

TALLINNA ÜLIKOOL

Informaatika Instituut

E-turunduse andmebaaside arendamine

Bakalaureusetöö

Autor: Sergei Selivanov

Juhendaja: lektor Jaagup Kippar

Autor: ,,2011

Juhendaja: ,,2011

Instituudi direktor: ,,2011

Tallinn 2011

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(autor)

Sisukord

Autorideklaratsioon	1
Sisukord.....	2
Sissejuhatus	4
1. Probleemi tutvustamine	5
1.1. Tabelid, veerud ja read.....	5
1.2. Seosed relatsioonilises andmebaasis	5
1.2.1. Seos (1:*).....	5
1.2.2. Seos (1:1).....	6
1.2.3. Seos (*:*).....	7
1.3. Miks traditsioonilise relatsioonilise disaini tehnikad hästi ei tööta.....	9
1.3.1. Normaliseerimine tavalise keelega.....	9
1.3.2. Normaliseerimisega seotud probleemid	9
1.4. Kommentaarid ja üldvaade	10
2. Eksisteerivate e-turunduse süsteemide analüüs.....	11
2.1. Süsteemide ülesleidmise meetodid.....	11
2.2. Ülevaade süsteemidest.....	11
2.2.1. Digistore	11
2.2.2. CubeCart.....	14
2.2.3. eCommerce.....	17
2.2.4. Magento.....	19
2.2.5. OpenCart.....	21
2.2.6. osCommerce	24
2.2.7. PrestaShop	27
2.2.8. VirtueMart	30
2.2.9. ZenCart	33
3. Andmemudelid	38
3.1. “ROW” andmemudel.....	38
3.2. “EAV” andmemudel	38

3.3. "ROW" ja "EAV" võrdlemine	38
4. Süsteemide andmebaaside arendamine	40
4.1. Piirangud ja kommentaarid.....	40
4.2. Võrdlemine ning vajalikud arendussammud	40
4.2.1. Teoreetiline võrdlemine	40
4.2.2. Praktiline võrdlemine	41
Kokkuvõte	46
Summary.....	47
Kasutatud kirjandus.	49

Koodid ja süsteemid CD peal.

Sissejuhatus

Tarkvara arendatakse päris tihti. Sellel ilmuvad kogu aeg mingid uued nõudmised. Tihedam on see e-turunduses, kus kasutajal on kogu aeg nõudmised uusi atribuute lisada. Selleks, et neid nõudmisi ellu viia, tuleb arendajal iga kord palju tööd teha nii programmis, kuid ka andmebaasi struktuuris.

Nagu teema pealkirjas on mainitud, selles töös räägitakse ainult andmebaasi struktuuridest ja mitte koodidest, mis andmebaasidega töötavad.

Töö peamiseks eesmärgiks on teha ülevaade sellest, kui hästi on internetis olevates süsteemides lahendatud võimalused andmebaasi universaalse arendamiseks ja kui lihtne see üldse seal on arendajatele. Eesmärgiks on leida kõige paindlikum süsteem.

Uuringuplaaniks on:

- Teha teoreetiline ülevaade sellest, millised andmestruktuurid saavad olla e-turunduse andmebaasides kasutatud.
- Otsida e-turunduse valmis süsteeme, mis oleksid tasuta vabatarkvaralised süsteemid, mis töötaksid Linuxi operatsioonsüsteemiga, PHP programmeerimiskeelega, Apache veebiserveriga ning MySQL andmebaasiga (mis on tegelikult LAMP tehnoloogia).
- Teha väike ülevaade süsteemidest.
- Teha ülevaade andmebaasi struktuuridest millega valitud süsteemid töötavad.
- Võrrelda arenduseprotsesse, nii et saab näidata, kui palju aega võtab süsteemide arendamine.
- Anda kommentaare ning kirjutada kokkuvõtte, nii et oleks nähtav, missugused praktilised võimalused süsteemides on olemas ning mis süsteemid on kõige paindlikumad.

1. Probleemi tutvustamine

1.1. Tabelid, veerud ja read

Andmebaas koosneb ühest või rohkematest tabelitest. Rida tabelis on ühe asja kolleksioon (näiteks üks töötaja või üks rida ostuarvest). Veerg on ühe tüüpi informatsiooni kolleksioon, ja üks kiri ühest veerust ja reast on kõige väiksem andmeühik, mida saab saada andmebaasist päringute keele (query) abil. Näiteks, töötajate informatsiooni tabelis saab olla veerg nimega NAME, kus asuvad kõikide töötajate nimed. Andmeid saadetakse tabelist rea ja veergu filtreerimise abil (Bryla, 2004). Koodinäites 1 saab näha selle struktuuri loomist.

```
mysql> CREATE TABLE `workers`(id INT PRIMARY KEY auto_increment, name VARCHAR(100));
Query OK, 0 rows affected (0.29 sec)
```

```
mysql> INSERT INTO `workers`(name)VALUES('Sergei Selivanov');
Query OK, 1 row affected (0.03 sec)
```

```
mysql> DESCRIBE `workers`;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(100)	YES		NULL	

2 rows in set (0.00 sec)

```
mysql> SELECT name FROM `workers` WHERE name LIKE 'Sergei%' AND id = 1;
```

name
Sergei Selivanov

1 row in set (0.00 sec)

Koodinäide 1: Lihtsa tabeli loomine ning andmete filtreerimine.

1.2. Seosed relatsioonilises andmebaasis

Seosed relatsioonilises andmebaasis klassifitseeritakse nagu “üks-ühele” (1:1), “üks-mitmele” (1:*) ja “mitu-mitmele” (*:*). Seos (1:*) on relatsioonilise modelemise ideaal. See seos on hea igas relatsioonilise disaini andmebaasis. Seos (1:1) ei kasuta kuigi tihti relatsioonilise disaini andmebaasis. Erinevad andmete modelemisnoteerimised toetavad (*:*) seost erinevalt. Näiteks (*:*) seosed tuleb selles olukorras muuta kahte (1:*) seosesse (Rob, Coronel, Crockett, 2008).

1.2.1. Seos (1:*)

Et näidata kuidas selline seos on modelleeritud ja implementeeritud, teeme kaks tabelit ja mõtleme sellest, et näiteks inimestel on autod mingi registri sees. Esimene tabel oleks „INIMENE“ ja teine siis „AUTO“. Koodinäites 2 saab näha (1:*) seose loomist.

```
mysql> CREATE TABLE `INIMENE`(INIMENE_NUM INT PRIMARY KEY auto_increment, INIMENE_LNAME
```

```

VARCHAR(100), INIMENE_FNAME VARCHAR(100), INIMENE_INITIAL VARCHAR(1));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE `AUTO`(AUDO_NUM INT PRIMARY KEY auto_increment, AUDO_IDENTIFIER VARCHAR(255),
INIMENE_NUM INT);
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO `INIMENE`(INIMENE_LNAME, INIMENE_FNAME)VALUES('Selivanov','Sergei');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO `INIMENE `(INIMENE_LNAME, INIMENE_FNAME)VALUES('Kippar','Jaagup');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO `AUTO`(AUTO_IDENTIFIER, INIMENE_NUM)VALUES('345AOF',1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO `AUTO`(AUTO_IDENTIFIER, INIMENE_NUM)VALUES('345FKD',2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO `AUTO`(AUTO_IDENTIFIER, INIMENE_NUM)VALUES('569FSD',1);
Query OK, 1 row affected (0.00 sec)

mysql> DESCRIBE `INIMINE`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| INIMENE_NUM    | int(11)       | NO   | PRI | NULL    | auto_increment |
| INIMENE_LNAME  | varchar(100)  | YES  |     | NULL    |                |
| INIMENE_FNAME  | varchar(100)  | YES  |     | NULL    |                |
| INIMENE_INITIAL| varchar(1)    | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> DESCRIBE `AUTO`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| AUTO_NUM       | int(11)       | NO   | PRI | NULL    | auto_increment |
| AUTO_IDENTIFIER| varchar(255)  | YES  |     | NULL    |                |
| INIMENE_NUM    | int(11)       | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT * FROM `INIMENE`;
+-----+-----+-----+-----+
| INIMENE_NUM | INIMENE_LNAME | INIMENE_FNAME | INIMENE_INITIAL |
+-----+-----+-----+-----+
|          1 | Selivanov     | Sergei         | NULL             |
|          2 | Kippar        | Jaagup        | NULL             |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT * FROM `AUTO`;
+-----+-----+-----+
| AUTO_NUM | AUTO_IDENTIFIER | INIMENE_NUM |
+-----+-----+-----+
|          1 | 345AOF          |            1 |
|          2 | 345FKD          |            2 |
|          3 | 569FSD          |            1 |
+-----+-----+-----+
3 rows in set (0.00 sec)

```

Koodinäide 2: (1:) seose loomine.*

Iga auto on ainult ühe inimese oma ning ühel inimesel saab olla mitu autot.

1.2.2. Seos (1:1)

Selles seoses iga üks rida esimesest tabelist saab olla ühendatud ainult ühe reaga teisest tabelist.

Näiteks üks õpetaja saab olla ainult ühe klassi klassiõpetaja ning ühel klassil saab olla maksimaalselt üks klassiõpetaja. Koodinäites 3 saab näha (1:1) seose loomist.

```
mysql> CREATE TABLE `OPETAJA`(OPETAJA_NUM INT PRIMARY KEY auto_increment, OPETAJA_LNAME
VARCHAR(100), OPETAJA_FNAME VARCHAR(100), OPETAJA_INITIAL VARCHAR(1));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE `KLASS`(KLASS_NUM INT PRIMARY KEY auto_increment, KLASS_NAME VARCHAR(100)
UNIQUE, OPETAJA_NUM INT UNIQUE);
Query OK, 0 rows affected (0.14 sec)

mysql> INSERT INTO `OPETAJA`(OPETAJA_LNAME,OPETAJA_FNAME)VALUES('Kippar','Jaagup');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO `KLASS`(KLASS_NAME,OPETAJA_NUM)VALUES('IF-06',1);
Query OK, 1 row affected (0.00 sec)

mysql> DESCRIBE `OPETAJA`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| OPETAJA_NUM    | int(11)       | NO   | PRI | NULL    | auto_increment |
| OPETAJA_LNAME  | varchar(100)  | YES  |     | NULL    |                |
| OPETAJA_FNAME  | varchar(100)  | YES  |     | NULL    |                |
| OPETAJA_INITIAL| varchar(1)    | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> DESCRIBE `KLASS`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| KLASS_NUM      | int(11)       | NO   | PRI | NULL    | auto_increment |
| KLASS_NAME     | varchar(100)  | YES  | UNI | NULL    |                |
| OPETAJA_NUM    | int(11)       | YES  | UNI | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT * FROM `OPETAJA`;
+-----+-----+-----+-----+
| OPETAJA_NUM | OPETAJA_LNAME | OPETAJA_FNAME | OPETAJA_INITIAL |
+-----+-----+-----+-----+
| 1           | Kippar        | Jaagup        | NULL            |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM `KLASS`;
+-----+-----+-----+
| KLASS_NUM | KLASS_NAME | OPETAJA_NUM |
+-----+-----+-----+
| 1         | IF-06     | 1           |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Koodinäide 3: (1:1) seose loomine.

1.2.3. Seos (*:*)

Näiteks, e-poe rakenduses üks toode saab olla mitmetes füüsilistes ladudes. Koodinäites 4 saab näha vaid (*:*) seose loomist.

```
mysql> CREATE TABLE `TOOTED`(TOODE_NUM INT PRIMARY KEY auto_increment, TOODE_NAME VARCHAR(255),
TOODE_DESC TEXT);
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE `LADU`(LADU_NUM INT PRIMARY KEY auto_increment, LADU_LOC VARCHAR(255));
Query OK, 0 rows affected (0.00 sec)
```



```
mysql> CREATE TABLE `LAOS`(LAOS_NUM INT PRIMARY KEY auto_increment, TOODE_NUM INT, LADU_NUM INT,
UNIQUE `item`(`TOODE_NUM`,`LADU_NUM`));
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> INSERT INTO `TOOTED`(TOODE_NAME,TOODE_DESC)VALUES('arvuti 1','kirjeldus');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO `LADU`(LADU_LOC)VALUES('Tallinn');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO `LADU`(LADU_LOC)VALUES('Tartu');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO `LAOS`(LADU_NUM,TOODE_NUM)VALUES(1,1);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO `LAOS`(LADU_NUM,TOODE_NUM)VALUES(1,2);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> DESCRIBE `TOOTED`;
```

Field	Type	Null	Key	Default	Extra
TOODE_NUM	int(11)	NO	PRI	NULL	auto_increment
TOODE_NAME	varchar(255)	YES		NULL	
TOODE_DESC	text	YES		NULL	

3 rows in set (0.00 sec)

```
mysql> DESCRIBE `LADU`;
```

Field	Type	Null	Key	Default	Extra
LADU_NUM	int(11)	NO	PRI	NULL	auto_increment
LADU_LOC	varchar(255)	YES		NULL	

2 rows in set (0.00 sec)

```
mysql> DESCRIBE `LAOS`;
```

Field	Type	Null	Key	Default	Extra
LAOS_NUM	int(11)	NO	PRI	NULL	auto_increment
TOODE_NUM	int(11)	YES	MUL	NULL	
LADU_NUM	int(11)	YES		NULL	

3 rows in set (0.00 sec)

```
mysql> SELECT * FROM `TOOTED`;
```

TOODE_NUM	TOODE_NAME	TOODE_DESC
1	arvuti 1	kirjeldus

1 row in set (0.00 sec)

```
mysql> SELECT * FROM `LADU`;
```

LADU_NUM	LADU_LOC
1	Tallinn
2	Tartu

2 row in set (0.00 sec)

```
mysql> SELECT * FROM `LAOS`;
```

LAOS_NUM	TOODE_NUM	LADU_NUM
1	1	1
2	1	2

2 row in set (0.00 sec)

Koodinäide 4: (*:*) seose loomine.

