

# Relational Databases

**February 8, 2018**

# Data Science Roles

ROLES	RESPONSIBILITIES
<b>Data Architect</b> 	<p>Develops data architecture to effectively capture, integrate, organize, centralize and maintain data. Core responsibilities include:</p> <ul style="list-style-type: none"> <li>✓ Data Warehousing Solutions</li> <li>✓ Extraction, Transformation and Load (ETL)</li> <li>✓ Data Architecture Development</li> <li>✓ Data Modeling</li> </ul>
<b>Data Engineer</b> 	<p>Develop, test and maintain data architectures to keep data accessible and ready for analysis. Key tasks are:</p> <ul style="list-style-type: none"> <li>✓ Extraction Transformation and Load (ETL)</li> <li>✓ Installing Data Warehousing Solutions</li> <li>✓ Data Modeling</li> <li>✓ Data Architecture Construction and Development</li> <li>✓ Database Architecture Testing</li> </ul>
<b>Data Analyst</b> 	<p>Processes and interprets data to get actionable insights for a company. Responsibilities include:</p> <ul style="list-style-type: none"> <li>✓ Data Collection and Processing</li> <li>✓ Programming</li> <li>✓ Machine Learning</li> <li>✓ Data Munging</li> <li>✓ Data Visualization</li> <li>✓ Applying Statistical Analysis</li> </ul>
<b>Data Scientist</b> 	<p>Data analysis once data volume and velocity reaches a level requiring sophisticated technical skills. Core tasks are:</p> <ul style="list-style-type: none"> <li>✓ Data Cleansing and Processing</li> <li>✓ Predictive Modeling</li> <li>✓ Machine Learning</li> <li>✓ Identifying Questions</li> <li>✓ Running Queries</li> <li>✓ Applying Statistical Analysis</li> <li>✓ Correlating Disparate Data</li> <li>✓ Storytelling and Visualization</li> </ul>

Sources:  
 KDnuggets - [www.kdnuggets.com/2015/11/different-data-science-roles-industry.html](http://www.kdnuggets.com/2015/11/different-data-science-roles-industry.html)  
 Udacity - [blog.udacity.com/2014/12/data-analyst-vs-data-scientist-vs-data-engineer.html](http://blog.udacity.com/2014/12/data-analyst-vs-data-scientist-vs-data-engineer.html)  
 RMetrics - [rmetrics.com/resources/reports/the-state-of-data-science/](http://rmetrics.com/resources/reports/the-state-of-data-science/)



# Course Outline

- **Relational databases**

- Relational model
- Relationships
- Constraints
- Indexing
- Stored procedures
- Normalization

