



Голубь Павел
MicroOLAP Technologies

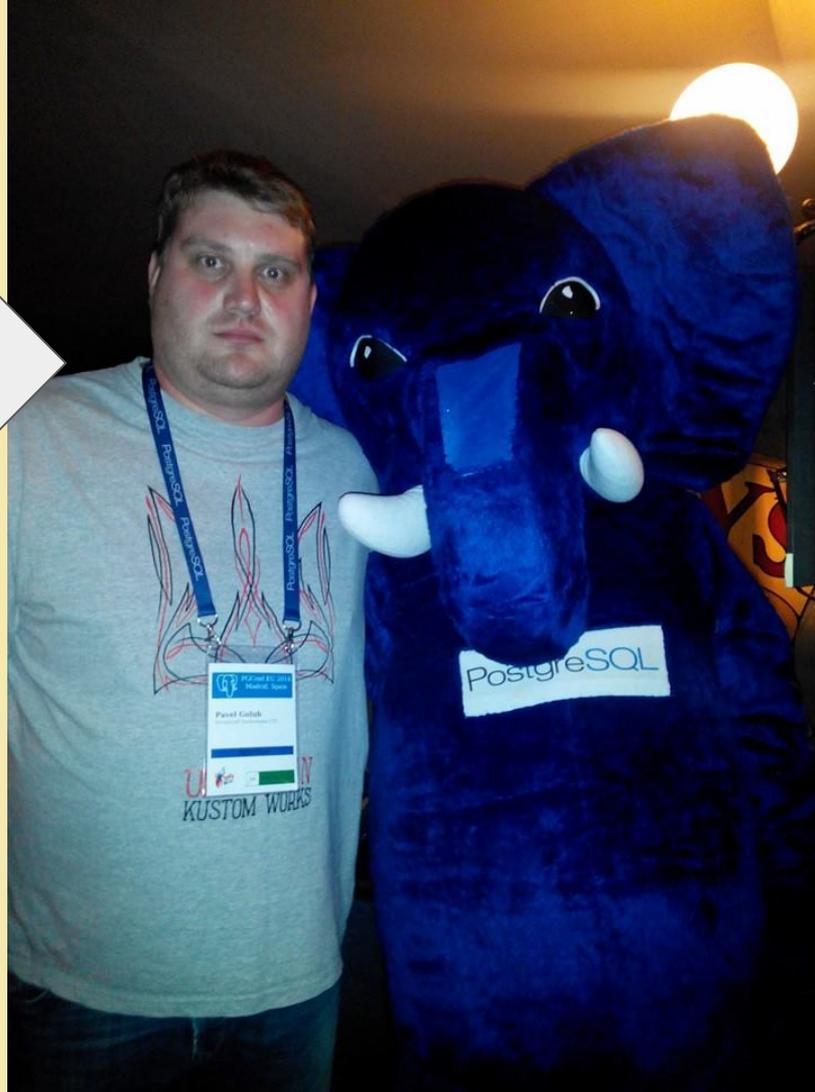
PGDAY'17
RUSSIA

КОНФЕРЕНЦИЯ
ПО БАЗАМ ДАННЫХ

Визуальное моделирование баз данных



ГОЛУБЬ!!!



СЛОНИК!!!

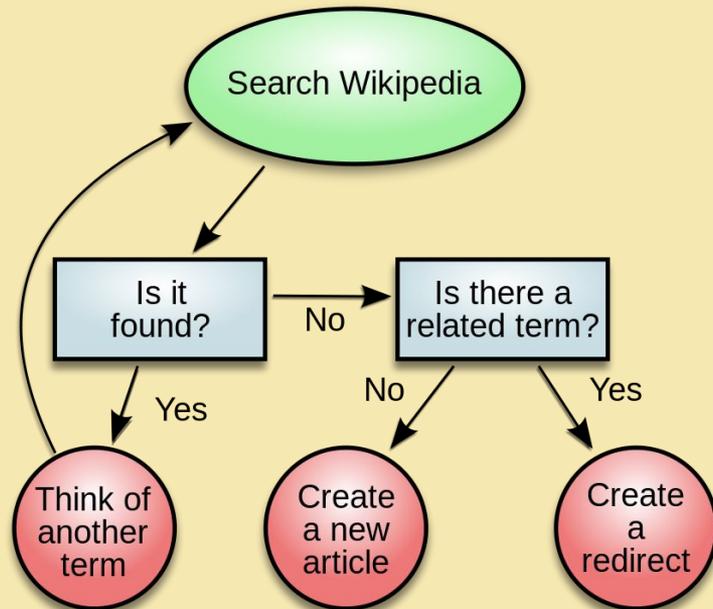
Визуальное моделирование

Определение

Определение моделирования

- Визуальное моделирование использует стандартное графическое представление

Adding an article to Wikipedia



Пример представления принятия решения о новой статье Wikipedia.

Определение моделирования

- Визуальное моделирование использует стандартное графическое представление
- **Коллекция визуальных объектов для описания данных, связей, семантики и ограничений**

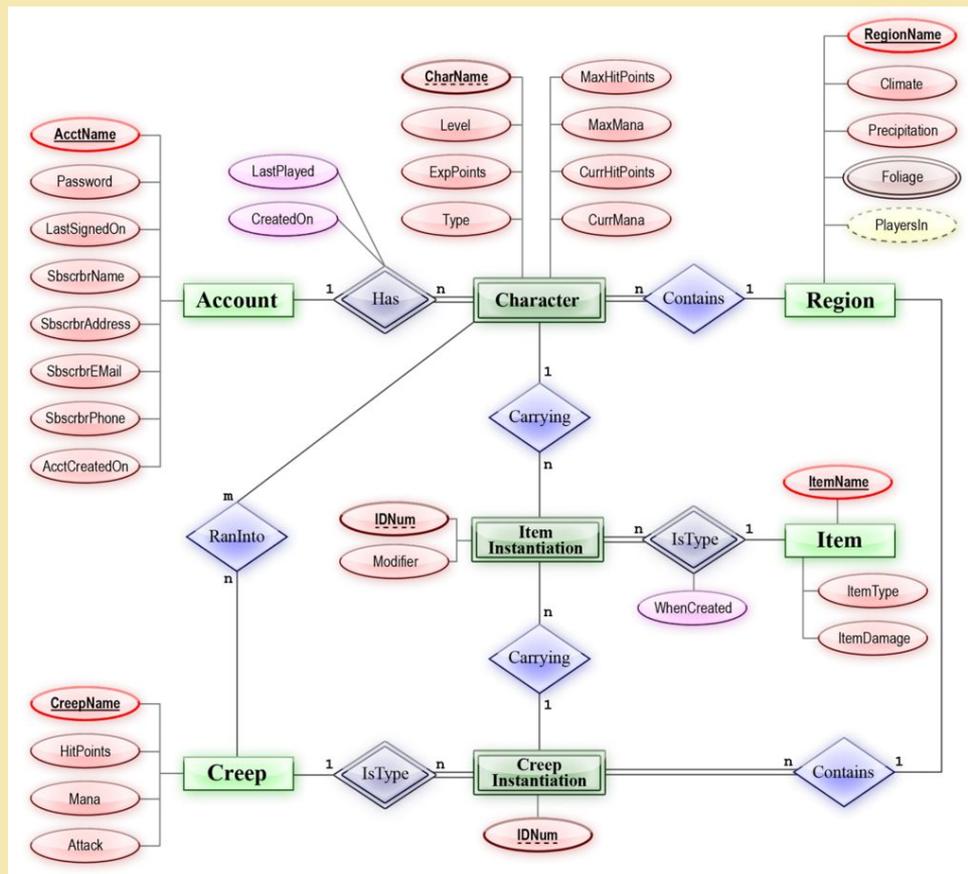


Диаграмма «сущность-связь» в нотации Чена

Определение моделирования

- Визуальное моделирование использует стандартное графическое представление
- Коллекция визуальных объектов для описания данных, связей, семантики и ограничений
- **Модель данных — концептуальное представление структур, отображающих и отвечающих требованиям бизнеса**

Определение моделирования

- Визуальное моделирование использует стандартное графическое представление
- Коллекция визуальных объектов для описания данных, связей, семантики и ограничений
- Модель данных — концептуальное представление структур, отображающих и отвечающих требованиям бизнеса
- **Модель данных визуализирует природу данных и ее будущую организацию в БД**

Визуальное моделирование

Цели визуального моделирования

Цели визуального моделирования

- **Средство коммуникации:**
 - фиксируем бизнес-логику, анализируем, проектируем свою базу данных



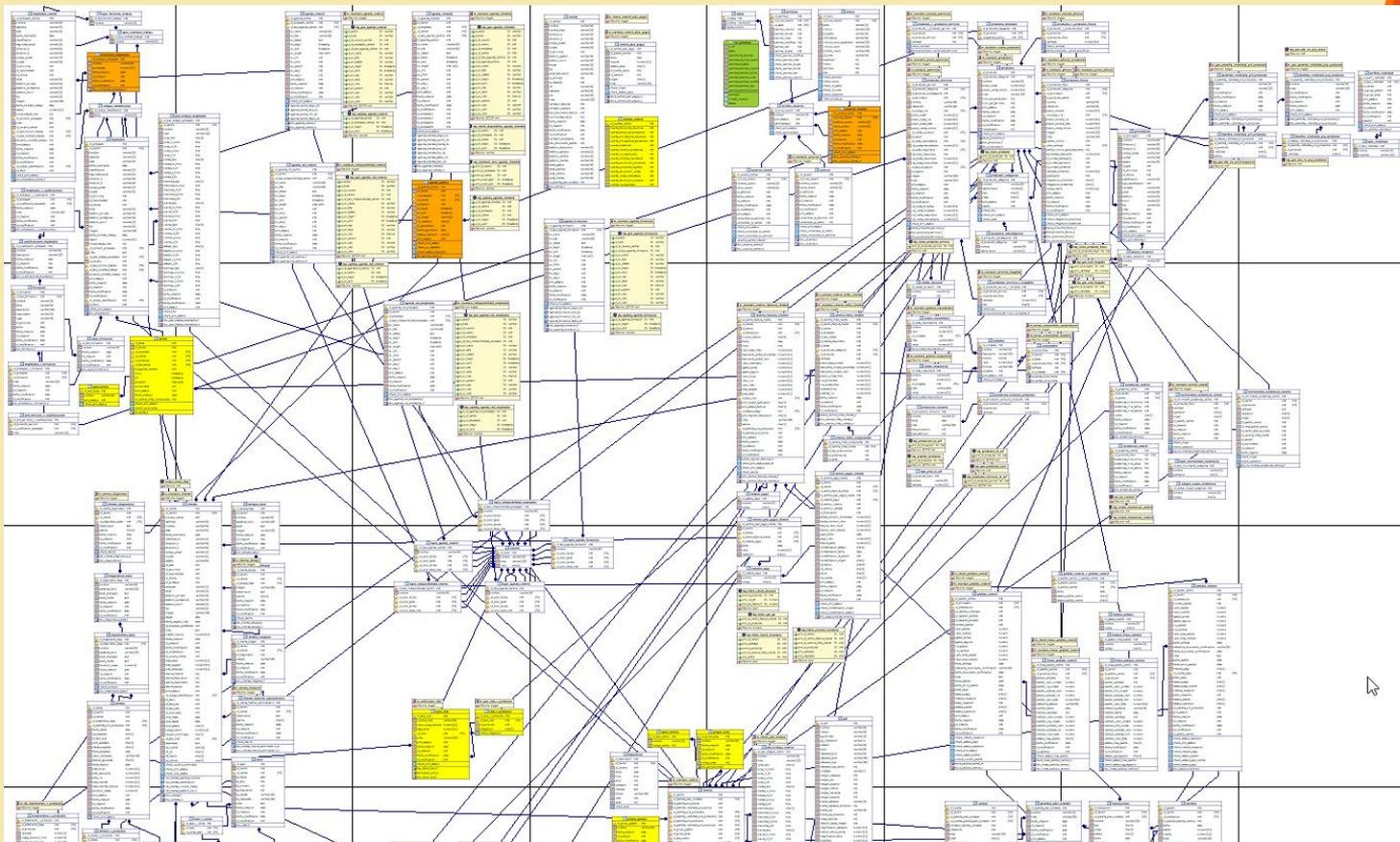
Общение — ключ к успеху

Цели визуального моделирования

- **Средство коммуникации:**
 - фиксируем бизнес-логику, анализируем, проектируем свою базу данных
- **Управление сложностью:**
 - мозг одновременно может удерживать 7 объектов



