

**СОФИЙСКИ УНИВЕРСИТЕТ  
"СВ. КЛИМЕНТ ОХРИДСКИ"**

---

**Факултет по Математика и Информатика  
Катедра „Информационни Технологии“**

**ДИПЛОМНА РАБОТА**

**на тема:**

**Система за проектиране и моделиране на  
бази от данни CWD4ALL**

*Дипломант: Стефка Георгиева Канова, фак. № М-21191*

*Специалност: Приложна Математика*

*Специализация: Математическо моделиране в икономиката*

*Научен ръководител: доц. Силвия Илиева*

**София, 2006 г.**

# СЪДЪРЖАНИЕ

<u>Увод .....</u>	<u>85</u>
<u>ОСНОВНИ ЦЕЛИ НА РАЗРАБОТКАТА .....</u>	<u>96</u>
<u>СТРУКТУРА НА ДИПЛОМНАТА РАБОТА.....</u>	<u>96</u>
<u>ОБЗОР НА СИСТЕМИ ЗА ПРОЕКТИРАНЕ И МОДЕЛИРАНЕ НА БАЗИ ОТ ДАННИ .....</u>	<u>107</u>
<b><u>ГЛАВА 1. МОДЕЛ НА ДАННИ .....</u></b>	<b><u>139</u></b>
<b><u>ВЪВЕДЕНИЕ.....</u></b>	<b><u>139</u></b>
<b><u>ЦИКЪЛ НА МОДЕЛИРАНЕ.....</u></b>	<b><u>1510</u></b>
<b><u>Логически дизайн .....</u></b>	<b><u>1813</u></b>
<b><u>Документиране на Логическия дизайн .....</u></b>	<b><u>1813</u></b>
<b><u>Нормализация.....</u></b>	<b><u>1813</u></b>
<b><u>Реляционен дизайн .....</u></b>	<b><u>2015</u></b>
<b><u>Трансформация на Логически към Реляционен модел.....</u></b>	<b><u>2116</u></b>
<b><u>Денормализация.....</u></b>	<b><u>2217</u></b>
<b><u>Индексиране .....</u></b>	<b><u>2319</u></b>
<b><u>Физически дизайн .....</u></b>	<b><u>2419</u></b>
<b><u>Разширяване на Реляционния модел до Физически .....</u></b>	<b><u>2419</u></b>
<b><u>Индексиране .....</u></b>	<b><u>2420</u></b>
<b><u>Физическо съхранение .....</u></b>	<b><u>2520</u></b>
<b><u>ОБЕКТИ НА ЛОГИЧЕСКИЯ МОДЕЛ .....</u></b>	<b><u>2621</u></b>
<b><u>Ентити .....</u></b>	<b><u>2621</u></b>
<b><u>Атрибут.....</u></b>	<b><u>2722</u></b>
<b><u>Ключове .....</u></b>	<b><u>2722</u></b>
<b><u>Първичен ключ .....</u></b>	<b><u>2823</u></b>
<b><u>Кандидат-ключ .....</u></b>	<b><u>2823</u></b>
<b><u>Чужд ключ.....</u></b>	<b><u>2924</u></b>
<b><u>Връзка .....</u></b>	<b><u>2924</u></b>
<b><u>Група .....</u></b>	<b><u>3025</u></b>
<b><u>Наследяване .....</u></b>	<b><u>3025</u></b>
<b><u>Изглед .....</u></b>	<b><u>3227</u></b>
<b><u>ОБЕКТИ НА РЕЛЯЦИОННИЯ МОДЕЛ .....</u></b>	<b><u>3227</u></b>
<b><u>Таблица .....</u></b>	<b><u>3227</u></b>
<b><u>Колона.....</u></b>	<b><u>3428</u></b>
<b><u>Ключове .....</u></b>	<b><u>3429</u></b>
<b><u>Първичен ключ .....</u></b>	<b><u>3429</u></b>
<b><u>Уникален ключ .....</u></b>	<b><u>3429</u></b>
<b><u>Чужд ключ.....</u></b>	<b><u>3429</u></b>
<b><u>Индекс.....</u></b>	<b><u>3529</u></b>
<b><u>Изглед .....</u></b>	<b><u>3530</u></b>
<b><u>Група .....</u></b>	<b><u>3630</u></b>
<b><u>ОБЕКТИ НА МНОГОМЕРНИЯ МОДЕЛ .....</u></b>	<b><u>3631</u></b>
<b><u>Куб .....</u></b>	<b><u>3631</u></b>
<b><u>Мярка .....</u></b>	<b><u>3731</u></b>
<b><u>Measure Folder .....</u></b>	<b><u>3832</u></b>
<b><u>Измерение.....</u></b>	<b><u>3832</u></b>
<b><u>Иерархия.....</u></b>	<b><u>3933</u></b>
<b><u>Ниво.....</u></b>	<b><u>3933</u></b>
<b><u>Rollup Link.....</u></b>	<b><u>4034</u></b>
<b><u>ГЛАВА 2. ПРОЦЕС МОДЕЛ .....</u></b>	<b><u>4135</u></b>
<b><u>ВЪВЕДЕНИЕ.....</u></b>	<b><u>4135</u></b>

<b>Цикъл на моделиране.....</b>	<b>4236</b>
<b>Обекти на процес модела.....</b>	<b>4337</b>
Външен агент.....	4337
Информационен поток.....	4438
Информационна структура.....	4539
Информационно хранилище.....	4639
Процес.....	4640
Иерархия на процеси.....	4942
Роля.....	5143
Събитие.....	5144
<b>ГЛАВА 3. СТЕПЕН НА ПОДДРЪЖКА НА CWM™.....</b>	<b>5346</b>
<b>ВЪВЕДЕНИЕ.....</b>	<b>5346</b>
<b>ОБЪКТ MODEL PACKAGE.....</b>	<b>5648</b>
Core package.....	5748
Behavioral package.....	5849
Relationships package.....	5950
Instance package.....	6253
<b>FOUNDATION PACKAGE.....</b>	<b>6353</b>
Business Information package.....	6354
Data Types package.....	6455
Expressions package.....	6556
Keys and Indexes package.....	6656
Software Deployment package.....	6757
Type Mapping package.....	6858
<b>RESOURCE PACKAGE.....</b>	<b>6959</b>
Relational package.....	6959
Record package.....	7060
Multidimensional package.....	7161
XML package.....	7161
<b>ANALYSIS PACKAGE.....</b>	<b>7262</b>
Transformation package.....	7262
OLAP package.....	7464
Data Mining package.....	7565
Information Visualization package.....	7665
Business Nomenclature package.....	7665
<b>MANAGEMENT PACKAGE.....</b>	<b>7765</b>
Warehouse Process package.....	7765
Warehouse Operation package.....	7766
<b>ГЛАВА 4. ИЗПОЛЗВАНЕ НА CWD4ALL ПРИ РЕАЛИЗИРАНЕ НА</b>	
<b>КОНКРЕТНИ МОДЕЛИ.....</b>	<b>7967</b>
<b>МОДЕЛ HANDY HARDWARE COMPANY.....</b>	<b>7967</b>
<b>МОДЕЛ SH.....</b>	<b>8573</b>
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>8775</b>
<b>РЕЧНИК НА ИЗПОЛЗВАНИТЕ СЪКРАЩЕНИЯ И ТЕРМИНИ.....</b>	<b>8876</b>
<b>A.....</b>	<b>8876</b>
<b>B.....</b>	<b>8876</b>
<b>C.....</b>	<b>8876</b>

<b>D</b> .....	<b>8977</b>
<b>E</b> .....	<b>9077</b>
<b>F</b> .....	<b>9077</b>
<b>I</b> .....	<b>9078</b>
<b>M</b> .....	<b>9178</b>
<b>N</b> .....	<b>9178</b>
<b>P</b> .....	<b>9278</b>
<b>R</b> .....	<b>9279</b>
<b>S</b> .....	<b>9379</b>
<b>T</b> .....	<b>9379</b>
<b>U</b> .....	<b>9379</b>
<b>V</b> .....	<b>9379</b>
<b>ИЗПОЛЗВАНА ЛИТЕРАТУРА</b> .....	<b>9480</b>

<b>ДИПЛОМНА РАБОТА</b> .....	<b>1</b>
<b>Увод</b> .....	<b>5</b>
<b>ОСНОВНИ ЦЕЛИ НА РАЗРАБОТКАТА</b> .....	<b>5</b>
<b>СТРУКТУРА НА ДИПЛОМНАТА РАБОТА</b> .....	<b>5</b>
<b>ГЛАВА 1. МОДЕЛ НА ДАННИ</b> .....	<b>7</b>
<b>ВЪВЕДЕНИЕ</b> .....	<b>7</b>
<b>ЦИКЪЛ НА МОДЕЛИРАНЕ</b> .....	<b>8</b>
<b>Логически дизайн</b> .....	<b>11</b>
<b>Документиране на Логическия дизайн</b> .....	<b>11</b>

Нормализация.....	12
<b>Релационен дизайн.....</b>	<b>14</b>
Трансформация на Логически към Релационен модел.....	14
Денормализация.....	16
Индексиране.....	17
<b>Физически дизайн.....</b>	<b>18</b>
Разширяване на Релационния модел до Физически.....	18
Индексиране.....	18
Физическо съхранение.....	19
<b>ОБЕКТИ НА ЛОГИЧЕСКИЯ МОДЕЛ.....</b>	<b>20</b>
Ентити.....	20
Атрибут.....	20
Ключове.....	21
Primary key.....	21
Candidate key.....	21
Foreign Key.....	22
Връзка.....	23
Subview.....	23
Inheritance.....	23
View.....	25
<b>ОБЕКТИ НА РЕЛАЦИОННИЯ МОДЕЛ.....</b>	<b>25</b>
Таблица.....	25
Колона.....	27
Keys— Primary key, Foreign key.....	27
Primary key.....	27
Unique key.....	28
Foreign Key.....	28
Индекс.....	28
View.....	28
Subview.....	29
<b>ОБЕКТИ НА МНОГОМЕРНИЯ МОДЕЛ.....</b>	<b>29</b>
Cube.....	29
Measure.....	30
Measure Folder.....	31
Dimension.....	31
Hierarchy.....	32
Level.....	32
Rollup Link.....	33
<b>ГЛАВА 2. ПРОЦЕС МОДЕЛ.....</b>	<b>34</b>
<b>ВЪВЕДЕНИЕ.....</b>	<b>34</b>
<b>ЦИКЪЛ НА МОДЕЛИРАНЕ.....</b>	<b>35</b>
<b>ОБЕКТИ НА ПРОЦЕС МОДЕЛА.....</b>	<b>36</b>
External Agents.....	36
Information Flows.....	37
Information Structure.....	38
Information Stores.....	39
Processes.....	39
Process Hierarchy.....	42
Role.....	43
Process-Triggering Events.....	44
<b>ГЛАВА 3. СТЕПЕН НА ПОДДРЪЖКА НА CWM™.....</b>	<b>45</b>
<b>ВЪВЕДЕНИЕ.....</b>	<b>45</b>

<b>ОБЪКТ MODEL PACKAGE</b> .....	<b>48</b>
Core package .....	48
Behavioral package .....	49
Relationships package .....	50
Instance package .....	53
<b>FOUNDATION PACKAGE</b> .....	<b>54</b>
Business Information package .....	54
Data Types package .....	55
Expressions package .....	56
Keys and Indexes package .....	57
Software Deployment package .....	58
Type Mapping package .....	59
<b>RESOURCE PACKAGE</b> .....	<b>60</b>
Relational package .....	60
Record package .....	61
Multidimensional package .....	62
XML package .....	62
<b>ANALYSIS PACKAGE</b> .....	<b>63</b>
Transformation package .....	63
OLAP package .....	65
Data Mining package .....	66
Information Visualization package .....	67
Business Nomenclature package .....	67
<b>MANAGEMENT PACKAGE</b> .....	<b>68</b>
Warehouse Process package .....	68
Warehouse Operation package .....	68
<b>ЗАКЛЮЧЕНИЕ</b> .....	<b>69</b>
<b>РЕЧНИК НА ИЗПОЛЗВАНИТЕ СЪКРАЩЕНИЯ И ТЕРМИНИ</b> .....	<b>69</b>
<b>A</b> .....	<b>69</b>
<b>B</b> .....	<b>70</b>
<b>C</b> .....	<b>70</b>
<b>D</b> .....	<b>70</b>
<b>E</b> .....	<b>71</b>
<b>F</b> .....	<b>71</b>
<b>I</b> .....	<b>71</b>
<b>K</b> .....	<b>72</b>
<b>L</b> .....	<b>72</b>
<b>M</b> .....	<b>72</b>
<b>N</b> .....	<b>72</b>
<b>O</b> .....	<b>73</b>
<b>P</b> .....	<b>73</b>
<b>R</b> .....	<b>73</b>
<b>S</b> .....	<b>74</b>
<b>T</b> .....	<b>74</b>
<b>U</b> .....	<b>74</b>
<b>V</b> .....	<b>74</b>
<b>ИЗПОЛЗВАНА ЛИТЕРАТУРА</b> .....	<b>75</b>



## Увод

Тенденциите за разработване на все по-сложен бизнес софтуер водят до сложни дизайни и на бази от данни сложни, особено за системи, които използват стотици таблици. Администраторите на бази от данни и разработчиците често се справят с тази сложност като разработват/създават модел на данните преди да генерират базата.

Поради тази причина използването на ефикасна система за проектиране и моделиране на бази от данни е абсолютно наложително в днешно време. С помощта на подобен софтуер се унагледяват графично моделите и връзките между тях и лесно се редактират, докато се достигне до финалния/окончателния изчистен дизайн на базата данни. Като резултат се генерира автоматично DDL за създаване на дефинираните обекти в базата. Обратно, вече съществуващи дизайни биха могли да се модифицират, като се въведе (import) импортне съответстващия DDL, XMI или друг формат, в който се пазят.

Освен това, една такава система предлага опростяване и ускоряване на анализирането, проектирането и реализирането на моделите на данни, бизнес процесите и Business Intelligence средите, включва традиционните OLTP системи, data marts и data warehouse за аналитични обработки. Нещо повече – вече не са достатъчни стандартните средства за моделиране – необходима е поддръжка на CWM™ съвместимо многомерно моделиране и CWM™ съвместими аналитични обработки. Тя осигурява преносимост на метаданните между различни формати метаданни, което води до преносимост на дизайни, разработвани с помощта на различни системи за моделиране.

Дизайните на бази от данни могат да бъдат сложни, особено за системи, които използват стотици таблици. Администраторите на бази от данни (DBA) и разработчиците често се справят с тази сложност като разработват модел на данните преди да генерират базата. Опитните DBA и разработчици обикновено обръщат внимание на системи, които опростяват моделирането на данни. Един такъв продукт, предназначен за SQL и online analytical processing (OLAP), (online

**Comment [S1]:** Уводът да е по-дълъг – 1-2 страници  
Стила да е по-литературен, не разговорен

**Formatted:** Body Text, Left

**Formatted:** Bulgarian

**Formatted:** Bulgarian

**Formatted:** Bulgarian

**Formatted:** Bulgarian

**Formatted:** Bulgarian

**Formatted:** Russian

**Formatted:** Russian

**Formatted:** Font color: Black

**Formatted:** Font color: Black

**Formatted:** Bulgarian

**Formatted:** Bulgarian

**Formatted:** Bulgarian



analytical processing) средите, е CWD4ALL, особено за тези, които използват Oracle, IBM DB2 и Microsoft SQL Server. CWD4ALL е предназначен за warehouse разработчици и администратори, технологични мениджъри и крайни потребители. Той предоставя среда за лесно извличане, моделиране, управление, използване и обмен на метаданни.

Formatted: Bulgarian

## Основни цели на разработката

Целта на дипломната работа е да се проучи и приложи системата за проектиране и моделиране на бази от данни CWD4ALL. От поставената цел произлизат следните задачи:

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

- Да се представи модела на данни
- Да се покаже как се моделират бизнес функции и процеси
- Да се анализира степента на поддръжка на CWM™
- Да се използва системата при реализиране на конкретни модели

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm, Widow/Orphan control

## Структура на дипломната работа

Дипломната работа включва увод, четири глави, заклучение, речник на използваните съкращения и термини и списък с използвана литература.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Уводът съдържа кратко въведение в областта на дипломната работа, описват се темата и начинът на структуриране на изложението.

Глава 1 е посветена на моделирането на данни в CWD4ALL. Представени са видовете модели на данни в CWD4ALL, обектите, които включват, връзките и трансформациите между тях, както и стъпките при тяхното моделиране.

Глава 2 представя начина, по който с помощта на CWD4ALL се моделират процесите в дадена информационна система. Дефинирани са обектите на Процес модела с конкретни примери.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Left, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm, Widow/Orphan control

Глава 3 анализира степента на поддръжка на CWD4ALL на Common Warehouse Metamodel (CWM™).

В глава 4 са показани конкретни модели, реализирани със системата. Примерите, използвани в предишните глави 1 и 2, са взайствани тъкмо от тези модели.

**Comment [S2]:** предишните глави и конкретно изреди намирата на главите

В заключението са описани възможните насоки за развитие на системата с оглед съвременните потребности на анализаторите.

Следват речник на използваните в дипломната работа съкращения и термини и списък на информационните източници, използвани за дипломната работа.

~~Приложението съдържа генериран от CWD4ALL DDL от реализираните модели.~~

**Comment [S3]:** защо без приложение?

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

## **Обзор на системи за проектиране и моделиране на бази от данни**

**Formatted:** Russian

За да задоволят съвременните потребности на анализаторите, вече съществуват десетки системи за моделиране на бази данни. Някои от тях се налагат като най-използваните и поддържащи най-адекватна на нуждите на потребителите функционалност.

Embarcadero ER/Studio (вж. [13]) поддържа повече бази от CWD4ALL – освен IDM DB2 и UDB, SQL Server и Oracle, поддържа MySQL и Sybase. ER/Studio предлага доста удобен графичен интерфейс с множество wizards. Реализацията на сигурността за различни платформи, поддръжката на код за базите, инструментите за оценка и подобряване на бързодействието в базите определено са предимства.

**Comment [S4]:** Цитира се само в средни скоби номера Навсякъде в текста

Не са цитирани всички изчници в текста!!!!!!!!!!!!

За разлика от ER/Studio, CWD4ALL представя напълно разделено различните модели, като едновременно с това са ясни съответствията и начините за преобразуване между тях. Реализирана е концепцията за Релационен дизайн – междинно ниво между Логическия дизайн и Физическата му имплементация. Въведени са Многомерен модел, Процес модел, Модел на типовете, Business Information модел.

CA Erwin Data Modeler (вж. [15]) е система за моделиране само на модела на данни, AllFusion Process Modeler моделира Процес модела, а AllFusion Data Model Validator анализира базите и осигурява графични справки за ефективността и

бързодействието на Физическия дизайн. Erwin поддържа импорт и експорт на метаданни от различни системи – повече формати от CWD4ALL. Въпреки, че поддържа много бази (Access, DB2, FoxPro, Informix, Ingres, iSeries, Oracle, Progress, Red Brick, SAS, SQL Server, Sybase, Teradata), не е възможно да се работи едновременно върху няколко физически дизайна. Тук отново се проявява уникалното концептуално предимство на CWD4ALL – наличие на Релационен дизайн. Логическият дизайн може да се преобразува до няколко Релационни дизайна, всеки от които може да има различни Физически имплементации. Освен това, за разлика от CWD4ALL, с Erwin не могат да се моделират многомерни данни.

Formatted: Russian

Formatted: Russian

Sybase Powerdesigner (вж. [14]) поддържа следните модели: Business Process Model, Conceptual Data Model, Information Liquidity Model, Object-Oriented Model, Physical Data Model, Requirements Model, XML Model, Repository Model. Поддържа над 50 СУБД, както и UML диаграми – Use Case, Sequence, Data Flow, Component, Class и др. Той обединява инструменти за моделиране на приложения, на бизнес-процеси и на бази данни. Средствата за работа с бизнес процеси позволяват да се събират, анализират и документират изискванията, да се моделират процесите и потока данни и да се прави оптимизация. На основата на създаване на диаграми на класовете се генерират изходни текстове на програмите на различни обектно ориентирани езици, сред които VB.Net, C++, C# и Java.

Formatted: Russian

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: Font: Times New Roman, 12 pt, Font color: Black

Formatted: Font: Times New Roman, 12 pt, Font color: Black

Друго предимство на Sybase Powerdesigner пред CWD4ALL е централизираният репозитар, осигуряващ съвместна работа на няколко специалисти над един и същ модел. Той притежава и ефективен механизъм за съхраняване на версии на моделите и другите документи, за търсене на обекти в съществуващите модели и тяхното повторно използване в нови, както и за управление на взаимовръзки между моделите.

Formatted: Russian

Formatted: Russian

Formatted: Font: Times New Roman, 12 pt, Font color: Black

IBM Rational Data Architect (вж. [16]) предоставя стандартната функционалност за работа с диаграми на взаимовръзките между елементите и преобразуване на Физически към Логически модел. Той е насочен повече към извличане на данни от съществуващи бази и обработването им. Поддържа DB2, Derby, Informix, SQL Server, Oracle и Sybase. В сравнение с него CWD4ALL

Formatted: Russian

предлага доста по-богата функционалност, свързана с Релационен, Многомерен, Типов, Процес Модел, съответствията между тях, импорт и експорт на различни формати метаданни, както и поддръжка на голяма част от пакетите на CWM.

**Formatted:** Russian

**Formatted:** Font: Not Bold

**Formatted:** Heading 2; Знак Знак  
Знак

# Глава 1. Модел на данни

## Въведение

Моделът на данни представя значимите за бизнеса неща и връзките между тях. Той дефинира структурата на бизнес информацията и правилата, които я управляват. Моделът на данни се състои от няколко части – логически, реляционен, физически и многомерен дизайн.

Логическия модел се разработва преди многомерния, реляционния и физическия. Той набляга на бизнес и функционалните изисквания на системната разработка. Логическия дизайн позволява да се определи организацията на данните, които ще се съхраняват в базата, преди да се създаде самата база.