- **SQL**

- History / Alternatives
- Design
- Syntax

- **Data Marts**

- OLAP Vs. OLTP
- Star schema
- Snowflake schema

- **Tidy data**

- Principles
- Benefits
- Tidying messy data

# Relational Databases

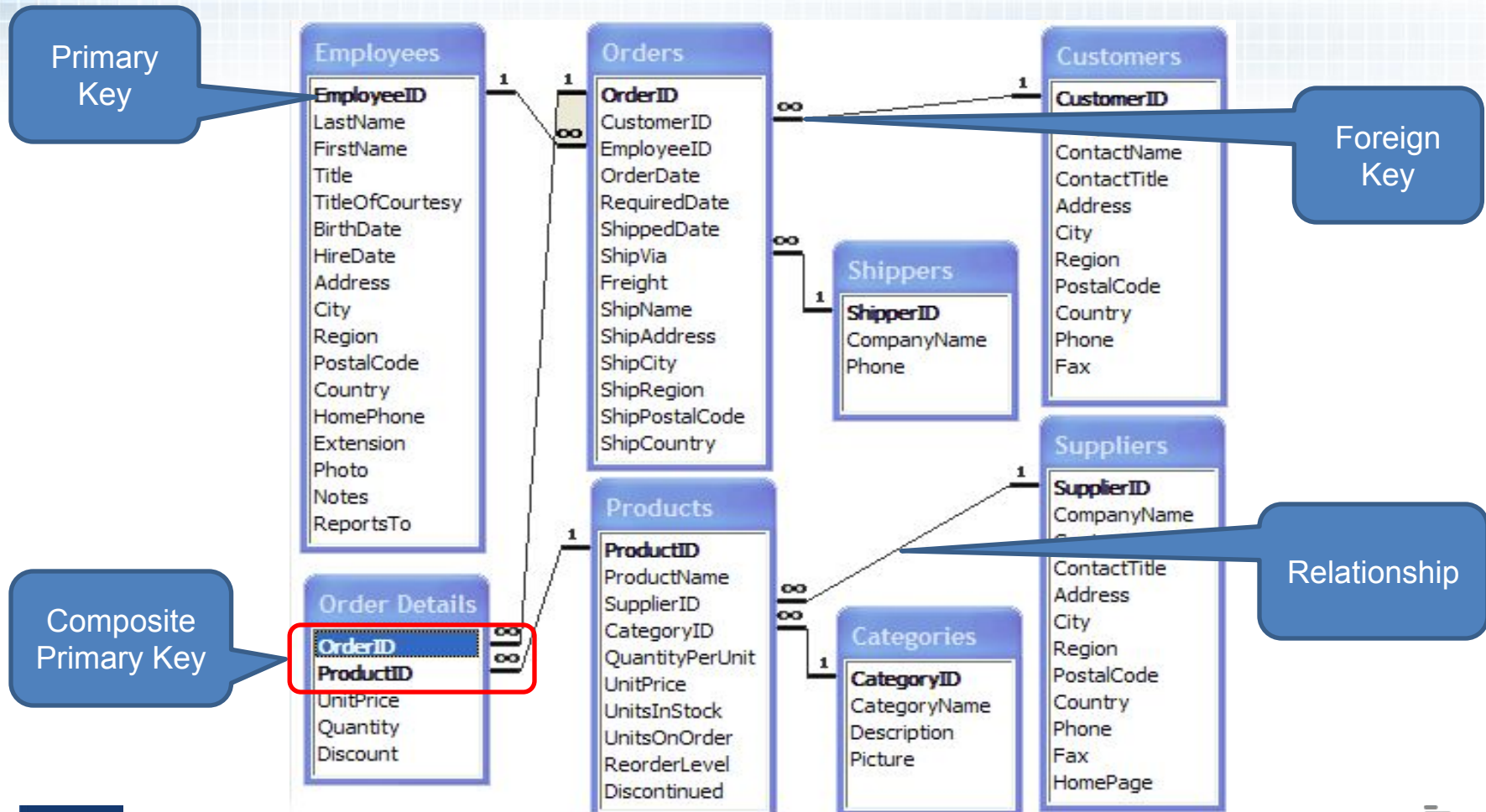
- Relational databases like MySQL, PostgreSQL and SQLite3 represent and store data in tables and rows.
- Relational databases use Structured Querying Language (SQL)
  - Good for applications that involve the management of several transactions
- The structure of a relational database allows you to link information from different tables through the use of foreign keys, which are used to uniquely identify any atomic piece of data within that table.
- Other tables may refer to that foreign key, so as to create a link between their data pieces and the piece pointed to by the foreign key.
  - Comes in handy for applications that are heavy into data analysis.

# Relational Databases (Cont.)

- A relational database at its simplest is a set of tables used for storing data. Each table has a unique name and may relate to one or more other tables in the database through common values.
- A table in a database is a collection of rows and columns. Tables are also known as entities or relations.
- A row contains data pertaining to a single item or record in a table. Rows are also known as records or tuples.
- A column contains data representing a specific characteristic of the records in the table. Columns are also known as fields or attributes.
- A relationship is a link between two tables (i.e., relations). Relationships make it possible to find data in one table that pertains to a specific record in another table.



# Example of Relational Database



# Datatypes

- **Each of a table's columns has a defined datatype that specifies the type of data that can exist in that column, for example:**

- **String**

- Variable Character (can define character set, i.e. ASCII)
- BLOB (Binary Large Object)
- Computer code, e.g. JSON
- Text, etc.

- **Numeric can be in a form of**

- Integer (small, big, medium)
- Double (fixed, floating)
- Large Numeric

- **Logical**

- Boolean

- **Various formats for date and time**

- **Unfortunately, datatypes vary widely between databases and analytical tools**

# Constraints

- **Constraints are used to specify rules for the data in a table**
- **Constraints are used to limit the type of data that can go into a table to ensure the accuracy and reliability**
- **Constraints can be column level or table level**
- **The commonly used constraints are:**
  - NOT NULL - Ensures that a column cannot have a NULL value
  - UNIQUE - Ensures that all values in a column are different
  - PRIMARY KEY - A combination of a NOT NULL and UNIQUE Uniquely identifies each row in a table
  - FOREIGN KEY - Uniquely identifies a row/record in another table
  - CHECK - Ensures that all values in a column satisfies a specific condition
  - DEFAULT - Sets a default value for a column when no value is specified
  - INDEX - Used to create and retrieve data from the database very quickly



# Indexing

- **Indexes are used to retrieve data from the database very fast**
- **The users cannot see the indexes, they are just used to speed up searches/queries**
- **Updating a table with indexes takes more time than updating a table without (because the indexes also need an update)**
- **Create indexes on columns that will be frequently searched against!**
- **Indexes can be unique or not unique**
  - **Recommend unique indexes**

