CH_3

OH

H_3C

H_3C

O

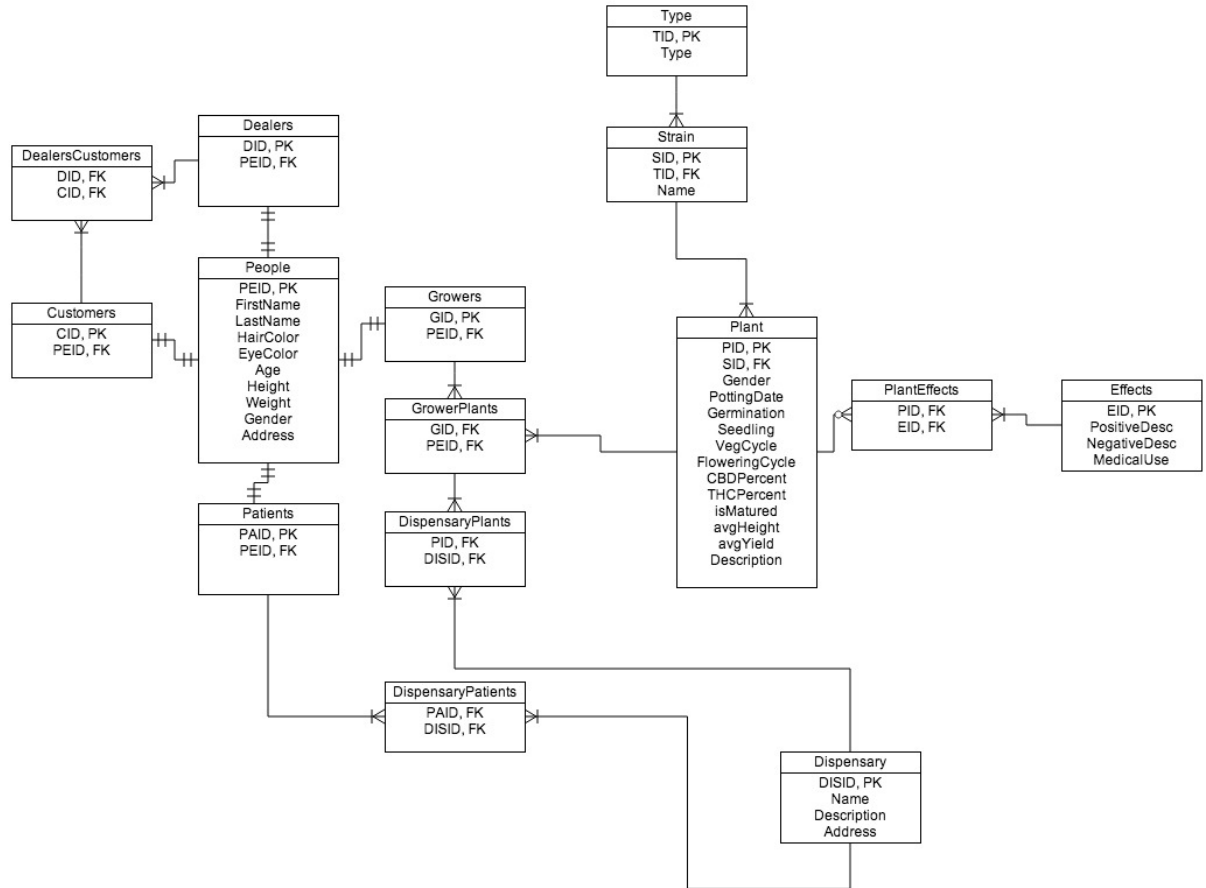$CH_2(CH_2)_3CH_3$

Anthony Santostefano

Table of Contents

Executive Summary

This document is a small look inside the complex world of the legal, and illegal, sale, distribution, and growth of marijuana. It is meant to show many things, but primarily the growth cycles of the individual strains, their affects on the human brain, and an average look into their yields. Even though data is relative, it is never fake or meant to be misleading, all data is based upon real numbers whether it be street, dispensary or research level of the industry.

Firstly the Entity Relationship Diagram is shown in order to display the tables of the database, and how closely related the data is in actuality. Next, their SQL scripts written to build on Postgres, these will include the framework of the database, its sample data, views, triggers and stored procedures. Lastly, the security of the data is viewable and explained.

