**Database Document**

**Automated Store Replenishment Using Distributed Objects (AutoRep)**

**CS 441**

**Group members**

Anirudh Ramnath,

Gowdhaman Sadhasivam,

Karthik Hariharan,

Venkatesan Ravichandran

University of Illinois at Chicago

Fall 2014

Table of Contents

Entity – Relationship Model 3

Relational Database Tables 4

SCHEMA 4

Category 4

Department 4

Groups 4

ManagerPermissions 5

Person 5

PersonGroups 5

Product 6

Region 6

RepSuggestions 7

SKU 8

Store 9

Subcategory 9

Vendor 10

Warehouse 10

WarehouseSKU 10

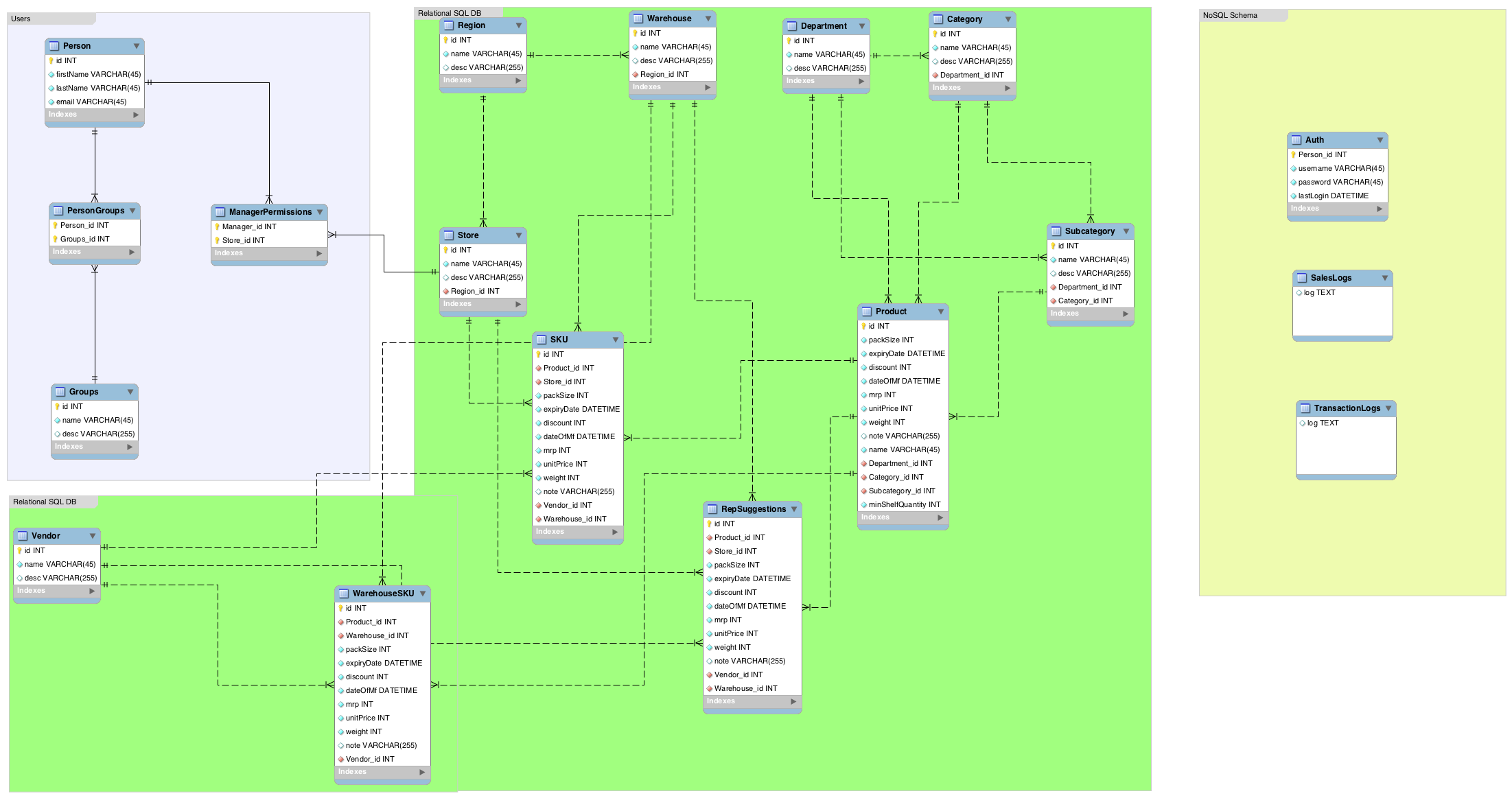
NoSQL Tables 11

Warehouse Transaction Logs 11

Store Transaction Logs 11

User authentication 11

# Entity – Relationship Model



# Relational Database Tables

## SCHEMA

CREATE DATABASE `autorep` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

## Category

The Category table is used to give a basic outline of the available product categories available in stores. This table is used to define classifications of products. It consists of details of categories such as the name, a description and the department that the category is associated with.

CREATE TABLE `category` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `name` varchar(45) NOT NULL,

 `desc` varchar(255) DEFAULT NULL,

 `Department\_id` int(11) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `Department\_id\_idx` (`Department\_id`),

 CONSTRAINT `Department\_id` FOREIGN KEY (`Department\_id`) REFERENCES `department` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Department

The Department table is used define the departments with which various product categories are associated to.

CREATE TABLE `department` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `name` varchar(45) NOT NULL,

 `desc` varchar(255) DEFAULT NULL,

 PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Groups

The Group table defines the access groups for the system to map with various users of the system. It consists of a description field for each access level.

CREATE TABLE `groups` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `name` varchar(45) NOT NULL,

 `desc` varchar(255) DEFAULT NULL,

 PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## ManagerPermissions

The ManagerPermissions table is used to map various Store Manager users of the system to their corresponding Stores.

CREATE TABLE `managerpermissions` (

 `Manager\_id` int(11) NOT NULL,

 `Store\_id` int(11) NOT NULL,

 PRIMARY KEY (`Manager\_id`,`Store\_id`),

 KEY `Store\_id\_idx` (`Store\_id`),

 CONSTRAINT `Manager\_id` FOREIGN KEY (`Manager\_id`) REFERENCES `person` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `Store\_id` FOREIGN KEY (`Store\_id`) REFERENCES `store` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Person