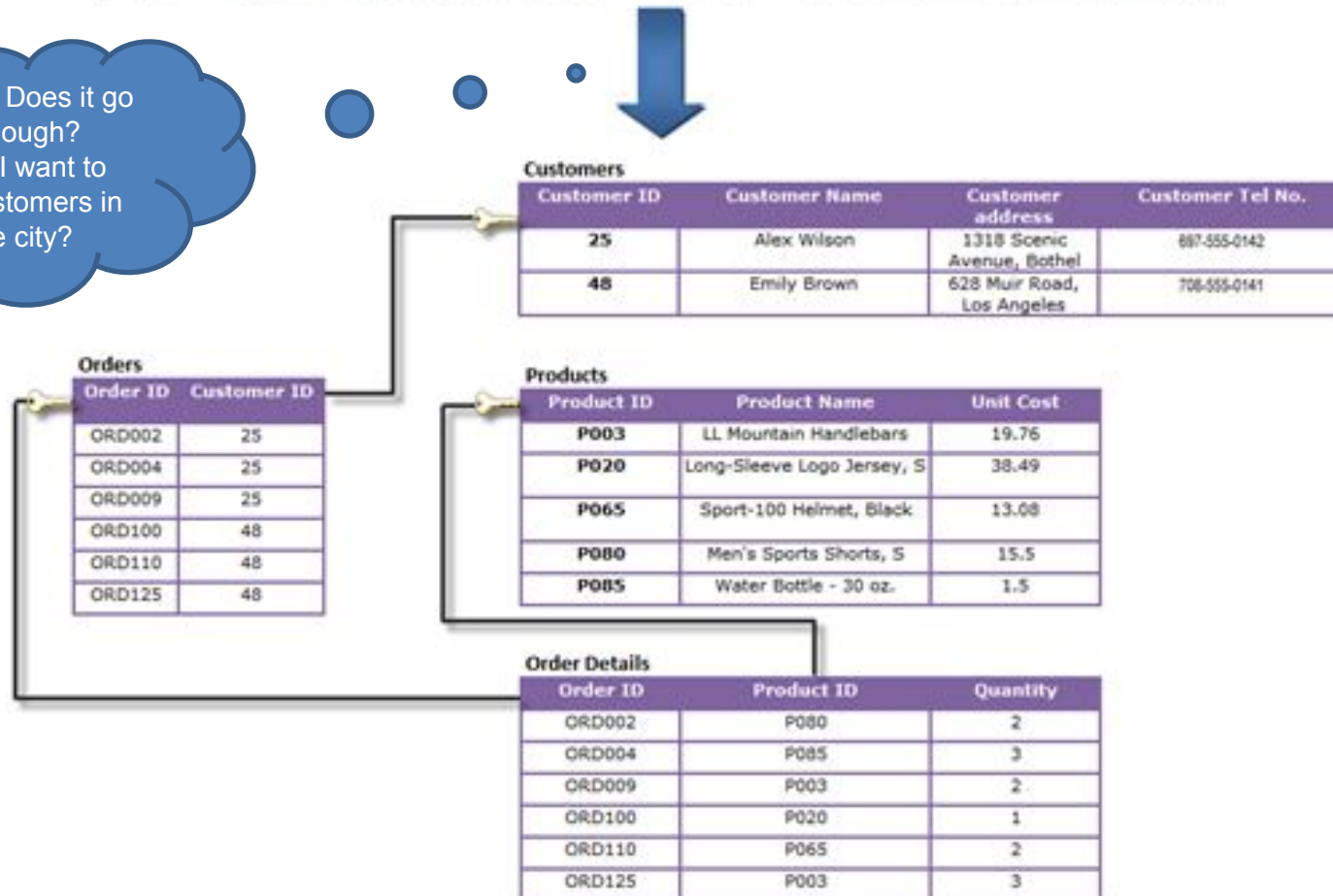
# Database Normalization

- **The concept of Normalization was introduced in 1969 by Edgar F. Codd as an integral part of his relational model**
- **Basic objective was to permit data to be queried and manipulated using a "universal data sub-language" grounded in first-order logic ("If X is Socrates and X is a man, then Socrates is a man")**
- **Has multiple states / forms**
- **Objectives of First Normal Form**
  - **Free the collection of relations from undesirable insertion, update and deletion dependencies**
  - **Reduce the need for restructuring the collection of relations, as new types of data are introduced, and thus increase the life span of application programs**
  - **Make the relational model more informative to users**
  - **Make the collection of relations neutral to the query statistics, i.e. query performance measurements**

# Database Normalization Example

Customer Name	Customer Address	Customer Tel No.	Product Name	Unit Cost	Quantity	Total Cost
Alex Wilson	1318 Scenic Avenue, Bothel	697-555-0142	Men's Sports Shorts, S	15.5	2	31
Alex Wilson	1318 Scenic Avenue, Bothel	697-555-0142	Water Bottle - 30 oz	1.5	3	4.5
Alex Wilson	1318 Scenic Avenue, Bothel	697-555-0142	LL Mountain Handlebars	19.76	2	39.52
Emily Brown	628 Muir Road, Los Angeles	708-555-0141	Long-Sleeve Logo Jersey, S	38.49	1	38.49
Emily Brown	628 Muir Road, Los Angeles	708-555-0141	Sport-100 Helmet, Black	13.08	2	26.16
Emily Brown	628 Muir Road, Los Angeles	708-555-0141	LL Mountain Handlebars	19.76	3	59.28

Question: Does it go far enough?  
What if I want to know customers in same city?



