

Dinein.co.uk – Database analysis

Document Information

|  |  |
| --- | --- |
| **Current Version** | 1.0 |
| **Document Owner** | Alex Popov,  Sergey Vodotyka,  Sasha Anisimov |
| **Status** | Draft |

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Created Date** | **Description** | **Author** |
| 1.0 | 01/20/2015 | Initial Release | Alex Popov,  Sergey Vodotyka,  Sasha Anisimov |

# Global modifications

Every table should have the following columns that will be used for data tracking:

* `record\_type` enum(‘Active’,’Inactive’,’Deleted’) NOT NULL DEFAULT ‘Active’,

Active record – the row is currently in use. Inactive record – the row is currently unused. Deleted – the row was deleted and can’t be restored by UI.

* `created\_on` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

The date when the row was created.

* `last\_update` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP.

The date when the row was updated. For the new row it will be as ‘created\_on’

All tables should have the default charset ‘utf8mb4’.

# Modifications by tables

## algorithms

#### Description:

Not in use

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

Do not transfer to the new database

## appdriverlocations

#### Description:

Contains info about driver’s physical location such as latitude and longitude.

#### Relations:

1. ONE-TO-ONE with ‘driverinfo’ (2.22)

#### Modifications:

Do not transfer to new DB

## appdriverlocationhistory

#### Description:

Driver location history

#### Modifications:

Do not transfer to new DB

## appdriverpushnotifications

#### Description:

Is used for tracking the push notification for drivers.

#### Relations:

1. MANY-TO-ONE with ‘driverinfo’ (2.22)

#### Modifications:

Do not transfer to new DB

## apprestheartbeat

#### Description:

#### Is used by Restaurant Apps to determine in the system which Applications are live

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.59)
2. MANY-TO-ONE with ‘users’ (2.80)

#### Modifications:

1. Rename to ‘app\_rest\_heartbeat’
2. Change columns:

* ‘restId’ to ‘restaurant\_id’
* ‘userId’ to ‘user\_id’
* ‘appVersion’ to ‘app\_version’

1. Add relations

## apprestpushnotifications

#### Description:

Used to push notifications to Restaurant Apps.

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.59)

#### Modifications:

1. Rename to ‘app\_restaurant\_push\_notifications’
2. Change columns:

* ‘restId’ to ‘restaurant\_id’

1. Add relations

## areaaddress

#### Description:

Contains the info of available address areas.

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

1. Rename to ‘area\_address’
2. Rename columns:

* ‘areaId’ to ‘id’,

1. Remove columns:

* ‘addedDate’,
* ‘areaSeoName’
* ‘status’
* ‘areadDesp’
* ‘areaAddr’

1. Add columns:

* ‘country\_id’ bigint(20) unsigned NOT NULL
* `name\_key’ varchar(50) NOT NULL,
* `native\_name’ varchar(50) NOT NULL,

1. Add relations:

* MANY-TO-ONE with ‘country’ (3.1)

## assigngroup

#### Description:

Contains the user’s groups.

#### Relations:

1. MANY-TO-MANY with ‘users’ (2.80)
2. MANY-TO-MANY with ‘usergroup’ (2.78)

#### Modifications:

1. Rename to ‘user\_user\_group’
2. Rename columns:

* ‘assignId’ to ‘id’
* ‘groupId’ to ‘group\_id’
* ‘userId’ to ‘user\_id’

1. Add relations
2. Remove ‘addDate’

## assignpickup

#### Description:

Contains the relation between drivers and orders

#### Modification:

Do not transfer to the new DB. If we need assign driver to order we could make it in ‘order’ (2.51) table.

## assignproject

#### Description:

Not in use

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

Do not transfer to the new database

## bestfor

#### Description:

Contains the list of ‘Best For’ items which is used to filter the restaurant list

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

1. Rename to ‘best\_for\_items’
2. Rename columns ‘bestforId’ to ‘id’
3. Remove columns:

* ‘bestSEOName’
* ‘bestDescription’

1. Add columns:

* ‘client\_id’ bigint(20) unsigned NOT NULL

1. Add relations:

* MANY-TO-ONE with ‘client’ (3.48)

## associateuser

#### Description:

Contains the relation between users.

#### Relations:

1. MANY-TO-MANY with ‘users’ (2.80)
2. MANY-TO-MANY with ‘users’ (2.80)

#### Modifications:

This table is ambiguous as all users are tied only to themselves (SELECT \* FROM associateuser where userId <> associateUser). So we don’t want to use it in the new DB.

## company

#### Description:

The list of companies for Corporate Accounts

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

1. Rename columns:

* ‘companyName’ to ‘name’
* ‘clientCode’ to ‘code’
* ‘paymentFrequency’ to ‘payment\_frequency’
* ‘paymentFrequencyAmount’ to ‘payment\_frequency\_amount’,
* ‘salesFee’ to ‘sales\_fee’,
* ‘vatType’ to ‘is\_vat\_exclusive’,
* ‘vatNumber’ to ‘vat\_number’,
* ‘daily’ to ‘dayly\_limit’,
* ‘weekly’ to ‘weekly\_limit’,
* ‘monthly’ to ‘monthly\_limit’,
* ‘limitType’ to ‘is\_limit\_hard’,
* ‘authorizeDomain’ to ‘authorize\_domain’,
* ‘minMTimeFrom` to ‘min\_m\_time\_from’,
* `minMTimeTo` to ‘min\_m\_time\_to’,
* ‘minM` to ‘min\_m’,
* ‘minETimeFrom` to ‘min\_e\_time\_from’,
* `minETimeTo` to ‘min\_e\_time\_to’,
* `minE` to ‘min\_e’,

1. Remove columns:

* ‘companyHouse’ (address)
* ‘address’
* ‘city’
* ‘postcode’
* ‘phone’,
* ‘clientName’ (company\_contact)
* ‘clientEmail’
* ‘clientPhone’
* ‘clientAddress’
* ‘clientCity’
* ‘clientPostCode’,
* ‘billingName’,
* ‘billingEmail’,
* ‘billingAddress’,
* ‘billingCity’
* ‘billingPostCode’,

1. Change columns:

* ‘vatType’ bit(1) NOT NULL DEFAULT 1,
* ‘limitType’ bit(1) NOT NULL DEFAULT 1,
* ‘paymentFrequency’ enum(‘Daily’,’Weekly’,’BiMonthly’,’Monthly’) NOT NULL

1. Add columns:

* ‘client\_id’ bigint(20) unsigned NOT NULL

1. Add relations:

* MANY-TO-ONE with ‘client’ (3.48)

## companydeliveryaddress

#### Description:

Contains the list of delivery addresses for particular company

#### Modification:

Will be replaced by ‘company\_address’ (3.6)

## contents

#### Description:

The list of static pages

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

Do not transfer to the new DB. Replaced by ‘page’ (3.19)

## conversion\_history

#### Discuss:

What is BI for this table? Do we need it in the new DB?

Deleted

## corporateorder

#### Description:

Contains information related to corporate orders

#### Relations:

1. MANY-TO-ONE with ‘users’ (2.80) - removed
2. MANY-TO-ONE with ‘project’ (2.58)
3. MANY-TO-ONE with ‘expensetype’ (2.26)
4. MANY-TO-ONE with ‘order’ (3.12)

#### Modifications:

1. Rename to ‘corporate\_order’
2. Change columns:

* ‘userId’ to ‘user\_id’
* ‘projectCode’ to ‘project\_code’
* ‘projectId’ to ‘company\_project\_id’
* ‘exptypeid’ to ‘company\_expense\_id’
* ‘limitType’ to ‘limit\_type’ enum(‘Soft’,’Hard’) NULL,
* ‘limitMax’ to ‘limit\_max’,
* ‘allocationLimit’ to ‘allocation\_limit’
* ‘totalAllocated’ to ‘total\_allocated’

1. Remove columns:

* ‘projectCode’ (we have it in project table)
* ‘orderNumber’ (we have it in order table)
* ‘loginUserId’ (we have it in user table)
* ‘firstName’ (we have it in order table)
* ‘lastName’ (we have it in order table)
* ‘email’ (we have it in order table),
* ‘cardPayment’ (we have it in order table),
* ‘companyId’ (we have it in project table),
* ‘companyName’ (we have it in company table)
* ‘orderStatus’ (we have it in order table)
* ‘date’ (we have it in order table)

1. Add columns:

* ‘order\_id’ bigint(20) unsigned NOT NULL

1. Add relations

## cuisine

#### Description:

The list of cuisines

#### Relations:

1. MANY-TO-ONE with ‘restGroup’ (2.61) – should be removed

#### Modifications:

1. Change column names:

* ‘cuisineName’ to ‘name\_key’
* ‘restGroupId’ to ‘restaurant\_group\_id’
* ‘cuisineDescription’ to ‘description\_key’

1. Remove columns:

* ‘addDate’
* ‘cuisineSEOName’

1. Add columns:

* ‘client\_id’ bigint(20) unsigned NOT NULL

1. Add relations:

* MANY-TO-ONE with ‘client’ (3.48)

## defaultdeliverycharges

#### Description:

The list of charges depended on the distance from restaurant.

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

Do not transfer to the new DB

## deliverycharges

#### Description:

The list of charges assigned to the particular restaurant.

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.59)

#### Modifications:

Do not transfer to the new DB

## deliverycredits

#### Description:

Looks like unused.

#### Modifications:

Do not transfer to the new DB.

