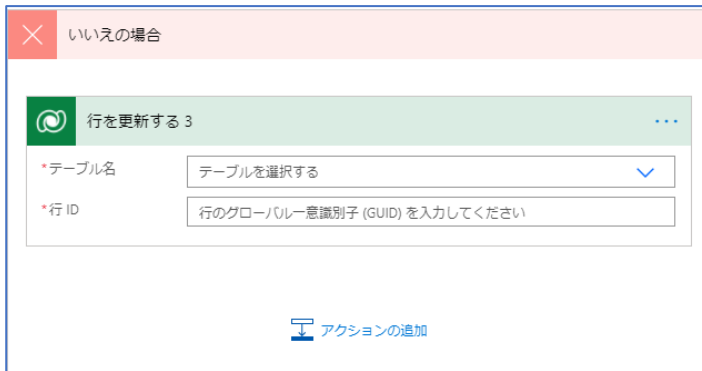
前の手順と同様に、[テーブル名] と [行 ID] を設定してください。

その後、下表に従って更新内容を設定してください。

No.	列名	値
1	クローズド	いいえ
2	承認/否認日時	完了日
3	承認結果	はい
4	願書送付予定日	@{addDays(outputs('開始して承認を待機')?['body/completionDate'], 30)} ※ "完了日の 30 日後" の指定です。
5	願書送付担当者	申請者
6	願書送付済み	いいえ
7	願書送付要否	はい

② 否認時の処理を作成します。

承認結果の条件分岐で、[いいえの場合] に [行を更新する] アクションを追加してください。

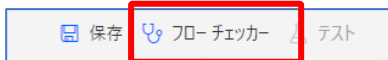


前の手順と同様に、[テーブル名] と [行 ID] を設定してください。

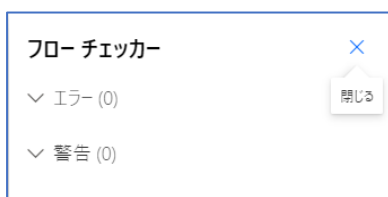
その後、下表に従って更新内容を設定してください。

No.	列名	値
1	クローズド	はい
2	承認/否認日時	完了日
3	承認結果	いいえ
4	願書送付予定日	null
5	願書送付担当者	null
7	願書送付要否	いいえ

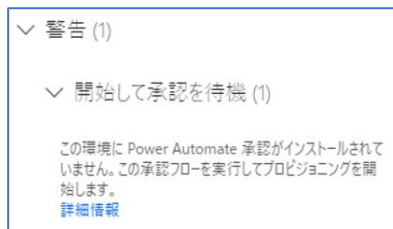
2. すべての入力が完了したら [フローチェッカー] をクリックしてエラーがないことを確認してください。



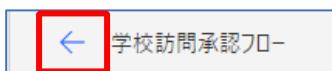
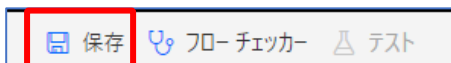
問題なければ [X] でフローチェッカーを閉じてください。



※注：現在の環境で初めて [承認] コネクタを使用する場合、下図のような表示となります。この警告が出ることは正常なので、そのまま次の手順に進んでください。



3. これでフローは完成です。[保存] をクリック後、[←] をクリックしてフロー一覧に戻ってください。



6.8. フローをテストする

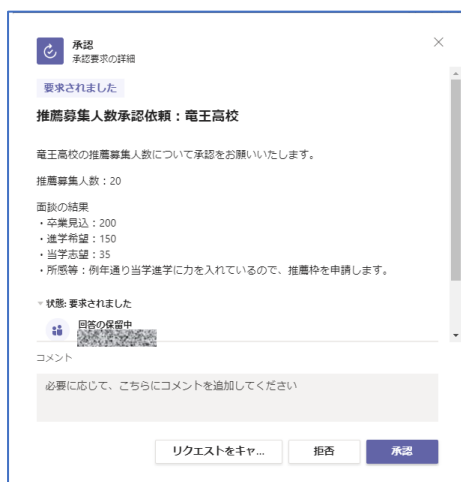
作成したフローをテストしましょう。

モデル駆動型アプリを起動して、学校訪問履歴情報を新規作成し、[学校訪問] ステージで [承認に進む] に [はい] と入力し、[推薦募集人数申請] ステージで [推薦募集人数] に 1 以上の数字を入力してください。

このステージで注意すべき点は、申請者、承認者ともご自身のメールアドレスを入力することです。誤って他人にテスト用の承認依頼や承認結果が通知されないよう気を付けてください。

正常に動作している場合、以下のような通知がなされます。

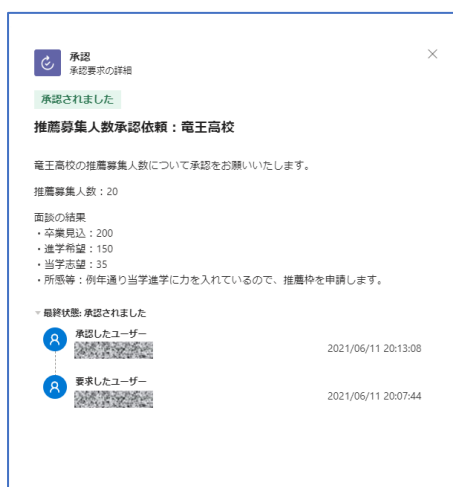
サンプル画像は Microsoft Teams のものです。



この通知に対して [承認] します。上の承認依頼にある [承認] をクリックしてください。

少し待つと以下のような承認結果が通知されます。

サンプル画像は Microsoft Teams のものです。



7. 演習 5（オプション 1）キャンバス アプリの作成

モデル駆動型アプリはノーコーディングでアプリを開発できるという便利さがありますが、その反面入力画面のレイアウトや入力内容のチェックなどは自由に設定できないという制約があります。

キャンバス アプリは、モデル駆動型アプリと比べると作り込みが必要ではありますが、入力画面をレイアウトしたり、動的に画面を切り替えたりといった機能を簡単に開発することができます。

また、キャンバス アプリは携帯電話やタブレット端末の画面サイズに合わせて作成することができるため、オフィス外でデータを参照したり入力したりする業務に適しています。

この演習では、これまでの演習で作成したモデル駆動型アプリの、[学校訪問] と [推薦募集人数] ステージのデータ入力をするためのキャンバス アプリを作成します。

※注：この演習はオプションですので実施は必須ではありません。実施する場合は [4.3. ビューを作成する] のオプション【学校訪問履歴情報】テーブルの [学校訪問アポイント情報] 及び [学校訪問実績情報] のビューを作成していることが前提です。これらのビューを作成していない場合はまずこれらを作成してから演習に進んでください。

7.1. 事前準備

このアプリは、学校訪問した先で面談の内容などを入力することを主な目的とします。

つまりモデル駆動型アプリの作成時に定義した業務フローでは以下のステージの入力をするのが主な機能となります。



キャンバス アプリでこの機能を作成するメリットは、面談内容の入力時に、推薦募集人数の申請をするかどうかを入力することにより申請に関する項目の入力チェックをするかどうかを自動的に切り替えたり、[クローズド] のような運用項目を自動的に更新したりできることです。これは業務フローにおける以下のルートにおいても画面から一括して対応できることを意味します。

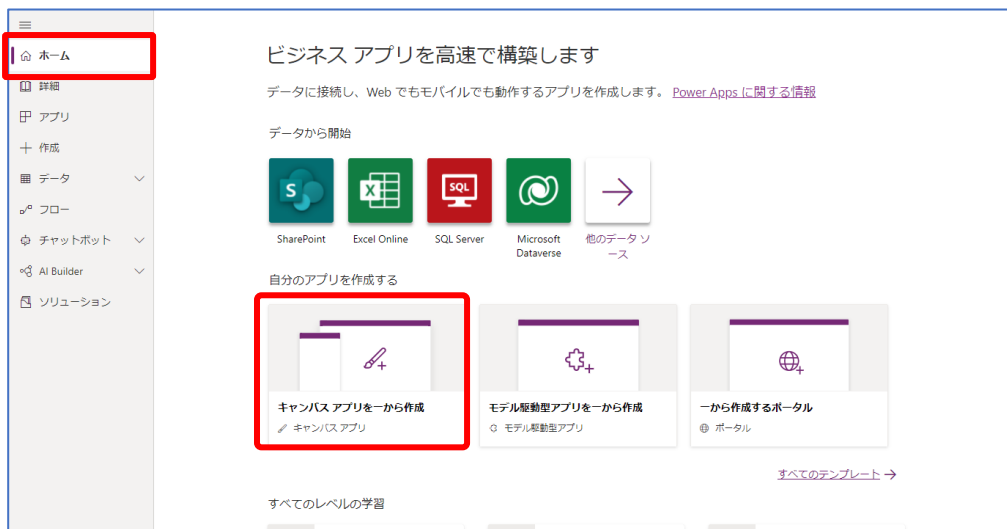


いずれのケースにしても、このアプリで更新する対象は [アポイント] ステージまで完了しているデータです。

ここまでの演習で登録したデータはすべて [学校訪問] ステージを完了してしまっており、対象となるデータが存在しない状態です。アプリ作成は対象データで動作を確認したりできる方が進めやすいので、事前準備として新規の訪問データを登録し、[アポイント] ステージまで完了させておいてください。

7.2. キャンバス アプリの新規作成

1. キャンバス アプリを新規作成するには [ホーム] から [キャンバス アプリを一から作成] をクリックします。

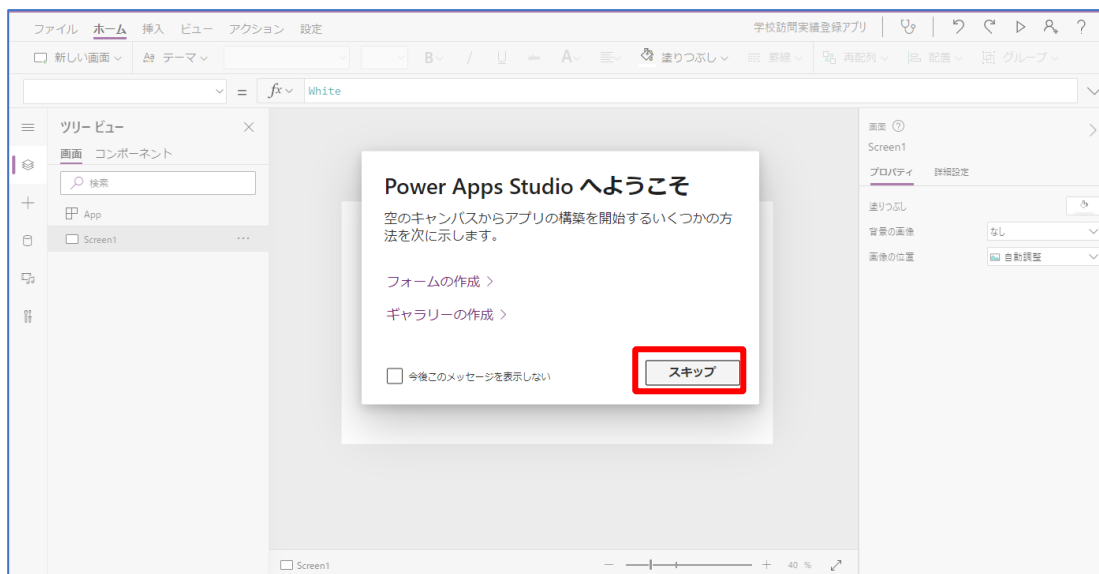


以下の画面が表示されますので、[アプリ名] に “学校訪問実績登録アプリ” と入力し、[形式] は [タブレット] を選択して、[作成] をクリックしてください。