ER Diagram

Type
TID, PK
Type

Strain
SID, PK
TID, FK
Name

DealersCustomers
DID, FK
CID, FK

Dealers
DID, PK
PEID, FK

Customers
CID, PK
PEID, FK

People
PEID, PK
FirstName
LastName
HairColor
EyeColor
Age
Height
Weight
Gender
Address

Growers
GID, PK
PEID, FK

GrowerPlants
GID, FK
PEID, FK

Patients
PAID, PK
PEID, FK

DispensaryPlants
PID, FK
DISID, FK

Plant
PID, PK
SID, FK
Gender
PottingDate
Germination
Seedling
VegCycle
FloweringCycle
CBDPercent
THCPercent
isMatured
avgHeight
avgYield
Description

PlantEffects
PID, FK
EID, FK

Effects
EID, PK
PositiveDesc
NegativeDesc
MedicalUse

DispensaryPatients
PAID, FK
DISID, FK

Dispensary
DISID, PK
Name
Description
Address

Type Table

This table contains the marijuana strain type, which can be Indica, Sativa, or Hybrid.

Functional Dependency:

  tid => type

Create Table Statement:

```
CREATE TABLE Type
(
    tid INTEGER NOT NULL PRIMARY KEY,
    type VARCHAR(6) UNIQUE check(type ='Indica' OR type = 'Sativa' OR type ='Hybrid')
);
```

Sample Data:

	tid integer	type character varying(6)
1	1	Indica
2	2	Sativa
3	3	Hybrid

Strains Table

This table contains the strains and their names.

Functional Dependency:

      sid => tid, name

Create Table Statement:

```sql
CREATE TABLE Strains
(
    sid INTEGER NOT NULL PRIMARY KEY,
    tid INTEGER NOT NULL,
    name TEXT, -- The name of the plant.
    FOREIGN KEY (tid) REFERENCES Type(tid)
);
```

Sample Data:

	sid integer	tid integer	name text
1	1	1	Afghan Kush
2	2	1	Northern Lights
3	3	1	Bubba Kush
4	4	1	LA Confidential
5	5	1	Skywalker
6	6	1	G13
7	7	1	Death Star
8	8	1	Granddaddy Purple
9	9	1	Strawberry Kush
10	10	1	Obama Kush
11	11	2	Sour Diesel
12	12	2	Purple Power
13	13	2	Cannalope Haze
14	14	2	Candy Jack
15	15	2	Ultimate Trainwreck
16	16	2	Voodoo
17	17	2	Very Berry Haze
18	18	2	Mango Dream
19	19	2	OG
20	20	2	AK 47
21	21	3	Barneys Tangerine Dream
22	22	3	Sour Diesel
23	23	3	Super Silver Haze
24	24	3	Super Lemon Haze
25	25	3	White Widow
26	26	3	Trainwreck Kush Haze
27	27	3	Blue Dream
28	28	3	Atomic Northern Lights
29	29	3	Skywalker OG
30	30	3	Pineapple Express

Plants Table

The Plants table contains a list of the plants and their characteristics. It contains information about their gender, their potting date (the date they were planted), the amount of time until germination, the time they spend (in days) as a seedling, the time they spend during their vegetative stage where they do most of their initial growing measured in days, their flowering stage where they may double in height and begin their budding process measured in days, their CBD or cannabinoid percentage the chemical is found mostly to have an effect physically, THC or tetrahydrocannabinol percentage the chemical primarily effects the psyche, if they are fully matured or developed, their average height measure in inches, their average yield measured in grams, and a brief description about the plant.

Functional Dependency:

pid => sid, gender, pottingdate, germination, seedling, vegcycle, floweringcycle, cbdpercent, thcpercent, ismatured, avgheight, description

Create Table Statement:

```
CREATE TABLE Plants
(
    pid INTEGER NOT NULL PRIMARY KEY,
    sid INTEGER NOT NULL,
    gender VARCHAR(6), -- The plants gender either male, female, or hermaphrodite. (IMPORTANT: Test data is only in the female gender
see executive                        summary for more details)
    pottingdate DATE,   -- The date it was planted.
    germination INTEGER,   -- In days how long it will take the seed to sprout.
    seedling INTEGER,   -- In days how long it the plant will stay in it's seedling stage.
    vegcycle INTEGER,   -- In days how long the vegatation cycle will be.
    floweringcycle INTEGER, -- In days how long the flowering cycle will be.
    cbdpercent DECIMAL,     -- The percentage of cannabinoids.
    thcpercent DECIMAL,     -- The percentage of tetrahydrocannabinol (THC) the pyschoactive ingredient.
    ismatured BOOLEAN default(false), -- DEFAULT: False, True when the plant is fully matured.
    avgheight INTEGER, -- The average height of the plant in inches.
    avgyield INTEGER, -- The average yield in grams.
    description TEXT,
    FOREIGN KEY (sid) REFERENCES Strains(sid)
);
```