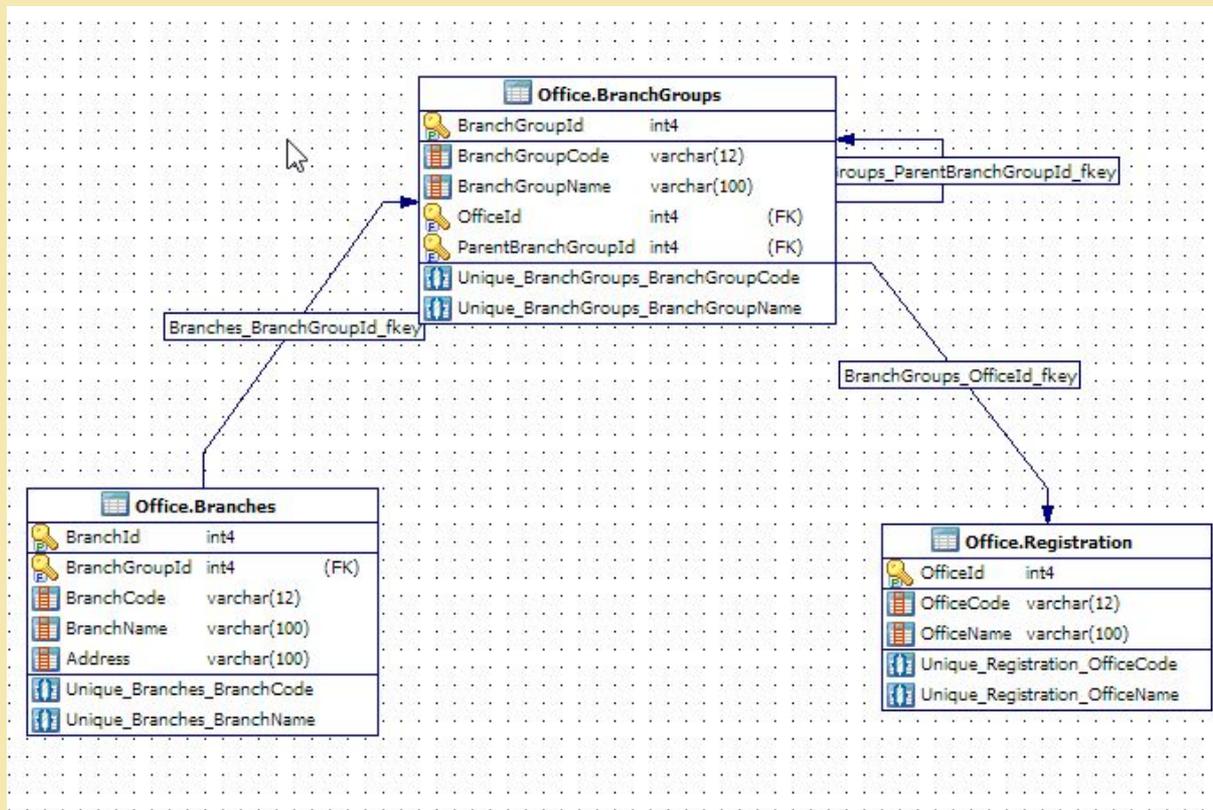
Blame Magnus... them! #blamemagnus



Нет, правда! Я знаю как выглядит боль...

Цели визуального моделирования

- **Средство коммуникации:**
 - фиксируем бизнес-логику, анализируем, проектируем свою базу данных
- **Управление сложностью:**
 - мозг одновременно может удерживать 7 объектов
- **Повторное использование:**
 - Модели могут быть частями чего-то большего или выступать в качестве шаблонов



Некая обыденность данная нам в ощущениях...

Визуальное моделирование

Уровни абстракции диаграмм

Уровни абстракции диаграмм:

Концептуальный

- Из ЧЕГО состоит предметная область
- Делает бизнес или системный аналитик
- Инфо черпается из требований бизнеса
- Представлено в человекочитаемом виде
- Требования СУБД не рассматриваются
- Диаграмма самая простая из всех

Уровни абстракции диаграмм:

Логический

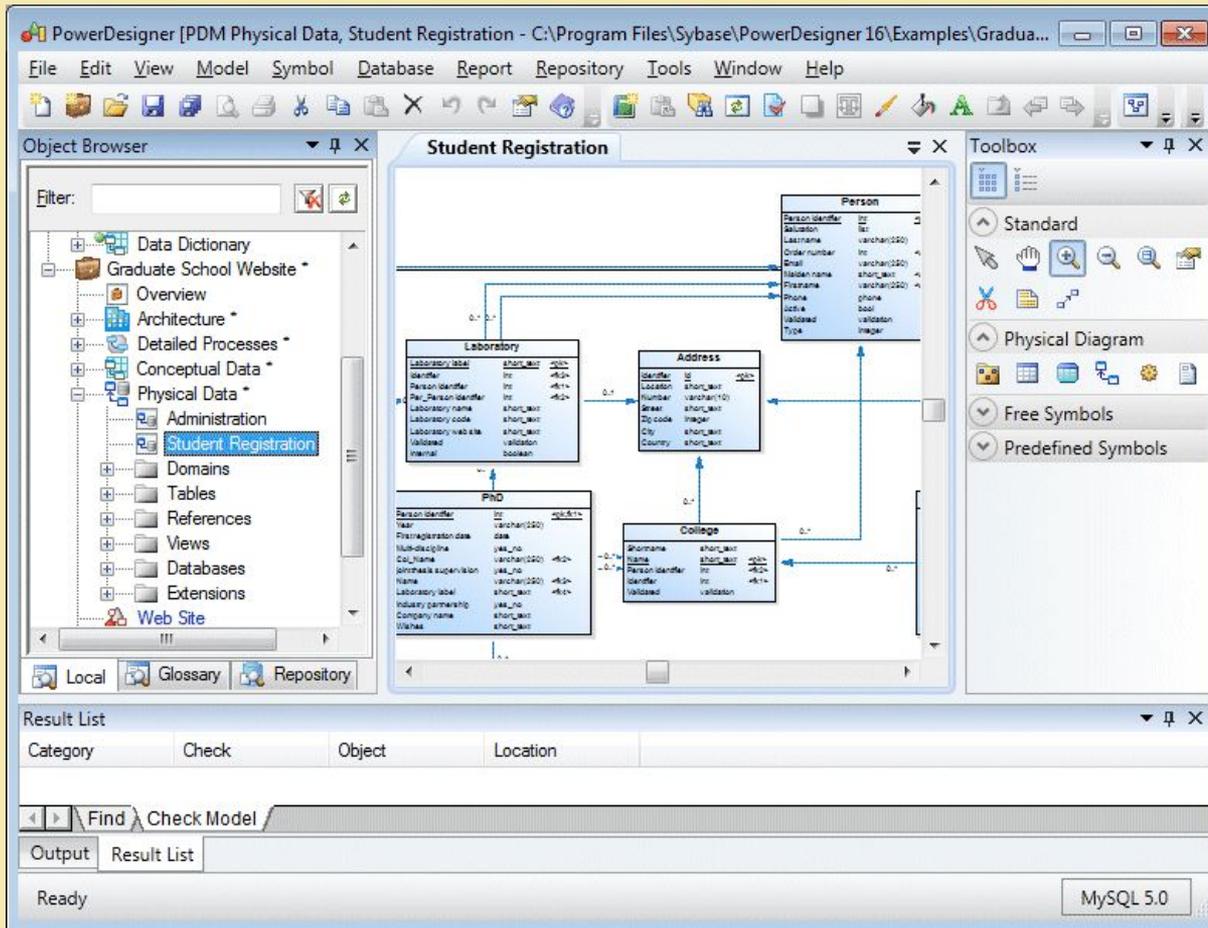
- КАК реализовать систему вне связи с СУБД
- Делает архитектор данных (data architect)
- Инфо также черпается из требований бизнеса
- Типы полей могут иметь типы
- Мы еще не имеем дело с созданием БД

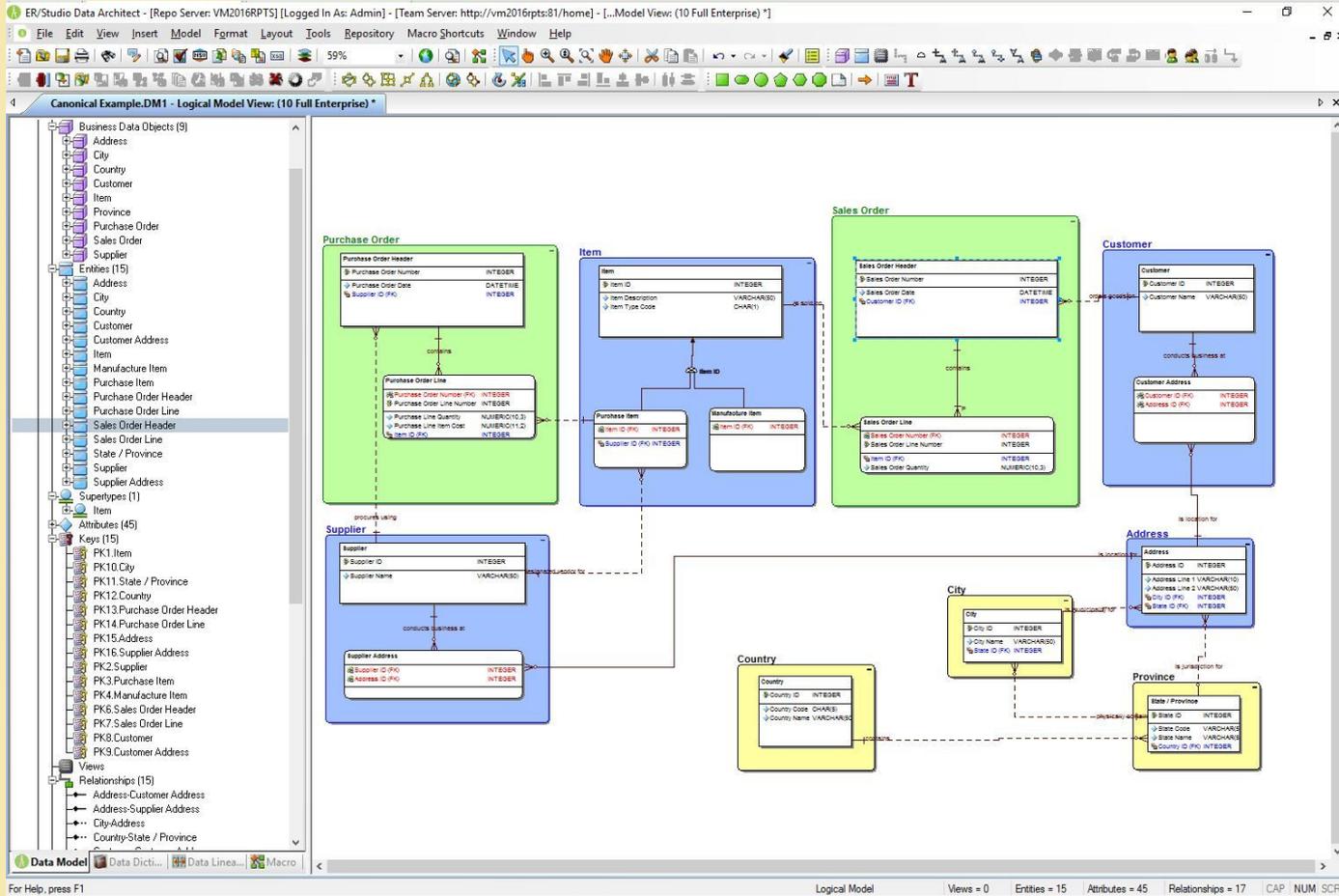
Уровни абстракции диаграмм:

Физический

- КАК реализована схема на конкретной СУБД
- Делает администратор БД
- Представляет фактический проект базы данных
- Следует учитывать ограничения СУБД
- Типы данных указаны для полей таблиц
- Специфические вещи прилагаются (views, rules, triggers etc.)

