

# CPSC 304 Project Cover Page

Milestone #: 2

Date: October 20th, 2023

Group Number: 134

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Benny Li	28772598	Cli66 e0p8y	cli66@student.ubc.ca
Helen Zhou	46073292	hzhou2 a2v5h	helenzhou212@gmail.com
Kelly Chen	36916120	chenkai y7v1z	kk.chen.1058@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

## Summary of Project

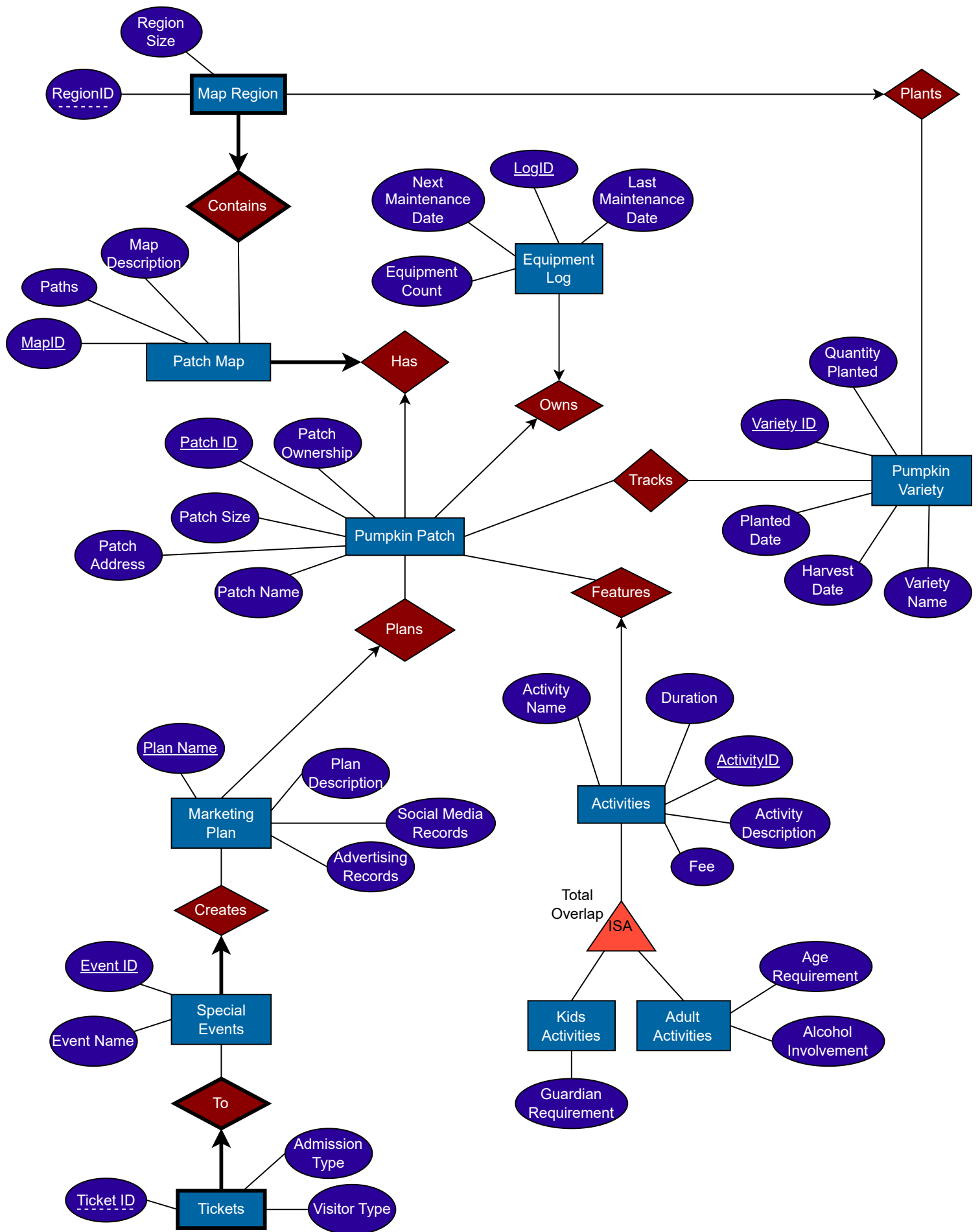
The pumpkin patch application is an agricultural management tool to track and manage the necessary aspects of a patch, including pumpkin variety, visitor management, and marketing events. The application can be used by owners, farmers, visitors, government and regulatory bodies, educational institutions, etc. to manage pumpkin patches across the region.