1.3. Miks traditsioonilise relatsioonilise disaini tehnikad hästi ei tööta.

Relatsioonilised andmebaasid on disainitud normaliseerimist silmas pidades (Hammergren, Simon, 2009). Järgmistes peatükkides kirjeldatakse mis see täpsemalt tähendab.

1.3.1. Normaliseerimine tavalise keelega.

Relatsiooniline andmebaas on lihtsalt tabel. Selle tabeli ühes reas korduvad andmegrupid eksisteerida küll ei saa (rohkem kui üks nimi ei saa olla vertikaalselt). Võib aga lisada ühe veeru, et oleks olemas kaks asju: "NIMI_1" ja "NIMI_2". Aga prograamiliselt sellega on päris raske töötada, sest see muutub üliiraskeks, kui ühel inimesel on üks nimi, ja teisel on neid tuhat (tegelikult see tihti ei juhtu, see on ainult näidis). Ametlikult seda lahendatakse niimoodi, et tehakse uus tabel, kus meil oleksid kasutajate nimed. Nagu see meil eelnevalt oli (1:*) seosega (vt. koodinäide 5).

```
mysql> DESCRIBE `INIMINE`;
```

Field	Type	Null	Key	Default	Extra
INIMENE_NUM	int(11)	NO	PRI	NULL	auto_increment
INIMENE_LNAME	varchar(100)	YES		NULL	
INIMENE_FNAME	varchar(100)	YES		NULL	
INIMENE_INITIAL	varchar(1)	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> DESCRIBE `AUTO`;
```

Field	Type	Null	Key	Default	Extra
AUTO_NUM	int(11)	NO	PRI	NULL	auto_increment
AUTO_IDENTIFIER	varchar(255)	YES		NULL	
INIMENE_NUM	int(11)	YES		NULL	

```
3 rows in set (0.00 sec)
```

Koodinäide 5: Ühe tabeli veergude parem lahendamine kahte tabelite abil.

Üks tähtsamatest normaliseerimise fookustest on piirata uuendamise anomaaliaid, kui me tahame uuendada üht veergu ühes reas ning ei taha, et teistes ridades sama veergudes oleks sama väärtus. Andmeteladu ei anna võimalust teha uuendamise operatioone, sest tihti see on seotud kogu toote uuendamisega, mitte väikeste uuendamisega. Nii et siis üks tähtsamatest normaliseerimise kasulikkumatest asjadest andmelaos ei tööta (Hammergren, Simon, 2009).

1.3.2. Normaliseerimisega seotud probleemid

Väga palju normaliseeritud andmebaasi üheks tulemuseks on see, et selle sees on väga palju tabeleid. Et saada tooteid, tuleb prograamiliselt teha väga palju liitumise operatsioone ning üldiselt

igale väiksele struktuurile vastab päris palju süsteemiinfot, ja see suurendab süsteemi koormust. See muutub ülikeeruliseks, kui tabelite ning atribuutide arv kasvab, mis juhtub päris tihti e-turunduse rakendustes.

Kui normaliseeritud andmebaasis on palju tabeleid ja toodete andmed on distributseeritud nende tabelite vahel ning uuendamise operatsioone meil pole, miks normaliseerida (Hammergren, Simon, 2009)?

1.4. Kommentaarid ja üldvaade

Ilmselt, kõige paindlikum andmebaasi seos on (*:*) seos. Aga see on ka kõige keerulisem struktuur, ning kõige suurem, sest selleks tuleb teha lisaks toodete tabelile veel kaks tabelit. Tegelikult mingi universaalse mootori tegemiseks tuleb veel palju mõelda kuidas teha niimoodi, et süsteem ei peaks iga kord töötama tabelite lisamisega ning muutmistega, aga töötaks vaid ainult eksisteerivate tabelite sisuga.

2. Eksisteerivate e-turunduse süsteemide analüüs

2.1. Süsteemide ülesleidmise meetodid

Süsteemide ülesleidmiseks olid kasutatud paralleelselt erinevad otsingumootorid koos erinevate võtmesõnadega. Igast päringust oli võtnud esimesed 10 tulemust ning igast tulemusest saime paar süsteemi. Tulemused olid muidugi ka filtreeritud vastavalt selle töö fookusele, mis tähendab LAMP tehnoloogiat.

Otsingumootorid:

- google.com
- yahoo.com
- bing.com
- amazon.com
- ebay.com

Võtmesõnad:

- “ecommerce”
- “ecommerce software”
- “ecommerce open source software”

Igast tulemusest saime ka paar viideid teistele süsteemidele. Kuna mõned viited olid erinevatest foorumitest pärit, seda küll ei saa nimetada halva kvaliteediga informatsiooniks, sest, esiteks, see näitab süsteemide populaarsust, ning teiseks, võrdlemine ja kommenteerimine toimub vaid selles töös.

2.2. Ülevaade süsteemidest

Järgmiselt vaatame ülesleituid süsteeme.

2.2.1. Digistore



Joonis 1: DigiStore logo

Ametlikuks veebileheküljeks on <http://www.digistore.co.nz/>. Töös kasutatakse 4.0 versiooni.

Digistore on osCommerce põhiline tarkvara. Andmebaasi struktuurid on üldiselt samad. Digistore tarkvara vajab serverit, millel on PHP toetus ning on olemas MySQL andmebaas. Toetakse Linux, Unix, BSD, Mac OS X ja Microsoft Windows operatsioonsüsteeme.

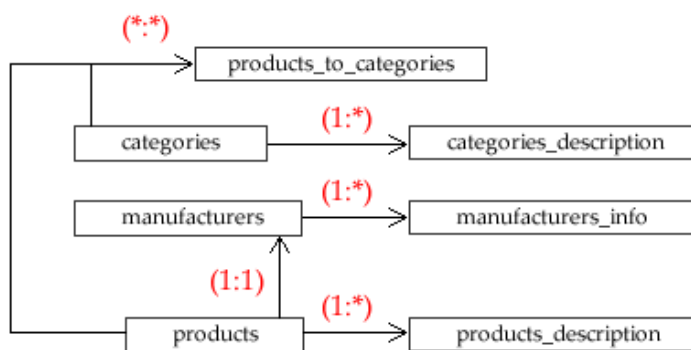
Ametliku arendajalt on kättesaadavad professionaalse disaini tegemine ning hostingu teenused.

Tabelis 1 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 1: DigiStore failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/DigistoreEcommerceV4.0.zip	8297271	7.91
/files/shops_zipped/DigistoreEcommerceV4.0.zip.md5	61	0.000058
Maht serveri kõvakettal:	18525365	17.67
Andmebaasi faili suurus:	347234	0.33

Andmebaasis igal individuaalsel tootel on olemas üks rida *products* tabelis, kus veergudes on olemas nii tavalised toote väärtused, kuid ka rea identifikaatorid teistest tabelitest. Nende identifikaatorite hulgas on 1:* seosed (*products* ja *products_description* tabelite vahel, kus *products_description* tabelis asuvad toodete kirjeldusetekstid erinevates keeltes), 1:1 seosed (*products* ja *manufacturers* tabelite vahel, kus *manufacturers* tabelis asub toodete firma nimed, kus on ka veel üks 1:* seos *manufacturers* ja *manufacturers_info* tabelite vahel, kus *manufacturers_info* tabelis asub firma infotekstid erinevates keeltes) ja *.* seosed (kus meil *products_to_categories* tabelis on kirjeldatud seosed *products* ja *categories* tabelite vahel). On ka veel üks 1:* seos *categories* ja *categories_description* tabelite vahel, mis *categories_description* tabelis asub kategooriate infotekstid erinevates keeltes (vt joonis 2 ja koodinäide 6).



Joonis 2: Digistore andmebaasi struktuur

```
mysql> DESCRIBE `categories`;
```

Field	Type	Null	Key	Default	Extra
categories_id	int(11)	NO	PRI	NULL	auto_increment
categories_image	varchar(64)	YES		NULL	
parent_id	int(11)	NO	MUL	0	
sort_order	int(3)	YES		NULL	
date_added	datetime	YES		NULL	
last_modified	datetime	YES		NULL	

6 rows in set (0.00 sec)

```
mysql> DESCRIBE `categories_description`;
```

Field	Type	Null	Key	Default	Extra
categories_id	int(11)	NO	PRI	0	
language_id	int(11)	NO	PRI	1	
categories_name	varchar(32)	NO	MUL	NULL	

3 rows in set (0.00 sec)

```
mysql> DESCRIBE `manufacturers`;
```

Field	Type	Null	Key	Default	Extra
manufacturers_id	int(11)	NO	PRI	NULL	auto_increment
manufacturers_name	varchar(32)	NO	MUL	NULL	
manufacturers_image	varchar(64)	YES		NULL	
date_added	datetime	YES		NULL	
last_modified	datetime	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> DESCRIBE `manufacturers_info`;
```

Field	Type	Null	Key	Default	Extra
manufacturers_id	int(11)	NO	PRI	NULL	
languages_id	int(11)	NO	PRI	NULL	
manufacturers_url	varchar(255)	NO		NULL	
url_clicked	int(5)	NO		0	
date_last_click	datetime	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> DESCRIBE `products`;
```

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	auto_increment

products_quantity	int(4)	NO		NULL	
products_model	varchar(12)	YES	MUL	NULL	
products_image	varchar(64)	YES		NULL	
products_image_med	varchar(64)	YES		NULL	
products_image_lrg	varchar(64)	YES		NULL	
products_image_sm_1	varchar(64)	YES		NULL	
products_image_xl_1	varchar(64)	YES		NULL	
products_image_sm_2	varchar(64)	YES		NULL	
products_image_xl_2	varchar(64)	YES		NULL	
products_image_sm_3	varchar(64)	YES		NULL	
products_image_xl_3	varchar(64)	YES		NULL	
products_image_sm_4	varchar(64)	YES		NULL	
products_image_xl_4	varchar(64)	YES		NULL	
products_image_sm_5	varchar(64)	YES		NULL	
products_image_xl_5	varchar(64)	YES		NULL	
products_image_sm_6	varchar(64)	YES		NULL	
products_image_xl_6	varchar(64)	YES		NULL	
products_price	decimal(15,4)	NO		NULL	
products_date_added	datetime	NO	MUL	NULL	
products_last_modified	datetime	YES		NULL	
products_date_available	datetime	YES		NULL	
products_weight	decimal(5,2)	NO		NULL	
products_status	tinyint(1)	NO		NULL	
products_tax_class_id	int(11)	NO		NULL	
manufacturers_id	int(11)	YES		NULL	
products_ordered	int(11)	NO		0	
products_bundle	tinytext	NO		NULL	

28 rows in set (0.00 sec)

```
mysql> DESCRIBE `products_description`;
```

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	auto_increment
language_id	int(11)	NO	PRI	1	
products_name	varchar(64)	NO	MUL	NULL	
products_description	text	YES		NULL	
products_url	varchar(255)	YES		NULL	
products_viewed	int(5)	YES		0	

6 rows in set (0.00 sec)