Программы на рынке





Toad Data Modeler 6.0 Trial [C:\Program Files\Delph\Toad Data Modeler 6.0\Samples\videorental.tpx]

File Edit View Objects Layout Model Tools Macros Settings Window Help

Physical Model Explorer

Videorental

- Work Spaces
- Entities
- Relationships
- Views
- Procedures
- Functions
- Categories
- Defaults
- Domains
- Notes
- Check Constraint Rules
- User Data Types
- User Groups
- Users
- Directories
- Images
- Java
- Materialized Views

All Items x

Customer: Customer ID NN (PK), Name (X), Address, Name (X)

Borrowing: Borrowing ID NN (PK), Customer ID NN (FK), Start Date, End Date, Total Price, Val

Order Record: Order Record ID NN (PK), Customer ID NN (FK), Film ID NN (FK), Order Date

Customer Rating: T.Rate NN (PK), Director NN (FK), Rating

Firm: Firm ID NN (PK), T.IDE NN (AK), Director NN (AK), Production Company, Genre ID NN (FK), Minimal Age, Film ID Episodes (FK)

Example: Example ID NN (PK (AK)), Film ID NN (FK), Medium ID NN (FK), Price per Day (AK)

Customer has firm: name (VARCHAR2(20) CHAR), city (EXPR), c.address.city (CHAR)

Project: Videorental, Model: Videorental, Author: Adam Brand, Company: Quest Software, Version: 3.15.2007.0315, Date of Creation: 3/15/2007 03:15, Last Change: 3/15/2007 03:15

Display notes:

- IE notation
- Customer and Customer Rating to Category
- indexes displayed
- Logical names displayed

Category: Customer Info

Name: All Items DB: Oracle 10g

Verification Log, Loupe, Overview - Videorental - All Items

Verification Log

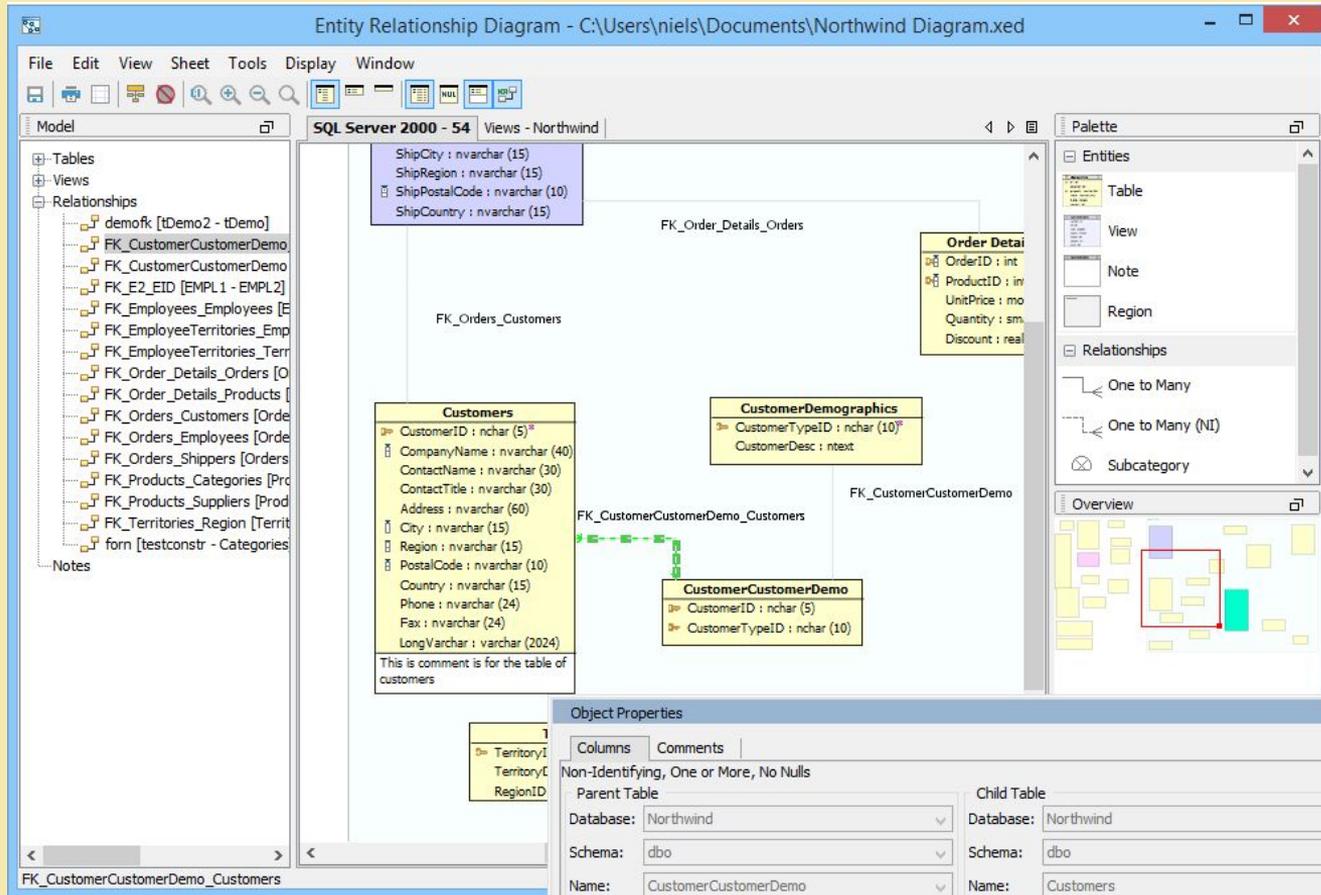
Object	Caption
Entity - T_CUSTOMER	Tablespace Assignment Check
Key - pk_T_CUSTOM...	Tablespace Assignment Check
Index - i_name	Tablespace Assignment Check
Entity - T_BORROWI...	Tablespace Assignment Check

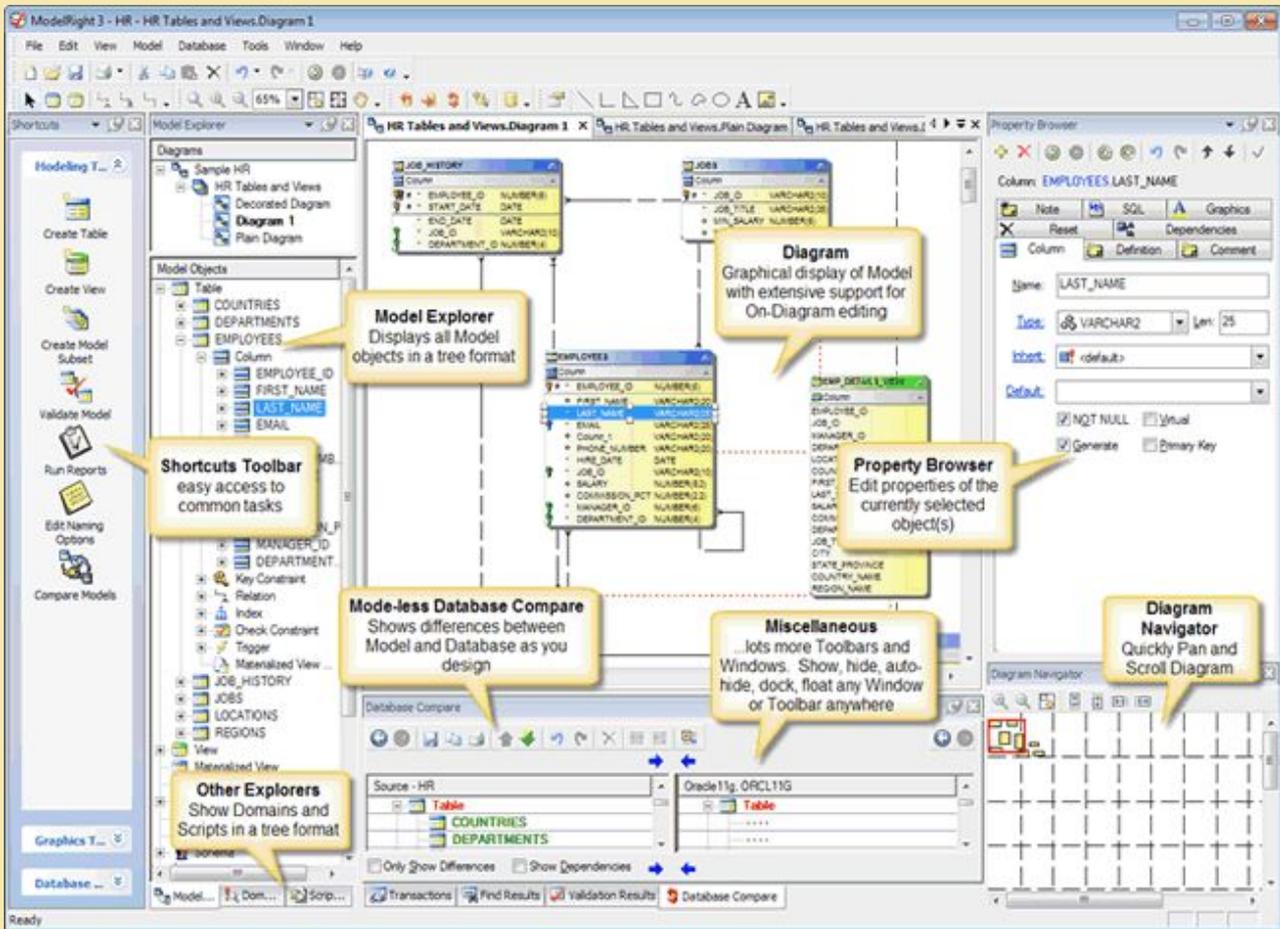
Videorental Errors: 0 Warnings: 1 Hints: 19