Sample Data:

	sid integer	name text	type character varying(6)	gender character varying(6)	pottingdate date	germination integer	seedling integer	vegcycle integer	floweringcycle integer	cbdpercent numeric	thcpercent numeric	ismatured boolean	avgheight integer	avgyield integer	description text
1	1	Afghan Kush	Indica	Female	2014-04-2(	3	28	45	51	0.25	15.13	f	60	500	The Afgha
2	20	AK 47	Sativa	Female	2014-04-2(	3	15	25	31	0.22	21.15	f	37	120	The AK-47
3	25	White Widow	Hybrid	Female	2014-04-2(	3	10	45	60	0.13	15.67	f	74	425	The White
4	26	Trainwreck Kush Haze	Hybrid	Female	2014-04-2(	3	10	50	65	0.22	18.13	f	80	350	The Train
5	27	Blue Dream	Hybrid	Female	2014-04-2(	3	15	45	70	0.41	16.62	f	60	600	The Blue
6	11	Sour Diesel	Sativa	Female	2014-04-2(	3	10	30	85	0.85	53.02	f	72	500	The Sour
7	17	Very Berry Haze	Sativa	Female	2014-04-2(	3	10	40	67	20.00	17.00	f	59	700	The Very
8	12	Purple Power	Sativa	Female	2014-04-2(	3	10	25	56	0.00	20.00	f	102	550	The Purpl
9	10	Obama Kush	Indica	Female	2014-04-2(	3	10	30	60	0.10	23.00	f	65	450	The Obama
10	18	Mango Dream	Sativa	Female	2014-04-2(	3	15	45	65	0.50	17.00	f	65	500	The Mango
11	2	Northern Lights	Indica	Female	2014-04-2(	3	10	45	50	0.18	11.51	f	98	575	The North
12	15	Ultimate Trainwreck	Sativa	Female	2014-04-2(	3	10	40	56	0.77	21.74	f	80	600	The Ultim
13	13	Cannalope Haze	Sativa	Female	2014-04-2(	3	10	25	63	0.33	20.33	f	40	350	The Canna
14	21	Barneys Tangerine Dream	Hybrid	Female	2014-04-2(	3	10	35	70	1.80	25.00	f	40	500	The Barne
15	5	Skywalker	Indica	Female	2014-04-2(	3	10	30	72	0.20	12.00	f	60	400	The Skywa
16	19	OG	Sativa	Female	2014-04-2(	3	10	45	56	0.70	20.00	f	80	525	The OG Pl
17	24	Super Lemon Haze	Hybrid	Female	2014-04-2(	3	10	50	75	0.21	19.33	f	74	800	The Super
18	8	Granddaddy Purple	Indica	Female	2014-04-2(	3	20	40	54	0.06	17.56	f	84	770	The Grand
19	28	Atomic Northern Lights	Hybrid	Female	2014-04-2(	3	10	20	50	0.36	11.10	f	90	500	The Atomi
20	30	Pineapple Express	Hybrid	Female	2014-04-2(	3	10	20	56	0.35	18.13	f	78	425	The Pinea
21	6	G13	Indica	Female	2014-04-2(	3	10	45	60	0.25	14.97	f	60	600	The G13 P
21	6	G13	Indica	Female	2014-04-2(	3	10	45	60	0.25	14.97	f	60	600	The G13 P
22	29	Skywalker OG	Hybrid	Female	2014-04-2(	3	10	30	58	0.13	20.24	f	74	550	The Skywa
23	16	Voodoo	Sativa	Female	2014-04-2(	3	10	25	56	0.15	23.92	f	72	625	The Voodo
24	4	LA Confidential	Indica	Female	2014-04-2(	3	10	22	54	0.48	17.72	f	55	375	The LA Co
25	23	Super Silver Haze	Hybrid	Female	2014-04-2(	3	10	45	77	0.41	19.25	f	68	750	The Super
26	22	Sour Diesel	Hybrid	Female	2014-04-2(	3	10	30	89	0.85	53.02	f	80	600	The O.D S
27	3	Bubba Kush	Indica	Female	2014-04-2(	3	10	40	63	0.26	18.75	f	48	800	The Bubba
28	14	Candy Jack	Sativa	Female	2014-04-2(	3	10	20	60	0.10	16.89	f	52	500	The Candy
29	9	Strawberry Kush	Indica	Female	2014-04-2(	3	10	35	64	0.40	20.30	f	92	1000	The Straw
30	7	Death Star	Indica	Female	2014-04-2(	3	10	30	65	0.43	18.00	f	80	525	The Death