# Popular Databases

## ▪ Commercial

- Oracle is the most popular relational database. It runs on both Unix and Windows. It used to be many times more expensive than SQL Server and DB2, but it has come down a lot in price.
- SQL Server is Microsoft's database and, not surprisingly, only runs on Windows. It has only a slightly higher market share than Oracle on Windows machines. Many people find it easier to use than Oracle.
- IBM's DB2 was one of the earliest players in the database market. It is still very commonly used on mainframes and runs on both Windows and Unix.

## ▪ Popular Open Source Databases

- Until recently, PostgreSQL was the most popular open source database until that spot was taken over by MySQL. It is certainly a featureful and robust database management system and a good choice for people who want some of the advanced features that MySQL doesn't yet have.
- Because of its small size, its speediness, and its very good documentation, MySQL has quickly become the most popular open source database. MySQL is available on both Windows and Unix. It catches up with PostgreSQL functionality.
- DSVD is using MySQL

# Brief History of SQL

## Structured Query Language

- In 1970, E. F. Codd published "A Relational Model of Data for Large Shared Data Banks," an article that outlined a model for storing and manipulating data using tables
- Shortly after, IBM began working on creating a relational database
- Between 1979 and 1982, Oracle (then Relational Software, Inc.), Relational Technology, Inc. (later acquired by Computer Associates), and IBM all put out commercial relational databases
- By 1986 they all were using SQL as the data query language.
- In 1986, the American National Standards Institute (ANSI) standardized SQL
  - This standard was updated in 1989, in 1992 (called SQL2)
  - In 1999 called SQL3
  - In 2003 called SQL 2003
  - In 2006 called SQL 2006
  - In 2008 called SQL 2008
- Standard SQL is sometimes called ANSI SQL. All major relational databases support this standard but each has its own proprietary extensions



# SQL Statements

- **Database Manipulation Language (DML) statements are used to work with data in an existing database. The most common DML statements are:**
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
- **Database Definition Language (DDL) statements are used to structure objects in a database. The most common DDL statements are:**
  - CREATE
  - ALTER
  - DROP
- **Database Control Language (DCL) statements are used for database administration. The most common DCL statements are:**
  - GRANT
  - DENY (SQL Server Only)
  - REVOKE

# Some Basics

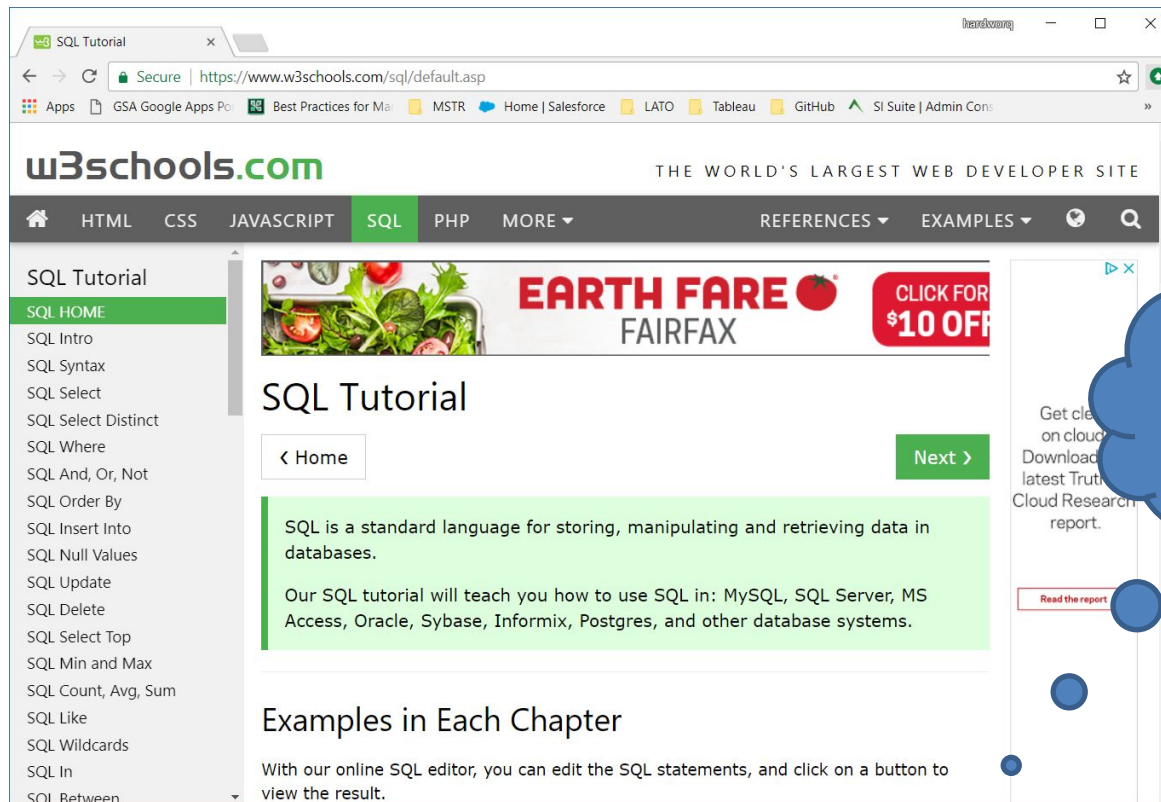
- Comments: the standard SQL comment is two hyphens (--). However, some databases use other forms of comments as shown in the table below.