This table is used to define all the users of the system regardless of their role. It consists of generic details, which all users of the system will have.

CREATE TABLE `person` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `firstName` varchar(45) NOT NULL,

 `lastName` varchar(45) NOT NULL,

 `email` varchar(45) NOT NULL,

 PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## PersonGroups

This table is used to map various Person users in the system to their corresponding groups. These groups define the level of access that these users will have in the system.

CREATE TABLE `persongroups` (

 `Person\_id` int(11) NOT NULL,

 `Groups\_id` int(11) NOT NULL,

 PRIMARY KEY (`Person\_id`,`Groups\_id`),

 KEY `Groups\_id\_idx` (`Groups\_id`),

 CONSTRAINT `Groups\_id` FOREIGN KEY (`Groups\_id`) REFERENCES `groups` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `Person\_id` FOREIGN KEY (`Person\_id`) REFERENCES `person` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Product

The Product table is used to define all the details of various products available at stores. This table has a comprehensive list and includes fields such as, expiry date, discount, manufacturing date, weight, unit price, pack size and so on. This table will have references to the Department and Category tables in order to refer to the type of the Product being described.

CREATE TABLE `product` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `packSize` int(11) NOT NULL,

 `expiryDate` datetime NOT NULL,

 `discount` int(11) NOT NULL,

 `dateOfMf` datetime NOT NULL,

 `mrp` int(11) NOT NULL,

 `unitPrice` int(11) NOT NULL,

 `weight` int(11) NOT NULL,

 `note` varchar(255) DEFAULT NULL,

 `name` varchar(45) NOT NULL,

 `Department\_id` int(11) NOT NULL,

 `Category\_id` int(11) NOT NULL,

 `Subcategory\_id` int(11) NOT NULL,

 `minShelfQuantity` int(11) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `Category\_id\_idx` (`Category\_id`),

 KEY `Department\_id\_idx` (`Department\_id`),

 KEY `Subcat\_id\_idx` (`Subcategory\_id`),

 CONSTRAINT `Prod\_Category\_id\_fk` FOREIGN KEY (`Category\_id`) REFERENCES `category` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `Prod\_Department\_id\_fk` FOREIGN KEY (`Department\_id`) REFERENCES `department` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `Prod\_Subcat\_id` FOREIGN KEY (`Subcategory\_id`) REFERENCES `subcategory` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Region

The Region table consists of id, a unique identifier for regions. This table includes fields like name and description of the region also holds a reference to warehouse and store tables to identify the stores and warehouses of that particular region.

