
Before we start...

This is the **Introduction to Databases Design and Implementation** workshop

- Download material: dartgo.org/db-design
 - Poll / Interactive questions: dartgo.org/poll
 - Optional software:
<https://dev.mysql.com/downloads/workbench/>
 - More info: rc.dartmouth.edu
-



DARTMOUTH

Information, Technology and Consulting

Introduction to Database Design and Implementation

Christian Darabos, Ph.D.

christian.Darabos@dartmouth.edu

Slides download: dartgo.org/db-design

Overview

- introduction to Databases and this workshop
 - development/production environments
 - tools (admin, browse, query, etc.)
 - DB design, UML and case study
(<http://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html>)
 - port model into MySQL Workbench
-

Right-click > Open link in new window
To keep open slides and poll

dartgo.org/poll

How many RC/RTL workshops have you attended? (excluding this one)

0 (this is my first)

1 - 2

2 - 5

5 - 10

10+



Research Computing

Resources and Services

- Software -

Get access to, and support for, research software applications as well as custom software development.

- Hardware -

Run compute intensive, large memory programs quickly and efficiently.
Store your data securely.

- Consulting -

Get advice and assistance from our application specialists in HPC, GIS, Humanities, Sciences and more. Strengthen your grant applications.

- Training -

Learn how to use research software and systems through our live training sessions and online classes.

Most of our services and resources are available at no-cost to members of the Dartmouth research community including faculty, post-docs, graduate, and undergraduate students

Set up an Account

at rc.dartmouth.edu

Contact Us

at Research.Computing@dartmouth.edu

Introduction

- Research Computing service offering
- Definition of a Relational Database
- Overview of this workshop



Right-click > Open link in new window
To keep open slides and poll

dartgo.org/poll

🖥️ When poll is active, respond at **PollEv.com/dartrc**

💬 Text **DARTRC** to **37607** once to join

In one word, what is a database to you? (word cloud)



Definition of a Relational Database (SQL)

- a database type structured to recognize relations among stored items of information
- designed to store text, dates/times, integers, floating-point number
- implemented as a series of tables



Mental Model

- Think of a database as a set of spreadsheets
 - Each spreadsheet (or table) represents a type of entity (person, object, concept, etc.)
 - Better than Excel because it also models the relationship between the entities
-

Why use a Relational Database

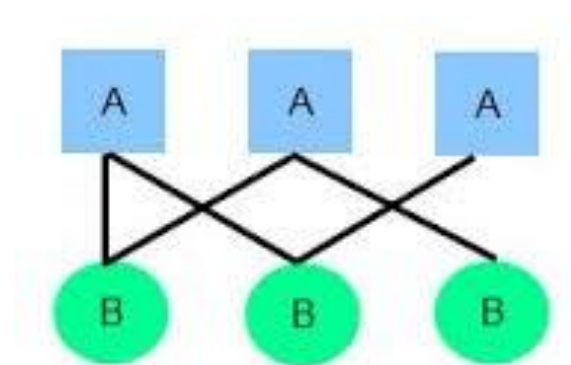
- concurrent (simultaneous) read and **write**
 - **powerful selecting, filtering and sorting** cross-referencing tables
 - large quantity of structured storage and standardized distribution
 - minimize post-processing (simple analytics tools pre-implemented)
 - automate using any scripting and programming languages (R, Matlab, Python, C++, Java, PHP)
 - web-proof
-

SQL vs. NoSQL

SQL	NoSQL
Relational Databases	distributed database
table based	document based, key-value pairs, graph databases or wide-column stores
predefined schema	dynamic schema for unstructured data
vertically scalable (more powerful hardware)	horizontally scalable (more hardware)
SQL (structured query language)	proprietary language
MySql, Oracle, Sqlite, Postgres, MariaDB, ...	MongoDB, BigTable, Redis, RavenDb, Cassandra, ...

Methodologies

- **(Enhanced) Entity-Relationship — (E)ER —model:** a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems.
- **Unified Modeling Language (UML):** a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system.



Development vs. Production Environments

- **Development**

- local to your machine
- used for development and update purposes
- flexible and robust
- **low** accessibility, availability, and performance

- **Production**

- distant server
 - used for client/customer access
 - safe, secure, redundant, and backed up
 - **high** accessibility, availability and performance
-

Disclaimer

- I do not endorse any software over any other
 - I am no expert in any particular software environment used in this workshop
 - This is a workshop about methodology, not about any particular software
 - Both EER and UML are methodologies that are assumption based, usually on incomplete problem sets. They are therefore open to interpretation and have multiple ways of modeling the same problem, based on those assumptions
 - We will not cover database management, optimization or administration
 - I am happy to help and answer questions. Always. If, however, you need troubleshooting or have questions that are very particular to your problem set, let's take those offline
 - I probably have too much material for a 2 hours workshop
-

Development Environment

- **MAMP / MAMP Pro**
- Windows & Mac OS X
- Web servers: Apache, Ngnix
- Database server: MySQL
- Web/DB admin tools: phpMyAdmin, phpLiteAdmin
- Scripting: PHP, Python, Perl
- User friendly: “Web Start”



Production Environment

- RTL / Research Computing
 - Institutional Information Systems & Services
 - Cloud services (AWS, Azure, Google Cloud, etc.)
 - Google: free database hosting :)
-

Database Design Environment

- pen and paper
 - MySQL Workbench
 - visualize and manage DB and its content
 - submit queries, visualize and export results
 - translates UML into a DB implementation
-

Case Study

- A retail back-office manager wants to track the sales at multiple retail locations (shops):
 - customers purchase products in a shop
 - purchased products are bundled in a sale
 - each sale is attributed to a vendor
-

Enhanced Entity-Relationship modeling steps

1. Identify Entities and their Attributes
 2. Identify Relationships
 3. Keys Assignment
 4. Attributes Data Types
 5. Normalization
-

Identify Entities (I)

- entities are the individuals and objects (concrete or abstract) of which you need to represent the interactions or relationships
 - **entities types** are stored as tables (i.e. customers, stores, products, etc.)
 - **entities** are stored as entries (line items) in the tables (ie. John, JCrew 747, cashmere sweater, etc.)
-

Go to poll window now :)

dartgo.org/poll

What are the possible entities in the case study presented?

Top



Identify Entities (II)

Customers

Products

Shops

Vendors

Sales

Entities: types of information.

Identify Attributes (I)

- attributes are the relevant characteristics of the entities
 - attribute types are stored as columns in entity type tables (i.e. customers number, stores' street address, products unit price, etc.)
 - attributes of each entity are stored as elements (column items) in the tables (ie. 12345, Main St, \$200, etc.)
-

What are relevant attribute types of 'Customers'?

Top



Start the presentation to see live content. For screen share software, share the entire screen. Get help at padl.eu.com/app

What are relevant attribute types of 'Sale'?

Top



Start the presentation to see live content. For screen share software, share the entire screen. Get help at padluc.com/app

Identify Attributes (II)



Entities: Entities with attributes.

Identify Relationships (I)