## driverinfo

#### Description:

Looks like unused.

#### Modifications:

Do not transfer to the new DB.

## driverpool

#### Description:

The pool of drivers assigned to the restaurant group.

#### Relations:

1. MANY-TO-ONE with restGroup (2.61)

#### Modifications:

Do not transfer to the new DB.

## email\_templates

#### Description:

Looks like unused.

#### Modifications:

Do not transfer to the new DB.

## expensename

#### Description:

The list of available expense types

#### Relations:

1. MANY-TO-ONE with company (2.13)

#### Modifications:

1. Rename to ‘expense\_type’
2. Rename column ‘exptypeId’ to ‘id’
3. Remove columns and relations:

* ‘companyId’
* ‘addDate’

## expensetype

#### Description:

Contains the expenses tied to the company

#### Relations:

1. ONE-TO-ONE with ‘expensename’ (2.25)
2. MANY-TO-ONE with ‘usergroup’ (2.78)

#### Modifications:

1. Rename to ‘company\_expense\_type’
2. Rename columns:

* ‘assignId’ to ‘id’
* ‘exptypeId’ to ‘expense\_type\_id’
* ‘groupId’ to ‘user\_group\_id’
* ‘limitOrder’ to ‘limit\_order’
* ‘limitType’ to ‘limit\_type’
* ‘softLimitMax’ to ‘soft\_limit\_max’

1. Remove columns:

* addDate,

1. Change columns:

* ‘limitType’ enum(‘Soft’,’Hard’) NULL

1. Add relations

## expensetypeopening

#### Description:

Contains information when the expenses are effective

#### Relations:

1. MANY-TO-ONE with ‘expensetype’ (2.26)

#### Modifications:

1. Rename to ‘expense\_type\_schedule’
2. Rename columns:

* ‘assignId’ to ‘expense\_type\_id’

1. Add columns:

* ‘schedule\_id’ bigint(20) NOT NULL,

1. Add primary key for ‘expense\_type\_id’ and ‘schedule\_id’
2. Add relations:

* MANY-TO-ONE with ‘schedule’ (3.28)

1. Remove columns:

* ‘openingid’
* ‘openDay’
* ‘openDay’
* ‘delMTimeFrom’
* ‘delMTimeTo’
* ‘delETimeFrom’
* ‘delETimeto’

1. Add relations.

## faqcatagory

#### Description:

The list of FAQ categories

#### Relation:

1. MANY-TO-ONE with ‘restGroup’ (2.61)

#### Modification:

Do not transfer to the new DB.

## faqquestion

#### Description:

The list of FAQ and answers

#### Relations:

1. MANY-TO-ONE with ‘faqcatagory’ (2.28)
2. MANY-TO-ONE with ‘faqsubcatagory’ (2.30)

#### Modifications:

Do not transfer to the new DB

## faqsubcategory

#### Description:

The list of FAQ sub categories

#### Relations:

1. MANY-TO-ONE with ‘faqcatagory’ (2.28)

#### Modifications:

Do not transfer to the new DB

## feedback

#### Description:

The users feedback

#### Modifications:

It looks like unused. So do not transfer to the new DB

## feedbacksite

#### Description:

The user’s feedback

#### Modifications:

It looks like unused. So do not transfer to the new DB

## grorddetail

#### Description:

Menu items related to group order users

## grordmember

#### Description:

Contains the users assigned to the orders

#### Relations:

1. MANY-TO-ONE with ‘grouporder’ (2.35)
2. MANY-TO-ONE with ‘users’ (2.80) – should be replaced by MANY-TO-ONE with ‘contact’ (3.9)

#### Modifications:

1. Rename to ‘group\_order\_contact’
2. Rename columns:

* ‘grodid’ to ‘group\_order\_id’
* ‘userId’ to ‘contact\_id’

1. Remove columns:

* ‘grodmemId’
* ‘orderNumber’
* ‘name’
* ‘email’

1. Add relations

## grouporder

#### Description:

Contains the list of group orders that was made.

#### Relations:

1. MANY-TO-ONE with ‘users’ (2.80) – should be replaced by MANY-TO-ONE with ‘contact’ (3.9)
2. MANY-TO-ONE with ‘restaurant’ (2.59)

#### Modifications:

All this info should be in ‘order’ table.

## interimmeta

#### Description:

Contains temporary order data about order until it’s not ordered. After the order it looks like the data is deleted from this table

#### Modifications:

Do not transfer to the new DB. Cache will be responsible for it. Also we added the status to ‘order’ table ‘InProgress’

## interimmeta\_trans

#### Description:

Looks like unused. No data there.

#### Modifications:

Do not transfer to the new DB.

## interimorder

#### Description:

Contains temporary order data about order until it’s not ordered. After the order it looks like the data is deleted from this table

#### Modifications:

Do not transfer to the new DB. Cache will be responsible for it

## interimorder\_trans

#### Description:

Looks like unused. No data there.

#### Modifications:

Do not transfer to the new DB.

## ipvote

#### Discuss:

What is the BI for it? Do we need to transfer it to the new DB?

## likemenu

#### Description:

The liked menu items (basically favorite menus)

#### Relations:

1. MANY-TO-ONE with ‘user’ (2.80)
2. MANY-TO-ONE with ‘restaurant’ (2.59) – will be removed
3. MANY-TO-ONE with ‘menu’ (2.44)

#### Modifications:

1. Rename to ‘menu\_item\_like’
2. Rename columns:

* ‘userId’ to ‘user\_id’
* ‘menuItemId’ to ‘menu\_item\_id’

1. Remove columns:

* ‘likeMenuId’
* ‘restId’
* ‘likeDate’

1. Add unique key for ‘user\_id’ and ‘menu\_item\_id’
2. Add relations

## likesrestaurant

#### Description:

The liked restaurants (favorite restaurants)

#### Relations:

1. MANY-TO-ONE with ‘user’ (2.80)
2. MANY-TO-ONE with ‘restaurant’ (2.59)

#### Modifications:

1. Rename to ‘restaurant\_like’
2. Rename columns:

* ‘restId’ to ‘restaurant\_id’
* ‘userId’ to ‘user\_id’

1. Remove columns:

* ‘likeRestId’
* ‘likeDate’

1. Add unique key for ‘user\_id’ and ‘restaurant \_id’
2. Add relations

## mailinglist

#### Discuss:

What is the BI for it? Do we need it in the new DB?

## menu

#### Description:

The list of available menu items

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.59)
2. MANY-TO-ONE with ‘subMenu’ (2.70) – will be removed
3. MANY-TO-ONE with ‘menuTypeName’ (2.46) – will be removed
4. MANY-TO-ONE with ‘vat’ (2.82)

#### Modifications:

1. Rename to ‘menu\_item’
2. Add columns:

* ‘menu\_id’ bigint(20) unsigned NOT NULL,
* ‘nutritional’ varchar(500) NULL,
* ‘menu\_allergy\_id’ bigint(20) unsigned NOT NULL
* ‘menu\_option\_id’ bigint(20) unsigned NULL
* ‘is\_imported’ bit(1) NOT NULL Default 0

1. Remove columns:

* ‘restId’ (will be in ‘menu’ table)
* ‘addDate’,
* ‘subMenuId’
* ‘menuTypeId’
* ‘status’

1. Change column names:

* ‘menuId’ to ‘id’
* ‘menuName’ to ‘name\_key’
* ‘price’ to ‘resaurant\_price’
* ‘webPrice’ to ‘web\_price’
* ‘sortOrder to ‘order’

1. Add existing and following relations:

* MANY-TO-ONE with ‘menu’ (3.23)
* MANY-TO-ONE with ‘menu\_allergy’ (3.27)
* MANY-TO-ONE with ‘menu\_option’ (4.1)

## menutype

#### Description:

The list of menus related to the restaurant

#### Modification:

Will be replaced by ‘menu’ (3.23)

## menutypename

#### Description:

The list of menu types (breakfast, lunch, dinner, etc.)

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

1. Rename to ‘menu\_type’
2. Rename columns:

* ‘menuTypeId’ to ‘id’
* ‘menuTypeName’ to ‘name’

1. Remove columns:

* ‘addedDate’
* ‘oneMenuType’

1. Add columns:

* ‘client\_id’ bigint(20) unsigned NOT NULL
* ‘schedule\_id’ bigint(20) unsigned NOT NULL

1. Add relations:

* MANY-TO-ONE with ‘client’ (3.48)
* MANY-TO-ONE with ‘schedule’ (3.28)

## openingdetails

#### Description:

The time when restaurant works and delivery time

#### Modifications:

Will be moved to ‘restaurant\_schedule’ (3.29)

## order\_ivr\_history

#### Description:

The history of orders which was made by IVR calls

#### Relations:

1. MANY-TO-ONE with ‘order’(3.12)

#### Modifications:

1. Delete columns:

* ‘orderNumber’
* ‘calldatetime’
* ‘restGroupId’

1. Add columns:

* ‘order\_id’ bigint(20) unsigned NOT NULL

1. Change columns:

* ‘callstatus’ to ‘call\_status’ enum(‘Queued’,’Completed’,NoAnswer’,’Busy’,’Failed’) NOT NULL
* ‘calledno’ to ‘phone\_number’
* ‘duration’

1. Add relations

## ordercount

#### Description:

The number of orders per day

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

Do not transfer to the new DB

## orderhistory

#### Description:

The history of orders (Tracks the Dispatcher and order status history)

#### Relations:

1. MANY-TO-ONE with ‘order’ (3.12)

#### Modifications:

1. Rename to ‘order\_history’
2. Delete columns:

* ‘userId’
* ‘orderNumber’

1. Add columns:

* ‘order\_id’

1. Change columns:

* ‘hisotryoforder’ to ‘history’
* ‘vsms’ to ‘v\_sms’

1. Add relations

#### Discuss:

What is ‘vsms’?