➤ Example	-- Comment	# Comment	/* Comment */
➤ ANSI	YES	NO	NO
➤ SQL Server	YES	NO	YES
➤ Oracle	YES	NO	YES
➤ MySQL	YES	YES	YES

- Whitespace is ignored in SQL statements. Multiple statements are separated with semi-colons. The two statements in the sample below are equally valid.
  - SELECT \* FROM Employees;
  - SELECT \*  
FROM Employees;
- SQL is not case sensitive. It is common practice to write reserved words in all capital letters. User-defined names, such as table names and column names may or may not be case sensitive depending on the operating system used.

# How to Learn SQL

- <https://www.webucator.com/tutorial/learn-sql/simple-selects/introduction-the-northwind-database-reading.cfm#tutorial>
  - Uses Microsoft Northwind database incl. in Access
- <https://www.w3schools.com/sql/default.asp>
  - More inclusive: offers MySQL, Oracle, and MS Access specifics



You will get a certificate after passing a quiz

# MySQL Workbench

- MySQL is an open source relational database that is cross platform
- MySQL supports multiple storage engines which greatly improve the server performance tuning and flexibility
- MySQL server can be administered using a number of server access mysql tools which include both commercial and open source products:
  - phpMyAdmin - cross platform web based open source server access tool
  - SQLYog - targeted at the windows platform, desktop commercial server access tool
  - MySQL workbench - cross platform open source server access tool.
- MySQL workbench is an integrated development environment for MySQL server
- It has utilities for database modeling and designing, SQL development and server administration
- MySQL workbench is included in DSVD
- <https://www.guru99.com/introduction-to-mysql-workbench.html>

# MySQL Workbench Tutorial

SQL Tutorial

SQL Tryit Editor v1.5

MySQL Workbench Tutorial

Secure | https://www.guru99.com/introduction-to-mysql-workbench.html


Apps | GSA Google Apps Po | Best Practices for Mail | MSTR | Home | Salesforce | LATO | Tableau | GitHub | SI Suite | Admin Cons

GURU99

Home | Testing | SAP | Web | Must Learn! | Big Data | Live Projects


Blog

## MySQL Workbench Tutorial & MySQL Introduction



**Google Home.**  
Hands-free help from the Google Assistant.

Shop now



### What is MySQL?


MySQL is an open source relational database.



MySQL is cross platform which means it runs on a number of different platforms such as Windows, Linux, and Mac OS etc.

**In this tutorial, you will learn-**

- What is MySQL?
- Why use MySQL?

- Introducing MySQL Workbench
- MySQL workbench- Modeling and Design tool
- MySQL workbench - SQL development tool
- MySQL workbench - Administration tool
- Install MySQL workbench Guide

