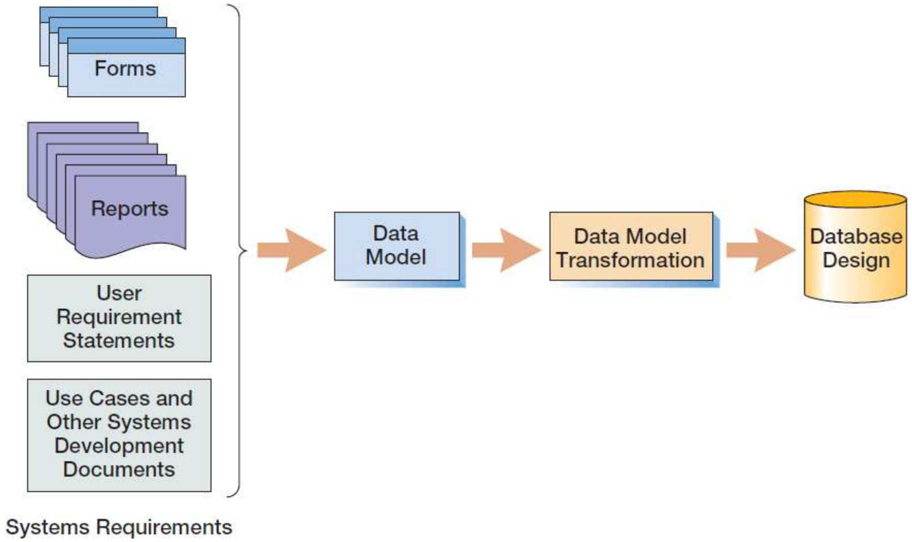
Database Systems

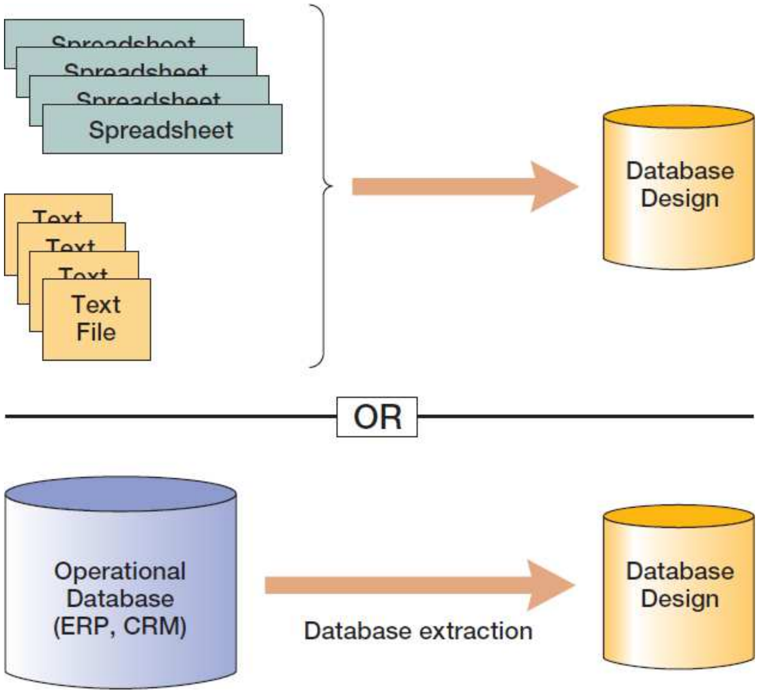
Database Modeling

Database Design Approaches

Database Design from New Systems Development


Forms
Reports
User Requirement Statements
Use Cases and Other Systems Development Documents
Systems Requirements
Data Model
Data Model Transformation
Database Design

Database Design from Existing Data



Database Modeling : Step-by-step

Database Modeling

- The process of producing a detailed data model to meet an end user's requirements
  http://www.prowareness.com/blog/database-design-conceptual-design-logical-design-physical-design/
- Qualities of good database design:
  - Reflects real-world structure of the problem
  - Can represent all expected data over time
  - Avoids redundancy and ensures consistency
  - Provides efficient access to data
  - Supports the maintenance of data integrity over time
  - Supports the needs of the database users

3 Phases of Database Design

- Conceptual database design
  - Constructing a data model for each view of the real world problem
    - Constructing the ER Model
    - Checking it for redundancy
    - Validating it against user transactions to ensure all scenarios are supported
- Logical database design
- Physical database design

Step 0 of Conceptual Database Design

- Understanding the real world structure of the problem!

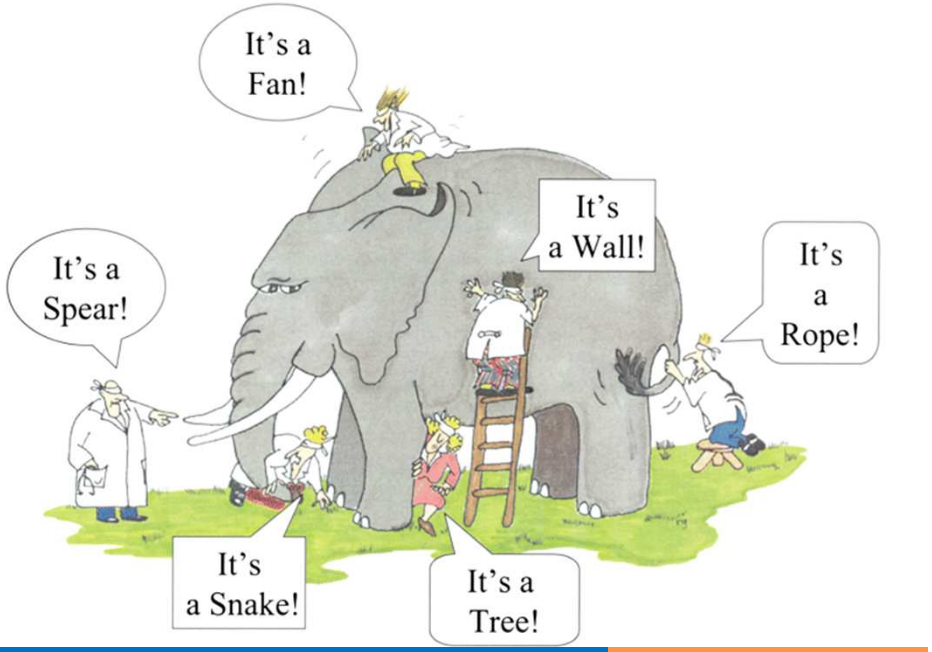
Step 0: Know your customer

- In order ultimately to design databases to support an organization, one should have:
  - a clear understanding of how the organization is structured
  - how it functions
  - understand its components, what they do and how they relate to each other.
  - There must be a way of recording (diagramming) the business
- This is the principle of **DATA MODELING**.
- What happens if we don't try to know our customer?! → see next slide!)
- **Questions to begin with:**
  - Who are the stakeholders?
    - More on this: Rational Unified Process (RUP) and software engineering
  - What data is important to them?
  - What tasks do they have to do with the data?

پیل اندر خانهٔ تاریک بود   عرضه را آورده بودندش هُنود
از برای دیدنش مردم بسی   اندر آن ظلمت همی شد هر کسی
دیدنش با چشم چون ممکن نبود   اندر آن تاریکی اش کف می بسود
آن یکی را کف به خرطوم اوفتاد   گفت همچون ناودان است این نهاد
آن یکی را دست بر گوشش رسید   آن برو چون بادبیزن شد پدید
آن یکی بر پشت او بنهاد دست   گفت خود این پیل چون تختی بدست
از نظرگه، گفتشان شد مختلف   آن یکی دالَش لقب داد این الف

در کف هریک اگر شمعی بدی   اختلاف از گفتشان بیرون شدی
چشم حس همچون کف دست است و بس   نیست کف را بر همه او دسترس
چشم دریا دیگرست و کف دگر   کف بهل وز دیده دریا نگر
جنبش کف‌ها ز دریا روز و شب   کف همی بینی و دریانی عجب
ما چو کشتی‌ها به هم بر می‌زنیم   تیره چشمیم و در آب روشنیم
ای تو در کشتی تن رفته به خواب   آب را دیدی؟ نگر در آبِ آب
آب را آبی است، کو می رانَدَش   روح را روحی است، کو می خوانَدَش

- Managing this complexity:
  - Layered database design
  - Different UML diagrams



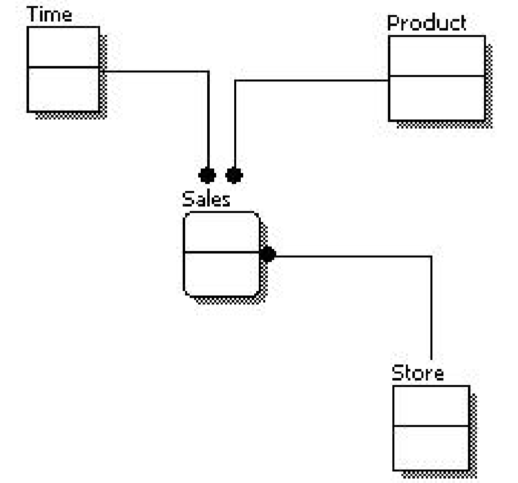
Database Design Levels

Feature	Conceptual	Logical	Physical
Entity Names	√	√	
Entity Relationships	√	√	
Attributes		√	
Primary Keys		√	√
Foreign Keys		√	√
Table Names			√
Column Names			√
Column Data Types			√

- There are three levels of data modeling
  - ✓ Conceptual
  - ✓ Logical
  - ✓ Physical