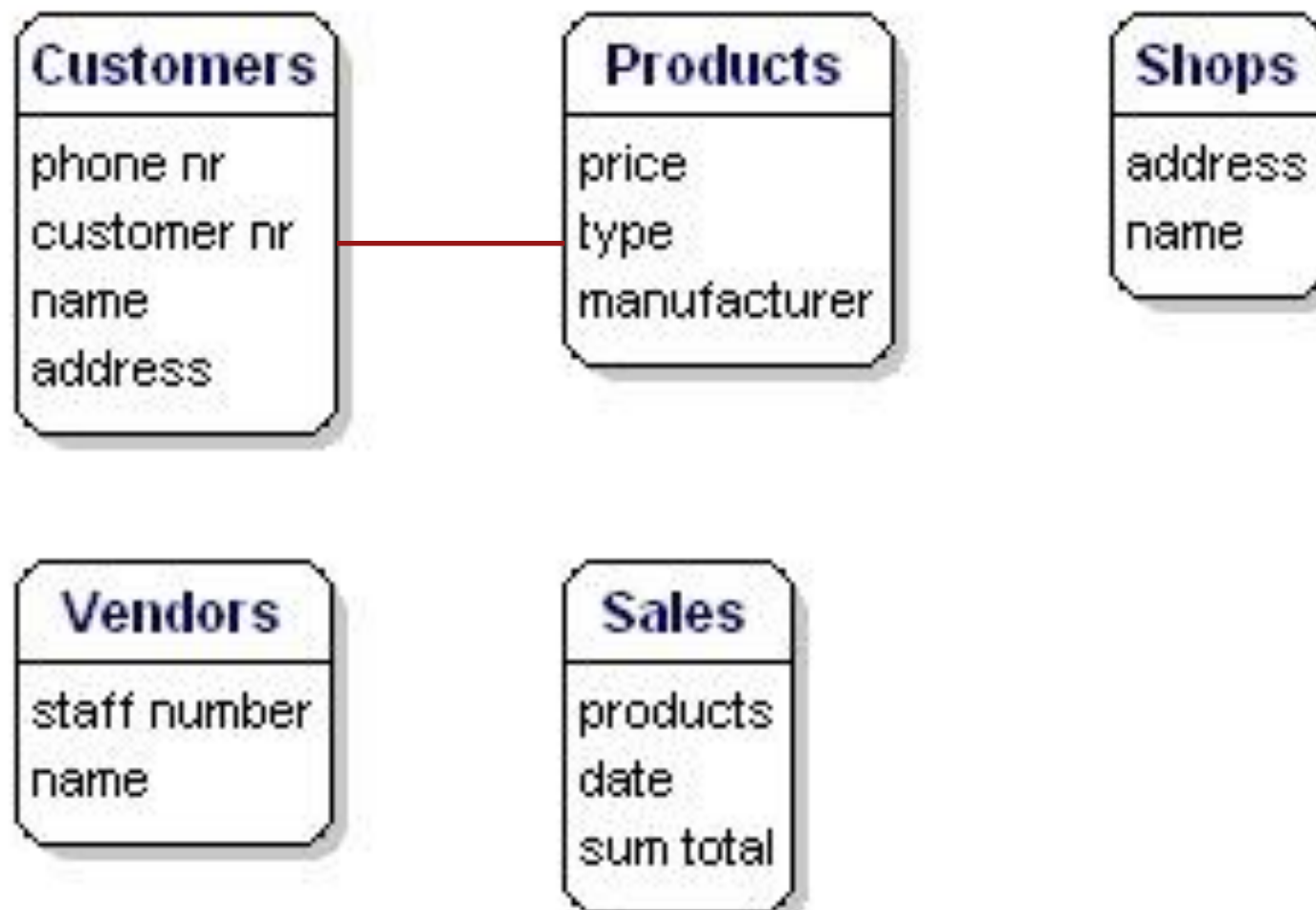
English grammar structure	EER structure	Example
Common noun	Entity type	<i>customer</i>
Proper noun	Entity	<i>Jane Doe</i>
Transitive verb	Relationship type	<i>Jane buys a pen</i>
Intransitive verb	Attribute type	<i>Jane moved</i>
Adjective	Attribute for entity	<i>Jane is blond</i>
Adverb	Attribute for relationship	<i>Jane buys a pen impulsively</i>

Identify Relationships (II)



Entities: Entities with attributes.

Identify Relationships (II)



Entities: Entities with attributes.

Identify Relationships (II)



Entities: Entities with attributes.

Identify Relationships (II)



Entities: Entities with attributes.

Identify Relationships (II)



Entities: Entities with attributes.

Identify Relationships (II)



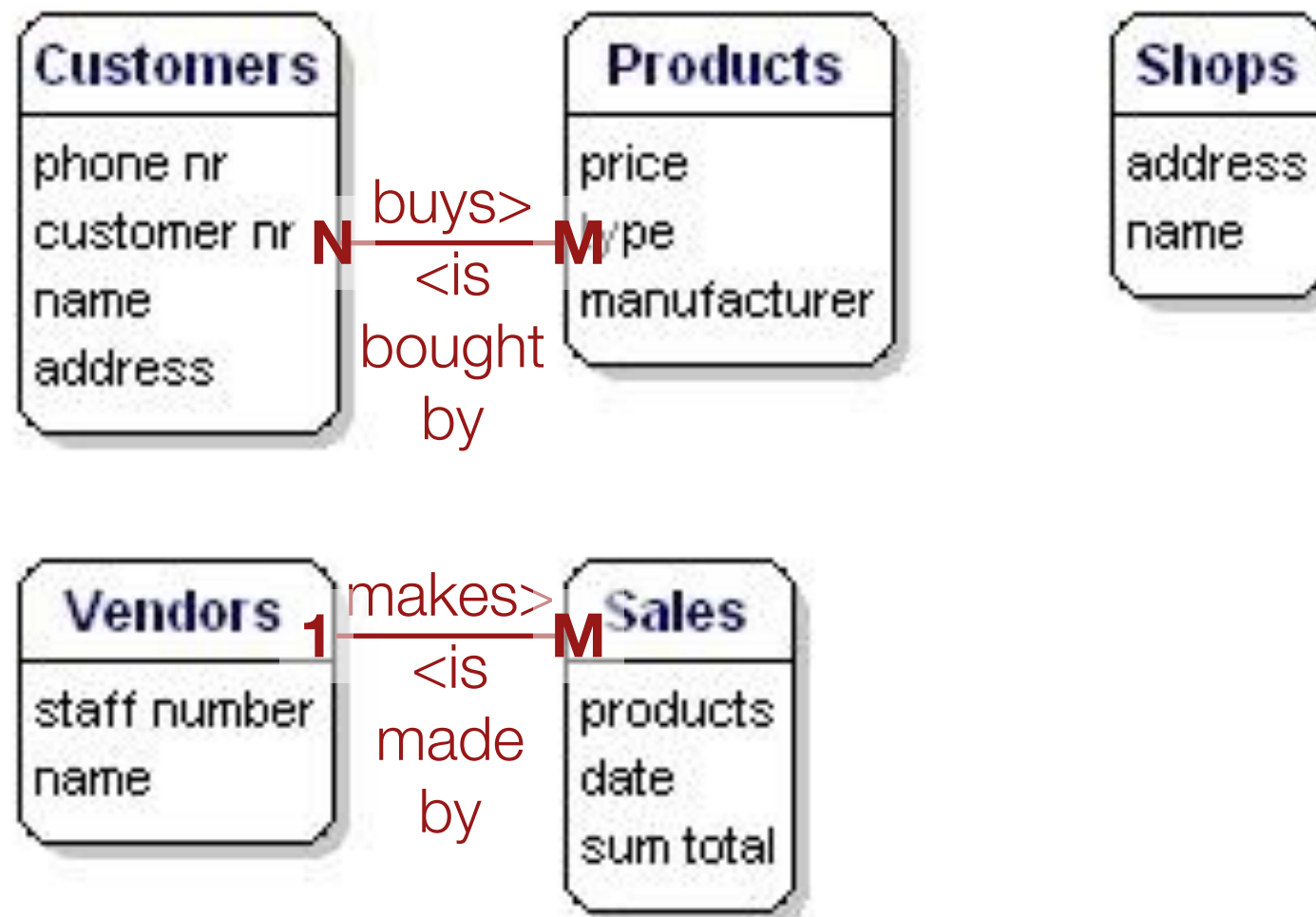
Entities: Entities with attributes.

What's a possible relationship between "Sales" and "Vendors"?