Логическият дизайн на CWD4ALL се представя от Entity-Relationship модел. Основната му идея се състои в следното: всяко понятие в бизнеса може да бъде обобщение на абстрактен или архитипен идеал, който се нарича единица/елемент/ентити (entity). Тези ентитита имат определени характеристики или атрибути. Освен това ентититата са свързани с други ентитита чрез действия, които всяко ентити извършва върху едно или повече други ентитита. Тези действия се наричат връзки relationship (relationship връзка).

Реляционният модел е базиран на реляционната алгебра и се фокусира върху независимостта на данните от физическата структура, където се съхраняват. Тази концепция е разработена от Чарли Бахман и е описана в [4] и [5]. По този начин се улеснява миграцията към по-големи и по-нови системи с минимални физически разлики между устройствата за съхранение (главно свързани с типовете данни и домейните).

Formatted: Font: Not Bold, Bulgarian

Formatted: Font: Not Bold

Comment [S5]: Така да е форматирани всеки параграф – с отстояние на първия ред, без празен ред след това, шрифт 12

Formatted: Font: Not Italic

Formatted: Bulgarian

Formatted: Bulgarian

Formatted: Font: Not Italic

Formatted: Bulgarian

Comment [S6]: Трябва да има български еквивалент – ентити не е българска дума

Formatted: Bulgarian

Formatted: Bulgarian

Formatted: Font: Verdana, Font color: Black, Not Hidden

Formatted: Bulgarian

Formatted: Font: Verdana, Not Italic

Formatted: Font: Not Italic

Formatted: Russian

Formatted: Russian

Силата на релационния модел се състои в неговата простота. В релационния модел данните са организирани в таблици от редове и колони. Всяка таблица представя един тип данни. Всеки ред представя една данна от този тип. Всяка колона или домейн представят един тип от информацията за типа данни съхранявани в таблицата.

Физическият дизайн представя техническата реализация на модела на данни, т.е. начина, по който данните се съхраняват в базата. Например, определя се типа на данните на всяка колона в таблиците и как всяка таблица ще се запазва в базата. Физическият модел е напълно зависим от вида на базата данни. CWD4ALL поддържа IBM DB/2 v7 & v8 за Z/OS, IBM DB/2 UDB v7 & v8, Oracle 8i, 9i, 10g и Microsoft SQL Server 2000 & 2005.

Многомерния дизайн е обобщено представяне на многомерна база от данни. Чрез него се постига моделиране за data warehouses, data marts и OLAP системи. В многомерните бази от данни, ключови OLAP конструкции (измерения, йерархии и т.н.) са представени от вътрешни структури от данни на многомерния сървър на базата, а OLAP операциите (consolidation, drill-down и т.н.) се изпълняват от сървъра върху тези структури от данни.

Многомерното представяне е оптимизирано за работа със сложни данни. Например, ако в базата данни се съхранява информация за продажби на дадена компания, променливата Sales може да се измерва с TIME, PRODUCT и GEOGRAPHY, така че няколко стъпки ще са необходими, за да се намерят 10-те града с най-много продажби на палатки за изминалите 3 месеца. В релационна база, за това ще е нужна сложна SQL заявка.

За разработване на възможно най-ефективния модел данни се препоръчва логическият дизайн да се разработи преди релационния и физическия. И логическият, и релационния, и физическият дизайн са сложни, затова е най-добре да се разделят, отколкото да се смесват.

**Comment [S7]:** Това може да е отделно изречение  
И в двете скоби на български и на английски думите

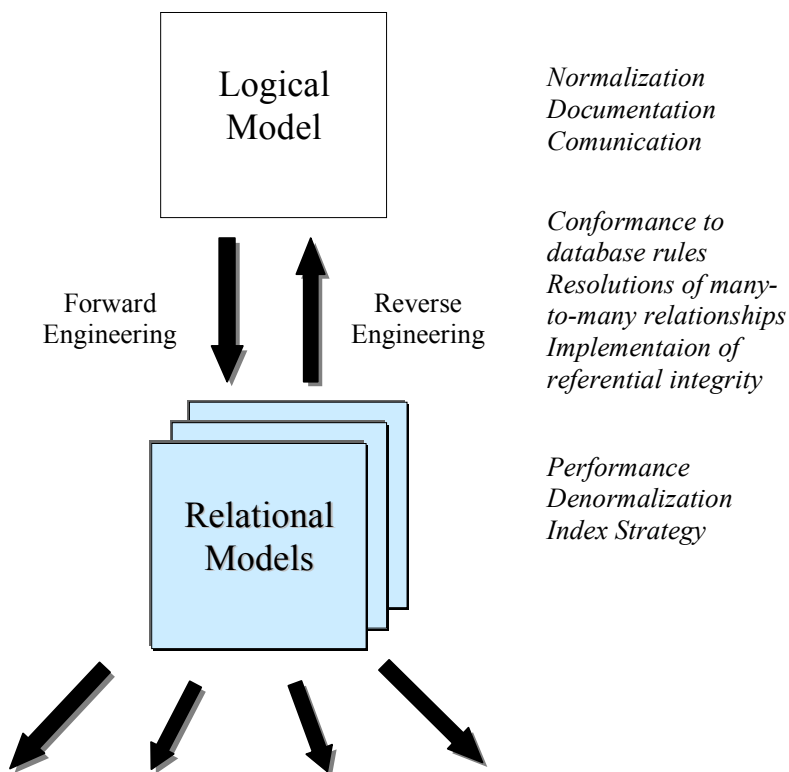
**Formatted:** Font: Verdana, 12 pt,  
Font color: Black, Not Hidden

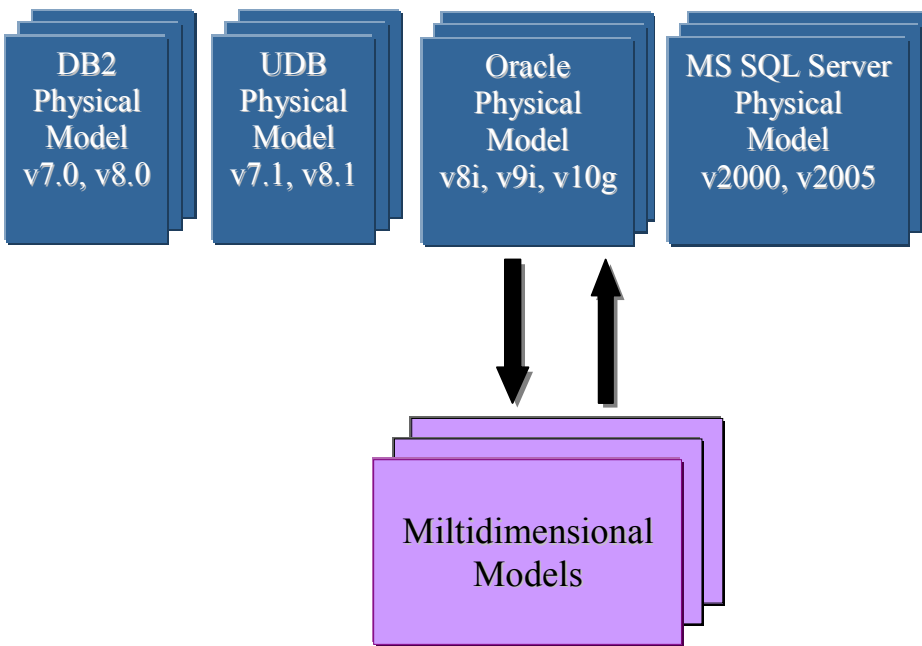
**Comment [S8]:** По-ясно за този пример – изведнъж се появяват палатки ....  
Например, ако в базата данни се съхранява информация ....

## Цикъл на моделиране

Не съществува напълно последователен път за дефиниране на добър модел на данни за даден бизнес. Моделирането на данните е сложен итеративен процес и най-добре се постига чрез метода на опита и грешката. Колкото повече детайли се дефинират за даден модел, толкова повече начини за удовлетворяване на едно и също състояние или правило могат да бъдат открити.

Няма универсален подход за последователността на реализация на моделите в рамките на модела на данни. От кой модел ще започне разработчика зависи от





Фиг 1. Модели, съставлящи Модела на данни в CWD4ALL, фирмените стандарти и избрания подход. На фиг. 1 са представени моделите, от които се състои модела на данни в CWD4ALL, както и връзките между тях. Стрелките показват възможността за автоматично преобразуване на един модел в друг, които предоставя CWD4ALL.

CWD4ALL улеснява естествения прогрес на проектирането и оставя дизайнера да се фокусира върху ключовите моменти. По време на разработването на логическия дизайн, основната задача е постигането на добре документиран нормализиран модел на данни. Визуалното представяне и репортите-справките в CWD4ALL позволяват лесно валидиране на елементите на дизайна.

Когато логическия модел е задоволително завършен, той може да бъде трансформиран до реляционен. CWD4ALL предоставя функционалност за настройване на трансформацията, включително и на съвместимостта на референсите – преобразуване на логическите ключове към индекси или

Formatted: Centered

Formatted: Russian

Formatted: Russian

Formatted: Indent: First line: 0 cm

Formatted: Centered

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm



констрейнти на таблици. И разбира се, CWD4ALL преобразува връзките много-към-много като автоматично генерира съответните таблици.

След като логическия дизайн е завършен и трансформиран до реляционен, дизайнерът има свободата да се насочи към подобряване на бързодействието по време на разработването на реляционния модел. CWD4ALL помага при формулирането и извършването на денормализацията, индексването и стратегията за физическо съхранение.

Многомерен модел може да бъде получен чрез преобразуване (reverse engineering) на физически модел за Oracle. Овен това, има съответствие между обектите на многомерния и логическия модел.

Докато диаграма на взаимовръзките между елементите (entity-relationship diagram) обикновено се асоциира с нормализирани модели, при многомерния дизайн не е така (фиг. 2). Многомерното моделиране се характеризира не с търсене на атомарни единици информация (като ентитета и атрибути) и връзките между тях, а с откриване коя информация спада към централна таблица (fact table) и коя информация спада към нейните асоциирани таблици на измеренията (dimension tables).

OLTP		Data Warehouse
Complex data structures (3NF databases)		Multidimensional data structures
Few	Indexes	Many
Many	Joins	Some
Normalized DBMS	Duplicated Data	Denormalized DBMS
Rare	Derived Data and Aggregates	Common

Фиг. 2 Реляционна и многомерна структура на данните

**Comment [S9]:** Еквивалент на български – “диаграма на взаимовръзките между елементите”

**Comment [S10]:** Всички такива английски термини до края на изложението да имат и български еквивалент

**Comment [S11]:** За всичките фигури – номерация, заглавие, цитиране в текста. По възможност фигурите да са с български текст вътре

**Formatted:** Centered

## Логически дизайн

Разработването често започва още преди критичните за дизайна **неща** **изисквания** да бъдат установени и посочени. Тези обстоятелства могат да доведат до бъгове, пренаписване и в някои случаи провалени проекти. Затова е важно **логическия** дизайн да се изчисти като се следват **общоприети практики – документиране и нормализация**.

**Comment [S12]:** Какви неща?

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Comment [S13]:** Какви?  
Нещо по-конкретно

## Документиране на Логическия дизайн

**Ако некате** **За да бъдат** идеите **на логическия дизайн ви да бъдат** разбираеми **за другите**, е необходимо **се да инвестирате** **отдели** време в документиране на значението на всяко ентити, връзка и атрибут. Добре документирания модел на данни е по-лесен за поддръжка и вероятно ще се ползва по-добре в компанията.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

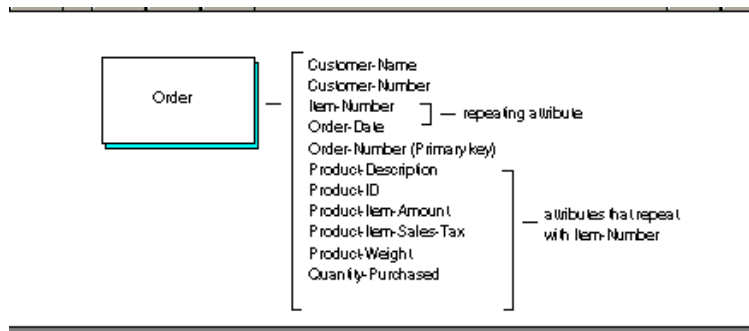
**Comment [S14]:** Не е добре да има такива форми на изказване  
Стила е безличен – за да бъдат разбираеми идеите ..... да се отдели време .....

## Нормализация

Нормализацията представлява процес на идентифициране и отстраняване на повторенията и зависимостите между ентитита и атрибути. Всяко ниво на достигнато опростяване се нарича *нормална форма*. Моделът на данни се нормализира, за да се осигури целостта на данните и, впоследствие, целостта на данните в базата, след като модела се подложи на forward engineeringed.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

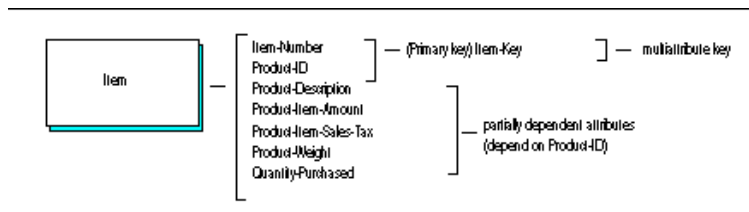
**Първа нормална форма** Дадено ентити е в *Първа нормална форма* ако няма *повтарящи се атрибути*, или атрибути, които могат да имат различни стойности за едно срещане на ентитито. Например, ентитито Order в **в следната** **фиг. 3.9а** не е в Първа нормална форма, тъй като атрибутите Product-Description, Product-ID, Product-Item-Amount, Product-Item-Sales-Tax, Product-Weight и Quantity-Purchased могат да имат различни стойности за всяка стойност на Item-Number, което се повтаря и Order-Number (ключа на ентитито).



**Фиг. 3 Ненормализирано ентити Order**

За нормализиране на ентити Order до Първа нормална форма, е необходимо да се създаде ново ентити Item, и да се преместят повтарящите се атрибути в него. Item-Number става първичен ключ в новото ентити.

**Втора нормална форма** Дадено ентити е във *Втора нормална форма*, ако всичките му неключови атрибути зависят от всички компонентите на ключа. Неключов атрибут, който зависи само от част от ключа има *частична ключова зависимост (partial-key dependency)*. Например, ентити Item във *долната* [фиг. 4](#) *тура* не е във Втора нормална форма, тъй като атрибутите Product-Description, Product-Item-Amount, Product-Item-Sales-Tax, и Product-Weight зависят само от компонентата Product-ID на ключа.



**Фиг. 4 Ентити Order в Първа нормална форма**

За нормализиране на ентити, атрибута Product-ID трябва да се премести в ново ентити, като стане първичен ключ на новото ентити, и да се преместят зависимите атрибути в новото ентити.

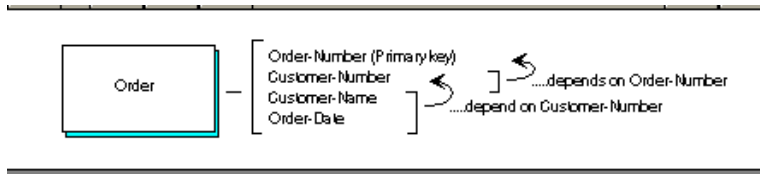
**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Трета нормална форма** Дадено ентити е в *Трета нормална форма*, ако никой не ключов атрибут не зависи от никой друг не ключов атрибут. Неключов атрибут, който зависи от друг не ключов атрибут има *транзитивна зависимост*. Например, ентити Order **навъв долната**-фиг. 5ура не е в Трета нормална форма, тъй като атрибутите Customer-Name и Order-Date зависят от атрибута Customer-Number, който от своя страна зависи директно ключа на Order, Order-Number.



**Фиг. 5 Ентити Order във Втора нормална форма**

За нормализиране на ентитито, атрибута Customer-Number трябва да се премести в ново ентити, като стане първичен ключ на новото ентити, и да се преместят зависимите атрибути в новото ентити.

### Релационен дизайн

По време на разработването на релационния дизайн се набляга на бързодействието и преобразуването на **нешата данните** за определена платформа. Нормализираният дизайн често води до проблеми в бързодействието, обикновено с големи структури от данни и многобройни конкурентни клиенти. Това е резултат от сложни заявки с множество join-клаузи. Затова релационния дизайн се занимава със следните въпроси:

- Трансформация на Логически към Релационен модел – как елементите на релационния модел съответстват на елементите на логическия
- Денормализация – как да се модифицира логическия модел, за да се повиши ефективността на операциите в базата
- Индексиране – Как да се създадат индекси, за да се повиши ефективността на операциите за четене (заявките) без да се жертва ефективността на операциите за четене (insert, update, delete)

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Comment [S15]:** Думата неща не трябва да е в текста – да се замени с конкретни думи

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

## Трансформация на Логически към Релационен модел

Когато CWD4ALL генерира релационен дизайн от логически, той трансформира обектите на логическия дизайн до техните аналози от релационния.

Долната Таблица 1 показва съответствието между логическите и релационните обекти.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Обект на Логическия модел	Обект на Релационния модел	Забележка
Entity	Table	Ентититите се преобразува директно до таблица. Ако то е част от йерархия, се прилага избраната forward стратегия В многомерния модел таблиците са автоматично определени типове (fact, dimension, snowflake или undefined).
Attribute	Column	Атрибутът се преобразува директно до колона на таблица.
View	View	View-то се преобразува директно между логическия и релационния модел. След това при физическия някои характеристики могат

		да му бъдат добавени или тялото изменено според спецификата на базата.
Relationship	Foreign Key, <u>Table</u>	<u>Връзките с cardinality много-към-едно се преобразуват във чужд ключ. Връзките с cardinality много-към-много се преобразуват в трета таблица с индекс. Връзките едно-към-едно не се преобразуват.?</u>
Primary Key	Primary Key	Primary Key се преобразува директно до Primary Key в релационния модел.
Alternate Key	Unique Index	
Inversion Entry Key	Non-unique Index	

Таблица 1 Съответствие между логически и релационни обекти

## Денормализация

Денормализацията е неразделна част от моделирането на бази от данни. Независимо колко елегантен е логическия дизайн на теория, той често не е ефективен на практика, заради сложните заявки, които той изисква. В действителност, денормализацията е може би най-важната причина за разделянето на логическия от релационния дизайн.

Ето няколко общи сценарии, които трябва да бъдат разгледани при денормализирането на релационен дизайн.

Formatted: English (U.S.)

Formatted: English (U.S.)

Comment [S16]: За таблиците важи същото като за фигурите

Formatted: Centered

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

### ***Дублирани неключови колони***

Възможно е дублирането на колони, които не участват в ключ в други таблици с цел да елиминирате нуждата от достъп (join) до таблицата в заявка. В този случай задължително трябва процедурната логика да синхронизира данните в колоната, така че тя не се различава от стойностите на реферираната таблица.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

### ***Хоризонтално разцепване на таблици***

Когато таблиците имат много редове с данни, е възможно разделянето на данните чрез разцепване на таблицата на няколко таблици. Всяка таблица ползва една и съща схема, но съхранява различна поредица от стойности на първичния ключ. Това позволява да се намали размера и гъстотата на индексите, ползвани за извличане на данните, оттам и да се подобри тяхната ефективност. От друга страна, достъпа до данните в няколко таблици вместо в една, води до използването на по-сложна процедурна логика при управление на таблиците.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

### ***Вертикално разцепване на таблици***

Когато таблици с много колони са обект на множество операции за четене и писане, е удачно да се разцепи таблицата на няколко таблици с различни множества от колони; и всяко множество от колони да ползва един и същ първичен ключ. Това води до разпространение на update операциите сред много таблици и така намаляване на поддръжката на много индекси. Както при хоризонталното разцепване на таблици, вертикалното разцепване води до необходимостта от ползване на по-сложна процедурна логика при управление на таблиците.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

### ***Индексиране***

Една от целите на индексите е подобряване на бързодействието чрез осигуряване на по-резултатен механизъм за намиране на данните. Индексите работят като каталог в библиотека: вместо да се търси измежду книгите на рафта се търси указателя към книгата в каталога, който сочи към местоположението на книгата. Логическите индекси пазят указатели към данните, така не се налага търсене в самите данни.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Индексите са един от най-важните механизми за повишаване на бързодействието на заявките. Въпреки това, безразборното използване на индекси може да се отрази негативно на бързодействието. Затова трябва да се намери оптималния брой, тип и място на индексите, така че да се максимизира ефективността на заявките.

Въпреки, че индексите могат да подобрят действието на заявките за четене, те забавят операциите за писане (insert, update и delete). Това е така, тъй като самите индекси се модифицират при всяко модифициране на данните. Затова ако дадена таблица е обект на множество операции за писане, броят на индексите в нея трябва да се ограничи. Обратно – ако таблицата е основно статична, то големия брой индекси в нея няма да навреди на бързодействието.

## Физически дизайн

### Разширяване на Релационния модел до Физически

CWD4ALL поддържа преобразуване на един релационен до няколко физически дизайна. Целта е да се разшири релационния модел според потребностите на конкретната база, като се добавят специфични обекти и характеристики. Затова физическия дизайн се занимава със следните въпроси:

- Индексиране – Как да се създадат индекси, за да се повиши ефективността на операциите за четене (заявките) без да се жертва ефективността на операциите за четене (insert, update, delete)
- Физическо съхранение – как да се определи физическото съхранение на таблиците и индексите в базата данни, за да се максимизира ефективността и да позволява разрастване

## Индексиране

### Тип на индекс

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm



Общо взето, има два типа заявки: *–конкретни заявки*, които връщат малко множество от данни, и *не дотам конкретни заявки*, които връщат голямо множество от данни. Клъстеризираните индекси са подходящи при изпълнение на по-общите заявки, докато не-клъстеризираните – на конкретните заявки.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

### **Местоположение на индекс**

Повечето SQL бази от данни предоставят механизъм за управление на физическо съхранение на таблиците и индексите. Добре е да се разделят местата на таблиците и индексите, така че базата да може да чете паралелно индексите и таблиците, и така да се подобри бързодействието.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

### **Физическо съхранение**

Планирането на съхранението на таблиците и индексите се прави най-добре по време на разработването на физическия дизайн с цел да се повиши ефективността и да се наблегне на задачите за администриране на данните.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Бързодействието на приложенията с бази данни зависи най-вече от входно-изходните операции. За повишаване на входно-изходния капацитет, таблиците и индексите, които се ползват с тях се разделят физически. Освен това се разделят и таблиците от техните индекси. Целта е да се извършва паралелно четене и писане колкото в възможно повече.

