1. Business Description and Rules

We need to keep data for a company that sells plants to customers. The data that we need and the way it is organized is similar to the Altgeld Mart system. We store data about the plants we sell, the vendors we can get the plants from, our customers, and their orders.

2. Tables, Keys, and Relationships

We have two tables with data for plants that we have in inventory for sale.

The plants table includes the common name of the plant, the list price, the acquired date the date we started selling this plant) and the discontinued date (the date we stopped selling this plant) and how many of this plant we have in inventory (on_hand). The primary key is (plant_id). We sell plants based on the plants in this plants table.

The plant_taxonomy table includes the plant family and the scientific name of the plant (the genus and species). For example, the California poppy is in the family Papaveraceae and has the scientific name ***Eschscholzia californica.*** The primary key is (plant_id).Each row in the plant_taxonomy table is associated with one row in the plants table. But we can have some plants in the plants table where we do not have a row in the plant_taxonomy table. We could have some new variety of plants that does not yet have a scientific name- but we are going to sell it.

The tables for customers and orders are similar to those in the Altgeld_Mart database.

The order_headers table includes an order id number and a customer number and an order date. The primary key is (order_id)

The order_details table includes the plant ordered (plant_id), the actual price for the plant (price_per) and the number ordered (quantity). The primary key is a compound primary key (order_id, plant_id)

The customers table includes the customer's name and state. The primary key is (cust_id).

We sell only the plants listed in the plants table; this is enforced by the foreign key from the order_details table to the plants table.

We sell only to customers who are in the customer table; this is enforced by the foreign key from the order_headers table to the customers table.

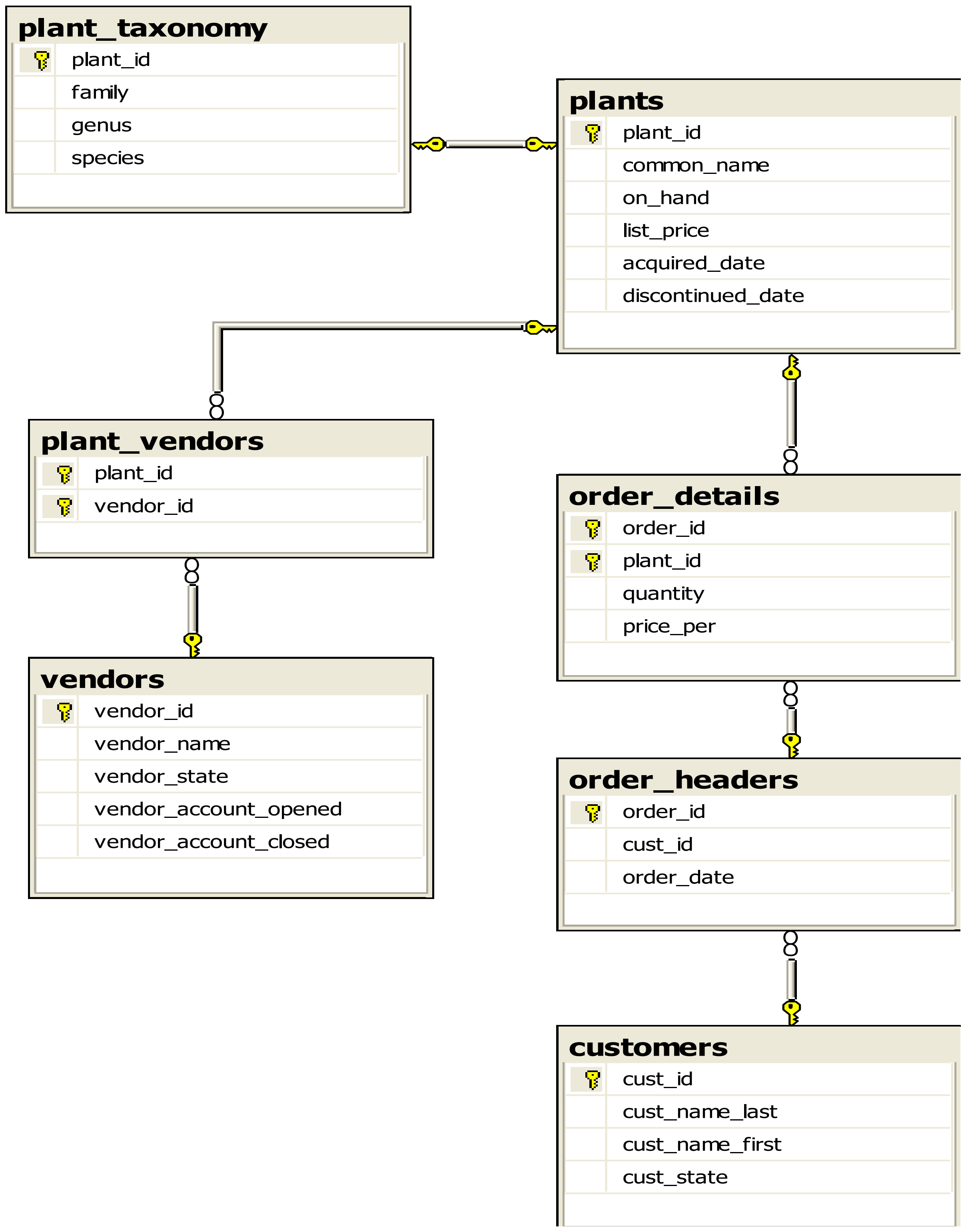
We get the plants that we sell from vendors. The vendors table has a primary key of (vendor_id).We store the vendor name, the state the vendor is in and the date we started buying plants from the vendor and the date we stopped doing business with the vendor.

