

# Расчёт требуемого количества сборщиков и курьеров

Логика расчета одинакова для сборщиков и курьеров. Прогноз трудоемкости автоматически преобразуется в требуемое количество сотрудников. Алгоритм включает следующие шаги:

## 1. Оценка скорости сборки сотрудников

Для каждого сотрудника рассчитывается средняя скорость сборки на основе фактических данных за последний месяц. Скорость измеряется в количестве строк заказов, собираемых за час. Расчет выполняется в преобразовании `scheduleBindings`, код которого хранится в БД приложения. Из представления `time_series_col_hourly` извлекается сумма собранных строк, деленная на общее количество часов работы. Учитываются только часы с продолжительностью сборки от 1000 до 3600 секунд, что исключает неполные или чрезмерно длинные интервалы и повышает надежность оценки.

```
$userStatAvgRaw := $query($queryQuery, {
  'table': 'time_series_col_hourly',
  'from': $userStatFrom,
  'restaurantIds': $restaurantIds,
  'query': [
    {
      'select': {
        'user_id': 'user_id'
      },
      'sum': {
        'sum_qty': 'qty'
      },
      'count': {
        'cnt': '*'
      },
      'where': {
        'len >=': 1000000,
```

```

        'len <=': 3600000
    }
}
];
$userStatAvg := $userStatAvgRaw.tquery{
    $string(user_id): {
        'speed': $number(sum_qty) / $number(cnt)
    }
};

```

## 2. Расчет плановой производительности

Далее для каждого часа прогнозируемого периода рассчитывается ожидаемое количество собранных строк на основе списка запланированных сборщиков и их индивидуальной скорости. Расчет выполняется в настройке `dayPlaceTransform`. Если дата относится к будущему, сотрудники фильтруются по позиции "Сборщик". Для каждого сотрудника извлекается индивидуальная скорость (`userSpeed`). Интервал события разбивается на часы; для каждого часа вклад рассчитывается как длительность работы  $\times$  скорость /  $\sim 3,6$  млн миллисекунд (для перевода в часы). Вклады суммируются в `sum_qty` для часа. В массиве `$hours` (0–23 часа) значение `plan` устанавливается как `sum_qty` (или 0 при отсутствии данных). Таким образом формируется плановая производительность — ожидаемое количество собранных строк заказов на основе текущего плана.

```

$eventsStat := $inPast
    ? $lookup($lookup($ttStat, $string(place.id)), $dateStr)
    : $statEvents[position~>/Сборщик/].(
        $userSpeed := [$lookup($users, $string(userId)).userSpeedAvg, $speed][!=$null][0];
        $user := $lookup($users, $string(userId));
        $event := $user.isActive ? $range(beginAt, endAt).intersect($day);
        $event ? $array($event.snapTo('hour').by('hour')).(
            $r := $.range('hour').intersect($event);
            $r ? {'h': $.format('H'), 't': $r.diff()*$userSpeed/3599999}
        );
    ){
        h: { 'sum_qty': $round($sum(t)) }
    };

$hours := [0..23].(
    ...

```

```
$plan := $number([$lookup($eventsStat, $string($)).sum_qty, 0][0]);  
...  
);
```

### 3. Сравнение с прогнозом

Далее рассчитывается разность между прогнозируемым количеством строк на сборку и плановой производительностью. Этот шаг выполняется в настройке `dayPlaceTransform`. В массиве `$hours` для каждого часа извлекается прогнозное значение (`$stat`) и план (`$plan`). Если дата в будущем, разность (`$delta`) вычисляется как (прогноз - план) / средняя скорость сборки. Положительное значение указывает на нехватку сотрудников, отрицательное — на избыток. Если прогноз отсутствует, `$delta` устанавливается в 0, чтобы избежать ошибок в рекомендациях.

```
$hours := [0..23].(  
  $statObj := $lookup($hourStat, $string($));  
  $stat := $number($inPast ? $statObj.cnt : $statObj.cnt) * $extraNorm;  
  $plan := $number([$lookup($eventsStat, $string($)).sum_qty, 0][0]);  
  $delta := $stat  
    ? $inPast  
      ? $round($plan - $stat, 1)  
      : $round(($stat - $plan) / $speed)  
    : 0;  
  ...  
);
```