## orders

#### Description:

Contains the ordered menu items (Historical data copy for reports).

#### Relations:

1. MANY-TO-ONE with ‘menu\_item’ (2.44)
2. MANY-TO-ONE with ‘users’ (2.80) – should be removed

#### Modifications:

1. Rename to ‘order\_item’
2. Change columns:

* ‘orderId’ to ‘id’
* ‘menuId’ to ‘menu\_item\_id’
* ‘price’ to ‘web\_price’
* ‘restPrice’ to ‘restaurant\_price’

1. Remove columns:

* ‘addDate’
* ‘restId’
* ‘menuName’
* ‘userType’
* ‘description’
* ‘unitPrice’
* ‘unitRestPrice’
* ‘orderNumber’
* ‘restItemNo’
* ‘vatAmt’
* ‘userId’

1. Add columns:

* ‘oder\_id’ bigint(20) unsigned NOT NULL,
* ‘vat\_id’ bigint(20) unsigned NOT NULL

1. Add existing relations and following:

* MANY-TO-ONE with ‘order’ (3.12)
* MANY-TO-ONE with ‘vat’ (2.82)

## ordersmeta

#### Description:

Contains information about al orders ((Historical data copy for reports).

#### Modification:

Will be replaced by ‘order’ (3.12) table

## paymenttext

#### Description:

Contains the payment costs

#### Modification:

Will be replaced by ‘order\_payment\_cost’ (3.14)

## postcode

#### Description:

The list of postcodes

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

1. Rename columns:

* ‘poscodeId’ to ‘id’
* ‘postcode’ to ‘code’

1. Add columns

* `country\_id` smallint(5) unsigned NOT NULL,
* `area\_address\_id` int(10) unsigned DEFAULT NULL,

1. Add relations:

* MANY-TO-ONE with ‘area\_address’ (2.7)
* MANY-TO-ONE with ‘country’ (3.1)

1. Change columns types:

* `latitude` decimal(18,15) DEFAULT NULL,
* `longitude` decimal(18,15) DEFAULT NULL,

## postcode\_copy

#### Description:

Copy of ‘poscode’ table

#### Modification:

Do not transfer to the new DB.

## pramotion

#### Description:

The list of promo codes/vouchers.

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.60)

#### Modifications:

1. Rename to ‘voucher’
2. Change columns:

* ‘pracodeId’ to ‘id’
* ‘codeValue’ to ‘code’
* ‘discountValue’ to ‘value’
* ‘restId’ to ‘restaurant\_id’
* ‘startDate’ to ‘start\_date’
* ‘endDate’ to ‘end\_date’
* ‘valueType’ to ‘value\_type’ enum(‘Fixed’,’Percent’)
* ‘priceValue’ to ‘value’
* ‘itemQuantity’ to ‘quantity’
* ‘orderAfter’ to ‘order\_after’
* ‘maxTimePerUser’ to ‘limit\_per\_user’
* ‘generateBy’ to ‘generate\_by’ enum(‘M’,’C’)

1. Remove columns:

* ‘praStatus’
* ‘addDate’
* ‘pramotionCat’
* ‘promotionType’

1. Add columns:

* ‘total\_limit’ int(10) unsigned NULL

1. Add relations

## pramotionmenu

#### Description:

Promo codes assigned to the menu items

#### Relations:

1. MANY-TO-ONE with ‘menu\_item’ (2.44)
2. MANY-TO-ONE with ‘pramotion’ (2.56)
3. MANY-TO-ONE with ‘restaurant’ (2.60) – should be removed

#### Modifications:

1. Rename to ‘vaucher\_menu\_item’
2. Change columns:

* ‘promoMenuId’ to ‘id’
* ‘menuId’ to ‘menu\_id’
* ‘buyQuantity’ to ‘quantity’
* ‘pracodeId’ to ‘vaucher\_id’

1. Remove columns:

* ‘restId’
* ‘discType’
* ‘dicValue’ (we have it in ‘voucher’ table)

1. Add relations

## project

#### Description:

Contains information about company projects for corporate orders

#### Relations:

1. MANY-TO-ONE with ‘company’ (2.13)
2. MANY-TO-ONE with ‘users’ (2.80)

#### Modifications:

1. Rename to ‘company\_project’
2. Change columns:

* ‘projectId’ to ‘id’
* ‘userId’ to ‘user\_id’
* ‘companyId to ‘company\_id’
* ‘projectCode’ to ‘code’
* ‘projectName’ to ‘name’
* ‘daily’ to ‘daily\_limit’
* ‘weekly’ to ‘weekly\_limit’
* ‘monthly’ to ‘monthly\_limit’
* ‘limitType’ to ‘limit\_type’ enum(‘Soft’,’Hard’) NOT NULL

1. Remove columns:

* ‘clientId’
* ‘addDate’

1. Add relations

## ratings

#### Description:

The rating of restaurants

#### Relations:

1. MANY-TO-ONE with ‘order’ (3.12) – will be removed
2. MANY-TO-ONE with ‘restaurant’ (2.60) – will be removed
3. MANY-TO-ONE with ‘users’ (2.80) – will be removed
4. MANY-TO-ONE with ‘review’ (2.66)

#### Modifications:

1. Rename to ‘rating’
2. Change columns:

* ‘rateId’ to ‘id’
* ‘reviewId’ to ‘review\_id’

1. Remove columns:

* ‘rateDate’
* ‘orderId’ (we have it in review table)
* ‘restId’ (we have it in review table)
* ‘userId’ (we have it in review table)

1. Add relations

## restaurant

#### Description:

Contains the restaurant list

#### Relations:

1. MANY-TO-ONE with ‘restGroup’ (2.61)
2. MANY-TO-ONE with ‘setbasepostcode’ (2.67)

#### Modifications:

1. Change columns:

* ‘companyName’ to ‘name’
* ‘SEOComName’ to ‘slug’
* ‘paymentMode’ to ‘payment\_method\_id’
* ‘avgPrepareTime’ to ‘avg\_prepare\_time’
* ‘minOrder’ to ‘min\_order\_amount’
* ‘fixCharge’ to ‘fix\_charge’
* ‘isNewest’ to ‘is\_newest’
* ‘isRestOfWeek’ to ‘is\_rest\_of\_week’
* ‘restOfWeekDate’ to ‘restOfWeekDate’
* ‘endDayReportDate’ to ‘end\_day\_report\_date’
* ‘restGroupId’ to ‘restaurant\_group\_id’
* ‘seoTitle’ to ‘seo\_title\_key’
* ‘metaText’ to ‘seo\_text\_key’
* ‘metaDesc’ to ‘seo\_description\_key’
* ‘basecodeid’ to ‘address\_base\_id’
* ‘haveApp’ to ‘have\_app’ bit(1) NOT NULL

1. Remove columns:

* ‘companyName’
* ‘openTimeFrom’
* ‘openTimeTo’
* ‘conFirstName’
* ‘conLastName’
* ‘conPhone’
* ‘conEmail’
* ‘conRole’
* ‘cuisine’
* ‘openingDay’
* ‘restImage’
* ‘restStatus’
* ‘addDate’
* ‘mobPhone1’
* ‘mobPhone2’
* ‘mobPhone3’
* ‘email1’
* ‘email2’
* ‘email3’
* ‘fax1’
* ‘fax2’
* ‘fax3’
* ‘bestFor’
* ‘aboutDelivery’
* ‘serviceType’
* ‘collectApprox’
* ‘deliveryChargeType’
* ‘avgMenuPrice’
* ‘deliveryRange’
* ‘restaurantType’
* ‘tradingName’
* ‘restCity’
* ‘comHouseNo’
* ‘restPhone’
* ‘billFirstName’
* ‘billLastName’
* ‘billPhone’
* ‘billEmail’
* ‘billRole’
* ‘phone1Name’
* ‘phone1’
* ‘phone1Alert’
* ‘phone2Name’
* ‘phone2’
* ‘phone2Alert’
* ‘phone3Name’
* ‘phone3’
* ‘phone3Alert’
* ‘mob1Name’
* ‘mob1Charge’
* ‘mob1Alert’
* ‘mob2Name’
* ‘mob2Charge’
* ‘mob2Alert’
* ‘mob3Name’
* ‘mob3Charge’
* ‘mob3Alert’
* ‘email1Name’
* ‘email1Alert’
* ‘email2Name’
* ‘email2Alert’
* ‘email3Name’
* ‘email3Alert’
* ‘fax1Location’
* ‘fax1Alert’
* ‘fax2Location’
* ‘fax2Alert’
* ‘fax3Location’
* ‘fax3Alert’
* ‘accHolderName’
* ‘bankName’
* ‘sortCoder’
* ‘accNumber’
* ‘paypalEmail’
* ‘creditCardNumber’
* ‘creditStartDate’
* ‘creditExpDate’
* ‘creditSecCode’
* ‘mapLink’
* ‘paymentFrequncy’
* ‘paymentFrequencyAmt’
* ‘userId’
* ‘restGrAdmin’
* ‘salesFee’
* ‘salesFeeCol’
* ‘chargeMethodCol’
* ‘chargeMethod’
* ‘latitudeValue’
* ‘longitudeValue’
* ‘mms1Charge’
* ‘mms2Charge’
* ‘mms3Charge’
* ‘mmsdelay1’
* ‘mmsdelay2’
* ‘mmsdelay3’
* ‘area’
* ‘vatEx’
* ‘vatExCol’
* ‘vatNo’
* ‘deliverymodel’
* ‘deliveryalgo’
* ‘defaultFoodPrepTime’
* ‘currentFoodPrepTime’
* ‘appHeartbeat’
* ‘driverDelayTime’
* ‘ivr\_phone\_1’
* ‘ivr\_phone\_2’
* ‘ivr\_phone\_3”
* ‘ivr\_1\_alert’
* ‘ivr\_2\_alert’
* ‘ivr\_3\_alert’
* ‘ivr\_delay\_1’
* ‘ivr\_delay\_2’
* ‘ivr\_delay\_3’
* ‘ivr\_ph\_charge\_1’
* ‘ivr\_ph\_charge\_2’
* ‘ivr\_ph\_charge\_3’
* ‘ivr\_role\_name\_1’
* ‘ivr\_role\_name\_2’
* ‘ivr\_role\_name\_3’