Effects Table

The effects table contains the possible effects after user, and the common medical uses.

Functional Dependency:

eid => PostiveDesc, NegativeDesc, MedicalUse

Create Table Statement:

```
CREATE TABLE Effects
(
    eid INTEGER NOT NULL PRIMARY KEY,
    PositiveDesc text,
    NegativeDesc text,
    MedicalUse text
);
```

Sample Data:

	eid integer	positivedesc text	negativedesc text	medicaluse text
1	1	Euphoric, Happy, Relaxed, Sleepy, Uplifted	Dizzy, Dry Mouth, Dry Eyes, Headache, Parano	Anxiety, Migraines, ADD/ADHD, PMS, Arthritis
2	2	Creative, Energetic, Euphoric, Happy, and Up	Dizzy, Dry Mouth, Dry Eyes, Headache, Parano	Anxiety, Migraines, ADD/ADHD, Bipolar Disord
3	3	Creative, Energetic, Euphoric, Happy, and Up	Dizzy, Dry Mouth, Dry Eyes, Headache, Parano	Anxiety, Migraines, ADD/ADHD, Bipolar Disord

Plant Effects Table

The Plant Effects table is the join table between the plants and their effects.

Functional Dependency:

pid, eid =>

Create Table Statement:

```
CREATE TABLE PlantEffects
(
    pid INTEGER,
    eid INTEGER,
    FOREIGN KEY (pid) REFERENCES Plants(pid),
    FOREIGN KEY (eid) REFERENCES Effects(eid),
    PRIMARY KEY (pid, eid)
);
```

Sample Data:

	pid integer	eid integer
1	1	1
2	11	1
3	27	1
4	24	1
5	15	1
6	21	1
7	30	1
8	18	1
9	29	1
10	9	1
11	6	2
12	8	2
13	13	2
14	28	2
15	12	2
16	23	2
17	7	2
18	10	2
19	16	2
20	2	2
21	14	3
22	26	3
23	25	3
24	17	3
25	3	3
26	4	3
27	5	3
28	19	3
29	22	3
30	20	3

People Table

The people table contains information about the people entered into the database.

Functional Dependency:

peid => firstname, lastname, age, haircolor, eyecolor, height, weight, gender, address

Create Table Statement:

```
CREATE TABLE People
(
    peid INTEGER NOT NULL PRIMARY KEY,
    firstname VARCHAR(20),
    lastname VARCHAR(30),
    age INTEGER,
    haircolor VARCHAR(10),
    eyecolor VARCHAR(10),
    height INTEGER, -- Height of the person in inches.
    weight INTEGER, -- Weight of the person in pounds.
    gender VARCHAR(6) check(gender ='Male' OR gender ='Female'),
    address TEXT
);
```

Example Data:

	peid integer	firstname character varying(20)	lastname character varying(30)	age integer	haircolor character varying(10)	eyecolor character varying(10)	height integer	weight integer	gender character varying(6)	address text
1	1	John	Doe	35	Black	Brown	64	150	Male	19 Star
2	2	Suzi	Queue	18	Blonde	Blue	62	125	Female	25 Star
3	3	Jessica	Morgan	23	Brunette	Brown	72	145	Female	19 High
4	4	Tony	McJackson	42	Blonde	Hazel	68	165	Male	29 Star
5	5	Danielle	McGrubersen	38	Red	Brown	70	145	Female	35 Left
6	6	Jessica	Jorgen	25	Brunette	Brown	72	145	Female	14 Mill
7	7	Daniel	Bates	22	Brunette	Blue	76	185	Male	25 Mair
8	8	Norman	Bates	19	Brunette	Brown	70	150	Male	25 Mair
9	9	Norma	Bates	35	Brunette	Blue	70	145	Female	25 Mair
10	10	Ryan	Pool	21	Brown	Blue	68	175	Male	35 West
11	11	Vincent	Stallman	32	Brown	Blue	78	178	Male	30 West
12	12	Erika	Manchest	27	Blonde	Blue	68	125	Female	92 Wind
13	13	Tracy	Detka	24	Blonde	Brown	65	105	Female	325 Wor
14	14	Nancy	Wilcrest	47	Blonde	Blue	72	125	Female	22 Scar
15	15	Hayley	Consen	20	Black	Hazel	65	140	Female	24 Mair

Dispensaries Table

The Dispensaries table contains the name, address, and location of the dispensaries.

Functional Dependency:

disid => name, description, address

Create Table Statement:

```
CREATE TABLE Dispensaries
(
    disid INTEGER NOT NULL PRIMARY KEY,
    name VARCHAR(50),
    description TEXT,
    address TEXT
);
```

Example Date:

	disid integer	name character varying(50)	description text	address text
1	1	Capitola Healing Association	Great customer service and top qua	3088 Winkle Ave, Santa Cruz, California, 95065
2	2	Herbal Cruz	Great selection, and a large quant	1001 41st Ave, Santa Cruz, California, 95062
3	3	La Brea Compassionate Caregivers	Very friendly staff, and a well cl	735 N La Brea Ave, Los Angeles, California, 90038
4	4	Exhale Med Center	Super friendly associates, prices	980 N La Cienega Blvd, Los Angeles, California, 90069
5	5	The Giving Tree of Denver	Number 1 rated dispenary in Denver	2707 W 38th Ave, Denver, Colorado, 80211
6	6	Kind Love	Awesome selection of CBD rich buds	4380 E Alameda Ave Denver, Colardo, 80246

Growers Table

The Growers table inherits the primary key from people in order to display what people are also growers.

Functional Dependency:

gid => peid

Create Table Statement:

```
CREATE TABLE Growers
(
    gid INTEGER NOT NULL PRIMARY KEY,
    peid INTEGER,
    FOREIGN KEY (peid) REFERENCES People(peid)
);
```

Sample Data:

	gid integer	peid integer
1	1	10
2	2	2
3	3	4
4	4	5
5	5	9

Grower Plants Table

The Grower Plants table is the join table between Plants and Growers in order to show what plants are grown by what person.

Functional Dependency:

       gid, pid =>

Create Table Statement:

```
CREATE TABLE GrowerPlants
(
    gid INTEGER,
    pid INTEGER,
    FOREIGN KEY (gid) REFERENCES Growers(gid),
    FOREIGN KEY (pid) REFERENCES Plants(pid),
    PRIMARY KEY (gid, pid)
);
```

Example Data:

	gid integer	pid integer
1	1	23
2	1	2
3	1	20
4	1	13
5	1	4
6	1	1
7	2	26
8	2	25
9	2	5
10	2	21
11	2	30
12	3	19
13	3	17
14	3	22
15	3	7
16	3	4
17	3	28
18	3	9
19	4	29
20	4	20
21	4	15
22	4	12
23	4	24
24	4	8
25	5	6
26	5	14
27	5	15
28	5	18
29	5	3
30	5	10

Dispensary Plants Table

The Dispensary Plants table is the join table between Dispensaries and Plants to show which Plants are purchased by what dispensaries.

Function Dependency:

      pid, disid =>

Create Table Statement:

```
CREATE TABLE DispensaryPlants
(
    pid INTEGER,
    disid INTEGER,
    FOREIGN KEY (pid) REFERENCES Plants(pid),
    FOREIGN KEY (disid) REFERENCES Dispensaries(disid),
    PRIMARY KEY (pid, disid)
);
```

Sample Data:

	pid integer	disid integer
1	30	1
2	21	1
3	5	1
4	6	1
5	10	1
6	4	1
7	19	1
8	29	1
9	29	2
10	8	2
11	24	2
12	20	2
13	12	2
14	15	2

	pid integer	disid integer
15	4	2
16	19	2
17	3	3
18	19	3
19	8	3
20	9	3
21	14	3
22	18	3
23	6	3
24	29	3
25	30	4
26	21	4
27	5	4
28	25	4
29	14	4

28	25	4
29	14	4
30	15	4
31	23	4
32	29	4
33	3	5
34	30	5
35	2	5
36	15	5
37	25	5
38	17	5
39	20	5
40	19	5
41	3	6
42	18	6
43	6	6

44	15	6
45	7	6
46	22	6
47	8	6
48	24	6

15

Patients Table

The Patients table inherits the primary key from People in order to show what people are patients.