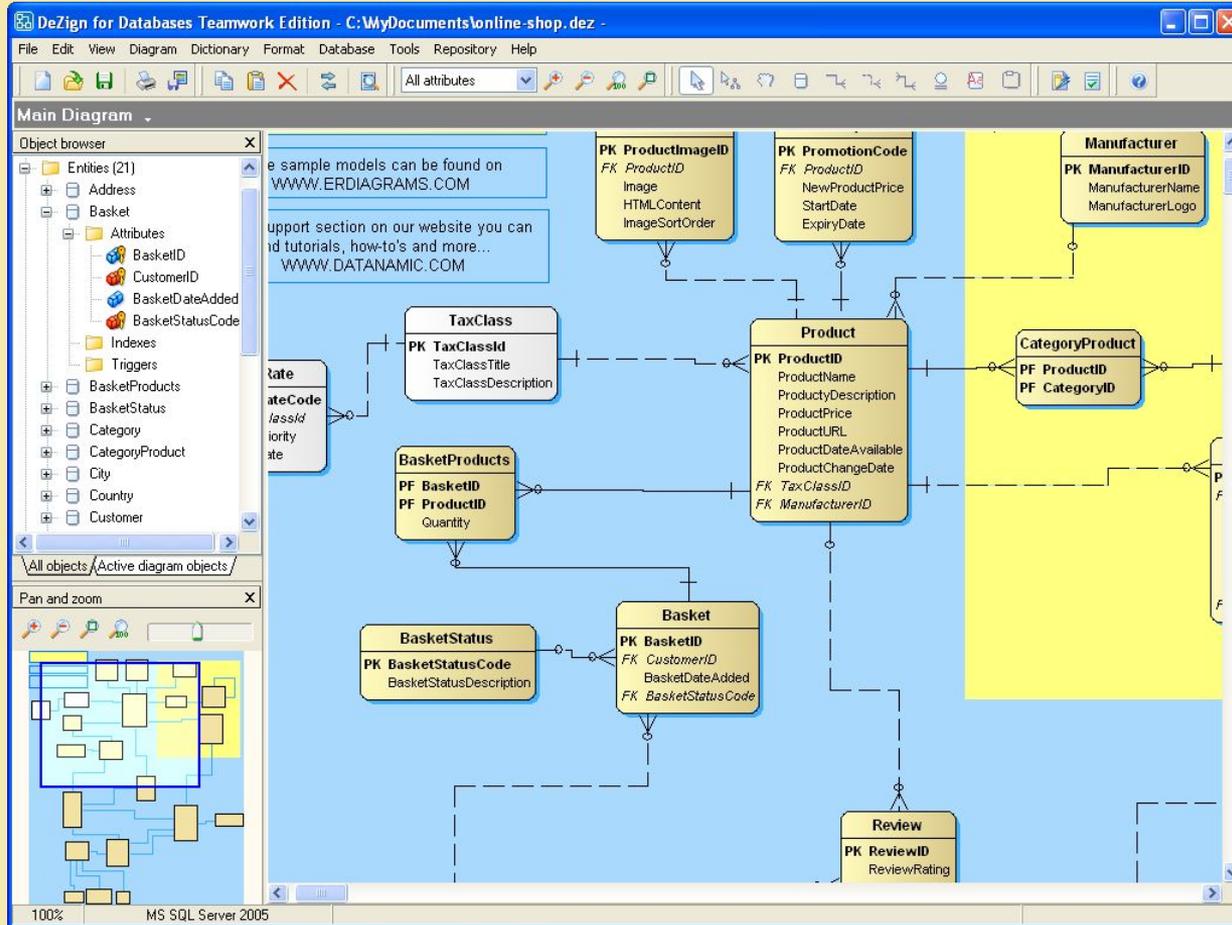
Overview - Videorental - All Items

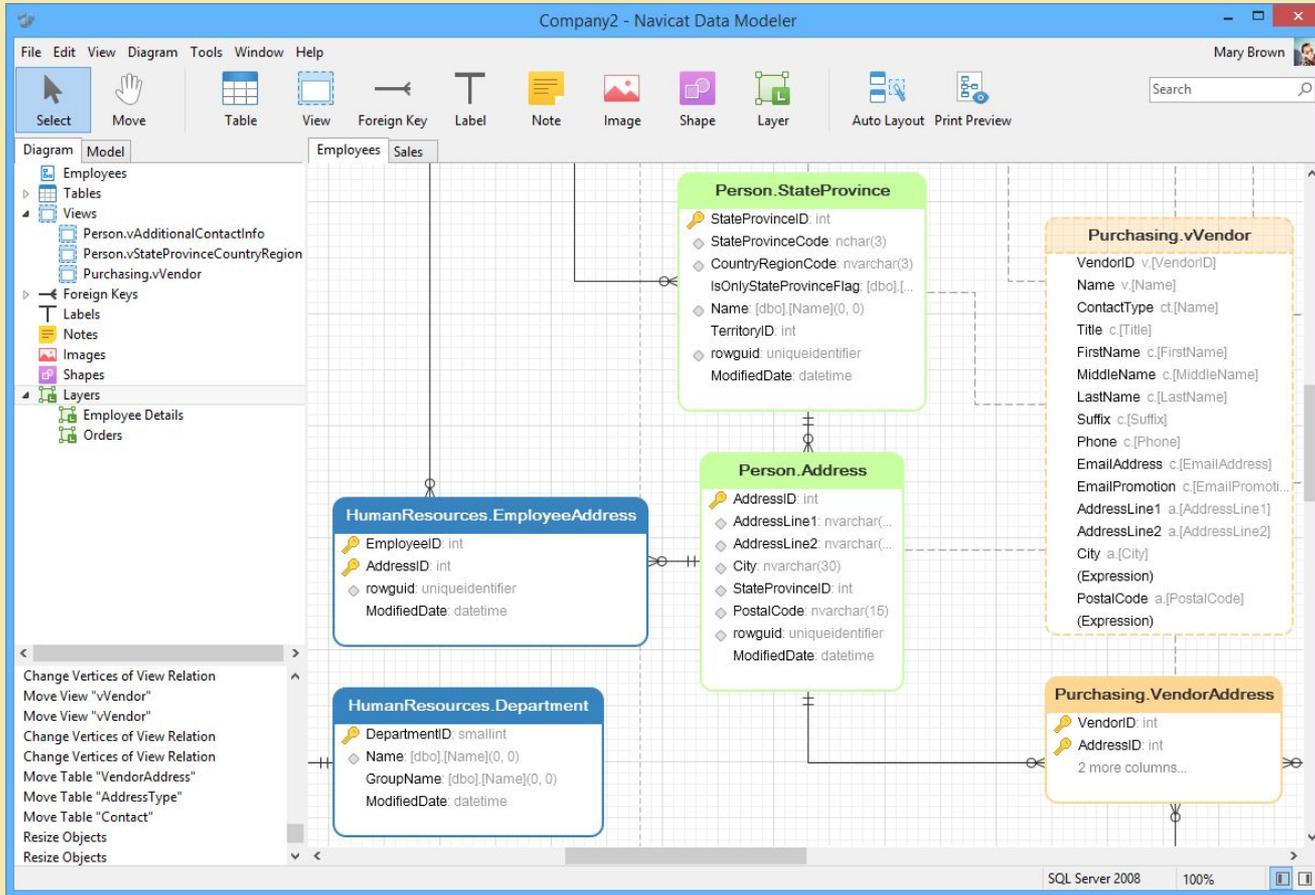
Loupe

Customer I	
name	Varchar2
city	[expr]
title	Varchar2
director	Varchar2









Valentina Studio Pro

Start Page | Schema Editor

Save all changes... | Discard all changes...

Search

Controls

- Comment
- Procedure
- Label
- Table
- Picture
- Type
- View

Customers

- ID Long:1:U
- Company VarChar[50]:1:N
- Last Name VarChar[50]:1:N
- First Name VarChar[50]:1:N
- E-mail Address VarChar[50]:N
- Job Title VarChar[50]:N
- Business Phone VarChar[25]:N
- Home Phone VarChar[25]:N
- Mobile Phone VarChar[25]:N
- Fax Number VarChar[25]:N
- Address Text:N
- City VarChar[50]:1:N
- State/Province VarChar[50]:1:N
- ZIP/Postal Code VarChar[15]:1:N
- Country/Region VarChar[50]:N
- Web Page Text:N
- Notes Text:N
- Attachments BLOB:N

Orders

- Order ID Long:1:U
- Employee ID Long:1:N
- Customer ID Long:1:N
- Order Date DateTime:N
- Shipped Date DateTime:N
- Shipper ID Long:1:N
- Ship Name VarChar[50]:N
- Ship Address Text:N
- Ship City VarChar[50]:N
- Ship State/Province VarChar[50]:N
- Ship ZIP/Postal Code VarChar[50]:N
- Ship Country/Region VarChar[50]:N
- Shipping Fee Double:N
- Taxes Double:N
- Payment Type VarChar[50]:N
- Paid Date DateTime:N
- Notes Text:N
- Tax Rate Double:N
- Tax Status Short:1:N
- Status ID Short:1:N

Orders Status

- Status ID Short:1:U
- Status Name VarChar[50]:N

Order Details

- ID Long:1:U
- Order ID Long:1
- Product ID Long:1:N
- Quantity Double
- Unit Price Double:N
- Discount Double
- Status ID Long:1:N
- Date Allocated DateTime:N
- Purchase Order ID Long:N
- Inventory ID Long:N