## ER Diagram

See next page for the ER diagram.

Summary of changes made to the ER diagram from Milestone 1:

- Added a constraint (total overlap) to the ISA relationship as per TA's suggestion.
- Added an entity Adults Activity to the ISA relationship as per TA's suggestion.
- Added EventID and EventName attributes to the Special Event entity.
- Removed Seasonal Trends entity. We decided that while analyzing seasonal trends can be valuable for other businesses, for a pumpkin patch it is somewhat inherent that their sales and activities will revolve around the Halloween season.
- Removed Reviews entity. Upon closer examination, it became apparent that the relationship between Activities and Reviews didn't align with the essence of our project, as collecting reviews from activities may not be the most intuitive way to gather feedback for a pumpkin patch.
- Changed the Equipment entity into an Equipment Log and adjusted the nature of the Owns relationship from a many-to-many to a one-to-one relationship. This modification is driven by the aim to diversify the types of relationships present in our database. Given the project's existing complexity, which already includes numerous many-to-many relationships, this change helps create a more balanced and structured schema.
- Changed the Features relationship between Pumpkin Patch and Activities from a many-to-many relationship to a one-to-many relationship. This adjustment aligns our database design more closely with real-world scenarios where a pumpkin patch typically creates one or multiple signature activities, each of which is specific to that particular pumpkin patch.
- Change the Plants relationship between Map Region and Pumpkin Variety from a many-to-many relationship to a one-to-many relationship. This alteration is a reflection of the real-world scenario where each map region is typically associated with the planting of a single pumpkin variety. On the other hand, a single pumpkin variety can be planted in multiple regions.
- Changed the Creates relationship between Marketing Plan and Special Events to a one-to-many relationship with a total participation constraint on the Special Events side. This reflects the real-world scenario where a marketing plan typically encompasses and promotes multiple special events. By making this change, we highlight that a single marketing plan can be associated with numerous special events, while each special event is uniquely tied to one marketing plan.



## Schema

For the following schemas below, attributes which are underlined denote primary key attributes, and attributes which are bolded denote foreign key attributes:

- Pumpkin Patch: (PatchID: int, PatchOwnership: varchar, PatchSize: int, PatchAddress: varchar, PatchName: varchar NOT NULL)
- Patch Map (MapID: int, **PatchID**: int NOT NULL, Paths: varchar, MapDescription: varchar)
- Map Region (RegionID: int, **MapID**: int, RegionSize: int, **VarietyID**: int)
- Pumpkin Variety (VarietyID: int, QuantityPlanted: int, PlantedDate: date, HarvestDate: date, VarietyName: varchar UNIQUE)
- Patch Tracks Variety(**PatchID**: int, **VarietyID**: int)
- EquipmentLog (LogID: int, LastMaintenanceDate: date, NextMaintenanceDate: date, EquipmentCount: int, **PatchID**: int)
- Marketing Plan (PlanDescription: varchar, SocialMediaRecords: varchar, AdvertisingRecords: varchar, PlanName: varchar, **PatchID**: int)
- Special Event(EventID: int, EventName: varchar NOT NULL, **PlanName**: varchar NOT NULL)
- Tickets(**EventID**: int, TicketID: int, AdmissionType: varchar, VisitorType: varchar)
- Activities(Duration: int, ActivityID: int, Fee: int, ActivityDescription: varchar, ActivityName: varchar NOT NULL, **PatchID**: int)
- Kids Activities(**ActivityID**: int, Guardian Requirement: int)
- Adult Activities(**ActivityID**: int, Age Requirement: int, Alcohol Involvement: int)

