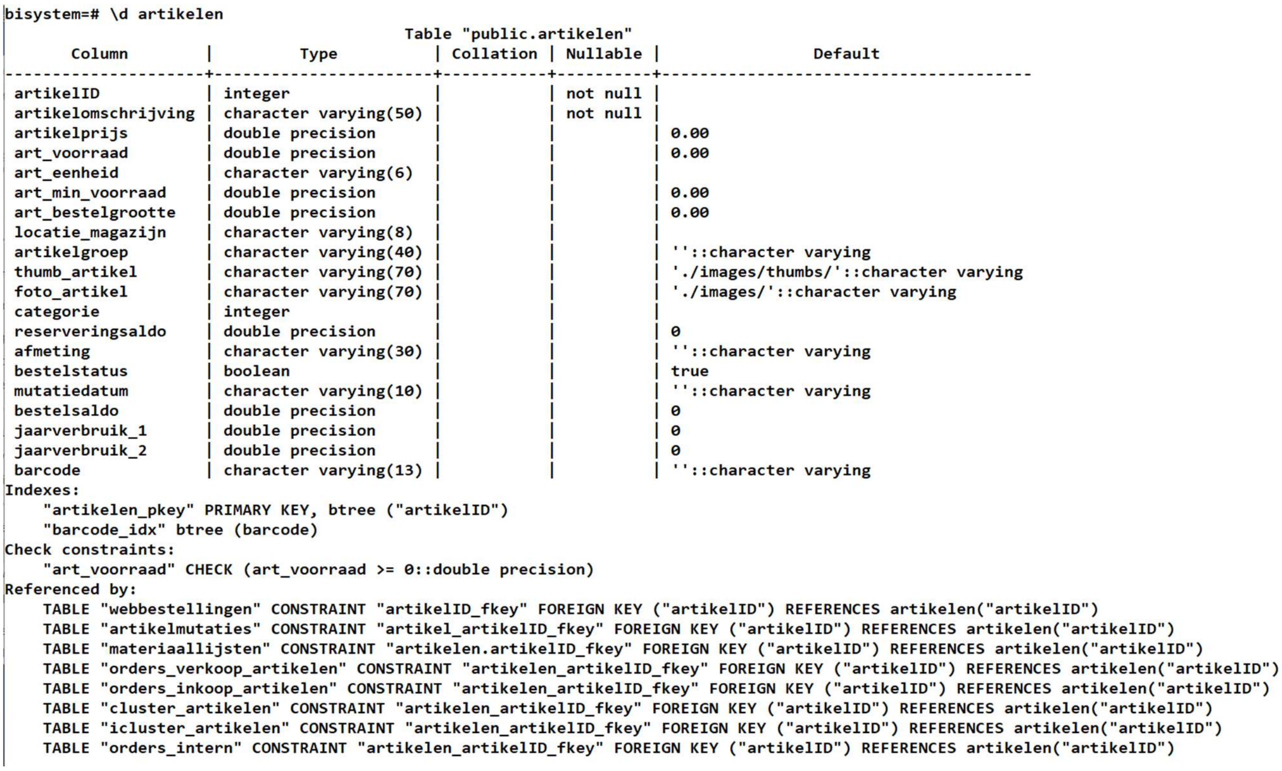
Documentation Pandora ERP system.