Top

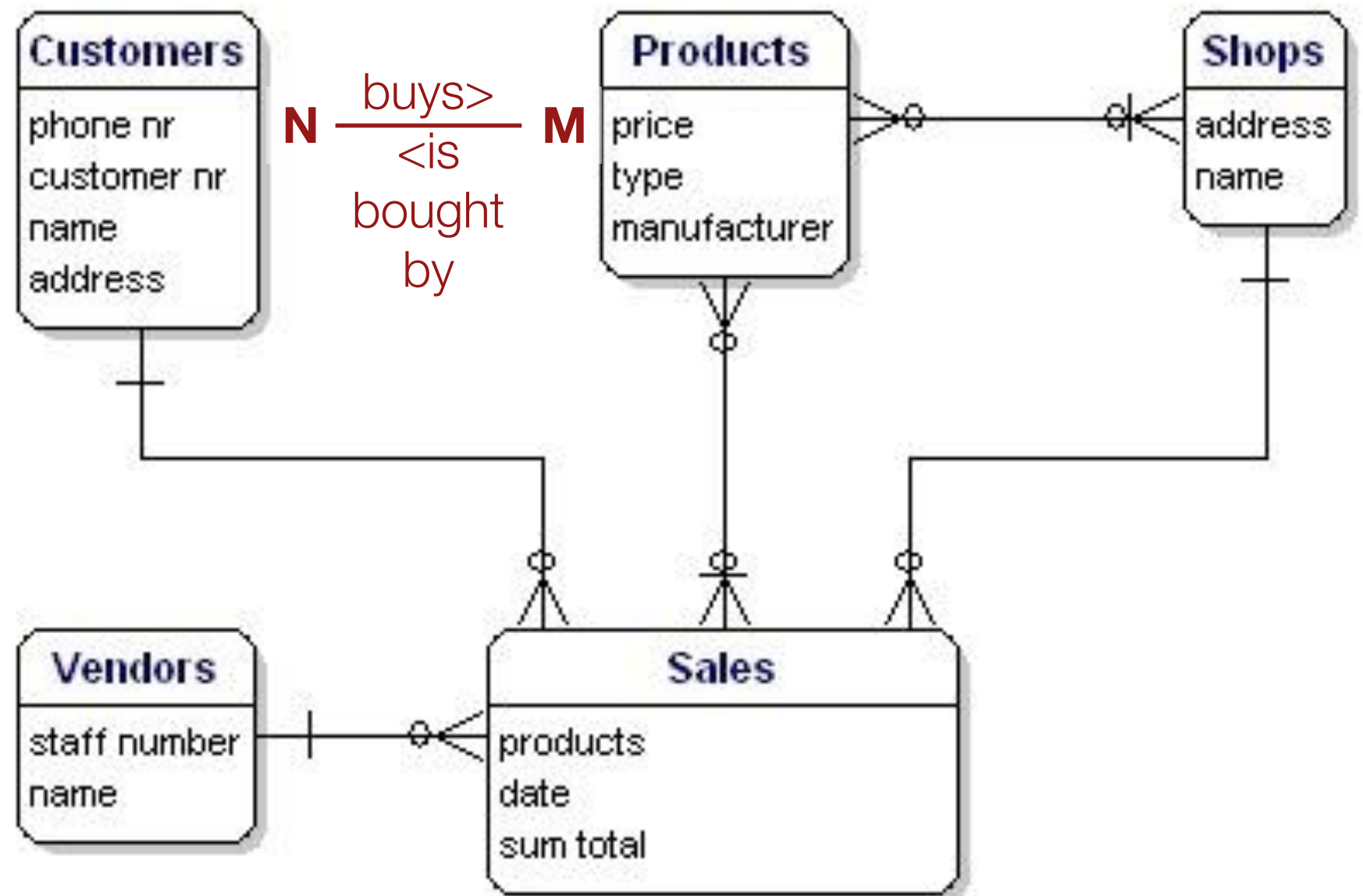


Identify Relationships (II)



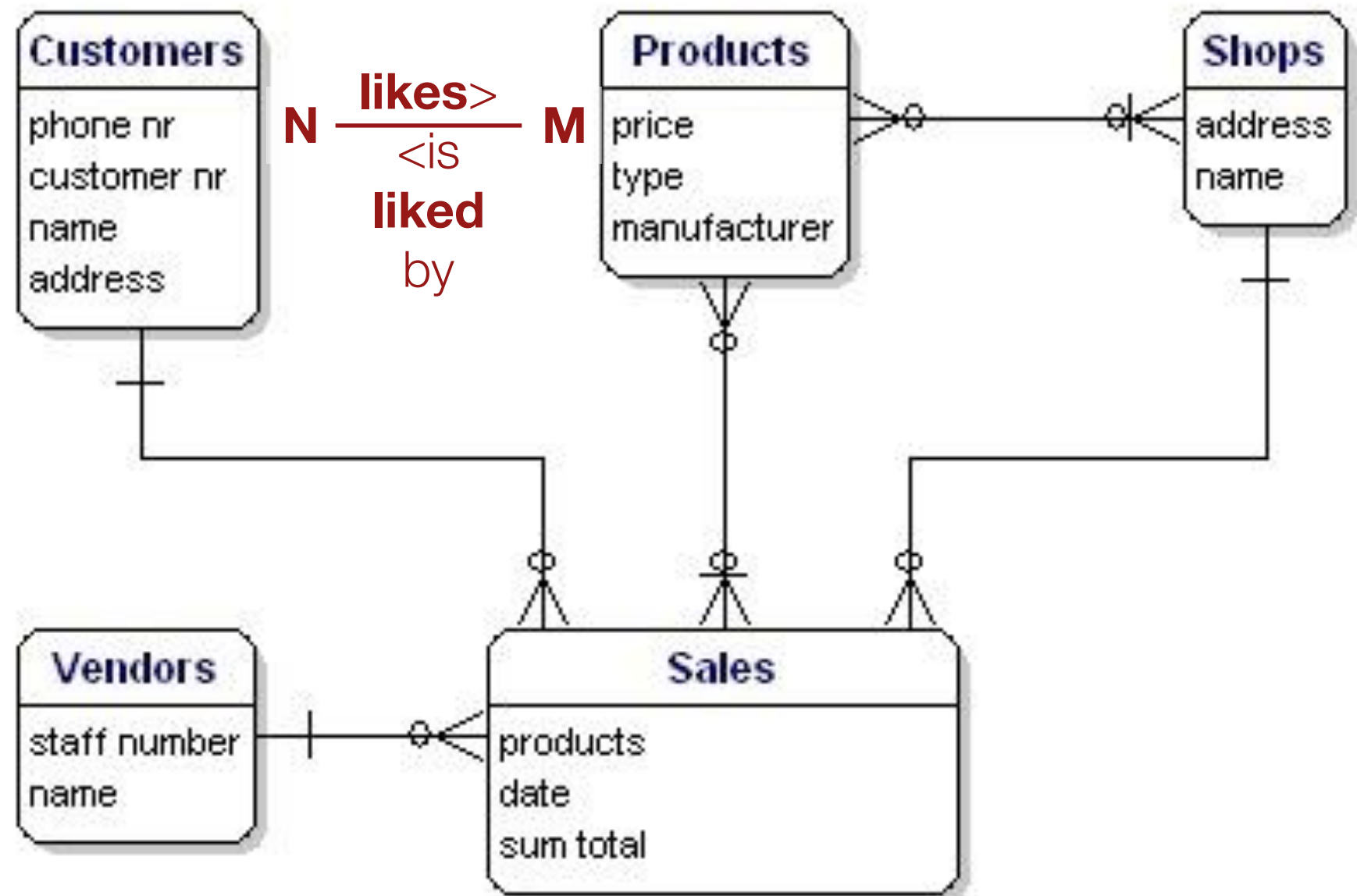
Entities: Entities with attributes.

Identify Relationships (III)



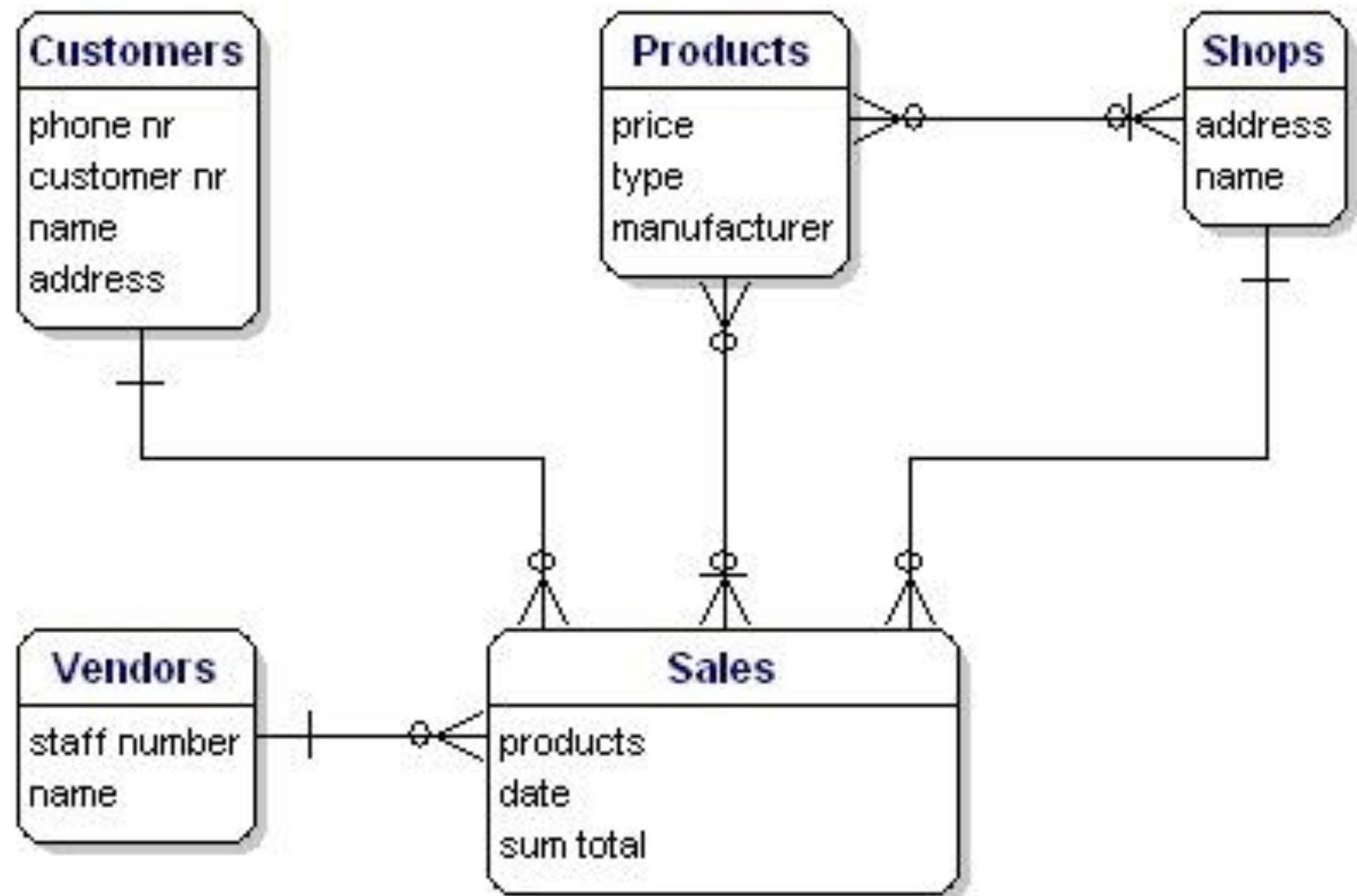
Relationships between the entities.