1. Add columns:

* ‘address\_base\_id’ bigint(20) unsigned NOT NULL
* ‘is\_pre\_signed’ bit(1) NOT NULL Default 0
* ‘is\_featured’ bit(1) NOT NULL Default 0

1. Add relations:

* MANY-TO-ONE with ‘address\_base’ (3.3)
* MANY-TO-ONE with ‘payment\_method’ (3.55)

## restgroup

#### Description:

The list of all restaurant groups

#### Relations:

1. MANY-TO-ONE with ‘users’ (2.80) – will be removed

#### Modification:

1. Rename to ‘restaurant\_group’
2. Change columns:

* ‘group\_name’ to ‘name’

1. Remove columns:

* ‘admin\_id’

1. Add columns:

* ‘restaurant\_chain\_id’ bigint(20) NOT NULL,

1. Add relations:

* MANY-TO-ONE with ‘restaurant\_chain’ (3.30)

## restphoto

#### Description:

The list of all restaurant photos

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.60)

#### Modification:

1. Rename to ‘restaurant\_photo’
2. Change columns:

* ‘photoId’ to ‘id’
* ‘restId’ to ‘restaurant\_id’
* ‘imageName’ to ‘image\_name’
* ‘title’ to ‘title\_key’
* ‘photoOrderNo’ to ‘order’

1. Remove columns:

* ‘addDate’

1. Add columns:

* ‘is\_default’ bit(1) NOT NULL,

1. Add relations

## restsignup

#### Description:

It looks like this table contains the same data as the ‘restaurant’

#### Modification:

Do not transfer to the new DB. We will use flag ‘is\_from\_signup’ in ‘restaurant’ table (2.60)

## restsnaps

#### Description:

It looks like this table contains the same data as the ‘restPhoto’ (2.62)

#### Discuss:

What is the BI for this table? Do we need to move it to the new DB?

## reststats

#### Description:

The restaurant statuses

#### Modification:

Do not transfer to the new DB.

## review

#### Description:

Contains the restaurant reviews

#### Relations:

1. MANY-TO-ONE with ‘order’ (3.12)
2. MANY-TO-ONE with ‘restaurant’ (2.60)
3. MANY-TO-ONE with ‘users’ (2.80) – will be replaced by MANY-TO-ONE with ‘contact’ (3.9)

#### Modifications:

1. Rename to ‘feedback’
2. Change columns:

* ‘reviewId’ to ‘id’
* ‘reviewTitle’ to ‘title’
* ‘restId’ to ‘restaurant\_id’
* ‘city’ to ‘city\_id’
* ‘userId’ to ‘contact\_id’
* ‘orderId’ to ‘order\_id’

1. Remove columns:

* ‘voterName’ (we have it in user table)
* ‘reviewDate’
* ‘status’

1. Add columns

* ‘feedback\_type\_id’ bigint(20) NOT NULL

1. Add existing and following relations:

* MANY-TO-ONE with ‘feedback\_type’ (3.54)

## setbasepostcode

#### Description:

The list of all address bases.

#### Modification:

Will be replaced by ‘address\_base’ (3.3)

## smscost

#### Description:

The cost of sms sending

#### Relations:

No MANY-TO-ONE relations

#### Modification:

1. Rename to ‘sms\_cost’
2. Change columns:

* ‘smsCost’ to ‘cost’

## smsrecord

#### Description:

The tracking of all sent sms

#### Relations:

1. MANY-TO-ONE with ‘order’ (2.51)
2. MANY-TO-ONE with ‘restaurant’ (2.60)

#### Modification:

1. Rename to ‘sms\_history’
2. Change columns:

* ‘smsrecordId to ‘id’
* ‘orderNo’ to ‘order\_id’
* ‘smsType’ to ‘sms\_type’ enum(‘R’,’D’) NOT NULL
* ‘smsCharge’ to ‘charge’

1. Remove columns:

* ‘sendDate’
* ‘restId’

1. Add relations

## submenu

#### Description:

The list of available menu categories

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.60) – should be removed (every ‘menu\_item’ has reference to it)
2. MANY-TO-ONE with ‘menutype’ (2.45) – should be removed

#### Modifications:

1. Rename to ‘menu\_category’
2. Change columns:

* ‘submenuId’ to ‘id’
* ‘subMenu’ to ‘name’

1. Remove columns:

* ‘restId’
* ‘addDate’
* ‘menutypeId’

1. Add column

* ‘client\_id’ bigint(20) unsigned NOT NULL
* ‘is\_option’ bit(1) NOT NULL Default 0

1. Add relations:

* MANY-TO-ONE with ‘client’ (3.48)

## suggestedrestaurants

#### Description:

Contains the list of restaurants which suggested by user

#### Modification:

It looks like duplicate of suggestrestaurant (2.74)

## suggester

#### Description:

Contains emails of person who suggest the restaurant

#### Modification:

The table ‘contact’ (3.9) will be used instead of it.

## suggestrestarea

#### Description:

Contains the suggested restaurant address

#### Modification:

The table ‘address’ (3.2) will be used instead of it.

## suggestrestaurant

#### Description:

Contains the restaurants which suggested by user

#### Relations:

1. MANY-TO-ONE with ‘cuisine’ (2.18)

#### Modification:

1. Rename to ‘restaurant\_suggested’
2. Change columns:

* ‘suggRestId’ to ‘id’
* ‘sugRestName’ to ‘name’
* ‘cuisineId’ to ‘cuisine\_id’

1. Remove columns:

* ‘restArea’
* ‘restPhoneNo’
* ‘restPostCode’
* ‘suggEmail’
* ‘addDate’
* ‘sugRank’

1. Add columns:

* ‘address\_id’ bigint(20) unsigned NOT NULL
* ‘contact\_id’ bigint(20) unsigned NOT NULL

1. Add existing and following relations:

* MANY-TO-ONE with ‘address’ (3.2)
* MANY-TO-ONE with ‘contact’ (3.9)

## suggrestranking

#### Description:

Contains the ranks of suggested restaurants

#### Relations:

1. MANY-TO-ONE with ‘suggestrestaurant’ ()

#### Modifications:

1. Rename to ‘resturant\_suggested\_rank’
2. Change columns:

* ‘rankingId’ to ‘id’
* ‘suggRestId’ to ‘restaurant\_suggested\_id’
* ‘ranking’ to ‘rank’

1. Remove columns:

* ‘rankingDate’

1. Add relations

## unique\_offer

#### Description:

Contains the data which is similar to ‘users\_offer’ table (2.81).

#### Modification:

The ‘users\_offer’ table (2.81) will be used instead of it.

## useraddress

#### Description:

Contains the list of user addresses

#### Modification:

Table ‘user\_contact’ (3.11) will be user instead of it.

## usergroup

#### Description:

The list of user group for corporate accounts.

#### Relations:

1. MANY-TO-ONE with ‘company’ (2.13)

#### Modifications:

1. Rename to ‘company\_user\_group’
2. Change columns:

* ‘groupName’ to ‘name’
* ‘ordCounter’ to ‘order\_counter’

1. Add columns
2. Delete columns:

* ‘codeId’ (will be in ‘user\_group\_code’ table )
* ‘addDate’

1. Add existing and following relations:

* MANY-TO-ONE with ‘client’ (3.48)

## userpromocodehistory

#### Description:

The history of vouchers applying

#### Relations:

1. MANY-TO-ONE with ‘users’ (2.80) – should be replaced by MANY-TO-ONE with ‘contact’ (3.9)

#### Modifications:

1. Rename to ‘voucher\_use\_history’
2. Add columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘voucher\_id’ bigint(20) unsigned NOT NULL

1. Change columns:

* ‘userId’ to ‘contact\_id’

1. Remove columns:

* ‘usedPromoCodes’

1. Add existing and following relations:

* MANY-TO-ONE with ‘pramotion’ (2.56)