Database Design Levels

Conceptual



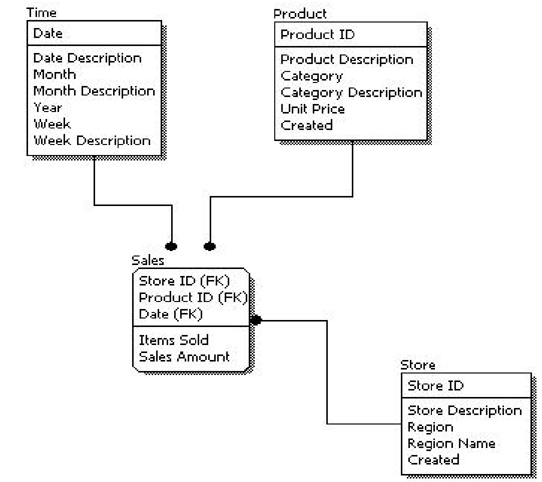
Time

Product

Sales

Store

Logical



Time

Date

Date Description
Month
Month Description
Year
Week
Week Description

Product

Product ID

Product Description
Category
Category Description
Unit Price
Created

Sales

Store ID (FK)
Product ID (FK)
Date (FK)

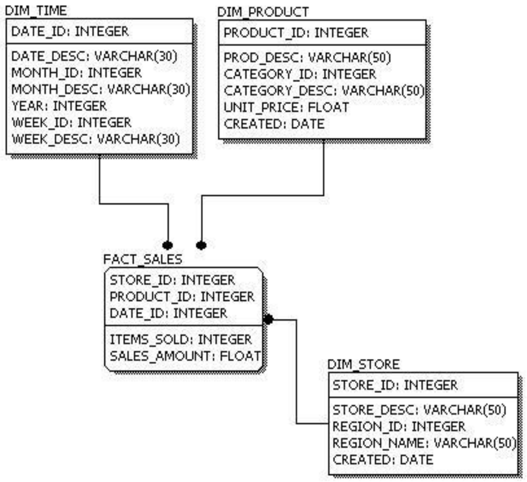
Items Sold
Sales Amount

Store

Store ID

Store Description
Region
Region Name
Created

Physical



DIM_TIME

DATE_ID: INTEGER

DATE_DESC: VARCHAR(30)
MONTH_ID: INTEGER
MONTH_DESC: VARCHAR(30)
YEAR: INTEGER
WEEK_ID: INTEGER
WEEK_DESC: VARCHAR(30)

DIM_PRODUCT

PRODUCT_ID: INTEGER

PROD_DESC: VARCHAR(50)
CATEGORY_ID: INTEGER
CATEGORY_DESC: VARCHAR(50)
UNIT_PRICE: FLOAT
CREATED: DATE

FACT_SALES

STORE_ID: INTEGER
PRODUCT_ID: INTEGER
DATE_ID: INTEGER

ITEMS_SOLD: INTEGER
SALES_AMOUNT: FLOAT

DIM_STORE

STORE_ID: INTEGER

STORE_DESC: VARCHAR(50)
REGION_ID: INTEGER
REGION_NAME: VARCHAR(50)
CREATED: DATE

Data Modeling

The analysis of data objects and their relationships to other data objects.

- Types of data models:

    1. Conceptual: describes **WHAT** the system contains
    2. Logical: describes **HOW** the system will be implemented, regardless of the DBMS
    3. Physical: describes **HOW** the system will be implemented using a specific DBMS

Types of Data Models

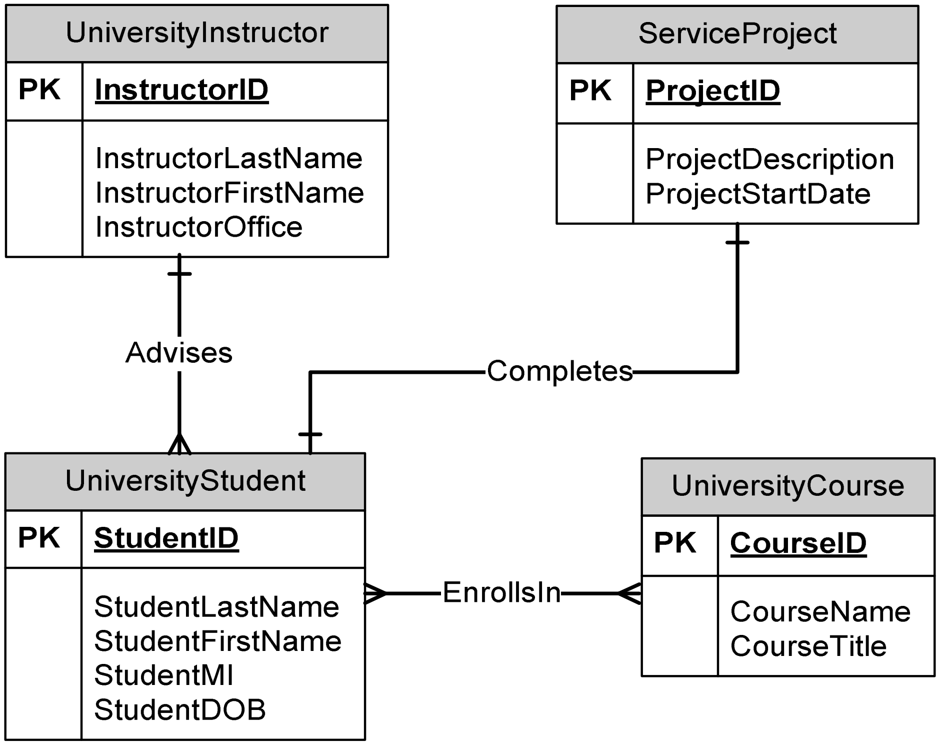
- Entity-Relationship (E-R) Models
- The most common method for database modelling
  - Only addresses data and relationships
  - Classic, simplest
    - Best for deriving a sound table design
  - Basis for most other modeling approaches

- Also : UML (unified modeling language)
  - Class models
  - Goes beyond data, also models behaviors

Steps to Create ERDs

1. Identify Entities	Identify the roles, events, locations, tangible things or concepts about which the end-users want to store data.
2. Find Relationships	Find the natural associations between pairs of entities using a relationship matrix.
3. Draw Rough ERD	Put entities in rectangles and relationships on line segments connecting the entities.
4. Fill in Cardinality	Determine the number of occurrences of one entity for a single occurrence of the related entity.
5. Define Primary Keys	Identify the data attribute(s) that uniquely identify one and only one occurrence of each entity.
6. Draw Key-Based ERD	Eliminate Many-to-Many relationships and include primary and foreign keys in each entity.
7. Identify Attributes	Name the information details (fields) which are essential to the system under development.
8. Map Attributes	For each attribute, match it with exactly one entity that it describes.
9. Draw fully attributed ERD	Adjust the ERD from step 6 to account for entities or relationships discovered in step 8.
10. Check Results	Does the final Entity Relationship Diagram accurately depict the system data?

ERD Model Example



UniversityInstructor	
PK	<u>**InstructorID**</u>
	InstructorLastName InstructorFirstName InstructorOffice

ServiceProject	
PK	<u>**ProjectID**</u>
	ProjectDescription ProjectStartDate

UniversityStudent	
PK	<u>**StudentID**</u>
	StudentLastName StudentFirstName StudentMI StudentDOB

UniversityCourse	
PK	<u>**CourseID**</u>
	CourseName CourseTitle

Now Its Your Turn!

- **SCENARIO:**

A company has several departments. Each department has a supervisor and at least one employee.

Employees must be assigned to at least one, but possibly more departments.

At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects.

The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.

More Options: Historical data are important.

Solving the Problem

1. Identify Entities

- The entities in this system are Department, Employee, Supervisor and Project. One is tempted to make Company an entity, but it is a false entity because it has only one instance in this problem. True entities must have more than one instance.

2. Find Relationships

- We construct the following Entity Relationship Matrix:

	Department	**Employee**	**Supervisor**	**Project**
Department		is assigned	run by	
Employee	belongs to			works on
Supervisor	runs			
Project		uses		

Solving the Problem

3. Fill in Cardinality

From the description of the problem we see that:

- Each department has exactly one supervisor.
- A supervisor is in charge of one and only one department.
- Each department is assigned at least one employee.
- Each employee works for at least one department.
- Each project has at least one employee working on it.
- An employee is assigned to 0 or more projects.

Solving the Problem

4. **Identify Attributes**

- The only attributes indicated are:
  - Department names
  - projects
  - supervisors
  - employees
  - supervisor number
  - employee number
  - project number