CREATE TABLE `region` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `name` varchar(45) NOT NULL,

 `desc` varchar(255) DEFAULT NULL,

 PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## RepSuggestions

This is a Replenishment suggestion table used to store all the products which reaches below its minimum threshold value and display this list to corresponding store manager using the reference Store\_id and its Warehouse\_id.

CREATE TABLE `repsuggestions` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `Product\_id` int(11) NOT NULL,

 `Store\_id` int(11) NOT NULL,

 `packSize` int(11) NOT NULL,

 `expiryDate` datetime NOT NULL,

 `discount` int(11) NOT NULL,

 `dateOfMf` datetime NOT NULL,

 `mrp` int(11) NOT NULL,

 `unitPrice` int(11) NOT NULL,

 `weight` int(11) NOT NULL,

 `note` varchar(255) DEFAULT NULL,

 `Vendor\_id` int(11) NOT NULL,

 `Warehouse\_id` int(11) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `Product\_id\_idx` (`Product\_id`),

 KEY `Vendor\_id\_idx` (`Vendor\_id`),

 KEY `Warehouse\_id\_idx` (`Warehouse\_id`),

 KEY `Store\_id\_idx` (`Store\_id`),

 CONSTRAINT `RepSug\_Product\_id` FOREIGN KEY (`Product\_id`) REFERENCES `product` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `RepSug\_Store\_id` FOREIGN KEY (`Store\_id`) REFERENCES `store` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `RepSug\_Vendor\_id` FOREIGN KEY (`Vendor\_id`) REFERENCES `vendor` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `RepSug\_Warehouse\_id` FOREIGN KEY (`Warehouse\_id`) REFERENCES `warehouse` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## SKU

Stock Keeping Unit(SKU) table used to keep track amount of products inventory and units of billable entities sold. This table consists of relative fields of products like product\_id, unitPrice, expiry Date, discount etc and relative fields of its location like Warehouse\_id, Store\_id.

CREATE TABLE `sku` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `Product\_id` int(11) NOT NULL,

 `Store\_id` int(11) NOT NULL,

 `packSize` int(11) NOT NULL,

 `expiryDate` datetime NOT NULL,

 `discount` int(11) NOT NULL,

 `dateOfMf` datetime NOT NULL,

 `mrp` int(11) NOT NULL,

 `unitPrice` int(11) NOT NULL,

 `weight` int(11) NOT NULL,

 `note` varchar(255) DEFAULT NULL,

 `Vendor\_id` int(11) NOT NULL,

 `Warehouse\_id` int(11) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `Product\_id\_idx` (`Product\_id`),

 KEY `Warehouse\_id\_idx` (`Warehouse\_id`),

 KEY `Store\_id\_idx` (`Store\_id`),

 KEY `Vendor\_id\_idx` (`Vendor\_id`),

 CONSTRAINT `Product\_id\_fk` FOREIGN KEY (`Product\_id`) REFERENCES `product` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `Store\_id\_fk` FOREIGN KEY (`Store\_id`) REFERENCES `store` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `Vendor\_id\_fk` FOREIGN KEY (`Vendor\_id`) REFERENCES `vendor` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `Warehouse\_id\_fk` FOREIGN KEY (`Warehouse\_id`) REFERENCES `warehouse` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Store

This table contains all the information about an individual store associated with an unique- id which holds the reference to the tables like SKU, Region, Replenishment suggestion

to identify its store using Store\_id.

CREATE TABLE `store` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `name` varchar(45) NOT NULL,

 `desc` varchar(255) DEFAULT NULL,

 `Region\_id` int(11) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `Region\_id\_idx` (`Region\_id`),

 CONSTRAINT `Region\_id` FOREIGN KEY (`Region\_id`) REFERENCES `region` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Subcategory

This table maintains the list of possible sub-categories a product can belong to. It maintains referential integrity with Department and Category tables using Department\_id and Category\_id foreign keys.