1. Stock and inventory control.

Description operation article table.



```
bisystem=# \d artikelen
                                        Table "public.artikelen"
       Column        |         Type          | Collation | Nullable |                Default
---------------------+-----------------------+-----------+----------+---------------------------------------
 artikelID           | integer               |           | not null |
 artikelomschrijving | character varying(50) |           | not null |
 artikelprijs        | double precision      |           |          | 0.00
 art_voorraad        | double precision      |           |          | 0.00
 art_eenheid         | character varying(6)  |           |          |
 art_min_voorraad    | double precision      |           |          | 0.00
 art_bestelgrootte   | double precision      |           |          | 0.00
 locatie_magazijn    | character varying(8)  |           |          |
 artikelgroep        | character varying(40) |           |          | ''::character varying
 thumb_artikel       | character varying(70) |           |          | './images/thumbs/'::character varying
 foto_artikel        | character varying(70) |           |          | './images/'::character varying
 categorie           | integer               |           |          |
 reserveringsaldo    | double precision      |           |          | 0
 afmeting            | character varying(30) |           |          | ''::character varying
 bestelstatus        | boolean               |           |          | true
 mutatiedatum        | character varying(10) |           |          | ''::character varying
 bestelsaldo         | double precision      |           |          | 0
 jaarverbruik_1      | double precision      |           |          | 0
 jaarverbruik_2      | double precision      |           |          | 0
 barcode             | character varying(13) |           |          | ''::character varying
Indexes:
    "artikelen_pkey" PRIMARY KEY, btree ("artikelID")
    "barcode_idx" btree (barcode)
Check constraints:
    "art_voorraad" CHECK (art_voorraad >= 0::double precision)
Referenced by:
    TABLE "webbestellingen" CONSTRAINT "artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
    TABLE "artikelmutaties" CONSTRAINT "artikel_artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
    TABLE "materiaallijsten" CONSTRAINT "artikelen.artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
    TABLE "orders_verkoop_artikelen" CONSTRAINT "artikelen_artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
    TABLE "orders_inkoop_artikelen" CONSTRAINT "artikelen_artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
    TABLE "cluster_artikelen" CONSTRAINT "artikelen_artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
    TABLE "icluster_artikelen" CONSTRAINT "artikelen_artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
    TABLE "orders_intern" CONSTRAINT "artikelen_artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
```

Above a screenshot of the table with its connections with other tables.
We won’t explain common fields, for I assume this is known.
The fields and its purpose we explain are:
art_min_voorraad
art_bestelgrootte
categorie
reserververingsaldo
bestelsaldo
bestelstatus
mutatiedatum
jaarverbruik_1
jaarverbruik_2
This fields are used for inventory control.
The field art_voorraad (stock) is a check constraint field, with control of positive amount.

Inventory Control.

art_min_voorraad:

When this article is driven by minimum stock and not reservation, which is determined by the field categorie, the article must be ordered when stock becomes lower then this value.
The minimum stock is calculated by annual consumption and the category (field categorie), by which the number of delivery times per year is determined.

The decision or an article is ordered by stock-control (mainly online orders and countersales) should be made by the circumstances of the annual consumption is equally spread over the year. If not the article should be ordered by reservation.

art_bestelgrootte:
This value is determined by the Camp formula, that recognize the order size based on annual comsumption. See the explanation of annual consumption by
jaarverbruik_1 and jaarverbruik_2. Camp formula: Q = √2DF/HP
Stands for:
Q = Quantity

- D = Demand / year
- F = Fixed Costs (order costs conversion costs)
- H = Stock costs as a percentage of the price
- P = Price of the product

The orders are made with the program voorraadbeheersing and the table materiaallijsten,

here is the Camp formula used for ordering.

By starting Pandora.py the first time after Januari 1st, the minimum stock and order size is determined for all articles and stored in the database.

So it's possible to switch from ordering by reservation towards ordering by stock driven.

The edited articles with values are added on a table artikelmutaties and afdrachten for accountancy and paying taxes.

All stock values are added or deducted in the column jaarverbruik_1 or jaarverbruik_2

Even years jaarverbruik_2 odd years jaarverbruik_1, so always a year-consumption is available.

After a year has passed fields of a year agoo are set to 0, so that the countings start from scratch.

This is established within the module Pandora.py. Also in this module the monthly values of the stock are counted, and is determined (with the mutatiedatum – last mutation) wether the mutatiedatum (transactiondate) is longer then a year agoo. Then it is called dead stock (incourant in Dutch).

With this items graphs are produced.

The data for the graphs are stored in the table magazijnvoorraad (warehouse stock).

Products are ordered direct online, by countersales or through article lists of clustercalculaties or iclustercalculaties.