Functional Dependency:

      paid => peid

Create Table Statement:

```
CREATE TABLE DispensaryPatients
(
    paid INTEGER,
    disid INTEGER,
    FOREIGN KEY (paid) REFERENCES Patients(paid),
    FOREIGN KEY (disid) REFERENCES Dispensaries(disid),
    PRIMARY KEY (paid, disid)
);
```

Example Data:

	paid integer	peid integer
1	1	1
2	2	1
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9

Dispensary Patients Table

The Dispensary Patients table is the join table between Dispensaries and Patients to show which Patients belong to which Dispensaries.

Functional Dependency:

paid, disid =>

Create Table Statement:

```
CREATE TABLE DispensaryPatients
(
    paid INTEGER,
    disid INTEGER,
    FOREIGN KEY (paid) REFERENCES Patients(paid),
    FOREIGN KEY (disid) REFERENCES Dispensaries(disid),
    PRIMARY KEY (paid, disid)
);
```

Example Data:

	paid integer	disid integer
1	1	4
2	2	3
3	3	1
4	4	2
5	5	1
6	6	6
7	7	5
8	8	5
9	9	6

Dealers Table

The Dealers table inherits the primary key from People in order to show what people are Dealers.

Functional Dependency:

did => peid

Create Table Statement:

```
CREATE TABLE Dealers
(
    did INTEGER NOT NULL PRIMARY KEY,
    peid INTEGER,
    FOREIGN KEY (peid) REFERENCES People(peid)
);
```

Example Data:

	did integer	peid integer
1	1	15
2	2	11
3	3	10

Customers Table

The Customers table inherits the primary key from People in order to show what people are Customers.

Functional Dependency:

cid => peid

Create Table Statement:

```sql
CREATE TABLE Customers
(
    cid INTEGER NOT NULL PRIMARY KEY,
    peid INTEGER,
    FOREIGN KEY (peid) REFERENCES People(peid)
);
```

Sample Data:

	cid integer	peid integer
1	1	15
2	2	11
3	3	12
4	4	13
5	5	14

Dealers Customers Table

The Dealers Customers table is the join table between Dealers and Customers to show which Customers purchase from what Dealers.

Functional Dependency:

Create Table Statement:

```
CREATE TABLE DealersCustomers
(
    did INTEGER,
    cid INTEGER,
    FOREIGN KEY (did) REFERENCES Dealers(did),
    FOREIGN KEY (cid) REFERENCES Customers(cid),
    PRIMARY KEY (did, cid)
);
```

Example Data:

	did integer	cid integer
1	3	1
2	3	2
3	1	4
4	1	3
5	2	2
6	2	4
7	1	5

View – Marijuana Strains

The Marijuana Strains view is useful in order to show the strains types.

```sql
CREATE VIEW MarijuanaStrains AS
SELECT Strains.sid AS "Strain_ID",
       Type,
       Strains.name AS "Strain_Name"
FROM Strains FULL OUTER JOIN Type ON (Strains.tid = Type.tid)
ORDER BY Strains.sid ASC;

SELECT * FROM MarijuanaStrains WHERE Type ='Indica';
```

	Strain_ID integer	type character varying(6)	Strain_Name text
1	1	Indica	Afghan Kush
2	2	Indica	Northern Lights
3	3	Indica	Bubba Kush
4	4	Indica	LA Confidential
5	5	Indica	Skywalker
6	6	Indica	G13
7	7	Indica	Death Star
8	8	Indica	Granddaddy Purple
9	9	Indica	Strawberry Kush
10	10	Indica	Obama Kush