```
mysql> DESCRIBE `products_to_categories`;
```

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	
categories_id	int(11)	NO	PRI	NULL	

2 rows in set (0.00 sec)

Koodinäide 6: DigiStore tabelite ülevaade.

2.2.2. CubeCart



Joonis 3: CubeCart logo

Ametlikuks veebileheküljeks on <http://www.cubecart.com/>. Töös kasutatakse 3.0.20 versiooni.

CubeCart on valmis e-turunduse ostukorvi tarkvara lahendus, mis on kirjutatud tööks serverites, mis toetavad PHP programmeerimiskeelt ning on olemas MySQL andmebaas. Et kasutada CubeCart tarkvarat on vaja sobiv server: Linux/Unix (soovitatud) või Windows, PHP versioon $\geq 5.1.0$, MySQL versioon $\geq 4.1.0$, GD Image Library ≥ 2.0

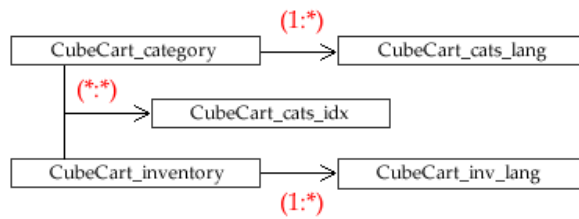
Eksisteerib kaks versiooni: tasuta versioon 3 ning tasuline versioon 4. Selles töös kirjeldatakse tasuta versiooni 3.

Tabelis 2 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 2: *CubeCart* failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/cubecart-3.0.20.zip	2354283	2.25
/files/shops_zipped/cubecart-3.0.20.zip.md5	54	0.000052
Maht serveri kõvakettal:	5237800	5
Andmebaasi faili suurus:	55258	0.05

Andmebaasis igal individuaalsel tootel on olemas üks rida *CubeCart_inventory* tabelis, kus veergudes on olemas nii tavalised toote väärtused, kuid ka rea identifikaatorid teistest tabelitest. Nende identifikaatorite hulgas on kaks 1:* seost (*CubeCart_inventory* ja *CubeCart_inv_lang* tabelite vahel, kus *CubeCart_inv_lang* tabelis asuvad toodete kirjeldusetekstid erinevates keeltes, ning *CubeCart_category* ja *CubeCart_cats_lang* tabelite vahel, kus *CubeCart_cats_lang* tabelis on olemas kategooriate infotekstid erinevates keeltes. Isegi kui *_cats_lang tabelites unikaalsuse reegleid pole, kuna väärtused on lihtsalt tekstid, isegi kui need tekstid on igasugused, need ongi erinevad andmeühikud). Lisaks on veel üks **: seos (kus *CubeCart_cats_idx* tabelis on meil kirjeldatud seosed *CubeCart_inventory* ja *CubeCart_category* tabelite vahel) (vt joonis 4 ja koodinäide 7).



Joonis 4: CubeCart andmebaasi struktuur

```

mysql> DESCRIBE `CubeCart_category`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| cat_name       | text          | NO   |     | NULL    |                |
| cat_id         | int(16)       | NO   | PRI | NULL    | auto_increment |
| cat_father_id  | int(16)       | NO   |     | 0        |                |
| cat_image      | varchar(250)  | NO   |     |          |                |
| per_ship       | decimal(20,2) | NO   |     | 0.00    |                |
| item_ship      | decimal(20,2) | NO   |     | 0.00    |                |
| item_int_ship  | decimal(20,2) | NO   |     | 0.00    |                |
| per_int_ship   | decimal(20,2) | NO   |     | 0.00    |                |
| noProducts     | int(11)       | YES  |     | 0        |                |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql> DESCRIBE `CubeCart_cats_lang`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   | MUL | NULL    | auto_increment |
| cat_master_id  | int(11)       | NO   |     | 0        |                |
| cat_lang       | varchar(20)   | NO   |     |          |                |
| cat_name       | varchar(255)  | NO   |     |          |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> DESCRIBE `CubeCart_cats_idx`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   | MUL | NULL    | auto_increment |
| cat_id         | int(11)       | NO   |     | 0        |                |
| productId      | int(11)       | NO   |     | 0        |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DESCRIBE `CubeCart_inventory`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| productId      | int(11)       | NO   | PRI | NULL    | auto_increment |
| productCode    | varchar(60)   | NO   |     |          |                |
| quantity       | int(16)       | NO   |     | 1        |                |
| description     | text          | NO   |     | NULL    |                |
| image          | varchar(250)  | NO   |     |          |                |
| noImages       | int(11)       | NO   |     | 0        |                |
| price          | decimal(30,2) | NO   |     | 0.00    |                |
| name           | varchar(250)  | NO   |     | 0        |                |
| cat_id         | int(16)       | NO   |     | 0        |                |
| popularity     | bigint(64)    | NO   |     | 0        |                |
| sale_price     | decimal(30,2) | NO   |     | 0.00    |                |
| stock_level    | int(11)       | YES  |     | 0        |                |
| useStockLevel  | int(11)       | YES  |     | 1        |                |
| digital        | int(11)       | NO   |     | 0        |                |
| digitalDir     | varchar(255)  | YES  |     | NULL    |                |
| prodWeight     | decimal(10,3) | YES  |     | NULL    |                |
| taxType        | int(11)       | YES  |     | NULL    |                |
| showFeatured   | int(11)       | NO   |     | 1        |                |
+-----+-----+-----+-----+-----+-----+
  
```

18 rows in set (0.00 sec)

```
mysql> DESCRIBE `CubeCart_inv_lang`;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	MUL	NULL	auto_increment
name	varchar(255)	NO			
prod_lang	varchar(20)	NO			
prod_master_id	int(11)	NO		0	
description	text	NO		NULL	

5 rows in set (0.00 sec)

Koodinäide 7: CubeCart tabelite ülevaade.

2.2.3. eCommerce



Joonis 5: Drupal logo

Ametlikuks veebileheküljeks on <http://drupal.org/project/ecommerce>. Töös kasutatakse 4.2 eCommerce mooduli versiooni koos 6.20 Drupal versiooniga.

eCommerce on Drupal CMS süsteemid moodul. See pole iseseisev tarkvaralahendus.

Drupal on vabatarkvaraline sisuhaldussüsteem. Seda saab panna nii Apache veebiserveritele, kuid ka Microsofti IIS serveritele. On toetatud Apache 1.3 või Apache 2.x operatsioonisüsteemidel nagu Linux/Unix, Mac OS X või Windows. Suurem osa Drupal arendusest toimub Apache veebiserveri abil, nii et seal on rohkem infot ning kogemusi Drupal ühiskonnas, mis on seotud Apache veebiserveriga. Drupal 6 toetab MySQL 4.1 või vanemat versiooni. Drupal toetab MyISAM ja InnoDB tabeli mootorit. NDB tabelid (MySQL Cluster) pole toetud. Microsoft SQL Server ja Oracle on toetud valise mooduli abil. Drupal 5 and 6 vajab seda, et PHP 5.2 oleks kättesaadav.

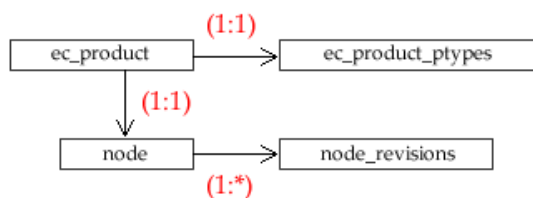
Tabelis 3 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 3: Drupal failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/ecommerce-6.x-4.2.zip	314657	0.3
/files/shops_zipped/ecommerce-6.x-	56	0.000054

4.2.zip.md5		
/files/dependencies/drupal-6.20.zip	1285970	1.23
/files/dependencies/drupal-6.20.zip.md5	50	0.000048
/files/dependencies/ctools-6.x-1.8.zip	721594	0.69
/files/dependencies/ctools-6.x-1.8.zip.md5	53	0.000051
Maht serveri kõvakettal:	7337108	7
Andmebaasi faili suurus:	1690432	1.61

Andmebaasis igal individuaalsel toodel on olemas üks rida *ec_product* tabelis, kus veergudes on olemas nii tavalised toote väärtused, kuid ka rea identifikaatorid teistest tabelitest. Nende identifikaatorite hulgas on kaks 1:1 seost (*ec_product* ja *ec_product_ptypes* tabelite vahel ning *ec_product* ja *node* tabelite vahel). On ka veel üks 1:* seos *node* ja *node_revisions* tabelite vahel (vt joonis 6 ja koodinäide 8).



Joonis 6: Drupal eCommerce andmebaasi struktuur

```

mysql> DESCRIBE `ec_product_ptypes`;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ptype | varchar(32)   | NO   | PRI | NULL    |       |
| name  | varchar(32)   | NO   |     | NULL    |       |
| description | longtext     | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DESCRIBE `ec_product`;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nid   | int(10) unsigned | NO   |     | 0        |       |
| vid   | int(10) unsigned | NO   | PRI | 0        |       |
| ptype | varchar(75)     | NO   | MUL |          |       |
| sku   | varchar(75)     | YES  |     | NULL    |       |
| price | decimal(10,2)   | NO   |     | 0.00    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> DESCRIBE `node_revisions`;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nid   | int(10) unsigned | NO   | MUL | 0        |       |
| vid   | int(10) unsigned | NO   | PRI | NULL    |       |
| uid   | int(11)         | NO   | MUL | 0        |       |
| title | varchar(255)    | NO   |     |          |       |
| body  | longtext       | NO   |     | NULL    |       |
| teaser | longtext       | NO   |     | NULL    |       |
| log   | longtext       | NO   |     | NULL    |       |
| timestamp | int(11)       | NO   |     | 0        |       |
+-----+-----+-----+-----+-----+-----+

```

```
| format | int(11) | NO | | 0 | |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

```
mysql> DESCRIBE `node`;
```

Field	Type	Null	Key	Default	Extra
nid	int(10) unsigned	NO	PRI	NULL	auto_increment
vid	int(10) unsigned	NO	UNI	0	
type	varchar(32)	NO	MUL		
language	varchar(12)	NO			
title	varchar(255)	NO	MUL		
uid	int(11)	NO	MUL	0	
status	int(11)	NO	MUL	1	
created	int(11)	NO	MUL	0	
changed	int(11)	NO	MUL	0	
comment	int(11)	NO		0	
promote	int(11)	NO	MUL	0	
moderate	int(11)	NO	MUL	0	
sticky	int(11)	NO		0	
tnid	int(10) unsigned	NO	MUL	0	
translate	int(11)	NO	MUL	0	

```
15 rows in set (0.00 sec)
```

Koodinäide 8: Drupal tabelite ülevaade.

2.2.4. Magento



Joonis 7: Magento logo

Ametlikuks veebileheküljeks on <http://www.magentocommerce.com/>. Töös kasutatakse 1.5.0.1 Magento versiooni.

Magento on vabatarkvaraline e-turunduse veebirakendus, mis ilmus 31. Märtsil, 2008. Seda loodi Varien Zend Frameworki baasil (Magento creates huge success with enterprise e-commerce platform & community built on Zend Framework, 2008).

Magento Enterprise Edition, tasuline Magento versioon, mille fookus on suured firmad, ilmus 15. Aprillil, 2009. See versioon annab teenuste osutamise lepingut Varieniga (Announcing the Magento Enterprise Edition, 2009).

Varien, firma mis omab Magento süsteemi, varem töötas osCommerce süsteemiga. Esialgu kavatsetakse lõpeda tööd osCommerce süsteemiga, aga hiljem otsustati seda ümber teha Magento nime all. Magento arendust alustati 2007 aasta alguses (Interview: Roy Rubin About Magento Open

Source eCommerce, 2007).

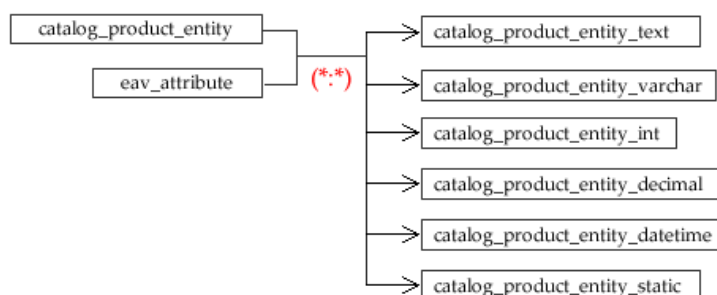
Et see oleks paindlik, Magento andmebaasi kasutab Entity-Attribute-Value(EAV) andmemudelit. Nagu see tihti on, hinnaks on keerukus. Andmetega töötamine Magentos on tihti palju keerulisem, kui see on traditsioonilistes relatsioonilistes tabelites. Nii et EAV printsiibide arusaamine on väga tähtis kui on vaja andmebaasis midagi muutma (Magento Database, 2011).

Tabelis 4 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 4: Magento failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/magento-1.5.0.1.zip	23809883	22.71
/files/shops_zipped/magento-1.5.0.1.zip.md5	54	0.000052
Maht serveri kõvakettal:	158284198	150.95
Andmebaasi faili suurus:	568991	0.54

Andmebaasis igal individuaalsel tootel on olemas üks rida *catalog_product_entity* tabelis. Kõik võimalikud atribuutid (see sisaldab üldse kõik süsteemi atribuudid, mitte ainult toodete atribuutid) asuvad *eav_attribute* tabelis ning kõikide atribuutide väärtused asuvad tabelites nimetega *catalog_product_entity_varchar*, *catalog_product_entity_decimal*, *catalog_product_entity_etc* (Magento for Developers: Part 7 - Advanced ORM - Entity Attribute Value, n.d.). (vt joonis 8 ja koodinäide 9). Võimalikud struktuurid on: static, datetime, decimal, int, text ja varchar. Tegelikult see on hübriidne lahendus. Selle kirjeldamisel, saab rääkida nii kuuest (*:*), kuid ka lihtsalt ühest (*:*) seostest. Kuna kasutatakse ainult üht toodete tabelit ning üht atribuutide tabelit, räägime, et see ongi üks (*:*) seos.



Joonis 8: Magento andmebaasi struktuur