## Functional Dependencies (FDs)

- Pumpkin Patch
  - PatchID → PatchAddress, PatchName, PatchOwnership, PatchSize
- Patch Map
  - MapID → Paths, MapDescription
  - RegionID, MapID → RegionSize
- Pumpkin Variety
  - VarietyID → VarietyName, PlantedDate, HarvestDate, QuantityPlanted
  - PlantedDate, VarietyName → HarvestDate
- Equipment Log
  - LogID → EquipmentCount, LastMaintenanceDate, NextMaintenanceDate
  - LastMaintenanceDate → NextMaintenanceDate
- Activities
  - ActivityID → ActivityName, Duration, Fee, ActivityDescription, GuardianRequirement, AgeRequirement, AlcoholInvolvement
- Marketing Plan
  - PlanName → PlanDescription, SocialMediaRecords, AdvertisingRecords
- Special Events

- EventID → EventName, PlanName
- EventID, TicketID → AdmissionType, VisitorType

## Normalization

### Pumpkin Patch:

- Table before normalization:
  - Pumpkin Patch: (PatchID: int, PatchOwnership: varchar, PatchSize: int, PatchAddress: varchar, PatchName: varchar NOT NULL UNIQUE)
- Relevant FD's & Normalization:
  - PatchID → PatchAddress, PatchName, PatchOwnership, PatchSize
    - PatchID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Pumpkin Patch: (PatchID: int, PatchOwnership: varchar, PatchSize: int, PatchAddress: varchar, PatchName: varchar NOT NULL)

### Patch Map:

- Table before normalization:
  - Patch Map (MapID: int, **PatchID**: int NOT NULL, Paths: varchar, MapDescription: varchar)
- Relevant FD's & Normalization:
  - MapID → Paths, MapDescription
    - MapID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Patch Map (MapID: int, **PatchID**: int NOT NULL, Paths: varchar, MapDescription: varchar)

### Map Region:

- Table before normalization:
  - Map Region (RegionID: int, MapID: int, RegionSize: int, **VarietyID**: int)
- Relevant FD's & Normalization:
  - RegionID, MapID → RegionSize
    - RegionID + MapID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Map Region (RegionID: int, MapID: int, RegionSize: int, **VarietyID**: int)

### Pumpkin Variety

- Table before normalization:
  - Pumpkin Variety (VarietyID: int, QuantityPlanted: int, PlantedDate: date, HarvestDate: date, VarietyName: varchar)
- Relevant FD's & Normalization:
  - VarietyID → VarietyName, PlantedDate, HarvestDate, QuantityPlanted

- VarietyID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
  - PlantedDate, VarietyName → HarvestDate
    - PlantedDate + VarietyName is NOT a super key. Therefore this FD violates BCNF and we need to decompose it.
    - (VarietyID, QuantityPlanted (PlantedDate, VarietyName) HarvestDate)
    - R1(PlantedDate, VarietyName, HarvestDate)
      - The only FD relevant in this table is:
        - PlantedDate, VarietyName → HarvestDate
      - PlantedDate + VarietyName is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
      - We will name this table “Harvest Schedule”
    - R2(VarietyID, QuantityPlanted, PlantedDate, VarietyName)
      - The only FD relevant in this table is:
        - VarietyID → VarietyName, PlantedDate, QuantityPlanted
      - VarietyID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
      - We will name this table “Pumpkin Variety”
- Tables after normalization:
  - Pumpkin Variety (VarietyID: int, QuantityPlanted: int, PlantedDate: date, VarietyName: varchar)
  - Harvest Schedule (PlantedDate: date, VarietyName: varchar, HarvestDate: date)

#### Patch Tracks Variety:

- Table before normalization:
  - Patch Tracks Variety(**PatchID**: int, VarietyID: int)
- Relevant FD's & Normalization:
  - There are no relevant FD's for this table. Therefore it does not violate BCNF.
- Table after normalization:
  - Patch Tracks Variety(**PatchID**: int, VarietyID: int)

#### Equipment Log:

- Table before normalization:
  - Equipment Log (LogID: int, LastMaintenanceDate: date, NextMaintenanceDate: date, EquipmentCount: int, **PatchID**: int)
- Relevant FD's & Normalization:
  - LogID → EquipmentCount, LastMaintenanceDate, NextMaintenanceDate
    - LogID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
  - LastMaintenanceDate → NextMaintenanceDate
    - LastMaintenanceDate is NOT a super key. Therefore this FD violates BCNF and we need to decompose it.