~~Два важни~~ За всеки администратор на бази данни са важни въпросите за управлението на свободната памет и за фрагментирането на данни. Ако не са правилно предвидени обема и разрастването на таблиците и индексите, тези две неща определено могат да повлияят на достъпа до системата и бързодействието. Следователно, при моделирането на физическия дизайн трябва да се вземе пред вид първоначалния размер на пространството и логическото му разделяне.

**Comment [S17]:** На български словоредът е обратен

Като начало се определя размера на всяка таблица и нейните индекси, на базата на броя на редовете ѝ. За бази, за които се определя начален размер на

пространството (extent), като Oracle, трябва да се зададе очакван размер, за да се избегне фрагментиране на данните когато таблицата се разрасне. Чрез разполагане на таблиците в отделен extent, може да се намали времето за достъп до данните и да се избегне реорганизацията на таблиците.

След като са определени места и размери за таблиците и индексите в базата, може да се пресметне общия размер на базата. Това ще помогне да не се стигне до липса на свободно пространство за базата.

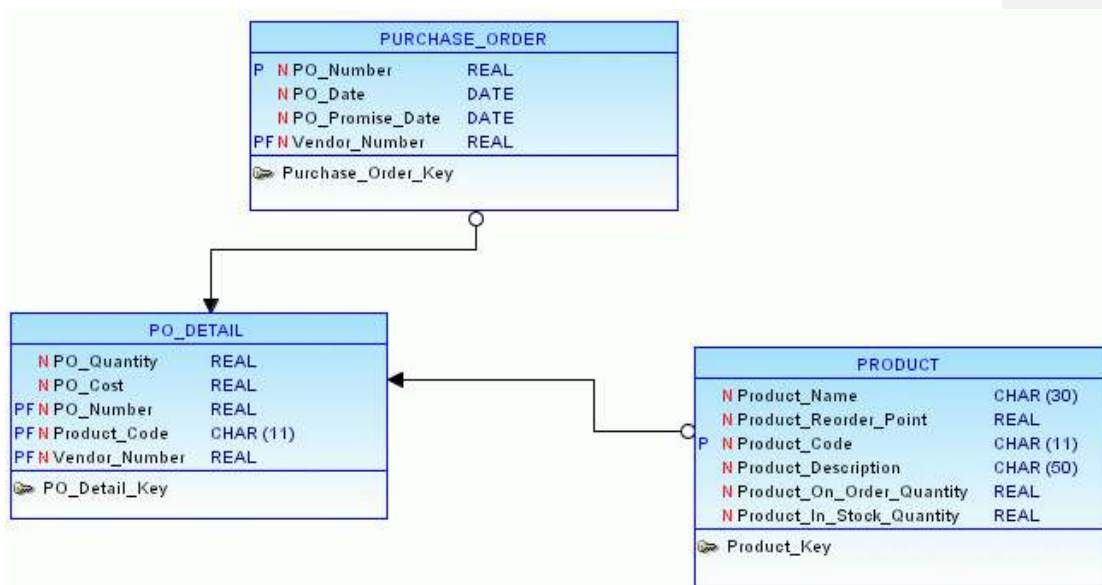
## **Обекти на Логическия модел**

### **Ентити**

*Ентити* е обект от логическия модел, който представя човек, място, предмет, понятие или действие. То е основната градивна единица информация в логическия модел. Всяко ентити има уникално в рамките на логическия дизайн име и се описва с факти, наречени атрибути.

Бизнесът се моделира чрез връзките, които съществуват между ентититата. Например, в вв ~~долната~~ фигура 6 ентититото PO\_Detail в Handy Hardware има връзки с ентититата Product и Purchase\_Order.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm



**Фиг. 6** Ентитата PURCHASE\_ORDER, PO-DETAIL и PRODUCT и връзките между тях

- Formatted: Bulgarian
- Formatted: Centered
- Formatted: Bulgarian
- Formatted: Bulgarian
- Formatted: Bulgarian

## Атрибут

Атрибутът е парче информация за бизнеса. Когато се конструира модела на данни, се дефинират атрибути и се асоциират с ентитита. Определят се колкото атрибути са необходими, за да се опише дадено ентити.

- Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Възможно е няколко атрибута в логическия модел да бъдат с еднакви имена, но в рамките на всяко ентитита имената трябва да са уникални. Всяко срещане на атрибут има специфични стойности.

**Пример** Нека е дадено ентити Sale (в модела на Handy Hardware Company от глава 4), което представя продажба на даден клиент. Атрибутите в това ентити могат да включват Sale\_Number, Sale\_Date, Subtotal, Total\_Tax и Grand\_Total.

- Formatted: Russian
- Formatted: Russian
- Formatted: Russian

## Ключове

Ключът е именован обект от логическия модел, който уникално идентифицира едно срещане на ентити. Ключът може да бъде съставен от един или повече атрибути. CWD4ALL позволява да се дефинират един първичен и един или няколко кандидат ключа за всяко ентити в модела.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Primary key** **Първичен ключ**

Първичен ключ (*primary key*) е ключ, който се използва за идентифициране на срещане на ентити.

Formatted: Font: Italic, English (U.S.)

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Font: Italic

Formatted: Font: Italic, Russian

Formatted: Font: Italic, English (U.S.)

Formatted: Font: Italic

**Candidate key** **Кандидат-ключ**

Ключове, които не са първични се наричат кандидат-ключове (*candidate key*), тъй като са кандидати да станат първичен ключ.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

**Пример** Срещанията на ентити Customer от фиг. 7 могат да бъдат идентифицирани като се използва ключ, съставен от атрибута Customer ID или ключ, съставен от атрибутите Customer Name и Customer Address. Ако единия ключ се избере за първичен ключ, другия автоматично става кандидат-ключ. В този случай е по-удачно да се използва за първичен ключ този, съставен от атрибута Customer ID, тъй като той достъпва информацията за клиента по най-сбит начин.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

CUSTOMER	
N Customer_Name	CHAR (30)
N Customer_Address	CHAR (25)
N Customer_City	CHAR (25)
N Customer_State	CHAR (2)
N Customer_Zip	REAL
N Customer_Phone	REAL
N Customer_Credit_Limit	REAL
N Customer_Last_Purchase_Date	DATE
P N Customer_Number	REAL
Customer-Key	

**Фиг. 7 Ентити CUSTOMER**

Тъй като ключа уникално идентифицира едно срещане на ентити, той не може да включва атрибути с нулеви стойности. CWD4ALL запазва съвместимостта на

връзките в модела като предотвратява създаването на ключове в компоненти, неотговарящи на тези ограничения.

Първичните ключове, подложени на forward engineering се преобразуват в първични ключове в релационния модел. Кандидат-ключовете нямат аналози в релационния модел.

### **Чужд ключ Foreign Key**

Чужд ключ (Foreign key) е първичен ключ в едно ентити, който е рефриран в друго ентити. Foreign key Чуждият ключ е рефрираща информация, а не обект на модела. Той не може да бъде създаден или манипулиран самостоятелно.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

При всяко създаване на primary първичен или candidate key кандидат-ключ, който включва атрибути от връзка, CWD4ALL проверява дали има foreign чужд ключ key, асоцииран с тази връзка.

- Ако има foreign key чужд ключ, CWD4ALL
  - Включва атрибутите на foreign key чуждия ключ в primary първичния или candidate key кандидат-ключа на ентитито
  - Маркира атрибутите на foreign key чуждия ключ като РФК първичен чужд ключ или CFK кандидат чужд ключ
- Ако няма чужд ключ foreign key, CWD4ALL завършва операциите по създаване на primary key първичен ключ за партньорското ентити.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 2 + Aligned at: 2,86 cm + Tab after: 3,56 cm + Indent at: 3,35 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

### **Връзка**

Връзката между две или повече ентитита в модела на данни описва как те взаимодействат помежду си.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Възможно е да се укаже максимален брой на срещанията на ентитита, в който може да участва дадена връзка. Този номер се нарича *максимално cardinality* на връзката. Срещанията на ентитита могат да участват във връзки в следните отношения:

**Formatted:** Russian

- Одно-към-едно (1:1) – на всяко срещане на ентити А, може да има най-много едно срещане на ентити В; и на всяко срещане на ентити В може да има най-много едно срещане на ентити А.
- Одно-към-много (1:M) или много-към-едно (M:1) ~~—че—~~ на всяко срещане на ентити А, може да има максимум много (повече от едно) срещания на ентити В; за всяко срещане на ентити В, може да има най-много едно срещане на ентити А.
- Много-към-много (M:M) ~~—че—~~ на всяко срещане на ентити А, може да има много срещания на ентити В; и за всяко срещане на ентити В, може да има много срещания на ентити А.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

### **SubviewГрупа**

Група (*Subview*) е обект от модела на данни, който логически групира ентитита. Чрез него се постига разделяне на логическия модел измежду много анализатори и по-добро документиране.

**Formatted:** Font: Italic

**Formatted:** Font: Italic, English (U.S.)

**Formatted:** Font: Italic

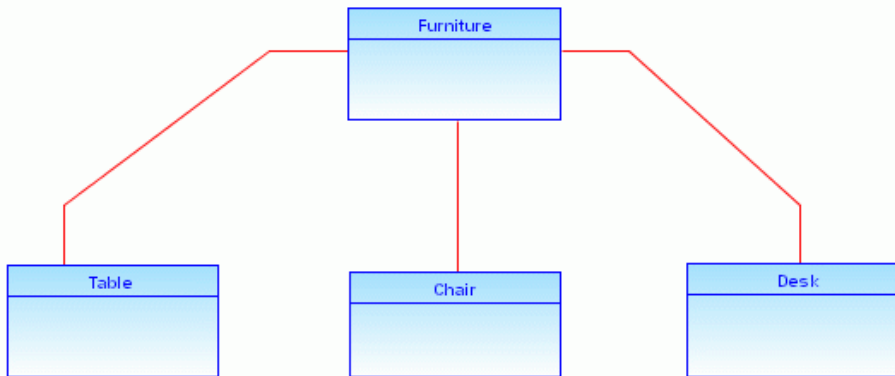
**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

### **InheritanceНаследяване**

*Йерархия на типове ентитита* е обект на модела на данни, който описва йерархична структура, в която дадено супер тип ентити е свързано с едно или няколко подтип ентитита (ентитита, които представляват различни специализирани нива на супер тип ентитито). Супер тип ентитито притежава всички атрибути и връзки на различните подтипове. Подтиповете притежават атрибутите и връзките, които ги правят уникални.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Възможно е да се създаде отделна връзка между подтип ентитита или между подтип ентитита и други ентитита. Например, нека е дадена ~~долната~~ йерархия та, показана на фиг. 8, –с ентити Furniture като супер тип и Table, Chair и Desk ентитита като подтипове.



**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

Фиг. 8 Йерархия на ентитета Furniture, Table, Chair и Desk

Тази йерархия описва връзката между подтип ентитета и супер типа, но не казва нищо за връзки, които могат да съществуват между подтиповете, като следните:

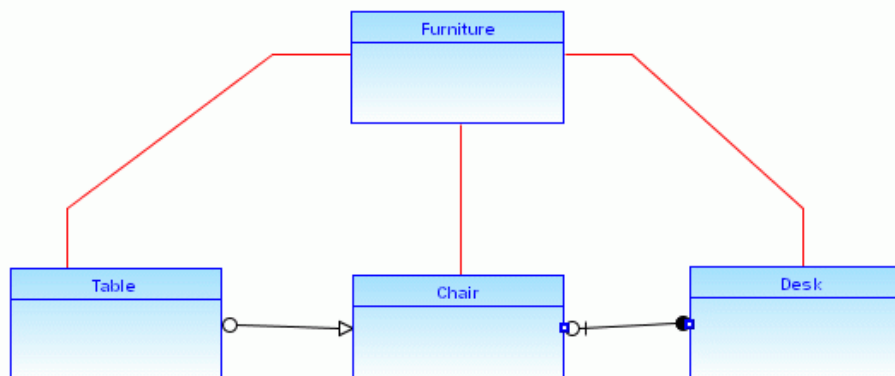
**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

- Всяко бюро трябва да има стол.
- Масите идват с нула или повече стола.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Долната фигура показва как може да се моделира тази ситуация в рамките на същата йерархия на типове ентитета.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm



**Formatted:** Centered

Фиг. 9 Връзки в йерархията на ентитета Furniture, Table, Chair и Desk

## ViewИзглед

Изглед (View) е структура от данни, получена като резултат от операции върху една или повече таблици. За улеснение CWD4ALL позволява view-изглед да се дефинира по два начина:

- чрез рефериране на атрибути на ентитета
- чрез дефиниране на SQL заявка, в резултат от изпълнението на която би се получили това view този изглед

## Обекти на Релационния модел

### Таблица

Таблицата е структура от данни, която съдържа данните в релационна база. Таблицата се състои от редове и колони.

Таблицата може да представя едно ентити от системата. Този тип таблица може да представя списък с продукти например, както е показано на фиг. 10.

PRODUCT		
N	Product_Name	CHAR (30)
N	Product_Reorder_Point	REAL
P	Product_Code	CHAR (11)
N	Product_Description	CHAR (50)
N	Product_On_Order_Quantity	REAL
N	Product_In_Stock_Quantity	REAL
	Product_Index	

**Фиг. 10** Таблица PRODUCT

Таблицата може освен това да представя връзка между две ентитета. Този тип таблици изпълнява ролята на връзка например между продуктите и техните доставки като таблицата PRODUCT\_DELIVERY. В рамките на таблиците тези връзки се представят с foreign keys.

Formatted: Centered

Formatted: Bulgarian

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Left

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

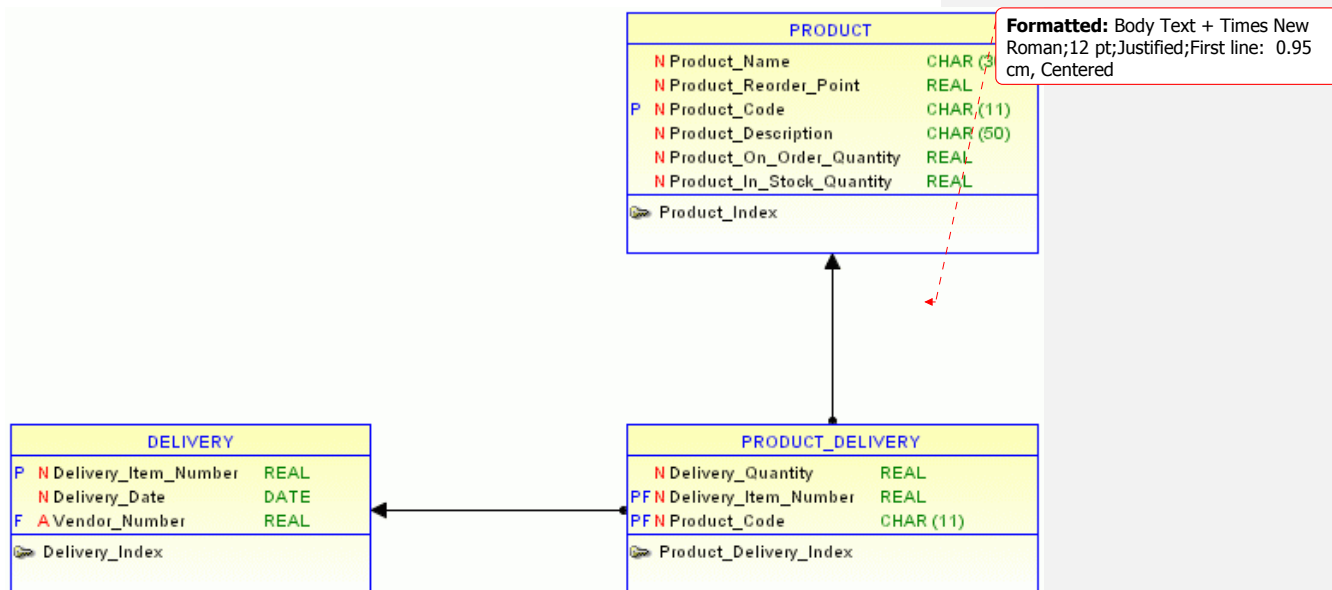
Formatted: English (U.S.)

Formatted: Russian

Formatted: English (U.S.)

Formatted: Russian





**Фиг. 11 Таблицата PRODUCT\_DELIVERY като връзка между две таблици**

Въпреки, че някои добре проектирани таблици могат да представят и ентити, и да описват връзка между това ентити и друго ентити, повечето таблици трябва да представят само едното.

▲ Следните правила се прилагат при проектирането на таблици:

- Използват се описателни имена за таблици, колони и индекси
- Използва се единна абревиатура и единствен или множествени форми на имената на таблици и колони
- Документира се значението на всяка таблица в полето Notes
- Всяка таблица се нормализира
- Избират се подходящи типове данни за всяка колона
- Дефинират се колони, които позволяват нулеви стойности най-накрая, за да се запази свободното пространство за съхранение

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

## Колона

Колоната дефинира какви данни ще съдържа таблицата. Няма ограничение на броя на колоните в дадена таблица, но е препоръчително да има поне една. Дефинират се колкото колони са необходими, за да се опишат данните в таблицата.

Възможно е няколко колони в реляционния модел да бъдат с еднакви имена, но в рамките на всяка таблица имената трябва да са уникални.

## Keys — Primary key, Foreign key Ключове

За колоните могат да бъдат дефинирани *integrity constraints*, с цел да се приложат бизнес правила на върху данните в таблиците. Бизнес правилата установяват закономерности и връзки, които винаги трябва да са изпълнени или винаги не трябва да са изпълнени.

Когато integrity constraint се приложи за дадена таблица, всички данни в таблицата трябва да следват съответното правило. При задаване на SQL заявка, която модифицира таблицата, базата автоматично проверява дали новите данни удовлетворяват зададения integrity constraint.

## Първичен ключ primary key

Първичен ключ (*primary key*) е ключ, който се използва за идентифициране на ред от таблицата. Стойността му е уникална в рамките на таблицата и не може да бъде null. Всяка таблица може да има най-много един първичен ключ.

## Уникален ключ Unique key

Уникален ключ (*Unique key*) е ключ, чиято стойност в таблицата е уникална. За разлика от първичния ключ, той не е идентификатор на таблицата, стойността му не може да бъде null. Всяка таблица може да има неограничен брой *unique keys* уникални ключове.

## Чужд ключ Foreign Key

Чужд ключ (*Foreign key*) е първичен ключ в дадена таблица, който е рефериран в друга таблица.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Comment [S18]: Еквиваленти и на български  
Първичен ключ  
Уникален ключ  
Чужд/друг/.....

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: English (U.S.)

Formatted: Russian

Formatted: English (U.S.)

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

## Индекс

Индексът е обект на релационния модел, които се дефинира върху една или няколко колони, които са най-често достъпвани. Индексирането им помага за бързодействие при заявките за четене и писане.

Всички ~~unique~~ уникални и ~~primary keys~~ първични ключове изискват ~~require~~ съответстващи индекси. Следните правила се прилагат при създаването на индекси:

- Уникалните и първични ключове могат да ползват както неуникални, така и уникални индекси
- Най-много един уникален или първичен ключ може да ползва всеки неуникален индекс
- Не е задължително реда на колоните в индекса и constraint-а да съвпадат

- ~~Unique и primary keys могат да ползват както не уникални, така и уникални индекси:~~
- ~~Най-много един unique или primary key може да ползва всеки не уникален индекс:~~
- ~~Не е задължително реда на колоните в индекса и constraint-а да съвпадат.~~

Състояние на индекса:

- Primary Constraint (Primary Key)
- Unique Constraint (Unique Key)
- Plain Index
- Unique Plain Index

## Изглед View

Изглед (View) е логическо представяне на друга таблица или комбинация от таблици. View-то извлича данните си от таблиците, върху които е базирано. Тези таблици се наричат *базови таблици*. Базовите таблици от своя страна могат да бъдат истински таблици или също view-та.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

**Formatted:** Bullets and Numbering

**Formatted:** Bullets and Numbering

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

**Comment [S19]:** Възможен български превод е изглед

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Всички операции, изпълнени върху **view** **изглед**, всъщност въздействат върху базовите таблици на view-то. View-тата се използват почти по същия начин като таблиците – заявките са от тип select, update, insert, delete.

**View тата** **Изгледите** предоставят различно представяне (като подмножества или надмножества) на данните, които се съхраняват в други таблици или **view-та** **изгледи**. **Изгледите** **View тата** са изключително мощно средство, тъй като позволяват нагаждане на представянето на данните за различни типове потребители.

### **Subview** **Група**

Аналогично на логическия модел, и релационния съдържа обект **Група** (**Subview**) - логическа група таблици и връзките между тях. Чрез него се постига разделяне на релационния модел измежду много анализатори и по-добро структуриране.

### **Обекти на Многомерния модел**

#### **Куб** **Cube**

**Кубът** (**Cube**) е обект на многомерния модел, който съхранява бизнес-измеренията. Обикновено кубът има два типа колони: такива, които съдържат числени факти (наречени **мерки** - *measurements*), и такива, които са **foreign keys** **чужди ключове** към **dimension** **измерения**.

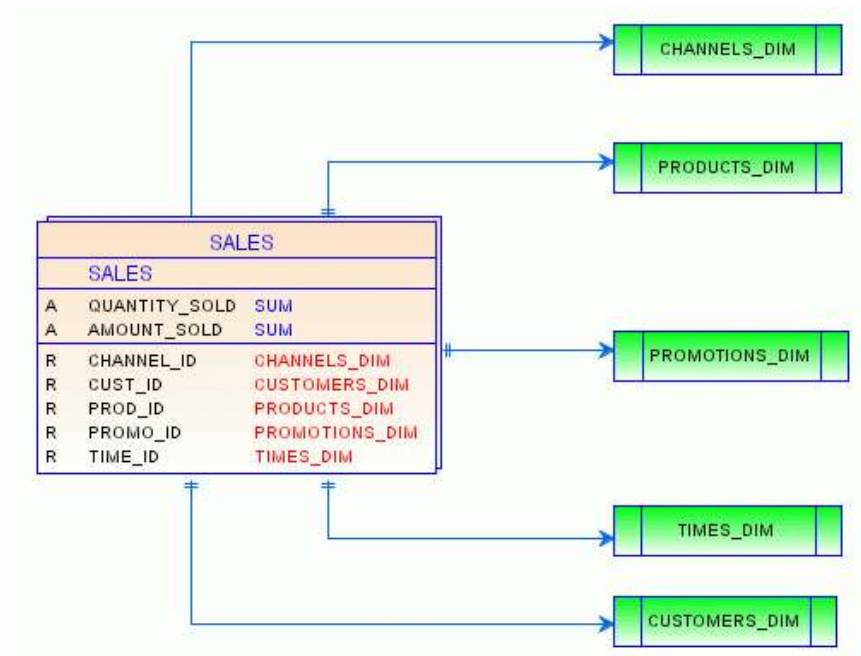
**Comment [S20]:** Така не може да остане – термин на български и в скоби view  
Вжи за трите параграфа и за следващия

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Comment [S21]:** Български ....