Properties

Diagram

name: d1
gamma: Light
style: Rounded
style_links: Default

Control

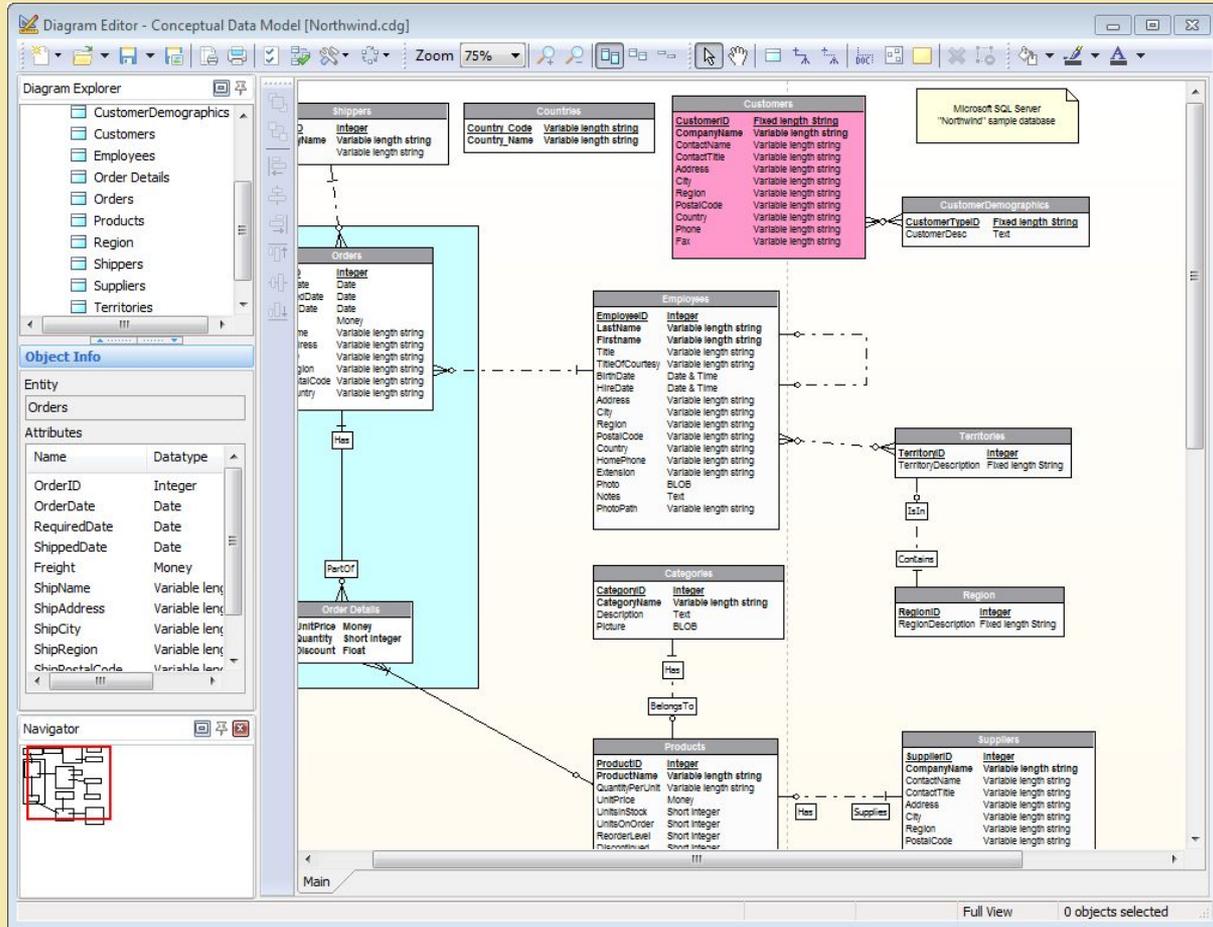
Navigator

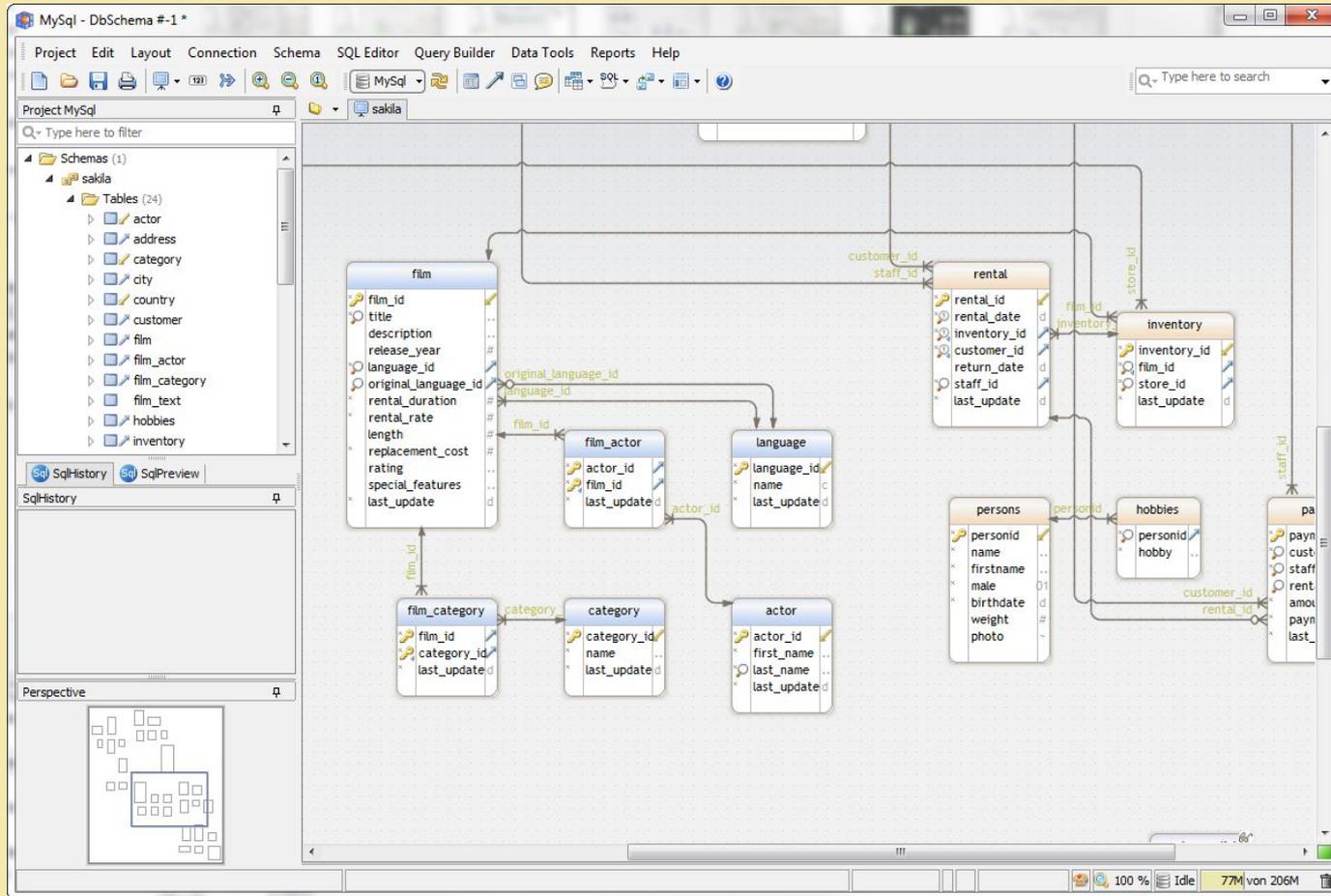
History

Orders

Table	Fields (20)	Checks (0)	Indexes (6)	Links (8)	Methods (0)	Primary Key	Properties (1)	Triggers (0)	Uniques (0)
	Name			Type	N	U			
Order ID	Order ID			Long	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	default_value		
Employee ID	Employee ID			Long	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	comment none		
Customer ID	Customer ID			Long	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Order Date	Order Date			DateTime	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Shipped Date	Shipped Date			DateTime	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Shipper ID	Shipper ID			Long	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Ship Name	Ship Name			VarChar	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Ship Address	Ship Address			Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Ship City	Ship City			VarChar	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Ship State/Province	Ship State/Province			VarChar	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Ship ZIP/Postal Code	Ship ZIP/Postal Code			VarChar	<input checked="" type="checkbox"/>	<input type="checkbox"/>			

Ready





Vertabelo

My models Share requests My account Invite people Help

Jarosław Bład Log out

Simple shop (Edit mode)

MODEL STRUCTURE

Model

- Tables
 - client
 - order
 - order_item
 - product
 - product_category
- References
 - client_order
 - order_order_item
 - product_category_product
 - product_category_product_cat...
 - product_order_item
- Sequences
- Text notes

PROBLEMS

This is example of simple database model.
You can change, enhance or delete it.

client

id	int	N PK
full_name	varchar(255)	
email	varchar(255)	

order

id	int	PK
order_no	char(12)	
client_id	int	FK

order_item

id	int	PK
order_id	int	FK
product_id	int	FK
amount	int	

TABLE PROPERTIES SQL Preview

Primary data

Name: client

Description:

Columns + Add column

Name	Type	N	PK
id	int		<input checked="" type="checkbox"/>
full_name	varchar(255)		<input type="checkbox"/>
email	varchar(255)		<input type="checkbox"/>

Alternate keys

Indexes

Checks

TABLE PROBLEMS

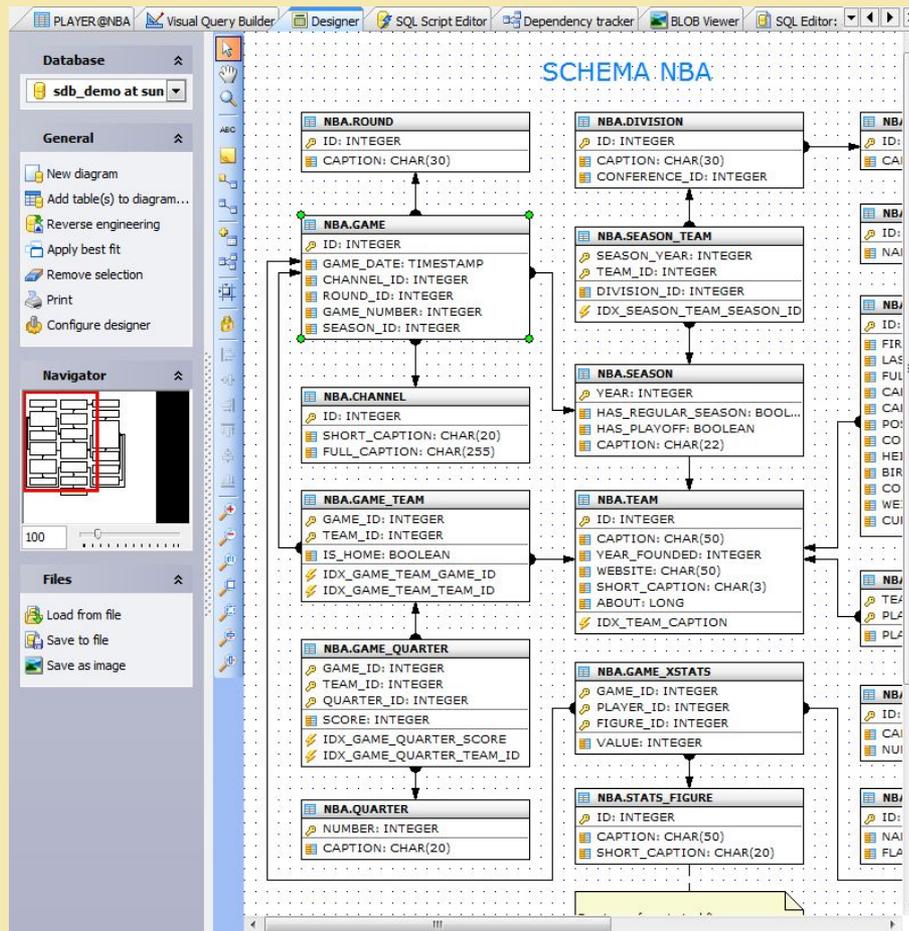
Errors (1)

id Primary key column can't be null.

Warnings (0)

Hints (0)

Vertabelo, от \$111 (\$9/месяц, до 5 ДБ, до 25 таблиц)



DbWrench Database Design and Synchronization

File Edit View Database Diagram Tools Window Help

60 % Connection: ...

DbExplorer Script Explorer Database Compare Software License For...

Sakila

Schemas (1)

public

Tables (17)

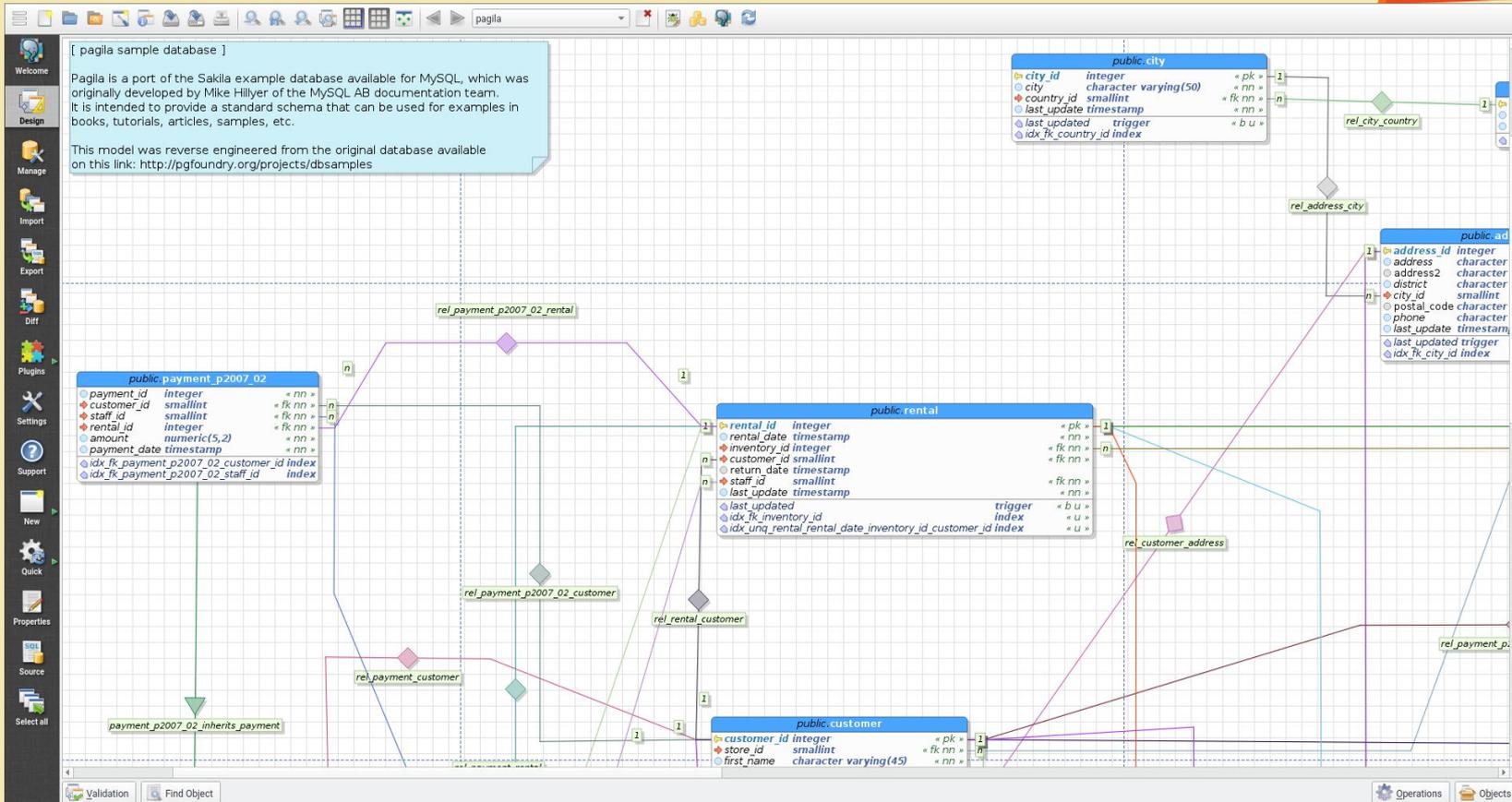
- actor
- address
- category
- city
- country
- customer
- film
- film_actor
- film_category
- film_text
- inventory
- language
- payment
- rental
- staff