```
mysql> DESCRIBE `catalog_product_entity`;
```

Field	Type	Null	Key	Default	Extra
-------	------	------	-----	---------	-------

entity_id	int(10) unsigned	NO	PRI	NULL	auto_increment
entity_type_id	smallint(8) unsigned	NO	MUL	0	
attribute_set_id	smallint(5) unsigned	NO	MUL	0	
type_id	varchar(32)	NO		simple	
sku	varchar(64)	YES	MUL	NULL	
has_options	smallint(1)	NO		0	
required_options	tinyint(1) unsigned	NO		0	
created_at	datetime	NO		0000-00-00 00:00:00	
updated_at	datetime	NO		0000-00-00 00:00:00	

9 rows in set (0.00 sec)

mysql> DESCRIBE `eav_attribute`;

Field	Type	Null	Key	Default	Extra
attribute_id	smallint(5) unsigned	NO	PRI	NULL	auto_increment
entity_type_id	smallint(5) unsigned	NO	MUL	0	
attribute_code	varchar(255)	NO			
attribute_model	varchar(255)	YES		NULL	
backend_model	varchar(255)	YES		NULL	
backend_type	enum('static','datetime','decimal','int','text','varchar')	NO		static	
backend_table	varchar(255)	YES		NULL	
frontend_model	varchar(255)	YES		NULL	
frontend_input	varchar(50)	YES		NULL	
frontend_label	varchar(255)	YES		NULL	
frontend_class	varchar(255)	YES		NULL	
source_model	varchar(255)	YES		NULL	
is_required	tinyint(1) unsigned	NO		0	
is_user_defined	tinyint(1) unsigned	NO		0	
default_value	text	YES		NULL	
is_unique	tinyint(1) unsigned	NO		0	
note	varchar(255)	NO		NULL	

17 rows in set (0.00 sec)

mysql> DESCRIBE `catalog_product_entity_decimal`;

Field	Type	Null	Key	Default	Extra
value_id	int(11)	NO	PRI	NULL	auto_increment
entity_type_id	smallint(5) unsigned	NO		0	
attribute_id	smallint(5) unsigned	NO	MUL	0	
store_id	smallint(5) unsigned	NO	MUL	0	
entity_id	int(10) unsigned	NO	MUL	0	
value	decimal(12,4)	YES		NULL	

6 rows in set (0.00 sec)

Koodinäide 9: Magento tabelite ülevaade.

2.2.5. OpenCart



Joonis 9: OpenCart logo

Ametlikuks veebileheküljeks on <http://www.opencart.com/>. Töös kasutatakse 1.4.9.3 OpenCart versiooni.

OpenCart töötab järgmistes keskkondades:

- Apache või Microsoft IIS veebiserverid
- PHP versioon 5
- MySQL andmebaas

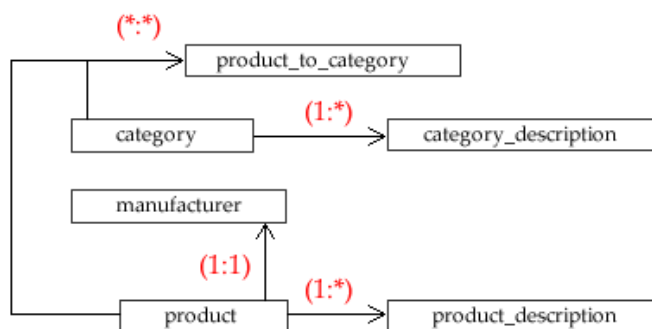
OpenCart süsteem on vabatarkvaraline PHP-põhjaline e-turunduse süsteem. Ametlikul arendajal on tasuta kasutajatugi kõikidele kasutajatele. Süsteemi on sisse ehitatud erinevad tellimuste juhtimisvahendid ning makseviisi.

Tabelis 5 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 5: OpenCart failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/opencart_v1.4.9.3.zip	6311195	6.02
/files/shops_zipped/opencart_v1.4.9.3.zip.md5	56	0.000054
Server disk space required	12521768	11.94
Andmebaasi faili suurus:	238044	0.23

Andmebaasis igal individuaalsel tootel on olemas üks rida *product* tabelis, kus veergudes on olemas nii tavalised toote väärtused, kuid ka rea identifikaatorid teistest tabelitest. Nende identifikaatorite hulgas on 1:* seosed (*product* ja *product_description* tabelite vahel, kus *product_description* tabelis asuvad toodete kirjeldusetekstid erinevates keeltes), 1:1 seosed (*product* ja *manufacturer* tabelite vahel, kus *manufacturer* tabelis asub toodete firma nimed) ja *: * seosed (kus meil *product_to_category* tabelis on kirjeldatud seosed *product* ja *category* tabelite vahel). On ka veel üks 1:* seos *category* ja *category_description* tabelite vahel, mis *category_description* tabelis asub kategooriate infotekstid erinevates keeltes. Tegelikult struktuur on sarnane Digistore struktuuriga, mis tähendab seost osCommerce tarkvaraga (Digistore on osCommerce süsteemi põhine) (vt joonis 10 ja koodinäide 10).



Joonis 10: OpenCart andmebaasi struktuur

```
mysql> DESCRIBE `category`;
```

Field	Type	Null	Key	Default	Extra
category_id	int(11)	NO	PRI	NULL	auto_increment
image	varchar(255)	YES		NULL	
parent_id	int(11)	NO		0	
sort_order	int(3)	NO		0	
date_added	datetime	NO		0000-00-00 00:00:00	
date_modified	datetime	NO		0000-00-00 00:00:00	
status	int(1)	NO		1	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE `category_description`;
```

Field	Type	Null	Key	Default	Extra
category_id	int(11)	NO	PRI	NULL	
language_id	int(11)	NO	PRI	NULL	
name	varchar(255)	NO	MUL		
meta_keywords	varchar(255)	NO		NULL	
meta_description	varchar(255)	NO		NULL	
description	text	NO		NULL	

6 rows in set (0.00 sec)

```
mysql> DESCRIBE `manufacturer`;
```

Field	Type	Null	Key	Default	Extra
manufacturer_id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(64)	NO			
image	varchar(255)	YES		NULL	
sort_order	int(3)	NO		NULL	

4 rows in set (0.00 sec)

```
mysql> DESCRIBE `product`;
```

Field	Type	Null	Key	Default	Extra
product_id	int(11)	NO	PRI	NULL	auto_increment
model	varchar(64)	NO		NULL	
sku	varchar(64)	NO		NULL	
location	varchar(128)	NO		NULL	
quantity	int(4)	NO		0	
stock_status_id	int(11)	NO		NULL	
image	varchar(255)	YES		NULL	
manufacturer_id	int(11)	NO		NULL	
shipping	int(1)	NO		1	
price	decimal(15,4)	NO		0.0000	
tax_class_id	int(11)	NO		NULL	
date_available	date	NO		NULL	
weight	decimal(5,2)	NO		0.00	

weight_class_id	int(11)	NO		0	
length	decimal(5,2)	NO		0.00	
width	decimal(5,2)	NO		0.00	
height	decimal(5,2)	NO		0.00	
length_class_id	int(11)	NO		0	
status	int(1)	NO		0	
date_added	datetime	NO		0000-00-00 00:00:00	
date_modified	datetime	NO		0000-00-00 00:00:00	
viewed	int(5)	NO		0	
sort_order	int(11)	NO		0	
subtract	int(1)	NO		1	
minimum	int(11)	NO		1	
cost	decimal(15,4)	NO		0.0000	

26 rows in set (0.00 sec)

```
mysql> DESCRIBE `product_description`;
```

Field	Type	Null	Key	Default	Extra
product_id	int(11)	NO	PRI	NULL	auto_increment
language_id	int(11)	NO	PRI	NULL	
name	varchar(255)	NO	MUL	NULL	
meta_keywords	varchar(255)	NO		NULL	
meta_description	varchar(255)	NO		NULL	
description	text	NO		NULL	

6 rows in set (0.00 sec)

```
mysql> DESCRIBE `product_to_category`;
```

Field	Type	Null	Key	Default	Extra
product_id	int(11)	NO	PRI	NULL	
category_id	int(11)	NO	PRI	NULL	

2 rows in set (0.00 sec)

Koodinäide 10: OpenCart tabelite ülevaade.

2.2.6. osCommerce



Joonis 11: osCommerce logo

Ametlikuks veebileheküljeks on <http://www.oscommerce.com/>. Töös kasutatakse 2.3.1 osCommerce versiooni. osCommerce on ühe vanematest e-turunduse süsteemidest. Selle põhjal alustati tegemist palju alternatiivsüsteeme, nagu Digistore, ZenCart ja teised süsteemid.

Varien, firma mis on Magento süsteemi omanik, töötas varem osCommerce süsteemiga. Eialgu kavatsetakse lõpeda tööd osCommerce süsteemiga, aga hiljem otsustati seda ümber teha Magento nime all. Magento arendus alustati 2007 aasta alguses (Interview: Roy Rubin About Magento Open Source eCommerce, 2007).

Serveri nõudmised:

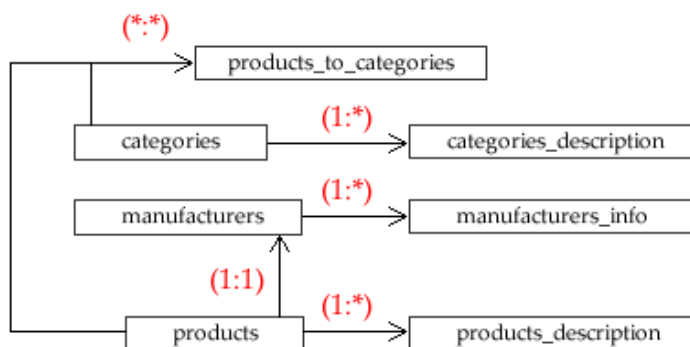
- PHP v4+ (PHP v5+ soovitatud)
- MySQL v3+ (MySQL v5+ soovitatud)

Tabelis 6 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 6: osCommerce failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/oscommerce-2.3.1.zip	5267948	5.02
/files/shops_zipped/oscommerce-2.3.1.zip.md5	55	0.000053
Maht serveri kõvakettal:	5059857	4.83
Andmebaasi faili suurus:	135431	0.13

Andmebaasis igal individuaalsel toodel on olemas üks rida *products* tabelis, kus veergudes on olemas nii tavalised toode väärtused, kuid ka rea identifikaatorid teistest tabelitest. Nende identifikaatorite hulgas on 1:* seosed (*products* ja *products_description* tabelite vahel, kus *products_description* tabelis asuvad toodete kirjeldusetekstid erinevates keeltes), 1:1 seosed (*products* ja *manufacturers* tabelite vahel, kus *manufacturers* tabelis asub toodete firma nimed, kus on ka veel üks 1:* seos *manufacturers* ja *manufacturers_info* tabelite vahel, kus *manufacturers_info* tabelis asub firma infotekstid erinevates keeltes) ja *.* seosed (kus meil *products_to_categories* tabelis on kirjeldatud seosed *products* ja *categories* tabelite vahel). On ka veel üks 1:* seos *categories* ja *categories_description* tabelite vahel, mis *categories_description* tabelis asuvad kategooriate infotekstid erinevates keeltes. Kuna eelnevalt kirjeldatud DigiStore süsteem on selle süsteemi põhine, andmebaasi struktuur on siin tegelikult sama (vt joonis 12 ja koodinäide 11).