Фиг. 12 Куб SALES

### Мярка Measure

Мярката (*measure*) е —представява данни, обикновено числови и акумулативни, които могат да бъдат анализирани и обработвани. Обикновено, всяка мярка се категоризира с едно или повече *dimension-измерения*.

Кубът съдържа detail-level мерки или мерки, които са общи (aggregated). Кубове, които съдържат *aggregated measurements* агрегирани мерки, се представят в логическия дизайн чрез *summary* ентитета. В противен случай, кубовете се представят като *fact* ентитета.

Мерките могат да бъдат:

- Fully-additive – могат да бъдат агрегирани посредством просто аритметично събиране. Например QUANTITY\_SOLD в куба Sales.
- Semi-additive – могат да бъдат агрегирани заедно с някои dimensions, а с други – не. Например, ако това е ниво на инвентаризация, нивото не може да се определи просто като се види какво е

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

- Non-additive – не могат да бъдат агрегирани. Например, средни стойности.

## Measure Folder

*Measure folder* може да съдържа мерки от различни кубове, както и subfolders (списък от measure folders).

Measure folder няма графично представяне, а се създава от дървото на браузъра.

## DimensionИзмерение

Измерението (Dimension) е структура, често съставена от една или няколко йерархии, която категоризира данните. Атрибутите на dimension-измерението помагат да се опише стойността на-мydimension-a. Това са обикновено описателни, текстови полета. Няколко различни dimension-измерения, комбинирани с кубове, помагат в отговорите на бизнес въпроси. В горния пример на фиг. 12 dimension-измерения са CHANNELS\_DIM, CUSTOMERS\_DIM, PRODUCTS\_DIM, PROMOTIONS\_DIM и TIMES\_DIM.

Dimension-аИзмерението дефинира йерархични връзки между двойки колони или множества от колони. Тази връзка се нарича функционална зависимост на едно ниво от друго. Dimension-аИзмерението е контейнер на логически връзки между колоните и не съществува хранилище на данни, свързано с него.

При дефинирането на dimension-измерение се задават:

- Множество нива (levels), всеки от които идентифицира една или няколко колони в dimension-измерението
- Една или няколко йерархии (hierarchy), които специфицират връзки parent/child между нивата
- Незадължителни атрибути, всеки от които идентифицира допълнителна колона, асоциирана с ниво

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Колоните в dimension-измерението могат да идват или от същата таблица (денормализирана), или от множество таблици (напълно или частично нормализирани). За дефиниране на колони от много таблици се използва JOIN клауза на HIERARCHY клаузата.

## Hierarchy Йерархия

Йерархиите (hierarchy) са логически структури, които използват подредени нива в смисъла на организиране на данни. Йерархиите могат да бъдат използвани за дефиниране на агрегации на данни.

В рамките на йерархия, всяко ниво е логически свързано с нивата под и над него. Стойностите на данните на по-ниските нива се агрегират в стойностите на данните на по-високите нива. Dimension-Дадено измерение може да бъде съставено от една или няколко йерархии. Например, в dimension-измерението Product има две йерархии – една за продуктовете категории и една за доставчиците.

## Level Ниво

Нивото (level) представя позиция в дадена йерархия. Нивата в дадено dimension-измерение могат да са организирани в една или няколко йерархии. Например, на фиг. 13 dimension-измерението Product може да има йерархия, която представя данните в нива PRODUCT, SUBCATEGORY, CATEGORY и PROD\_TOTAL.

Връзките между нивата определят отгоре-надолу top-to-bottom подреждане на нивата от най-общото (корена) до най-частните данни. Тези правила дефинират parent-child връзки между нивата в йерархията.

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

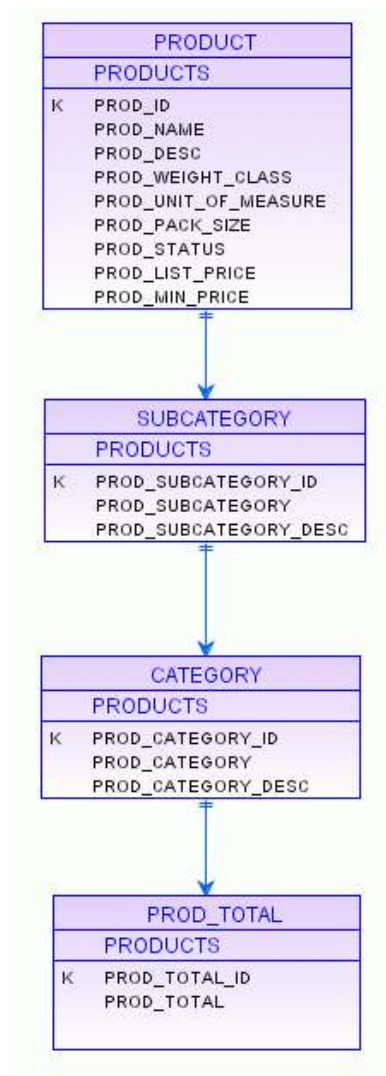
Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian



Фиг. 13 Йерархия от нива PRODUCTS, SUBCATEGORY, CATEGORY, PROD\_TOTAL

### Rollup Link

Йерархиите в dimension-ите/измеренията представляват естествени 1:n връзки между колони или групи от колони (нивата на йерархията), които не могат да бъдат представени с ограничаващи условия. Изкачването на ниво в йерархията се нарича **roll up** на данни, а спускането на ниво в йерархията - **drill dow**.

- Formatted: Russian
- Formatted: Russian
- Formatted: Russian
- Formatted: Russian
- Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm
- Formatted: Russian
- Formatted: Russian
- Formatted: Russian
- Formatted: Russian
- Formatted: Russian
- Formatted: Russian
- Formatted: Russian
- Formatted: Russian
- Formatted: Russian



## Глава 2. **Процес Модел**

### Въведение

Процесите съществуват, за да трансформират суровите данни в информация. Процес модела идентифицира и дефинира процесите, които използват информацията, дефинирана в модела на данни. Процес моделът е концептуално представяне на функциите в дадена информационна система или бизнес. Той допълва модела на данни, който представя съдържанието, структурата и връзките на информацията, използвана от предприятието.

В CWD4ALL, Процес моделът се представя визуално от една или повече Диаграми на потока данни (Data Flow Diagrams - (DFD)). Обектите на Процес модела включват процеси и подпроцеси, събития, информационни хранилища, външни агенти и информационни потоци. Например, модела за Handy Hardware Company съдържа няколко DFD: Point of Sale Project, Returned Goods Processing Project, ~~and~~ и Store Inventory Control Project.

~~Data flow diagram (DFD)~~ Диаграмата на потока данни представя потока и обработването на информацията в даден бизнес. DFD показва цялостния системен поток на информацията между функционалните единици (процеси), които трансформират информацията, единиците за съхранение (~~information~~ информационни хранилища—stores), които държат информацията, и източниците и предназначенията на информацията, които се намират извън системата (~~external agents~~ външните агенти). В CWD4ALL, DFD още показва дали даден процес се въздейства от събития (event).

DFD дава изчерпателноа и цялостно представяне на информационната система, която се моделира. Тя представя функционална ориентация на информацията като се движи през системата от перспективата на информацията. Като такава DFD е ценно средство за изпълнение на анализ на изискванията и системна спецификация. Обикновено ползвана за моделиране на информационни системи, DFD се използва също за бизнес и стратегическо планиране.

**Comment [S22]:** На български не е ли по-добре Модел на процес

**Comment [S23]:** Никъде не става дума в тази глава за CWD4ALL., дори не се споменава  
Трябва да се започне по принцип и после спецификата на CWD4ALL.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

**Formatted:** Russian

**Comment [S24]:** Диаграма на потока данни

**Comment [S25]:** ???  
не може хем женски, хем среден род

Тъй като DFD дава функционално представяне на системата, тя не представя детайлите на информацията или как информацията се обработва. Тези детайли се представят от диаграми на взаимовръзките между елементите (ERD) entity relationship diagram (ERD) на модела на данни. По този начин информацията като цялостен системен поток се моделира от DFD заедно с ERD.

### Цикъл на моделиране

Последователността от действия при моделиране зависи от подхода на моделиране и стандартите на проекта. В таблица 2 е показан примерен цикъл за моделиране, който съвсем не е задължителен.

Стъпка	Задача	Действие
1	Създаване на DFD	Създава се DFD
2	Добавят се обекти към DFD	Създаване на обекти в DFD
3	Започване на функционална декомпозиция	Създаване на нива от процеси и подпроцеси в DFD
4	Изчистване на функционалната декомпозиция	Аранжиране на позицията на процесите в DFD
5	Създаване на процес йерархия в DFD Изчистване на описанията на обектите	Попълване на текст във формите на обектите
6	Модифициране на обектите на Процес модела	Променяне на DFD

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Comment [S26]:** едно въвеждащо изречение за таблицата и оформяне на текста в таблица

**Formatted:** Font: (Default) Times New Roman

**Formatted:** Font: (Default) Times New Roman

**Formatted:** Font: (Default) Times New Roman

**Formatted:** Font: (Default) Times New Roman

**Formatted:** Font: (Default) Times New Roman

**Formatted:** Font: (Default) Times New Roman

**Formatted:** Font: (Default) Times New Roman

	Редактиране на обектите през формите им	
	Изтриване на обекти	
	Редактиране на обекти чрез трансформации	
7	Обединение на две DFD	Добавяне или комбиниране на обекти на две DFD
8	Отпечатване	Отпечатване на диаграма или репорт
9	Валидиране на Процес модела	Изпълнение на Design Rules за Процес модела

Таблица 2 Примерен цикъл на моделиране на Процес модел

## Обекти на Процес модела

### External Agents Външен агент

Външния агент (external agent) е човек, група хора или друга система, която изпраща информация на или получава информация от процесите на системата, която се дефинира. Външните агенти са източници и получатели на информация, които са извън границите на системата.

Външен агент може да се класифицира като един от четири типа:

- **Система:** Друга система в предприятието, например система за продажби или система за плащане.
- **Организационна единица:** Клон или отдел на предприятието, например, изпълнително ръководство или директор.
- **Роля:** Функционална единица в рамките на предприятието, дефинирана отделно от организационните единици или други системи, например, главен купувач или лидер на проект.

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Russian

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Russian

Formatted: Font: (Default) Times New Roman

Formatted: Centered

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

- **Друго:** Външен агент, който не е нито система, нито организационна единица, нито роля; например, търговец или правителствена регулаторна агенция.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Пример** В модела Handy Hardware Company, DFD Point of Sale Project дефинира следните типове външни агенти:

- Product Manager, Sales Person (Организационна единица)
- Store Inventory Control Project (Система)
- Customer (Роля)
- Bank, Credit Bureau (Друго)

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

### **Information Flows** Информационен поток

*Информационния поток (information flow)* е пътя, по който минава информацията (дефинирана във вид на ентитета). Потокът може да свързва агент и хранилище с процес или процес с хранилище, агент или друг процес.

**Formatted:** Russian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

Следните правила важат за потоците в CWD4ALL:

- Когато поток свързва примитивен или непримитивен (декомпозитен) процес, примитивния процес се добавя автоматично към декомпозицията DFD на непримитивния процес.
- Ако се създаде поток между хранилище и процес на най-ниското ниво на декомпозиция, същия поток автоматично се появява на по-високите нива на декомпозиция.
- Ако поток се създаде между хранилище и непримитивен процес, на най-ниското ниво се създаде поток (със същата посока) между същото хранилище и подпроцес, родителския поток може да бъде избран за новосъздадения поток. Няколко потока могат да имат един родител – това са т. нар. съставни потоци.

**Formatted:** Russian

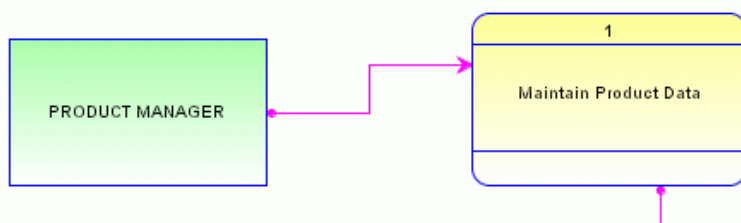
**Formatted:** Russian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

**Formatted:** Russian

CRUD (Create, Read, Update and Delete) операции могат да бъдат дефинирани за потока и примитивния процес – Read за поток от хранилище към процес, и Create, Update, Delete за поток от процес към хранилище. На практика тези операции дефинират връзката между процеса и информационната структура, дефинирани за този поток.

**Пример** На фиг. 14 като част от диаграма е показан информационен поток от външния агент PRODUCT MANAGER до процес Maintain Product Data.



Фиг. 14 Информационен поток от външния агент PRODUCT MANAGER до процес Maintain Product Data

### Information Structure Информационна структура

Информационната структура (information structure) дефинира типове ентитета и атрибути, които ще се използват от един или повече процеси. След като е създадена една информационна структура, тя може да бъде използвана неограничен брой пъти в процес модела във всяка DFD.

Структурите се създават и прикачат във формата на Информационния поток.

Дадена информационна структура може да съдържа един или повече ентитета заедно с един или повече от неговите собствени или наследените му атрибути

Следните общи правила се прилагат за информационните структури:

- Даден информационен поток може да има нула или повече асоциирани структури.
- Ентити или атрибут може да се съдържа в повече от една информационна структура.

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text

Formatted: Bulgarian

Formatted: Bulgarian

Formatted: Bulgarian

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

- Обсега на информационната структура е модела на данни, следователно може да бъде ползвана от произволен брой процеси.

## **Information Stores** Информационно хранилище

Информационното хранилище (*information store*) е склад на информация, която е дефинирана като ентитета в модела на данни. Всяко информационно хранилище има уникално име и номер.

В завършен Процес модел, всяко информационно хранилище има поне един информационен поток от примитивен процес в или от него. Веднъж дефинирано, хранилището може да се използва във всяко ниво на декомпозиране.

Информационните хранилища могат да бъдат включени в рамките на родителски процес. Така, заедно с подпроцесите, информационното хранилище става подчинен обект в йерархията, която се показва от DFD.

**Пример** В модела Handy Hardware Company, DFD Point of Sale дефинира информационно хранилище Product Information, както е показано на [фиг. 15](#).



[Фиг. 15 Информационно хранилище Product Information](#)

## **Processes** Процес

Процесът е операция в бизнеса, която променя информацията според специфицираната бизнес логика. Всеки процес в Процес модела има уникално име и номер. [На фиг. 16 е показан процесът Generate Purchase Orders.](#)

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Font: Not Bold

Formatted: Body Text

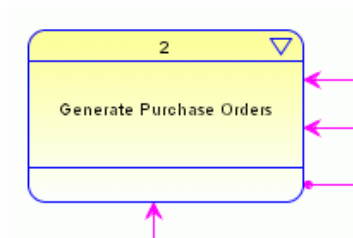
Formatted: Russian

Formatted: Font: Not Bold

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian



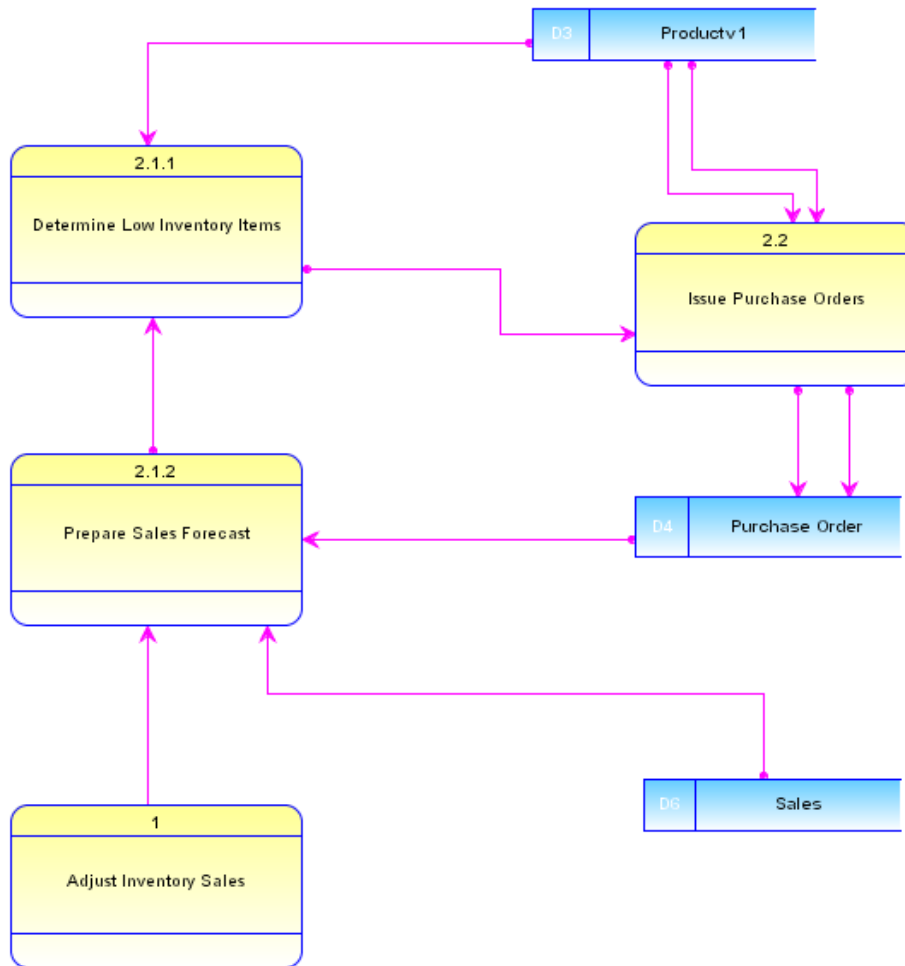
Фиг. 16 Процес Generate Purchase Orders

Процесът може да бъде разделен на два или повече по-прости *подпроцеси*. Ако е необходимо, всеки подпроцес от своя страна може да бъде разделен на друго ниво подпроцеси. Разделянето на процеси е основното при разработването на йерархия на процесите и има смисъл на определяне на процесите, за които се специфицира логика. Родителския процес може да съдържа и информационни хранилища, които са подчинени, или вътрешни, за него, както е показано на фиг. 17.

Formatted: Russian

Formatted: Russian

Formatted: Russian



**Фиг. 17 Процеси и подпроцеси**

*Примитивен процес* е процес, който не е разделен; той няма подпроцеси. Примитивният процес е най-ниското ниво в йерархията на процеси. Примитивните процеси са процесите, за които се определя логика.

Даден примитивен процес има още:

- Информационен поток, идващ в него или излизащ от него, или и двете

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm



- Не задължително, събитие, което показва причините процеса да възникне и кога процеса е започнал

Само примитивните процеси могат да имат събития и информационни структури.

Умението да се различава примитивен от непримитивен процес е важно за изпълнението на множество операции като премахването на процеса и свързването отново на информационния поток.

Когато процесът е съставен, той има своя диаграма (собствена DFD) която описва подпроцесите и потоците в него. Тази диаграма е подобна на DFD от най-горно ниво, но има и разлики. Тя показва подпроцесите, потоците и използваните от тях хранилища (те са глобални за процес модела), но показва и процеси от по-горно. И така разбивката на процеса на подпроцеси и техни диаграми може да продължава безкрай.

### Йерархия на процеси

## **Process Hierarchy**

Йерархията на процеси (*process hierarchy*) е връзката между всички процеси и подпроцеси в Процес модела.

Йерархията на процеси дава поглед “отгоре-надолу” на функционалните единици в моделираната информационна система. Това помага да се идентифицират примитивните процеси, за които да се специфицира процес логика.

**Пример** В модела Handy Hardware Company, DFD Store Inventory Control Проект дефинира процес йерархия на три нива (фиг. 18).

- Ниво 1\_ - съдържа процесите Adjust Inventory Sales, Generate Purchase Orders и Receive Goods.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

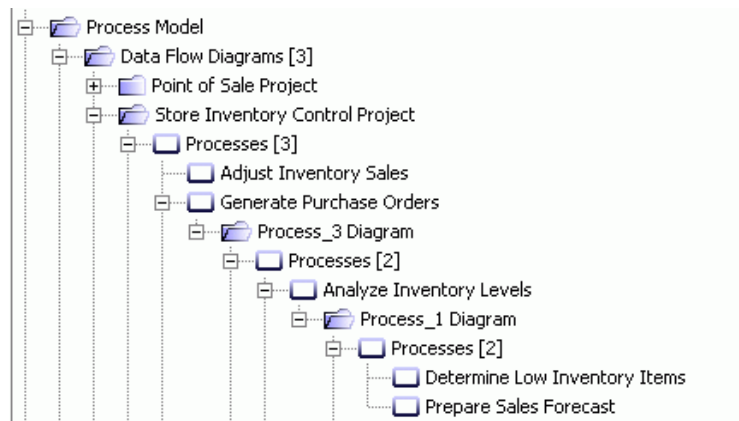
Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

- Ниво 2 – съдържа процесите Analyze Inventory Levels и Issue Purchase Orders.
- Ниво 3 – съдържа процесите Determine Low Inventory Items и Prepare Sales Forecast.