Database Con... Navigator

Output Object Details: Sakila Query Results Sakila [Sakila_screenshot.xml]

Edit Designs from Diagrams
Edit Columns, Data Types and Other Entities Directly from Diagrams

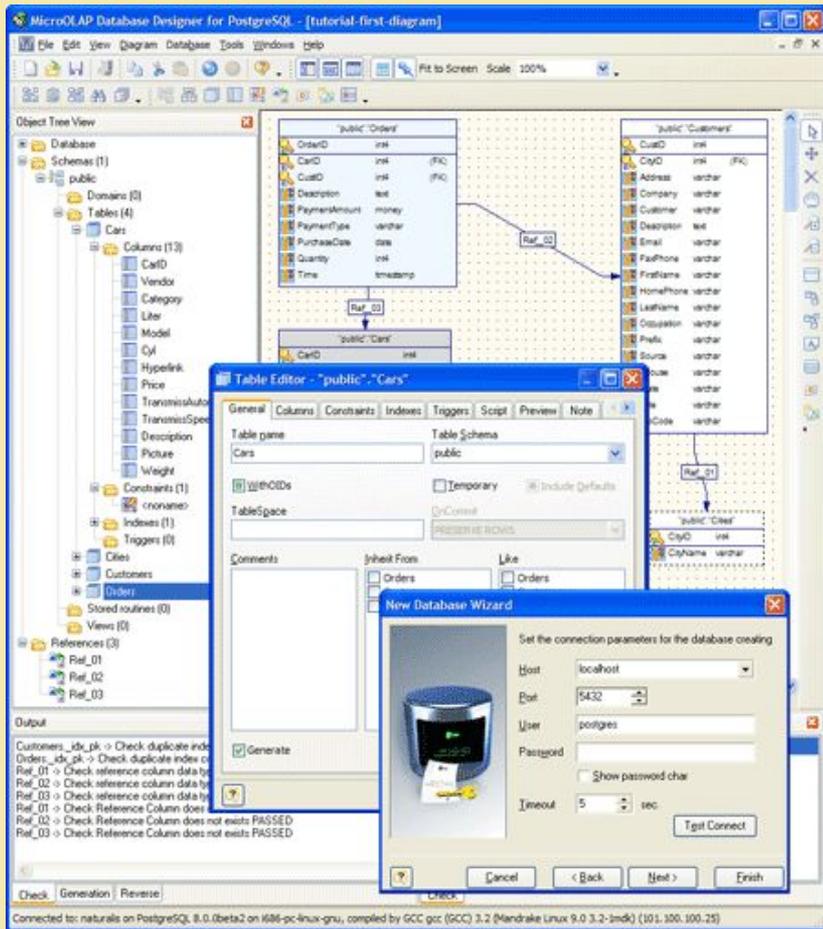
The screenshot shows the DbWrench interface with a database diagram for the Sakila database. The diagram consists of several tables and their relationships. A callout box with a blue arrow points to the 'actor' table, indicating that users can edit columns, data types, and other entities directly from the diagram. The interface includes a menu bar, a toolbar, a left-hand pane for navigation, and a main workspace for the diagram. The bottom status bar shows the current database and file name.



PgMDD

расшифровывается как

MicroOLAP Database Designer for PostgreSQL



Визуальные образы объектов

Используются в PgMDD

Таблицы

- В виде прямоугольников
- Атрибуты перечислены в строках
- Свойства атрибутов описаны в колонках
- Комментарии могут отображаться внизу

customer			
	customer_id	int4	(NN)
	store_id	int2	<i1> (NN) (FK)
	first_name	varchar(45)	(NN)
	last_name	varchar(45)	<i2> (NN)
	email	varchar(50)	
	address_id	int2	<i0> (NN) (FK)
	activebool	bool	(NN)
	create_date	date	(NN)
	last_update	timestamp	
	active	int4	
	chkCreatedUpdated		
	idx_fk_address_id		
	idx_fk_store_id		
	idx_last_name		
	Rule0		
	Rule1		
	last_updated		
	Customer information		

rental_customer_id_fkey

rental			
	rental_id	int4	(NN)
	rental_date	timestamp	<i1> (NN)
	inventory_id	int4	<i0,i1> (NN) (FK)
	customer_id	int2	<i1> (NN) (FK)
	return_date	timestamp	
	staff_id	int2	(NN) (FK)
	last_update	timestamp	(NN)
	last_year	year	(D)
	idx_fk_inventory_id		
	idx_unq_rental_rental_date_inventory_id_customer_id		
	last_updated		

Связи (внешние ключи)

- В виде линий, связывающих таблицы
- Могут иметь вид стрелки (нотация Бахмана)
- Могут иметь вид “вороньей лапки”
- Ярлычки с дополнительной информацией
- Возможность добавления пользовательских точек

customer			
	customer_id	int4	(NN)
	store_id	int2	<i1> (NN) (FK)
	first_name	varchar(45)	(NN)
	last_name	varchar(45)	<i2> (NN)
	email	varchar(50)	
	address_id	int2	<i0> (NN) (FK)
	activebool	bool	(NN)
	create_date	date	(NN)
	last_update	timestamp	
	active	int4	
	chkCreatedUpdated		
	idx_fk_address_id		
	idx_fk_store_id		
	idx_last_name		
	Rule0		
	Rule1		
	last_updated		
	Customer information		

|| [0..*]

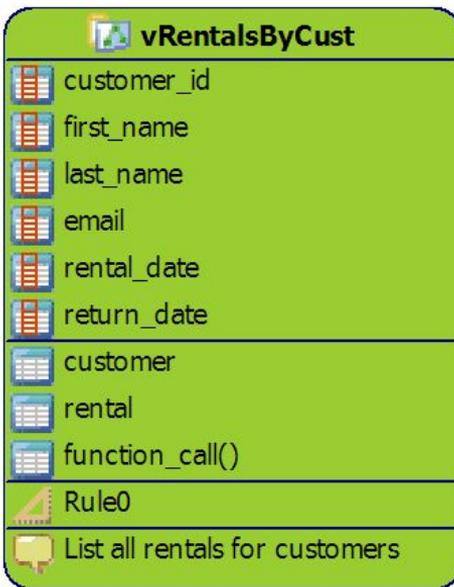
rental_customer_id_fkey(customer_id=customer_id):

rental			
	rental_id	int4	(NN)
	rental_date	timestamp	<i1> (NN)
	inventory_id	int4	<i0,i1> (NN) (FK)
	customer_id	int2	<i1> (NN) (FK)
	return_date	timestamp	
	staff_id	int2	(NN) (FK)
	last_update	timestamp	(NN)
	last_year	year	(D)
	idx_fk_inventory_id		
	idx_unq_rental_rental_date_inventory_id_customer_id		
	last_updated		



Представления

- В виде скругленных прямоугольников
- Атрибуты берутся из разобранного SQL
- Комментарии могут отображаться внизу



Хранимые процедуры

- В виде прямоугольников
- Параметры перечислены в строках
- Отображается язык функции
- Могут отображаться комментарии

 **example_area**

 minArea IN float8
 name OUT char
 capital OUT char
 area OUT float8
 population OUT float8

 Language: plpgsql

 Returns: SETOF record

 Returns countries with area bigger then input

 **isleapyear**

 year_number IN int4

 Language: plpgsql

 Returns: bool

 **checkSum**

 Language: PLpgSQL

 Returns: trigger

 Check for some integer

Заметки и комментарии

- Специальные объекты модели без связи с БД
- Используются для отображения дополнительных данных

Physical Data Diagram

Project: Pagila

Diagram: Paglia Light

Company: pgfoundry.org

Author: Christopher Kings-Lynne, Robert Treat, Mike Hillyer

Date: 13.07.2016

Copyright: BSD license

Version: 0.44

Pagila is a port of the Sakila example database available for MySQL, which was originally developed by Mike Hillyer of the MySQL AB documentation team. It is intended to provide a standard schema that can be used for examples in books, tutorials, articles, samples, etc.

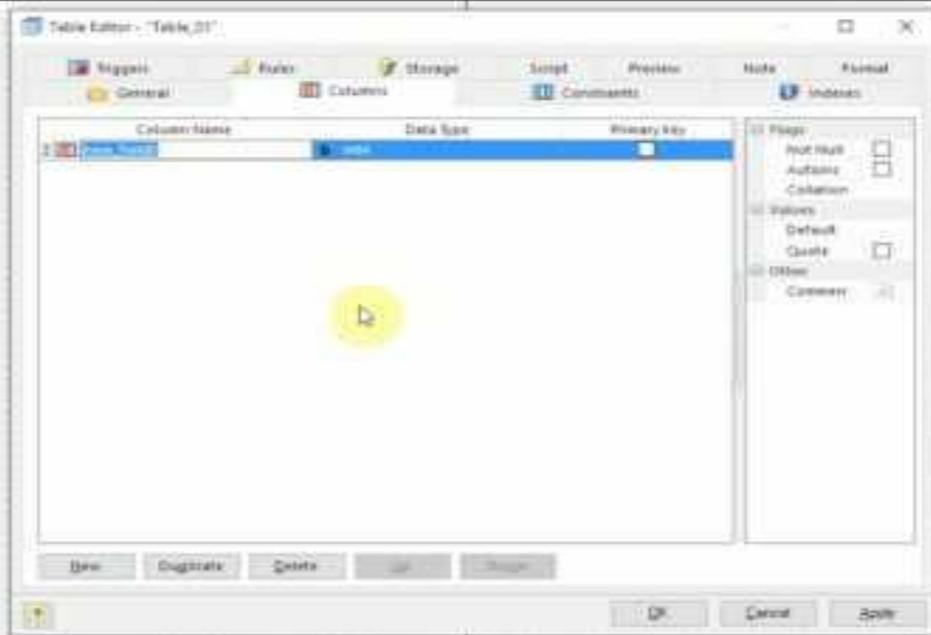
The schema and data for the Sakila database were made available under the BSD license which can be found at <http://www.opensource.org/licenses/bsd-license.php>. The pagila database is made available under this license as well.