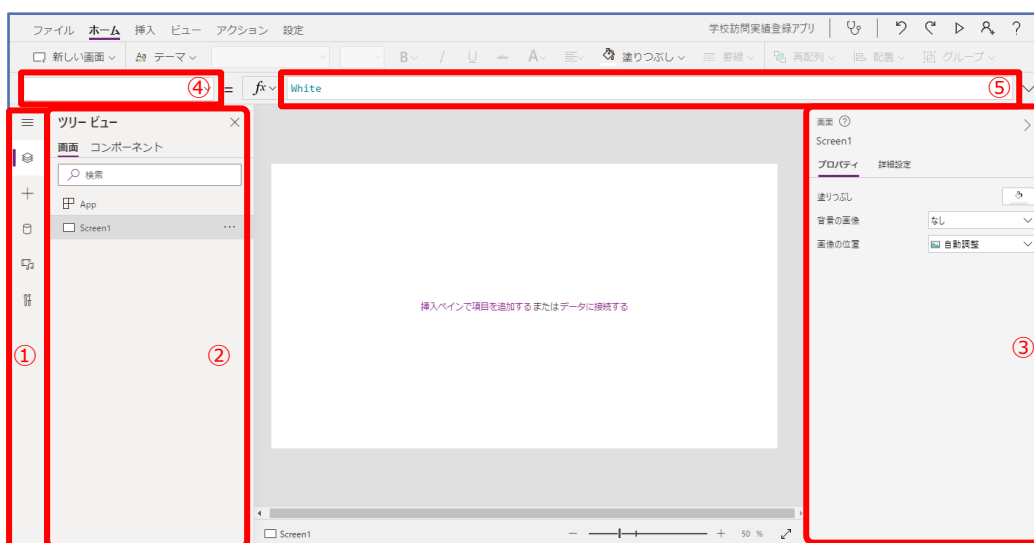


ブラウザの新しいタブで Power Apps Studio が起動します。

[Power Apps Studio へようこそ] のメッセージは [スキップ] をクリックして閉じてください。

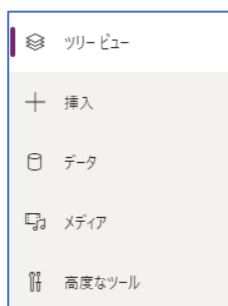


2. Power Apps Studio の画面構成を簡単に整理しておきましょう。



① 画面左端のサイドバーでは②のペインに表示する機能を選択します。

上部の [≡] をクリックすると機能の名前の表示・非表示を切り替えることができます。



② このペインには①のサイドバーで選択した機能が表示されます。

ここではこの演習で使用するツリービュー、挿入、データの 3 機能について説明します。

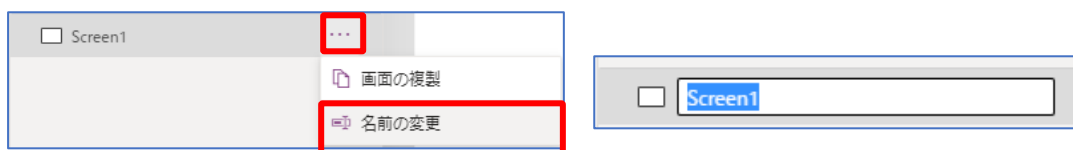
a. [ツリービュー] には現在のアプリ内の画面、画面上のコントロールなどが階層表示されます。

ここに表示された名前をクリックすることによりそのオブジェクトを選択することができます。

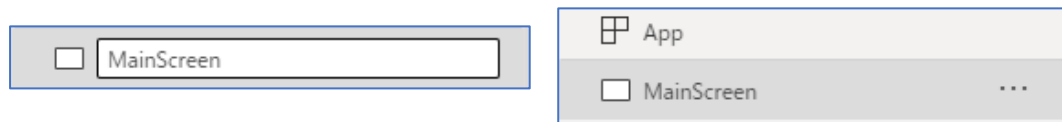
また、オブジェクトの横にある [...] をクリックして、そのオブジェクトに対する操作のメニューを表示することができます。

操作の例として、現在表示されている画面 [Screen1] の名前を変更してみましょう。

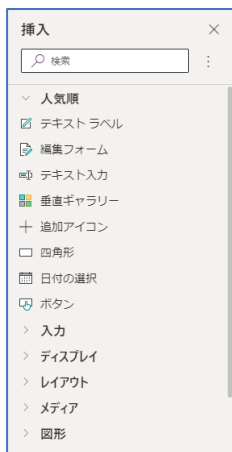
[Screen1] の横にある [...] をクリックしてメニューを表示し、[名前の変更] をクリックすると、名前が入力可能な状態になります。



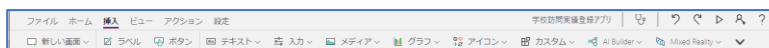
この画面の名前を "MainScreen" とします。入力完了後は Enter で確定してください。



b. [挿入] では画面上にコントロールを追加することができます。

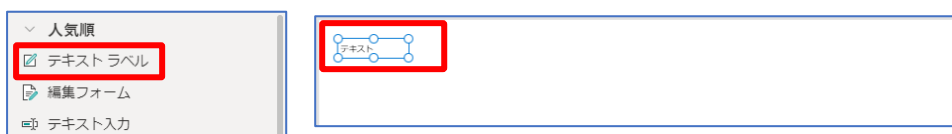


※注：メニューバーから [挿入] を選択して表示されるリボンからもコントロールを挿入可能です。



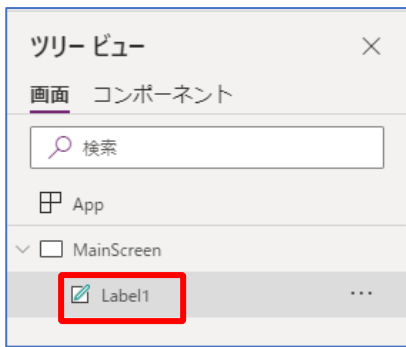
例として [テキストラベル] でタイトルバーを挿入してみましょう。

リストから [テキストラベル] を選択してください。



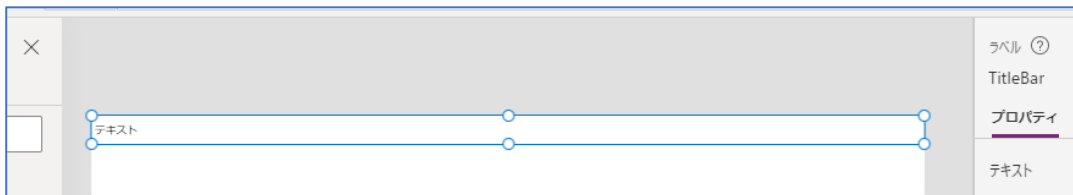
MainScreen 上に [テキスト] と表示されました。

ツリービューを表示すると、Label1 という名前で追加されたことがわかります。

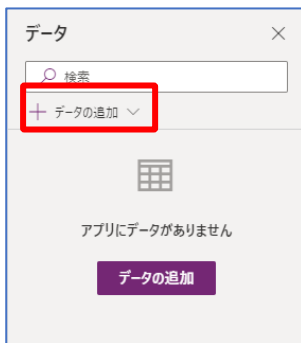


この場で Label1 の名前を "TitleBar" と変更しておいてください。

次に、画面中央部のデザインペインで [TitleBar] を選択し、[MainScreen] の上部に幅いっぱいとなるよう、ドラッグしてサイズ変更してください。



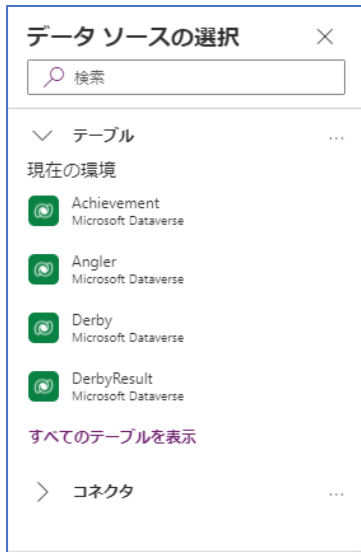
c. [データ] ではこのアプリで使用するデータソースを管理します。



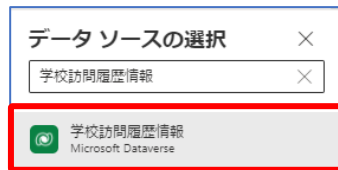
[データの追加] でテーブルや各種コネクタを追加することができます。

このアプリでは主に [学校訪問履歴情報] テーブルを使用するので、このテーブルを追加しましょう。

[データの追加] をクリックすると次のパネルが表示されます。

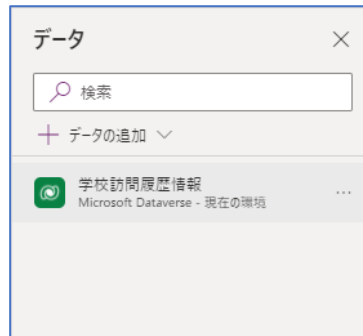


[検索] に “学校訪問履歴情報” と入力して一覧を絞り込みます。



表示された [学校訪問履歴情報] を選択してください。

テーブルが追加され、アプリから参照できるようになりました。



- ③ プロパティペインでは現在選択中のオブジェクトの各種プロパティを設定します。

[プロパティ] と [詳細設定] のタブがあり、[詳細設定] では [プロパティ] タブよりも詳細なプロパティを設定することができます。

上で挿入した TitleBar のテキスト、文字色、背景色、テキストのアラインメントを設定してみましょう。

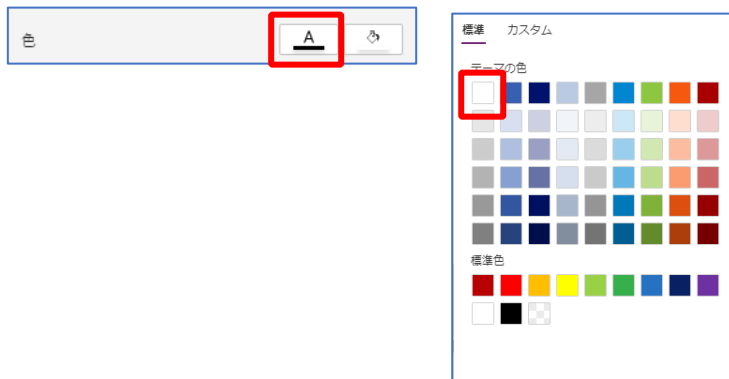
- a. [テキスト] に “学校訪問” と入力してください。



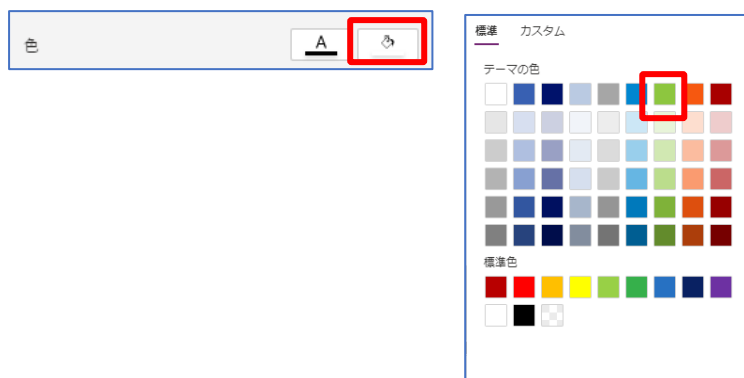
- b. [テキストのアラインメント] で [左右中央揃え] を選択してください。