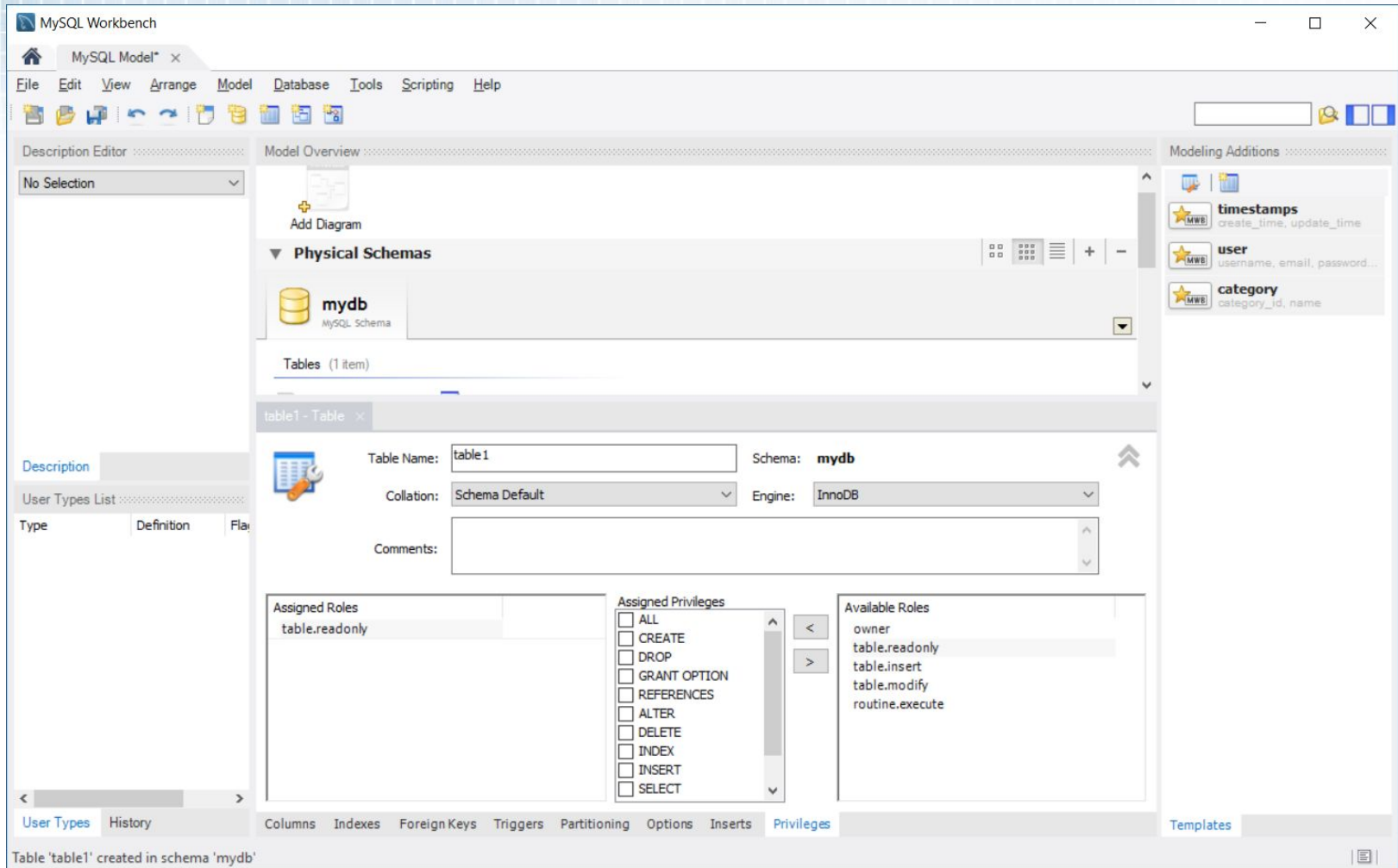


Connect your application quickly to billions of data points across the globe.