Основные сценарии

Создание модели и
генерация базы данных

Создание и редактирование таблиц

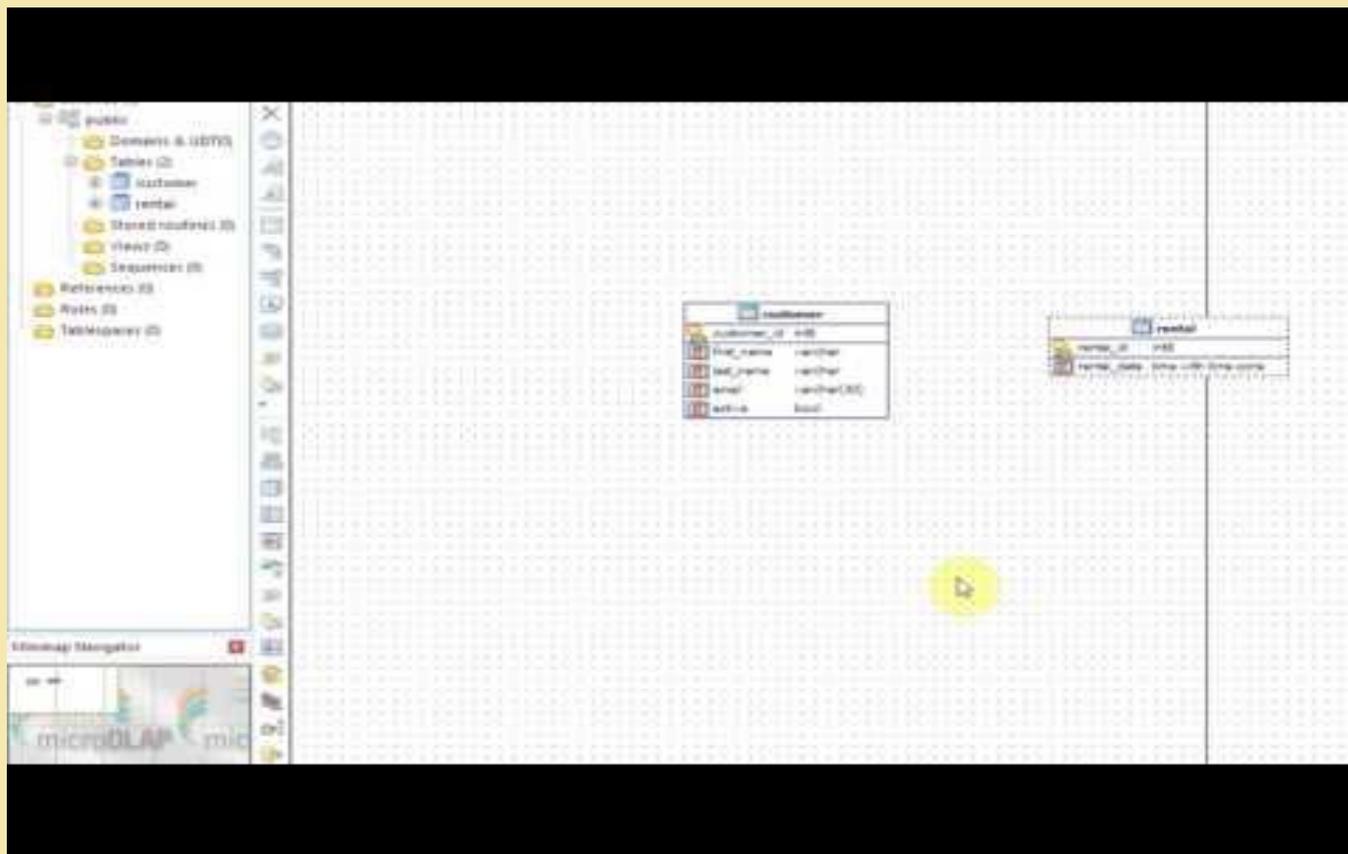
- Определяем атрибуты (колонки, поля)
- Определяем тип данных
- Назначаем ключи
- Вводим ограничения
- Добавляем индексы, правила, триггеры
- Форматируем внешний вид



Как создать таблицу: https://youtu.be/kO3Xd_9nEjo

Определяем внешние ключи

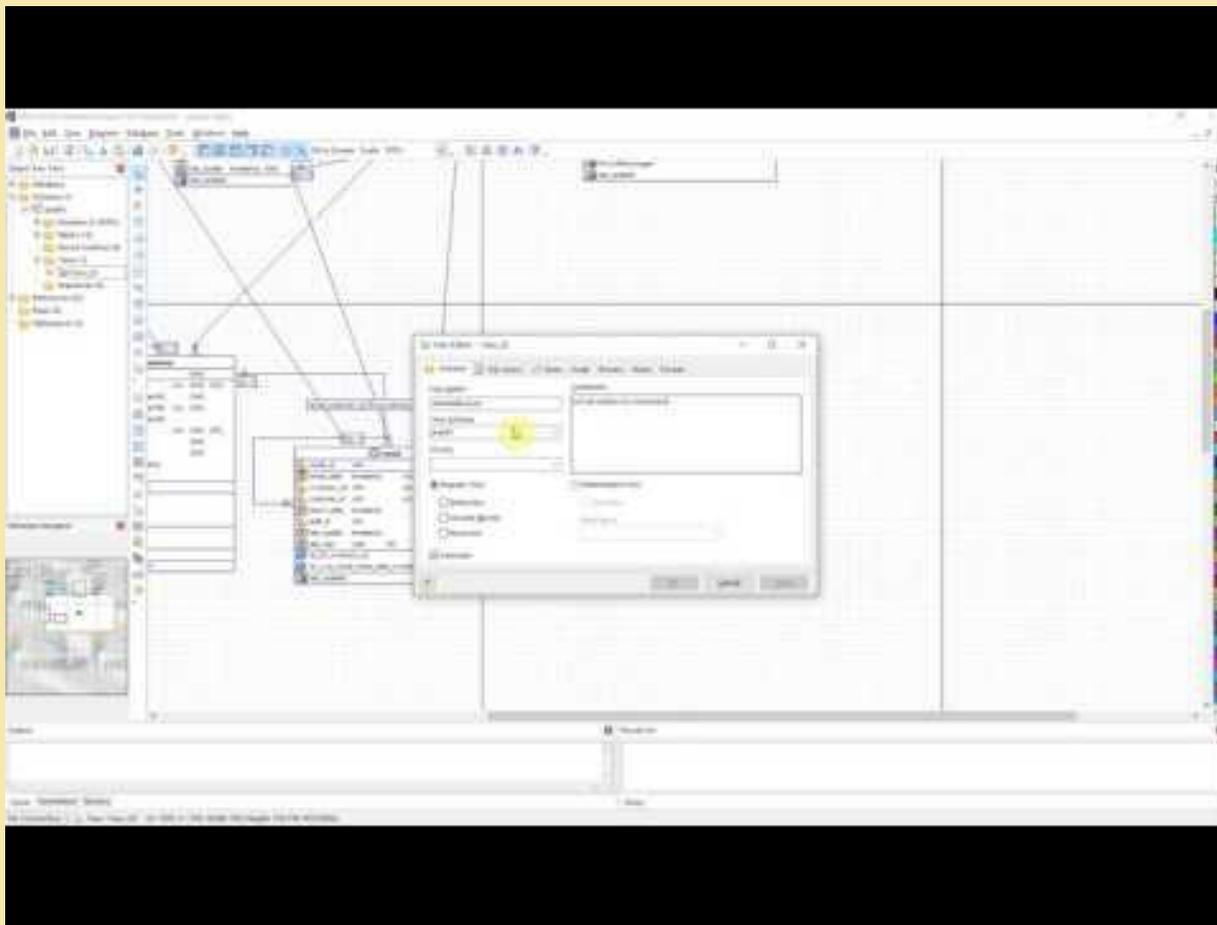
- Определяем мощность (cardinality)
- Определяем ключ или уникальное ограничение для связи
- Управляем действиями для сохранения связности



Как создать связь: <https://youtu.be/wRvjBf5HVZI>

Создание представления

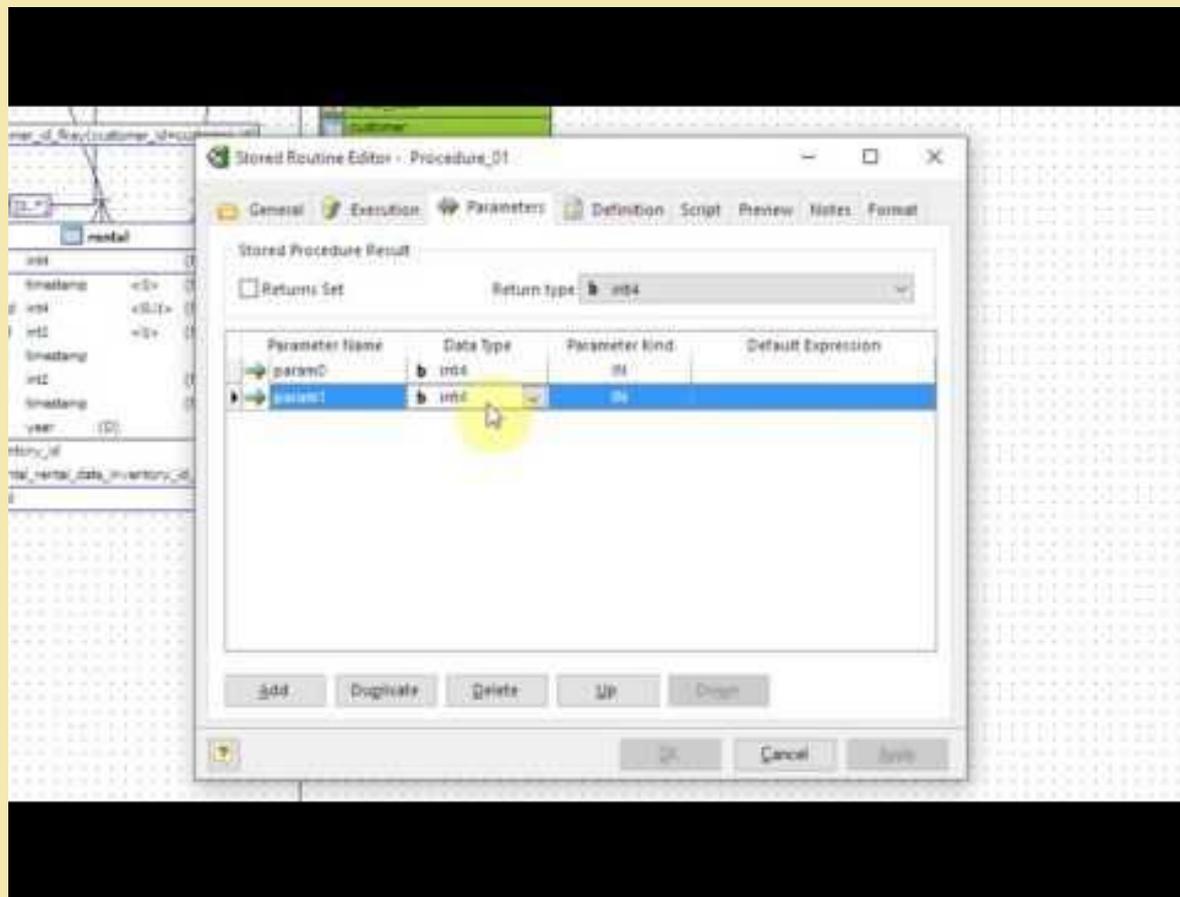
- SQL
- Правила
- Форматирование внешнего вида