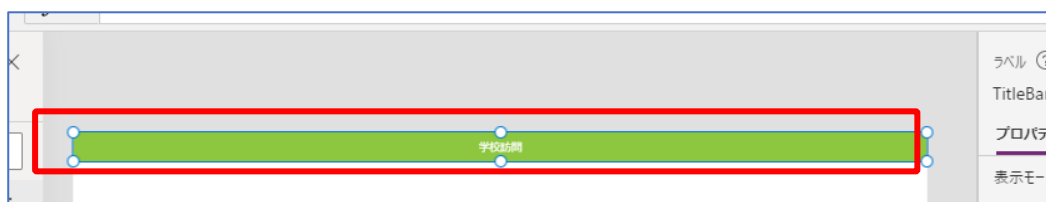
- c. [色] のプロパティで [色] を選択し、カラーパレットから白を選択してください。



d. 同様に [塗りつぶし] を選択して、カラーパレットから緑を選択してください。



TitleBar の見た目はこのように変わっています。



④ プロパティ名のコンボボックスでは、現在選択中のオブジェクトのプロパティ名を選択します。

ここで選択したプロパティに対して、④の数式バーでプロパティ値を設定します。

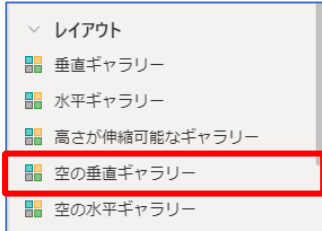
⑤ 数式バーではプロパティ名コンボボックスで選択されたプロパティ値を編集します。

キャンバス アプリでは様々な関数を組み合わせてプロパティに指定することにより、Excel のような作法で画面の動作を定義します。数式バーを使用することにより、関数に指定する引数やオブジェクトのプロパティに関する入力ヘルプを参照しながら設定することが可能です。

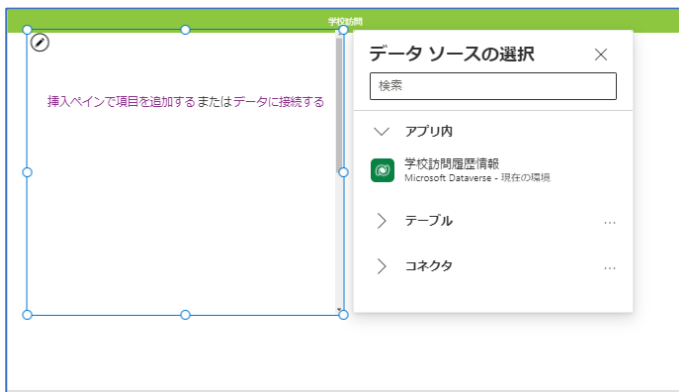
7.3. 照会画面を作成する

1. [MainScreen] にギャラリーを配置してみましょう。

① [挿入] ペインで [レイアウト] を展開し、[空の垂直ギャラリー] を選択してください。



ギャラリーが追加され、接続するデータ ソースの選択パネルが表示されます。



[データ ソースの選択] には先ほど追加した [学校訪問履歴情報] が表示されています。

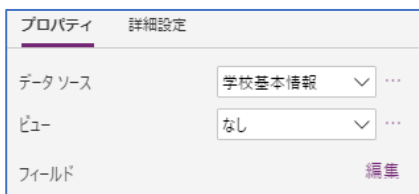
ここではこのテーブルは使用せず、新たに [学校基本情報] をデータ ソースとして選択します。

先ほどの手順同様、[検索] に "学校基本情報" と入力してください。



リストが絞り込まれて [学校基本情報] が表示されるので、これを選択します。

選択後、プロパティペインの [データ ソース] を確認してください。



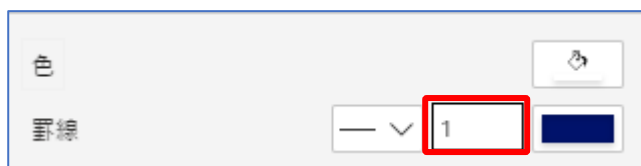
[学校基本情報] が [データ ソース] プロパティに設定されています。

※注：このプロパティを指定してギャラリーのデータ ソースを設定することもできます。

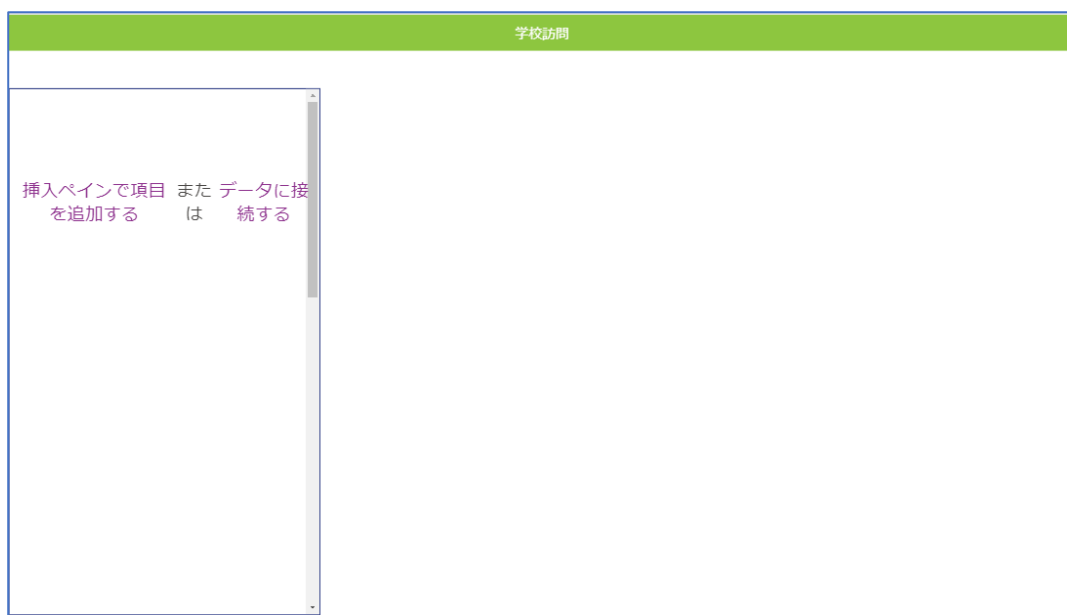
このギャラリーの名前を "SchoolListGallery" としてください。

次に画面内の位置とサイズを調整するのですが、その前にギャラリーの外枠がわかりやすいように罫線を設定しておきましょう。罫線は後で非表示にしてもかまいません。

[罫線] の太さに 1 と入力してください。※罫線を削除する際は 0 に戻してください。



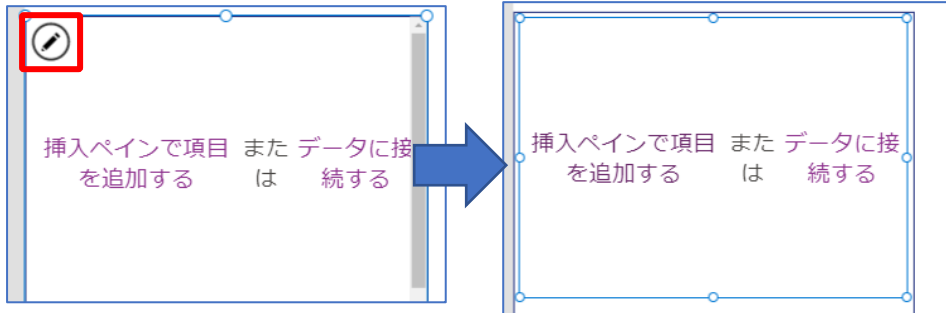
ギャラリーを [MainScreen] の左側に寄せて図のように配置してください。



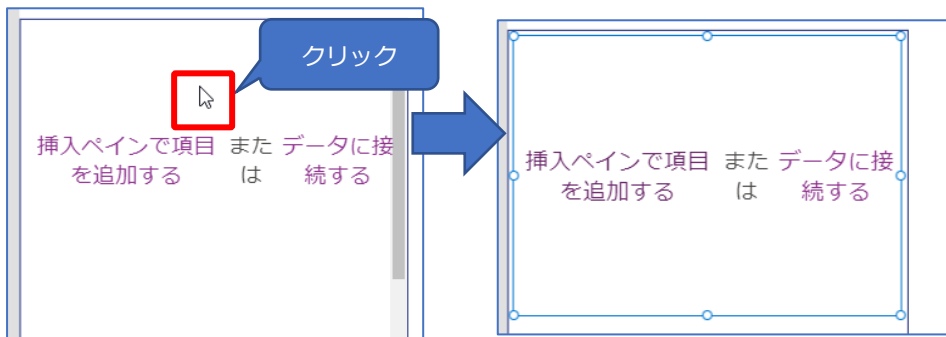
ギャラリーの上に少し余白を開けておいてください。後でこの場所にフィルター用のテキスト入力コントロールを配置します。

② ギャラリーに表示する項目を追加します。

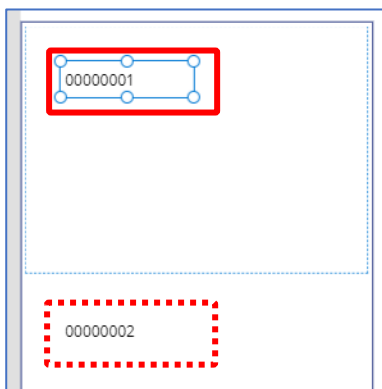
ギャラリーを編集するにはまずギャラリーのテンプレートをアクティブにする必要があります。
テンプレートをアクティブにする方法はギャラリー左上に表示される編集アイコンをクリックするか、



1 つめのデータが表示される (と予測される) 場所を直接クリックする方法のどちらも可能です。



テンプレートがアクティブになったらその状態のまま、[挿入] ペインからコントロールを追加します。
テキストラベルを 1 個追加してください。図のような状態になります。



表示されている値が何に由来しているのかは、プロパティペインの [詳細設定] タブで [Text] プロパティを確認するか、画面上部のプロパティ名プルダウンで [Text] を選択して数式バーの値を確認します。