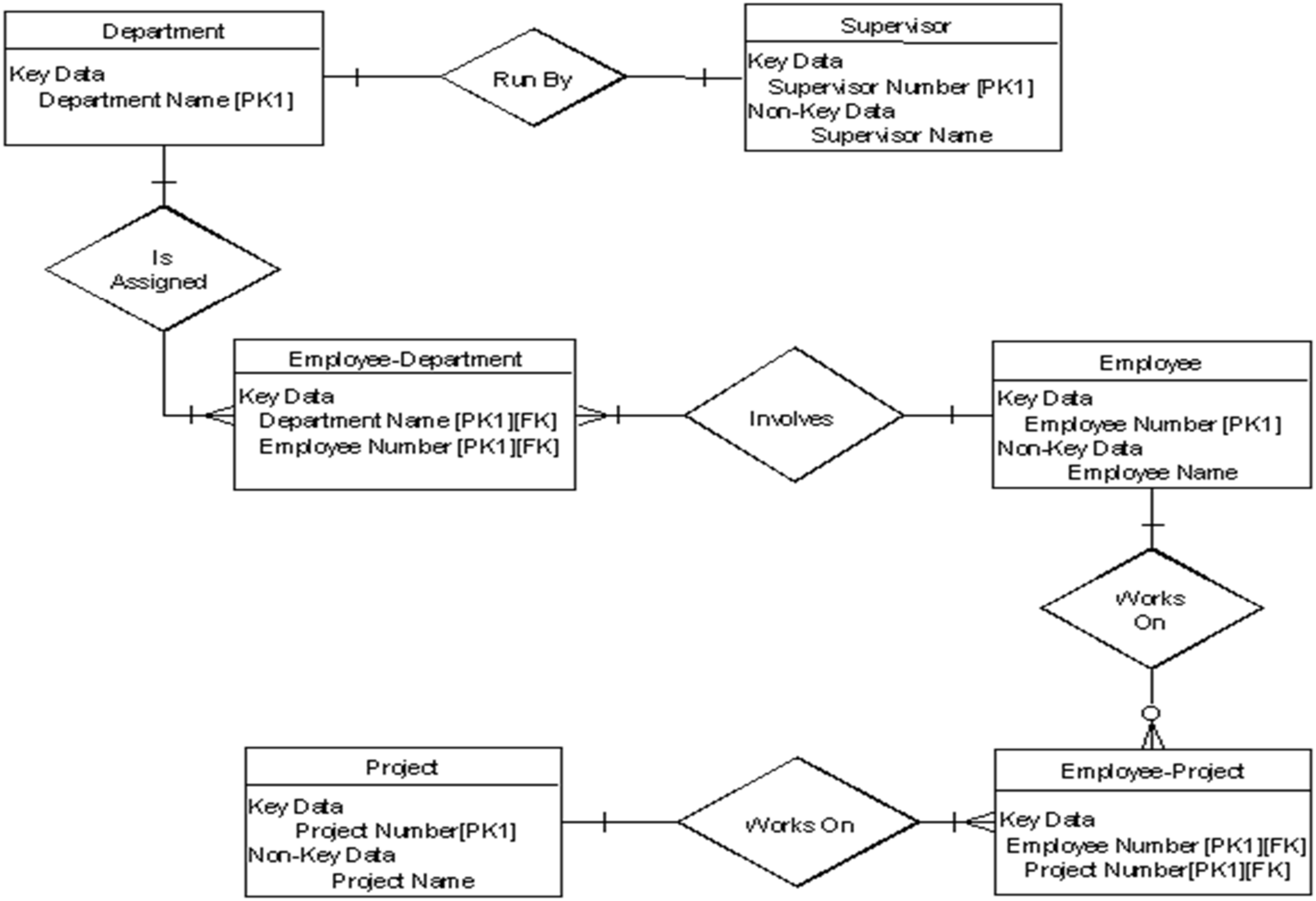
5. **Define Primary Keys**

- The primary keys are:
  - Department Name
  - Supervisor Number
  - Employee Number
    Project Number

6. **Draw Key-Based ERD**

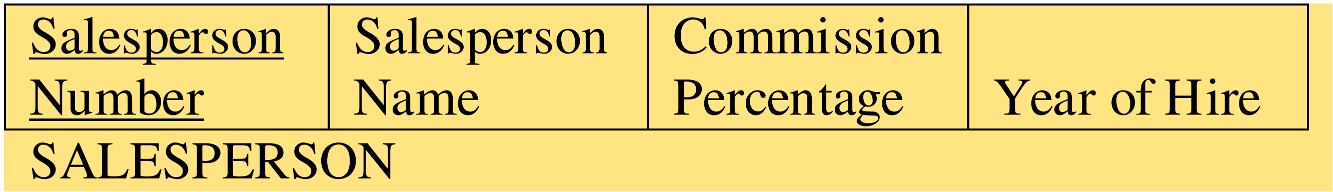
Problem Solved!

7. Draw fully attributed ERD



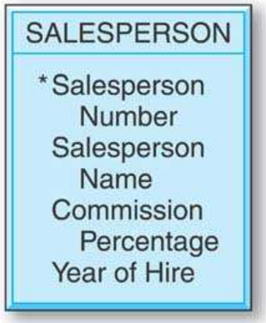
Database Modeling : More Examples

Converting a Simple Entity



<u>Salesperson Number</u>	Salesperson Name	Commission Percentage	Year of Hire

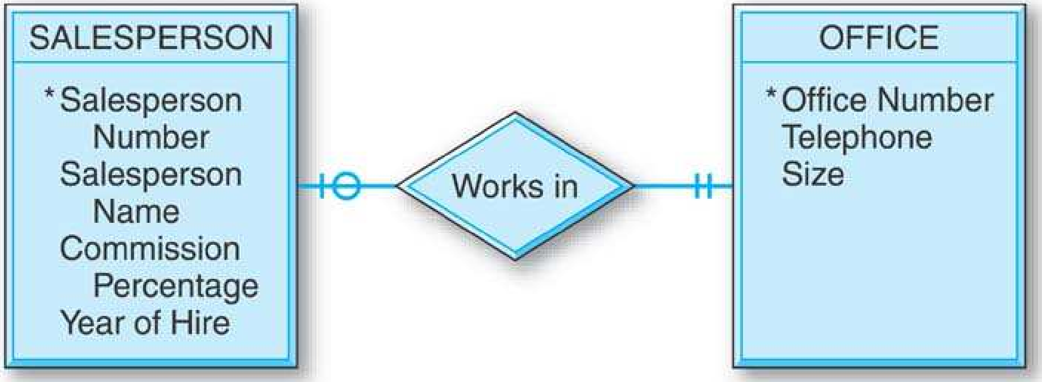
SALESPERSON



- The table simply contains the attributes that were specified in the entity box.

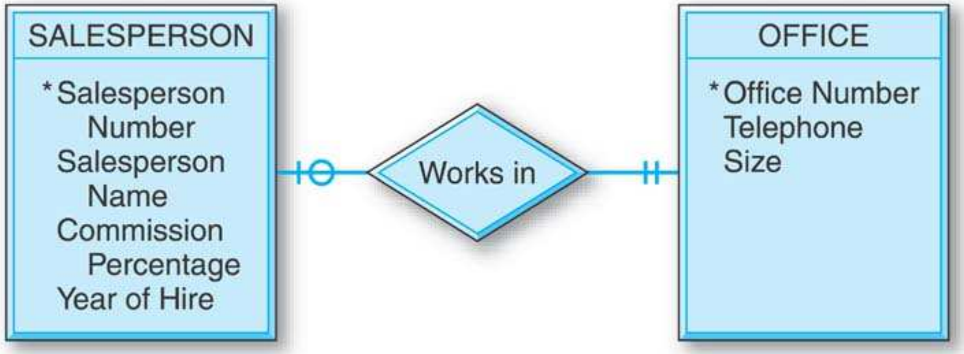
- Salesperson Number is underlined to indicate that it is the unique identifier of the entity and the primary key of the table.

Converting Entities in Binary Relationships: One-to-One



- There are three options for designing tables to represent this data.

One-to-One: Option #1

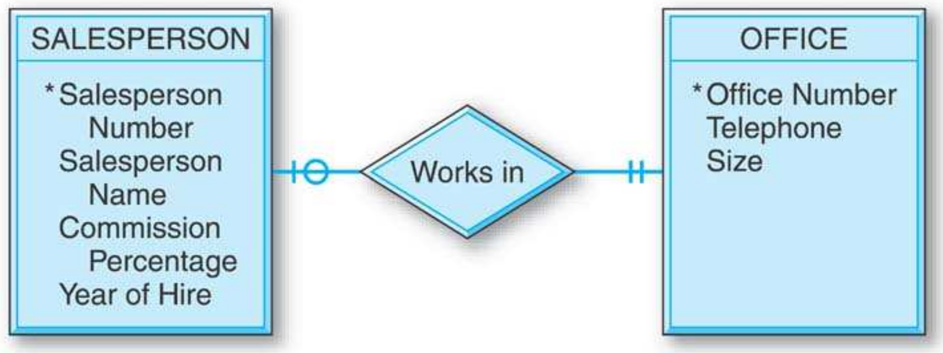


- The two entities are combined into one relational table.

Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire	Office Number	Telephone	Size

SALESPERSON/OFFICE

One-to-One: Option #2



- Separate tables for the SALESPERSON and OFFICE entities, with Office Number as a foreign key in the SALESPERSON table.

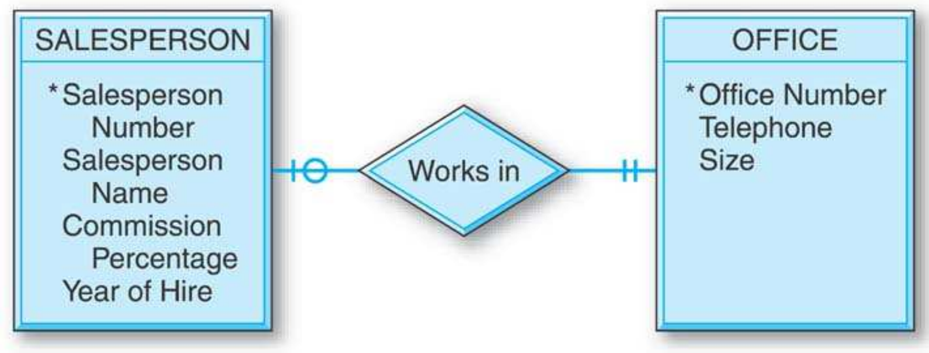
Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire	Office Number

SALESPERSON

Office Number	Telephone	Size

OFFICE

One-to-One: Option #3



- Separate tables for the SALESPERSON and OFFICE entities, with Salesperson Number as a foreign key in the OFFICE table.