Identify Relationships (III)



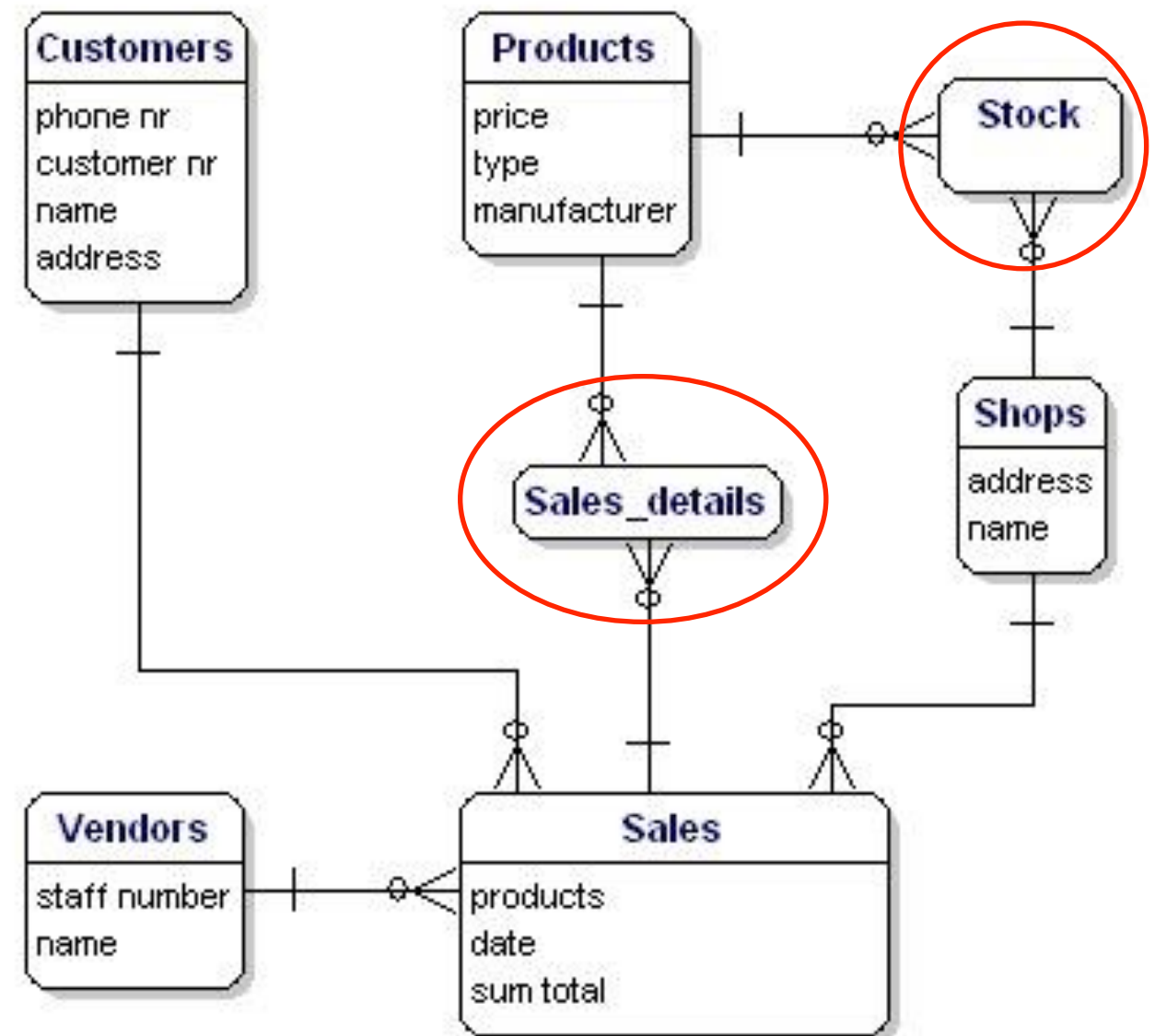
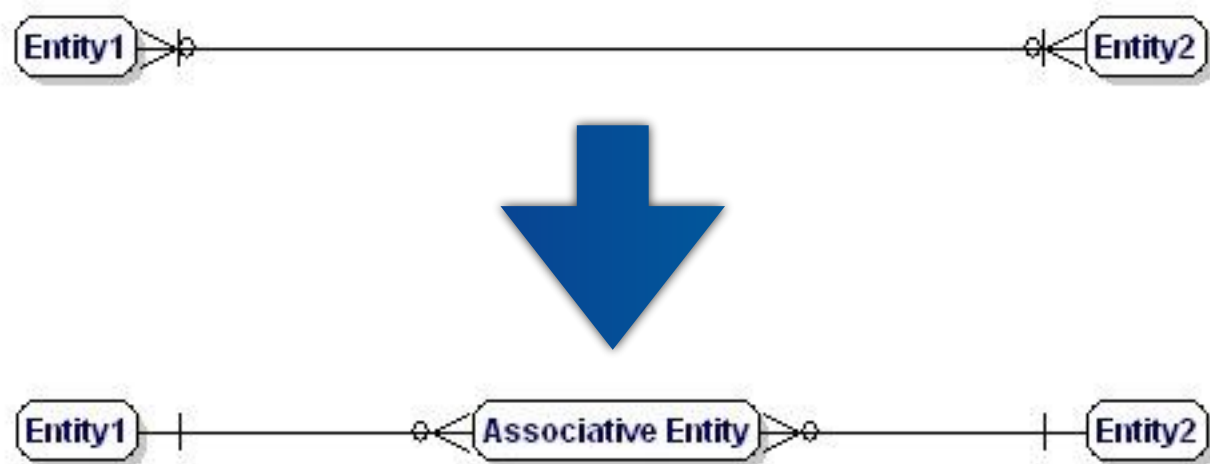
Relationships between the entities.

Identify Relationships (III)

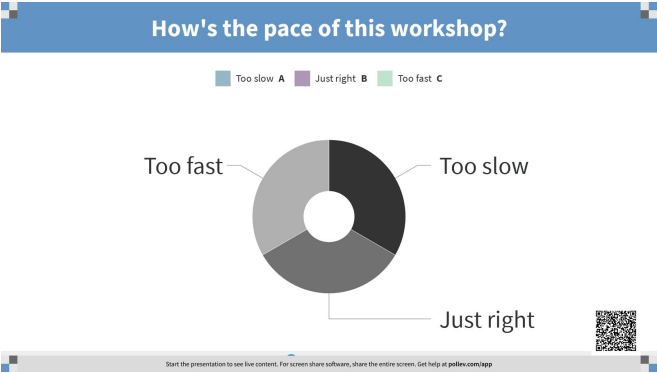


Relationships between the entities.

Identify Relationships (III)



Relationships between the entities.