Joonis 12: osCommerce andmebaasi struktuur

```
mysql> DESCRIBE `categories`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+

```

categories_id	int(11)	NO	PRI	NULL	auto_increment
categories_image	varchar(64)	YES		NULL	
parent_id	int(11)	NO	MUL	0	
sort_order	int(3)	YES		NULL	
date_added	datetime	YES		NULL	
last_modified	datetime	YES		NULL	

6 rows in set (0.00 sec)

mysql> DESCRIBE `categories_description`;

Field	Type	Null	Key	Default	Extra
categories_id	int(11)	NO	PRI	0	
language_id	int(11)	NO	PRI	1	
categories_name	varchar(32)	NO	MUL	NULL	

3 rows in set (0.00 sec)

mysql> DESCRIBE `manufacturers`;

Field	Type	Null	Key	Default	Extra
manufacturers_id	int(11)	NO	PRI	NULL	auto_increment
manufacturers_name	varchar(32)	NO	MUL	NULL	
manufacturers_image	varchar(64)	YES		NULL	
date_added	datetime	YES		NULL	
last_modified	datetime	YES		NULL	

5 rows in set (0.00 sec)

mysql> DESCRIBE `manufacturers_info`;

Field	Type	Null	Key	Default	Extra
manufacturers_id	int(11)	NO	PRI	NULL	
languages_id	int(11)	NO	PRI	NULL	
manufacturers_url	varchar(255)	NO		NULL	
url_clicked	int(5)	NO		0	
date_last_click	datetime	YES		NULL	

5 rows in set (0.00 sec)

mysql> DESCRIBE `products`;

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	auto_increment
products_quantity	int(4)	NO		NULL	
products_model	varchar(12)	YES	MUL	NULL	
products_image	varchar(64)	YES		NULL	
products_image_med	varchar(64)	YES		NULL	
products_image_lrg	varchar(64)	YES		NULL	
products_image_sm_1	varchar(64)	YES		NULL	
products_image_xl_1	varchar(64)	YES		NULL	
products_image_sm_2	varchar(64)	YES		NULL	
products_image_xl_2	varchar(64)	YES		NULL	
products_image_sm_3	varchar(64)	YES		NULL	
products_image_xl_3	varchar(64)	YES		NULL	
products_image_sm_4	varchar(64)	YES		NULL	
products_image_xl_4	varchar(64)	YES		NULL	
products_image_sm_5	varchar(64)	YES		NULL	
products_image_xl_5	varchar(64)	YES		NULL	
products_image_sm_6	varchar(64)	YES		NULL	
products_image_xl_6	varchar(64)	YES		NULL	
products_price	decimal(15,4)	NO		NULL	
products_date_added	datetime	NO	MUL	NULL	
products_last_modified	datetime	YES		NULL	
products_date_available	datetime	YES		NULL	
products_weight	decimal(5,2)	NO		NULL	
products_status	tinyint(1)	NO		NULL	
products_tax_class_id	int(11)	NO		NULL	

manufacturers_id	int(11)	YES		NULL	
products_ordered	int(11)	NO		0	
products_bundle	tinytext	NO		NULL	

13 rows in set (0.00 sec)

```
mysql> DESCRIBE `products_description`;
```

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	auto_increment
language_id	int(11)	NO	PRI	1	
products_name	varchar(64)	NO	MUL	NULL	
products_description	text	YES		NULL	
products_url	varchar(255)	YES		NULL	
products_viewed	int(5)	YES		0	

6 rows in set (0.00 sec)

```
mysql> DESCRIBE `products_to_categories`;
```

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	
categories_id	int(11)	NO	PRI	NULL	

2 rows in set (0.00 sec)

Koodinäide 11: osCommerce tabelite ülevaade.

2.2.7. PrestaShop



Joonis 13: PrestaShop logo

Ametlikuks veebileheküljeks on <http://www.prestashop.com/>. Töös kasutatakse 1.3.7.0 PrestaShop versiooni.

PrestaShop on tehtud PHP ja MySQL tehnoloogiate abil. Arendaja poolt on kättesaadav PrestaBox hostingu teenus, kuhu saab oma e-poe panna. PrestaShop süsteemil on olemas paar hargüsteemi, mis on aga pole nii populaarsed.

Et kasutada PrestaShop süsteemi, server peab toetama järmiseid asju:

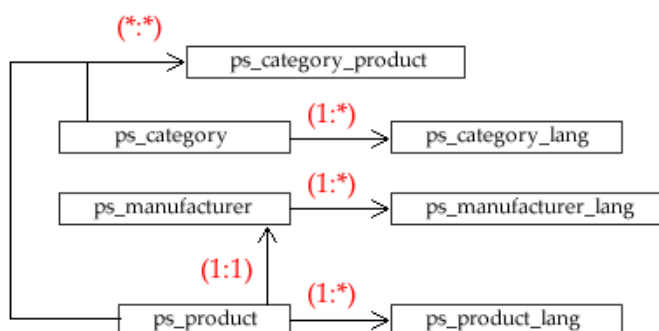
- Linux/Unix või Windows operatsioonsüsteem
- Apache veebiserver
- PHP versioon 5.0
- MySQL versioon 4.1.14

Tabelis 7 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 7: PrestaShop failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/prestashop_1.3.7.0.zip	8565739	8.17
/files/shops_zipped/prestashop_1.3.7.0.zip.md5	57	0.000054
Maht serveri kõvakettal:	16234915	15.48
Andmebaasi faili suurus:	289414	0.28

Andmebaasis igal individuaalsel toodel on olemas üks rida *ps_product* tabelis, kus veergudes on olemas nii tavalised toode väärtused, kuid ka rea identifikaatorid teistest tabelitest. On olemas üks 1:* seos (*ps_products* ja *ps_product_lang* tabelite vahel ning *ps_product* ja *ps_manufacturer* tabelite vahel). Üldiselt on olemas ka veel kaks 1:* seost (*ps_category* ja *ps_category_lang* tabelite vahel ning *ps_manufacturer* ja *ps_manufacturer_lang* tabelite vahel). On ka veel üks *:.* seos *ps_category* ja *ps_product* tabelite vahel, mis on kirjeldatud *ps_product_category* tabelis (vt joonis 14 ja koodinäide 12).



Joonis 14: PrestaShop andmebaasi struktuur

```
mysql> DESCRIBE `ps_category`;
```

Field	Type	Null	Key	Default	Extra
id_category	int(10) unsigned	NO	PRI	NULL	auto_increment
id_parent	int(10) unsigned	NO	MUL	NULL	
level_depth	tinyint(3) unsigned	NO		0	
active	tinyint(1) unsigned	NO		0	
date_add	datetime	NO		NULL	
date_upd	datetime	NO		NULL	

6 rows in set (0.00 sec)

```
mysql> DESCRIBE `ps_category_lang`;
```

Field	Type	Null	Key	Default	Extra
id_category	int(10) unsigned	NO	PRI	NULL	
id_lang	int(10) unsigned	NO	PRI	NULL	
name	varchar(128)	NO	MUL	NULL	
description	text	YES		NULL	
link_rewrite	varchar(128)	NO		NULL	
meta_title	varchar(128)	YES		NULL	

meta_keywords	varchar(255)	YES		NULL	
meta_description	varchar(255)	YES		NULL	

8 rows in set (0.00 sec)

mysql> DESCRIBE `ps_category_product`;

Field	Type	Null	Key	Default	Extra
id_category	int(10) unsigned	NO	MUL	NULL	
id_product	int(10) unsigned	NO	MUL	NULL	
position	int(10) unsigned	NO		0	

3 rows in set (0.00 sec)

mysql> DESCRIBE `ps_manufacturer`;

Field	Type	Null	Key	Default	Extra
id_manufacturer	int(10) unsigned	NO	PRI	NULL	auto_increment
name	varchar(64)	NO		NULL	
date_add	datetime	NO		NULL	
date_upd	datetime	NO		NULL	

4 rows in set (0.00 sec)

mysql> DESCRIBE `ps_manufacturer_lang`;

Field	Type	Null	Key	Default	Extra
id_manufacturer	int(10) unsigned	NO	PRI	NULL	
id_lang	int(10) unsigned	NO	PRI	NULL	
description	text	YES		NULL	
short_description	varchar(254)	YES		NULL	
meta_title	varchar(128)	YES		NULL	
meta_keywords	varchar(255)	YES		NULL	
meta_description	varchar(255)	YES		NULL	

7 rows in set (0.00 sec)

mysql> DESCRIBE `ps_product`;

Field	Type	Null	Key	Default	Extra
id_product	int(10) unsigned	NO	PRI	NULL	auto_increment
id_supplier	int(10) unsigned	YES	MUL	NULL	
id_manufacturer	int(10) unsigned	YES	MUL	NULL	
id_tax	int(10) unsigned	NO	MUL	NULL	
id_category_default	int(10) unsigned	YES	MUL	NULL	
id_color_default	int(10) unsigned	YES	MUL	NULL	
on_sale	tinyint(1) unsigned	NO		0	
ean13	varchar(13)	YES		NULL	
ecotax	decimal(17,2)	NO		0.00	
quantity	int(10) unsigned	NO		0	
price	decimal(20,6)	NO		0.000000	
wholesale_price	decimal(20,6)	NO		0.000000	
reduction_price	decimal(17,2)	YES		NULL	
reduction_percent	float	YES		NULL	
reduction_from	datetime	NO		1970-01-01 00:00:00	
reduction_to	datetime	NO		1970-01-01 00:00:00	
reference	varchar(32)	YES		NULL	
supplier_reference	varchar(32)	YES		NULL	
location	varchar(64)	YES		NULL	
weight	float	NO		0	
out_of_stock	int(10) unsigned	NO		2	
quantity_discount	tinyint(1)	YES		0	
customizable	tinyint(2)	NO		0	
uploadable_files	tinyint(4)	NO		0	
text_fields	tinyint(4)	NO		0	
active	tinyint(1) unsigned	NO		0	
indexed	tinyint(1)	NO		0	
date_add	datetime	NO	MUL	NULL	
date_upd	datetime	NO		NULL	

29 rows in set (0.00 sec)

```
mysql> DESCRIBE `ps_product_lang`;
```

Field	Type	Null	Key	Default	Extra
id_product	int(10) unsigned	NO	PRI	NULL	
id_lang	int(10) unsigned	NO	PRI	NULL	
description	text	YES		NULL	
description_short	text	YES		NULL	
link_rewrite	varchar(128)	NO		NULL	
meta_description	varchar(255)	YES		NULL	
meta_keywords	varchar(255)	YES		NULL	
meta_title	varchar(128)	YES		NULL	
name	varchar(128)	NO	MUL	NULL	
available_now	varchar(255)	YES		NULL	
available_later	varchar(255)	YES		NULL	

11 rows in set (0.00 sec)

Koodinäide 12: PrestaShop tabelite ülevaade.

2.2.8. VirtueMart



Joonis 15: VirtueMart logo

Ametlikuks veebileheküljeks on <http://www.virtuemart.net/>. Töös kasutatakse 1.1.7a VirtueMart versiooni koos 1.5.22 Joomla! versiooniga.

VirtueMart on vabatarkvara e-turunduse lahendus, mida kasutatakse moodulitena Joomla! sisuhaldussüsteemiga. Joomla! ja VirtueMart on kirjutatud PHP programmeerimiskeele abil ning need kasutatakse standardses PHP/MySQL keskkondades (VirtueMart, 2011).

VirtueMart on phpShopil põhinev e-turunduse süsteem (Differences to phpShop, 2011). phpShop süsteemi selles töös aga ei vaadeta, sest see on VirtueMart süsteemiga sarnane.

Server, kuhu saab VirtueMart süsteemi panna, peab toetama järmiseid asju:

- Apache 1.3.19 veebiserver (2.2.x soovitatud).
- PHP 4.2.x toetust (5.2.x soovitatud).
- MySQL 3.23.x (5.0.x soovitatud).
- Joomla! 1.0.x (1.5.x soovitatud).

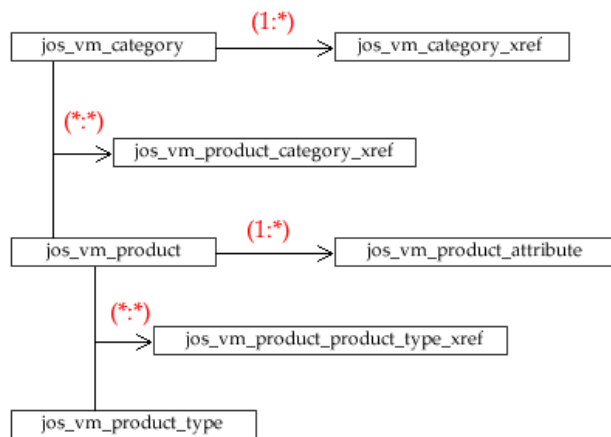
Tuleb arvestada ka sellega, et PHP peab toetama ka MySQL, XML ning Zlib funktsioone. PHP peab olema kompileeritud openssl ning cURL toetamisega.