## users

#### Description:

The user list that have access to the DineIn system

#### Relations:­­­­­­­

1. MANY-TO-ONE with ‘company’ (2.13)

#### Modifications:

1. Rename to ‘user’
2. Change columns:

* ‘userId’ to ‘id’
* ‘userName’ to ‘username’
* ‘userType’ to ‘role’ enum(‘Admin’,’User’,’RestaurantOwner’,RestaurantApp’,’CorporateUser’,’RestaurantTeam’,’Dispatcher’,’CorporateAdmin’,’RestaurantGroupAdmin’,’Finance’)
* ‘lasvisit’ to ‘last\_visit\_date’
* ‘activationCode’ to ‘activation\_hash’
* ‘profilePhoto’ to ‘photo’
* ‘knowAbout’ to ‘know\_about’ enum(‘‘)
* ‘emailPreferance’ to ‘email\_preferance’
* ‘termCondCheck’ to ‘term\_and\_cond’ bit(1) NOT NULL
* ‘termCondWebUse’ to ‘term\_and\_cond\_web’ bit(1) NOT NULL
* ‘termCondAccPol’ to ‘term\_and\_cond\_acc\_pol’ bit(1) NOT NULL
* ‘companyId’ to ‘company\_id’
* ‘orderAlert’ to ‘order\_alert’
* ‘apiToken’ to ‘api\_token’

1. Remove columns:

* ‘emailId’
* ‘country’
* ‘registerDate’
* ‘pool\_id’
* ‘status’
* ‘modifiedDate’
* ‘paypalEmail’
* ‘biography’
* ‘hobby’
* ‘firstName’
* ‘lastName’
* ‘city’
* ‘address’
* ‘postcode’
* ‘homeTel’
* ‘mobileNo’
* ‘address1’
* ‘address2’
* ‘address3’
* ‘title’
* ‘addressName’
* ‘addressId’
* ‘work\_phone’
* ‘confirmPassword’
* ‘confirmEmail’
* ‘gugLatUsername’
* ‘gugLatPassword’
* ‘loginStatus’
* ‘ownbikecar’
* ‘vehicle’
* ‘make’
* ‘pizzabox’
* ‘phonerecmail’
* ‘deleveryexp’
* ‘phchargattach’
* ‘restaurantId’
* ‘restauranGroupId’
* ‘forgetPassStatus’

1. Add columns:

‘affiliate\_id’ bigint(20) unsigned NULL

1. Add existing and following relations:

* MANY-TO-ONE with ‘affeliate’ (3.50)

## users\_offer

#### Description:

Contains the vouchers assigned to the users.

#### Relations:

1. MANY-TO-ONE with ‘pramotion’ (2.56)
2. MANY-TO-ONE with ‘users’ (2.80)

#### Modifications:

1. Rename to ‘voucher\_user’
2. Change columns:

* ‘userId’ to ‘user\_id’
* ‘emarkType’ to ‘emark\_type’ enum (‘NeverOrder’,’LastVisit’)
* ‘applicableOffer’ to ‘applicable\_offer’
* ‘startDate’ to ‘start\_date’
* ‘endDate’ to ‘end\_date’

1. Add key:

* PRIMARY KEY(‘user\_id’,’promo\_id’)

1. Remove columns:

* ‘id’
* ‘user\_email’

1. Add relations

## vat

#### Description:

The list of VATs

#### Relations:

No MANY-TO-ONE relations

#### Modifications:

1. Change columns:

* ‘vartype’ to ‘vat\_type’ enum(‘Zero’,’Standart’) NOT NULL,
* ‘vatper’ to ‘value’

1. Remove columns:

* ‘addDate’

# New Tables

## country

#### Description:

The list of countries which are used by system.

#### Columns:

* `id` smallint(5) unsigned NOT NULL AUTO\_INCREMENT,
* `name\_key` varchar(50) NOT NULL,
* `native\_name` varchar(50) NOT NULL,
* `iso\_code` char(2) NOT NULL COMMENT ‘ISO 3166-1-alpha-2 code’

#### Keys:

* PRIMARY KEY (`id`),
* UNIQUE KEY `name` (`name`),
* UNIQUE KEY `iso\_code` (`iso\_code`),

## address

#### Description:

The list of all addresses

#### Columns:

* `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* `address\_type` enum(‘Billing’, ‘Delivery’) NOT NULL DEFAULT ‘Delivery’,
* `address1` varchar(50) NOT NULL,
* `address2` varchar(50) DEFAULT NULL,
* `address3` varchar(50) DEFAULT NULL,
* `instructions` varchar(250) NULL,
* `city\_id` int(10) unsigned NOT NULL,
* `postcode\_id` int(10) unsigned DEFAULT NULL,
* ‘address\_base\_id’, bigint(20) unsigned NULL,
* `latitude` decimal(18,15) DEFAULT NULL,
* `longitude` decimal(18,15) DEFAULT NULL,

#### Keys:

* PRIMARY KEY (`id`)

#### Relations:

1. MANY-TO-ONE with ‘city’ (3.4)
2. MANY-TO-ONE with ‘postcode’ (2.54)
3. MANY-TO-ONE with ‘postcode’ (2.54)
4. MANY-TO-ONE with ‘address\_base’ (3.3)

## address\_base

#### Description:

The list of address bases for addresses grouping.

#### Columns:

* `id` bigint(10) unsigned NOT NULL AUTO\_INCREMENT,
* ‘name’ varchar(255) NOT NULL
* ‘delivery\_delay\_time’ TIME NOT NULL,
* ‘max\_delivery’ float
* ‘client\_id’ bigint(20) NOT NULL

#### Keys:

* PRIMARY KEY(‘id’)

#### Relations:

1. MANY-TO-ONE with ‘client’ (3.48)

## city

#### Description:

The list of cities

#### Columns:

* `id` int(10) unsigned NOT NULL AUTO\_INCREMENT,
* `name\_key’ varchar(50) NOT NULL,
* `native\_name’ varchar(50) NOT NULL,
* `postcode\_id` int(10) unsigned DEFAULT NULL,

#### Keys:

* PRIMARY KEY (`id`)

#### Relations:

1. MANY-TO-ONE with ‘postcode’ (2.54)

## phone

#### Description:

The list of phone numbers

#### Columns:

* `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* `number` varchar(20) NOT NULL,
* ‘type’ enum(‘Mobile’,’Static’) NULL

#### Keys:

* PRIMARY KEY (`id`)

#### Relations:

No MANY-TO-ONE relations

## company\_address

#### Description:

The list of company addresses

#### Columns:

* `company\_id` bigint(20) unsigned NOT NULL,
* `address\_id` bigint(20) unsigned NOT NULL,

#### Keys:

* PRIMARY KEY (`company\_id`,` address\_id`),

#### Relations:

1. MANY-TO-ONE with ‘company’ (2.13)
2. MANY-TO-ONE with ‘address’ (3.2)

## company\_phone

#### Description:

The list of company phone numbers

#### Columns:

* `company\_id` bigint(20) unsigned NOT NULL,
* `phone\_id` bigint(20) unsigned NOT NULL,

#### Keys:

* PRIMARY KEY (`company\_id`,` phone\_id`),

#### Relations:

1. MANY-TO-ONE with ‘company’ (2.13)
2. MANY-TO-ONE with ‘phone’ (3.5)

## person

#### Description:

The list of personal data

#### Columns:

* `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* `first\_name` varchar(255) NOT NULL,
* `middle\_name` varchar(255) NULL,
* `last\_name` varchar(255) NOT NULL,
* `title` enum(‘Ms’,’Mrs’,’Mr’) NULL

#### Keys:

* PRIMARY KEY (`id`),

#### Relations:

No MANY-TO-ONE relations

## contact

#### Description:

The list of contact persons

#### Columns:

* `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* `person\_id` bigint(20) unsigned NOT NULL,
* `email` varchar(255) NULL,
* ‘is\_opt\_in’ bit(1) NOT NULL Default ‘0’
* ‘type’ enum(‘Primary’,’Billing’, ‘Delivery’) NOT NULL DEFAULT ‘Primary’,
* `phone\_id` bigint(20) unsigned NULL,
* `address\_id` bigint(20) unsigned NULL,

#### Keys:

* PRIMARY KEY (`id`),

#### Relations:

1. MANY-TO-ONE with ‘phone’ (3.5)
2. MANY-TO-ONE with ‘address’ (3.2)
3. MANY-TO-ONE with ‘person’ (3.8)

## company\_contact

#### Description:

The list of company contact persons

#### Columns:

* ‘company\_id` bigint(20) unsigned NOT NULL,
* `contact\_id` bigint(20) unsigned NOT NULL,

#### Keys:

* PRIMARY KEY (`company\_id`,` contact\_id`),

#### Relations:

1. MANY-TO-ONE with ‘company’ (2.13)
2. MANY-TO-ONE with ‘contact’ (3.9)

## user\_contact

#### Description:

The list of user contact persons

#### Columns:

* ‘user\_id` bigint(20) unsigned NOT NULL,
* `contact\_id` bigint(20) unsigned NOT NULL,