Как создать представление: <https://youtu.be/saKDIUgal0s>

Создание хранимых процедур

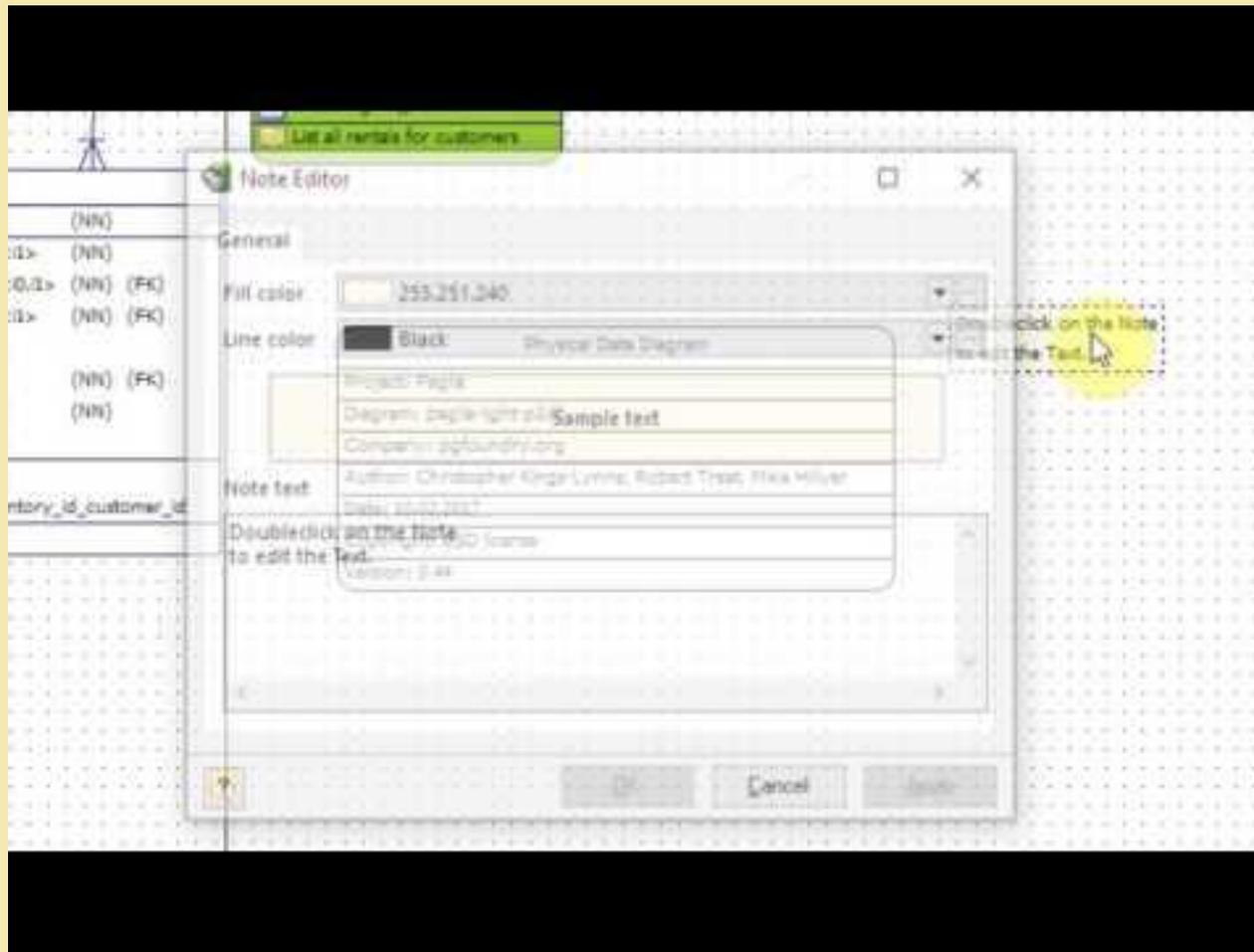
- Параметры
- Исходный код



Создание хранимых процедур: <https://youtu.be/lZROhgMQaus>

Заметки и комментарии

- Заметки
- Комментарии

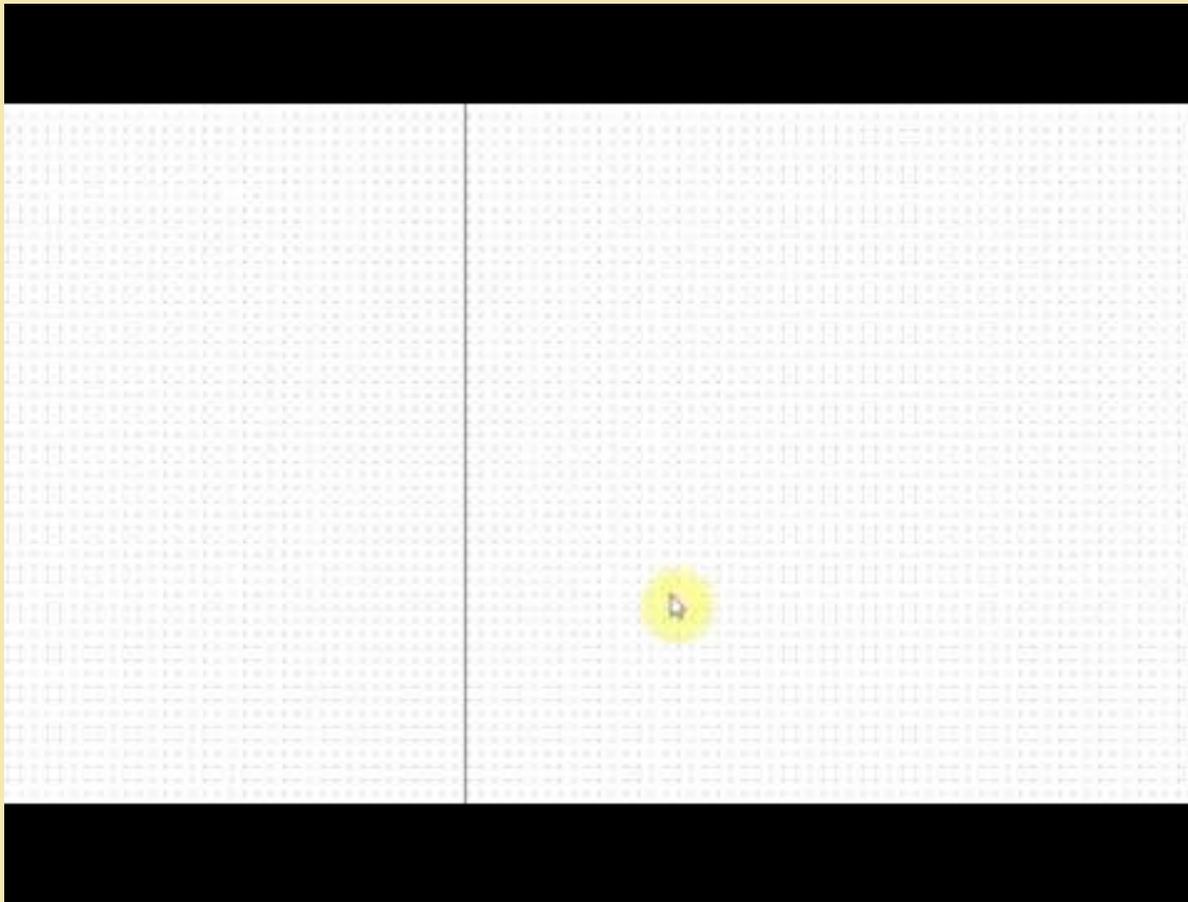


Основные сценарии

Реверс-инжиниринг
(импорт) базы данных

Реверс-инжиниринг PostgreSQL

- Соединение с сервером
- Выбор объектов для реверса



Другие виды реверс-инжиниринга

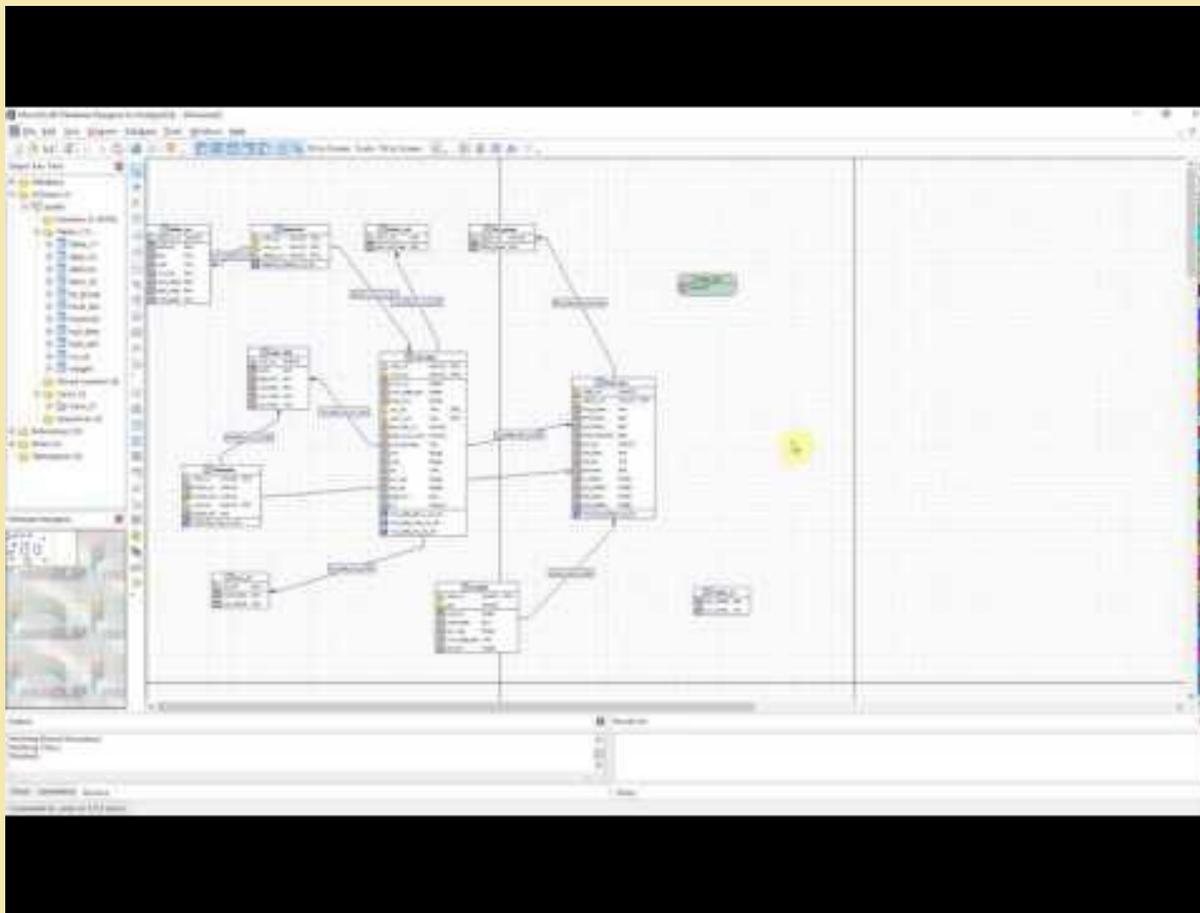
- Импорт из скрипта SQL
- Импорт из БД MS Access
- Universal Reverse Engineering (ODBC)

Основные сценарии

Изменение базы данных

Изменение базы данных

- К каким объектам применить изменения
- Что сделать с измененными объектами
- Генерация скрипта модификации для выполнения



Основные сценарии

Документирование

Документирование

- Экспорт модели как изображения
 - .jpg, .png, .gif, .bmp
 - .emf
 - один файл на страницу
- Печать и экспорт в .pdf
- Создание отчета в .html

Print Preview

File View Insert Format Go Tools

50%

Margins: Left: 0,5 in Top: 0,5 in Right: 0,5 in Bottom: 0,5 in Header: 0,2 in Footer: 0,2 in

Page: 1 of 6 Pages Paper Size: 8,3 in x 11,7 in Status: One or more margins are set outside the printable area of the page



Голубь Павел
MicroOLAP Technologies
pavel@microolap.com



pasha_golub

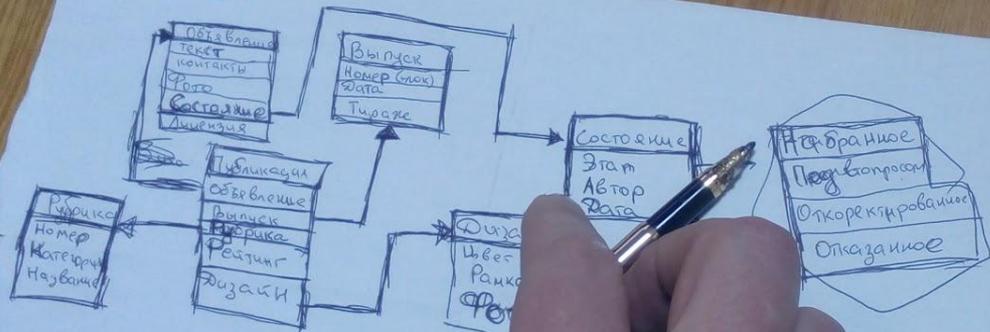
**PGDAY'
RUSSIA 17**

**КОНФЕРЕНЦИЯ
ПО БАЗАМ ДАННЫХ**

Вопросы?



Купон на скидку на все продукты: **pgday2017**



то обыв
 рциям
 а
 ке
 ке

