





```
Enum user_role {  
    admin [note: "Владелец, полный доступ: биллинг, настройки, аналитика, управление персоналом"]  
    manager [note: "Менеджер точки: управляет стоп-листами, заказами и видимостью меню. Нет доступа к биллингу"]  
    cashier [note: "Кассир: Только список входящих заказов и кнопка \"Выдан\". Минимальные права"]  
    customer [note: "Клиент: заказ еды — хранится в users, но роль определяется ОТСУТСТВИЕМ записи в staff_roles."] } 
```

```
Enum order_status {  
    NEW [note: "Создан в PWA, Заказ создан, деньги не списаны. Можно отменить без возврата"]  
    WAITING_PAYMENT [note: "Ожидает оплаты эквайрингом. Корзина зафиксирована, платёж инициирован. Ждём webhook от iiko на кухню. Платёж подтверждён, заказ передан в iiko. Отмена только через возврат"]  
    COOKING [note: "Успешно ушёл в iiko на кухню. Платёж подтверждён, заказ передан в iiko. Отмена только через возврат"]  
    READY [note: "Готов к выдаче. iiko прислал статус готовности. PWA показывает клиенту крупный номер заказа"]  
    COMPLETED [note: "Завершён/ Клиент забрал заказ или прошло 2 часа с момента READY"]  
    CANCELLED [note: "Отменён (с возвратом или без). Причина в orders.cancelled_reason, возврат в orders.refund_id"] } 
```

```
Enum pos_sync_status {  
    NOT_SENT [note: "Ожидает отправки в iiko. Воркер берёт эти записи каждые 5 сек и пытается отправить в iiko"]  
    SENT [note: "Успешно принято iiko. iiko вернул 200 OK и pos_order_id. Заказ принят кухней"]  
    ERROR [note: "Ошибка API (требует внимания админа). iiko вернул ошибку. Детали в pos_sync_logs.response_payload. Требует внимания"] } 
```

```
Enum billing_status {  
    ACTIVE [note: "Сервис оплачен. paid_until в будущем. Заказы принимаются в штатном режиме"]  
    PAST_DUE [note: "Задолженность < 3 дней (предупреждение). paid_until просрочен менее 3 дней. Баннер-предупреждение в админке"]  
    BLOCKED [note: "Сервис отключён за неуплату. paid_until просрочен более 3 дней. PWA не принимает заказы, только страница оплаты"] } 
```

```
Enum payment_status {  
    PENDING [note: "Платёж инициирован. Платёж создан в БД и у банка. Ждём webhook. Деньги не списаны"]  
    SUCCESS [note: "Деньги списаны. Банк подтвердил списание. Заполняем paid_at, меняем order.status COOKING"]  
    FAILED [note: "Ошибка оплаты. Банк отказал. Заказ возвращается в статус NEW для повторной оплаты"]  
    REFUNDED [note: "Возврат выполнен. Возврат выполнен через API банка. Заполняем orders.refunded_at"]  
    CANCELLED [note: "Платёж отменён до списания (клиент закрыл страницу или истёк таймаут 15 мин)"] } 
```

```
Enum theme_preset {  
    light [note: "Светлая (по умолчанию). Белый фон, тёмный текст. Универсально. Дефолт для всех новых точек"]  
    dark [note: "Тёмная. Тёмный фон. Подходит для баров и кофеен с атмосферой"]  
    coffee [note: "Кофейная. Тёплые коричневые тона. Для кофеен и пекарен"]  
    fastfood [note: "Фастфуд. Яркие акценты, крупные кнопки. Для точек быстрого обслуживания"] } 
```

```
Enum discount_type {  
    PERCENT [note: "Скидка в процентах. Значение = %, итог: total * (1 - value/100). Валидировать: 1-100"]  
    FIXED [note: "Скидка фиксированной суммой. Значение = рубли, итог: MAX(total - value, 0). Защита от отрицательных значений"] } 
```

```
Enum order_type {  
    takeaway [note: "Самовывоз (по умолчанию в MVP). Единственный тип в MVP. Клиент называет номер заказа"]  
    dine_in [note: "В зале (будущее). Зарезервировано. Потребует поле table_number в orders"]  
    delivery [note: "Доставка (будущее). Зарезервировано. Потребует адрес и интеграцию с курьерами"] } 
```