#### Keys:

* PRIMARY KEY (`user\_id`,` contact\_id`),

#### Relations:

1. MANY-TO-ONE with ‘user’ (2.80)
2. MANY-TO-ONE with ‘contact’ (3.9)

## order

#### Description:

Will contains the list of all orders

#### Columns:

* `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* ‘order\_number’ bigint(20) unsigned NOT NULL,
* ‘type’ enum (‘Single’,’Group’,’Corporate’,’GroupCorporate’) NOT NULL
* ‘contact\_id’ bigint(20) unsigned NOT NULL
* ‘driver\_id’ bigint(20) unsigned NULL
* ‘voucher\_id’ bigint(20) unsigned NULL
* ‘delivery\_type’ enum(‘Delivered’,’ToCollect’) NOT NULL
* ‘delivery\_time\_type’ enum(‘Now’,’Later’) NOT NULL,
* ‘later\_date’ TIMESTAMP NULL
* ‘order\_payment\_id’ bigint(20) unsigned NULL,
* ‘description’ varchar(500) NULL,
* ‘order\_status’ enum (‘InProgress’,’FoodRecieved’,’FoodPrepare’,’Cancel’,’EstDelivery’,’FoodReady’,’TransToReset’, DeliveryAccept’,’ReadyBy’,’ArrivedAtCustomer’,’DriverAssigned’) NOT NULL Default ‘InProgress’
* ‘previous\_order\_status’ enum (‘FoodReceived,’FoodPrepare’,’Cancel’,’EstDelivery’,’FoodReady’,’TransToReset’,’ DeliveryAccept’ ,’ReadyBy’,’ArrivedAtCustomer’,’DriverAssigned’) NULL
* ‘is\_amend’ bit(1) NOT NULL Default 0,
* ‘is\_term\_cond’ bit(1) NOT NULL Default 0,
* ‘is\_term\_cond\_web\_use’ bit(1) NOT NULL Default 0,
* ‘is\_term\_cond\_acc\_pol’ bit(1) NOT NULL Default 0,
* ‘restaurant\_notes’ varchar(500) NULL
* ‘pickup\_step’ enum(1,3,5,7,9,11,13,15,17,19) NULL
* ‘drop\_off\_step’ enum (1,2,4,6,8,10,12,14,16,18,20) NULL
* ‘size’ varchar(50) NULL
* ‘is\_in\_dispatch’ bit(1) NOT NULL Default 0
* ‘affeliate\_id’ bigint(20) unsigned NULL
* ‘delivery\_charges’ float NULL
* ‘driver\_charges’ float NULL

#### Keys:

* PRIMARY KEY (`id`),
* UNIQUE KEY (`order\_number’),

#### Relations:

1. MANY-TO-ONE with ‘driver’ (3.31)
2. MANY-TO-ONE with ‘promotion’ (2.56)
3. MANY-TO-ONE with ‘contact’ (3.9)
4. MANY-TO-ONE with ‘order\_payment’ (3.14)
5. MANY-TO-ONE with ‘order\_time\_track’ (3.13)
6. MANY-TO-ONE with ‘affeliate’ (3.50)

#### Discuss:

What are the values for ‘pickup\_step’ and ‘drop\_off\_step’?

## order\_time\_track

#### Description:

Contains the info about order time, when it was created, delivered, canceled etc.

#### Columns:

* ‘id’ bigint(20) NOT NULL AUTO\_INCREMENT,
* ‘estimated\_time’ TIME NULL,
* ‘transfer\_at’ TIMESTAMP NULL
* ‘confirm\_at’ TIMESTAMP NULL
* ‘food\_ready\_at’ TIMESTAMP NULL
* ‘food\_prepare\_at’ TIMESTAMP NULL
* ‘food\_delivery\_at’ TIMESTAMP NULL
* ‘delivery\_accepted\_at’ TIMESTAMP NULL
* ‘delivery\_assigned\_at’ TIMESTAMP NULL
* ‘food\_pick\_at’ TIMESTAMP NULL
* ‘food\_estimate\_at’ TIMESTAMP NULL
* ‘food\_recieve\_at’ TIMESTAMP NULL
* ‘cancel\_at’ TIMESTAMP NULL
* ‘on\_way\_restaurant\_at’ TIMESTAMP NULL
* ‘arrived\_to\_restaurant\_at’ TIMESTAMP NULL
* ‘waiting\_for\_food\_at’ TIMESTAMP NULL
* ‘picked\_up\_at’ TIMESTAMP NULL
* ‘food\_route\_at’ TIMESTAMP NULL
* ‘arrived\_to\_customer\_at’ TIMESTAMP NULL
* ‘send\_to\_driver\_at’ TIMESTAMP NULL

#### Relations:

No MANY-TO-ONE relations.

#### Keys:

* PRIMARY KEY (‘id’)

## order\_payment

#### Description:

Contains order payment cost

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* ‘credit\_card\_type’ enum(‘VisaDebit’,’VisaElectron’,’VisaCredit’,’MasterCard’,’Maestro’,’Amex’) NOT NULL,
* ‘commison\_rate’ float NULL,
* ‘charge\_method’ enum(0,1,2)
* ‘payment\_cost’ float NULL,
* ‘payment\_cost\_type’ enum(‘Percent’,’Fixed’)
* ‘customer\_cost\_type’ enum(‘Percent’,’Fixed’)
* ‘customer\_cost’ float NULL,
* ‘payment\_charge’ float NULL,
* ‘total’ float NULL,
* ‘restaurant\_total’ float NULL,
* ‘msd\_cost’ float NULL,
* ‘delivery\_charge’ float NULL,
* ‘refund\_amount’ float NULL,
* ‘restaurant\_charge’ float NULL,
* ‘restaurant\_refund’ float NULL,

#### Keys:

* PRIMARY KEY(‘id’)

#### Relations:

No MANY-TO-ONE relations

#### Discuss:

What are the values for ‘charge\_method’?

## order\_ivr

#### Description:

The relation between order and IVR calls

#### Columns:

* ‘id’ bigint(20) NOT NUL AUTO\_INCREMENT,
* ‘order\_id’ bigint(20) NOT NULL,
* ‘ivr\_type’ enum(1,2,3) NOT NULL

#### Keys:

* PRIMARY KEY(‘id’)

#### Relations:

1. MANY-TO-ONE with ‘order’ (3.12)

#### Discuss:

What are the values for ‘ivr\_type’?

## order\_contact

#### Description:

Contains the list of contacts for particular order

#### Columns:

* ‘order\_id’ bigint(20) unsigned NOT NULL,
* ‘contact\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY(‘order\_id’, ‘contact\_id’)

#### Relations:

1. MANY-TO-ONE with ‘order’ (3.12)
2. MANY-TO-ONE with ‘contact’ (3.9)

## order\_phone

#### Description:

Contains the list of orders phone numbers

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL,
* ‘order\_id’ bigint(20) unsigned NOT NULL,
* ‘phone\_id’ bigint(20) unsigned NOT NULL,

#### Keys:

* PRIMARY KEY(‘id’)

#### Relations:

1. MANY-TO-ONE with ‘phone’ (3.5)
2. MANY-TO-ONE with ‘order’ (3.12)

## navigation

#### Description:

The list of navigation items in menus

#### Columns:

* `id` smallint(5) unsigned NOT NULL AUTO\_INCREMENT,
* `page\_id` int(10) unsigned NOT NULL,
* `language\_id` smallint(5) unsigned NOT NULL,
* `parent\_id` smallint(5) unsigned DEFAULT NULL,
* ‘client\_id’ bigint(20) unsigned NOT NULL,
* `value` varchar(30) NOT NULL,
* `position` enum(‘Footer’,’Header’) NOT NULL DEFAULT ‘Header’,
* `order` tinyint(3) unsigned NOT NULL DEFAULT ‘0’,
* `open\_from` datetime NOT NULL,
* `open\_to` datetime NOT NULL,

#### Keys:

* PRIMARY KEY (`id`),
* KEY `page\_id` (`page\_id`),
* KEY `language\_id` (`language\_id`),
* KEY `parent\_id` (`parent\_id`),

#### Relations:

1. MANY-TO-ONE with ‘page’ (3.19)
2. MANY-TO-ONE with ‘language’ (3.20)
3. MANY-TO-ONE with ‘navigation’ (3.18)
4. MANY-TO-ONE with ‘client’ (3.48)

## page

#### Description:

The list of all static pages

#### Columns:

* `id` int(10) unsigned NOT NULL AUTO\_INCREMENT,
* `language\_id` smallint(5) unsigned NOT NULL,
* `title` varchar(255) NOT NULL,
* `slug` varchar(100) NOT NULL,
* `content` text NOT NULL,
* `description` varchar(160) DEFAULT NULL,
* `robots` varchar(30) DEFAULT NULL,
* ‘client\_id’ bigint(20) unsigned NOT NULL,
* `open\_from` datetime NOT NULL,
* `open\_to` datetime NOT NULL,

#### Keys:

* PRIMARY KEY (`id`),
* UNIQUE KEY `language\_slug` (`language\_id`,`slug`),
* KEY `language\_id` (`language\_id`),
* KEY `slug` (`slug`),

#### Relations:

1. MANY-TO-ONE with ‘language’ (3.20)
2. MANY-TO-ONE with ‘client’ (3.48)

## language

#### Description:

The list of languages

#### Columns:

* `id` smallint(5) unsigned NOT NULL AUTO\_INCREMENT,
* `name` varchar(50) NOT NULL,
* `iso\_code` char(2) NOT NULL COMMENT ‘ISO 639-2’,

#### Keys:

1. PRIMARY KEY (`id`),
2. UNIQUE KEY `name` (`name`),
3. UNIQUE KEY `iso\_code` (`iso\_code`)

#### Relations:

No MANY-TO-ONE relations

## label

#### Description:

The list of label that will be used in web site and will be translated

#### Columns:

* `id` int(10) unsigned NOT NULL AUTO\_INCREMENT,
* `code` varchar(190) NOT NULL COMMENT ‘A unique, human-readable code that identifies the label. This code will be used in our HTML templates.’,
* ‘description` varchar(250) DEFAULT NULL COMMENT ‘A short description. This is a helper for the CMS users.’,