### 4. Корректировка численности

Затем разность преобразуется в дополнительное или избыточное количество сотрудников с использованием средней скорости сборки по всей торговой точке. Эта скорость рассчитывается в `scheduleBindings` аналогично индивидуальной, но для всех сборщиков вместе. Из представления `time_series_col_hourly` извлекается сумма собранных строк с группировкой по торговой точке и времени, а также агрегированные метрики: суммы, средние и количество записей. Ограничения по секундам сборки здесь не применяются.

```
$ttStatRaw := $query($tqueryQuery, {  
  'table': 'time_series_col_hourly',  
  'from': $statFrom,  
  'to': $to,  
  'restaurantIds': $restaurantIds,  
  'query': [  
    {
```

```

'select': {
  'restaurant_id': 'restaurant_id',
  'tstamp': 'tstamp'
},
'avg': {
  'avg_cnt': 'cnt',
  'avg_qty': 'qty'
},
'sum': {
  'sum_qty': 'qty',
  'sum_cnt': 'cnt'
},
'count': {
  'cnt': '*'
}
}
]
});

```

```

$ttStat := $ttStatRaw.tquery{
$string(restaurant_id): $.{
'time': $moment(tstamp),
'sum_cnt': sum_cnt,
'sum_qty': sum_qty,
'avg_cnt': avg_cnt,
'avg_qty': avg_qty,
'cnt': cnt
}{
time.format('YYYY-MM-DD'): ${time.format('H')}: {
'sum_cnt': sum_cnt,
'sum_qty': sum_qty,
'avg_cnt': avg_cnt,
'avg_qty': avg_qty,
'cnt': cnt
}}
}
};

```

## 5. Вывод итогового количества

Наконец, полученное число используется для формирования рекомендаций о том, сколько сотрудников добавить или убрать в конкретный час. Пороговые значения определяются в настройке `dayPlaceTransform`. В функции `$formatRowPlan` (применяемой для будущих дней) извлекается объект часа с разностью (`$delta`). На основе этого значения формируется текст рекомендации: если разность равна нулю, отображается "Норма" в зеленом цвете без изменений; если разность положительная, показывается "Нехватка X человек" в красном (X — значение разности, что означает необходимость добавить сотрудников); если разность отрицательная, отображается "Избыток X человек" в оранжевом (X — абсолютное значение разности, что предполагает возможность убрать сотрудников). Если данных нет, выводится сообщение "Нет данных".

```
$formatRowPlan := function($t) {(
    $h := $hours[$t];
    $info := $h.stat ? $h.plan & ' из ' & $h.stat : '-';
    $span := $h.delta = 0
        ? ''
        : $h.delta > 0
            ? $redSpan
            : $amberSpan;
    $spanEnd := $h.delta = 0
        ? ''
        : '';

    $span & $formatNumber($t, '00') & ':00 - ' & $formatNumber($t+1, '00') & ':00' & $spanEnd & ' |

    ' & $span & $info & $spanEnd & '
    ' &
    ($h.stat
    ? $h.delta = 0
    ? $greenBox & 'Норма'
    : $h.delta > 0
    ? $redBox & 'Нехватка ' & $h.delta & ' чел.'
    : $amberBox & 'Избыток ' & -$h.delta & ' чел.'
    : 'Нет данных') & '
    ';
});
```

# Формула расчета требуемого количества сборщиков или курьеров

Все вышеперечисленные шаги можно описать формулой, которая рассчитывает рекомендуемое общее количество работников, необходимое для обработки заказов в конкретный час.

$$R_t = N_t + \left\lceil \frac{D_t - \sum_{i=1}^{N_t} v_i}{\bar{v}} \right\rceil$$

Где:

$t$  — номер часа, в который производится расчет

$D_t$  — общее количество заказов, поступивших в час  $t$

$N_t$  — количество работников, занятых в час  $t$

$v_i$  — время, необходимое для обработки заказа  $i$ -го, поступившего в час  $t$

$\bar{v}$  — среднее время, необходимое для обработки заказа

$R_t$  — общее количество работников, необходимое для обработки заказов в час  $t$

---

Revision #7

Created 19 August 2025 07:39:06 by Семен Долгов

Updated 26 September 2025 06:29:11 by Семен Долгов