View – Marijuana Full Details

A full details view would be useful to a dispensary who wanted to know more about the

plant or a grower interested in growing.

```sql
CREATE VIEW MarijuanaFullDetails AS
SELECT Plants.pid AS "Plant ID",
       Strains.sid AS "Strain ID",
       name,
       type,
       gender,
       pottingdate AS "Date Planted",
       germination AS "Germination Days",
       seedling AS "Days as Seedling",
       vegcycle AS "Vegetative Phase Days",
       floweringcycle AS "Flowering Cycle Days",
       cbdpercent,
       thcpercent,
       ismatured AS "Ready",
       avgheight AS "Average Height in.",
       avgyield AS "Average Yield gr",
       description
  FROM Plants FULL OUTER JOIN Strains ON (Plants.sid = Strains.sid)
  INNER JOIN Type ON (Strains.tid = Type.tid)
  ORDER BY Plants.pid ASC;

SELECT * FROM MarijuanaFullDetails WHERE cbdpercent > 0.56;
```

	Plant ID integer	Strain ID integer	name text	type character varying(6)	gender character varying(6)	Date Planted date	Germination integer	Days as Seedling integer	Vegetative Pha integer	Flowering Cycle Days integer	cbdpercent numeric	thcpercent numeric	Ready boolean	Average He integer	Average Yi integer	description text
1	6	11	Sour	Sativa	Female	2014-04-20	3	10	30	85	0.85	53.02	f	72	500	The Sour
2	7	17	Very	Sativa	Female	2014-04-20	3	10	40	67	20.00	17.00	f	59	700	The Very
3	12	15	Ultin	Sativa	Female	2014-04-20	3	10	40	56	0.77	21.74	f	80	600	The Ultin
4	14	21	Barne	Hybrid	Female	2014-04-20	3	10	35	70	1.80	25.00	f	40	500	The Barne
5	16	19	OG	Sativa	Female	2014-04-20	3	10	45	56	0.70	20.00	f	80	525	The OG PT
6	26	22	Sour	Hybrid	Female	2014-04-20	3	10	30	89	0.85	53.02	f	80	600	The O.D S

View – Plant Strain Effects

The Plant Strain Effects view would be useful for purchaser before purchasing.

```
CREATE VIEW PlantStrainEffects AS
SELECT name,
       Description,
       PositiveDesc AS "Positive Characteristics",
       NegativeDesc AS "Negative Chataceristics",
       MedicalUse AS "Medical Uses"
FROM Strains FULL OUTER JOIN Plants ON (Plants.sid = Strains.sid)
INNER JOIN PlantEffects ON (Plants.pid = PlantEffects.pid)
INNER JOIN Effects ON (PlantEffects.eid = Effects.eid);

SELECT * FROM PlantStrainEffects WHERE name = 'Granddaddy Purple';
```

	name text	description text	Positive Characteristics text	Negative Chataceristi text	Medical Uses text
1	Granddaddy Purple	The Granddaddy Purple Plant k	Euphoric, Happy, Relaxed, Sleepy,	Dizzy, Dry Mouth,	Anxiety, Migraines, ADD/ADHD, PMS, Arthritis

View – Dispensary Strains

The Dispensary Strains view would be useful until an actual inventory view is developed.

It gives us minimal knowledge of what the dispensaries may carry.

```sql
CREATE VIEW DispensaryStrains AS
SELECT Dispensaries.disid AS "Dispensary ID",
       Dispensaries.name AS "Dispensary Name",
       Strains.name AS "Strain"
FROM Dispensaries INNER JOIN DispensaryPlants ON (Dispensaries.disid = DispensaryPlants.disid)
INNER JOIN Plants ON (DispensaryPlants.pid = Plants.pid)
INNER JOIN Strains ON (Strains.sid = Plants.sid)
ORDER BY Dispensaries.name ASC;

SELECT * FROM DispensaryStrains;
```

	Dispensary ID integer	Dispensary Name character varying(50)	Strain text
1	1	Capitola Healing Association	Death Star
2	1	Capitola Healing Association	G13
3	1	Capitola Healing Association	Blue Dream
4	1	Capitola Healing Association	Sour Diesel
5	1	Capitola Healing Association	Mango Dream
6	1	Capitola Healing Association	Trainwreck Kush Haze
7	1	Capitola Healing Association	Atomic Northern Lights
8	1	Capitola Healing Association	Strawberry Kush
9	4	Exhale Med Center	Voodoo
10	4	Exhale Med Center	Death Star
11	4	Exhale Med Center	G13