Tabelis 8 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 8: VirtueMart failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/VirtueMart_1.1.7a-COMPLETE_PACKAGE.j15.zip	2269272	2.17
/files/shops_zipped/VirtueMart_1.1.7a-COMPLETE_PACKAGE.j15.zip.md5	77	0.000074
/files/dependencies/Joomla_1.5.22-Stable-Full_Package.zip	6919357	6.6
/files/dependencies/Joomla_1.5.22-Stable-Full_Package.zip.md5	72	0.000069
Maht serveri kõvakettal:	28452077	27.14
Andmebaasi faili suurus:	312964	0.3

Andmebaasis igal individuaalsel tootel on olemas üks rida *jos_vm_product* tabelis, kus veergudes on olemas nii tavalised toote väärtused, kuid ka rea identifikaatorid teistest tabelitest. Üldiselt on olemas kaks 1:* seost (*jos_vm_product* ja *jos_vm_product_attribute* tabelite vahel, kus *jos_vm_product_attribute* tabelis asuvad erinevad toodete eritribuudid, ning *jos_vm_category* ja *jos_vm_category_xref* tabelite vahel, mis on kirjeldatud *jos_vm_category_xref* tabelis. See viimane saab olla tegelikult (*:*), aga kuna *category_child_id* tabelis on meil primary key väärtus, on see ka unikaalne) ning kaks **: * seost (*jos_vm_product* ja *jos_vm_product_type* tabelite vahel, mis on kirjeldatud *jos_vm_product_product_type_xref* tabelis, ning *jos_vm_product* ja *jos_vm_category* tabelite vahel, mis on kirjeldatud *jos_vm_product_category_xref* tabelis) (vt joonis 16 ja koodinäide 13).



Joonis 16: VirtueMart andmebaasi struktuur

```
mysql> DESCRIBE `jos_vm_category`;
```

Field	Type	Null	Key	Default	Extra
category_id	int(11)	NO	PRI	NULL	auto_increment
vendor_id	int(11)	NO	MUL	0	
category_name	varchar(128)	NO	MUL		
category_description	text	YES		NULL	
category_thumb_image	varchar(255)	YES		NULL	
category_full_image	varchar(255)	YES		NULL	
category_publish	char(1)	YES		NULL	
cdate	int(11)	YES		NULL	
mdate	int(11)	YES		NULL	
category_browsepage	varchar(255)	NO		browse_1	
products_per_row	tinyint(2)	NO		1	
category_flypage	varchar(255)	YES		NULL	
list_order	int(11)	YES		NULL	

13 rows in set (0.00 sec)

```
mysql> DESCRIBE `jos_vm_category_xref`;
```

Field	Type	Null	Key	Default	Extra
category_parent_id	int(11)	NO	MUL	0	
category_child_id	int(11)	NO	PRI	0	
category_list	int(11)	YES	MUL	NULL	

3 rows in set (0.00 sec)

```
mysql> DESCRIBE `jos_vm_product`;
```

Field	Type	Null	Key	Default	Extra
product_id	int(11)	NO	PRI	NULL	auto_increment
vendor_id	int(11)	NO	MUL	0	
product_parent_id	int(11)	NO	MUL	0	
product_sku	varchar(64)	NO	MUL		
product_s_desc	varchar(255)	YES		NULL	
product_desc	text	YES		NULL	
product_thumb_image	varchar(255)	YES		NULL	
product_full_image	varchar(255)	YES		NULL	
product_publish	char(1)	YES		NULL	
product_weight	decimal(10,4)	YES		NULL	
product_weight_uom	varchar(32)	YES		pounds.	
product_length	decimal(10,4)	YES		NULL	
product_width	decimal(10,4)	YES		NULL	
product_height	decimal(10,4)	YES		NULL	
product_lwh_uom	varchar(32)	YES		inches	
product_url	varchar(255)	YES		NULL	
product_in_stock	int(11)	NO		0	

product_available_date	int(11)	YES		NULL	
product_availability	varchar(56)	NO			
product_special	char(1)	YES		NULL	
product_discount_id	int(11)	YES		NULL	
ship_code_id	int(11)	YES	MUL	NULL	
cdate	int(11)	YES		NULL	
mdate	int(11)	YES		NULL	
product_name	varchar(64)	YES	MUL	NULL	
product_sales	int(11)	NO		0	
attribute	text	YES		NULL	
custom_attribute	text	NO		NULL	
product_tax_id	int(11)	YES		NULL	
product_unit	varchar(32)	YES		NULL	
product_packaging	int(11)	YES		NULL	
child_options	varchar(45)	YES		NULL	
quantity_options	varchar(45)	YES		NULL	
child_option_ids	varchar(45)	YES		NULL	
product_order_levels	varchar(45)	YES		NULL	

35 rows in set (0.00 sec)

mysql> DESCRIBE `jos_vm_product_attribute`;

Field	Type	Null	Key	Default	Extra
attribute_id	int(11)	NO	PRI	NULL	auto_increment
product_id	int(11)	NO	MUL	0	
attribute_name	char(255)	NO	MUL		
attribute_value	char(255)	NO			

4 rows in set (0.00 sec)

mysql> DESCRIBE `jos_vm_product_category_xref`;

Field	Type	Null	Key	Default	Extra
category_id	int(11)	NO	MUL	0	
product_id	int(11)	NO	MUL	0	
product_list	int(11)	YES	MUL	NULL	

3 rows in set (0.00 sec)

mysql> DESCRIBE `jos_vm_product_product_type_xref`;

Field	Type	Null	Key	Default	Extra
product_id	int(11)	NO	MUL	0	
product_type_id	int(11)	NO	MUL	0	

2 rows in set (0.00 sec)

mysql> DESCRIBE `jos_vm_product_type`;

Field	Type	Null	Key	Default	Extra
product_type_id	int(11)	NO	PRI	NULL	auto_increment
product_type_name	varchar(255)	NO			
product_type_description	text	YES		NULL	
product_type_publish	char(1)	YES		NULL	
product_type_browsepage	varchar(255)	YES		NULL	
product_type_flypage	varchar(255)	YES		NULL	
product_type_list_order	int(11)	YES		NULL	

7 rows in set (0.00 sec)

Koodinäide 13: VirtueMart tabelite ülevaade.

2.2.9. ZenCart



Joonis 17: ZenCart logo

Ametlikuks veebileheküljeks on <http://www.zen-cart.com/>. Töös kasutatakse 1.3.9h ZenCart versiooni.

ZenCart on e-turunduse vabatarkvara süsteem, mis on osCommerce süsteemi põhine.

ZenCart süsteemi arendavad e-poe kasutajad, programmeerijad, disainerid ja konsultandid, kes arvavad, et e-turunduse veebidisain saab olla ning peab olema tehtud teisiti (ZenCart, 2011).

Minimaalsed serveri omadused on PHP 4.3.2, Apache 1.3.x, MySQL 3.2.x. Ideaalsed on aga PHP 4.4.x, Apache 1.3.x ning MySQL 3.2.x või 4.0.x.

Isegi kui öeldakse, et ZenCart on tehtud teisiti, andmebaasi struktuurid selles on sarnased osCommerce süsteemiga.

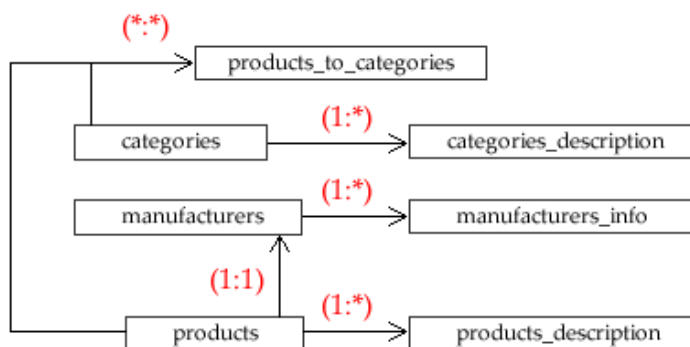
Tabelis 9 on kirjeldatud süsteemiga seotud failid suurustega.

Tabel 9: ZenCart failid.

Failinimi	Maht baitides	Maht megabaitides
/files/shops_zipped/zen-cart-v1.3.9h-full-fileset-10262010.zip	4463316	4.26
/files/shops_zipped/zen-cart-v1.3.9h-full-fileset-10262010.zip.md5	77	0.000074
Maht serveri kõvakettal:	10671555	10.18
Andmebaasi faili suurus:	526634	0.5

Andmebaasis igal individuaalsel tootel on olemas üks rida *products* tabelis, kus veergudes on olemas nii tavalised toote väärtused, kuid ka rea identifikaatorid teistest tabelitest. Nende identifikaatorite hulgas on 1:* seosed (*products* ja *products_description* tabelite vahel, kus *products_description* tabelis asuvad toodete kirjeldusetekstid erinevates keeltes), 1:1 seosed (*products* ja *manufacturers* tabelite vahel, kus *manufacturers* tabelis asub toodete firma nimed, kus on ka veel üks 1:* seos *manufacturers* ja *manufacturers_info* tabelite vahel, kus *manufacturers_info* tabelis asub firma infotekstid erinevates keeltes) ja *.* seosed (kus meil *products_to_categories*

tabelis on kirjeldatud seosed *products* ja *categories* tabelite vahel). On ka veel üks 1:* seos *categories* ja *categories_description* tabelite vahel, mis *categories_description* tabelis asub categoryate infotekstid erinevates keeltes. Tegelikult kuna ZenCart on osCommerce süsteemi põhine, andmebaasi struktuur selles süsteemis on sama (vt joonis 18 ja koodinäide 14).



Joonis 18: ZenCart andmebaasi struktuur

```

mysql> DESCRIBE `categories`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| categories_id  | int(11)       | NO   | PRI | NULL    | auto_increment |
| categories_image | varchar(64)   | YES  |     | NULL    |               |
| parent_id     | int(11)       | NO   | MUL | 0       |               |
| sort_order    | int(3)        | YES  |     | NULL    |               |
| date_added    | datetime      | YES  |     | NULL    |               |
| last_modified  | datetime      | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DESCRIBE `categories_description`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| categories_id  | int(11)       | NO   | PRI | 0       |               |
| language_id   | int(11)       | NO   | PRI | 1       |               |
| categories_name | varchar(32)   | NO   | MUL | NULL    |               |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DESCRIBE `manufacturers`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| manufacturers_id | int(11)       | NO   | PRI | NULL    | auto_increment |
| manufacturers_name | varchar(32)   | NO   | MUL | NULL    |               |
| manufacturers_image | varchar(64)   | YES  |     | NULL    |               |
| date_added    | datetime      | YES  |     | NULL    |               |
| last_modified  | datetime      | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> DESCRIBE `manufacturers_info`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| manufacturers_id | int(11)       | NO   | PRI | NULL    |               |
| languages_id    | int(11)       | NO   | PRI | NULL    |               |
+-----+-----+-----+-----+-----+-----+

```

manufacturers_url	varchar(255)	NO		NULL	
url_clicked	int(5)	NO		0	
date_last_click	datetime	YES		NULL	

5 rows in set (0.00 sec)

mysql> DESCRIBE `products`;

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	auto_increment
products_quantity	int(4)	NO		NULL	
products_model	varchar(12)	YES	MUL	NULL	
products_image	varchar(64)	YES		NULL	
products_image_med	varchar(64)	YES		NULL	
products_image_lrg	varchar(64)	YES		NULL	
products_image_sm_1	varchar(64)	YES		NULL	
products_image_xl_1	varchar(64)	YES		NULL	
products_image_sm_2	varchar(64)	YES		NULL	
products_image_xl_2	varchar(64)	YES		NULL	
products_image_sm_3	varchar(64)	YES		NULL	
products_image_xl_3	varchar(64)	YES		NULL	
products_image_sm_4	varchar(64)	YES		NULL	
products_image_xl_4	varchar(64)	YES		NULL	
products_image_sm_5	varchar(64)	YES		NULL	
products_image_xl_5	varchar(64)	YES		NULL	
products_image_sm_6	varchar(64)	YES		NULL	
products_image_xl_6	varchar(64)	YES		NULL	
products_price	decimal(15,4)	NO		NULL	
products_date_added	datetime	NO	MUL	NULL	
products_last_modified	datetime	YES		NULL	
products_date_available	datetime	YES		NULL	
products_weight	decimal(5,2)	NO		NULL	
products_status	tinyint(1)	NO		NULL	
products_tax_class_id	int(11)	NO		NULL	
manufacturers_id	int(11)	YES		NULL	
products_ordered	int(11)	NO		0	
products_bundle	tinytext	NO		NULL	

35 rows in set (0.00 sec)

mysql> DESCRIBE `products_description`;

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	auto_increment
language_id	int(11)	NO	PRI	1	
products_name	varchar(64)	NO	MUL	NULL	
products_description	text	YES		NULL	
products_url	varchar(255)	YES		NULL	
products_viewed	int(5)	YES		0	

6 rows in set (0.00 sec)

mysql> DESCRIBE `products_to_categories`;

Field	Type	Null	Key	Default	Extra
products_id	int(11)	NO	PRI	NULL	
categories_id	int(11)	NO	PRI	NULL	

2 rows in set (0.00 sec)

mysql> DESCRIBE `products_attributes`;

Field	Type	Null	Key	Default	Extra
products_attributes_id	int(11)	NO	PRI	NULL	auto_increment
products_id	int(11)	NO	MUL	NULL	
options_id	int(11)	NO		NULL	
options_values_id	int(11)	NO		NULL	
options_values_price	decimal(15,4)	NO		NULL	

price_prefix	char(1)	NO		NULL	
sort_order	int(11)	NO		50	
attribute_sort	int(10) unsigned	NO		0	

8 rows in set (0.00 sec)

mysql> DESCRIBE `products_options`;

Field	Type	Null	Key	Default	Extra
products_options_id	int(11)	NO	PRI	0	
language_id	int(11)	NO	PRI	1	
products_options_name	varchar(32)	NO		NULL	

3 rows in set (0.00 sec)

mysql> DESCRIBE `products_options_values`;

Field	Type	Null	Key	Default	Extra
products_options_values_id	int(11)	NO	PRI	0	
language_id	int(11)	NO	PRI	1	
products_options_values_name	varchar(64)	NO		NULL	

3 rows in set (0.00 sec)

mysql> DESCRIBE `products_options_values_to_products_options`;

Field	Type	Null	Key	Default	Extra
products_options_values_to_products_options_id	int(11)	NO	PRI	NULL	auto_increment
products_options_id	int(11)	NO		NULL	
products_options_values_id	int(11)	NO		NULL	

3 rows in set (0.00 sec)

Koodinäide 14: ZenCart tabelite ülevaade.