Keys Assignment (I)

- **primary key** (PK) is a set of one or more data attributes that uniquely identify an entity
 - **foreign key** (FK) in an entity is the reference to the primary key of another entity
 - **indexed** fields are “indexed” in a separate manner to increase make their referencing faster at the expense of space
-

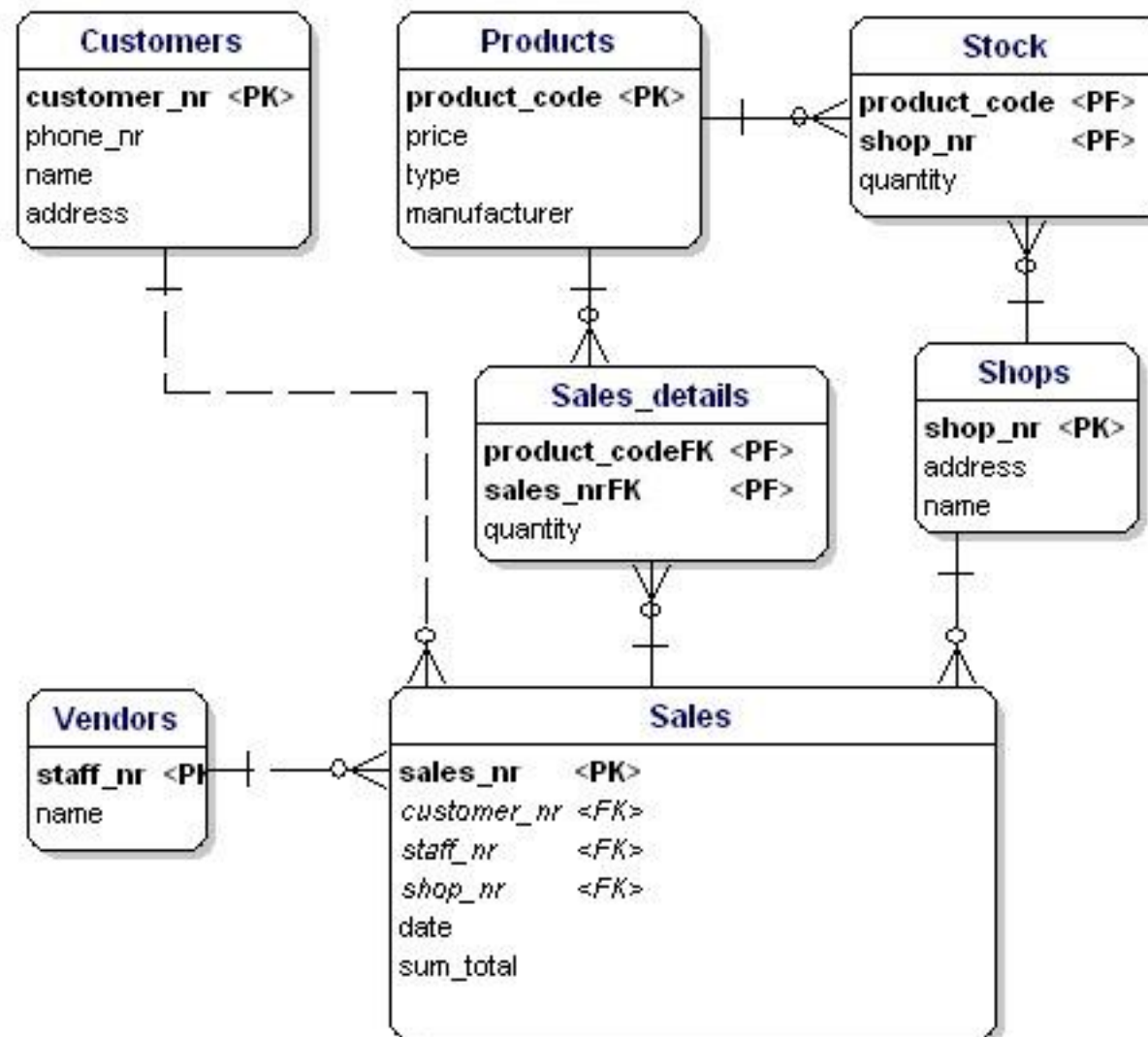
What are each entities' PRIMARY KEY? (response format:
table - attribute type)

Top



Start the presentation to see live content. For screen share software, share the entire screen. Get help at padluc.com/app

Keys Assignment (II)



Primary keys and foreign keys.

Keys Assignment (III)

MyISAM versus Innodb

The following table provides a brief comparison of the engine types. The abbreviation ACID stands for Atomicity, Consistency, Isolation, Durability.

MyISAM	Innodb
Not ACID-compliant and non-transactional	ACID-compliant and hence fully transactional with ROLLBACK and COMMIT and support for Foreign Keys
MySQL 5.0 default engine	Rackspace Cloud default engine
Offers compression	Offers compression
Requires full repair and rebuild of indexes and tables	Provides automatic recovery from crashes via the replay of logs
Changed database pages written to disk instantly	Dirty pages converted from random to sequential before commit and flush to disk
No ordering in storage of data	Row data stored in pages in PK order
Table-level locking	Row-level locking

Attributes Data Types (I)

Text:

- CHAR(length) - includes text (characters, numbers, punctuations...). CHAR saves a fixed amount of positions.
- VARCHAR(length) - includes text (characters, numbers, punctuation...). VARCHAR is the same as CHAR, the difference is that VARCHAR only takes as much space as necessary.
- TEXT - can contain large amounts of text. Depending on the type of database this can add up to gigabytes.

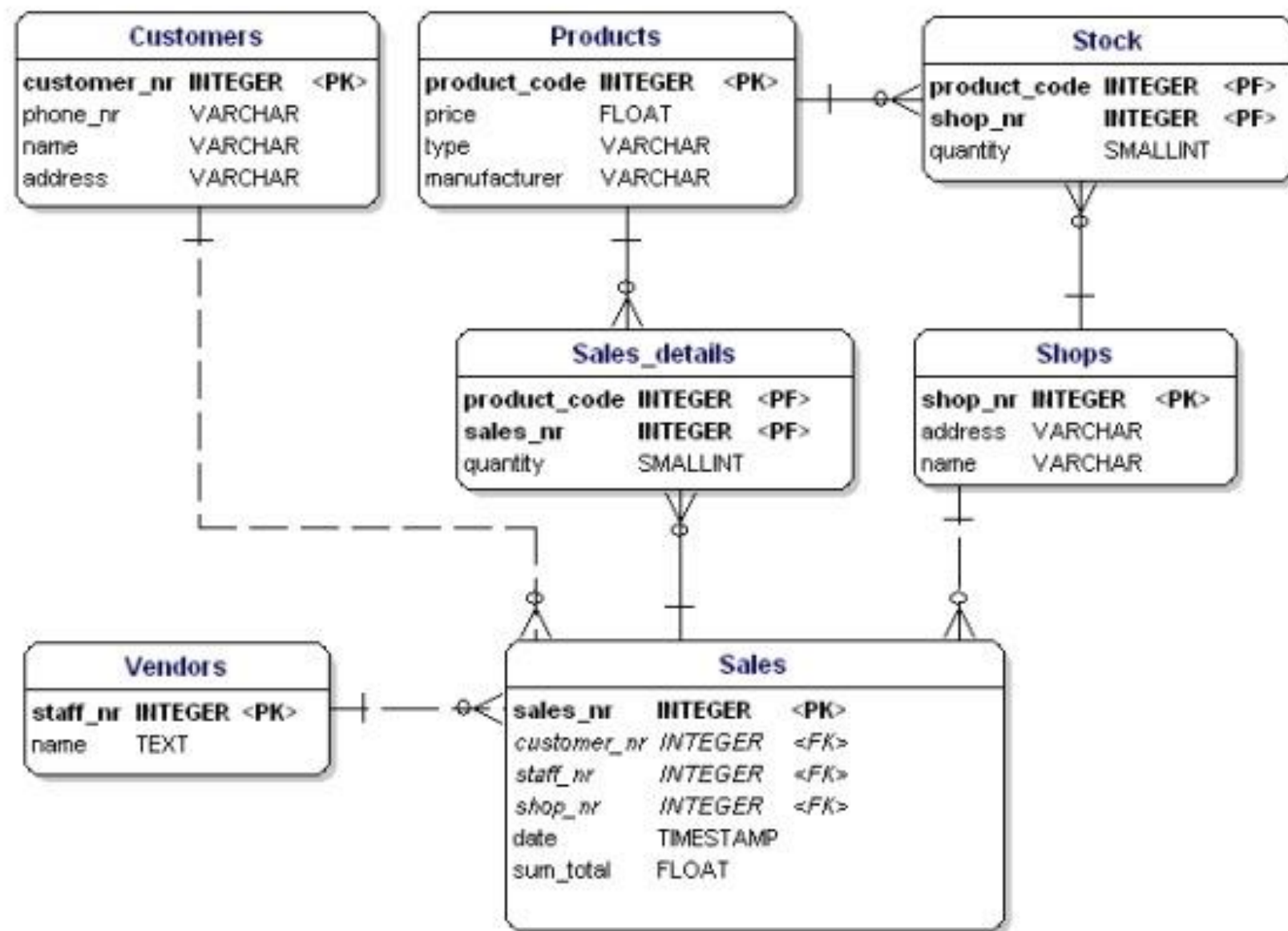
Numbers:

- INT - contains a positive or negative whole number. Variations: TINYINT, SMALLINT, MEDIUMINT, BIGINT...
 - INT is 4 bytes : -2147483647 to +2147483646, **UNSIGNED** from 0 to 4294967296.
 - INT8, or BIGINT, 0 to 18446744073709551616, but takes up to 8 bytes of disk space
- FLOAT, DOUBLE - MySQL calculating with floating point numbers is not perfect, $(1/3)*3$ will result in 0.9999999, not 1.