- (LogID, EquipmentCount, PatchID (LastMaintenanceDate) NextMaintenanceDate)
- R1(LastMaintenanceDate, NextMaintenanceDate)
  - Tables with 2 attributes are automatically BCNF.
  - We will name this table "Maintenance Schedule"
- R2(LogID, EquipmentCount, PatchID, LastMaintenanceDate)
  - The only FD relevant in this table is:
    - LogID → EquipmentCount, LastMaintenanceDate
  - LogID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
  - We will name this table "Equipment Log"
- Table after normalization:
  - Maintenance Schedule (LastMaintenanceDate: date, NextMaintenanceDate: date)
  - Equipment Log (LogID: int, LastMaintenanceDate: date, EquipmentCount: int, **PatchID**: int)

#### Marketing Plan:

- Table before normalization:
  - Marketing Plan (PlanDescription: varchar, SocialMediaRecords: varchar, AdvertisingRecords: varchar, PlanName: varchar, **PatchID**: int)
- Relevant FD's & Normalization:
  - PlanName → PlanDescription, SocialMediaRecords, AdvertisingRecords
    - PlanName is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Marketing Plan (PlanDescription: varchar, SocialMediaRecords: varchar, AdvertisingRecords: varchar, PlanName: varchar, **PatchID**: int)

#### Special Event:

- Table before normalization:
  - Special Event(EventID: int, EventName: varchar NOT NULL, **PlanName**: varchar NOT NULL)
- Relevant FD's & Normalization:
  - EventID → EventName, PlanName
    - EventID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Special Event(EventID: int, EventName: varchar NOT NULL, **PlanName**: varchar NOT NULL)

#### Tickets:

- Table before normalization:
  - Tickets(**EventID**: int, TicketID: int, AdmissionType: varchar, VisitorType: varchar)
- Relevant FD's & Normalization:
  - EventID, TicketID → AdmissionType, VisitorType
    - EventID + TicketID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Tickets(**EventID**: int, TicketID: int, AdmissionType: varchar, VisitorType: varchar)

#### Activities:

- Table before normalization:
  - Activities(Duration: int, ActivityID: int, Fee: int, ActivityDescription: varchar, ActivityName: varchar NOT NULL, **PatchID**: int)
- Relevant FD's & Normalization:
  - ActivityID → ActivityName, Duration, Fee, ActivityDescription
    - ActivityID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Activities(Duration: int, ActivityID: int, Fee: int, ActivityDescription: varchar, ActivityName: varchar NOT NULL, **PatchID**: int)

#### Kids Activities:

- Table before normalization:
  - Kids Activities(**ActivityID**: int, Guardian Requirement: int)
- Relevant FD's & Normalization:
  - ActivityID → GuardianRequirement
    - ActivityID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Kids Activities(**ActivityID**: int, Guardian Requirement: int)

#### Adult Activities:

- Table before normalization:
  - Adult Activities(**ActivityID**: int, Age Requirement: int, Alcohol Involvement: int)
- Relevant FD's & Normalization:
  - ActivityID → AgeRequirement, AlcoholInvolvement
    - ActivityID is the primary key in this table. Therefore it is a super key. Therefore this FD does not violate BCNF.
- Table after normalization:
  - Adult Activities(**ActivityID**: int, Age Requirement: int, Alcohol Involvement: int)