# MySQL Workbench Desktop

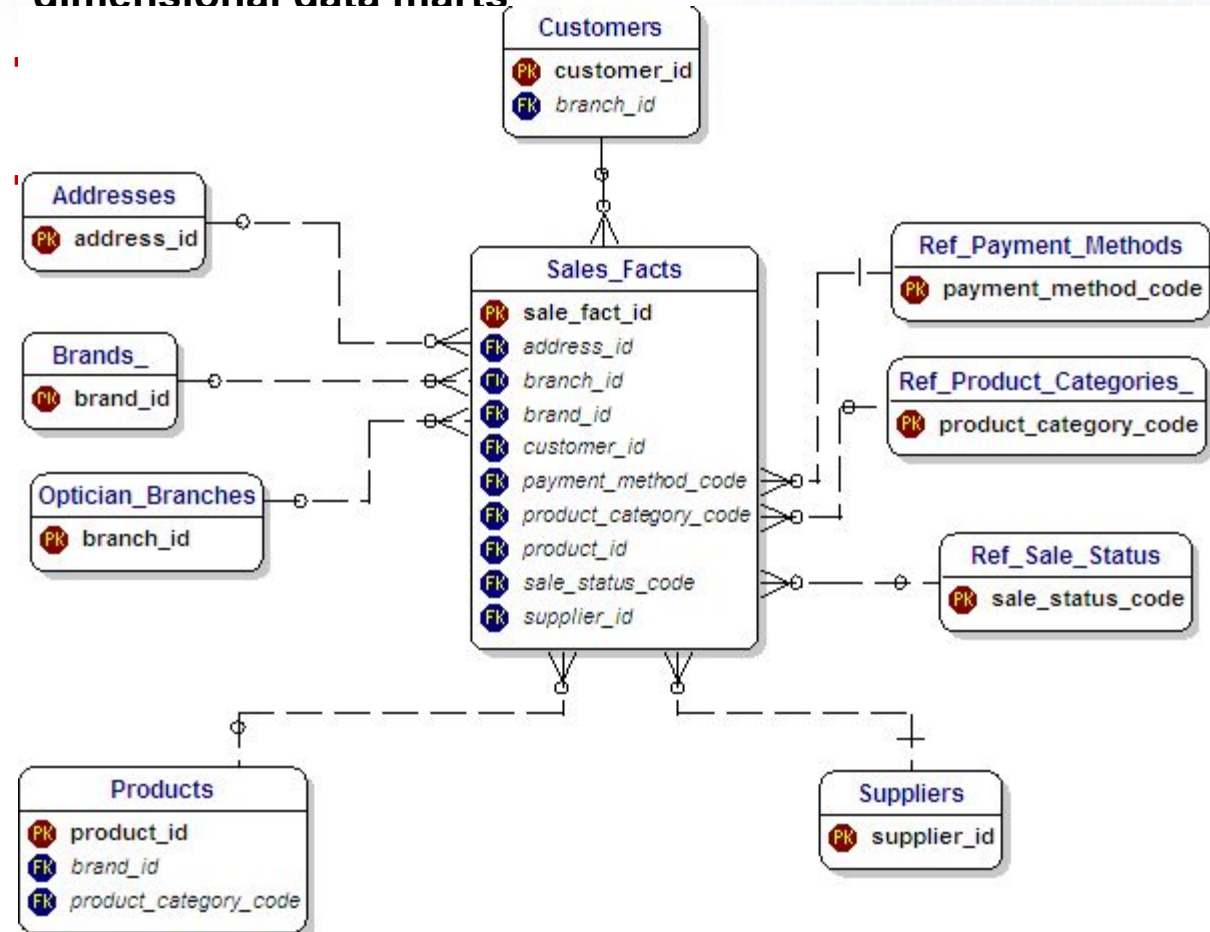


# OLTP Vs. OLAP

- **What is a prime use of your database?**
  - Operational – OLTP, or
  - Analytical – OLAP
- **OLTP (On-line Transaction Processing)**
  - Large number of short on-line transactions (INSERT, UPDATE, DELETE)
  - The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second.
  - OLTP database is used to store transactional databases is the entity model
- **OLAP (On-line Analytical Processing)**
  - Relatively low volume of transactions
  - Queries are often very complex and involve aggregations
  - For OLAP systems fast response time is desired
  - OLAP applications are widely used by Data Mining techniques.
  - In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema)

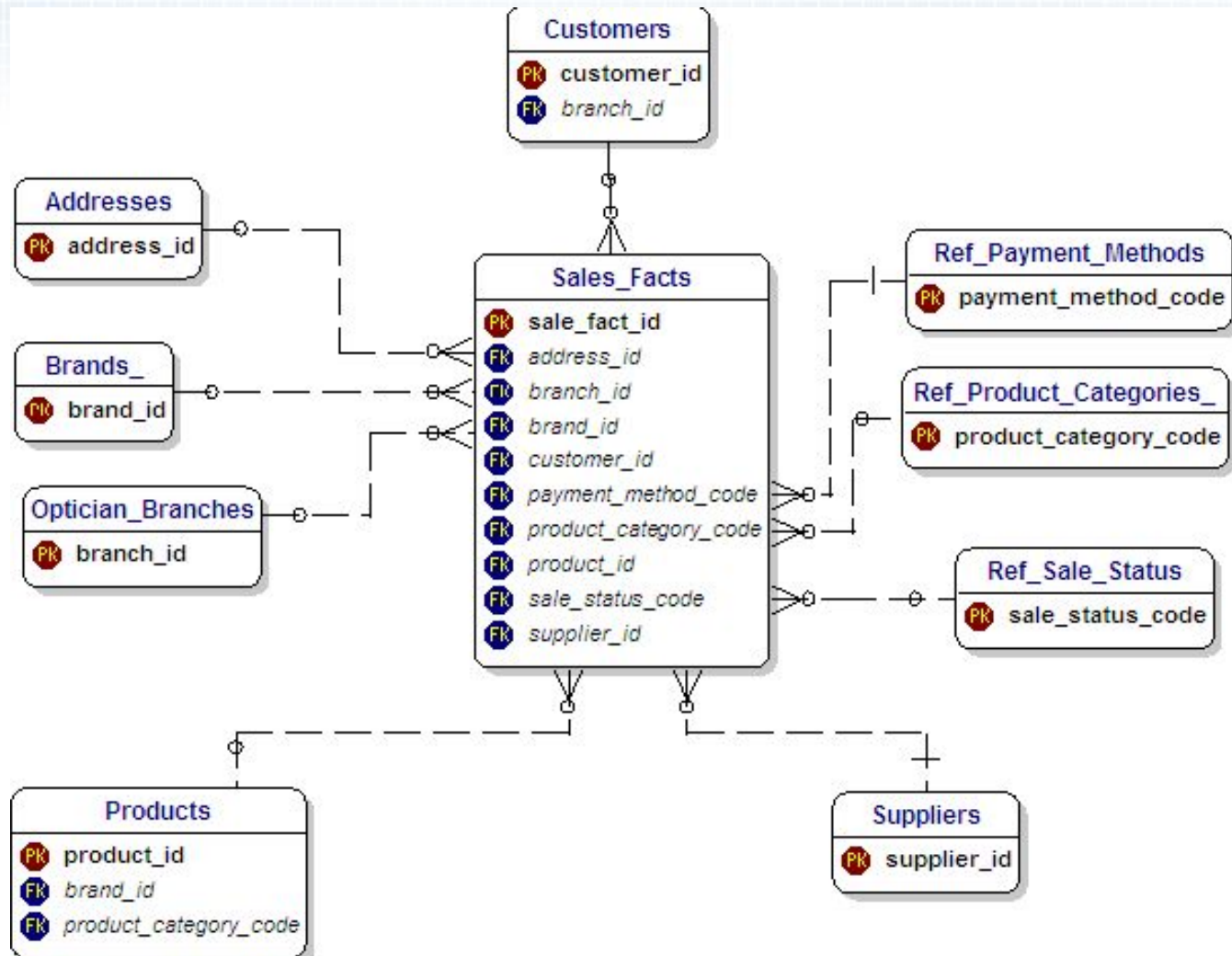
# Star Schema

- Star schema is the simplest style of data mart schema and is the approach most widely used to develop data warehouses and dimensional data marts.