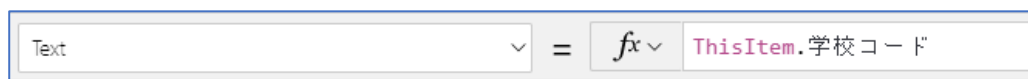


[ThisItem] は、ギャラリーに表示されるデータ (レコード) を表しています。

現在の値は、[学校コード] が表示されていることがわかりました。

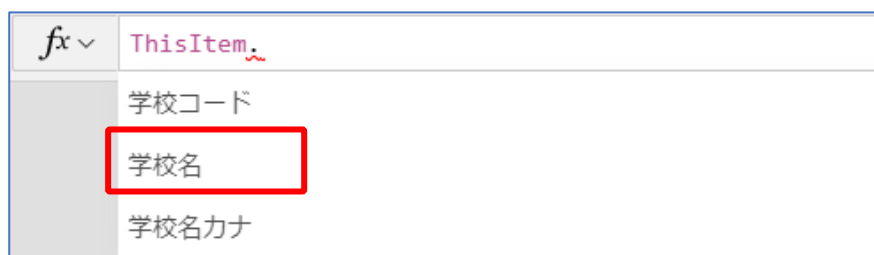
このテキストラベルに [学校名] を表示させるように変更します。

プロパティ名に [Text] を選択して、数式バーで編集を開始します。

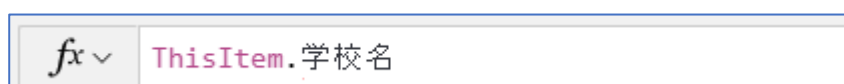


“学校コード” の部分を削除すると、[ThisItem.] に続けることができる候補がリスト表示されます。

この中から [学校名] を選択してください。



選択後の状態です。



これによりギャラリー内のテキストラベルは以下のような表示となります。



ツリービューペインでこのテキストラベルの名前を “SchoolName” としてください。

③ テキストラベルに書式を設定します。

ここでは上で挿入した [SchoolName] テキストラベルのフォント サイズとフォントの太さを設定します。プロパティペインで以下のように入力してください。

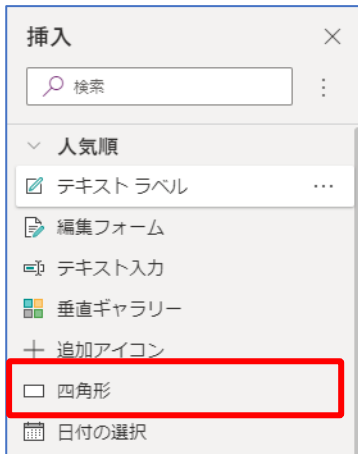
フォントサイズ	14
フォントの太さ	B 太字 ▼

ギャラリー内の見た目が以下のように変化していることを確認してください。



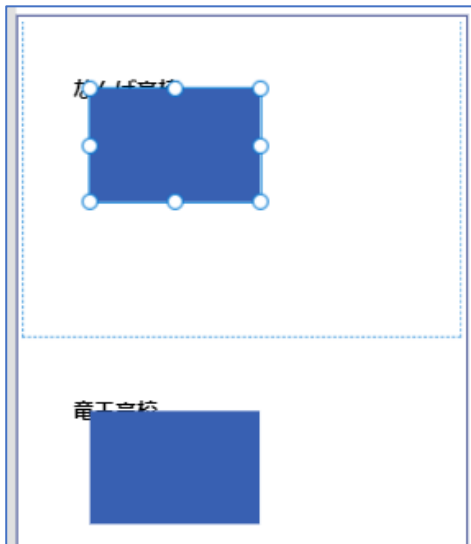
④ データの区切り線を挿入します。

ギャラリーのテンプレートがアクティブな状態で、[挿入] ペインから [四角形] を選択してください。



四角形がギャラリーに追加されました。

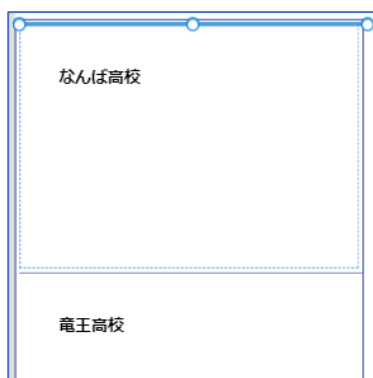
名前を "SchoolSeparator" としてください。



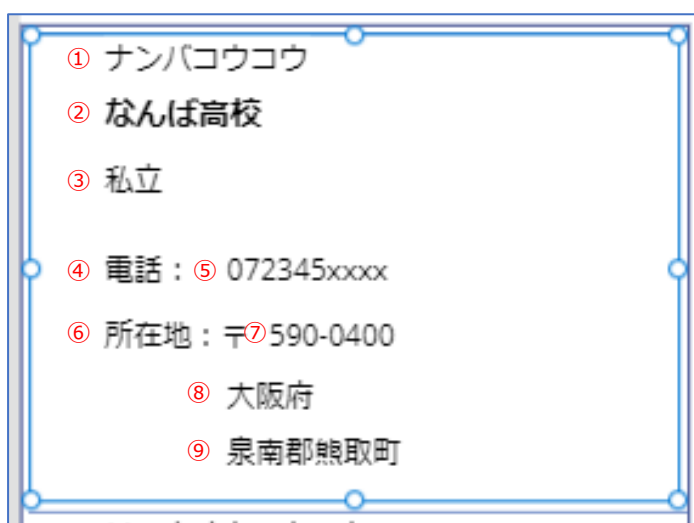
プロパティペインの [詳細設定] で以下のように設定してください。



ギャラリーの見た目はこのようになります。





[SchoolGallery] に必要な他のフィールドを追加して、下記のような見た目となるようにしましょう。



No.	コントロール名	コントロール種類	Text プロパティの設定値
1	SchoolNameKana	テキストラベル	ThisItem.学校名カナ
2	SchoolName	テキストラベル	ThisItem.学校名
3	SchoolType	テキストラベル	ThisItem.学校種別
4	SchoolPhoneLabel	テキストラベル	"電話:"
5	SchoolPhone	テキストラベル	ThisItem.電話番号
6	SchoolAddressLabel	テキストラベル	"所在地 : 〒"
7	SchoolPostalCode	テキストラベル	ThisItem.郵便番号
8	SchoolPrefecture	テキストラベル	ThisItem.都道府県
9	SchoolAddress	テキストラベル	ThisItem.所在地

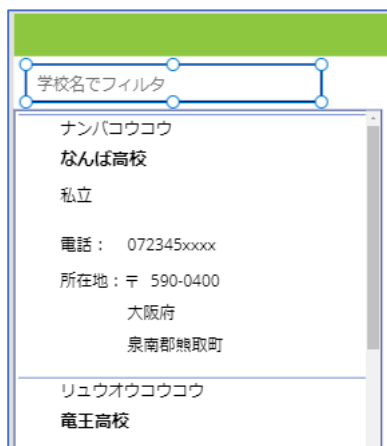
2. [SchoolListGallery] に学校名のフィルター機能をつけてみましょう。

- ① ギャラリーの上の余白に [テキスト入力] を挿入し、“SchoolNameFilter” という名前にしてください。
プロパティは以下のように設定します。

SchoolNameFilter 	
既定	値なし
書式	テキスト 
ヒントのテキスト	学校名でフィルタ

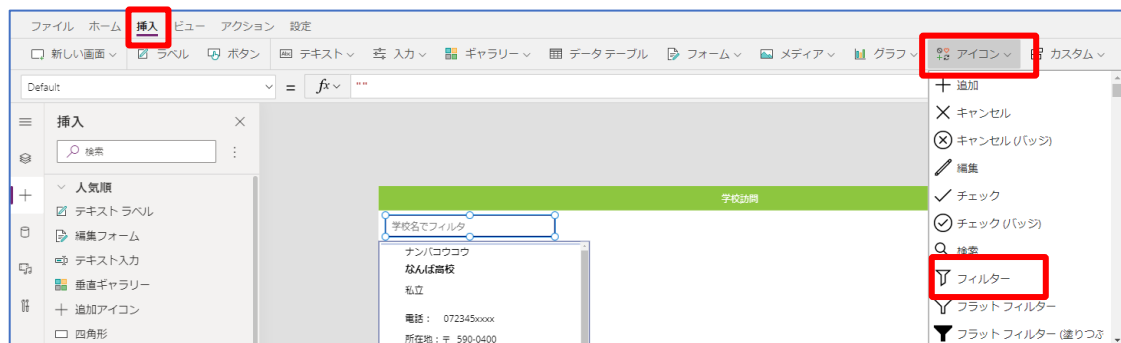
※注：[値なし] と表示されている箇所はもともと設定されていた値をクリアしています。

見た目はこのようになります。

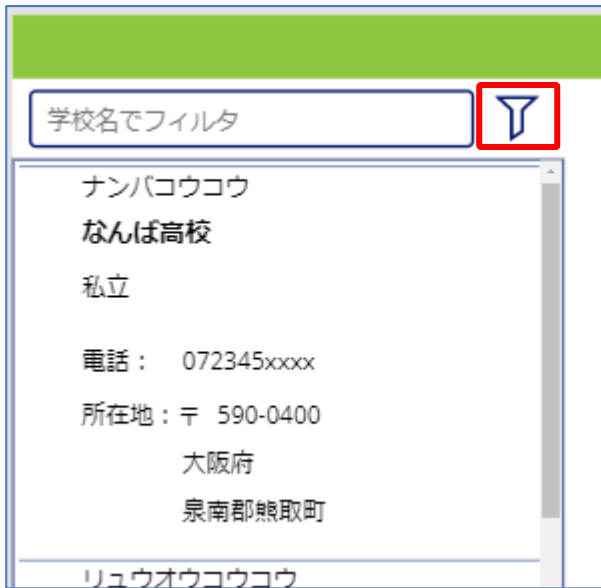


フィルターであることがわかりやすいよう、アイコンを追加しましょう。

アイコンは [挿入] リボンから選択します。

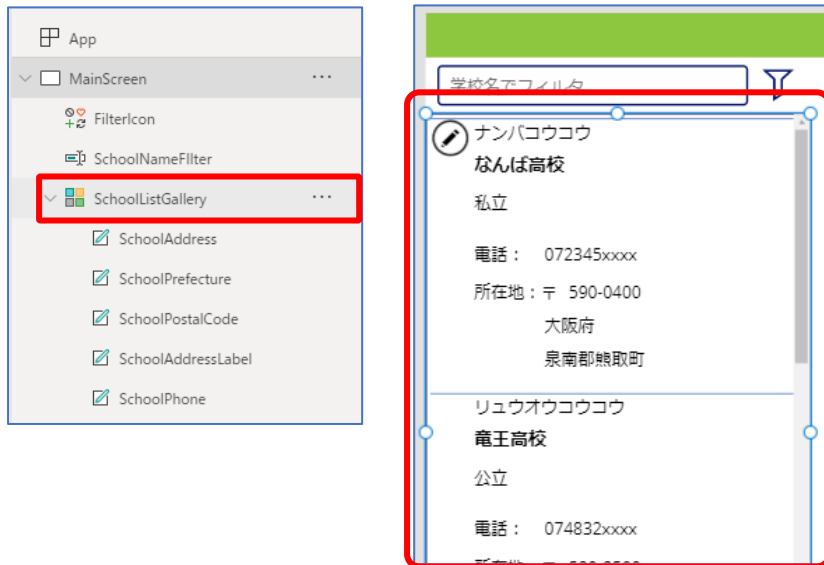


アイコンの位置とサイズを調整し、名前を“FilterIcon”としてください。



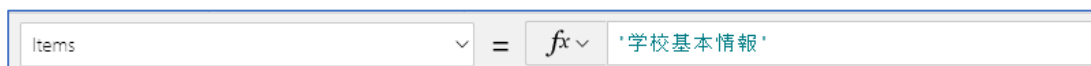
② [SchoolListGallery] にフィルターの入力が反映されるようにします。

まず、[SchoolListGallery] を選択してください。ツリービューペインから選択すると目的のオブジェクトに直接アクセスできるので安全です。



数式バーでプロパティを設定するため、[Items] プロパティを選択してください。

現在は '学校基本情報' が設定されています。



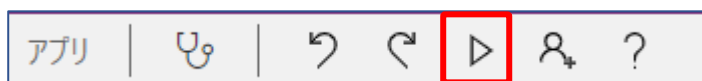
このプロパティを次の数式に変更してください。

```
Filter(  
  学校基本情報,  
  '学校基本情報 (ビュー)'.アクティブな学校基本情報,  
  StartsWith(  
    学校名,  
    SchoolNameFilter.Text  
  )  
)
```