## SQL DDL statements

```
CREATE TABLE PumpkinPatch (  
    PatchID INTEGER PRIMARY KEY,  
    PatchOwnership VARCHAR2(255),  
    PatchSize INTEGER,  
    PatchAddress VARCHAR2(255),  
    PatchName VARCHAR2(255) NOT NULL UNIQUE  
);
```

```
CREATE TABLE PatchMap (  
    MapID INTEGER PRIMARY KEY,  
    PatchID INTEGER NOT NULL,  
    Paths VARCHAR2(255),  
    MapDescription VARCHAR2(255),  
    FOREIGN KEY (PatchID) REFERENCES PumpkinPatch(PatchID) ON DELETE CASCADE  
);
```

```
CREATE TABLE MapRegion (  
    RegionID INTEGER PRIMARY KEY,  
    MapID INTEGER,  
    RegionSize INTEGER,  
    VarietyID INTEGER,  
    FOREIGN KEY (MapID) REFERENCES PatchMap(MapID) ON DELETE CASCADE,  
    FOREIGN KEY (VarietyID) REFERENCES PumpkinVariety(VarietyID)  
);
```

```
CREATE TABLE PumpkinVariety (  
    VarietyID INTEGER PRIMARY KEY,  
    QuantityPlanted INTEGER,  
    PlantedDate DATE,  
    VarietyName VARCHAR2(255) UNIQUE  
);
```

```
CREATE TABLE HarvestSchedule (  
    PlantedDate DATE,  
    VarietyName VARCHAR2(255),  
    HarvestDate DATE,  
    PRIMARY KEY (PlantedDate, VarietyName),  
    FOREIGN KEY (VarietyName) REFERENCES PumpkinVariety(VarietyName) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE PatchTracksVariety (  
    PatchID INTEGER,  
    VarietyID INTEGER,  
    PRIMARY KEY (PatchID, VarietyID),  
    FOREIGN KEY (PatchID) REFERENCES PumpkinPatch(PatchID) ON DELETE CASCADE,  
    FOREIGN KEY (VarietyID) REFERENCES PumpkinVariety(VarietyID) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE EquipmentLog (  
    LogID INTEGER PRIMARY KEY,  
    LastMaintenanceDate DATE,  
    EquipmentCount INTEGER,  
    PatchID INTEGER,  
    FOREIGN KEY (PatchID) REFERENCES PumpkinPatch(PatchID) ON DELETE CASCADE,  
    FOREIGN KEY (LastMaintenanceDate) REFERENCES  
MaintenanceSchedule(LastMaintenanceDate) ON DELETE CASCADE  
);
```

```
CREATE TABLE MaintenanceSchedule (  
    LastMaintenanceDate DATE PRIMARY KEY,  
    NextMaintenanceDate DATE  
);
```

```
CREATE TABLE MarketingPlan (  
    PlanName VARCHAR2(255) PRIMARY KEY,  
    PlanDescription VARCHAR2(255),  
    SocialMediaRecords VARCHAR2(255),  
    AdvertisingRecords VARCHAR2(255),  
    PatchID INTEGER,  
    FOREIGN KEY (PatchID) REFERENCES PumpkinPatch(PatchID) ON DELETE CASCADE  
);
```

```
CREATE TABLE SpecialEvent (  
    EventID INTEGER PRIMARY KEY,  
    EventName VARCHAR2(255) NOT NULL,  
    PlanName VARCHAR2(255) NOT NULL,  
    FOREIGN KEY (PlanName) REFERENCES MarketingPlan(PlanName) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE Tickets (  
    TicketID INTEGER PRIMARY KEY,  
    EventID INTEGER,  
    AdmissionType VARCHAR2(255),  
    VisitorType VARCHAR2(255),  
    FOREIGN KEY (EventID) REFERENCES SpecialEvent(EventID) ON DELETE CASCADE  
);
```

```
CREATE TABLE Activities (  
    ActivityID INTEGER PRIMARY KEY,  
    Duration INTEGER,  
    Fee INTEGER,  
    ActivityDescription VARCHAR2(1000),  
    ActivityName VARCHAR2(255) NOT NULL,  
    PatchID INTEGER,  
    FOREIGN KEY (PatchID) REFERENCES PumpkinPatch(PatchID) ON DELETE CASCADE  
);
```

```
CREATE TABLE KidsActivities (  
    ActivityID INTEGER PRIMARY KEY,  
    GuardianRequirement INTEGER,  
    FOREIGN KEY (ActivityID) REFERENCES Activities(ActivityID) ON DELETE CASCADE  
);
```

```
CREATE TABLE AdultActivities (  
    ActivityID INTEGER PRIMARY KEY,  
    AgeRequirement INTEGER,  
    AlcoholInvolvement INTEGER,  
    FOREIGN KEY (ActivityID) REFERENCES Activities(ActivityID) ON DELETE CASCADE  
);
```