View - Dispensary_Patients

The Dispensary_Patients view would be useful in order to see what patient purchase from

what dispensenary.

```
CREATE VIEW Dispensary_Patients AS
SELECT Dispensaries.name AS "Dispensary Name",
       Dispensaries.address AS "Dispensary Address",
       (People.firstname || ' ' || People.lastname) AS "Patient Name",
       age,
       People.address AS "Patient Address"
FROM Patients INNER JOIN People ON (Patients.peid = People.peid)
INNER JOIN DispensaryPatients ON (Patients.paid = DispensaryPatients.paid)
INNER JOIN Dispensaries ON (DispensaryPatients.disid = Dispensaries.disid)
ORDER BY People.lastname ASC;

SELECT * FROM Dispensary_Patients WHERE age < 35;
```

	Dispensary Name character varying(50)	Dispensary Address text	Patient Name text	age integer	Patient Address text
1	The Giving Tree of Denver	2707 W 38th Ave,	Daniel Bates	22	25 Main ST, Denver, Colorado, 80201
2	The Giving Tree of Denver	2707 W 38th Ave,	Norman Bates	19	25 Main ST, Denver, Colorado, 80201
3	Kind Love	4380 E Alameda A	Jessica Jorgen	25	14 Mill ST, Denver, Colorado, 80201
4	Capitola Healing Association	3088 Winkle Ave,	Jessica Morgan	23	19 High ST, Santa Cruz, California, 95062

Triggers/Stored Procedures

Update Maturity:

This trigger will update the isMatured boolean from false to true in the Plants table when

the growth cycle is complete.

```sql
CREATE OR REPLACE FUNCTION UpdateMaturity() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF OLD.ismatured = false AND seedling + vegcycle + floweringcycle =< date_part('date', pottingdate) + date_part('date', current_date) THEN
        NEW.ismatured = True;
    END IF;
    RETURN NEW;
END;
$BODY$
LANGUAGE 'plpgsql';

CREATE TRIGGER UpdateMaturity BEFORE UPDATE ON Plants FOR EACH ROW EXECUTE PROCEDURE UpdateMaturity();
```

Get Plants By Name:

Allows user to input a Strain name and returns information about the plant.

```sql
create or replace function GetPlants(text, REFCURSOR) returns refcursor as
$$
declare
    PlantName text       := $1;
    resultset   REFCURSOR := $2;
begin
    open resultset for
        SELECT Plants.pid,
               Strains.sid,
               Strains.name,
               Type.type,
               floweringcycle,
               cbdpercent,
               thcpercent,
               ismatured,
               avgheight,
               avgyield
        FROM Plants FULL OUTER JOIN Strains ON (Plants.sid = Strains.sid)
        INNER JOIN Type ON (Strains.tid = Type.tid)
        WHERE PlantName = Strains.name
        ORDER BY Plants.pid ASC;
    return resultset;
end;
$$
language plpgsql;
```

```
select GetPlants('Pineapple Express', 'results');
Fetch all from results;
```

	pid integer	sid integer	name text	type character varying(6)	floweringcycle integer	cbdpercent numeric	thcpercent numeric	ismatured boolean	avgheight integer	avgyield integer
1	20	30	Pinec	Hybrid	56	0.35	18.13	f	78	425

Security

Breaking down levels of security to different users is a must. First we start with a password protected Admin user.

```
CREATE USER Admin WITH PASSWORD 'alpaca';

GRANT SELECT, INSERT, UPDATE, DELETE
ON ALL TABLES IN SCHEMA public
TO Admin;
```

Scientists must be able to alter the Plants table in order to make changes in data to reflect the latest findings.

```
CREATE USER Scientist1 WITH PASSWORD 'science';

GRANT SELECT, INSERT, UPDATE
ON Plants
TO Scientist1;
```

A public user is created with select privileges to view certain public tables, but have no access to other tables.

```
CREATE USER PublicUser;

GRANT SELECT
ON Type, Strains, Plants, Dispensaries, Effects
TO PublicUser;
```

Known Issues/Future Enhancements

Known Issues:

- The UpdateMaturity function doesn't function properly.

- There is no view to show what customers buy from what dealers in Las Vegas.

Future Enhancements:

- A working UpdateMaturity function.

- A function that automatically inserts the plants into the Dispensary Plants table when they are fully matured.

- A prescribers table.

- A price guide table.