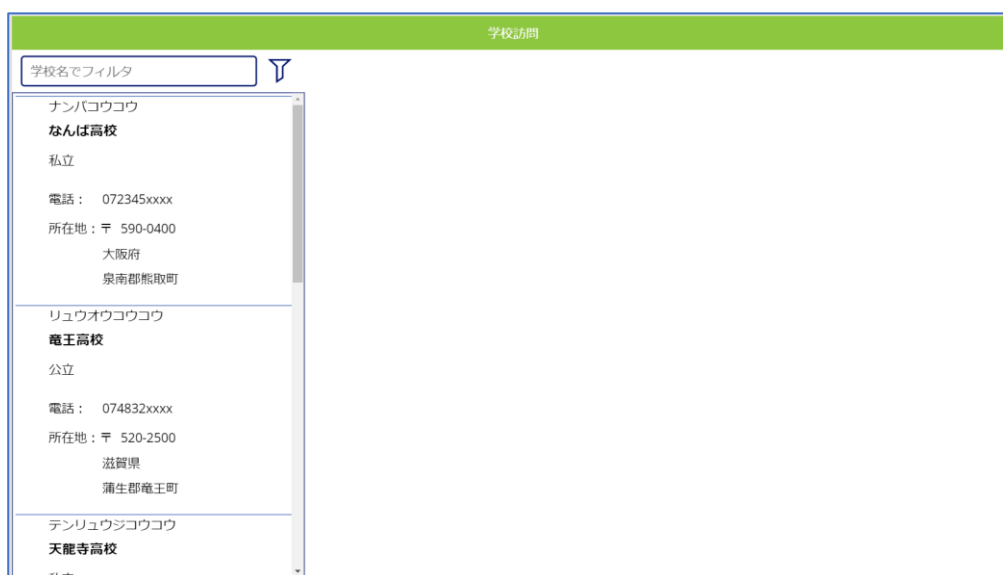
※注：読みやすさのために改行とインデントを使用しています。

入力できたら現在の状態でテストしてみましょう。

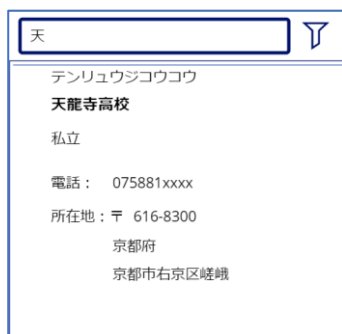
メニューの右側にある [アプリのプレビュー] をクリックしてください。



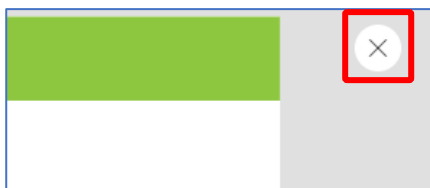
アプリの画面が起動します。



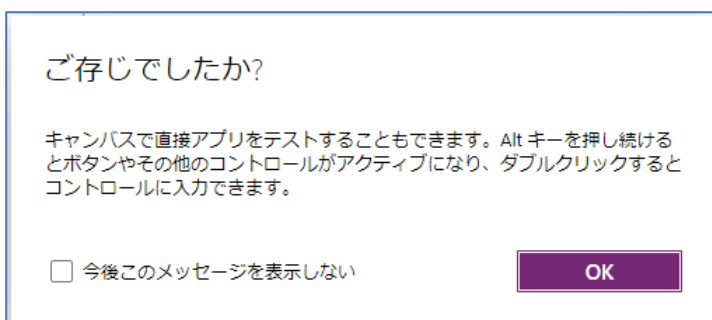
フィルターに“天”と入力してください。学校名が“天”で始まる学校に絞り込まれました。



フィルターのテストができたので、アプリ画面の横に表示されている [X] をクリックしてプレビューを終了してください。



次のメッセージが表示されるので [OK] で閉じてください。

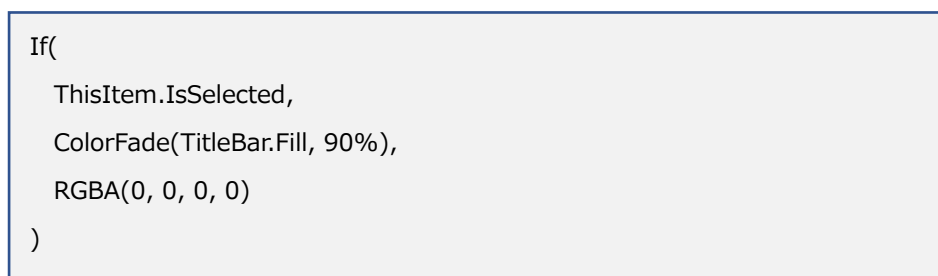


3. [SchoolListGallery] でどのデータが選択されているかを色で判別できるようにします。

① [SchoolListGallery] を選択し、[TemplateFill] プロパティを数式バーに表示してください。

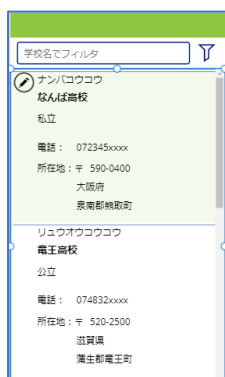


② このプロパティ値を次の数式に変更してください。



この数式は、選択されている場合に背景色を [TitleBar] の背景色を 90% フェードした色にします。

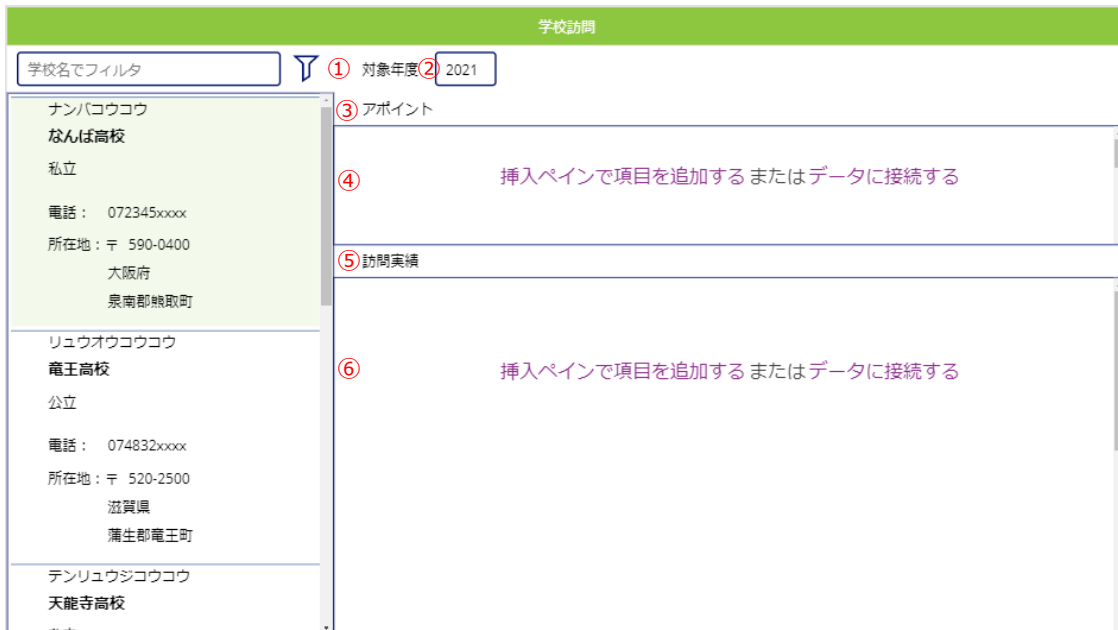
Alt キーを押しながら [SchoolListGallery] のデータから一つ選択 (クリック) してください。



クリックしたデータの背景色が薄い緑になります。

4. [MainScreen] にその他のコントロールを配置します。

図と表を参考にコントロールを配置してください。



No.	コントロール名	コントロール種類	プロパティ名	プロパティ値
1	TargetYearLabel	テキストラベル	Text	"対象年度"
2	TargetYearText	テキスト入力	Format	TextFormat.Number
			Default	Year(Today())
3	AppointLabel	テキストラベル	Text	"アポイント"
4	AppointmentGallery	空の垂直ギャラリー	-	※下記(a)参照
5	HistoryLabel	テキストラベル	Text	"訪問実績"
6	HistoryGallery	空の垂直ギャラリー	-	※下記(b)参照

(a) [AppointmentGallery] のプロパティ

[Items]

```
Filter(
    学校訪問履歴情報,
    '学校訪問履歴情報 (ビュー)'.学校訪問アポイント情報,
    訪問年度 = Value(TargetYearText.Text) &&
    学校.学校コード = SchoolListGallery.Selected.学校コード
)
```

[TemplateFill]

```
If(ThisItem.IsSelected, ColorFade(TitleBar.Fill, 90%), RGBA(0, 0, 0, 0))
```

(b) [HistoryGallery] のプロパティ

[Items]

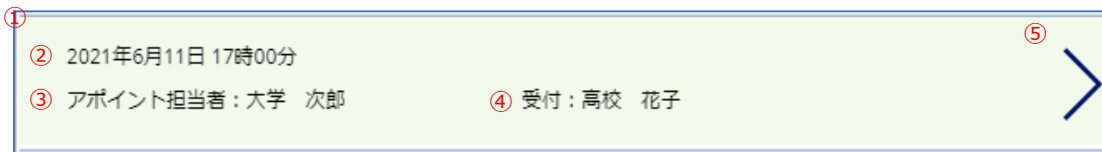
```
Filter(
  学校訪問履歴情報,
  '学校訪問履歴情報 (ビュー)'.学校訪問実績情報,
  訪問年度 = Value(TargetYearText.Text) &&
  学校.学校コード = SchoolListGallery.Selected.学校コード
)
```

[TemplateFill]

```
If(ThisItem.IsSelected, ColorFade(TitleBar.Fill, 90%), RGBA(0, 0, 0, 0))
```

5. [AppointmentGallery] にフィールドを配置します。

図と表を参考にコントロールを配置してください。



No.	コントロール名	コントロール種類	プロパティ名	プロパティ値
1	AppointSeparator	四角形	X	0
			Y	0
			Height	1
			Width	Parent.Width
2	AppointedDateTime	テキストラベル	Text	ThisItem.アポイント日時
3	Appointer	テキストラベル	Text	"アポイント担当者:" & ThisItem.アポイント担当者
4	Acceptor	テキストラベル	Text	"受付:" & ThisItem.受付担当者
5	VisitRegiserNavigateIcon	アイコン	OnSelect	※下記(a)参照

※注：プロパティ値を設定する際、“&” で接続することにより 2 つの値を結合することができます。

(a) [VisitRegiserNavigateIcon] のプロパティ

[OnSelect]

```
Select(Parent);
Set(
  RegisterArgs,
  {
    School:ThisItem.学校,
    Record:ThisItem,
    Mode:"Register"
  }
);
Navigate(RegisterVisitScreen)
```

※注：[OnSelect] は当該コントロールが選択（クリック）された時の動作を指定するプロパティです。

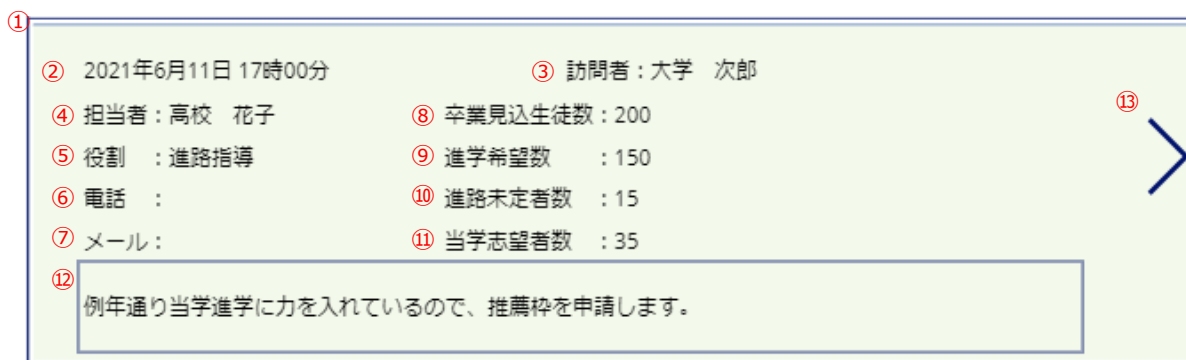
このような、動作のプロパティでは動作を表す関数をセミコロン [;] で複数設定することが可能です。上記例では ①親オブジェクトを選択し、②グローバル変数を設定し、③画面遷移を行う、という 3 つの関数を指定しています。