## **INSERT statements**

### PumpkinPatch:

```
INSERT INTO PumpkinPatch (PatchID, PatchOwnership, PatchSize, PatchAddress, PatchName) VALUES (1, 'Owner1', 100, 'Address1', 'Sunny Farm');
INSERT INTO PumpkinPatch (PatchID, PatchOwnership, PatchSize, PatchAddress, PatchName) VALUES (2, 'Owner2', 200, 'Address2', 'Happy Farm');
INSERT INTO PumpkinPatch (PatchID, PatchOwnership, PatchSize, PatchAddress, PatchName) VALUES (3, 'Owner3', 150, 'Address3', 'Green Farm');
INSERT INTO PumpkinPatch (PatchID, PatchOwnership, PatchSize, PatchAddress, PatchName) VALUES (4, 'Owner4', 300, 'Address4', 'Organic Farm');
INSERT INTO PumpkinPatch (PatchID, PatchOwnership, PatchSize, PatchAddress, PatchName) VALUES (5, 'Owner5', 250, 'Address5', 'Country Farm');
```

### PumpkinVariety:

```
INSERT INTO PumpkinVariety (VarietyID, QuantityPlanted, PlantedDate, VarietyName) VALUES (1, 50, '2023-04-01', 'Big Max');
INSERT INTO PumpkinVariety (VarietyID, QuantityPlanted, PlantedDate, VarietyName) VALUES (2, 75, '2023-04-15', 'Sugar Pie');
INSERT INTO PumpkinVariety (VarietyID, QuantityPlanted, PlantedDate, VarietyName) VALUES (3, 65, '2023-04-20', 'Cinderella');
INSERT INTO PumpkinVariety (VarietyID, QuantityPlanted, PlantedDate, VarietyName) VALUES (4, 80, '2023-05-01', 'Jarrahdale');
INSERT INTO PumpkinVariety (VarietyID, QuantityPlanted, PlantedDate, VarietyName) VALUES (5, 100, '2023-05-10', 'Kabocha');
```

### PatchMap:

```
INSERT INTO PatchMap (MapID, PatchID, Paths, MapDescription) VALUES (1, 1, 'Path1', 'Description1');
INSERT INTO PatchMap (MapID, PatchID, Paths, MapDescription) VALUES (2, 2, 'Path2', 'Description2');
INSERT INTO PatchMap (MapID, PatchID, Paths, MapDescription) VALUES (3, 3, 'Path3', 'Description3');
INSERT INTO PatchMap (MapID, PatchID, Paths, MapDescription) VALUES (4, 4, 'Path4', 'Description4');
INSERT INTO PatchMap (MapID, PatchID, Paths, MapDescription) VALUES (5, 5, 'Path5', 'Description5');
```

#### MapRegion:

```
INSERT INTO MapRegion (RegionID, MapID, RegionSize, VarietyID) VALUES (1, 1, 20, 1);
INSERT INTO MapRegion (RegionID, MapID, RegionSize, VarietyID) VALUES (2, 2, 25, 2);
INSERT INTO MapRegion (RegionID, MapID, RegionSize, VarietyID) VALUES (3, 3, 30, 3);
INSERT INTO MapRegion (RegionID, MapID, RegionSize, VarietyID) VALUES (4, 4, 35, 4);
INSERT INTO MapRegion (RegionID, MapID, RegionSize, VarietyID) VALUES (5, 5, 40, 5);
```