**Фиг. 18 Йерархия на процеси в 3 нива**

Йерархията на процеси може да се види от:

- DFD перспектива – Композитните процеси съдържат по една DFD за всеки процес от йерархията
- От браузъра – Това е директната перспектива, където йерархията се показва визуално в дървото

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Йерархията на процеси има следните характеристики:

- Нивата на процеси и подпроцеси са различни (Горните нива процеси са познати още като root процеси.)
- В DFD, процес йерархията се представя като кутийки в кутийки, които могат да включват и информационни хранилища, освен процеси.
- Подпроцесите могат да се появяват на всяко ниво от йерархията, с изключение на най-горното ниво, и могат да бъдат както примитивни, така и непримитивни.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

## Роля

Ролята е именовано специфично поведение на процес, участващ в специфичен контекст. Основната употреба на ролята е да дефинира съответствие между ролите и функциите (процесите).

**Comment [S27]:** да се остават само на английски думи – бълг+англ

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

## Събитие

Събитието (process triggering event) в Процес модела предизвиква примитивен процес да започне изпълнение на логиката, специфицирана за него. Събитието е само контролен обект.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Пример** В модела Handy Hardware Company, DFD Point of Sale Project съдържа примитивен процес Maintain Product Data, който се изпълнява, когато се получи нов продукт и информация за цената от продуктовия мениджър. Така получателят на информационния поток New Product and Price Changes от външния агент Product Manager задейства процедурната логика за Maintain Product Data. Процесът Maintain Product Data не се изпълнява, докато не възникне събитие за получен продукт и смяна на цена от продуктовия мениджър.

Събитията за примитивните процеси могат да бъдат четири типа:

- Събитие за информационния поток
- Събитие за часовник или календар
- Събитие за човек
- Събитие за съхранение

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Събитието за информационен поток се дефинира в потока, свързан с примитивния процес, но се появява в списъка със събития на процеса. Събитията за часовник, човек и съхранение се дефинират в примитивния процес.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

Следните правила се прилагат за събитията, независимо от типа им:

- Събитията са асоциирани с примитивни процеси

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

- Всеки напълно дефиниран примитивен процес трябва да има поне едно събитие, асоциирано с него.
- Всеки процес може да има повече от едно събития, но само едно събитие може да е дефинирано в потока.
- Събитията се дефинират за Процес модела и могат да се асоциират към повече от един примитивен процес, но събитието за информационен поток е свързано с един входен поток на примитивния процес и може да се асоциира с друг примитивен процес само, ако той има входящ поток, който свързва процеса с агент или друг примитивен процес – тогава информационното събитие се деасоциира от предишния процес и поток.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

## Глава 3. Степен на поддръжка на CWM™

### Въведение

Размерът на данните в дадена организация се удвоява на всеки 5 години. Повечето организации страдат от наличие на излишни и несъвместими данни, за които е трудно ефективно да се управляват, достъпват и използват за целите на взимане на решения.

*Data warehousing* осигурява отлична възможност за трансформиране на данни в удобна и надеждна информация за поддръжка на процес за взимане на бизнес решения и за постигане на *business intelligence*. Един от най-важните аспекти на *data warehousing* са метаданните (*metadata*). Метаданните се използват за построяване, поддръжка, управление и използване на *data warehouse*. За съжаление, разпространението на системи за управление и анализ на данни доведе до почти толкова различни представяния и манипулации на метаданните, колкото са самите системи.

Тъй като всяка система за управление и анализ на данни изисква различни метаданни и различен модел на метаданни (метамодел) за разрешаване на *data warehouse* метамодел проблема, не е възможно да има само едно хранилище на метаданни, което реализира само един метамодел в организацията. Вместо това, това, което е необходимо, е стандарт за обмен на *warehouse* метаданни.

CWM отговаря на тези нужди. Той осигурява framework-среда за представяне на метаданни като източници на данни (*data sources*), цели на данни (*data targets*), трансформации и анализи, процеси и операции, които създават и управляват *warehouse* данни и осигуряват последователна информация за нуждите на организацията.

Както е показано на фиг. 19.

Formatted: Russian

Comment [S28]: всички имена на пакети в тази глава и на български език – изключение са само XML, OLAP !!!!!!!!!

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

CWM Метамоделът се състои от под-метамодели, които представят общи warehouse метаданни в следните главни области на интерес на data warehousing и business intelligence:

- Data Resources – Включват метамодели, които представят обектно-ориентирани (object-oriented), релационни (relational), записващи (record), многомерни (multidimensional), и XML данни. В случая на обектно-ориентирани данни, CWM ползва отново базовия обектен модел.
- Data Analysis – Включват метамодели, които представят трансформации на данни, OLAP (On-line Analytical Processing), data mining, визуализация на информацията, и бизнес номенклатура.
- Warehouse Management – Включват метамодели, които представят warehouse процеси и резултати от warehouse операции.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Management	Warehouse Process			Warehouse Operation		
Analysis	Transformation		OLAP	Data Mining	Information Visualization	Business Nomenclature
Resource	Object Model	Relational	Record	Multidimensional		XML
Foundation	Business Information	Data Types	Expression	Keys and Indexes	Type Mapping	Software Deployment
	Object Model					

Formatted: Centered

### The CWM Metamodel

Management	Warehouse Process			Warehouse Operation		
Analysis	Transformation		OLAP	Data Mining	Information Visualization	Business Nomenclature
Resource	Object Model	Relational	Record	Multidimensional		XML
Foundation	Business Information	Data Types	Expression	Keys and Indexes	Type Mapping	Software Deployment
	Object Model					

Фиг. 19 CWM Метамодел

CWM Метамоделът е проектиран да увеличи максимално повторното ползване на Object Model (който е подмножество на UML) и поделянето на общи моделиращи конструкции, където е възможно. Най-известния пример е, че CWM използва/зависа от Обектния Модел за представяне на обектно-ориентирани данни. Като допълнение, където е подходящо, ключови елементи на метамоделите за други типове данни се състоят от елементи на същите модели в Обектния Модел, както е показано на долната таблица (клетките под Software System и Deployed Software System са примери.)

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

	Software System	Deployed Software System	Package	Class	Attribute
<b>Object Model</b>	Java	Java installation	Package	Class	Attribute
<b>Relational</b>	DB2 UDB, Oracle 8i, Teradata	DB2 UDB, Oracle 8i, Teradata installations	Catalog/Schema	Table	Column
<b>Record</b>	IMS, DMS II	IMS, DMS II installations	RecordFile	RecordDef	Field
<b>Multidimensional</b>	Essbase, Express	Essbase, Express installations	Schema	Dimension	Dimensioned Object
<b>XML</b>	XML4J	XML4J installation	Schema	ElementType	Attribute

Фиг. 20 Ресурси данни за CWM, Data Resources

### **Object Model package**

Обектния модел на CWM, предоставя основни конструкции за създаване и описване на класове на метамоделите на всички останали пакети на CWM. Обектния модел е подмножество на UML, което включва онези елементи, необходими за създаването и описанието на CWM.

Дефинирането на подмножество на UML, съдържащо само онези неща, необходими на CWM, позволява на CWM да използва UML концепциите и силата на моделиране без да обременява имплементациите с пълния обхват на възможностите на UML.

**Formatted:** Font: Verdana, 11 pt, Not Bold, Russian

**Formatted:** Font: Verdana, 11 pt, Not Bold

**Formatted:** Font: Verdana, 11 pt, Not Bold, Russian

**Formatted:** Font: Verdana, 11 pt, Not Bold

**Formatted:** Font: Verdana, 11 pt, Not Bold, Russian

**Formatted:** Font: Verdana, 11 pt, Not Bold

**Formatted:** Russian

**Formatted:** Russian

**Formatted:** Body Text + Times New Roman; 12 pt; Justified; First line: 0.95 cm

**Formatted:** Russian

**Formatted:** Russian

**Formatted:** Russian

**Formatted:** Russian

**Formatted:** Russian

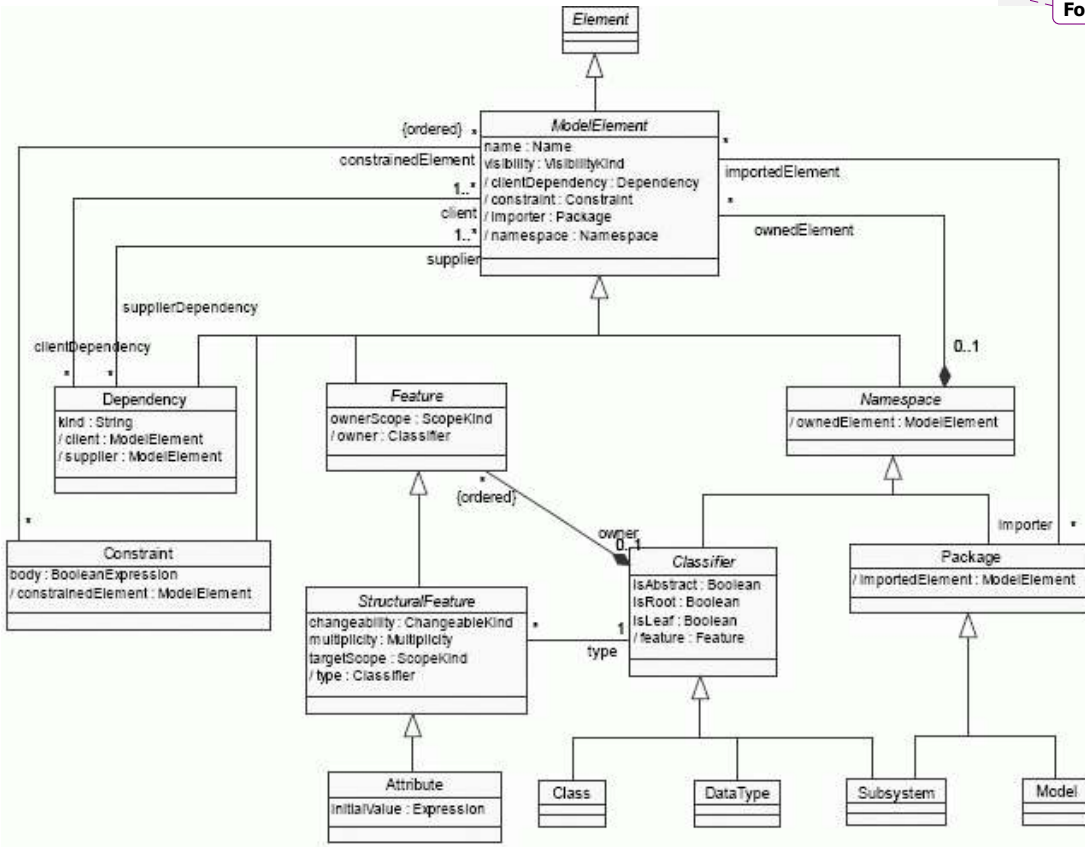


## Core package

Съдържа класове и асоциации, които формират ядрото на CWM обектния модел. Те се използват от всички останали CWM пакети, включително и от другите ObjectModel пакети.

Core metamodel не зависи от други пакети.

Класовете и асоциациите, които формират Core metamodel са показани на [дожната фигура 21](#).



Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Фиг. 21 Core  
metamodel

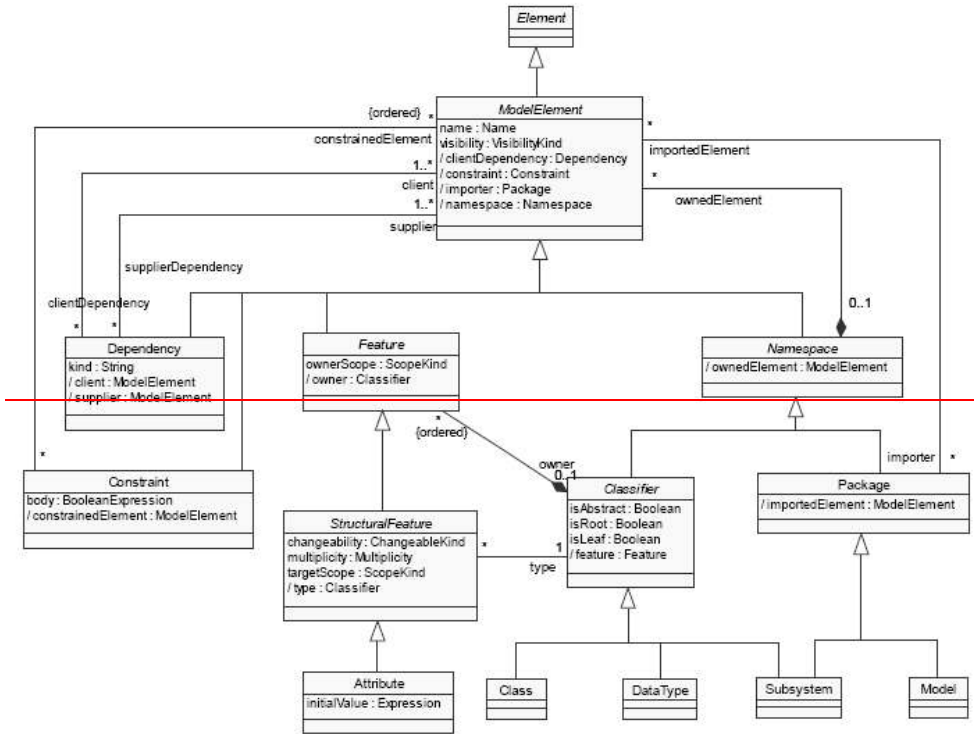


Figure 7-3-1 Core metamodel.

Core Metamodel дефинира понятия като Attribute, Feature, Class, Datatype, Constraint, Dependency, които са в основата на Логическия, Реляционния, Физическия, Многомерния, Процес Модела на CWD4ALL. Реализирано е естественото съответствие например между Attribute на Core Metamodel и Attribute на Логическия Модел или Column на Реляционния Модел.

### Behavioral package

Съдържа класове и асоциации, които описват поведението на CWM обектите и предоставя основи за описване на извикванията на дефинираните поведения.

Behavioral metamodel зависи от Core пакета на Обектния модел. Елементите му са показани на долната фигура. 22:

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

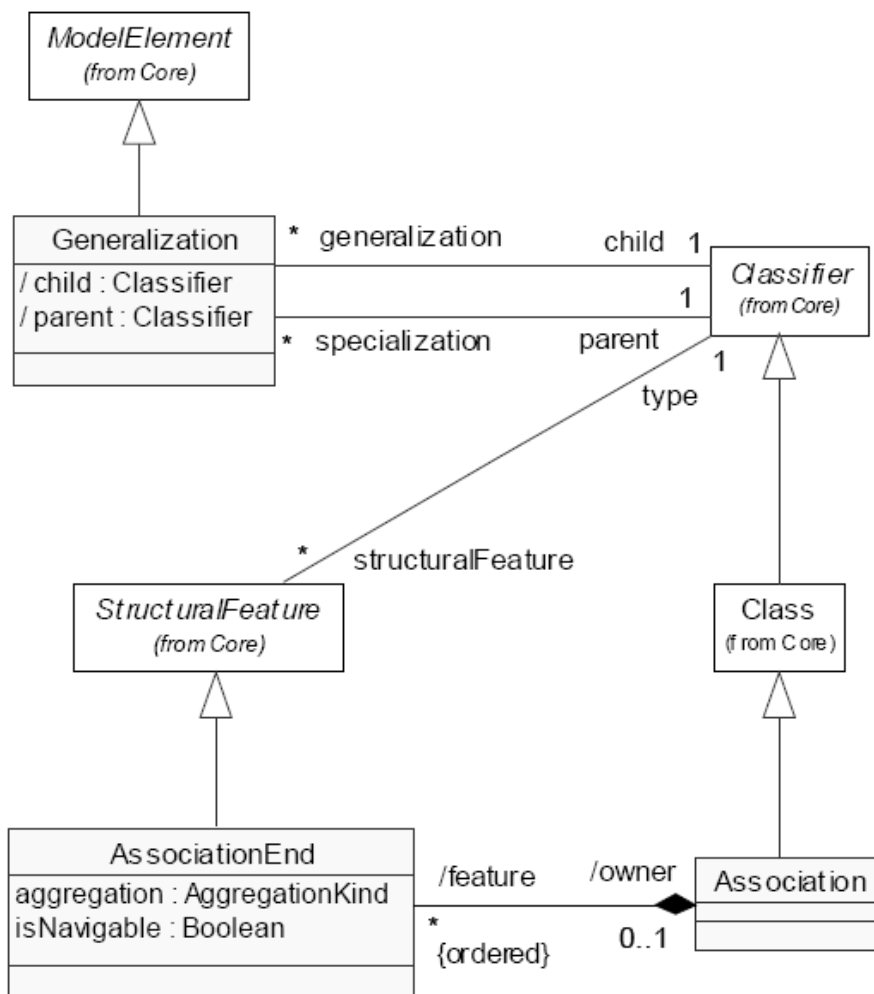
Formatted: Russian

Formatted: Russian

Formatted: Russian







**Фиг. 23 Relationships metamodel**

Отношенията на асоциация и генерализация са неразделна част от моделирането на всяка база от данни. Те правят модела на данни не просто съвкупност от ентитита или таблици, а дефинират връзките между тях.

Асоциацията в Relationships metamodel съответства на обекта Relation в Логическия модел и Foreign Key в Релационния и Физическия. За всеки се дефинира cardinality и има 2 края на асоциацията. Генерализацията определя йерархичната структура в Логическия, Многомерния, Туре и Процес Моделите.

Във всеки един е възможно дефиниране на parent и child обекти и по този начин

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Left

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

формиране на йерархия. В споменатите модели обекти подлежащи на генерализация са Entity в Логическия, Level в Многомерния, Structured Type в Type Модела, Process в Процес модела.

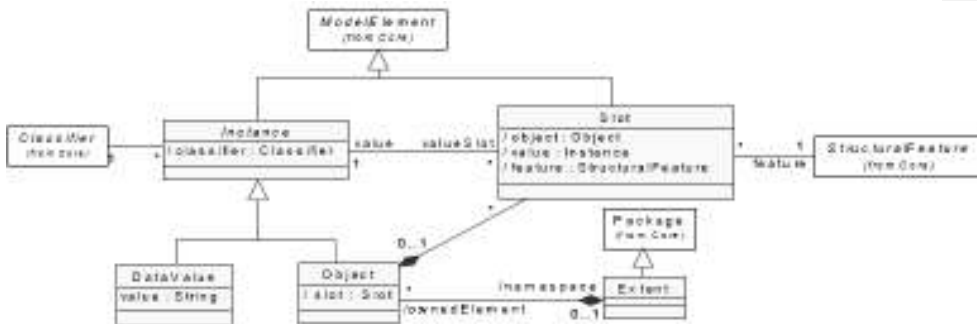
**Instance package**

Съдържа класове и асоциации, които представят инстанции на CWM класификатори.

Instance метамоделът зависи от Core package на Обектния модел.

Като допълнение към обмена на метаданни със CWM, понякога е полезно да се обменят специфични инстанции на данни. Instance метамоделът позволява включването на инстанции на данни в метаданните.

Instance метамоделът е илюстриран ~~не долуна~~ **фиг. 24:**



**Фиг. 24 Instance metamodel**

Тъй като целта на CWD4ALL е да моделира дефиницията на данните, а не тяхното съдържание (например дефинира какви данни ще съдържа една таблица, но не и самите данни), той не работи пряко с инстанции на обектите. Въпреки това, той специфицира някои характеристики и ограничения за инстанциите като **Primary** **Keupървичен ключ** на таблица в Релационния модел (който идентифицира

Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian  
Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered  
Formatted: Russian  
Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm  
Formatted: Russian

Formatted: Russian  
Formatted: Russian

уникално срещане на таблица) и Key в Логическия, Реляционния и Физически модел, който също е характеристика на инстанциите.

Formatted: Russian

## Foundation package

Foundation е множество от пакети, които представят концепции и структури, ползвани от други пакети на CWM. Следователно, елементите на модела предоставят по-обща функционалност от елементите в пакетите на по-високите нива в CWM.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Елементите на Foundation model се различават от елементите на ObjectModel, тъй като са специфични за целите на CWM. За разлика от тях, елементите на Обектния модел, са общи и приложими в различни области.

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

## Business Information package

Съдържа класове и асоциации, които представят бизнес информация за елементите на модела.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Business Information Metamodel предоставя услуги за дефиниране на бизнес-ориентирана информация за елементите на моделите.

Formatted: Russian

Formatted: Russian

Formatted: Russian

Бизнес-ориентираните услуги, описани тук са разработени за нуждите на data warehousing и business intelligence системи; те не са планирани като цялостно представяне на business intelligence metamodel.

Business Information Metamodel поддържа нотации на отговорни страни (responsible parties) и информация как се контактува с тях, идентификация на документи и поддръжка на описваща информация

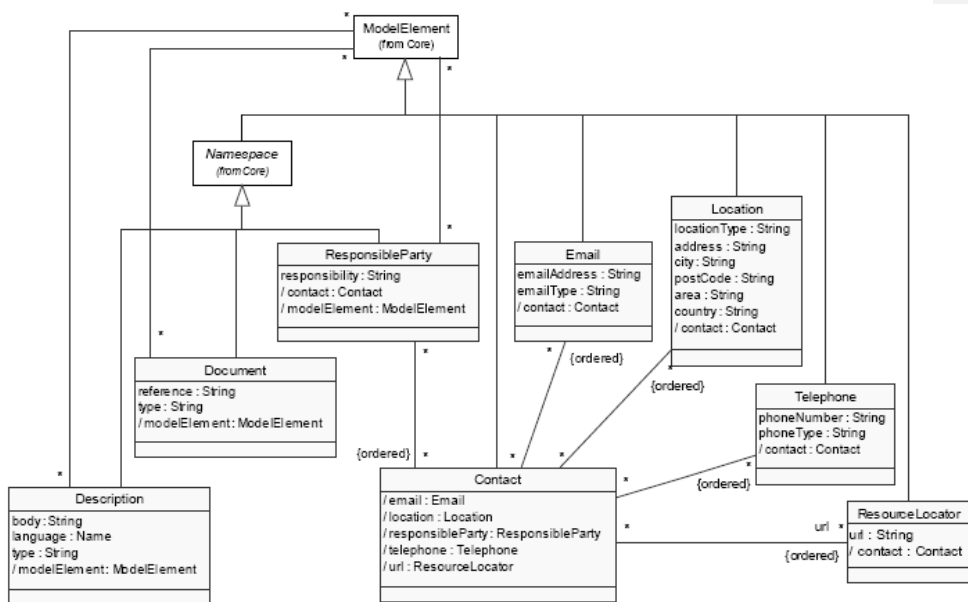
Formatted: Russian

Formatted: Russian

Formatted: Russian

Business Information метамодельт зависи от Core package на Обектния модел. Той е показан на ~~долната~~ фиг. 25ура.