Navigate 関数は、この時点では [RegisterVisitScreen] が存在しないためエラーとなります。

※注：グローバル変数はアプリ内のどの場所からでも参照できる変数です。このアプリでは照会画面から更新画面に遷移する際、登録モードと更新モードで処理を切り替える必要があり、また、照会画面で選択されたデータを更新画面に知らせる必要があるため、グローバル変数にこれらの情報を設定してから画面遷移をしています。

6. [HistoryGallery] にフィールドを配置します。

図と表を参考にコントロールを配置してください。



No.	コントロール名	コントロール種類	プロパティ名	プロパティ値
1	HistorySeparator	四角形	X	0
			Y	0
			Height	1
			Width	Parent.Width
2	VisitedDateTimeText	テキストラベル	Text	ThisItem.訪問日時
3	VisitorText	テキストラベル	Text	"訪問者 : &ThisItem.訪問者
4	IntervieweeText	テキストラベル	Text	"担当者 : " & ThisItem.担当者
5	IntervieweeRoleText	テキストラベル	Text	"役割 : &ThisItem.担当者役 割
6	IntervieweePhoneText	テキストラベル	Text	"電話 : &ThisItem.担当者電 話番号
7	IntervieweeMailText	テキストラベル	Text	"メール : &ThisItem.担当者メ ールアドレス
8	ExpectedGraduatesText	テキストラベル	Text	"卒業見込生徒数 : &ThisItem.卒業見込 生徒数
9	ExpectedExamineesText	テキストラベル	Text	"進学希望数 : &ThisItem.進学希望 数

10	CourseUndecidedsText	テキストラベル	Text	"進路未定者数 : "&ThisItem.進路未定者数
11	AspirantsForUsText	テキストラベル	Text	"当学志望者数 : "&ThisItem.当学志望者数
12	VisitSummaryText	テキストラベル	Text	ThisItem.訪問概要
13	HistoryEditNavigateIcon	アイコン	OnSelect	※下記(a)参照

(a) [HistoryEditNavigateIcon] のプロパティ

[OnSelect]

```

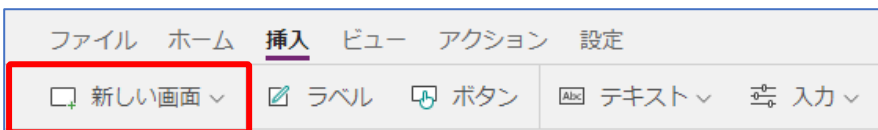
Select(Parent);
Set(
  RegisterArgs,
  {
    School:ThisItem.学校,
    Record:ThisItem,
    Mode:"Edit"
  }
);
Navigate(RegisterVisitScreen)

```

※注 : Navigate 関数は、この時点では [RegisterVisitScreen] が存在しないためエラーとなります。

7. 上記手順で発生しているエラーを解決するために、更新画面を作成します。

画面上部の [新しい画面] から空の画面を作成し、名前を "RegisterVisitScreen" としてください。



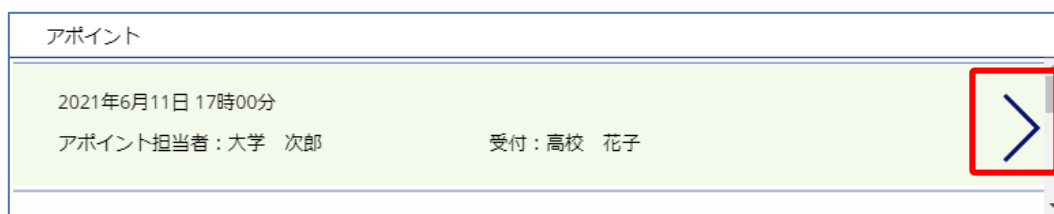
7.4. 更新画面を作成する

1. [RegisterVisitScreen] を開き、タイトルバーとしてテキストラベルを挿入してください。
名前を "TitleBarVisit" として、画面の幅いっぱいとなるよう位置とサイズを調整してください。
※注 : mainScreen の [TitleBar] をコピー & ペーストすると便利です。

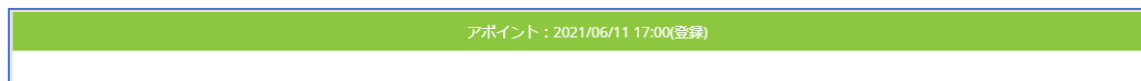
[Text] プロパティに次の数式を設定してください。

```
RegisterArgs.School.学校名 & " " & If(  
    RegisterArgs.Mode = "Register",  
    "アポイント : " & Text(  
        RegisterArgs.Record.アポイント日時,  
        "yyyy/mm/dd hh:mm"  
    ) & "(登録)",  
    "訪問日時 : " & Text(  
        RegisterArgs.Record.訪問日時,  
        "yyyy/mm/dd hh:mm"  
    ) & "(修正)"  
)
```

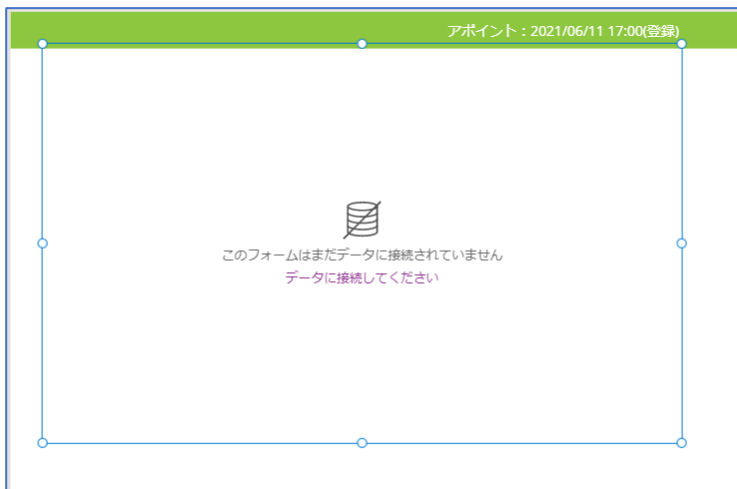
この数式の結果を確認するため、[MainScreen] を開いて、アイコン [VisitRegisterNavigateIcon] を Alt キーを押しながらクリックしてください。



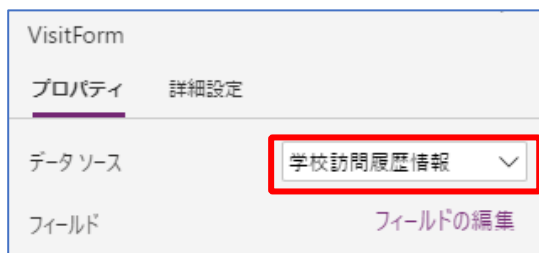
[RegisterVisitScreen] が開き、[TitleBarVisit] は図のように表示されます。



2. [RegisterVisitScreen] を開いたまま、挿入ペインから編集フォームを挿入してください。
このフォームの名前は " VisitForm" としてください。



追加した編集フォームのプロパティペインで、[データ ソース] を選択します。
このフォームが更新するのは学校訪問履歴情報テーブルなので、図のように指定してください。



いくつかのフィールドが自動的に展開されます。フォームのサイズを、タイトルバーを除いた画面いっぱいとなるよう調整してください。



プロパティペインで次のように入力してください。

The screenshot shows the 'VisitForm' property pane with the following settings:

- タブ: プロパティ (詳細設定)
- データソース: 学校訪問履歴情報
- フィールド: フィールドの編集
- 列へのスナップ: オン (スイッチ)
- 列: 1
- レイアウト: 縦
- 既定モード: 編集

フィールドの並びが図のようになります。

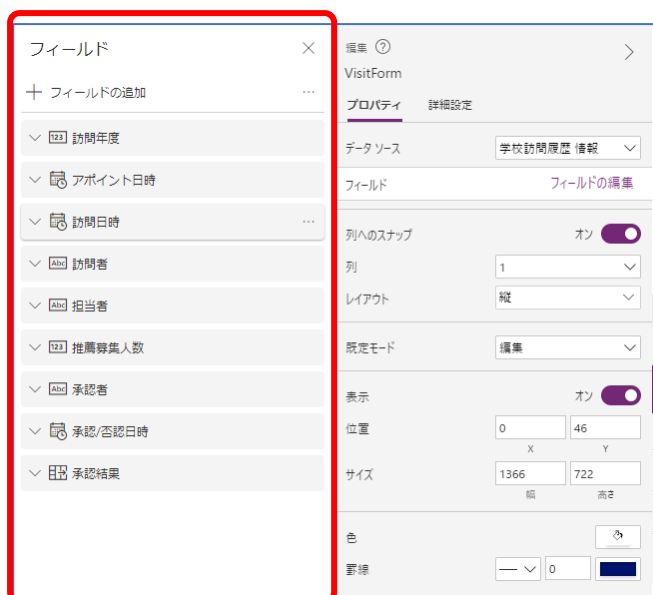
The screenshot shows the form layout with the following fields arranged vertically:

- * 訪問年度
- アポイント日時: 2001年12月31日 00:00
- 訪問日時: 2001年12月31日 00:00
- 訪問者
- 担当者
- 推薦募集人数
- 承認者

3. [VisitForm] で入力するフィールドを設定します。
プロパティペインの [フィールドの編集] をクリックしてください。

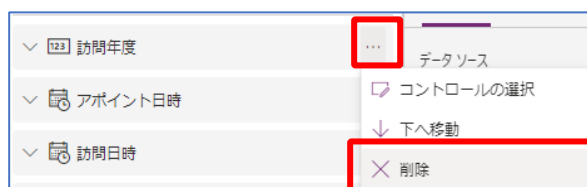
The screenshot shows the 'VisitForm' property pane with the 'フィールド' section highlighted by a red box, containing the text 'フィールドの編集'.

