Курсовой проект «Проектирование базы данных для управление складом» по дисциплине «Основы реляционных баз данных. MySQL».

Выполнил студент факультета python разработки: Бородич В.В.

Содержание

[Цели и задачи 3](#_Toc89428037)

[Описание основных процессов 4](#_Toc89428038)

[Установка и первоначальная настройка 4](#_Toc89428039)

[Распределение прав 4](#_Toc89428040)

[Назначение пользователей по складам 4](#_Toc89428041)

[При старте сессии 4](#_Toc89428042)

[Размещение заказа клиентом 4](#_Toc89428043)

[Размещение товаров на складах (Кладовщик) 4](#_Toc89428044)

[Подготовка заказа к сборке (менеджер) 4](#_Toc89428045)

[Назначение комплектовщика (менеджер либо автоматически) 5](#_Toc89428046)

[Сбор заказа нашей фирмой (комплектовщик) 5](#_Toc89428047)

[Просмотр статистических данных 5](#_Toc89428048)

[Тестовый пример 6](#_Toc89428049)

[Приложение 1 7](#_Toc89428050)

[Приложение 2 8](#_Toc89428051)

[Приложение 3 9](#_Toc89428052)

## Цели и задачи

1. Снижение рисков пересорта при сборе заказов.

2. Обеспечение незалеживаемости товаров.

3. Оперативная информация об остатках товара на складах.

## Описание основных процессов

### **Установка и первоначальная настройка**

Выполнить скрипты «Создание структуры БД.sql» и «Первоначальное заполнение.sql».

### **Распределение прав**

Предполагается использование двух ролей:

Менеджер: выполняет первичную обработку заказа, распределяет заказ между складами.

Кладовщик(Комплектовщик): выполняет сборку заказа, оформляет приход товара на склад.

В текущем решении роли формальные.

### **При старте сессии**

При входе в программу нужно выбрать текущий вариант настроек «select\_options» (приложение 3) и текущего пользователя. Для этого используется процедура «select\_user» (описание в приложении 3).

### **Назначение пользователей по складам**

При создании пользователей, для пользователей с ролью «кладовщик» обязательно указание склада, где работает пользователь. Для этого используется процедура «select\_user» (описание в приложении 3).

### **Размещение заказа клиентом**

Предполагается, что заказ поступает из внешней системы (например: интернет магазин).

Запрос «Добавление заказа» приложение 2.

### **Размещение товаров на складах (Кладовщик)**

Приход товара на склад, в конкретную ячейку согласно планограмме, процедура

«modify\_remains» (описание в приложении 3).

### **Подготовка заказа к сборке (менеджер)**

Заказы готовые к обработке (в статусе «Размещен») собраны в представлении «orders\_placed».

Менеджер распределяет товарные позиции в заказах по складам согласно их наличию: в таблице «Orders\_goods» поле «storehouses\_id». Остатки по товарным позициям по складам собраны в представлении «remains». В завершении устанавливает заказу статус «Готов к сборке». Если нужных позиций нет, заказ переводится в статус «Отменен». Статусы заказов перечислены приложении 1.

Для смены статуса используется процедура «change\_order\_status» (приложении 3).

### **Назначение комплектовщика (менеджер либо автоматически)**

Заказы готовые к сборке (в статусе «Готов к сборке») собраны в представлении «orders\_ready».

После распределения заказа по складам заказу назначается комплектовщик, менеджером либо регламентным заданием. Для этого предусмотрена процедура «set\_order\_picker» (приложение 3), она назначает заказу исполнителя и переводит заказ в статус «Выполняется». Если пользователь уже назначен на заказ, будет возвращена ошибка.

### **Сбор заказа нашей фирмой (комплектовщик)**

Сборка предполагается в прикладном решении, пользователю печатается сборочный лист с ячейками, из которых необходимо взять товар.

Процедура «assembly\_statement», приложение 3.

Комплектовщик, согласно сборочной ведомости, выбирает конкретную номенклатуру с конкретным сроком годности и из конкретной ячейки. Набранное количество заноситься в таблицу «Orders\_goods\_collected». Запрос «Сборка товарной позиции», приложение 2.

После сборки выбранным позициям заказа устанавливается «Выполнен». Процедура «pickup\_completed», приложение 3.

Когда все строки заказа переходят в статус «Выполнен», считается, что заказ передается в доставку, остатки списываются со склада.

### **Просмотр статистических данных**

«Остатки по складам» - представление «remains».

«Остатки на рабочем складе пользователя» - представление «remains\_on\_works\_storehouses».

Запрос «Заказы за период поступившие».