Formatted: English (U.S.)



**Фиг. 25 Business Information metamodel**

Business Information package е имплементиран изцяло в Business Information модела на CWD4ALL. Поддържат се всички обекти – Document, Responsible Party, Contact, E-mail, Location, Telephone, Resource Locator - и зависимостите между тях.

### Data Types package

Съдържа класове и асоциации, които представят конструкции, които анализаторите могат да използват за да създадат необходимите им типове данни.

DataTypes package зависи от Core пакета на Обектния модел.

CWM DataTypes метамоделът поддържа дефиниция на конструкции, които могат да се ползват за създаване на типове данни. Той е показан на фиг. 26

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

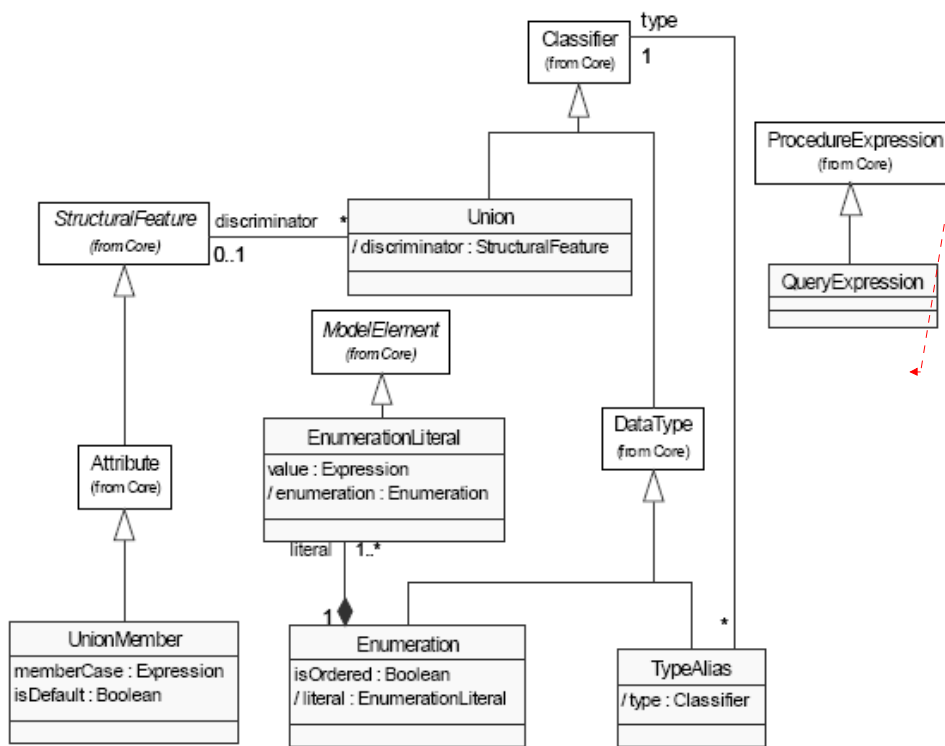
Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: English (U.S.)





**Фиг. 26 Data Types metamodel**

Data Types package е имплементиран изцяло в Data Types модела на CWD4ALL. Поддържат се Distinct Types, базирани на стандартните логически типове, Structured Types, при които е възможно дефиниране на отношения на генерализация и йерархии, и Collection Types, които представляват списъци с вече дефинирани типове.

### Expressions package

Съдържа класове и асоциации, които представят дървета от изрази.

CWM Expressions metamodel предоставя поддръжка за дефиниране на дървета от изрази (expression trees) в CWM.

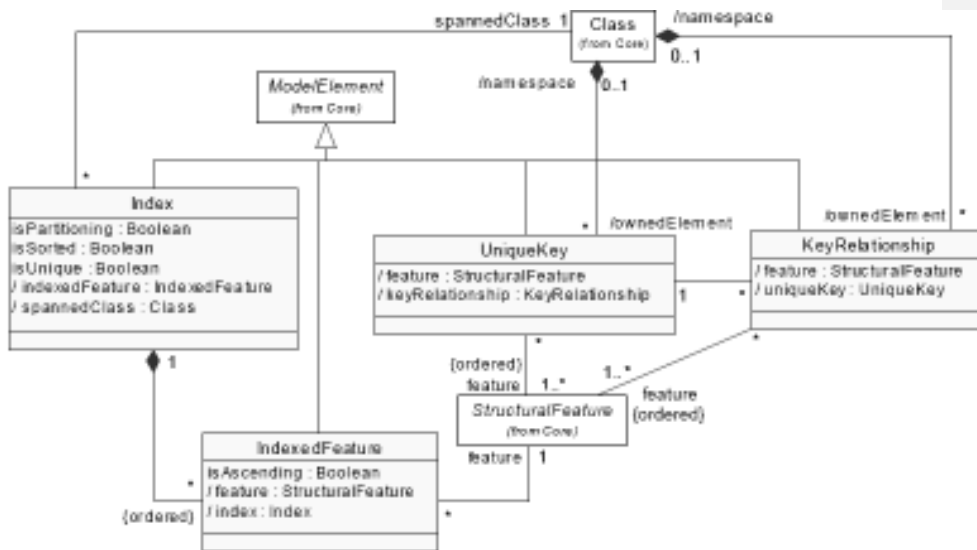
**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Left

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm





Фиг. 27 KeysIndexes metamodel

Два централни класа, представящи концепцията за ключове, са UniqueKey и KeyRelationship. UniqueKey инстанциите кореспондират с нотацията, че ключовете представляват идентичност на инстанции – подобно на Primary Key в Релационния и Физическия модел. KeyRelationship инстанциите кореспондират с нотацията, че ключове, вградени в инстанция могат да се използват за установяване на идентичност на други свързани инстанции – като Foreign Key в Релационния и Физическия модел.

### Software Deployment package

Съдържа класове и асоциации, които представят как софтуера се deploy-ва в data warehouse.

Software Deployment package съдържа класове за регистриране на начина, по който даден софтуер в data warehouse се използва.

Този пакет не е имплементиран в CWD4ALL, тъй като начина на Software Deployment не е обект на моделиране от CWD4ALL.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

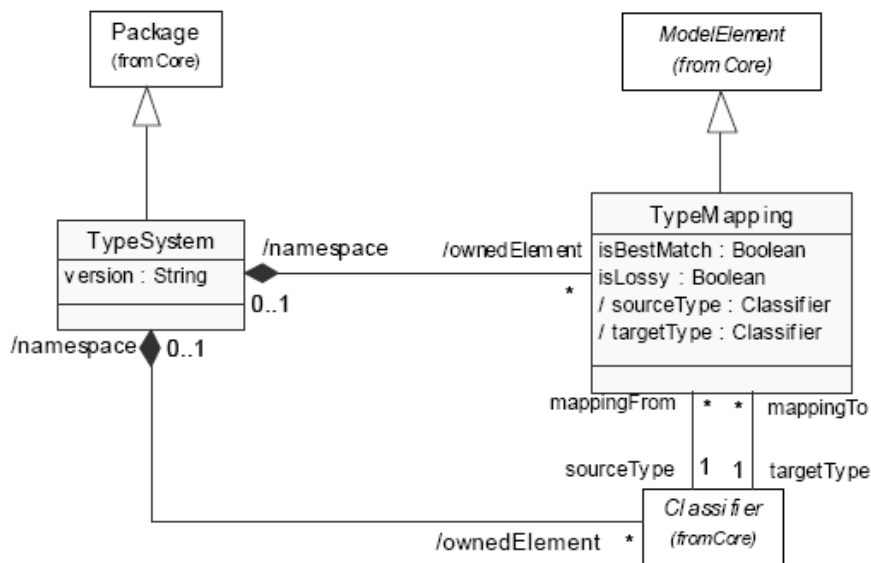
Formatted: Russian

Formatted: Russian

Formatted: Russian

## Type Mapping package

Съдържа класове и асоциации, които представят съответствието на типовете данни в различни системи. Той е показан на фиг. 28.



Фиг. 28 Type Mapping metamodel

Имплементацията на Type Mapping package в CWD4ALL е реализирана от Type Administrator (фиг. 29). Това е визуален интерфейс за по-лесно управление на типове и задаване на съответствие между типове в различни DBMS. Целта на тези съответствия е да се запази преносимостта на Реляционния и Физическите модели.

Formatted: Russian

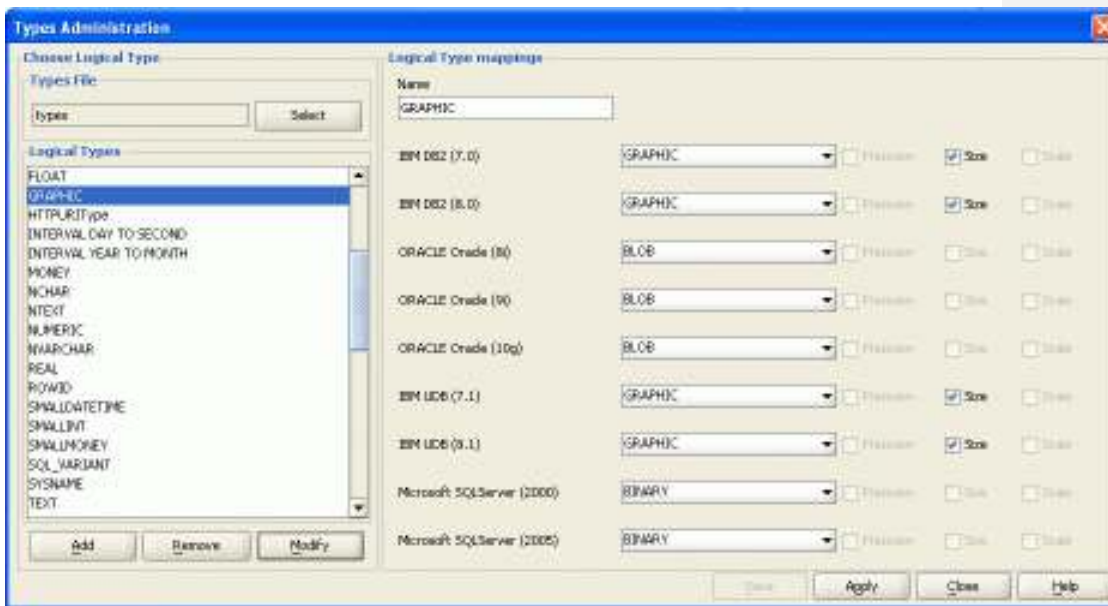
Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: English (U.S.)

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm



Фиг. 29 Type Administrator в CWD4ALL

Formatted: Centered

## Resource package

### Relational package

Съдържа класове и асоциации, които представят метаданни за реляционни данни.

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Relational package зависи от следните пакети:

- [org.omg::CWM::ObjectModel::Behavioral](#)
- [org.omg::CWM::ObjectModel::Core](#)
- [org.omg::CWM::ObjectModel::Instance](#)
- [org.omg::CWM::Foundation::DataTypes](#)
- [org.omg::CWM::Foundation::KeysIndexes](#)

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Formatted: Bullets and Numbering

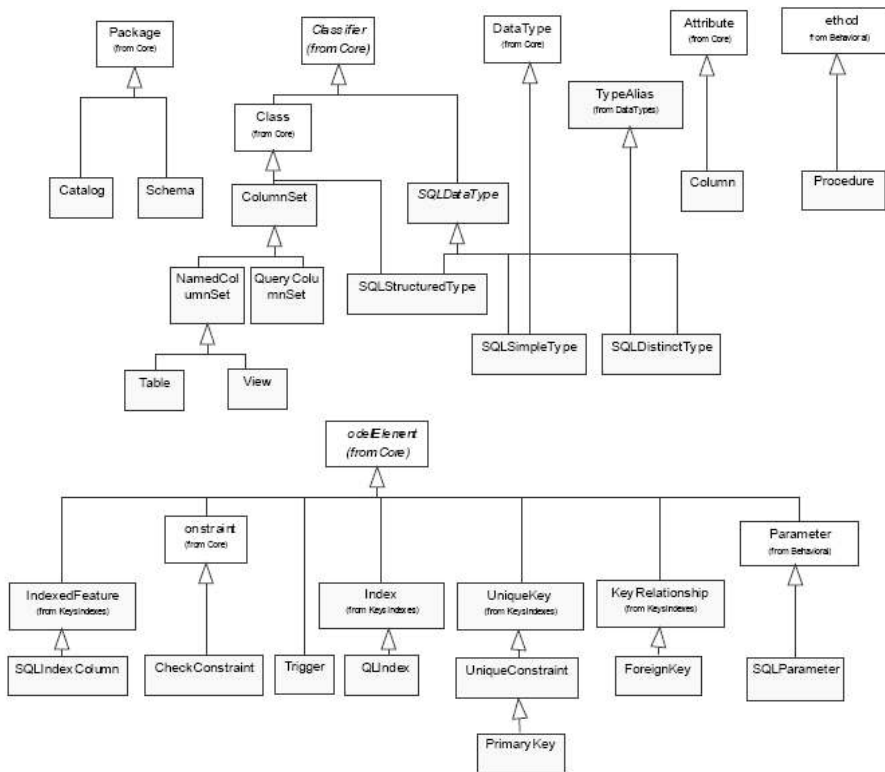
Relational package реферира the ObjectModel и Foundation пакетите. Той е представен на фиг. 30.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Relational пакетът е изцяло имплементиран в CWD4ALL от Реляционния модел и Type модела.

Formatted: Russian

Formatted: Russian



Фиг. 30 Relational metamodel

## Record package

Съдържа класове и асоциации, които представят метаданни за данни за запис.

Те са представени на фиг. 31.

Record пакетът обхваща основни концепции за запис и неговата структура. Той води до обширна представа на нотацията за запис, като включва както традиционните записи на данни като файлове и бази от данни, така и структурни типове на програмните езици.

Record пакетът зависи от следните пакети:

- org.omg::CWM::ObjectModel::Core
- org.omg::CWM::ObjectModel::Instance

Formatted: Russian

Formatted: Russian

Formatted: Centered

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

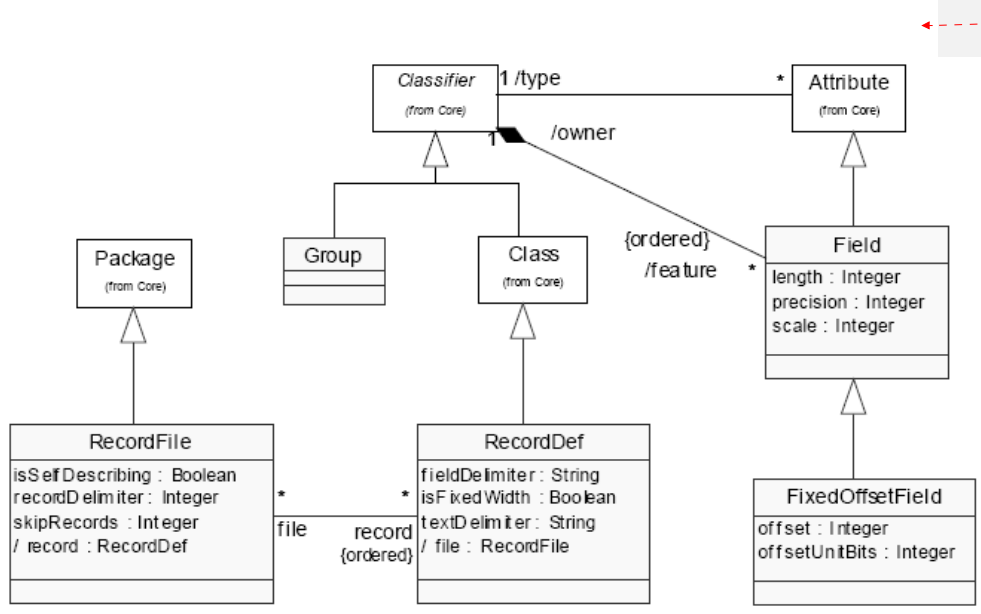
Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Formatted: Bullets and Numbering



Фиг. 31 Record metamodel

CWD4ALL реализира този пакет чрез Релационния, Физическия и Туре модела.

**Multidimensional package**

Съдържа класове и асоциации, които представят метаданни за многомерни данни.

Този пакет е изцяло имплементиран от [Многомерния модел](#) на CWD4ALL.

**XML package**

Съдържа класове и асоциации, които представят метаданни за XML данни.

Тъй като CWD4ALL е система за моделиране на бази от данни и данните се описват и съхраняват в базата, а не в отделни XML файлове, този пакет не е имплементиран. Разбира се, в базата могат да се пазят XML документи, но това става като всички останали данни.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

## Analysis package

### Transformation package

Ключов аспект в data warehousing е извличането, трансформацията и зареждането на данни от операционни ресурси в data warehouse или data mart за анализ. Извличането, трансформацията и зареждането могат да бъдат характеризирани като трансформации. На практика, дали трябва да се конвертират данни от една форма в друга в data warehousing, или за съхранение, възстановяване или представяне, трансформации са включени в процеса.

Transformation пакетът съдържа класове и асоциации, които представят метаданни за трансформация на данни, използвани в data warehousing. Той покрива основните трансформации на всички типове източници и цели: обектно-ориентирани, реляционни, за запис, многомерни, XML, OLAP, и data mining.

Transformation пакетът е разработен да:

- Свърже трансформацията с нейните data sources и targets. Тези data sources и targets могат да бъдат от всякакъв тип (например обектно-ориентирани, реляционни) или granularity (например клас, атрибут, таблица, колона).
- Обхване както "black box", така и "white box" трансформации. В случая на "black box" трансформации, data sources и targets са свързани с трансформацията и помежду си на общо ниво. Знае се, че data sources и targets са свързани с трансформацията, но не се знае как специфично парче data source е свързано със специфично парче data target. В случая на "white box" трансформации, е обратното.
- Позволи логическо групиране на трансформации.

Transformation package предполага наличието на следните пакети, които представят типове потенциални data sources или targets: ObjectModel, Relational, Record, Multidimensional, XML, OLAP, и Data Mining. Transformation пакетът е

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

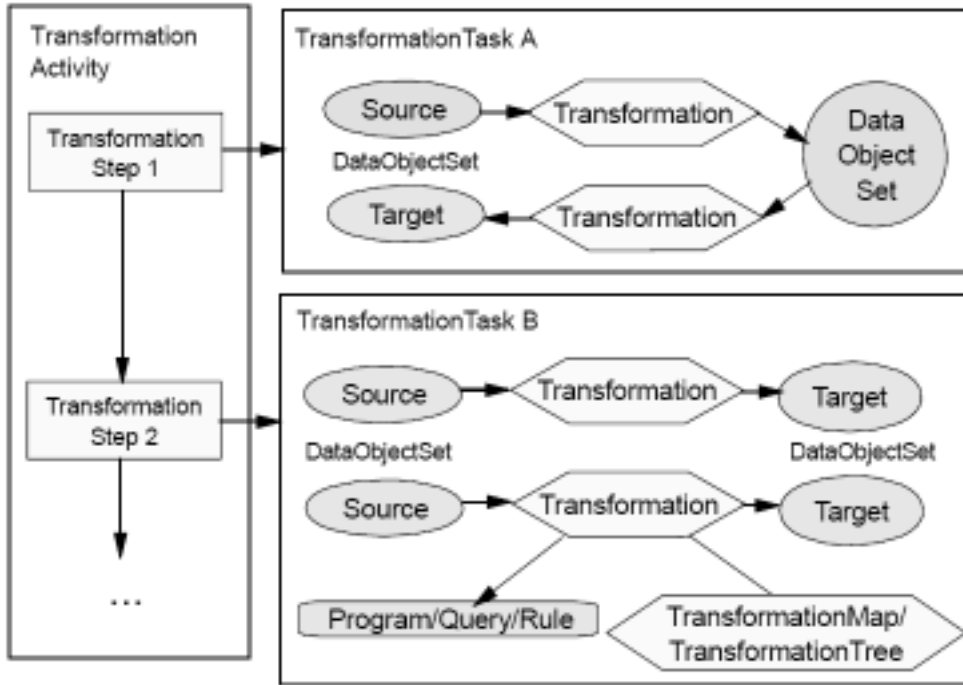
Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm



неразделна част от следните пакети: OLAP, Data Mining, Warehouse Process, и Warehouse Operation.

Семантиката на Transformation пакета е представена на ~~долната~~ ~~фиг.~~ ~~32.~~ ~~ура.~~



Фиг. 32 Семантика на Transformation metamodel

Трансформацията трансформира множество source обекти в множество target обекти. Множествата от обекти данни (data object set) могат да бъдат както sources, така и targets за различни трансформации.

Трансформациите могат да бъдат групирани в логически единици. На функционално ниво те се групират в transformation tasks, всеки от които дефинира множество от трансформации, които трябва да се изпълнят и завършат заедно – логическа единица работа. На ниво изпълнение, transformation steps се използват да координират потока на контрол между transformation tasks, всеки transformation step

Formatted: Russian

Formatted: Russian

Formatted: Bulgarian

Formatted: Centered

Formatted: Body Text + Times New Roman; 12 pt; Justified; First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

изпълнява една transformation task. Transformation steps от своя страна се групират в transformation activities.

Formatted: Russian

В CWD4ALL Transformation package има съответствие в Многомерния модел, тъй като там се дефинират source, target и transformation. Овен това един от обектите на Процес модела е TransformationPackage, който имплементира тъкмо transformation логиката.

Formatted: Russian

Formatted: Russian

### OLAP package

Съдържа класове и асоциации, които представят метаданни за системи за аналитични обработки на данни. Те са показани на фиг. 33.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Online Analytical Processing (OLAP) е клас аналитичен приложен софтуер, който разгръща бизнес данни в многомерен формат.