プロパティペインの横（左側）にフィールド選択用のパネルが開きます。



このアプリで [訪問年度] を更新することはありません。このフィールドを削除してください。

削除するにはフィールド名の横にある [...] をクリックしてメニューを表示し、[削除] をクリックします。



同様に、[アポイント日時]、[承認/否認日時]、[承認結果] を削除してください。

図のような状態となります。



次に、不足している更新対象フィールドを追加します。

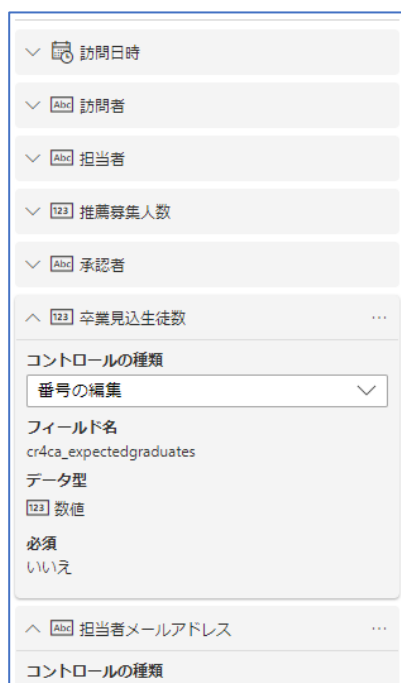
[フィールドの追加] をクリックしてください。以下のようなパネルが開きます。



このパネルで下表のフィールドを選択してチェックをオンにしてください。

No.	フィールド名
1	担当者役割
2	担当者電話番号
3	担当者メールアドレス
4	卒業見込生徒数
5	進学希望数
6	進路未定者数
7	当学志望者数
8	訪問概要
9	申請に進む
10	申請者
11	申請日時
12	クローズド

フィールドが追加されました。



これらのフィールドを、入力する順に並べ替えます。

フィールドの並べ替えは [...] をクリックしてメニューを表示し、[上(または下)へ移動] を選択してもできますが、フィールドをドラッグ & ドロップで行う方が直感的に操作できます。

下図の順番となるよう、フィールドを並べ替えてください。

- ▼ 訪問日時
- ▼ Abc 訪問者
- ▼ Abc 担当者
- ▼ Abc 担当者役割
- ▼ Abc 担当者電話番号
- ▼ Abc 担当者メールアドレス
- ▼ 123 卒業見込生徒数
- ▼ 123 進学希望数
- ▼ 123 進路未定者数
- ▼ 123 当学志望者数
- ▼ Abc 訪問概要
- ▼ 申請に進む
- ▼ 123 推薦募集人数
- ▼ Abc 承認者
- ▼ Abc 申請者
- ▼ 申請日時
- ▼ クローズド

更新対象のテーブルを設定することにより上記のようにフィールドの定義ができましたが、この編集フォームはまだ、具体的にどのデータ (レコード) を更新するかを知りません。更新対象となるデータは [Item] プロパティで設定します。

Item = fx RegisterArgs.Record

最後に、[OnSuccess] プロパティを設定します。

このプロパティには、更新が成功したときの動作を設定します。

OnSuccess = fx Back(Fade)

4. 各フィールドのプロパティを設定します。

フィールドを編集するためにはまずフィールドのロックを解除する必要があります。

例として [訪問日時] のデータカード (名前 : [訪問日時_DataCard? (? は数字)]) を選択して、プロパティペインの [詳細設定] を開きます。

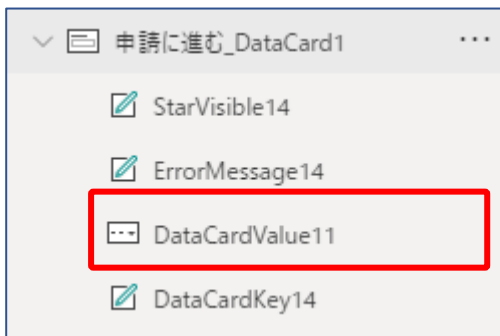
[プロパティを変更するためにロックを解除します] をクリックしてください。



これでプロパティを入力できる状態となりました。

フォームに展開される [カード] コントロールにはラベルや入力コントロールなど複数のコントロールが含まれます。通常、カードの名前にはデータフィールドの名前が含まれておりフィールドが識別可能なためコントロール名は変更しなくてもよいのですが、カードの中に含まれるコントロールはどのフィールドに紐づくのが識別できない名前となっています。このため、プロパティに設定する関数がカード内のコントロールを参照することがある場合、少なくともそのコントロールの名前は変更しておくべきです。

当アプリでは、[申請に進む] カードに含まれる値用コントロールを参照する関数がありますので、このコントロールの名前を変更しておきます。



※注： 値用の入力コントロールの名前は通常 [DataCardValue? (? は数字)] と生成されますが、念のためクリックして目的のコントロールであることを確認してください。

このコントロールの名前を " ProceedToApplicationValue" としてください。

下表に従って各フィールド（カード）のプロパティを設定してください。

No.	フィールド	プロパティ名	プロパティ値
1	訪問日時	Default	If(RegisterArgs.Mode="Register", Now(), ThisItem.訪問日時)
2		Required	true
3	訪問者	Default	If(RegisterArgs.Mode="Register", User().FullName, ThisItem.訪問者)
4		Required	true
5	担当者	Required	true
6	担当者役割	Required	true
7	訪問概要	Required	true
8	申請に進む	DisplayMode	If(RegisterArgs.Mode="Register", Parent.DisplayMode, DisplayMode.View)
9	承認者	DisplayMode	If(RegisterArgs.Mode="Register", If(ProceedToApplicationValue.Selected.Value, DisplayMode.Edit, DisplayMode.Disabled), DisplayMode.View)
10		Required	ProceedToApplicationValue.Selected.Value
11	申請者	Default	If(RegisterArgs.Mode="Register" And ProceedToApplicationValue.Selected.Value, User().Email ,ThisItem.申請者)
12		DisplayMode	If(RegisterArgs.Mode="Register", If(ProceedToApplicationValue.Selected.Value, DisplayMode.Edit, DisplayMode.Disabled), DisplayMode.View)
13		Required	ProceedToApplicationValue.Selected.Value
14	推薦募集人数	DisplayMode	If(RegisterArgs.Mode="Register", If(ProceedToApplicationValue.Selected.Value, DisplayMode.Edit, DisplayMode.Disabled), DisplayMode.View)
15		Required	ProceedToApplicationValue.Selected.Value
16	申請日時	Default	If(RegisterArgs.Mode="Register",Now(), ThisItem.申請日時)
17		Visible	false

18	クローズド	Default	If(RegisterArgs.Mode="Register", If(ProceedToApplicationValue.Selected.Value, Last(Choices(学校訪問履歴情報.クローズド).Value).Value, First(Choices(学校訪問履歴情報.クローズド).Value).Value), ThisItem.クローズド)
17		DisplayMode	DisplayMode.Disabled

5. 実行/キャンセル用にアイコンを挿入します。

タイトルバー上に X アイコンと ✓ アイコンを配置しましょう。



それぞれの名前のプロパティを設定してください。

No.	アイコン	名前	プロパティ名	プロパティ値
1	X	CancelIcon	OnSelect	Back(ScreenTransition.Fade)
			Color	RGBA(255, 0, 0, 1)
2	✓	ExecuteIcon	OnSelect	SubmitForm(VisitForm)
			Color	RGBA(32, 113, 45, 1)

6. アプリを保存します。

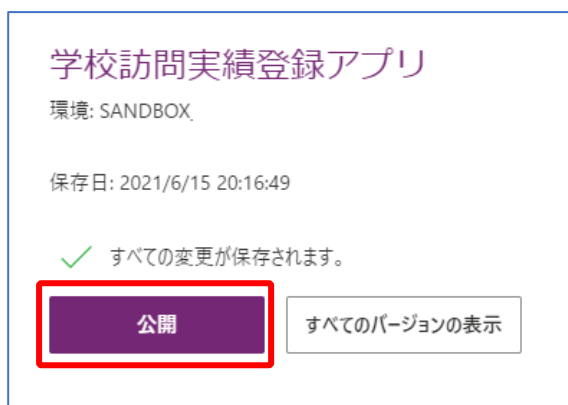
メニューから [ファイル] を選択してください。



次の画面で [保存] をクリックしてください。

The screenshot shows the '学校訪問実績登録アプリ' (School Visit Record Registration App) save screen. The title is '学校訪問実績登録アプリ' and the environment is 'Sandbox'. The save date is '2021/6/15 20:08:39'. There is a text area for 'バージョンに関するメモ' (Memo regarding version) with the text '実行済みの更新に関するメモを残します (オプション)'. At the bottom, there is a purple '保存' (Save) button highlighted with a red rectangular box.

最後に、他のユーザーがアクセスできるように、[公開] をクリックします。



以上でキャンバス アプリは完成です。

現在のブラウザータブを閉じて終了してください。

8. 演習 6 (オプション 2) リマインダーフローの作成

想定している業務フローでは、推薦募集人数の承認により願書送付予定日が自動的に登録されます。

このような場合、願書送付の担当者は願書送付をしなくてはならないということに気が付かないことがあります。

この演習ではそのようなケースをカバーするために、願書送付予定日が近づいたタイミングで願書送付担当者に対してリマインダーのメールを送信する機能を作成します。

この機能は Power Automate のクラウドフローで作成します。

※注：この演習はオプションです。実施しなくてもアプリの機能としては問題ありません。

8.1. クラウド フローの新規作成

1. クラウド フローを新規作成します。

ここで作成するフローは、毎日スケジュールされた時刻に起動して願書送付予定日が近づいたデータを抽出し、それぞれの願書送付担当者に対してメールで通知するという仕組みで作成します。

このため、新規作成の手順自体は演習 4 で行ったものと同様ですが、フロー起動のトリガーが異なります。

[新しいフロー] の選択肢から [スケジュール済みクラウドフロー] を選択してください。



この選択により次のような画面が表示されます。

表に従って設定項目を入力してください。

No.	項目名	入力
1	フロー名	願書送付リマインダーフロー
2	開始日	(任意 ※今日以前の日付)
3	時間	(任意)
4	繰り返し間隔	1
5	(単位)	日

スケジュール済みクラウド フローを構築する

追加の作業なしで何が重要かを把握し、フローを実行するタイミングと頻度を自分で選択します。

例:

- チームへの通知を自動化して経費精算書を送信する
- データを指定された記憶域に定期的に自動バックアップする

フロー名: ① 願書送付リマインダーフロー

このフローを実行する*

開始日: ② 21/6/16 ③ 10:00 AM

繰り返し間隔: ④ 1 ⑤ 日

このフローの実行頻度: 毎日

スキップ 作成 キャンセル

2. フローを構成していきます。

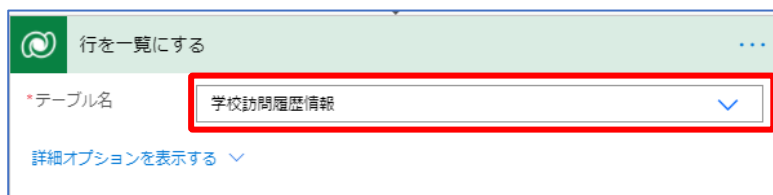
新規作成時にトリガー [Reccurence] が作成されています。この下にステップを追加します。

① データを抽出します。

[Reccurence] の下の [新しいステップ] をクリックし、図のアクションを探してください。



このアクションを選択し、[テーブル名] に [学校訪問履歴情報] を設定してください。



[詳細オプションを表示する] をクリックして各プロパティを設定します。
完成形はこのようになります。

No.	プロパティ名	入力
1	列を選択する	以降のステップで使用する列名をコンマで区切って指定します。※下記 a. 参照
2	行のフィルター	抽出したい行の条件を OData 形式で指定します。 ※下記 b. 参照

a. [列を選択する] で指定する列名

No.	列名 (表示名)
1	学校名
2	願書送付要否
3	願書送付済み
4	願書送付予定日
5	願書送付担当者
6	推薦募集人数

※注：プロパティ値として設定する際は上記表示名ではなく、[名前] を指定する必要があります。

b. [行のフィルター] に指定するクエリ式

```
[クローズド] eq false and  
[願書送付要否] eq true and  
[願書送付予定日] lt @{addDays(utcNow(), 7)}
```

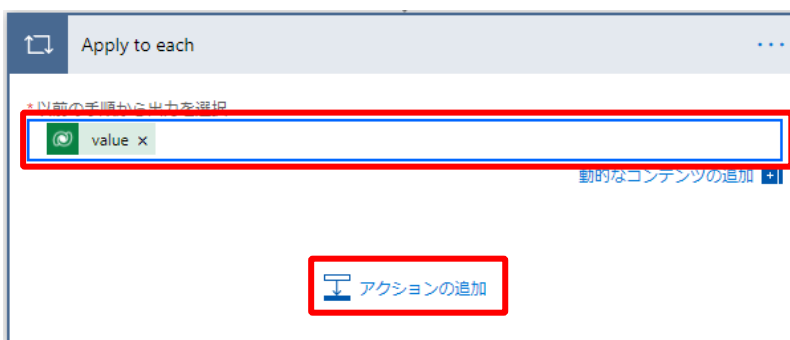
※注：列名は表示名で記述していますが、実際には [名前] を記述する必要があります。
“@{addDays(utcNow(), 7)}” は現在時刻から 7 日後を示す関数です。