3. Andmemudelid

3.1. "ROW" andmemudel

Row andmemudel tähendab, et andmed (näiteks toode atribuudid, populaarsus, kommentaarid jne.) on jagatud erinevate tabelite vahel. Toodete tabel sisaldab ainult staatilist toode infot, nagu hind, nimi ja kirjeldus. Samal ajal on olemas ka teised tabelid kommentaaride ning ostupäringute jaoks, kus on olemas andmed iga toode kohta.

See on traditsiooniline relatsioonilise andmebaasi andmemudel.

3.2. "EAV" andmemudel

EAV andmemudeliks nimetatakse Entity-attribute-value andmemudelit. Matemaatika valdkonnas seda struktuuri nimetatakse hõredaks maatriksiks. "EAV" andmemudel on traditsioonilise "ROW" andmemudeli üldistus.

EAV representatsiooni palju kasutatakse meditsiini valdkonnas patsientide kliiniliste kirjade hoidmiseks andmebaasides. Selle potentsiaal annab võimalust kasutada seda ka teistes biomeditsiini valdkondades – uurimisandmebaasides, mille skeemid on keerulised ning kogu aeg muutuvad, et saaks toetada teaduste arendusi kiiresti arenevas teadusvaldkonnas (Nadkarni, Marengo, Chen, Skoufos, Shepherd, Miller, 1999).

Traditsioonilised tabelid on tehtud niimoodi, et ühele asjale (tootele või atribuudile) vastab üks rida, mille sees on palju veerge, kus horisontaalselt kirjeldatud kõik parameetrid. Sellele saab ka lisada spetsiifilised igale atribuutile tabelid, kus on sama olukord. Tihti ridade arv, mis on seotud ühe loogilise asjaga – toodega või atribuutiga, on üks.

Kontrastiks, EAV tabel sisaldab ainult kolm veergu – objekti identifikaator, atribuudi identifikaator, ning väärtus. Selles tabelis üks rida on üks "atribuut-väärtus" ühik (Nadkarni et al., 1999).

3.3. "ROW" ja "EAV" võrdlemine

Vaatame mingi tavalise e-turunduse süsteemi andmebaasi struktuuri.

Traditsiooniliselt on olemas mingi *products* tabel, kus asuvad toodete tähtsamad väärtused, mille arv

on kogu aeg üks (üks veerg nimele, üks veerg hinnale jne.), ning nendel ei saa olla näiteks erinevaid tõlkimisi. Lisaks sellele on olemas tabelid, nagu *categories*, *sales* jne. Lisaks sellele on olemas ka veel *categories_lang* ja *products_to_categories*. Kõik üldised andmed asuvad *products* tabelis; kategooriate andmed asuvad *categories* tabelis, mille väärtuste tõlkimised asuvad *categories_lang* tabelis; toodete ja kategooriate seosed on kirjeldatud *products_to_categories* tabelis; ning *sales* tabelis on kirjeldatud müügi päringud hinnadega, kus igale reale vastab üks toode. Andmed on päris hästi jagatud erinevate tabelite vahel, mis aitab kasutajale toodete vaadates ei näe andmeid, mis on ainult müüjale kasulik. *Products* tabel on siis objektide tabel, *categories* tabel on atribuutide tabel ning *products_to_categoreis* on EAV tabel. Ilmselt selles struktuuris igale atribuutitüübile vastavad iseseisvad tabelid (kategooriad asuvad *categories* tabelis, võimalik aga veel lisada sinna mingi samatüübilisi *colors* või *types* tabelleid).

Tegelikult EAV andmemudel on ROW mudeli üldistus. Eelnevalt kirjeldatud e-turunduse süsteemi EAV kasutades saab ellu viia järgmiselt: näiteks meil on olemas *attributes* tabel, kui on kirjeldatud kõik omadused (kategooriad, hinnagrupid, suurusegrupid jne). On olemas ka *products* tabel, kus meil asuvad toodete objektide staatilised andmed (tegelikult saab kõik teha ilma seda tabelita, aga see oleks programmiselt keerulisem ellu viia). On olemas ka *eav_data* tabel, kus meil kõik EAV kirjad ning toodete objektide ja atribuutide seosed olemas.

4. Süsteemide andmebaaside arendamine

4.1. Piirangud ja kommentaarid

Nagu see oli eelnevalt mainitud, selles töös võrreldakse ainult andmebaasi struktuuri arendamisi ja mitte koodi arendamisi ja teist asju, mis saavad olla seotud muutmistega andmebaasis.

Analüüs ei tähenda tööd konkreetse andmete sisse panemisega või ülekandega uuetele struktuuridele, vaid lihtsalt näitab kui palju tööd tuleb arendajal teha, et ellu viia vajalikke muutmisi.

Analüüsitakse ka seda, kui palju sõltub süsteemi paindlikkus (*:*) seoste arvest süsteemide andmebaaside "ROW" ja "EAV" struktuurides.

Vaadetakse ainult toodete ning sellega seotuid tabeleid.

4.2. Võrdlemine ning vajalikud arendussammud

4.2.1. Teoreetiline võrdlemine

Vaatame missugused seosed on olemas meie süsteemides, ning kui palju neid seal on. Paneme andmeid tabeli sisse arvestades sellega, et (1:1) seos on kõige ebapaindlik, nii et kui süsteemi andmebaasi struktuuris on see olemas, siis arendajal tuleb mingil juhul kinslasti midagi käsiti tegema. Järgmiselt vaatame kui palju 1:* seoseid on olemas, see on natukene paindlikum, kui (1:1) seos, aga see pole ka paindlikum. Lõpuks, kui (1:1) ning (1:*) seoseid meil üldse pole, siis vaatame kui palju meil (*:*) seoseid olemas (kui neid üldse ka pole, siis süsteem on ühetabeline, kus andmeid on väga vähe või andmete paigaldamine on implementeeritud veerge kasutades, aga see on juba nagu (1:1) seos). Nii et kõige paindlikum on selline süsteem, kus üldse pole nii (1:1) seoseid, kuid ka (1:*) seoseid, ning (*:*) seoste arv on nullist erinev. Kui suurem (*:*) seoste arv, nii keerulisem süsteem on. Tabelis 10 on kirjeldatud tulemused.

Tabel 10: Süsteemide seoste võrdlemine

#	Süsteemi nimi	(1:1) seos	(1:*) seos	(*:*) seos
1.	Magento	0	0	1
2.	CubeCart	0	2	1
3.	VirtueMart	0	2	2
4.	OpenCart	1	2	1

5.	DigiStore, osCommerce, PrestaShop ja ZenCart	1	3	1
6.	eCommerce	2	1	0

Selle meetodi kasutamisel saab näha, et Magento e-turunduse süsteem on teoreetiliselt kõige paindlikum. Teisel kohal on meil CubeCart, kolmandal on VirtueMart ning neljandal on OpenCart. Viimane süsteem (OpenCart) võib olla paigutatud viiendale kohale teiste osCommerce põhinevate süsteemide suhtes, sest sellel on vähem (1:*) seost sellepärast, et *manufacturers_info* tabel siin lihtsalt puudub. Tegelikult viiendal kohal on neli süsteemi: DigiStore, osCommerce, PrestaShop ja ZenCart (need on osCommerce süsteemiga sarnased ning olid pannud tabeli sisse lihtsalt tähestikuliselt). Kuuendal kohal on aga eCommerce Drupal moodul, kus kategooriate lahendus on üldse kõige paindumatum.

4.2.2. Praktiline võrdlemine

4.2.2.1. Uue atribuudi lisamine

Nüüd vaatame mida tuleb meie süsteemides praktiliselt teha andmebaasi struktuuri arendamisel. Arvestame sellega, et tuleb igale süsteemile lisada üks uus atribuut (näiteks, kommentaarid, mis saaksid olla mitte ainult ühest tootest, vaid ka mingist kolleksioonist) (*:*) seose kasutamisel. Kuna süsteemide struktuurid on erinevad ning nende struktuuride võimalused on muidugi erinevad, vaatame lihtsalt kui palju tuleb tööd teha, kui tuleb süsteeme arendada, mis kasutavad konkreetset struktuuri. Alustame lõpust.

eCommerce süsteemi atribuudi lisamine on kirjeldatud koodinäites 15. Lihtsustab protsessi see asi, et süsteem ei toeta erinevaid tõlkimisi. Nii et tegeleme ainult kahte tabelite lisamisega.

```
mysql> CREATE TABLE `ec_comment`(comment_id INT PRIMARY KEY auto_increment, comment_message
VARCHAR(255), date_added datetime);
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> DESCRIBE `ec_comment`;
```

Field	Type	Null	Key	Default	Extra
comment_id	int(11)	NO	PRI	NULL	auto_increment
comment_message	varchar(255)	YES		NULL	
date_added	datetime	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> CREATE TABLE `ec_product_comment`(product_id INT, comment_id INT, PRIMARY KEY
(`product_id`,`comment_id`));
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> DESCRIBE `ec_product_comment`;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| product_id | int(11)   | NO   | PRI | 0       |       |
| comment_id | int(11)   | NO   | PRI | 0       |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Koodinäide 15: eCommerce süsteemi atribuudi lisamine.

Digistore, osCommerce, PrestaShop, ZenCart, CubeCart ja OpenCart süsteemide arendust saab kirjeldada universaalselt ühe lahendusega, mis sobib igale süsteemile (vt. koodinäide 16). Tuleb lisada kolm tabelit (kaks tabelit OpenCart süsteemis).

```
mysql> CREATE TABLE `comments`(comments_id INT PRIMARY KEY auto_increment, date_added datetime);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> DESCRIBE `comments`;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| comments_id | int(11)   | NO   | PRI | NULL    | auto_increment |
| date_added  | datetime  | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> CREATE TABLE `comments_description`(comments_id INT, language_id INT default 1,
comments_message VARCHAR(255) NOT NULL, PRIMARY KEY (`comments_id`,`language_id`),INDEX
`idx_comments_message`(`comments_message`));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> DESCRIBE `comments_description`;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| comments_id | int(11)   | NO   | PRI | 0       |       |
| language_id | int(11)   | NO   | PRI | 1       |       |
| comments_message | varchar(255) | NO   | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> CREATE TABLE `products_to_comments`(products_id INT, comments_id INT, PRIMARY KEY
(`products_id`,`comments_id`));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> DESCRIBE `products_to_comments`;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| products_id | int(11)   | NO   | PRI | 0       |       |
| comments_id | int(11)   | NO   | PRI | 0       |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Koodinäide 16: osCommerce süsteemist pärinevate süsteemide atribuudi lisamine.

VirtueMart süsteemiga tuleb töötada niimoodi, nagu see on kirjeldatud koodinäites 17. Lisame kolm tabelit.