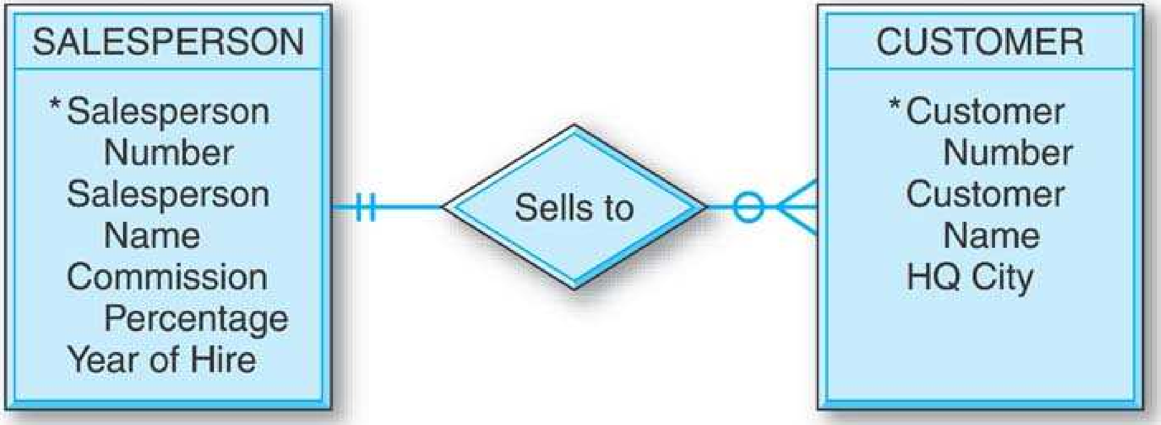
Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire

SALESPERSON

Office Number	Telephone	Salesperson Number	Size

OFFICE

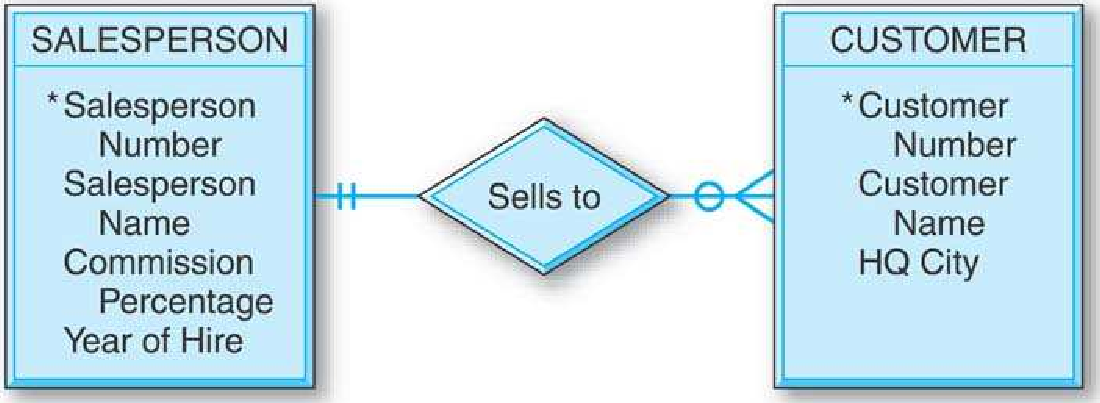
Converting Entities in Binary Relationships: One-to-Many



- The unique identifier of the entity on the “one side” of the one-to-many relationship is placed as a foreign key in the table representing the entity on the “many side.”

- So, the Salesperson Number attribute is placed in the CUSTOMER table as a foreign key.

Converting Entities in Binary Relationships: One-to-Many



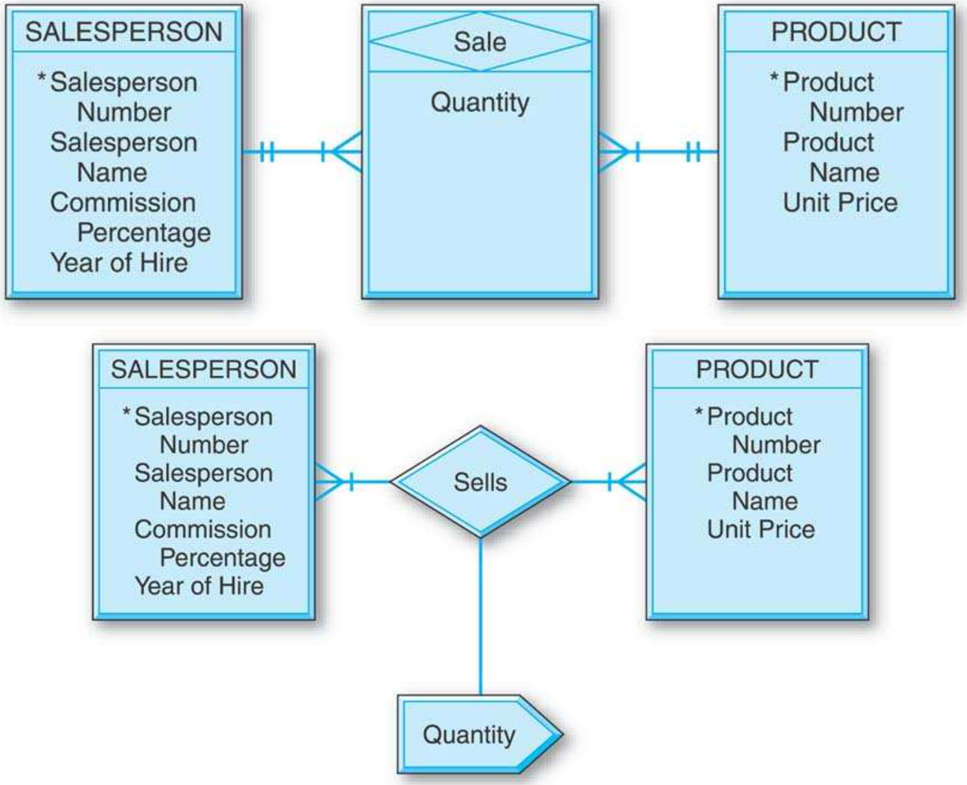
Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire

SALESPERSON

Customer Number	Customer Name	HQ City	Salesperson Number

CUSTOMER

Converting Entities in Binary Relationships: Many-to-Many



- E-R diagram with the many-to-many binary relationship and the equivalent diagram using an associative entity.

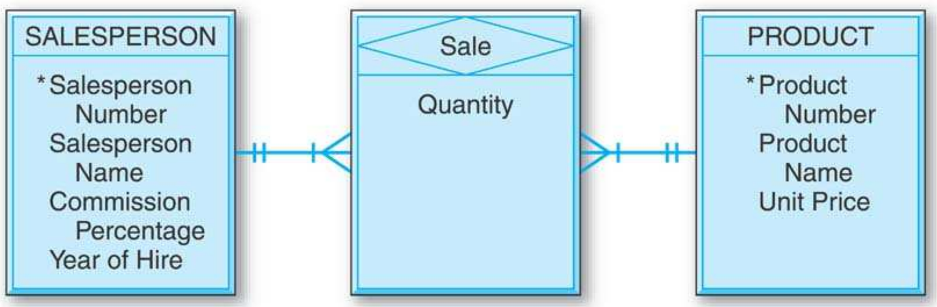
Converting Entities in Binary Relationships: Many-to-Many

- An E-R diagram with two entities in a many-to-many relationship converts to three relational tables.

- Each of the two entities converts to a table with its own attributes but with no foreign keys (regarding this relationship).

- In addition, there must be a third "many-to-many" table for the many-to-many relationship.

Converting Entities in Binary Relationships: Many-to-Many



- The primary key of SALE is the combination of the unique identifiers of the two entities in the many-to-many relationship. Additional attributes are the intersection data.

Product Number	Product Name	Unit Price

PRODUCT

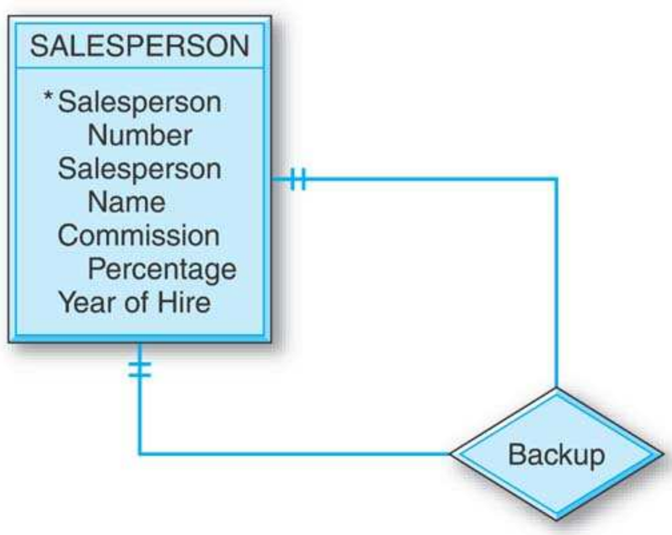
Salesperson Number	Product Number	Quantity

SALE

Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire

SALESPERSON

Converting Entities in Unary Relationships: One-to-One

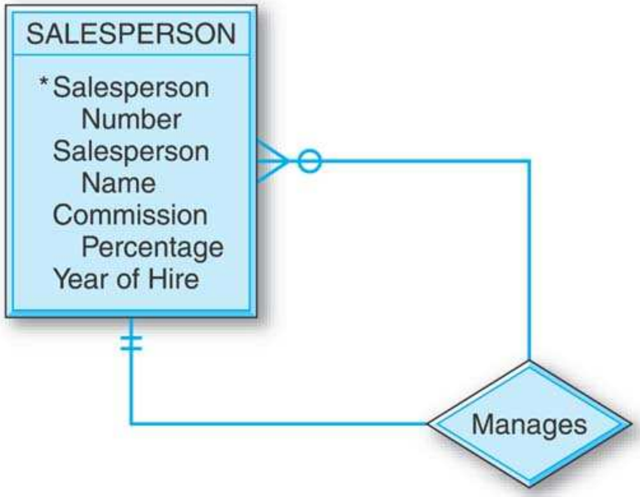


- With only one entity type involved and with a one-to-one relationship, the conversion requires only one table.

Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire	Backup Number

SALESPERSON

Converting Entities in Unary Relationships: One-to-Many

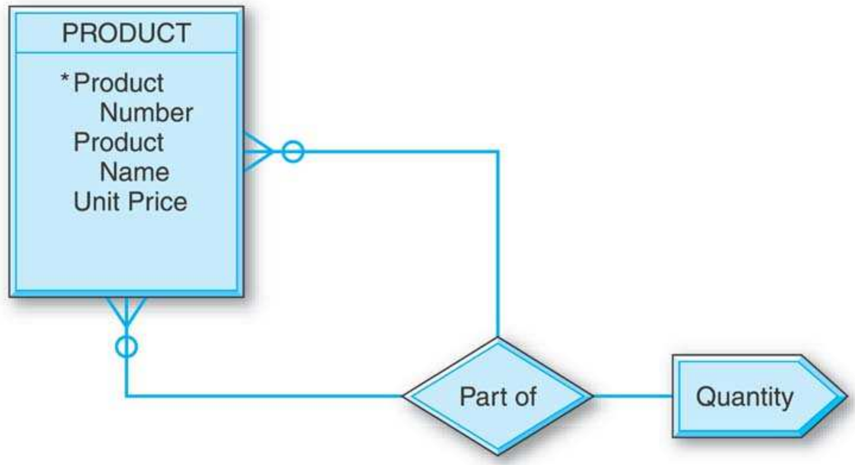


- Very similar to the one-to-one unary case.

Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire	Manager

SALESPERSON

Converting Entities in Unary Relationships: Many-to-Many



Product Number	Product Name	Unit Price

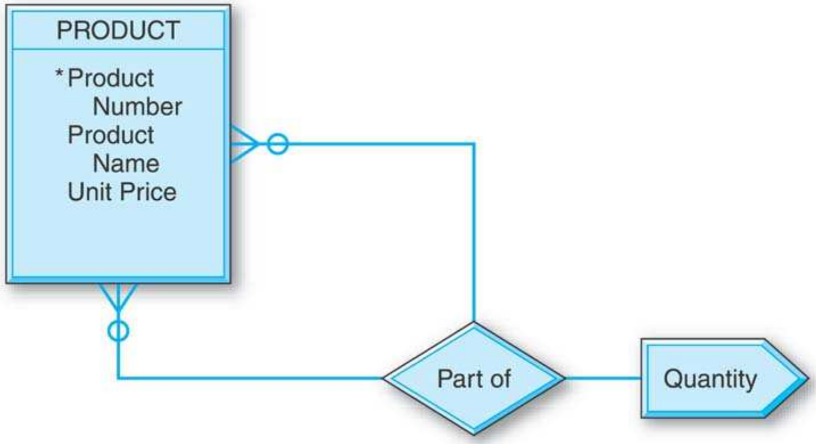
PRODUCT

Product Number	Sub-Assembly Number	Quantity

COMPONENT

- This relationship requires two tables in the conversion.
- The PRODUCT table has no foreign keys.

Converting Entities in Unary Relationships: Many-to-Many



Product Number	Product Name	Unit Price

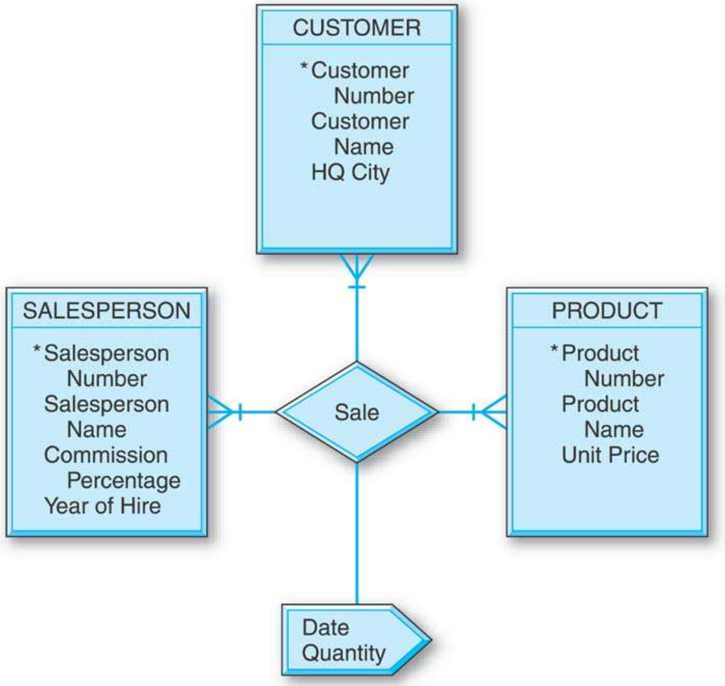
PRODUCT

Product Number	Sub-Assembly Number	Quantity

COMPONENT

- A second table is created since in the conversion of a many-to-many relationship of any degree — unary, binary, or ternary — the number of tables will be equal to the number of entity types (one, two, or three, respectively) plus one more table for the many-to-many relationship.

Converting Entities in Ternary Relationships



Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire

SALESPERSON

Salesperson Number	Customer Number	Product Number	Date	Quantity

SALE

Product Number	Product Name	Unit Price

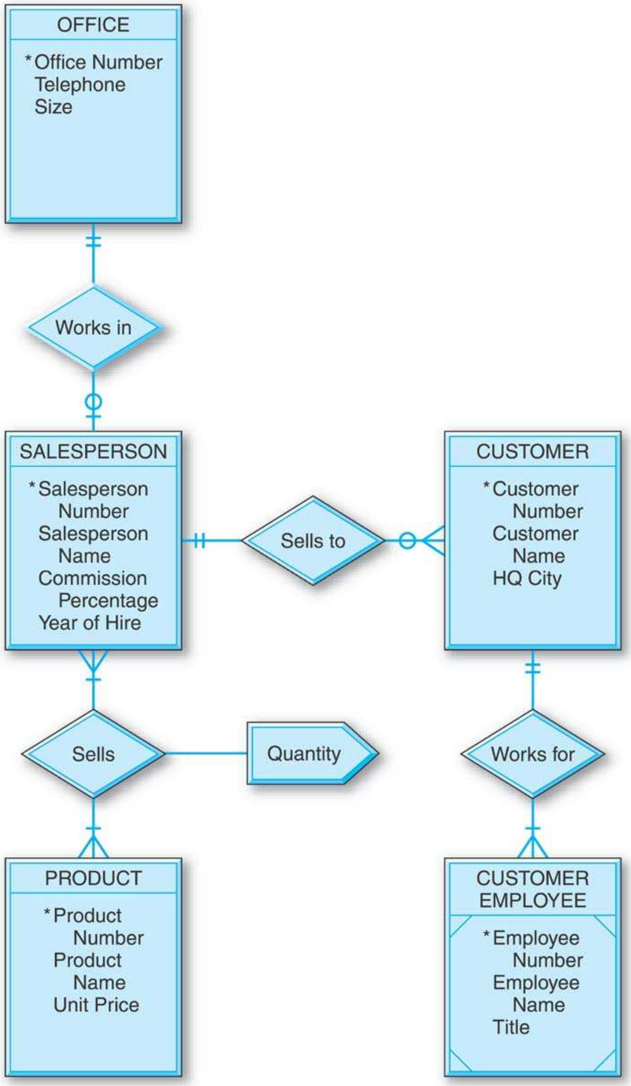
PRODUCT

Customer Number	Customer Name	HQ City

CUSTOMER

- The primary key of the SALE table is the combination of the unique identifiers of the three entities involved, plus the Date attribute.

Designing the General Hardware Company Database



Salesperson Number	Salesperson Name	Commission Percentage	Year of Hire	Office Number

SALESPERSON

Customer Number	Customer Name	Salesperson Number	HQ City

CUSTOMER

Customer Number	Employee Number	Employee Name	Title

CUSTOMER EMPLOYEE

Product Number	Product Name	Unit Price

PRODUCT

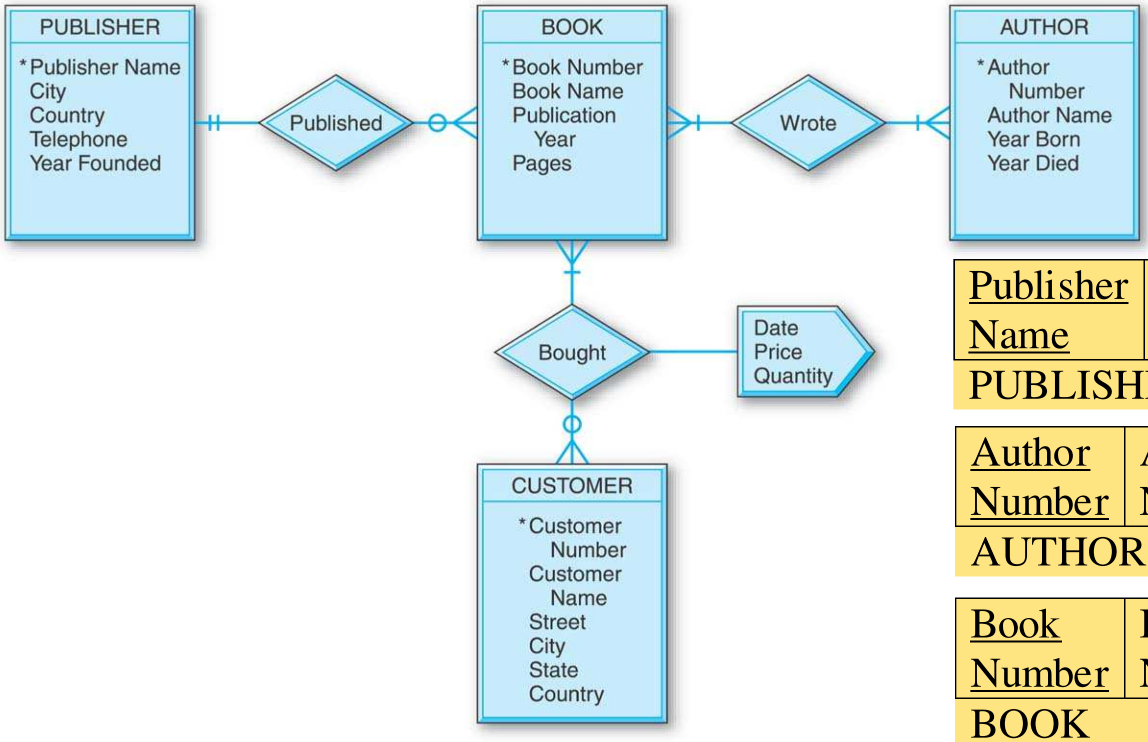
Salesperson Number	Product Number	Quantity