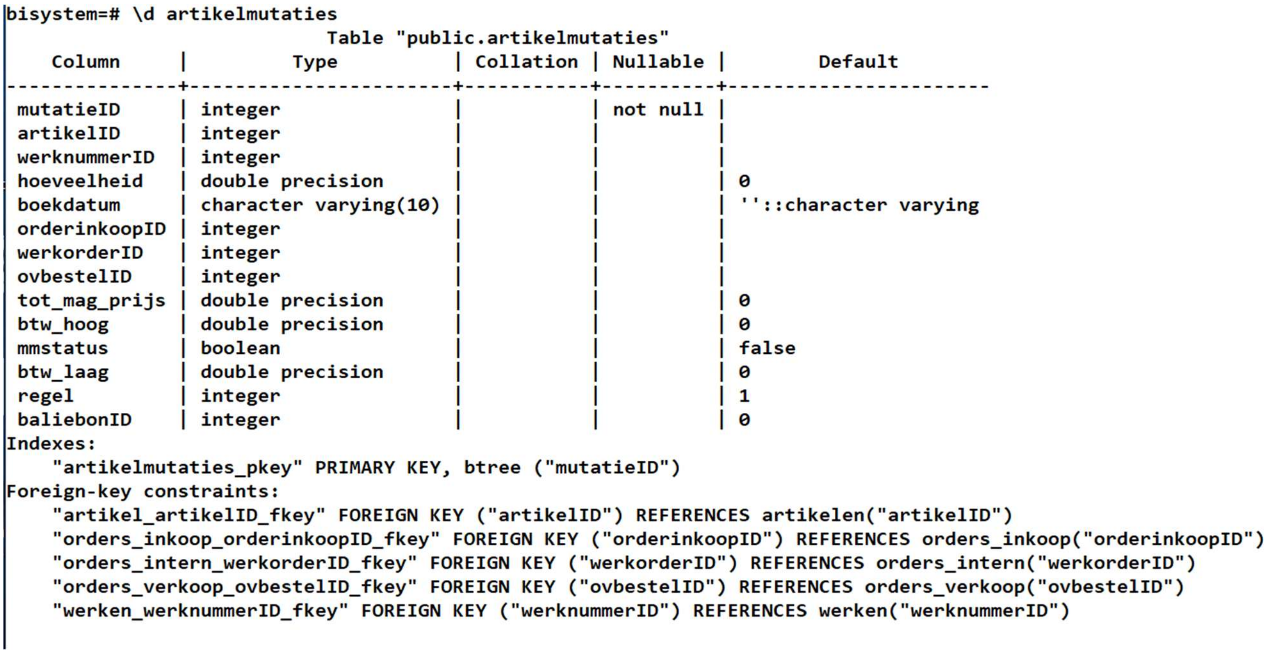
Purchasing, delivery, calling , orderpicking and processing.

1 Online sales

Thirst the client logs on and chooses products. The orderlist is established. The products are added on the field reservations in table artikelen, if ordered. Then the products are picked the fields reservations and stock are deducted and field jaarverbruik_1 or jaarverbruik_2 (see Inventory Control) is added.

The table afdrachten (payments) is filled with the total amount and total VAT.

The table artikelmutaties (article transactions) is filled with the delivered orderlines data.



```
bisystem=# \d artikelmutaties
                              Table "public.artikelmutaties"
    Column     |         Type          | Collation | Nullable |        Default
---------------+-----------------------+-----------+----------+-----------------------
 mutatieID     | integer               |           | not null |
 artikelID     | integer               |           |          |
 werknummerID  | integer               |           |          |
 hoeveelheid   | double precision      |           |          | 0
 boekdatum     | character varying(10) |           |          | ''::character varying
 orderinkoopID | integer               |           |          |
 werkorderID   | integer               |           |          |
 ovbestelID    | integer               |           |          |
 tot_mag_prijs | double precision      |           |          | 0
 btw_hoog      | double precision      |           |          | 0
 mmstatus      | boolean               |           |          | false
 btw_laag      | double precision      |           |          | 0
 regel         | integer               |           |          | 1
 baliebonID    | integer               |           |          | 0
Indexes:
    "artikelmutaties_pkey" PRIMARY KEY, btree ("mutatieID")
Foreign-key constraints:
    "artikel_artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
    "orders_inkoop_orderinkoopID_fkey" FOREIGN KEY ("orderinkoopID") REFERENCES orders_inkoop("orderinkoopID")
    "orders_intern_werkorderID_fkey" FOREIGN KEY ("werkorderID") REFERENCES orders_intern("werkorderID")
    "orders_verkoop_ovbestelID_fkey" FOREIGN KEY ("ovbestelID") REFERENCES orders_verkoop("ovbestelID")
    "werken_werknummerID_fkey" FOREIGN KEY ("werknummerID") REFERENCES werken("werknummerID")
```