Other types:

- BLOB - for binary data such as (serialized) files.
 - INET - for IP addresses. Also useable for netmasks.
-

Attributes Data Types (II)



Data model displaying data types.

For each PRIMARY KEY, what data type would you assign?
(response format: entity-attribute type-data type)

Top

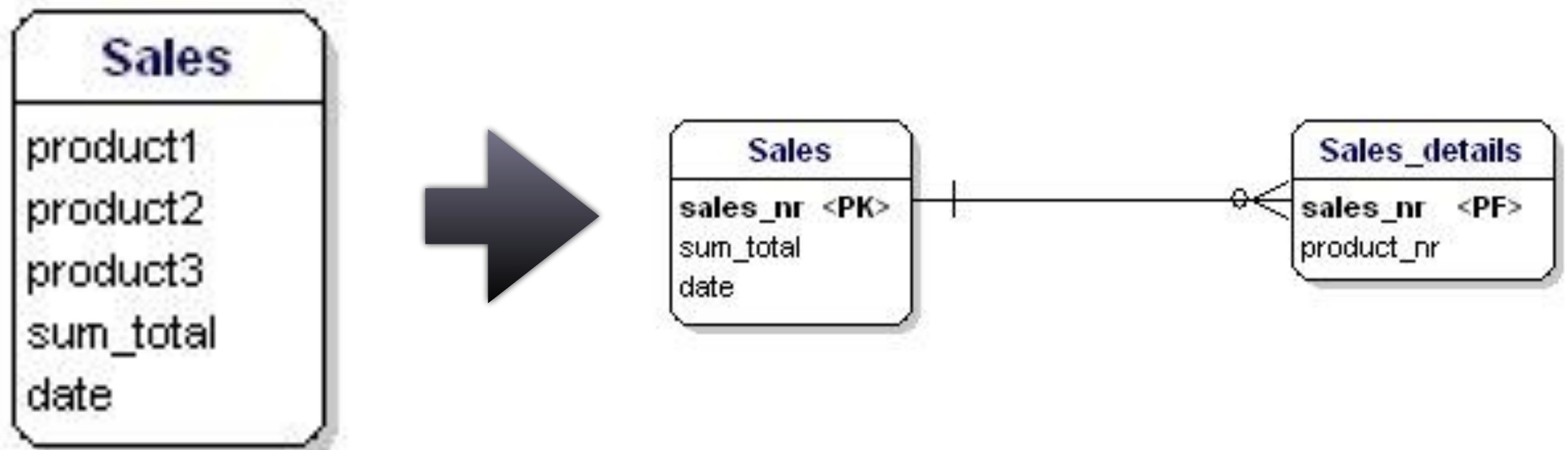


Start the presentation to see live content. For screen share software, share the entire screen. Get help at padluc.com/app

Normalization (I)

- **Normalization makes your data model flexible and reliable. It does generate some overhead because you usually get more tables, but it enables you to do many things with your data model without having to adjust it.**
-

Normalization (I)



1st normal form:

no repeating groups of columns in an entity

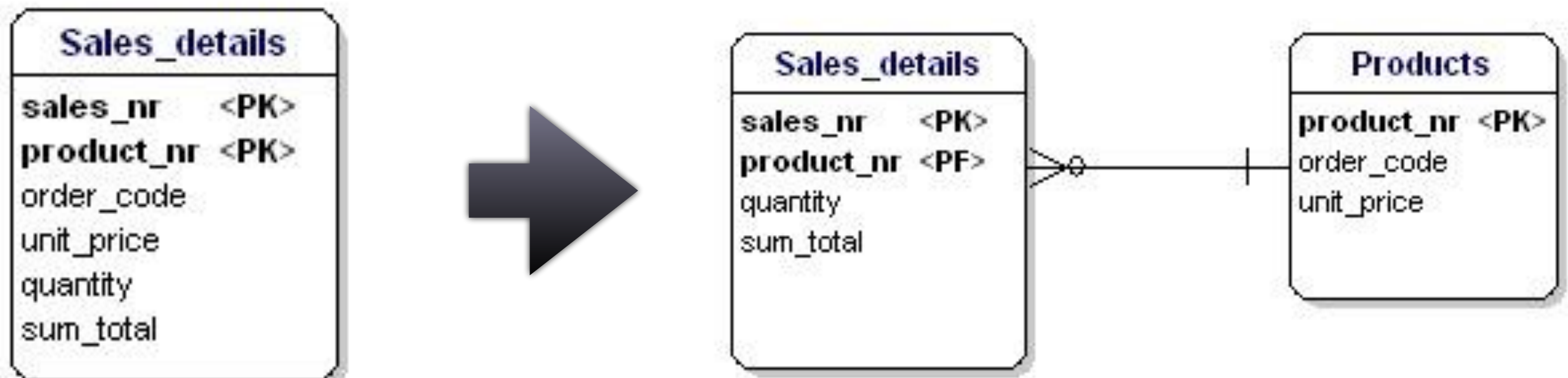
Normalization (II)



2nd normal form:

all attributes of an entity should be fully dependent on the whole primary key.

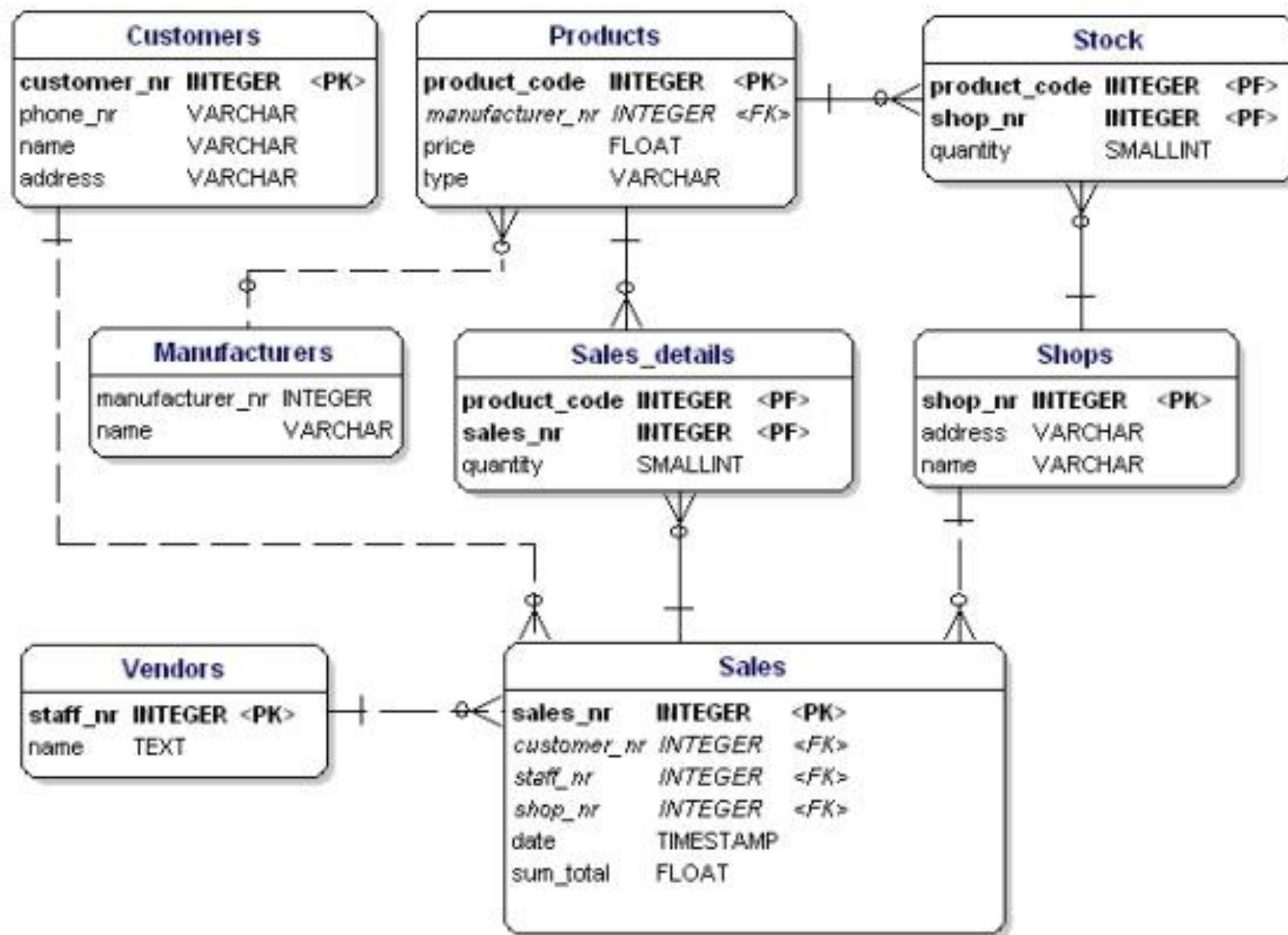
Normalization (III)