SALES

Office Number	Telephone	Size

OFFICE

Designing the Good Reading Bookstores Database



Publisher Name	City	Country	Telephone	Year Founded

PUBLISHER

Author Number	Author Name	Year Born	Year Died

AUTHOR

Book Number	Book Name	Publication Year	Pages	Publisher Name

BOOK

Customer Number	Customer Name	Street	City	State	Country

CUSTOMER

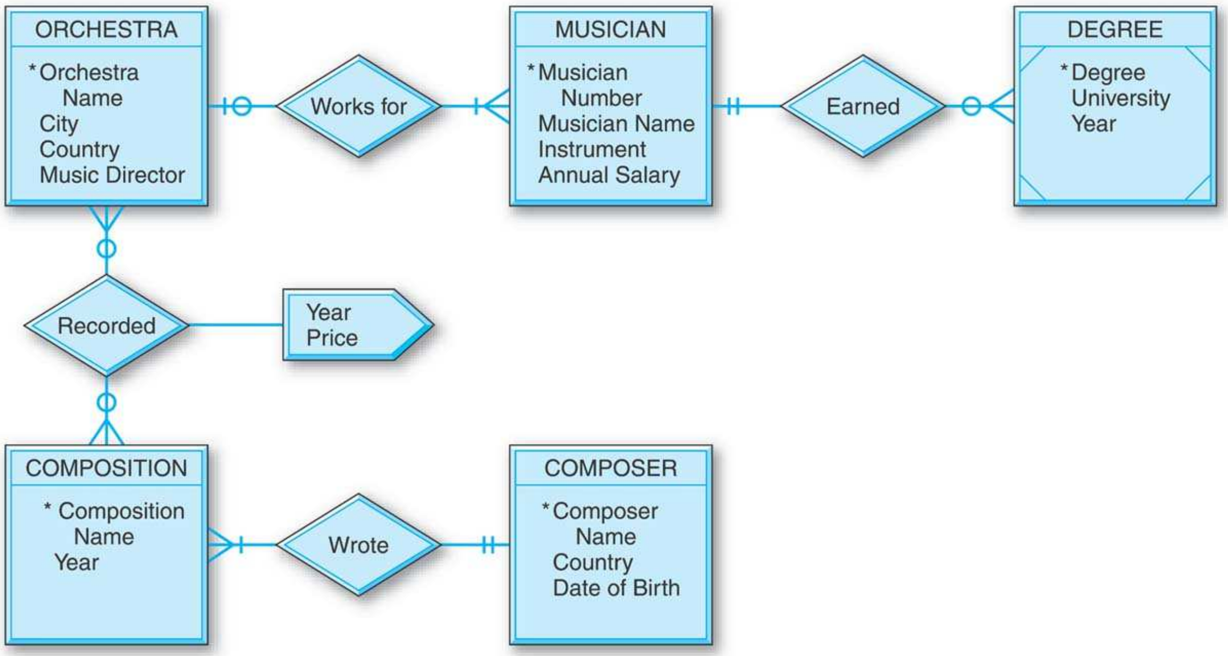
Book Number	Author Number

WRITING

Book Number	Customer Number	Date	Price	Quantity

SALE

Designing the World Music Association Database



Orchestra Name	City	Country	Music Director

ORCHESTRA

Musician Number	Degree	University	Year

DEGREE

Musician Number	Musician Name	Instrument	Annual Salary	Orchestra Name

MUSICIAN

Composer Name	Country	Date Of Birth

COMPOSER

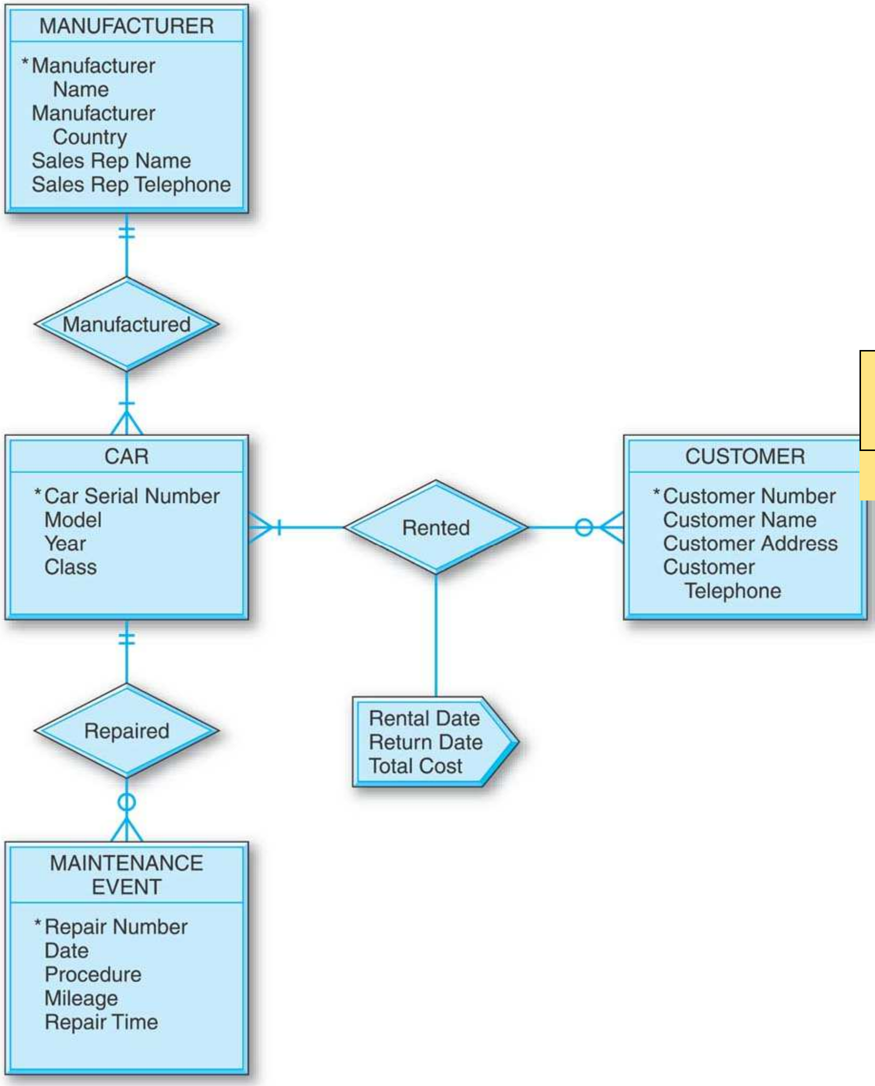
Composition Name	Composer Name	Year

COMPOSITION

Orchestra Name	Composition Name	Composer Name	Year	Price

RECORDING

Designing the Lucky Rent-A-Car Database



Manufacturer Name	Manufacturer Country	Sales Rep Name	Sales Rep Telephone

MANUFACTURER

Car Serial Number	Model	Year	Class	Manufacturer Name

CAR

Repair Number	Car Serial Number	Date	Procedure	Mileage	Repair Time

MAINTENANCE

Customer Number	Customer Name	Customer Address	Customer Telephone

CUSTOMER

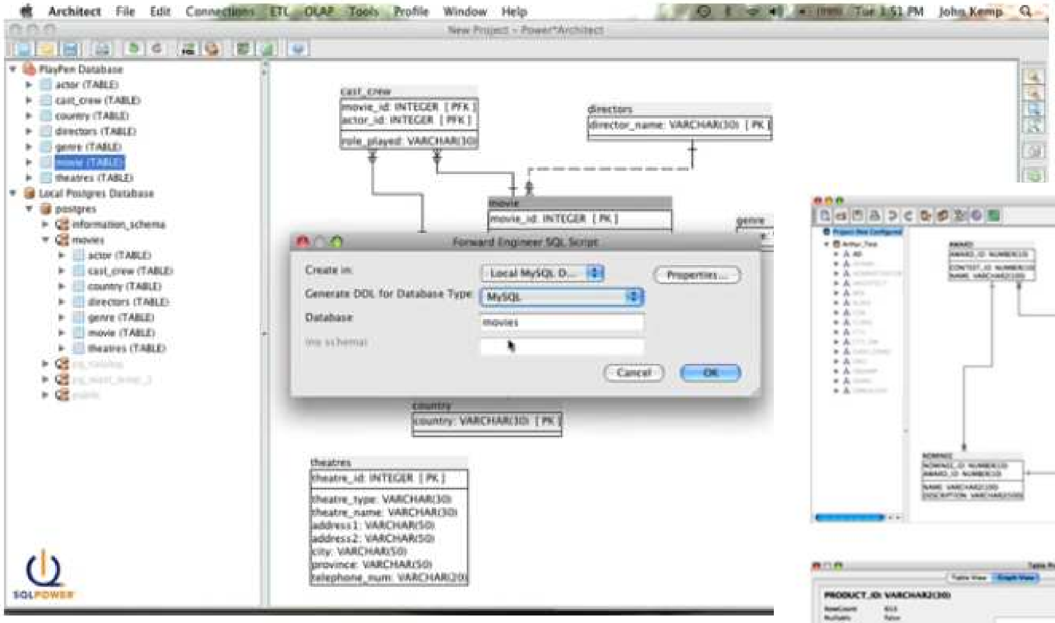
Car Serial Number	Customer Number	Rental Date	Return Date	Total Cost

RENTAL

Database Modeling Tools

SQL Power Architect

- Open source, supports most DBMSs

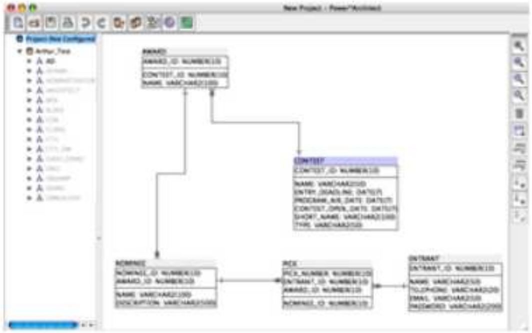


Create in: Local MySQL D...