② 抽出したデータごとに処理（ループ）します。

[行を一覧にする] の下の [アクションの追加] をクリックし、図のアクションを探してください。



選択して [] に動的なコンテンツから [value] を設定してください。
設定できたら、[アクションの追加] をクリックください。

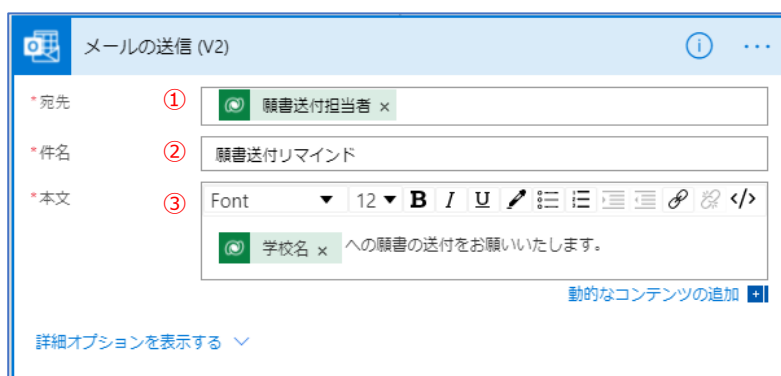


次のアクションを探して、選択してください。



下表を参考に各プロパティを設定してください。

完成形は以下のようになります。



No.	プロパティ	入力
1	宛先	※動的なコンテンツから [願書送付担当者] を設定してください
2	件名	“願書送付リマインド”
3	本文	@{items('Apply_to_each')}['学校名']}への願書の送付をお願いいたします。

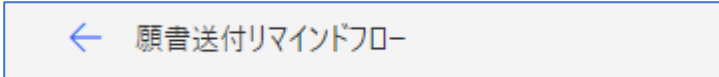
※注：列名を表示名で記述していますが、実際の入力時は [名前] で記述してください。

3. フローを有効化します。

フローチェッカーの結果が問題なければ、保存してフローの一覧に戻ります。



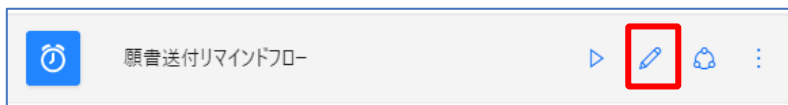
フローの一覧に戻るメニューは画面左上部にあります。



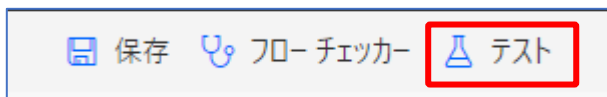
8.2. スケジュールされたフローのテスト

スケジュールトリガーのフローは演習 4 のようにデータを登録することで簡単にテストができません。このようなフローはフロー デザイナーの [テスト] 機能を利用することによりテストが実施できます。

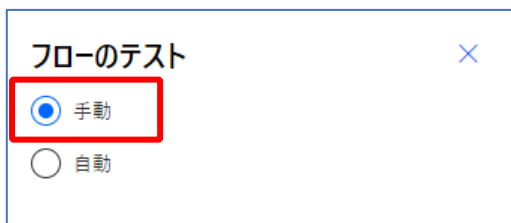
1. フローを編集モードで開くには、一覧で [編集] アイコンをクリックします。



フローが開いたら、画面右上にある [テスト] をクリックしてください。



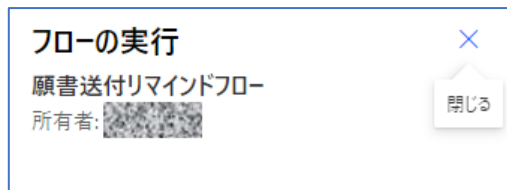
[フローのテスト] パネルが開くので、[手動] を選択してください。



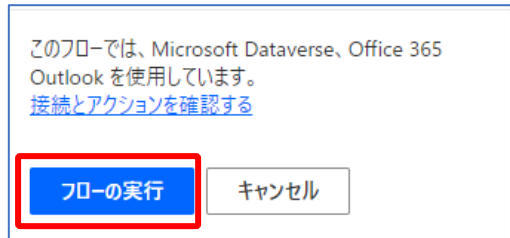
その後、パネルの下部にある [テスト] をクリックします。



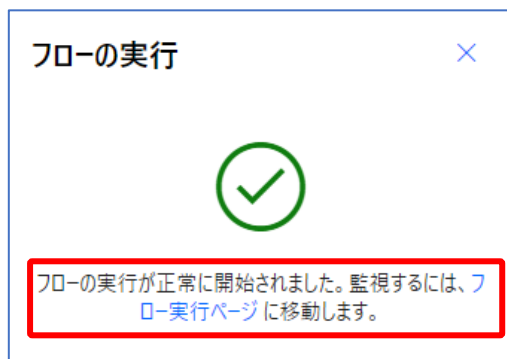
パネルの表示内容が以下のように変わります。



パネル下部にある [フローの実行] をクリックしてください。

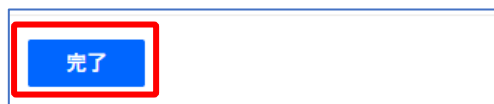


パネルが以下の表示に変わりますので、[フロー実行ページ] リンクをクリックしてください。



パネルの後ろの画面はフロー実行ページに遷移しています。

パネル下部の [完了] をクリックしてパネルを閉じてください。



フロー実行ページで実行状況を確認します。

[実行中] と表示されている場合はしばらく待ってください。処理が完了すると [成功] に変わります。



開始時刻	時間	状況
6月16日 11:42 (1分前)	00:00:01	成功

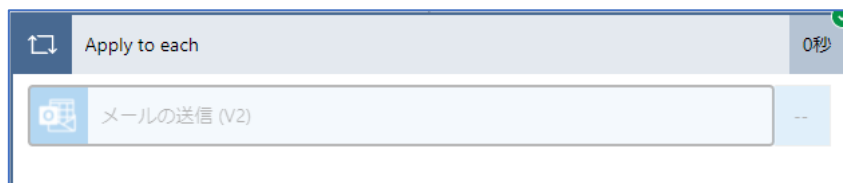
処理結果を確認するため、[開始時刻] のリンクをクリックしてください。



処理結果が下記のように表示されます。[Apply to each] をクリックして展開してください。



メールの送信がグレーアウトしています。[行を一覧にする] で抽出されたデータが 1 件もなかったため処理されなかったことが原因と考えられます。



検証のため、データを条件に合致するように変更して再度テストします。

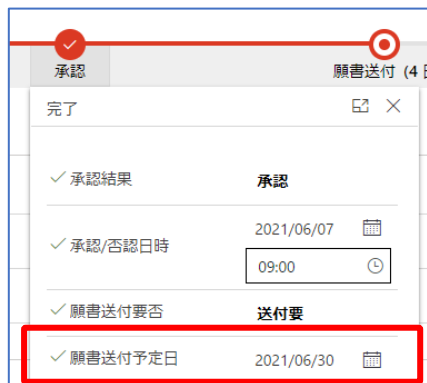
これまでの演習の手順どおりであれば、数件の願書送付予定データがありますので、そのデータを変更します。モデル駆動型アプリを開いてください。

学校訪問履歴情報を開き、ビューを [願書送付状況] に切り替えると以下のデータが表示されました。

願書送付状況			
学校名 (学校)	願書送付要否	推薦募集人数	願書送付予定日
なんば高校	送付要	15	2021/06/30
竜王高校	送付要	20	2021/07/11

この内の 1 件を変更します。変更するデータを選択して [編集] をクリックしてください。

[承認] ステージをクリックして、[願書送付予定日] を今日から 1 週間以内の日付に変更してから、[保存して閉じる] をクリックしてください。更新が確認できたらモデル駆動型アプリは閉じてください。



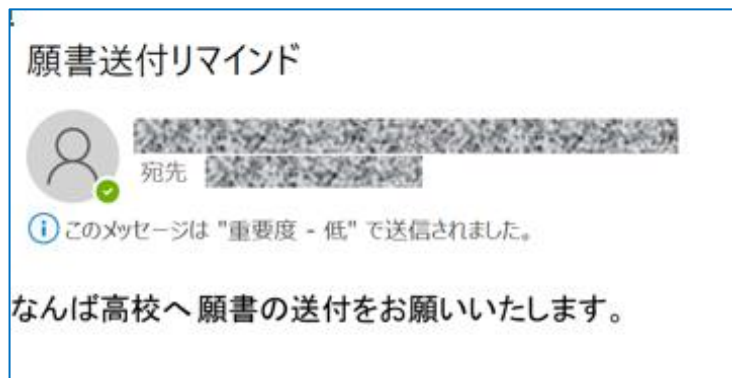
先ほどと同じ手順でフローのテストを実行してください。

開始	時間	状況
6月16日 12:24 (1分前)	00:00:01	成功
6月16日 11:42 (43分前)	00:00:01	成功

新しい開始時刻をクリックして結果を確認します。

今回は対象があったのでメールの送信が実行されています。

下記のメールが届きました。テストは成功です。



【 参考情報 】

メールの本文には上記サンプルの [学校名] のように動的なコンテンツとして様々な情報を埋め込むことが可能ですが、日付の情報 (例えば、[願書送付予定日]) を埋め込む場合は考慮が必要です。

これは Dataverse 上の日付関連データは「協定世界時 (UTC+00:00)」で保管されており、日本時間とは 9 時間の時差が生じるためです。

日付項目については、[タイムゾーンの変換] というアクションによって日本時間 (UTC+09:00) に変換した値を使用することにより、受信者にとって違和感のない日付を埋め込むことができます。

※ [タイムゾーンの変換] の出力を使用するには、例えば
“ 願書送付予定日は@{body('タイム_ゾーンの変換')}です。”
のように記述します。

9. おわりに

Power Apps と Power Automate を使って実際にアプリを作成していただきましたが、いかがでしたでしょうか。

経営判断や計画立案にスピードを求められる現代社会においては、データ分析の仕組みを手軽に自作できることには非常に大きな価値があります。

プログラム言語に馴染みのある方にはむしろ、ノーコーディング（またはローコーディング）であるが故の難しさを感じられる部分があるかもしれませんが、ミッションクリティカルで小さなアプリを開発する上で Power Apps は非常に強力なツールです。慣れれば Excel で関数を使うのと同じような感覚で素早くアプリを開発することができます。

当書を通じて Power Apps の価値を理解していただき、また Power Apps を活用した業務改善のヒントを見出していいただければ幸いです。

※ご参考までに、詳細な学習用ソースをご紹介します。

ラーニングコンテンツ

Power Platform 全般	https://docs.microsoft.com/ja-jp/learn/powerplatform/
Power Apps	https://docs.microsoft.com/ja-jp/learn/powerplatform/power-apps/
Power Automate	https://docs.microsoft.com/ja-jp/learn/powerplatform/power-automate/

技術情報

Power Platform 全般	https://docs.microsoft.com/ja-jp/power-platform/
Power Apps	https://docs.microsoft.com/ja-jp/powerapps/
Power Automate	https://docs.microsoft.com/ja-jp/power-automate/