#### Keys:

1. PRIMARY KEY (`id`),
2. UNIQUE KEY `code` (`code`)

#### Relations:

No MANY-TO-ONE relations

## label\_language

#### Description:

The list of label translations

#### Columns:

* `label\_id` int(10) unsigned NOT NULL,
* `language\_id` smallint(5) unsigned NOT NULL,
* `value` varchar(500) NOT NULL,

#### Keys:

* PRIMARY KEY (`label\_id`,`language\_id`),
* KEY `label\_id` (`label\_id`),
* KEY `language\_id` (`language\_id`),

#### Relations:

1. MANY-TO-ONE with ‘label’ (3.21)
2. MANY-TO-ONE with ‘language’ (3.20)

## menu

#### Description:

The list of label translations

#### Columns:

* `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* `name` varchar(150) NOT NULL,
* ‘menu\_type\_id’ bigint(20) unsigned NOT NULL
* ‘restaurant\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (`id`),

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.59)

## menu\_item\_similar

#### Description:

The relations between the similar menu items

#### Columns:

* ‘menu\_item\_id’ bigint(20) unsigned NOT NULL
* ‘menu\_item\_similar\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (`menu\_item\_id’, ‘menu\_item\_similar\_id’),

#### Relations:

1. MANY-TO-ONE with ‘menu\_item’ (2.44)
2. MANY-TO-ONE with ‘menu\_item’ (2.44)

## menu\_bundle

#### Description:

The list of menu bundles for the restaurant. So menu items can be grouped into Bundles - buy 3 items for $X;

#### Columns:

* `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* `name\_key` varchar(50) NOT NULL,
* ‘description\_key` varchar(250) NOT NULL,
* ‘restaurant\_id’ bigint(20) unsigned NOT NULL
* ‘web\_price’ float NOT NULL,
* ‘restaurant\_price’ float NOT NULL

#### Keys:

* PRIMARY KEY (`id`),

#### Relations:

* MANY-TO-ONE with ‘restaurant’ (2.59)

## menu\_item\_menu\_bundle

#### Description:

The relation between bundles and menu items.

#### Columns:

* ‘menu\_item\_id` bigint(20) unsigned NOT NULL
* ‘menu\_bundel\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (`menu\_item\_id`, ‘menu\_bundel\_id’),

#### Relations:

1. MANY-TO-ONE with ‘menu\_item’ (2.44)
2. MANY-TO-ONE with ‘menu\_bundel’ (3.25)

## menu\_allergy

#### Description:

The list of allergies

#### Columns:

* `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* `name\_key` varchar(150) NOT NULL,

#### Keys:

* PRIMARY KEY (`id`),

#### Relations:

No MANY-TO-ONE relations

## schedule

#### Description:

The list of schedules that can be assigned to the expense type, menu, restaurants etc.

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* ­‘from’ TIME NOT NULL
* ‘to’ TIME NOT NULL

#### Relations:

No MANY-TO-ONE relations

## restaurant\_schedule

#### Description:

The time when restaurant works and makes delivery

#### Columns:

* ‘restaurant\_id’ bigint(20) NOT NULL
* ‘schedule\_id’ bigint(20) NOT NULL
* ‘type’ enum (‘OpenTime’,’CloseTime’,’DeliveryTime’) NOT NULL
* ‘open\_day’ enum(‘Sunday’,’Monday’,’Thuesday’,’Wednesday’,’Thursday’,’Friday’,’Saturday’) NOT NULL

#### Keys:

* PRIMARY KEY(‘restaurant\_id’, ‘schedule\_id’)

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.59)
2. MANY-TO-ONE with ‘schedule’ (3.28)

## restaurant\_chain

#### Description:

The list of restaurant chains

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* ‘name’ varchar(150) NOT NULL
* ‘client\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY(‘id’)

#### Relations:

1. MANY-TO-ONE with ‘client’ (3.48)

## driver

#### Description:

The driver info

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT,
* ‘dispatch\_id’ varchar(50) NOT NULL

#### Keys:

* PRIMARY KEY(‘id’)

#### Relations:

No MANY-TO-ONE relations

## vaucher\_restaurant\_chain

#### Description:

The voucher assigned to the restaurant group

#### Columns:

* ‘vaucher\_id’ bigint(20) unsigned NOT NULL
* ‘restaurant\_chain\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘vaucher\_id’, ‘restaurant\_chain\_id’)

#### Relations:

1. MANY-TO-ONE with ‘vaucher’ (2.56)
2. MANY-TO-ONE with ‘restaurant\_chain’ (3.30)

## vaucher\_restaurant\_group

#### Description:

The voucher assigned to the restaurant group

#### Columns:

* ‘vaucher\_id’ bigint(20) unsigned NOT NULL
* ‘restaurant\_group\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘vaucher\_id’, ‘restaurant\_group\_id’)

#### Relations:

1. MANY-TO-ONE with ‘vaucher’ (2.56)
2. MANY-TO-ONE with ‘restaurant\_group’ (2.61)

## vaucher\_restaurant

#### Description:

The voucher assigned to the restaurant

#### Columns:

* ‘vaucher\_id’ bigint(20) unsigned NOT NULL
* ‘restaurant\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘vaucher\_id’, ‘restaurant\_id’)

#### Relations:

1. MANY-TO-ONE with ‘vaucher’ (2.56)
2. MANY-TO-ONE with ‘restaurant’ (2.60)

## restaurant\_contact

#### Description:

Contains the list of contact persons for restaurant

#### Columns:

* ‘restaurant\_id’ bigint(20) unsigned NOT NULL
* ‘contact\_id’ bigint(20) unsigned NOT NULL
* ‘role’ enum (‘Manager’,’Owner’,’Partner’,’Director’) NOT NULL

#### Keys:

* PRIMARY KEY (‘contact\_id’, ‘restaurant\_id’)

#### Relations:

1. MANY-TO-ONE with ‘contact’ (3.9)
2. MANY-TO-ONE with ‘restaurant’ (2.60)

## restaurant\_user

#### Description:

Contains the list of users which have access to the system and assigned to the particular restaurant

#### Columns:

* ‘restaurant\_id’ bigint(20) unsigned NOT NULL
* ‘user\_id’ bigint(20) unsigned NOT NULL
* ‘role’ enum (‘Admin’) NOT NULL

#### Keys:

* PRIMARY KEY (‘user\_id’, ‘restaurant\_id’)

#### Relations:

1. MANY-TO-ONE with ‘users’ (2.80)
2. MANY-TO-ONE with ‘restaurant’ (2.60)

## restaurant\_group\_user

#### Description:

Contains the list of users which have access to the system and assigned to the particular restaurant group

#### Columns:

* ‘restaurant\_group\_id’ bigint(20) unsigned NOT NULL
* ‘user\_id’ bigint(20) unsigned NOT NULL
* ‘role’ enum (‘Admin’) NOT NULL

#### Keys:

* PRIMARY KEY (‘user\_id’, ‘restaurant\_group\_id’)

#### Relations:

1. MANY-TO-ONE with ‘users’ (2.80)
2. MANY-TO-ONE with ‘restaurant\_group’ (2.61)

## restaurant\_chain\_user

#### Description:

Contains the list of users which have access to the system and assigned to the particular restaurant chain

#### Columns:

* ‘restaurant\_chain\_id’ bigint(20) unsigned NOT NULL
* ‘user\_id’ bigint(20) unsigned NOT NULL
* ‘role’ enum (‘Admin’) NOT NULL

#### Keys:

* PRIMARY KEY (‘user\_id’, ‘restaurant\_chain\_id’)

#### Relations:

1. MANY-TO-ONE with ‘users’ (2.80)
2. MANY-TO-ONE with ‘restaurant\_chain’ (3.30)

## restaurant\_contact\_order

#### Description:

Contains the list of contact types for the restaurant (SMS phone number, email, phone number)

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL
* ‘restaurant\_id’ bigint(20) unsigned NOT NULL
* ‘contact\_id’ bigint(20) unsigned NOT NULL
* ‘type’ enum (‘SMS’,’PhoneCall’,’Email’,’VoiceConfirmation’,’IVR’) NOT NULL
* ‘charge’ float NULL
* ‘alert’ bit(1) NOT NULL,
* ‘delay\_in\_min’ int(5) NULL

#### Keys:

* PRIMARY KEY (‘id’)

#### Relations:

1. MANY-TO-ONE with ‘contact’ (3.9)
2. MANY-TO-ONE with ‘restaurant’ (2.60)

## restaurant\_payment

#### Description:

Contains the list of contact types for the restaurant (SMS phone number, email, phone number)

#### Columns:

* ‘restaurant\_id’ bigint(20) unsigned NOT NULL
* ‘frequency\_type’ enum (‘BiMonthly,’Monthly’,’Weekly’,’Daily’,’IVR’) NOT NULL
* ‘delay\_in\_min’ int(5) NULL
* ‘charge\_value’ float NOT NULL
* ‘fee\_value’ float NOT NULL
* ‘fee\_type’ enum(‘VATExclusiev’,’VATInclusive’)
* ‘charge\_type’ enum (‘WebPrice’,’RestaurantPrice’)

#### Keys:

* PRIMARY KEY (‘restaurant\_id’)

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.60)

## restaurant\_best\_for

#### Description:

Contains the list of ‘best for’ items for restaurant

#### Columns:

* ‘restaurant\_id’ bigint(20) unsigned NOT NULL
* ‘best\_for\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘restaurant\_id’, ‘best\_for\_id’)