encing

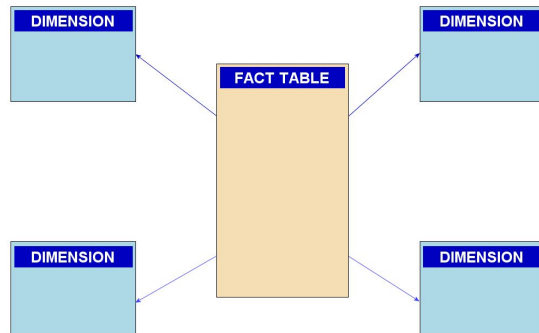
# Star Schema Diagram



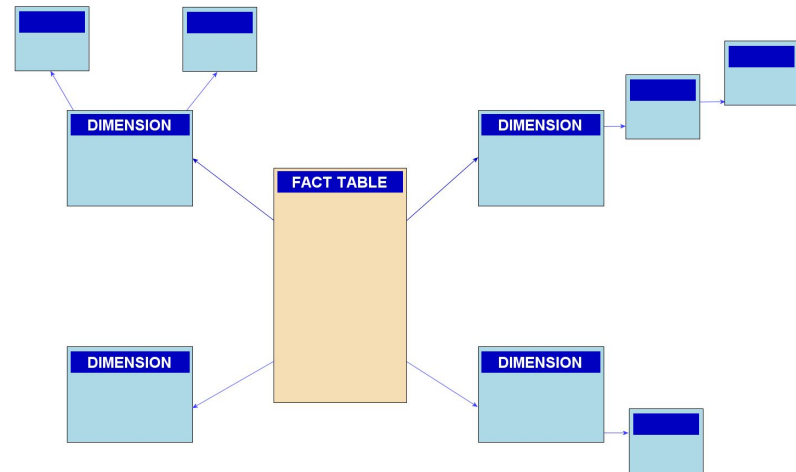
# Snowflake Schema

- The snowflake schema is expansion of the star schema
- In the snowflake schema, dimensions are normalized into multiple related tables, whereas the star schema's dimensions are de-normalized with each dimension represented by a single table

Star Schema



Snowflake Schema





# Snowflake Vs. Star

- **Snowflake is highly normalized**
- **Snowflake better enforces data integrity then star schema**
- **Requires less space**
- **The primary disadvantage of the snowflake schema is that the additional levels of attribute normalization adds complexity to source query joins**
- **Snowflake schemas, in contrast to flat single table dimensions, have been heavily criticized**
- **The goal of an efficient and compact storage of normalized data comes at the significant cost of poor performance when browsing the joins requires down highly normalized dimension**

Review Slide 11

# Tidy Data

- **Tidy data principles**

- Each variable forms a column
- Each observation (tuple) forms a row
- Each type of observational unit (dimension) forms a table

- **Five most common problems with messy datasets**

- Column headers are values, not variable names.
- Multiple variables are stored in one column.
- Variables are stored in both rows and columns.
- Multiple types of observational units are stored in the same table.
- A single observational unit is stored in multiple tables.

# Example of Messy Data

Pew data: relationship between income and religion

1: Columns are not names of the variable

religion	<\$10k		\$10-20k		\$20-30k		\$30-40k		\$40-50k		\$50-75k	
Agnostic	27	34	60	81	76	137						
Atheist	12	27	37	52	35	70						
Buddhist	27	21	30	34	33	58						
Catholic	418	617	732	670	638	1116						
Don't know/refused	15	14	15	11	10	35						
Evangelical Prot	575	869	1064		982	881	1486					
Hindu	1	9	7	9	11	34						
Historically Black Prot		228	244	236	238	197	223					
Jehovah's Witness	20	27	24	24	21	30						
Jewish	19	19	25	25	30	95						

2: Income categories turned into column names resulting in losing info: cannot compute average income per religion

# Tidied Pew Data

religion	income	noOfPeople
Agnostic	<\$10k	27
Agnostic	\$10-20k	34
Agnostic	\$20-30k	60
Agnostic	\$30-40k	81
Agnostic	\$40-50k	76
Agnostic	\$50-75k	137
Agnostic	\$75-100k	122
Agnostic	\$100-150k	109
Agnostic	>150k	84
Agnostic	Don't know/refused	96

Question: what else  
can we do?

# Q & A