2. Countersales

Balieverkoop barcodescanning. This products are straight away sold by means of barcodescanning.

The stock is directly deducted if the numbers don’t exceed art_voorraad minus reservations.

The field jaarverbruik_1 or jaarverbruik_2 (see Inventory Control) is added.

The table afdrachten (payments) is filled with the total amount and total VAT.

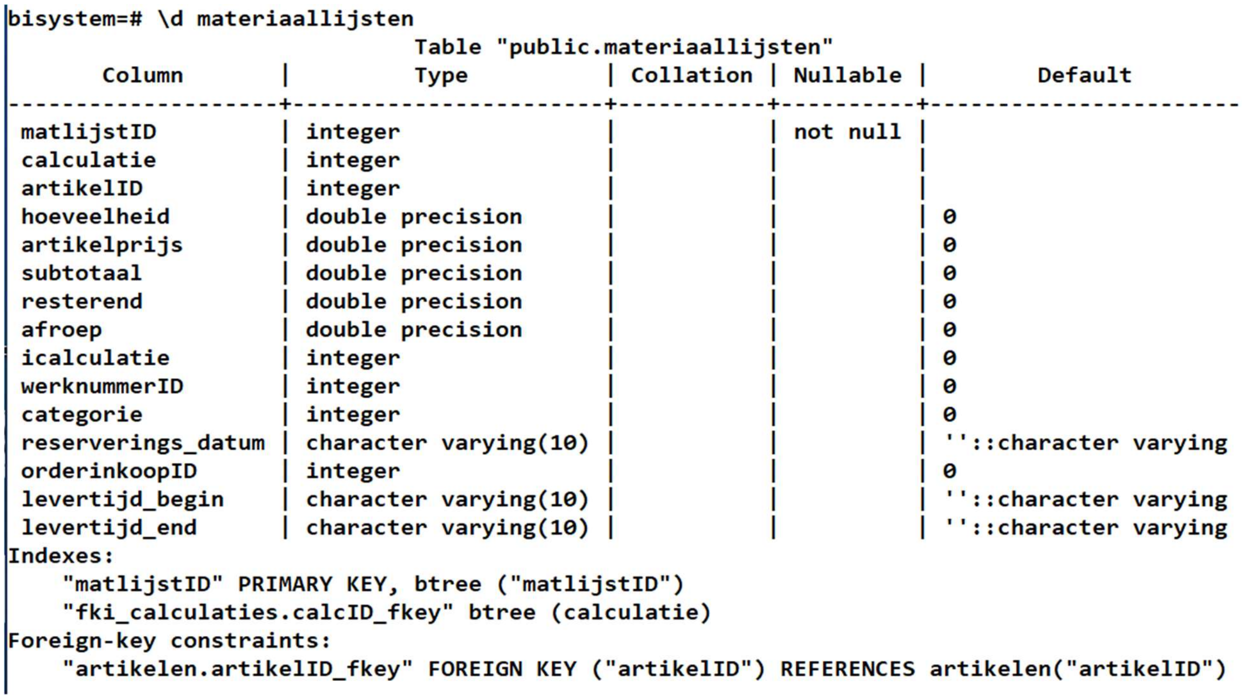
The table artikelmutaties (article transactions) is filled with the delivered orderlines data.