Запрос «Заказы, выполненные с разбивкой по комплектовщикам».

## Тестовый пример

Пример оформления операций в файле «Тестовый пример.sql».

ВЫПОЛНИТЬ СКРИПТ "тестовые данные.sql", ВНИМАНИЕ!!! Скрипт сотрет все данные в базе!

## Приложение 1

Таблица 1. Статусы заказов.

|  |  |
| --- | --- |
| **Ид** | **Описание** |
| 1 | Размещен |
| 2 | Готов к сборке |
| 3 | Собирается |
| 4 | Выполнен |
| 5 | Отменен |

Таблица 2. Коды ошибок.

|  |  |  |
| --- | --- | --- |
| **Нпп** | **Код ошибки** | **Описание** |
| 1 | '45001' | 'Статус не существует' |
| 2 | '45002' | 'Заказ не существует' |
| 3 | '45003' | 'Срок годности уже истек' |
| 4 | '45004' | 'Ячейка обязательна для заполнения' |
| 5 | '45005' | 'Склад не существует' |
| 6 | '45006' | 'Ячейка не существует' |
| 7 | '45007' | 'Товар не существует' |
| 8 | '45008' | 'Пользователь не существует' |
| 9 | '45009' | 'Пользователь уже назначен на другой заказ' |
| 10 | '45010' | 'У пользователя не задан склад' |
| 11 | '45011' | 'Вариант настроек не существует' |

## Приложение 2

**Файл «Основные запросы.sql»**

## Приложение 3

***«select\_options (options\_id, result)» -*** устанавливает текущий вариант настроек.

options\_id (BIGINT) – идентификатор настроек,

result (VARCHAR(200)) – результат выполнения

Пример:

set @result = '';

call select\_options (1, @result);

*select* @result;

***«select\_user (users\_id, user\_name, works\_storehouses\_id, result)» -*** создает, модифицирует и выбирает пользователя.

users\_id (BIGINT) – идентификатор пользователя,

user\_name(VARCHAR(150)) – имя пользователя,

works\_storehouses\_id (BIGINT) – идентификатор рабочего склада, для менеджера 0.

result (VARCHAR(200)) – результат выполнения

Пример:

set @result = '';

call select\_user(3, 'Гаврилов Иван Семёнович', 3, @result);

call select\_user(5, 'Лопатин Андрей Мирославович', 5, @result);

*select* @result;

***«modify\_remains (storehouses\_id, cells\_id, goods\_id, exp\_date, quantity, result)»*** приход или расход товара на склад.

storehouses\_id (bigint) – идентификатор склада,

cells\_id (bigint) – идентификатор ячейки,

goods\_id (bigint) – идентификатор товара,

exp\_date (varchar(15)) – срок годности товара,

quantity (SMALLINT) – количество (как положительное так и отрицательное),

result (varchar(200))) – результат выполнения

Пример:

*set @result = '';*

*call modify\_remains(1, 1, 1, '2022-01-04', 10, @result);*

*select @result;*

***«change\_order\_status (orders\_id, statuses\_id, result)»***изменение статуса заказа.

orders\_id (bigint) – идентификатор заказа,

statuses\_id (tinyint) – идентификатор статуса,

result (varchar(200))) – результат выполнения

Пример:

*set @result = '';*

*call change\_order\_status(4, 2, @result);*

*select @result;*

***«set\_order\_picker (orders\_id, users\_id)»*** - назначает комплектовщика заказу, переводит заказ в статус «Собирается»

orders\_id (bigint) – идентификатор заказа,

users\_id (BIGINT) – идентификатор пользователя,

result (varchar(200))) – результат выполнения.

Пример:

*set @result = '';*

*call set\_order\_picker(4, 3, @result);*

*select @result;*

*call set\_order\_picker(4, 5, @result);*

*select @result;*

***«assembly\_statement(result)»*** печатает сборочную ведомость для текущего пользователя, по тем заказам, которые ему назначены к сборке. Какой товар, из какой ячейки и в каком количестве нужно набрать и по какому заказу.

result (varchar(200))) – результат выполнения

***«pickup\_completed(orders\_id, users\_id, result)»*** завершает набор заказа, устанавливает строкам заказа статус «Выполнено», если весь заказ при этом собран списывает остатки со склада в таблице сборки сбрасывает флаг «is\_active».

orders\_id (bigint) – идентификатор заказа,

user\_id (bigint) – идентификатор комплектовщика,

result (varchar(200))) – результат выполнения

Пример:

*set @result = '';*

*call pickup\_completed(4, 3, @result);*

*call pickup\_completed(4, 5, @result);*

*select @result;*