3rd normal form:

all attributes need to be directly dependent on the primary key, and not on other attributes.

Normalization (IV)



Data model in accordance with 1st, 2nd and 3d normal form.

Questions?

Please take the 5 questions survey:
dartgo.org/poll

Workshop feedback

When survey is active, respond at [PollEv.com/darttc](https://poll-ev.com/darttc)

0 surveys done



0 surveys in progress

Start the presentation to see live content. For screen-share software, share the entire screen. Get help at poll-ev.com/app



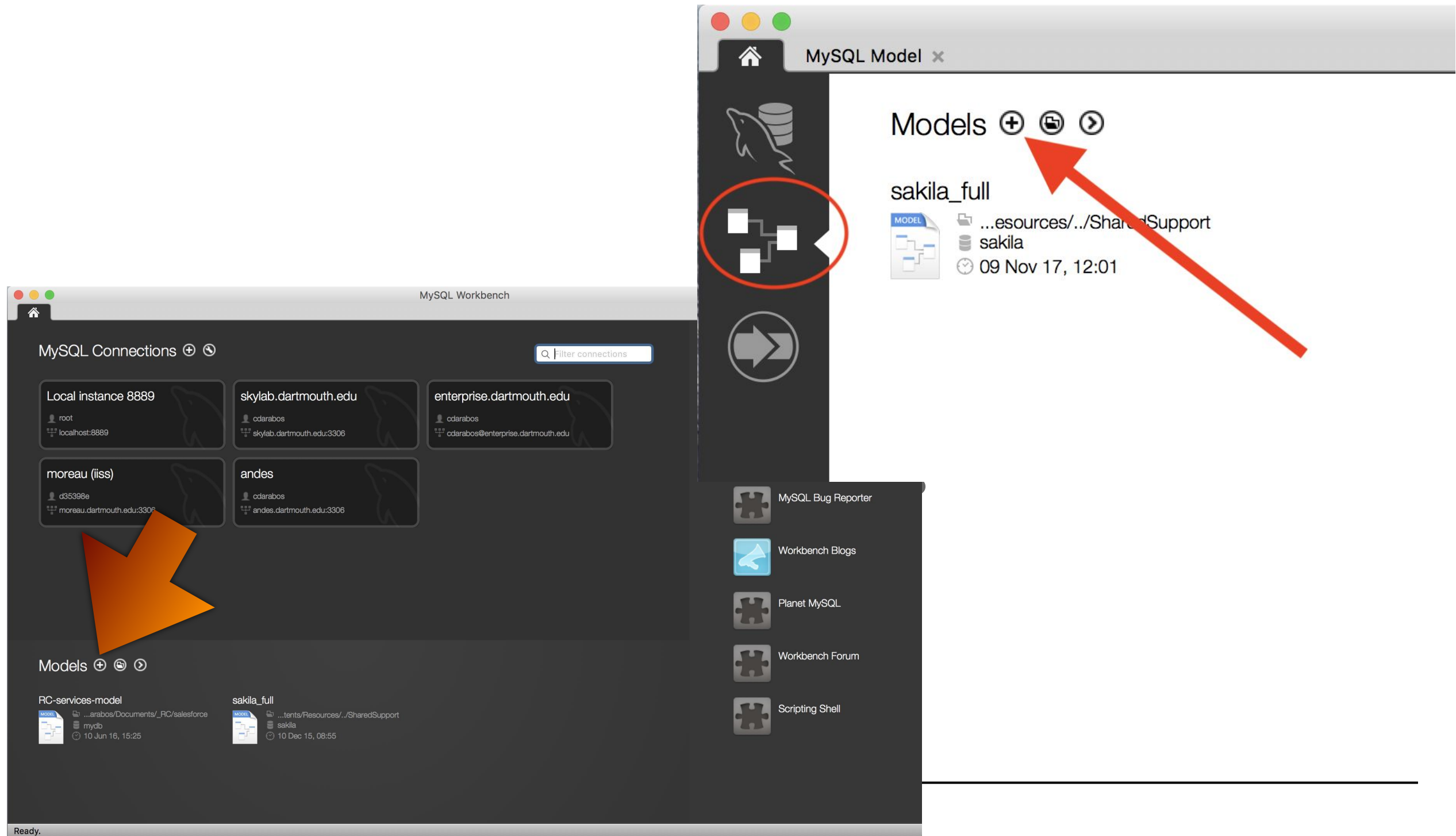
Use MySQL Workbench

- Use a new Model in MySQL Workbench
 - Declare new tables, attributes, and relationships
 - Including datatypes, primary/foreign key
-