Generate DDL for Database Type: MySQL

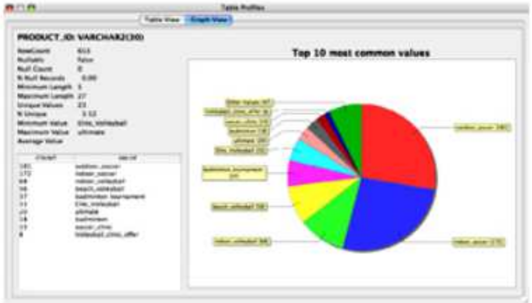
Database: movies

```
cast_crew
movie_id: INTEGER [ PFK ]
actor_id: INTEGER [ PFK ]
role_played: VARCHAR(30)

directors
director_name: VARCHAR(30) [ PK ]

movie
movie_id: INTEGER [ PK ]

country
country: VARCHAR(30) [ PK ]

theatres
theatre_id: INTEGER [ PK ]
theatre_type: VARCHAR(30)
theatre_name: VARCHAR(30)
address1: VARCHAR(50)
address2: VARCHAR(50)
city: VARCHAR(50)
province: VARCHAR(50)
telephone_num: VARCHAR(20)
```



"Playpen" Design Workspace

The tree view makes it easy to navigate through the database. Drag schemas or tables into the playpen, then adjust the layout manually or with the Auto-Layout function.



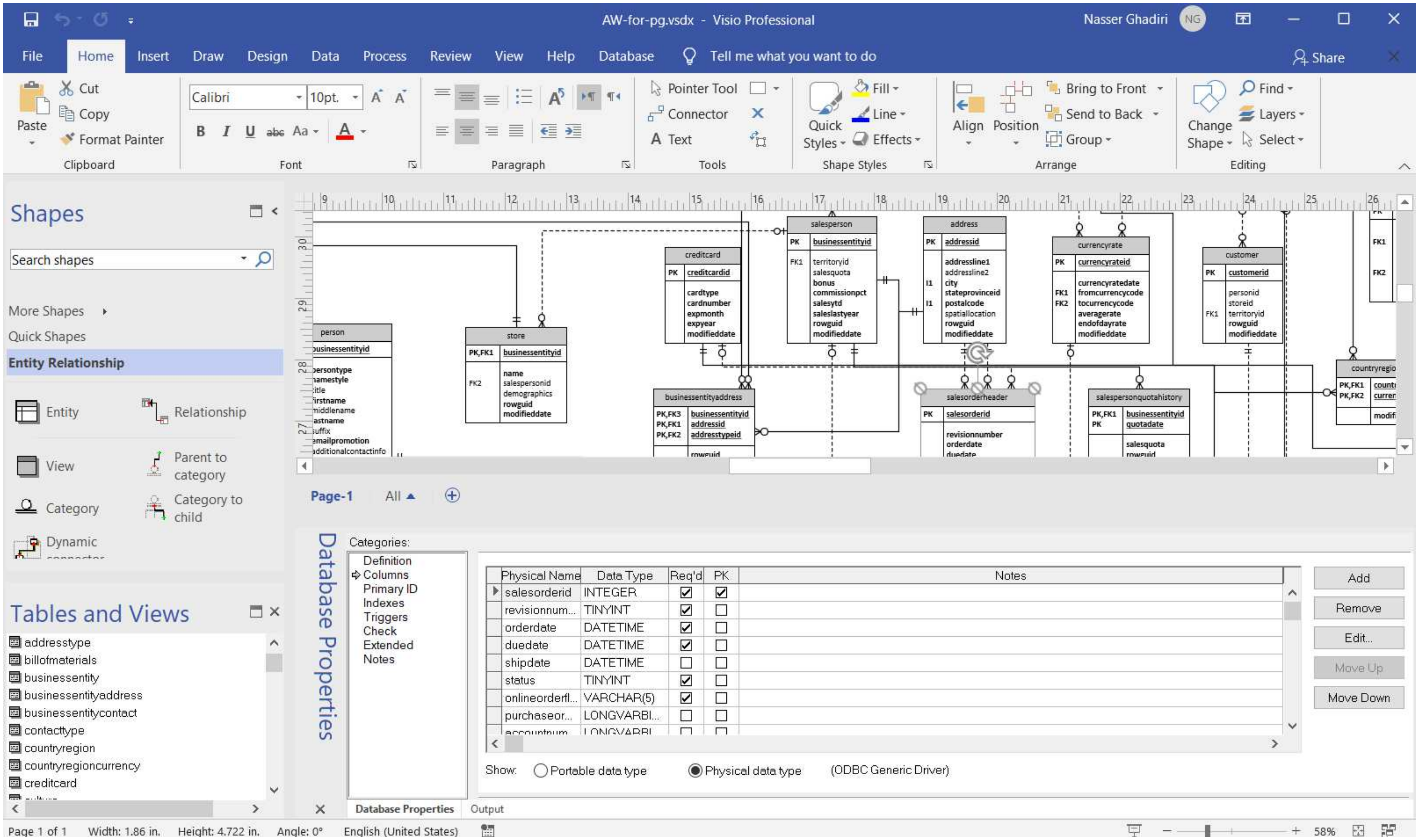
Quick & Easy Data Profiling

The Data Profiling feature is useful for exploring new or unfamiliar databases... you can see information about the size of the data in each column, maximum & minimum values, frequency distribution of values in a column and more.

Forward/Reverse Engineering with SQL Power Architect (3 parts)

https://www.youtube.com/watch?v=QY486ucLWMc

Visio



Categories:

- Definition
- Columns
- Primary ID
- Indexes
- Triggers
- Check
- Extended
- Notes

Physical Name	Data Type	Req'd	PK	Notes
salesorderid	INTEGER	☑	☑	
revisionnum...	TINYINT	☑	☐	
orderdate	DATETIME	☑	☐	
duedate	DATETIME	☑	☐	
shipdate	DATETIME	☐	☐	
status	TINYINT	☑	☐	
onlineorderfl...	VARCHAR(5)	☑	☐	
purchaseor...	LONGVARBI...	☐	☐	
accountnum...	LONGVARBI...	☐	☐	

Add

Remove

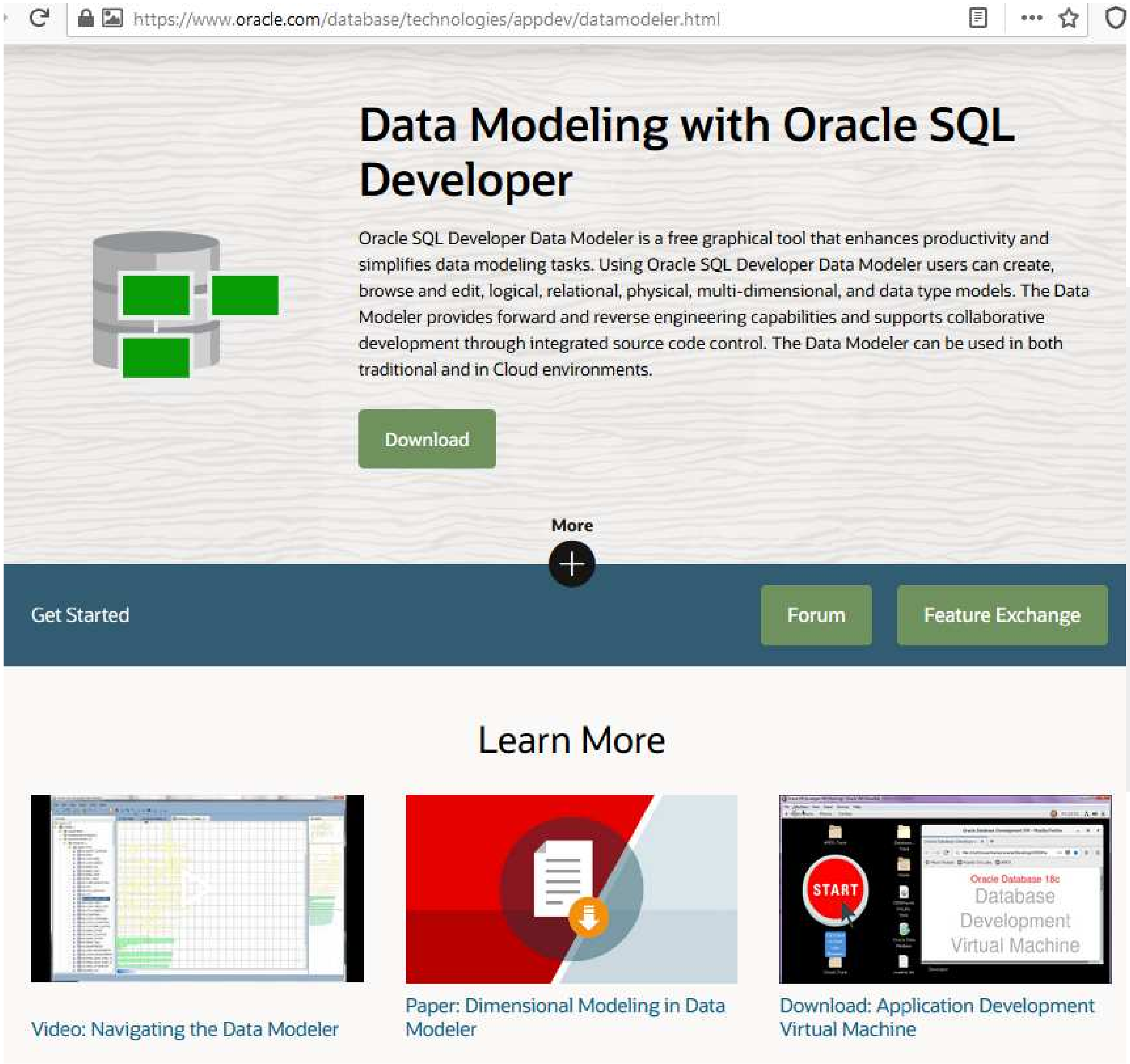
Edit...