#### HarvestSchedule:

```
INSERT INTO HarvestSchedule (PlantedDate, VarietyName, HarvestDate) VALUES
('2023-04-01', 'Big Max', '2023-10-01');
INSERT INTO HarvestSchedule (PlantedDate, VarietyName, HarvestDate) VALUES
('2023-04-15', 'Sugar Pie', '2023-10-15');
INSERT INTO HarvestSchedule (PlantedDate, VarietyName, HarvestDate) VALUES
('2023-04-20', 'Cinderella', '2023-10-20');
INSERT INTO HarvestSchedule (PlantedDate, VarietyName, HarvestDate) VALUES
('2023-05-01', 'Jarrahdale', '2023-10-25');
INSERT INTO HarvestSchedule (PlantedDate, VarietyName, HarvestDate) VALUES
('2023-05-10', 'Kabocha', '2023-10-30');
```

#### PatchTracksVariety:

```
INSERT INTO PatchTracksVariety (PatchID, VarietyID) VALUES (1, 1);
INSERT INTO PatchTracksVariety (PatchID, VarietyID) VALUES (2, 2);
INSERT INTO PatchTracksVariety (PatchID, VarietyID) VALUES (3, 3);
INSERT INTO PatchTracksVariety (PatchID, VarietyID) VALUES (4, 4);
INSERT INTO PatchTracksVariety (PatchID, VarietyID) VALUES (5, 5);
```

#### MaintenanceSchedule:

```
INSERT INTO MaintenanceSchedule (LastMaintenanceDate, NextMaintenanceDate) VALUES
('2023-03-01', '2023-09-01');
INSERT INTO MaintenanceSchedule (LastMaintenanceDate, NextMaintenanceDate) VALUES
('2023-03-15', '2023-09-15');
INSERT INTO MaintenanceSchedule (LastMaintenanceDate, NextMaintenanceDate) VALUES
('2023-04-01', '2023-10-01');
INSERT INTO MaintenanceSchedule (LastMaintenanceDate, NextMaintenanceDate) VALUES
('2023-04-15', '2023-10-15');
INSERT INTO MaintenanceSchedule (LastMaintenanceDate, NextMaintenanceDate) VALUES
('2023-05-01', '2023-11-01');
```

#### EquipmentLog:

```
INSERT INTO EquipmentLog (LogID, LastMaintenanceDate, EquipmentCount, PatchID)
VALUES (1, '2023-03-01', 5, 1);
INSERT INTO EquipmentLog (LogID, LastMaintenanceDate, EquipmentCount, PatchID)
VALUES (2, '2023-03-15', 7, 2);
INSERT INTO EquipmentLog (LogID, LastMaintenanceDate, EquipmentCount, PatchID)
VALUES (3, '2023-04-01', 6, 3);
INSERT INTO EquipmentLog (LogID, LastMaintenanceDate, EquipmentCount, PatchID)
VALUES (4, '2023-04-15', 8, 4);
INSERT INTO EquipmentLog (LogID, LastMaintenanceDate, EquipmentCount, PatchID)
VALUES (5, '2023-05-01', 9, 5);
```

#### MarketingPlan:

```
INSERT INTO MarketingPlan (PlanName, PlanDescription, SocialMediaRecords,
AdvertisingRecords, PatchID) VALUES ('Plan1', 'Description1', 'Record1', 'AdRecord1', 1);
INSERT INTO MarketingPlan (PlanName, PlanDescription, SocialMediaRecords,
AdvertisingRecords, PatchID) VALUES ('Plan2', 'Description2', 'Record2', 'AdRecord2', 2);
INSERT INTO MarketingPlan (PlanName, PlanDescription, SocialMediaRecords,
AdvertisingRecords, PatchID) VALUES ('Plan3', 'Description3', 'Record3', 'AdRecord3', 3);
INSERT INTO MarketingPlan (PlanName, PlanDescription, SocialMediaRecords,
AdvertisingRecords, PatchID) VALUES ('Plan4', 'Description4', 'Record4', 'AdRecord4', 4);
INSERT INTO MarketingPlan (PlanName, PlanDescription, SocialMediaRecords,
AdvertisingRecords, PatchID) VALUES ('Plan5', 'Description5', 'Record5', 'AdRecord5', 5);
```