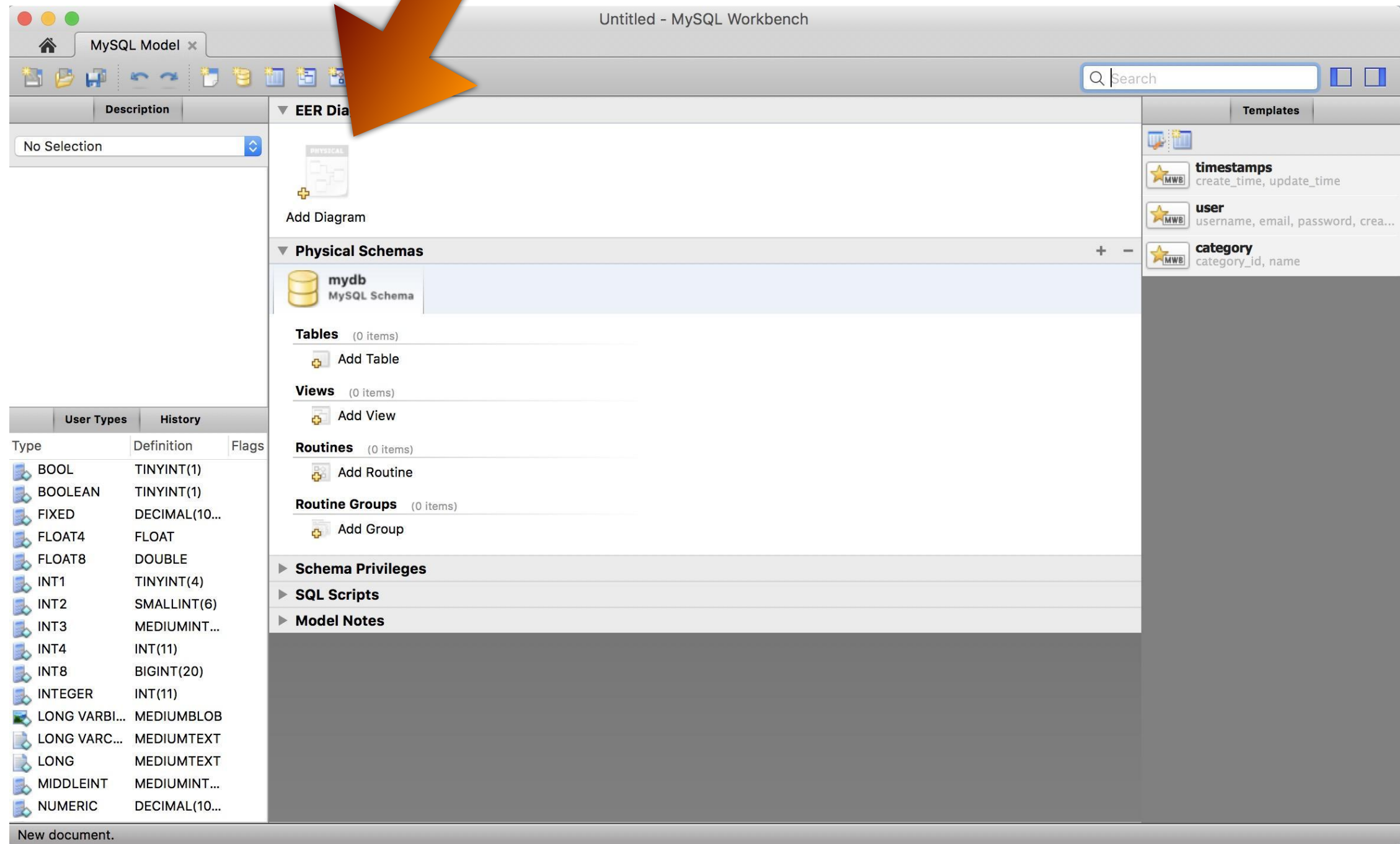
Generate DB/Schema

- from the EER Diagram, MySQL Workbench can generate the SQL script and submit it to the server to create the DB/Schema and all the tables (incl. attributes and properties)
 - select menu Database->Forward Engineer...
 - follow wizard steps
 - You can see the generate MySQL code to import in your DB manager
 - newly created DBs have no content
-

Environment Setup (I)



Environment Setup (II)



The screenshot shows the MySQL Workbench interface. A large orange arrow points to the 'EER Diagram' tab in the central workspace. The interface includes a top toolbar, a left sidebar with a 'Description' pane and a 'User Types' pane, and a right sidebar with a 'Templates' pane. The central workspace is divided into 'EER Diagram' and 'Physical Schemas' sections. The 'Physical Schemas' section shows a 'mydb' MySQL Schema with sub-sections for Tables, Views, Routines, and Routine Groups. The 'User Types' pane on the left lists various data types and their definitions.

Untitled - MySQL Workbench

MySQL Model x

Description

No Selection

EER Diagram

Physical Schemas

mydb
MySQL Schema

Tables (0 items)

Add Table

Views (0 items)

Add View

Routines (0 items)

Add Routine

Routine Groups (0 items)

Add Group

Schema Privileges

SQL Scripts

Model Notes

Templates

timestamps
create_time, update_time

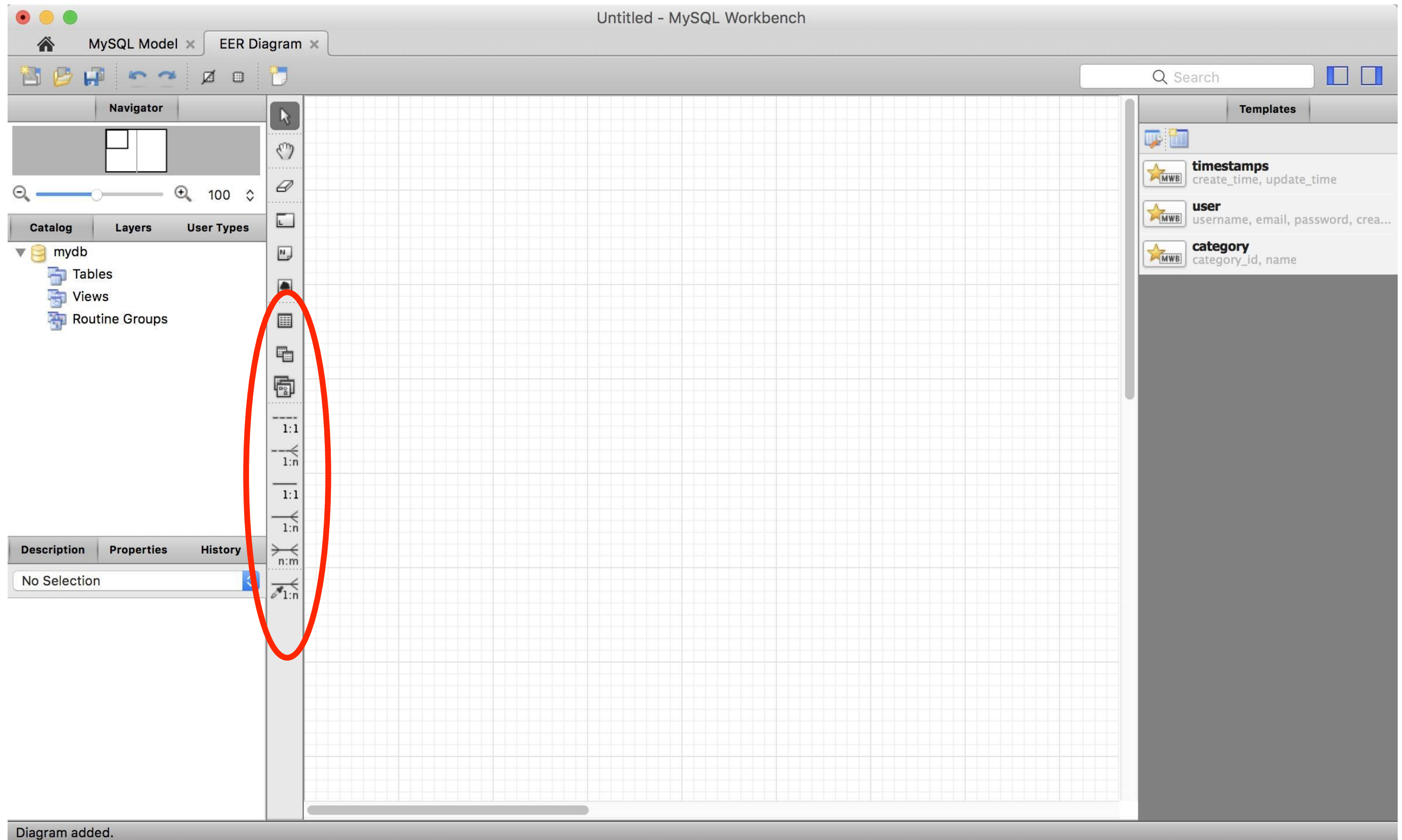
user
username, email, password, crea...

category
category_id, name

Type	Definition	Flags
BOOL	TINYINT(1)	
BOOLEAN	TINYINT(1)	
FIXED	DECIMAL(10...	
FLOAT4	FLOAT	
FLOAT8	DOUBLE	
INT1	TINYINT(4)	
INT2	SMALLINT(6)	
INT3	MEDIUMINT...	
INT4	INT(11)	
INT8	BIGINT(20)	
INTEGER	INT(11)	
LONG VARBI...	MEDIUMBLOB	
LONG VARC...	MEDIUMTEXT	
LONG	MEDIUMTEXT	
MIDDLEINT	MEDIUMINT...	
NUMERIC	DECIMAL(10...	

New document.

Environment Setup (III)



Environment Setup

(I)

- launch MAMP (skip MAMP Pro)
 - click on Start Servers
-

MySQL

MySQL can be administered v

To connect to the MySQL ser

Host	localhost
Port	3306
User	root
Password	root

The screenshot shows the MySQL Workbench interface with the 'Setup New Connection' dialog box open. The dialog box has several fields and tabs. Numbered callouts point to specific elements:

- 1: Points to the 'MySQL Connections' button in the top left of the main window.
- 2: Points to the 'Connection Name' field, which contains 'localhost'.
- 3: Points to the 'Username' field, which contains 'root'.
- 4: Points to the 'Password' field, which contains 'root'.
- 5: Points to the 'Test Connection' button at the bottom right of the dialog box.
- 6: Points to the 'OK' button at the bottom right of the dialog box.

The dialog box also includes a 'Connection Method' dropdown set to 'Standard (TCP/IP)', a 'Hostname' field with '127.0.0.1', a 'Port' field with '3306', and a 'Default Schema' field. The 'Parameters' tab is selected, and the 'Advanced' sub-tab is active. The 'Test Connection' button is highlighted with a blue border.

Environment Setup (II)

- start “Local instance XXXX”
- username: root — password
- select File -> New Model
- under “Physical Schemas” right-click and rename the “mydb” schema tab to “myShops”
- under “EER Diagrams, double click on “Add Diagram”

MySQL

MySQL can be administered 'MySQL

To connect to the MySQL server

Host	localhost
Port	3306
User	root
Password	root

Announcements

More RC workshops:

- <https://rc.dartmouth.edu> > Training

Support:

- christian.darabos@dartmouth.edu
 - research.computing@dartmouth.edu
-