The table artikelen helds a column barcode (String 13 positions).
In the module invoerArtikel.py (insert article) the field artikelID is transferred to barcode with the EAN 13 structure. The first 2 numbers is the country code. The next 5 numbers are the company numbers, the following 5 numbers is the product number en the last number is a validity check number.
In this module is a image saved of the barcode in the folder ./forms/Barcodelabels
This image can be printed for labeling the product or storage bin in the warehouse, so it’s enabled for scanning.
For more information about scanning see Barcodescanning document.

3. **Internal orders** (Semi finished product orders factory)

If the Iclustercalculatie is approved by the client and linked as a intern_order the articles from the article list are added in the column reservations (reserveringen)

Also the reservations are stored in the table materiaallijsten for financial reasons, purchasing and for producing orderpickinglists.



```
bisystem=# \d materiaallijsten
                              Table "public.materiaallijsten"
      Column        |          Type          | Collation | Nullable |        Default
--------------------+------------------------+-----------+----------+-----------------------
 matlijstID         | integer                |           | not null |
 calculatie         | integer                |           |          |
 artikelID          | integer                |           |          |
 hoeveelheid        | double precision       |           |          | 0
 artikelprijs       | double precision       |           |          | 0
 subtotaal          | double precision       |           |          | 0
 resterend          | double precision       |           |          | 0
 afroep             | double precision       |           |          | 0
 icalculatie        | integer                |           |          | 0
 werknummerID       | integer                |           |          | 0
 categorie          | integer                |           |          | 0
 reserverings_datum | character varying(10)  |           |          | ''::character varying
 orderinkoopID      | integer                |           |          | 0
 levertijd_begin    | character varying(10)  |           |          | ''::character varying
 levertijd_end      | character varying(10)  |           |          | ''::character varying
Indexes:
    "matlijstID" PRIMARY KEY, btree ("matlijstID")
    "fki_calculaties.calcID_fkey" btree (calculatie)
Foreign-key constraints:
    "artikelen.artikelID_fkey" FOREIGN KEY ("artikelID") REFERENCES artikelen("artikelID")
```

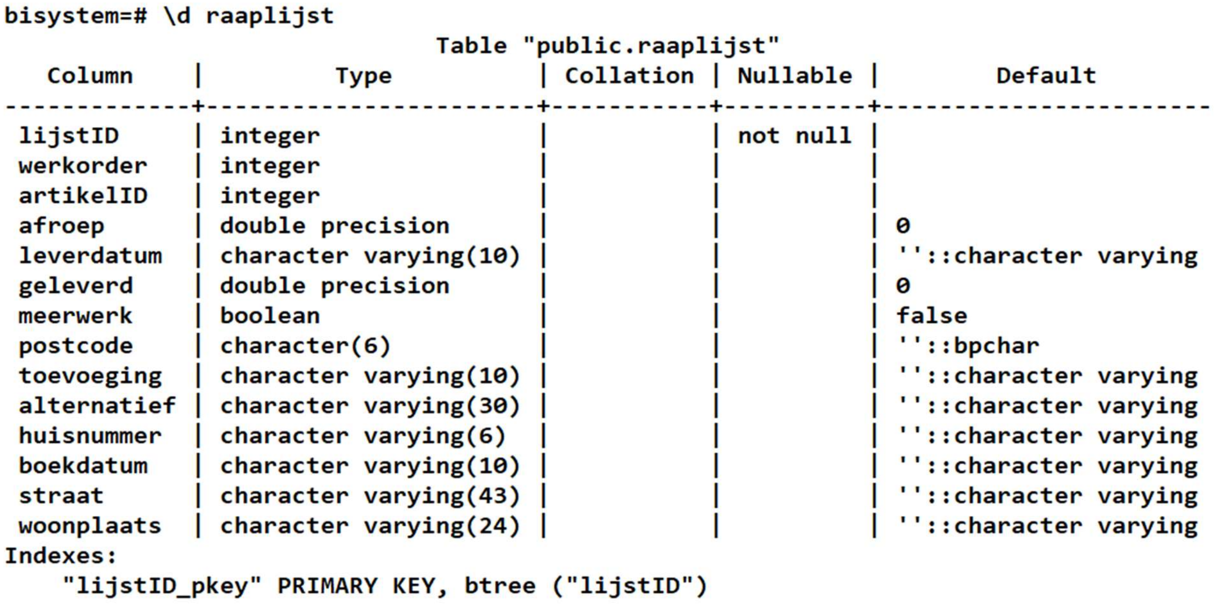
The total value of the articles from the Iclustercalculation is added on table orders_intern field begr_materialen.