#### SpecialEvent:

```
INSERT INTO SpecialEvent (EventID, EventName, PlanName) VALUES (1, 'Halloween Bash',
'Plan1');
INSERT INTO SpecialEvent (EventID, EventName, PlanName) VALUES (2, 'Harvest Festival',
'Plan2');
INSERT INTO SpecialEvent (EventID, EventName, PlanName) VALUES (3, 'Pumpkin Carving
Contest', 'Plan3');
INSERT INTO SpecialEvent (EventID, EventName, PlanName) VALUES (4, 'Thanksgiving
Sale', 'Plan4');
INSERT INTO SpecialEvent (EventID, EventName, PlanName) VALUES (5, 'Christmas Party',
'Plan5');
```

### Tickets:

```
INSERT INTO Tickets (TicketID, EventID, AdmissionType, VisitorType) VALUES (1, 1, 'Standard', 'Adult');
INSERT INTO Tickets (TicketID, EventID, AdmissionType, VisitorType) VALUES (2, 2, 'Premium', 'Child');
INSERT INTO Tickets (TicketID, EventID, AdmissionType, VisitorType) VALUES (3, 3, 'Standard', 'Senior');
INSERT INTO Tickets (TicketID, EventID, AdmissionType, VisitorType) VALUES (4, 4, 'Standard', 'Adult');
INSERT INTO Tickets (TicketID, EventID, AdmissionType, VisitorType) VALUES (5, 5, 'Premium', 'Adult');
```

### Activities:

```
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (1, 30, 5, 'Pumpkin carving', 'Carving Class', 1);
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (2, 60, 10, 'Pumpkin pie making', 'Cooking Class', 2);
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (3, 45, 7, 'Hayrides around the patch', 'Hayride', 3);
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (4, 120, 15, 'Halloween movie night', 'Movie Night', 4);
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (5, 90, 12, 'Pumpkin painting', 'Painting Class', 5);
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (6, 90, 20, 'Youth art class', 'Artistic Wonderland', 1);
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (7, 120, 50, 'Wine tasting event', 'Wine & Unwind', 1);
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (8, 60, 10, 'Cultural dance showcase', 'Dance Fiesta', 1);
INSERT INTO Activities (ActivityID, Duration, Fee, ActivityDescription, ActivityName, PatchID)
VALUES (9, 150, 60, 'Cooking class for adults', 'Culinary Masters', 1);
```

### KidsActivities:

```
INSERT INTO KidsActivities (ActivityID, GuardianRequirement) VALUES (1, 1);
INSERT INTO KidsActivities (ActivityID, GuardianRequirement) VALUES (3, 1);
INSERT INTO KidsActivities (ActivityID, GuardianRequirement) VALUES (5, 1);
INSERT INTO KidsActivities (ActivityID, GuardianRequirement) VALUES (6, 0);
INSERT INTO KidsActivities (ActivityID, GuardianRequirement) VALUES (7, 2);
INSERT INTO KidsActivities (ActivityID, GuardianRequirement) VALUES (8, 1);
INSERT INTO KidsActivities (ActivityID, GuardianRequirement) VALUES (9, 0);
```

AdultActivities:

```
INSERT INTO AdultActivities (ActivityID, AgeRequirement, AlcoholInvolvement) VALUES (2, 21, 0);
```

```
INSERT INTO AdultActivities (ActivityID, AgeRequirement, AlcoholInvolvement) VALUES (4, 21, 1);
```

```
INSERT INTO AdultActivities (ActivityID, AgeRequirement, AlcoholInvolvement) VALUES (6, 18, 0);
```

```
INSERT INTO AdultActivities (ActivityID, AgeRequirement, AlcoholInvolvement) VALUES (7, 25, 1);
```

```
INSERT INTO AdultActivities (ActivityID, AgeRequirement, AlcoholInvolvement) VALUES (8, 21, 0);
```

```
INSERT INTO AdultActivities (ActivityID, AgeRequirement, AlcoholInvolvement) VALUES (9, 25, 1);
```