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

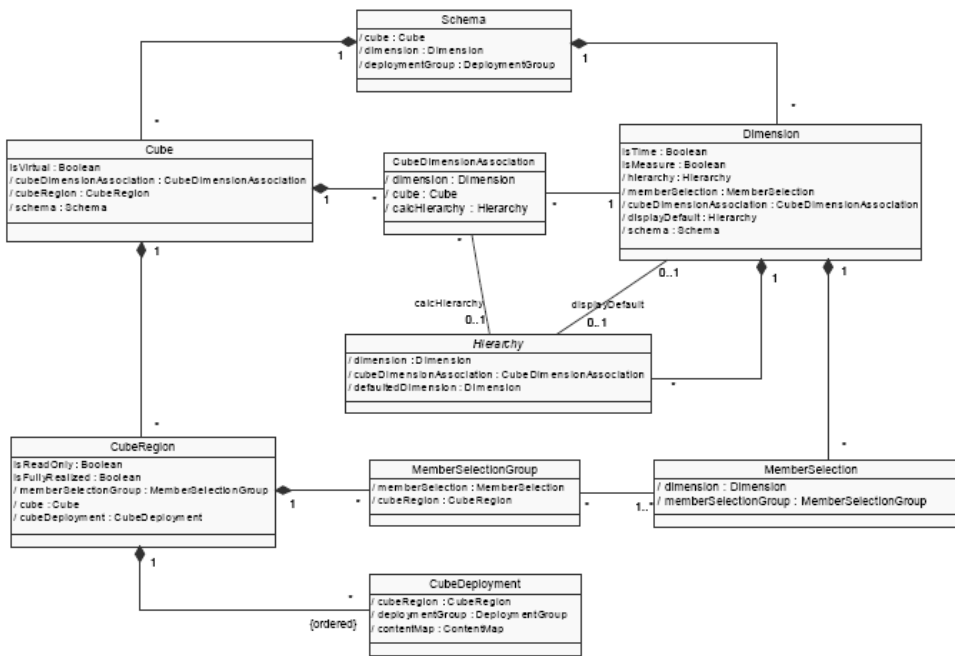
Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Има много начини за имплементация на OLAP. CWD4ALL поддържа логически OLAP структури в многомерния модел, които имат съответствие и в Логическия модел. Cube, Dimension, Hierarchy са основните обекти. Паралелно с това Физическия модел за Oracle също съдържа OLAP структури, включително materialized view и bitmap join indexes. При проектиране и управление на коректното поведение на релационния модел, CWD4ALL позволява на дизайнерите да изследват и определят актуалното поведение на системата – управление на пространството, undo datasets и пълен набор от специфични за имплементацията параметри могат да се задават.



Фиг. 33 OLAP metamodel

### Data Mining package

Съдържа класове и асоциации, които представят метаданни за data mining tools.

*Data mining* е приложение на математически и статистически процеси за целите на извличането на скрита информация от големи масиви с данни. Тази информация впоследствие се използва като business intelligence.

**Data Mining** метамоделят на CWD4ALL представя три концептуални области: цялостна дефиниция на самия модел, Settings, и Attributes.

Formatted: English (U.S.)

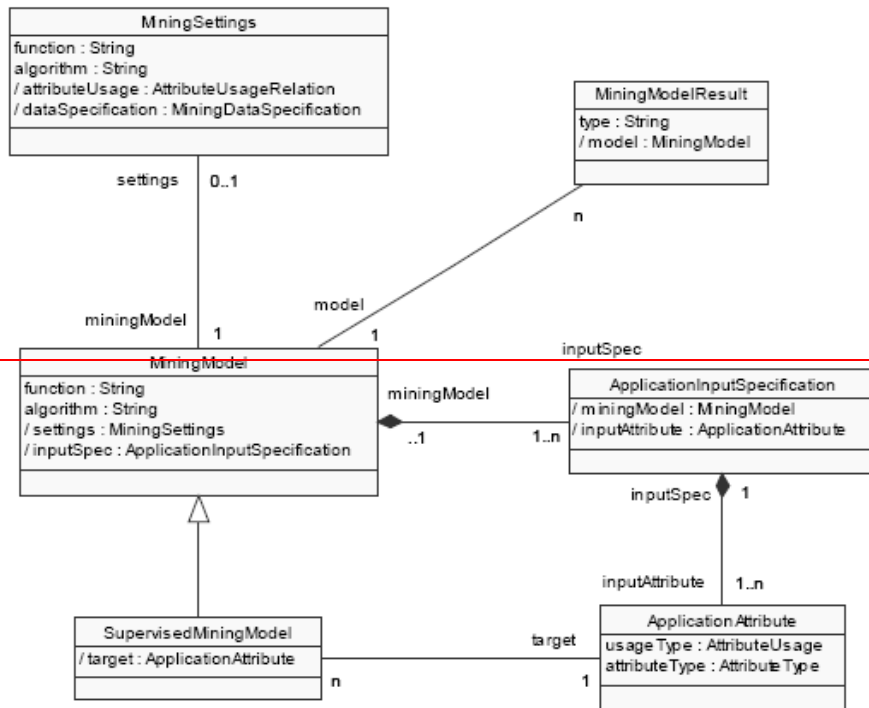
Formatted: Centered

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian



**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Left

Този пакет не е реализиран в CWD4ALL.

**Formatted:** Russian

### Information Visualization package

Съдържа класове и асоциации, които представят метаданни за системи за визуализация на информацията.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

**Formatted:** Russian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Самата система CWD4ALL визуализира на информация в подходящ форма.

**Formatted:** Russian

Целта ѝ, обаче, не е представяне на метаданни за подобна визуализация.

### Business Nomenclature package

Бизнес потребителите на data warehouses имат нужда да знаят каква информация се съдържа в data warehouse. Необходимо им е да разбират какво означава информацията от бизнес перспектива, откъде произхожда, от какви ресурси на данни произлиза и какви средства за анализ и справки съдържа.

**Formatted:** Bulgarian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Russian

BusinessNomenclature package съдържа класове и асоциации, които представят метаданни за класификация и речник на бизнеса. Той предоставя две основни конструкции:

- Таксономия (*Taxonomy*) е съвкупност от концепции, които предоставят контекста на значението на определен термин
- Речник (*Glossary*) е съвкупност от термини и различни свързани форми на термините

~~CWD4ALL представя тази информация визуално в дървото на обекти като то съдържа списък с модели и всеки модел има списък с неговите елементни имплементира този пакет.~~

## Management package

### Warehouse Process package

Съдържа класове и асоциации, които представят метаданни за warehouse процеси. Warehouse Process пакетът документира потока на процесите, използвани за изпълнение на трансформациите. Те могат да бъдат документирани на ниво на завършен TransformationActivity или неговите индивидуални TransformationSteps. WarehouseProcess обектите свързват трансформацията с множество събития, които ще се използват при изпълнението на трансформацията.

Този пакет не е имплементиран в CWD4ALL.

### Warehouse Operation package

Съдържа класове и асоциации, които представят метаданни за warehouse операции. Пакетът покрива три различни области:

- Transformation Executions – детайли за най-скорошните изпълнения на трансформации могат да бъдат записани, идентифицирани и след това завършени успешно.
- Measurements – тези обекти позволяват всеки елемент на модела да съдържа метрика.
- Change Requests – позволяват детайли за промените, засягащи всеки елемент на модела, да бъдат записани.

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Bulgarian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

•

CWD4ALL имплементира Measurement и Change Request за всеки обект на модела, но не и Transformation Execution. Система за моделиране като CWD4ALL не може да третира сурови данни, които са M1-level (Model Layer) според OMG metamodel architecture. CWD4ALL произвежда модели, които са M1-level. Затова, бидейки система за моделиране, не е възможно напълно да имплементира Warehouse Operation пакета.

Formatted: Bullets and Numbering

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

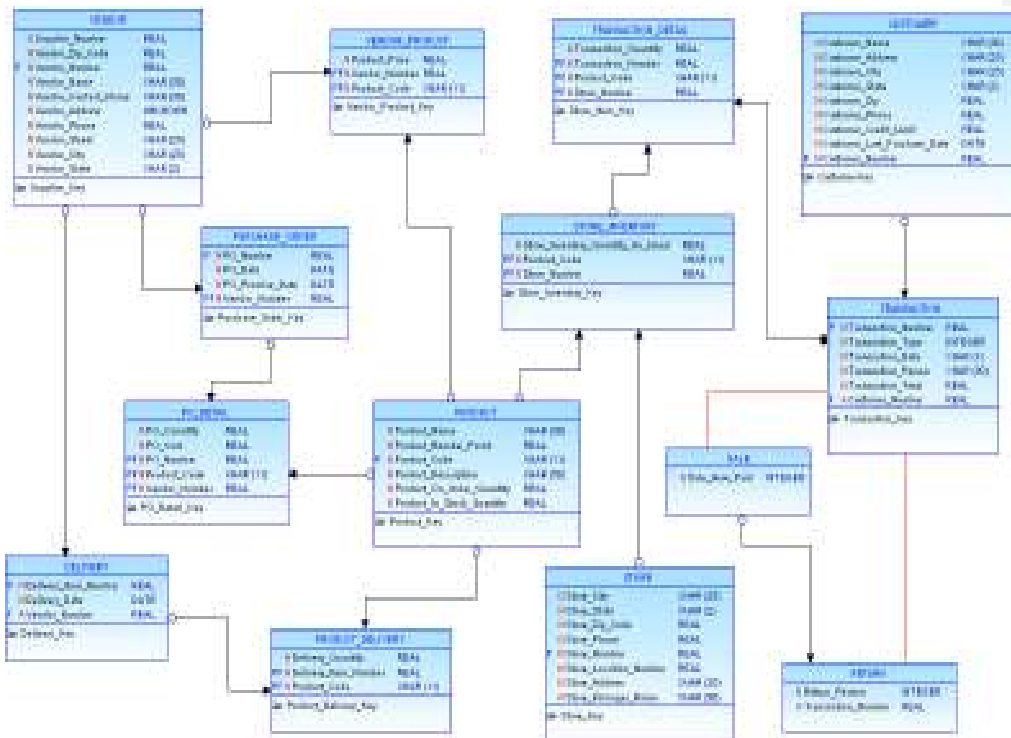
Formatted: Russian

## Глава 4. Използване на CWD4ALL при реализиране на конкретни модели

### Модел Handy Hardware Company

Това е модел на система за хипотетична хардуерна компания Handy Hardware. Тя съдържа информация, свързана с дейността на фирмата – за търговци, продукти, доставки, клиенти и продажби. Дизайнът ѝ се състои от следните подмодели:

- Логически модел
- Релационен модел
- Процес модел, който включва 3 Диаграми на потока на данни - Point of Sale Project, Store Inventory Control Project и Returned Goods Processing Project



Formatted: Russian

Formatted: Russian

Formatted: Bullets and Numbering

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 1,59 cm + Tab after: 2,22 cm + Indent at: 2,22 cm

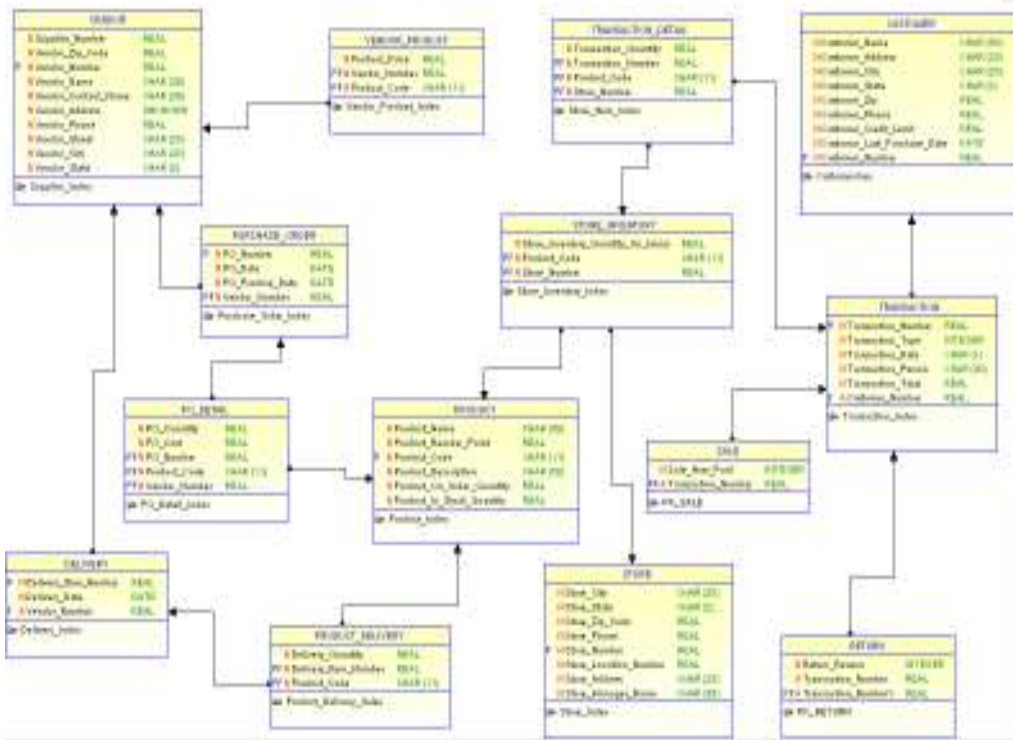
Formatted: Bullets and Numbering

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Фиг. 34** Логически модел на Handy Hardware Company

Логическият модел, показан на фиг. 34, определя основните логически единици данни в системата на компанията. Той се състои от ентитита като **VENDOR, DELIVERY, PRODUCT, SALE, STORE, CUSTOMER**. Всяко от тях е нормализирано според правилата, описани в секция 1.

Връзките между отделните ентитита са моделирани, както чрез връзки (relationships), така и чрез други ентитита, както в случая, описан в секция 1. Дефинирана е йерархия със супертип ентити **TRANSACTION** и подтип ентитита **SALE** и **RETURN**. Смисълът ѝ е, че и продажбата, и връщането на стока са вид транзакция и като такива имат общи характеристики, дефинирани в супертип ентитито.



**Фиг. 35** Релационен модел на Handy Hardware Company

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

**Formatted:** Russian

**Formatted:** Russian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

**Formatted:** Russian

**Formatted:** Russian

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm



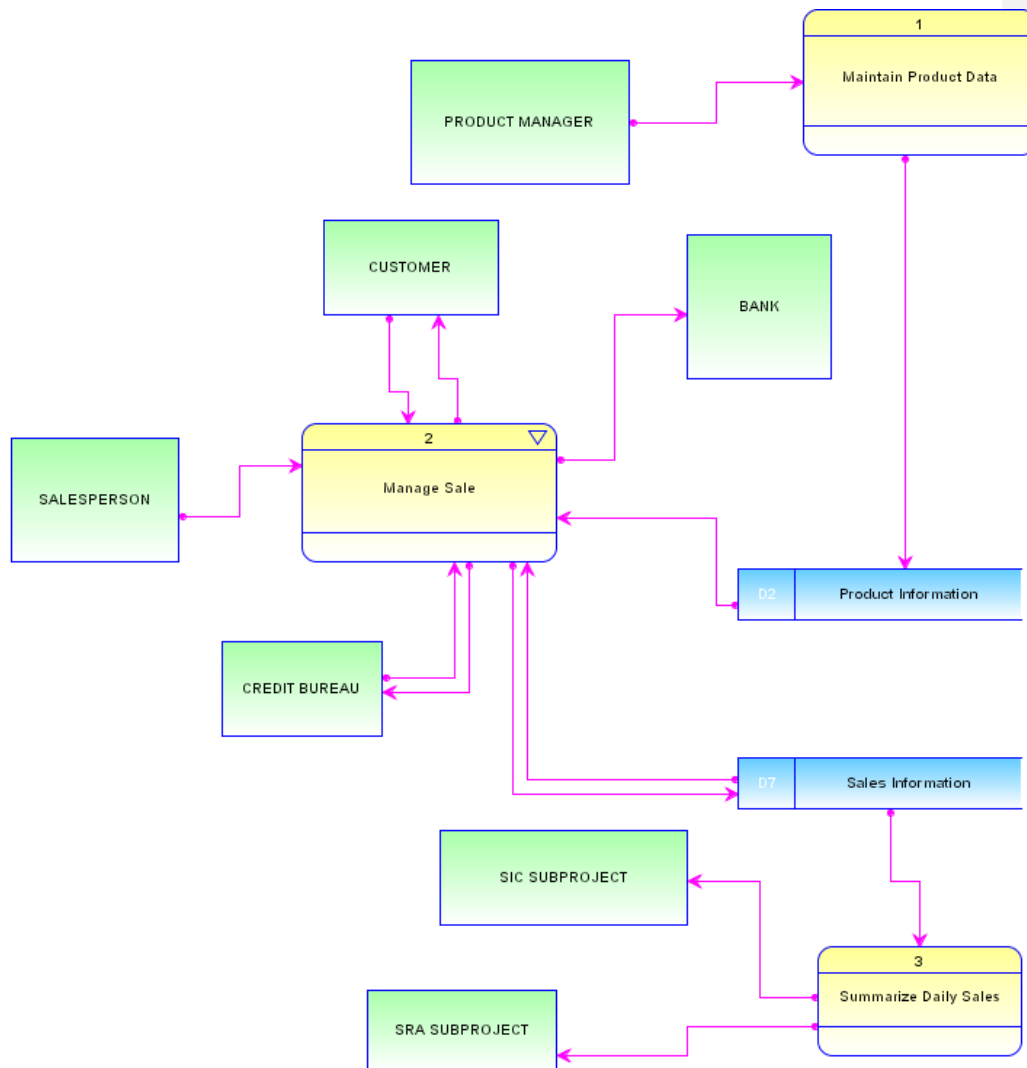
Релационният модел, показан на фиг. 35, е получен от директно преобразуване (forward engineering) на Логическия модел. На всяко ентити съответства таблица с колони, съответстващи на атрибутите на това ентити. Избраната стратегия за преобразуване на йерархията от ентитита със суперттип ентити TRANSACTION и подтип ентитита SALE и RETURN генерира съответните три таблици като към SALE и RETURN се добавя по един първичен ключ и чужд ключ към суперттип ентитито.

Създаденият вече Релационен модел би могъл да бъде разширен до Физически модел за някои от поддържаните от CWD4ALL бази – IBM DB2 7.0, 8.0, IBM UDB 7.1, 8.1, Oracle 8i, 9i, 10g или Microsoft SQL Server 2000, 2005.

След като са идентифицирани основните единици данни в системата, следва да се определят бизнес процесите между тях. Процес моделът на Handy Hardware Company се състои от 3 Диаграми на потока на данни.

Formatted: Russian

Formatted: Russian

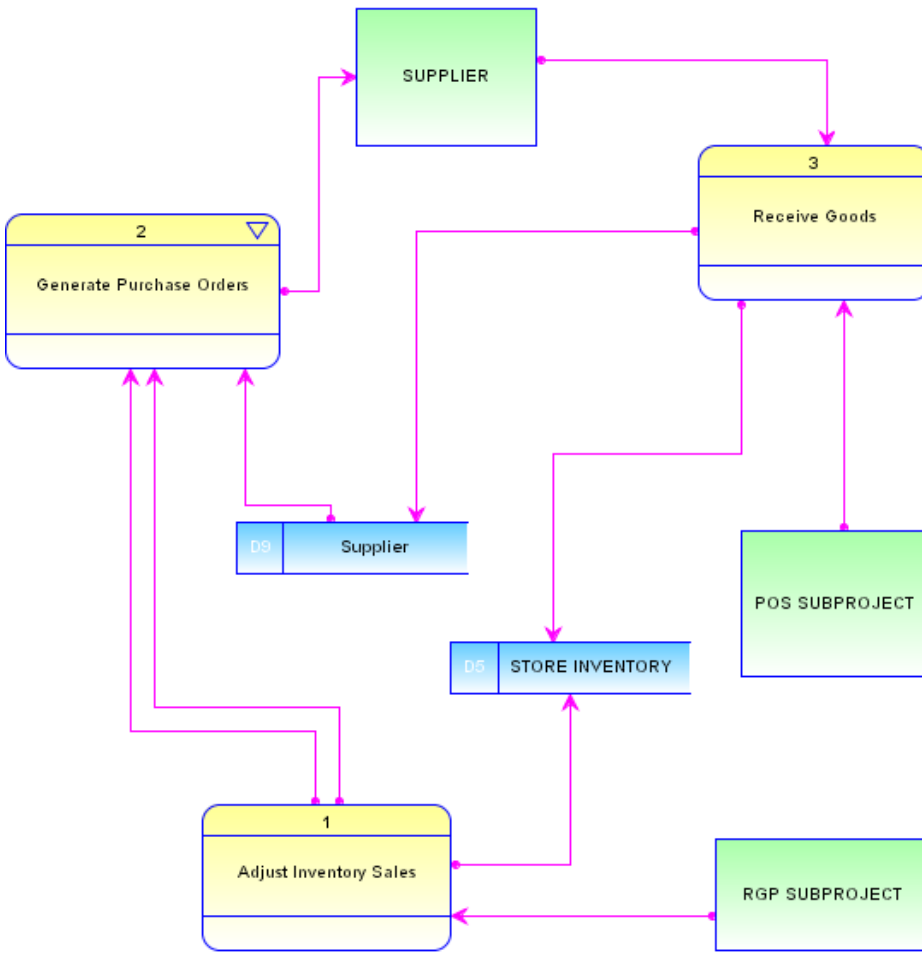


Фиг. 36 Диаграма на потока на данни Point of Sale Project

Диаграмата на потока на данни Point of Sale Project, показана на фиг. 36, моделира процесите, които протичат при продажба на даден продукт. Дефинирани са информационните хранилища Product Information и Sales Information, външните за системата агенти CUSTOMER, SALESPERSON, CREDIT BUREAU, PRODUCT MANAGER, BANK, SIC SUBPROJECT и SRA SUBPROJECT, процесите Maintain Product Data, Manage Sale и Summarize Daily Sales, и потоците между тях. В зависимост от бизнес правилата тези потоци могат да бъдат двупосочни – в такъв

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

случай се дефинират 2 потока с различни посоки. Процесът Manage Sale е композитен (вж. глава 2) и има собствена Диаграма на потока на данни.



Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

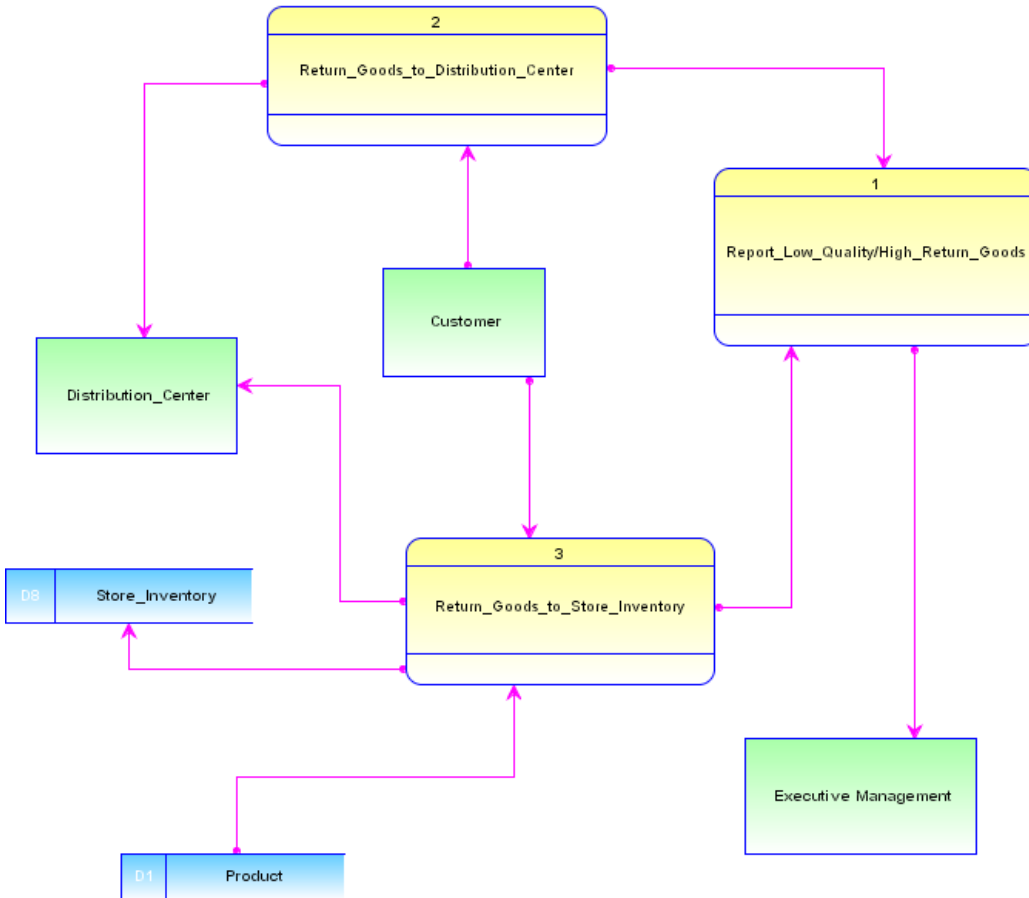
Фиг. 37 Диаграма на потока на данни Store Inventory Control Project

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

Диаграмата на потока на данни Store Inventory Control Project, показана на фиг. 37, моделира процесите, които протичат при съхранението на инвентара на компанията. Дефинирани са информационните хранилища STORE INVENTORY и Supplier, външните агенти SUPPLIER, SIC SUBPROJECT и SRA SUBPROJECT, процесите Adjust Inventory Sales, Generate Purchase Orders и Receive Goods, и

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

потоците между тях. Процесът Generate Purchase Orders е композитен (вж. глава 2) и има собствена Диаграма на потока на данни.



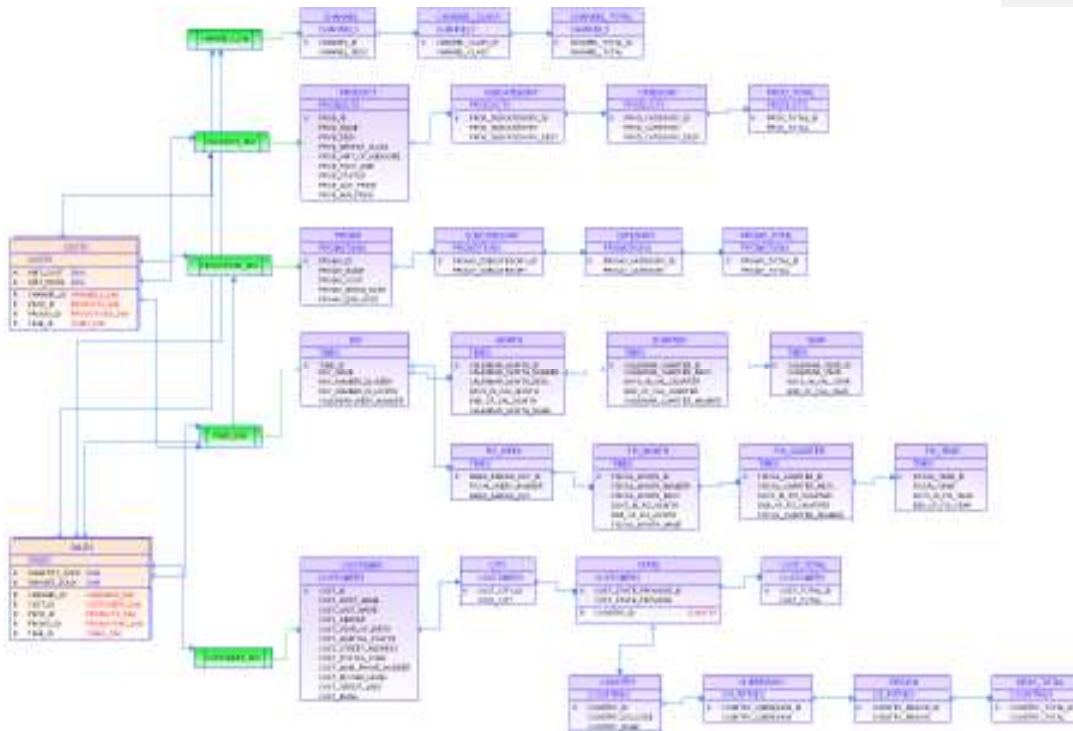
**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Centered

Фиг. 38 Диаграма на потока на данни Returned Goods Processing Project

Диаграмата на потока на данни Returned Goods Processing Project, показана на фиг. 38, моделира процесите, които протичат при връщането на стоки. Дефинирани са информационните хранилища STORE INVENTORY и Product, външните агенти Customer, Distribution\_Center и Executive Management, процесите Report\_Low\_Quality/High\_Return\_Goods, Return\_Goods\_to\_Distribution\_Center и Return\_Goods\_to\_Store\_Inventory, и потоците между тях.

## Модел SH

Моделът SH (Sales history) представлява дизайн на известната многомерна база SH, която е включена в дистрибуцията на последните инсталации на Oracle (вж. [10]). Тя съдържа огромно количество данни (само таблицата ѝ SALES съдържа над 1 милион записи) и подлежи на аналитични обработки (OLAP). В тази секция е описано как може да се моделира тя с помощта на CWD4ALL.



Фиг. 39 Многомерен модел на SH

Пред вид броя записи, които ще се съхраняват в базата и избора на реализация, която поддържа многомерни данни (Oracle), е ефективно да се моделира Многомерен дизайн.

Многомерния модел на SH, представен на фиг. 39, дефинира кубовете COSTS (за календарен период от време) и SALES (за фискален период от време), измеренията CHANNELS\_DIM, PRODUCTS\_DIM, PROMOTIONS\_DIM, TIMES\_DIM и CUSTOMERS\_DIM, както и следните йерархии от нива:

Formatted: Heading 2; Знак Знак  
Знак

Formatted: Font: (Default) Times New Roman

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman; 12 pt; Justified; First line: 0.95 cm, Centered

Formatted: Body Text + Times New Roman; 12 pt; Justified; First line: 0.95 cm

- CHANNEL, CHANNEL\_CLASS, CHANNEL\_TOTAL
- PRODUCT, SUBCATEGORY, CATEGORY, PROD\_TOTAL
- PROMO, SUBCATEGORY, CATEGORY, PROMO\_TOTAL
- DAY, MONTH, QUARTER, YEAR
- DAY, FIS\_WEEK, FIS\_MONTH, FIS\_QUARTER, FIS\_YEAR
- CUSTOMER, CITY, STATE, CUST\_TOTAL
- CUSTOMER, CITY, STATE, COUNTRY, SUBREGION, REGION, GEOG\_TOTAL

Моделът SH е използван за пример в глава 2, където има описание на всеки отделен обект.

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Bulleted + Level: 1 + Aligned at: 0,63 cm + Tab after: 1,27 cm + Indent at: 1,27 cm

**Formatted:** Bullets and Numbering

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

## Заклучение

Системата CWD4ALL улеснява работата на анализатори, администратори, мениджъри и крайни потребители като предоставя функционалност за извличане, моделиране, управление, използване и обмен на метаданни. Като следва CWM доколкото е възможно за една система за моделиране, CWD4ALL се придържа към широко разпространен стандарт за обмен на метаданни.

Възможните направления за развитие на CWD4ALL са поддръжка на повече бази данни, подобряване на визуалния интерфейс, импорт и експорт на все повече различни формати метаданни, конструиране на изрази и код за базите, доразвиване на инструментите за оценка и подобряване на бързодействието в базите.

~~Разработената дипломна работа представлява изследване на системата за проектиране и моделиране на бази от данни CWD4ALL. Изводите и обобщенията систематизират резултата от проучването на темата и предлагат разсъждения за възможностите и предимствата за приложение на системата в реални модели.~~

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Bulgarian

Comment [S29]: кои са изводите и обобщенията ???

Formatted: Russian

Comment [S30]: думата разсъждения не е подходяща – анализират се, изследват се, представят се .....възможностите

Formatted: Russian

Formatted: Indent: Left: 0 cm, First line: 0 cm

## Речник на използваните съкращения и термини

### A

#### Aggregation

Процес на комбиниране на няколко обекти в един.

#### Attribute

Приложимо свойство или характеристика на ентити.

### B

#### Business Rule

Бизнес политика или насока, ~~която~~която може да бъде приложена чрез модела на данни.

#### Business Intelligence (BI)

Обхваща процесите по събирането, управлението и анализирането на голям обем от данни с цел подпомагане на стратегическите бизнес решения на компанията, както и да улесни служителите от различните отдели а реагират гъвкаво на пазарните промени. С BI системата данните се трансформират в полезна информация.

### C

#### Candidate Key

Един или повече атрибути, които уникално идентифицират инстанция на ентити.

#### Cardinality

Във връзка, съотношението на свързаните parent и child инстанции на ентити.

#### Child Entity

Ентити, което наследява foreign key от друго ентити.

#### Column

Колоната представя приложимо свойство или характеристика на таблица.

Formatted: Russian

Formatted: Font: Bold, Italic

Formatted: Font: Italic, Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Russian

Formatted: Font: Bold, Italic

Formatted: Font: Italic, Russian

Formatted: Russian

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Font: Not Bold, Russian

Formatted: Font: Not Bold, Russian

Formatted: Russian

Formatted: Font: 12 pt, Font color: Black

Formatted: Font: 12 pt, Font color: Black

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Font: Not Bold, Russian

Formatted: Font: Not Bold, Russian

Formatted: Russian

Formatted: Font: Not Bold, Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Font: Not Bold, Russian

Formatted: Font: Not Bold, Russian

Formatted: Russian

Formatted: Russian

Formatted: Russian

Formatted: Font: Not Bold, Russian

Formatted: Russian



Constraint

Механизъм за поддържане на валидни стойности на данните.

CRUD

Матрица, която съответства на функциите Create, Replace, Update и Delete за дадена инстанция на ентити.

**D**

Database\*

Организирана съвкупност от данни, съхраняващи се в таблици.

Data Model

Логическа спецификация на структури от данни и бизнес правила, ръководени от бизнес област.

Datatype

Форма на данните, които могат да бъдат съхранявани в база от данни.

Data Definition Language

Езикът и синтаксисът за дадена The language and syntax for a given DBMS, който се използва за създаване, модифициране и изтриване на обекти на база used to create, modify and drop database objects данни.

Denormalization

Съзнателната модификация на модела на данни в нормална форма от по-нисък ред The intentional modification of a data model to a lower level normal form. Денормализацията обикновено се предприема, за да се постигне подобрене в бързодействието на физическия модел Denormalization is usually undertaken to achieve improved performance in implementing a physical design.

Derived Attribute

Атрибут, чиято стойност може да бъде определена от стойностите на други атрибути An attribute whose value can be determined from the values of other attributes.

Diagram

The data model in its entirety, including its logical and physical designs.

Domain

Валидните стойности, които може да приема даден атрибут The valid values than an attribute can take.

Formatted: Font: Not Bold, Russian

Formatted: Russian

Formatted: Font: Not Bold, Russian

Formatted ... [1]

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Font: Not Bold, Russian

Formatted: Russian

Formatted ... [2]

Formatted: Russian

Formatted: Font: Not Bold, Russian

Formatted ... [3]

Formatted ... [4]

Formatted: Font: Not Bold, Russian

Formatted ... [5]

Formatted ... [6]

Formatted ... [7]

Formatted ... [8]

Formatted: Font: Not Bold, Russian

Formatted ... [9]

## E

### Entity

Човек, място, понятие, събитие или концепция, за които се пази информация. A distinguishable person, place, thing, event or concept about which information is kept.

Formatted: Russian

Formatted: Font: Not Bold, Russian

Formatted: Body Text + Times New Roman; 12 pt; Justified; First line: 0.95 cm

Formatted ... [10]

### Entity Instance

A single occurrence or member of an entity.

Formatted: Bulgarian

Formatted: Russian

## F

### Forward Engineering

Процес на преобразуване на логически модел в реляционен. The process of converting a logical model to a physical design for deployment on a database platform.

Formatted ... [11]

Formatted: Body Text + Times New Roman; 12 pt; Justified; First line: 0.95 cm

Formatted ... [12]

### Foreign Key

Първичен ключ или неключов атрибут, наследен от друго ентити. A primary key or non key attribute that is inherited from another entity.

Formatted ... [13]

Formatted ... [14]

## I

### Index

Обект на базата данни, използван за налагане спазването на уникалността на колони в таблица. A database object used to enforce unique values in a table; an index can also be used to access table data more efficiently.

Formatted: Russian

Formatted: Body Text + Times New Roman; 12 pt; Justified; First line: 0.95 cm

Formatted: Font: Not Bold, Russian

Formatted ... [15]

### Inheritance

Процес на разпространение на чужди ключове от родителски ентитита към подентитита, базиран на правила за връзки. The process of propagating foreign keys from parent entities to child entities based on relationship rules.

Formatted: Font: Not Bold, Russian

Formatted ... [16]

### Instance

A single occurrence or member of an entity. Единично срещане.

### Integrity

A property of a data model in which all assertions hold.

### Integrity Constraint

Характеристика на база данни, която налага спазването на връзките с чужди ключове. A database feature that enforces foreign key relationships.

Formatted ... [17]

Formatted ... [18]

## K

### Key

Either a primary key, foreign key or alternate key.

## L

### Logical Design

The dimension of a data model that addresses real system requirements in the abstract, without consideration of data storage, performance or other physical implementation issues.

## M

### Metadata

All the information about a data warehouse that is not the actual stored data itself. Данни, които описват структурата и връзките между данните Metadata describes the structure and relationship of data.

### Metamodel

Модел за моделите. Метамоделът описва основната структура на модела A metamodel describes the underlying structure of a model.

## N

### Normalization

Процес на премахване на неточни, непълни, излишни и/или прекалено сложни твърдения в модела на данни. The process of removing inaccurate, inconsistent, redundant and/or overly complex assertions in a data model.

### Not Null

Състояние, при което винаги има стойност. The state of always having a value.

### Null

Състояние, при което няма стойност. The state of having no value.

## O

### Object Type

An abstract data type or object composed of a collection of similar types of data.

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Font: Not Bold, Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted ... [19]

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Font: Not Bold, Russian

Formatted ... [20]

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted: Font: Not Bold, Russian

Formatted ... [21]

Formatted ... [22]

Formatted ... [23]

Formatted: Font: Not Bold, Russian

Formatted ... [24]

## P

### Parent Entity

An entity that contributes a foreign key to another entity.

### Physical Design

The translation of a data model for implementation on a database platform. The physical design shows how the data is stored in the database.

### Primary Key

Атрибут или множество атрибути, избрани уникално да идентифицират всяко срещане на снтити. An attribute or set of attributes that have been chosen to uniquely identify every instance of an entity.

### Propagation

The process by which a parent entity contributes foreign keys to a child entity.

## R

### Referential Integrity

Характеристики на базата данни, които автоматично осигуряват факта, че всеки чужд ключ има съответна стойност на първичния ключ. Database features that automatically ensure that each foreign key value has a matching primary key value.

### Relationship

Връзка между две единици, която изразява някакво бизнес правило. In the IDEF1X methodology, there are three basic types of relationships: identifying relationships, non-identifying relationships, and non-specific relationships.

### Relational Model

A tabular data model in which data is represented in tables with records stored in rows and data elements expressed as table columns.

### Repository

A central database that stores information about the elements and structure of a data model.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted ... [25]

Formatted ... [26]

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted ... [27]

Formatted ... [28]

Formatted: Font: Not Bold, Russian

Formatted ... [29]

## Reverse-Engineering

Процес на преобразуване на релационен в логически модел. The process of extracting the definition of database objects, usually from a database's system catalog.

Formatted ... [30]

Formatted ... [31]

## S

### Subtype

A subset of entity instances that share common attributes or relationships distinct from other subsets.

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

### Supertype

The parent entity in a subtype cluster that represents the superset of the subtypes.

### Synonym

Алтернативно име на обект от базата данни. An alternate name for a database object.

Formatted: Font: Not Bold, Russian

Formatted ... [32]

## T

### Table

Основната единица в съхранението на базата данни. Таблицата съответстват на ентити в логическия модел. The basic unit of data storage in a database. Tables correspond to entities in the logical model.

Formatted: Russian

Formatted: Font: Not Bold, Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted ... [33]

## U

### Unique Index

Обект за достъп в базата, който осигурява уникалността на всеки ред в дадена таблица. A database access object that ensures that every row in a table can be uniquely identified.

Formatted: Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted ... [34]

Formatted ... [35]

## User-Defined Datatype

A customized and reusable datatype built from basic datatypes.

Formatted: Russian

## V

### View

Релационен обект, ползван за конструиране на виртуални изгледи на данни, без да се интересува къде са пазят данните. A relational object used for constructing virtual views of data without regard for where the data resides.

Formatted: Font: Not Bold, Russian

Formatted: Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm

Formatted ... [36]

## Използвана литература

- [1] <http://www.cwd4all.com/product.cfm>
- [2] [CWD4ALL Help](#)
- [3] [Global Software Innovators: CASE and Reengineering](#), Software Magazine, Jan, 1991 by Jessica Keyes
- [4] [Bachman, Charles W., "Data Structure Diagrams." DataBase: A Quarterly Newsletter of SIGBDP vol. 1, no. 2, Summer 1969.](#)
- [5] [Bachman, Charles W., "The Role Data Model Approach to Data Structures." International Conference on Data Bases, March 24, 1980.](#)
- [6] [Database Design Tool: CWD4ALL for Modeling SQL and Multidimensional Data, review SQL Summit, http://www.sqlsummit.com/Product/CWD4ALL.htm](#)
- [7] [Ikan Adds Modeling Capability to Oracle SQL Developer](#), Oracle Magazine, September/October 2006, <http://www.oracle.com/technology/oramag/oracle/06-sep/o56pnews.html>
- [8] [Bachman Documentation](#), DynaText Book
- [9] [DBA4ALL Documentation](#)
- [10] [Oracle9i Database Online Documentation \(Release 9.0.1; includes Windows books\)](#)
- [11] [http://david\\_hewins.tripod.com/papers/id12.html](http://david_hewins.tripod.com/papers/id12.html)
- [12] [Common Warehouse Metamodel \(CWM\) Specification](#), <http://www.omg.org/technology/documents/formal/cwm.htm>
- [13] [Embarcadero ER/Studio](#), <http://www.embarcadero.com/>
- [14] [Sybase Powerdesigner](#), <http://www.sybase.com/developer/products/pdtechcorner>
- [15] [CA ERwin](#), <http://www3.ca.com/solutions/Product.aspx?ID=260>
- [16] [IBM Rational Data Architect](#), <http://www-306.ibm.com/software/data/integration/rda/>

**Formatted:** Normal, Indent: Left: 0 cm, First line: 0 cm

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Indent: Left: 0,95 cm, First line: 0 cm

**Formatted:** Font: Times New Roman

**Formatted:** Font: Times New Roman, Not Bold

**Formatted:** Font color: Blue

**Formatted:** Font: Times New Roman, Not Bold

**Formatted:** Font: Times New Roman, Not Bold

**Formatted:** Font: Times New Roman, 12 pt

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Indent: Left: 0,95 cm, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

**Formatted:** Font: 12 pt

**Formatted:** Font: 12 pt, Font color: Blue

**Formatted:** Font: 12 pt

**Formatted:** German (Germany)

**Formatted:** Body Text + Times New Roman;12 pt;Justified;First line: 0.95 cm, Indent: Left: 0,95 cm, First line: 0 cm

**Formatted:** Font: (Default) Times New Roman, 12 pt

**Formatted:** Font: (Default) Times New Roman, 12 pt, Font color: Blue

**Formatted:** Font: (Default) Times New Roman, 12 pt

**Formatted:** Font: (Default) Times New Roman, 12 pt

**Formatted:** German (Germany)

**Formatted:** German (Germany)

**Formatted:** Font color: Blue













<b>Page 90: [12] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:11:00</b>
Russian		
<b>Page 90: [12] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:11:00</b>
Russian		
<b>Page 90: [12] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:11:00</b>
Russian		
<b>Page 90: [12] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:11:00</b>
Russian		
<b>Page 90: [13] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Font: Not Bold, Russian		
<b>Page 90: [13] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Font: Not Bold, Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [14] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [15] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		
<b>Page 90: [15] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Russian		



<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [16] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:20:00</b>
Russian		
<b>Page 90: [17] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Font: Not Bold, Russian		
<b>Page 90: [17] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
Font: Not Bold, Russian		
<b>Page 90: [18] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:22:00</b>
Russian		
<b>Page 90: [18] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:22:00</b>
Russian		
<b>Page 90: [18] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:22:00</b>
Russian		
<b>Page 90: [18] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:22:00</b>
Russian		





















<b>Page 93: [35] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:42:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [35] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:42:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian

<b>Page 93: [36] Formatted</b>	<b>Stefka Kanova</b>	<b>23.9.2006 г. 19:44:00</b>
--------------------------------	----------------------	------------------------------

Russian