We can have some plants where we do not have any current vendor; we can have some plants where we have exactly one vendor, we can have some plants where we have more than one vendor. We also can have some vendors who do not currently sell us any plants. (We might have just set them up as a vendor or maybe they supplied some plants in the past, but not at this time.) This means that we have a many-to-many relationship between plants and vendors so we have a junction table- plant_vendors. This has a primary key of (plant_id, vendor_id) and a foreign key to the plants table and a foreign key to the vendors table.

3. Definitions

The scientific name of the plant is the genus and species.
The extended cost is the quantity times the price_per

4. Graphic for tables



```
CREATE TABLE a_plants.plants(
  plant_id              int             NOT NULL
, common_name           varchar (30)    NOT NULL
, on_hand               int             NULL
, list_price            decimal(6, 2)   NOT NULL
, acquired_date         date            NOT NULL
, discontinued_date     date            NULL
, constraint plants_pk primary key (plant_id)
, constraint plant_id_range check(plant_id >= 100)
, constraint on_hand_range check(on_hand >= 0)
, constraint date_consistency check(discontinued_date>= acquired_date)
, constraint price_range check(list_price >= 0)
)engine = INNODB;

CREATE TABLE a_plants.plant_taxonomy(
  plant_id              int             NOT NULL
, family                varchar (30)    NOT NULL
, genus                 varchar (30)    NOT NULL
, species               varchar (30)    NOT NULL
, constraint plant_tax_pk primary key (plant_id)
, constraint plantTax_fk  foreign key (plant_id)
     references a_plants.plants (plant_id)
, constraint plantTax_un  unique (genus, species)
)engine = INNODB;


CREATE TABLE a_plants.vendors (
  vendor_id               int           not null
, vendor_name             varchar (25) not null
, vendor_state            char(2)       not null
, vendor_account_opened   date          not null
, vendor_account_closed   date          null
, constraint vendors_pk   primary key (vendor_id)
, constraint vendor_id_range check (vendor_id >0)
)engine = INNODB;



CREATE TABLE a_plants.plant_vendors(
  plant_id                int         not null
, vendor_id               int         not null
, constraint plant_vendors_pk primary key (plant_id, vendor_id)
, constraint vendor_fk foreign key (vendor_id)
             references a_plants.vendors (vendor_id)
, constraint plant_fk foreign key (plant_id)
             references a_plants.plants (plant_id)
)engine = INNODB;

CREATE TABLE a_plants.customers (
  cust_id             int               NOT NULL
, cust_name_last      varchar (25)      NULL
, cust_name_first     varchar (25)      NULL
, cust_state          char (2)          NOT NULL
, constraint customers_pk  primary key (cust_id)
, constraint cust_id_range check(cust_id >= 100)
);
```

```
CREATE TABLE a_plants.order_headers(
  order_id              int             NOT NULL
, cust_id               int             NOT NULL
, order_date            date            NULL
, constraint OH_pk primary key (order_id)
, constraint order_id_range check(order_id >= 100)
, constraint cust_fk foreign key (cust_id)
     references a_plants.customers (cust_id)
)engine = INNODB;


CREATE TABLE a_plants.order_details(
  order_id              int             NOT NULL
, plant_id              int             NOT NULL
, quantity              int             NULL
, price_per             numeric(6,2)      NULL
, constraint OD_pk  primary key  (order_id, plant_id)
, constraint order_fk foreign key (order_id)
     references a_plants.order_headers (order_id)
, constraint plantorder_fk foreign key (plant_id)
     references  a_plants.plants (plant_id)
, constraint quantity_range check(quantity >= 1)
, constraint price_per_range check(price_per >= 0)
)engine = INNODB;
```