Order balancing this field (bestelsaldo) is filled when a inkooporder for materials is ordered (initiated by table materiaallijsten), also the orderstatus (bestelstatus) is blocked (False), so ordering is not possible until materials are delivered and approved.

Then purchaseorder products are delivered and approved the fields bestelsaldo is deducted , art_voorraad (stock) is added with the approved amounts and the field orderstatus (bestelstatus) is released (True).

In the table artikelmutaties (article transactions) the delivered products are being inserted.

When the products are provided on the work the column reservations is deducted and the field jaarverbruik_1 or jaarverbruik_2 (see Inventory Control) is added with the issued numbers. The stock (art_voorrraad) is also deducted and the table materiaallijsten is updated with the picklist deliveries. (quantity, demand, remaining) The picklist (table raaplijst) is produced from the materaallijsten (module artikelAfroep.py) when the products are called with the desired delivery date and delivery place. The products are picked in the warehouse and delivered on the desired time and place.



```
bisystem=# \d raaplijst
                              Table "public.raaplijst"
   Column    |         Type          | Collation | Nullable |        Default
-------------+-----------------------+-----------+----------+----------------------
 lijstID     | integer               |           | not null |
 werkorder   | integer               |           |          |
 artikelID   | integer               |           |          |
 afroep      | double precision      |           |          | 0
 leverdatum  | character varying(10) |           |          | ''::character varying
 geleverd    | double precision      |           |          | 0
 meerwerk    | boolean               |           |          | false
 postcode    | character(6)          |           |          | ''::bpchar
 toevoeging  | character varying(10) |           |          | ''::character varying
 alternatief | character varying(30) |           |          | ''::character varying
 huisnummer  | character varying(6)  |           |          | ''::character varying
 boekdatum   | character varying(10) |           |          | ''::character varying
 straat      | character varying(43) |           |          | ''::character varying
 woonplaats  | character varying(24) |           |          | ''::character varying
Indexes:
    "lijstID_pkey" PRIMARY KEY, btree ("lijstID")
```

The table artikelmutaties (article transactions) is inserted with the delivered picklist data and field jaarverbruik_1 or jaarverbruik_2 (see Inventory Control) is added.

The value of the delivered articles is added on table orders_intern field werk_materialen

4. External Orders (Installation orders at external workareas)

If the Clustercalculatie is approved by the client and linked as a werknumber the articles from the article list are added in the column reservations (reserveringen)

Also the reservations are stored in the table materiaallijsten for financial reasons, purchasing and for orderpickinglists.

The total value of the articles from the Clustercalculation is added on table werken field begr_materialen.

Order balancing this field (bestelsaldo) is filled when a inkooporder for materials is ordered (initiated by table materiaallijsten), also the orderstatus(bestelstatus) is blocked (False), so ordering is not possible until materials are delivered and approved.

Then purchaseorder products are delivered and approved the fields bestelsaldo is deducted , art_voorraad (stock) is added with the approved amounts and the field orderstatus (bestelstatus) is released (True).

In the table artikelmutaties (article transactions) the delivered products are being inserted.

When the products are provided on the work the column reservations is deducted and the field jaarverbruik_1 or jaarverbruik_2 (see Inventory Control) is added with the issued numbers. The stock (art_voorrraad) is also deducted and the table materiaallijsten is updated with the picklist deliveries. (quantity, demand, remaining) The picklist (table raaplijst) is produced from the materaallijsten (module artikelAfroep.py) when the products are called with the desired delivery date and delivery place. The products are picked in the warehouse and delivered on the desired time and place.

The table artikelmutaties (article transactions) is inserted with the delivered picklist data and field jaarverbruik_1 or jaarverbruik_2 (see Inventory Control) is added.

The value of the delivered articles is added on table werken field kosten_materialen.