Esimerkkikyselyitä

Lippukauppa ja e-liput

Montako e-lipputilausta on lunastettu tiettynä päivänä

python manage.py shell

```
from lippukala.models import Order
from dateutil.tz import tzlocal
from datetime import datetime

tz = tzlocal()
loerdag_start = datetime(2015, 9, 5, 0, 0, 0, tzinfo=tz)
soendag_start = datetime(2015, 9, 6, 0, 0, 0, tzinfo=tz)

Order.objects.filter(code__used_on__gte=loerdag_start, code__used_on__lt=soendag_start).distinct().count()
# 2600
```

Lunastettujen koodien määrä per minuutti tietyllä aikavälillä

```
SELECT date_trunc('minutes', used_on) as ts, count(order_id) FROM lippukala_code
WHERE used_on between '2015-09-05 00:07:00'::timestamp AND '2015-09-05 14:00:00'::timestamp
GROUP BY ts ORDER BY ts;
```

Lunastettujen tilausten määrä per minuutti tietyllä aikavälillä

```
SELECT date_trunc('minutes', used_on) as ts, count(*) FROM lippukala_order o, lippukala_code c
WHERE o.id=c.id AND c.used_on between '2015-09-05 00:07:00'::timestamp AND '2015-09-05 14:00:00'::timestamp
GROUP BY ts ORDER BY ts;
```

Tilauksia, joissa yli 4 kpl viikonloppulippuja (Tracon 2018 event_id/product_id:t)

```
SELECT ti_o.id,ti_op.count AS _cnt FROM tickets_order AS ti_o, tickets_orderproduct AS ti_op
WHERE ti_o.id=ti_op.order_id AND ti_o.event_id=49 AND ti_o.payment_date is not null
GROUP BY ti_o.id, ti_op.count, ti_op.product_id
HAVING (ti_op.product_id=130 AND ti_op.count > 4)
ORDER BY _cnt DESC;
```

Minkä kokoisia tilauksia ihmiset tilaa

python manage.py shell

```
from tickets.models import OrderProduct
from collections import Counter

ops = OrderProduct.objects.filter(
    order__event__slug='tracon2018',
    order__confirm_time__isnull=False,
    order__payment_date__isnull=False,
    order__cancellation_time__isnull=True,
    count__gte=1,
    product__name__icontains='viikonloppu',
)
c = Counter()
for op in ops:
   c[op.count] += 1
c
# Counter({1: 1296, 2: 631, 3: 199, 4: 63, 5: 33, 6: 7, 7: 2, 8: 1, 9: 3})
```

Siirrä valittujen tilausten majoittujat toiseen kouluun

python manage.py shell

```
from tickets.models import *

# order_ids = {101375, 101577, 101577, 101696, 101696, 101724, 101839, 101879, 101903, 102041, 102054, 102078,
102078, 102086, 102094, 102094, 102094, 102094, 102228, 102228, 102228, 102269, 102272, 102464, 102464, 102464,
102467, 102504, 102591, 102596, 102596, 102596, 102835, 102835, 102917, 103034, 103412, 103509, 103677, 103701,
103724, 103900, 104173}
order_ids = {104958}

product_map = {
    173: 175,
    174: 176,
}

lgroup_map = {
    135: 137,
    136: 138,
}

# Due to braindamage, shop leaves ops with count=0 lying around. This will cause IntegrityError when updating
product of ops. So clean up.
# OrderProduct.objects.filter(order__confirm_time__isnull=False, count=0).delete()

# Swap products in orders
for old_product_id, new_product_id in product_map.items():
    old_product = Product.objects.get(id=old_product_id)
    print(old_product)
    ops = OrderProduct.objects.filter(order__in=order_ids, product=old_product_id, count__gt=0)
    print(ops.count())

    # Uncomment this to actually move
    # ops.update(product=new_product_id)

# Update references in AccommodationInformation
for old_lg_id, new_lg_id in lgroup_map.items():
    old_lg = LimitGroup.objects.get(id=old_lg_id)
    new_lg = LimitGroup.objects.get(id=new_lg_id)
    ais = AccommodationInformation.objects.filter(order_product__order__in=order_ids, limit_groups=old_lg_id)

    print(old_lg)
    for ai in ais:
        print(ai)

        # Uncomment these to actually move
        # ai.limit_groups.remove(old_lg)
        # ai.limit_groups.add(new_lg)
```

Badget ja nimilistat

Kuinka monta badgea on noudettu milläkin tunnilla

psql

```
select date_trunc('hour', used_on) as hour, count(*)
from lippukala_code c
inner join lippukala_order o on (c.order_id = o.id)
where o.event = 'tracon2018'
group by hour
order by hour;
```

Ohjelmanhallinta

Ohjelmanumerot, joista puuttuu ohjelmanjärjestäjän sähköpostiosoite

python manage.py shell

```
from programme.models import Programme
for programme in Programme.objects.filter(category__event__slug='traconx', programmerole__person__email='').
distinct():
    print programme.title
```

Yleiset

Listaa kaikki taulut, joilla on vierasavain *core_personiin*

StackOverflow

```
SELECT
    tc.constraint_name, tc.table_name, kcu.column_name,
    ccu.table_name AS foreign_table_name,
    ccu.column_name AS foreign_column_name
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
      ON tc.constraint_name = kcu.constraint_name
    JOIN information_schema.constraint_column_usage AS ccu
      ON ccu.constraint_name = tc.constraint_name
WHERE constraint_type = 'FOREIGN KEY' AND ccu.table_name = 'core_person';
```