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.60)
2. MANY-TO-ONE with ‘best\_for’ (2.11)

## restaurant\_delivery

#### Description:

Contains the delivery restaurant info

#### Columns:

* ‘restaurant\_id’ bigint(20) unsigned NOT NULL
* ‘description’ varchar(500) NULL
* ‘algo’ varchar(50) NOT NULL
* ‘type’ enum(‘CloudDriver’,’DedicatedDriver’) NOT NULL
* ‘range\_in\_miles’ float NULL
* ‘has\_collection’ bit(1) NOT NULL
* ‘has\_collection’ bit(1) NOT NULL
* ‘has\_dinein’ bit(1) NOT NULL
* ‘has\_own’ bit(1) NOT NULL
* ‘collect\_time\_in\_min’ int(10) NULL
* ‘driver\_delay\_time’ TIME NULL
* ‘delivery\_per\_week’ int(5)

#### Keys:

* PRIMARY KEY (‘restaurant\_id’)

#### Relations:

1. MANY-TO-ONE with ‘restaurant’ (2.60)

## restaurant\_delivery\_charges

#### Description:

Contains the restaurant the delivery charges (depends to miles)

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘restaurant\_delivery\_id’ bigint(20) unsigned NOT NULL
* ‘distance\_in\_miles’ float NOT NULL
* ‘charge’ float NOT NULL

#### Keys:

* PRIMARY KEY (‘id’)

#### Relations:

1. MANY-TO-ONE with ‘restaurant\_delivery’ (3.42)

## restaurant\_cuisine

#### Description:

Contains the list of cuisine for restaurant

#### Columns:

* ‘restaurant\_id’ bigint(20) unsigned NOT NULL
* ‘cuisine\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘cuisine\_id’, ‘restaurant\_id’)

#### Relations:

1. MANY-TO-ONE with ‘cuisine’ (2.18)
2. MANY-TO-ONE with ‘restaurant’ (2.60)

## restaurant\_payment\_bank

#### Description:

Contains the restaurant payment info for bank accounts

#### Columns:

* ‘restaurant\_payment\_id’ bigint(20) unsigned NOT NULL
* ‘account\_holder\_name’ varchar(150) NOT NULL
* ‘bank\_name’ varchar(150) NOT NULL
* ‘sort\_code’ varchar(50) NOT NULL
* ‘account\_number’ varchar(50) NOT NULL

#### Keys:

* PRIMARY KEY (‘restaurant\_payment\_id’)

#### Relations:

1. MANY-TO-ONE with ‘restaurant\_payment’ (2.603.40)

## restaurant\_payment\_paypal

#### Description:

Contains the restaurant payment info for paypal accounts

#### Columns:

* ‘restaurant\_payment\_id’ bigint(20) unsigned NOT NULL
* ‘email’ varchar(150) NOT NULL

#### Keys:

* PRIMARY KEY (‘restaurant\_payment\_id’)

#### Relations:

1. MANY-TO-ONE with ‘restaurant\_payment’ (2.603.40)

## restaurant\_payment\_cc

#### Description:

Contains the restaurant payment info for credit cards

#### Columns:

* ‘restaurant\_payment\_id’ bigint(20) unsigned NOT NULL
* ‘cc\_name’ varchar(150) NOT NULL
* ‘stard\_date’ TIMESTAMP NOT NULL
* ‘end\_date’ TIMESTAMP NOT NULL
* ‘security\_code’ varchar(50) NOT NULL

#### Keys:

* PRIMARY KEY (‘restaurant\_payment\_id’)

#### Relations:

1. MANY-TO-ONE with ‘restaurant\_payment’ (2.603.40)

## client

#### Description:

Contains the list of DineIn clients

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘name’ varchar(150) NOT NULL
* ‘description’ varchar(500) NULL
* ‘url’ varchar(50) NOT NULL

#### Keys:

* PRIMARY KEY (‘id’)

#### Relations:

No MANY-TO-ONE relations.

## client\_contact

#### Description:

Contains the list of contacts/addresses for client

#### Columns:

* ‘client\_id’ bigint(20) unsigned NOT NULL
* ‘contact\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘client\_id’,’contact\_id’)

#### Relations:

1. MANY-TO-ONE with ‘client’ (3.48)
2. MANY-TO-ONE with ‘contact’ (3.9)

## user\_group\_code

#### Description:

Contains the list of codes for user groups

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘code’ varchar(150) NOT NULL
* ‘company\_user\_group\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘id’)

#### Relations:

1. MANY-TO-ONE with ‘company\_user\_group’ (2.78)

## affiliate

#### Description:

The list of affiliate for the client

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘code’ varchar(150) NOT NULL
* ‘client\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘id’)

#### Relations:

1. MANY-TO-ONE with ‘client’ (3.48)

## menu\_item\_menu\_type

#### Description:

The relations between ‘menu\_item’ and ‘menu\_type’

#### Columns:

* ‘menu\_item\_id’ bigint(20) unsigned NOT NULL
* ‘menu\_type \_id’ bigint(20) unsigned NOT NULL
* ‘web\_price’ float NULL
* ‘restaurant\_price’ float NULL

#### Keys:

* PRIMARY KEY (`menu\_item\_id’, ‘menu\_type\_id’),

#### Relations:

1. MANY-TO-ONE with ‘menu\_item’ (2.46)
2. MANY-TO-ONE with ‘menu\_type’ (2.44)

## menu\_item\_menu\_category

#### Description:

The relations between ‘menu\_item’ and ‘menu\_ category’

#### Columns:

* ‘menu\_item\_id’ bigint(20) unsigned NOT NULL
* ‘menu\_ category\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (`menu\_item\_id’, ‘menu\_ category\_id’),

#### Relations:

1. MANY-TO-ONE with ‘menu\_item’ (2.44)
2. MANY-TO-ONE with ‘menu\_category’ (2.70)

## menu\_item\_menu\_bundle

#### Description:

The relations between ‘menu\_item’ and ‘menu\_bundle’

#### Columns:

* ‘menu\_item\_id’ bigint(20) unsigned NOT NULL
* ‘menu\_bundle\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (`menu\_item\_id’, ‘menu\_ bundle \_id’),

#### Relations:

1. MANY-TO-ONE with ‘menu\_item’ (2.44)
2. MANY-TO-ONE with ‘menu\_bundle’ (3.23)

## feedback\_type

#### Description:

The listing of available feedback types for client

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘name\_key’ varchar(250) NOT NULL
* ‘client\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (`id’),

#### Relations:

1. MANY-TO-ONE with ‘client’ (3.48)

## payment\_method

#### Description:

The listing of available payment methods

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘name’ varchar(250) NOT NULL

#### Keys:

* PRIMARY KEY (`id’),

#### Relations:

No MANY-TO-ONE relations

# Menu options structure

## menu\_option

#### Description:

Contains the options associated with Menu Item. Every Menu Item could have only one Menu Option.

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘name\_key’ varchar(150) NOT NULL
* ‘web\_price’ float NULL,
* ‘restaurant\_price’ float NULL

#### Keys:

* PRIMARY KEY (‘id’)

#### Relations:

No MANY-TO-ONE relations

## menu\_option\_category

#### Description:

Contains the list of Option Categories associated with Menu Option. Every Menu Option could have any number of Menu Option Category.

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘menu\_option\_id’ bigint(20) unsigned NOT NULL
* ‘name\_key’ varchar(250) NOT NULL
* ‘description\_key’ varchar(250) NOT NULL
* ‘view\_type’ enum(‘Dropdown’,’RadioButton’) NOT NULL Default ‘Dropdown’
* ‘items\_limit’ smallint(5) NULL
* ‘price\_calc\_type’ enum(‘Addition’,’Fixed’) NOT NULL Default ‘Addition’
* ‘web\_price’ float NULL,
* ‘restaurant\_price’ float NULL

#### Keys:

* PRIMARY KEY (‘id’)

#### Relations:

1. MANY-TO-ONE relation with ‘menu\_option’ (4.1)

## menu\_option\_item

#### Description:

Contains the list of Option Items that will be displayed for every menu Option Category.

#### Columns:

* ‘id’ bigint(20) unsigned NOT NULL AUTO\_INCREMENT
* ‘name\_key’ varchar(150) NOT NULL
* ‘description\_key’ varchar(250) NOT NULL
* ‘order’ smallint(5) NOT NULL Default 1
* ‘web\_price’ float NULL,
* ‘restaurant\_price’ float NULL
* ‘menu\_category\_id’ bigint(20) unsigned NOT NULL

#### Keys:

* PRIMARY KEY (‘id’)

#### Relations:

1. MANY-TO-ONE relation with ‘menu\_ category’ (2.70)

## menu\_option\_category\_item

#### Description:

Defines the list of Option Items displayed in the Option Category. So the Option Category could be associated with Menu Category or/and with Option Items

#### Columns:

* ‘menu\_option\_category\_id’ bigint(20) unsigned NULL
* ‘menu\_option\_item\_id’ bigint(20) unsigned NULL

#### Keys:

* PRIMARY KEY(‘menu\_option\_item\_id’,’ menu\_option\_category\_id’,’ menu\_category\_id’)

#### Relations:

1. MANY-TO-ONE relation with ‘menu\_option\_category’ (4.2)
2. MANY-TO-ONE relation with ‘menu\_option\_item’ (4.3)