Move Up

Move Down

Show: ○ Portable data type ● Physical data type (ODBC Generic Driver)

Oracle Data Modeler



https://www.oracle.com/database/technologies/appdev/datamodeler.html

Data Modeling with Oracle SQL Developer

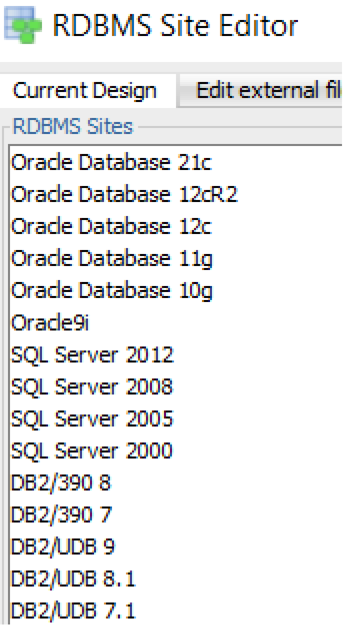
Oracle SQL Developer Data Modeler is a free graphical tool that enhances productivity and simplifies data modeling tasks. Using Oracle SQL Developer Data Modeler users can create, browse and edit, logical, relational, physical, multi-dimensional, and data type models. The Data Modeler provides forward and reverse engineering capabilities and supports collaborative development through integrated source code control. The Data Modeler can be used in both traditional and in Cloud environments.

Learn More

Video: Navigating the Data Modeler

Paper: Dimensional Modeling in Data Modeler

Download: Application Development Virtual Machine

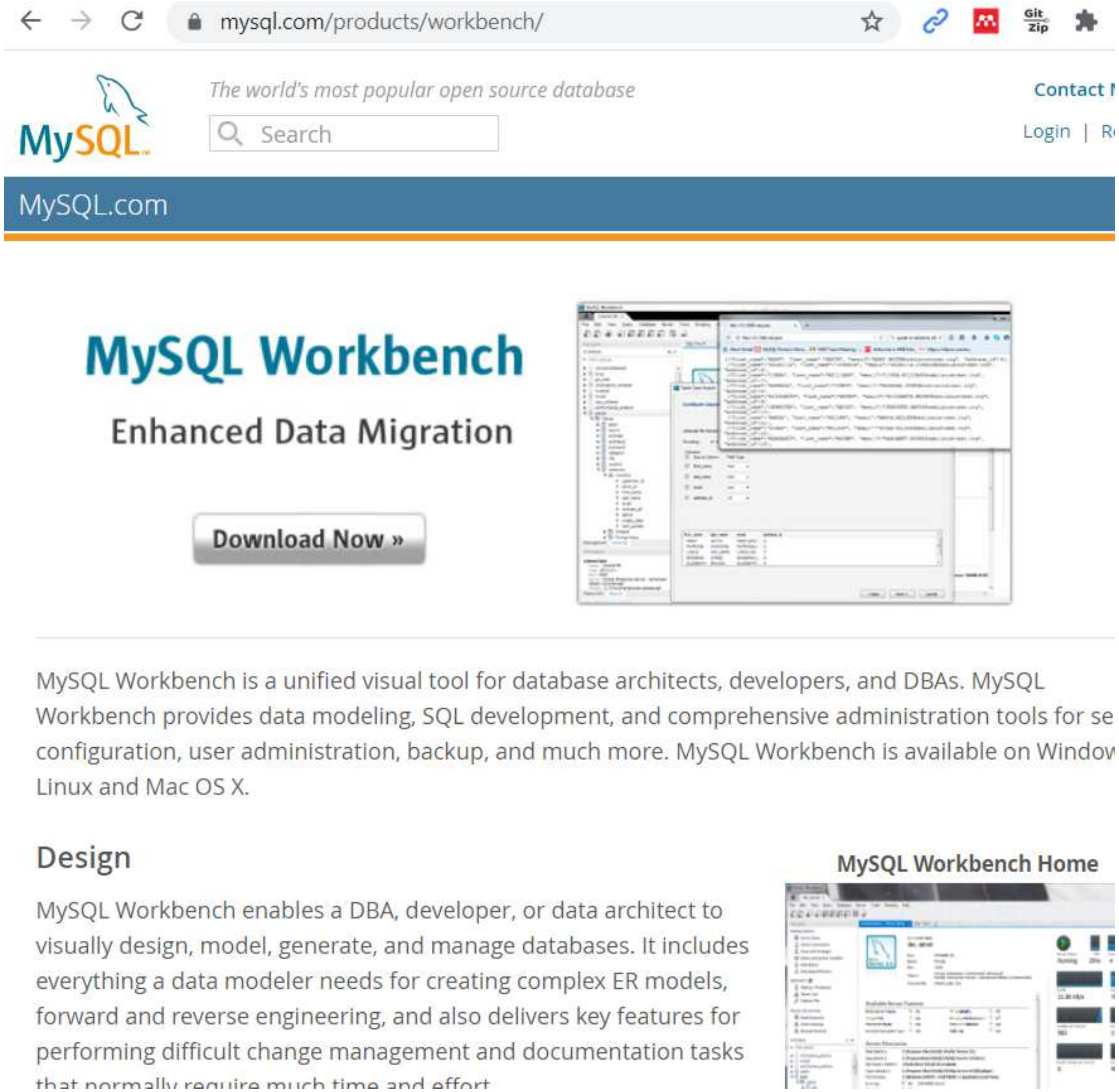


RDBMS Site Editor

RDBMS Sites

- Oracle Database 21c
- Oracle Database 12cR2
- Oracle Database 12c
- Oracle Database 11g
- Oracle Database 10g
- Oracle9i
- SQL Server 2012
- SQL Server 2008
- SQL Server 2005
- SQL Server 2000
- DB2/390 8
- DB2/390 7
- DB2/UDB 9
- DB2/UDB 8.1
- DB2/UDB 7.1

MySQL Workbench



mysql.com/products/workbench/
MySQL
The world's most popular open source database
Search
Contact I
Login | R
MySQL.com
Download Now »

MySQL Workbench

Enhanced Data Migration

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for se configuration, user administration, backup, and much more. MySQL Workbench is available on Windov Linux and Mac OS X.

Design

MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort

MySQL Workbench Home

Other ERD Tools

- CA ERWin
- Visual Paradigm UML

ca technologies ERwin

Microsoft Office Visio

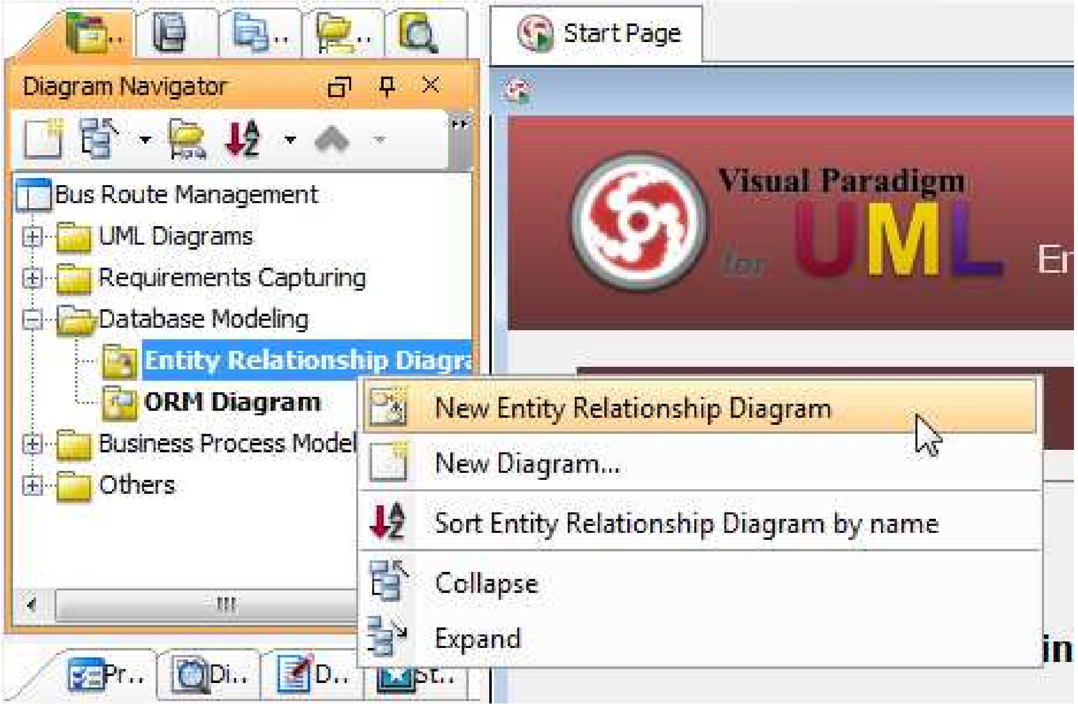


Diagram Navigator
Bus Route Management
UML Diagrams
Requirements Capturing
Database Modeling
Entity Relationship Diagra
ORM Diagram
Business Process Model
Others
Start Page
Visual Paradigm
for
UML
En
New Entity Relationship Diagram
New Diagram...
Sort Entity Relationship Diagram by name
Collapse
Expand
Pr..
Di..
D..
St..
in