```

mysql> CREATE TABLE `jos_vm_comment`(comment_id INT PRIMARY KEY auto_increment, comment_message
VARCHAR(255), cdate INT, INDEX `idx_category_name`(`comment_message`));
Query OK, 0 rows affected (0.07 sec)

mysql> DESCRIBE `jos_vm_comment`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| comment_id     | int(11)       | NO   | PRI | NULL    | auto_increment |
| comment_message | varchar(255)  | YES  | MUL | NULL    |                |
| cdate          | int(11)       | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> CREATE TABLE `jos_vm_comment_xref`(comment_parent_id INT NOT NULL default 0, comment_child_id
INT PRIMARY KEY default 0, INDEX `comment_xref_comment_parent_id`(`comment_parent_id`));
Query OK, 0 rows affected (0.04 sec)

mysql> DESCRIBE `jos_vm_comment_xref`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| comment_parent_id | int(11)       | NO   | MUL | 0       |                |
| comment_child_id  | int(11)       | NO   | PRI | 0       |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> CREATE TABLE `jos_vm_product_comment_xref`(comment_id INT NOT NULL default 0, product_id INT
NOT NULL default 0, INDEX `idx_product_category_xref_category_id`(`comment_id`), INDEX
`idx_product_category_xref_product_id`(product_id));
Query OK, 0 rows affected (0.03 sec)

mysql> DESCRIBE `jos_vm_product_comment_xref`;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| comment_id     | int(11)       | NO   | MUL | 0       |                |
| product_id     | int(11)       | NO   | MUL | 0       |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

Koodinäide 17: VirtueMart süsteemi atribuudi lisamine.

Magento süsteemis aga saab lihtsalt lisada uue atribuudi olemasolevasse tabelisse, mis oleks meil kommentaaridega seotud, ning pärast seda saab juba otse hakata süsteemi kasutama.

Süsteemide uute tabelite arv uue atribuudi lisamisel on kirjeldatud tabelis 11. Need on arvud, mille jagu süsteemid suurenevad iga arendamise ajal, mis on seotud toodete struktuuridega.

Tabel 11: Süsteemide uued tabelid uue atribuudi lisamisel

#	Süsteemi nimi	Uute tabelite arv
1.	Magento	0
2.	eCommerce	2
3.	OpenCart	2
4.	CubeCart, DigiStore, osCommerce, PrestaShop, ZenCart ja VirtueMart	3

Süsteemi arendamine on enam vähem samasugune igas süsteemis. Mõnedes on aga päris piiratud funktsionaalsus (eCommerce) või tõlkimisvõimaluste puudumine. Magento aga on üks erijuhtum. Selles süsteemis uue atribuudi lisamine on ülilihtne ning arendaja ei pea lisama uut tabelit. Muidugi tuleb PHP koodis muutmisi teha, aga see meil küll ka igas teises süsteemis juhtub.

4.2.2.2. Olemasolevate struktuuride arendamine

Et saada lõplikku pilti, vaatame olemasolevate struktuuride modifitseerimisprotsessi süsteemides. Kuna tegelikult kõikides süsteemides kategooriate toetus on lahendatud (*:*) seoste abil, seda me ei vaata, kuid küll areneme teist tabelleid, et teha neid maksimaalselt paindlikumalt. Mõtleme sellest, et igal olukorral tuleb lisada uue tabeli, kus meil oleksid kirjeldatud objekti ja atribuudi vahel, või lihtsalt atribuudid, seoste tabeli olemasolu korral.

eCommerce süsteemis, esiteks, saab muuta (1:1) seost *ec_product* ja *node* tabelite vahel. Selleks tuleb teha uus tabel, kus oleksid need seosed kirjeldatud nii, et ühe leheküljega saaks olla seotud mitu toodet ning üks toode saaks olla erinevatel lehekülgedel. Teiseks, tuleb kindlasti muuta (1:1) seost *ec_product* ja *ec_product_ptypes* tabelite vahel. Siin lisame ka uue tabeli, kus oleksid need seosed kirjeldatud. Üks (1:*) seos *node* ja *node_revisions* meil jääb, sest loogilisest seda küll ei saa paindlikumalt teha ning tegelikult selle osa arendamine (*:*) seoks meile paindlikkust ei anna. Seda ka küll võtta välja ei saa, teistes süsteemides sellist funktsiooni ei ole. Nii et meil jäävad kaks (*:*) ning üks (1:*) seost.

DigiStore, osCommerce, PrestaShop ja ZenCart süsteemides, esiteks, tuleb muuta kolm (1:*) seost: *categories* ja *categories_description* tabelite vahel, *manufacturers* ja *manufacturers_description* tabelite vahel ning *products* ja *products_description* tabelite vahel. Need on *ps_category* ja *ps_category_lang*, *ps_manufacturer* ja *ps_manufacturer_info* ning *ps_product* ja *ps_product_lang* tabelid PrestaShop süsteemis. Nii et üks kirjeldus saaks olla seotud mitme tootega, ning ühel tootel saaks olla mitu kirjeldust. Lisaks tuleb muuta üks (1:1) seos *products* ja *manufacturers* tabelite vahel (PrestaShop süsteemis need on *ps_product* ja *ps_manufacturer* tabelid), et, näiteks, saaks teha niimoodi, et mõned toode osad on erinevate tootjate omad. Tulemuseks on viis (*:*) seost.

OpenCart süsteemis muudame kaks (1:*) seost: *category* ja *category_description* tabelite vahel ja *product* ja *product_description* tabelite vahel. Pluss muudame üks (1:1) seos *product* ja *manufacturer*

tabelite vahel. Tulemus on: 4 (*:*) seost.

Sarnaste sammude kasutades same 4 (*:*) seost VirtueMart ning 3 (*:*) seost CubeCart süsteemides.

Magento süsteemiga midagi teha ei ole vaja, sest see on juba kõige optimaalsel kujul.

Tabelis 11 kirjeldatakse arendamistulemusi ning keerukustase, mis oleksid meie süsteemide kõige paindlikumatel kujudel.

Tabel 12: Arendatud süsteemid ning keerukus

#	Süsteemi nimi	(1:1) seos	(1:*) seos	(*:*) seos
1.	Magento	0	0	1
2.	CubeCart	0	0	3
3.	OpenCart	0	0	4
4.	VirtueMart	0	0	4
5.	DigiStore, osCommerce, PrestaShop ja ZenCart	0	0	5
6.	eCommerce	1	0	2

Tegelikult Magento süsteem sellel kujul on kõige keerulisem süsteem. Aga tuleb ka märgistada seda, et need kuus (*:*) seost on ainult üks “keeruline” seos – see on “EAV” andmemudel ning sellega me pole midagi teinud; ja teiste süsteemide seoste arv on moodustatud paljude tabelite kasutamisel, mis küll vastab “ROW” andmemudeli skeemile. Siin aga tegime andmebaasi struktuuris palju muutmisi. Üldiselt kuna “ROW” mudelis kogu aeg lisatakse minimaalselt kaks tabelit, mis suurendab andmebaasi üsna kiiresti, mis teeb sellega töötamise väga keerulisemaks, “EAV” mudelis töötakse ainult olemasolevate tabelitega, mis on üliefektiivne.

Kokkuvõte

“EAV” andmemudeli printsiibid e-turunduse valdkonnas on väga perspektiivsed ning efektiivsed. Selle kasutamist alustati vaid mõni aasta tagasi (Magento süsteem ilmus 2007-2008 aastatel). Vanad süsteemid, mis kasutavad traditsioonilist “ROW” andmemudelit, vajavad kogu aeg tähelepanu arendajate poolt, mis tähendab investeeringute vajadust. Nii et küsimus on selles, et miks investeerida süsteemidesse, mis vajavad pidevaid investeeringuid tulevikus, kui saab investeerida süsteemidesse, mis on paindlikumad, ning seal edasi arendamine toimub ainult kasutajaliideste kaudu.

Kuna traditsioonilistes “ROW” andmemudeli süsteemides saab paindlikkust ellu viia PHP mootori kasutamisel (mis tähendab, automaatset tabelite lisamist ning muutmist), see ka tähendab kogu andmebaasi piiranguteta suurendamist kiirusega 2-3 tabelit igal sammul. “EAV” süsteemides see suurendamine toimub aga ridade suurendamise abil, mis on aga rohkem tavaline andmebaasi arendamine.

Nagu töö peamine eesmärk oli pandud, töös tehti ülevaade sellest, kui hästi on internetis olevates süsteemides lahendatud võimalused andmebaasi universaalsuse arendamiseks ja kui lihtne see üldse seal on arendajatele ning kasutajatele. Tehti teoreetiline ülevaade sellest, millised andmestruktuurid saavad olla e-turunduse andmebaasides kasutanud; leiti vabatarkvara LAMP tehnoloogia e-turunduse valmissüsteemid; tehti ülevaade leitud süsteemidest; tehti ülevaade andmebaasi struktuuridest, millega leitud süsteemid töötavad; ning võrreldi arendusprotsesse, nii et oleks hästi nähtav kui palju aega võtab süsteemide arendamine, ning mitu sammu see võtab. Olid antud ka kommentaarid selle kohta, missugused teoreetilised ja praktilised võimalused süsteemides on olemas ning millised süsteemid on kõige paindlikumad.

Summary

Almost every software solution is being constantly developed. More often this is the case in e-commerce area, where administrators are constantly managing products and sometimes are in need to add new attributes. To make it possible for webshops to be constantly evolved to keep up with the market changes, e-commerce application developers need to do a lot of things, updating the codes as well as changing database structures, to meet new requirements.

This work is focused on “EAV” and “ROW” data structures.

“EAV” data model principles in e-commerce area are very perspective and effective. Their use in e-commerce systems was actively started only a few years ago (Magento e-commerce system was launched in 2007-2008). Old systems that are using “ROW” data model require constant developer’s attention that also means constant investments. The question is, why invest into a system that will require constant investments in future, when one can invest into a system that will be very flexible, and will enable it’s users to be developing it by using only graphical user interfaces.

In “ROW” data model databases flexibility could be achieved by PHP engine coding, that will be constantly altering and adding new tables with a speed of 2-3 tables per development stage. This development in “EAV” systems means simply additions of rows into regular and already existing tables, not additions of tables, which is more traditional way of database self-development.

This work is focused on database structures and not on actual codes that are working with the database. The main purpose of this work is to create an overview about how good database structures of existing systems are and how easy it is for developers to be managing them. The main criterion is universal self-development of database components. The idea is to find the best system.

To achieve the goal above, author does a theoretical overview about what database structures exist in e-commerce area applications; finds a suitable ready open source systems that are using Linux operation system, PHP programming language, Apache webserver as well as MySQL database engine (which are basically forming a so called LAMP technology); does a small overview about

found systems; does an overview about database structures that are used in found systems; compares development processes to see how much time it takes to develop the particular system; makes comments and writes a conclusion to show what practical options there are in found systems as well as what systems are the most flexible ones.

Kasutatud kirjandus.

1. Bryla, B. (2004). *Oracle database foundations*. San Francisco, London: SYBEX Inc.
2. Rob, P., Coronel, C., Crockett, K. (2008). *Database systems: design, implementation & management*. London: Cengage Learning EMEA.
3. Hammergren, T. C., Simon, A. R. (2009). *Data Warehousing for Dummies*. Indianapolis: Wiley Publishing, Inc.
4. Magento creates huge success with enterprise e-commerce platform & community built on Zend Framework. (2008). Retrieved April 20, 2011, from <http://framework.zend.com/casestudies/ZFCaseStudy-Magento.pdf>
5. Announcing the Magento Enterprise Edition. (2009). Retrieved April 20, 2011, from <http://www.magentocommerce.com/blog/comments/announcing-the-magento-enterprise-edition/>
6. Interview: Roy Rubin About Magento Open Source eCommerce. (2007). Retrieved April 20, 2011, from <http://www.ecommerceoptimization.com/articles/interview-roy-rubin-about-magento-open-source-e-commerce/>
7. Magento Database. (2011). Retrieved April 20, 2011, from http://www.magentocommerce.com/wiki/2_-_magento_concepts_and_architecture/magento_database_diagram
8. Magento for Developers: Part 7 - Advanced ORM - Entity Attribute Value (n.d.). Retrieved April 24, 2011, from <http://www.magentocommerce.com/knowledge-base/entry/magento-for-dev-part-7-advanced-orm-entity-attribute-value/magento-for-dev-part-5-magento-models-and-orm-basics>
9. VirtueMart. (2011). Retrieved April 20, 2011, from <http://virtuemart.net/>
10. Differences to phpShop. (2011). Retrieved April 20, 2011, from http://virtuemart.net/documentation/Developer_Manual/Differences_to_phpShop.html

11. ZenCart. (2011). Retrieved April 20, 2011, from <http://www.zen-cart.com/>

12. Nadkarni, P. M., Marenco, L.; Chen, R., Skoufos, E., Shepherd, G., Miller, P. (1999). Organization of Heterogeneous Scientific Data Using the EAV/CR Representation. *Journal of the American Medical Informatics Association*, 6, 478-493.