CREATE TABLE `subcategory` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `name` varchar(45) NOT NULL,

 `desc` varchar(255) DEFAULT NULL,

 `Department\_id` int(11) NOT NULL,

 `Category\_id` int(11) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `Department\_id\_idx` (`Department\_id`),

 KEY `Category\_id\_idx` (`Category\_id`),

 CONSTRAINT `Category\_id\_fk` FOREIGN KEY (`Category\_id`) REFERENCES `category` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `Department\_id\_fk` FOREIGN KEY (`Department\_id`) REFERENCES `department` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Vendor

This table stores a list of Vendors. Each SKU and Warehouse\_SKU is linked with a specific vendor, and that information is taken from this table.

CREATE TABLE `vendor` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `name` varchar(45) NOT NULL,

 `desc` varchar(255) DEFAULT NULL,

 PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Warehouse

This table stores a list of warehouses available. It links each warehouse with a specific region, by maintaining referential integrity with the Region table using the Region\_id field.

CREATE TABLE `warehouse` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `name` varchar(45) NOT NULL,

 `desc` varchar(255) DEFAULT NULL,

 `Region\_id` int(11) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `Region\_id\_idx` (`Region\_id`),

 CONSTRAINT `Region\_id\_fk` FOREIGN KEY (`Region\_id`) REFERENCES `region` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## WarehouseSKU

Warehouse Stock Keeping Unit(SKU) table used to keep track amount of products inventory and units of billable entities available in the warehouses. This table consists of relative fields of products like product\_id, unitPrice, expiry Date, discount etc and relative fields of its location like Warehouse\_id. When a product is replenished in a store, it is taken from the WarehouseSKU and moved to the SKU table.

CREATE TABLE `warehousesku` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `Product\_id` int(11) NOT NULL,

 `Warehouse\_id` int(11) NOT NULL,

 `packSize` int(11) NOT NULL,

 `expiryDate` datetime NOT NULL,

 `discount` int(11) NOT NULL,

 `dateOfMf` datetime NOT NULL,

 `mrp` int(11) NOT NULL,

 `unitPrice` int(11) NOT NULL,

 `weight` int(11) NOT NULL,

 `note` varchar(255) DEFAULT NULL,

 `Vendor\_id` int(11) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `Warehouse\_id\_idx` (`Warehouse\_id`),

 KEY `Vendor\_id\_idx` (`Vendor\_id`),

 KEY `Product\_id\_idx` (`Product\_id`),

 CONSTRAINT `WHSKU\_Product\_id\_fk` FOREIGN KEY (`Product\_id`) REFERENCES `product` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `WHSKU\_Vendor\_id\_fk` FOREIGN KEY (`Vendor\_id`) REFERENCES `vendor` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

 CONSTRAINT `WHSKU\_Warehouse\_id\_fk` FOREIGN KEY (`Warehouse\_id`) REFERENCES `warehouse` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

# NoSQL Tables

## Warehouse Transaction Logs

db.transaction\_Logs - Transaction log table in NoSQL

db.transaction\_Logs.insert(transaction\_id:”1”,transaction\_type:”update\_products”,created\_date:” 2014-18-10”,modified\_date:” 2014-18-10”,Duration:”154562”)

This table is used to maintain all replenishment request received from respective stores. For easy access to the logs, the details stored as JSON object.

## Store Transaction Logs

db.st\_transaction\_Logs - Store transaction log table in NoSQL

db.st\_transaction\_Logs.insert(store\_id:”1”,transaction\_type:”update\_products”,created\_date:”2014-18-10”,modified\_date:” 2014-18-10”,Duration:”154562”, store\_id:”store-123”,warehouse\_id:”ware-123”)

This is table is used to maintain transaction between store and warehouse. The logs are stored as JSON format to retrieve the information based on store id.

## User authentication

db.user\_authentication – User authentication table in NoSQL

db.user\_authentication.insert(user\_id:”sm\_1111”,user\_type:”store manager”, user\_name:”xyz\_123”,user\_email:”xyz@asd.com”, user\_password:”123jdgas13lasd\_\*&%asd”,store\_id:”store-123”,warehouse\_id:”ware-123”,)

This table is used to maintain user authentication details. The table has four major fields, store id, warehouse id, username and password.