Lesson 1: Relational Databases & SQL

December 22, 2016

Audience

- Those who are new to relational DBs and SQL.

Objective

To understand the ***reason for relational DBs***.

To learn about the ***anatomy of relational databases***.

Understand ***relational DB relationships***.

Understand the ***importance of good database design***.

Software & Resources Needed

- Database Server with Interface
  - MS SQL Server with SQL Server Management Studio

  OR

  - MySQL

  OR

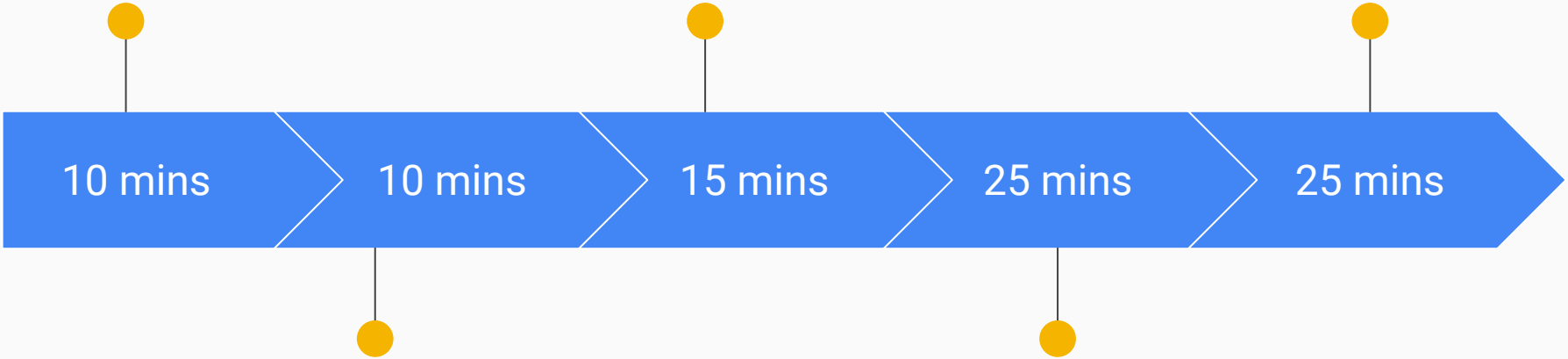
  - Online (sqlfiddle.com)
- Data
  - Test data on Github repo.
- Github repo for information

Overview of Day 1

Operational vs Analytical Databases

Relational DB Anatomy

Good database design



Why Relational Databases?

Relationships (one-to-one, one-to-many, many-to-many)

Operational vs Analytical DBs

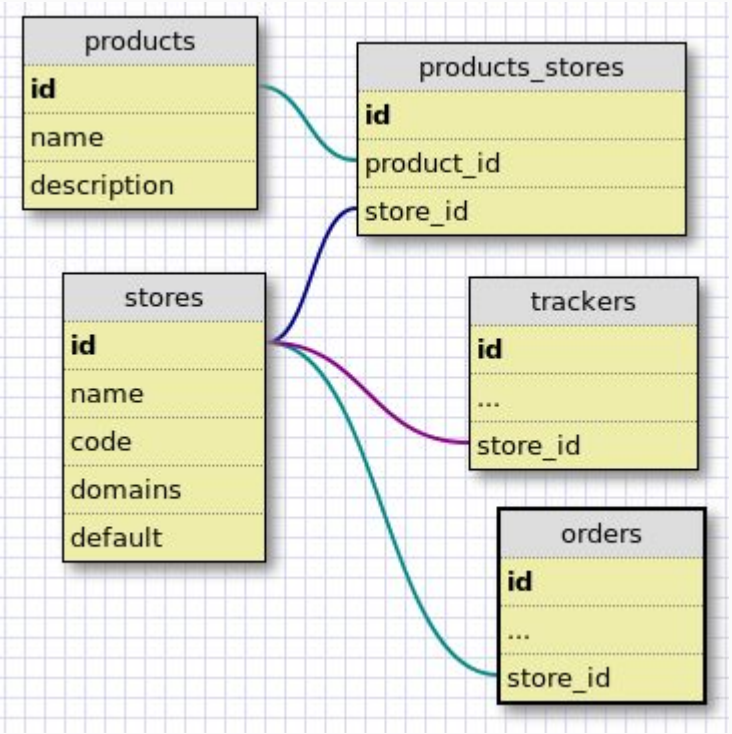
Operational DBs

Also called Transactional DBs.

Relational DBs are part of this category.

- Lots of CRUD operations performed
- Think e-commerce store. Best for dynamic, living data.

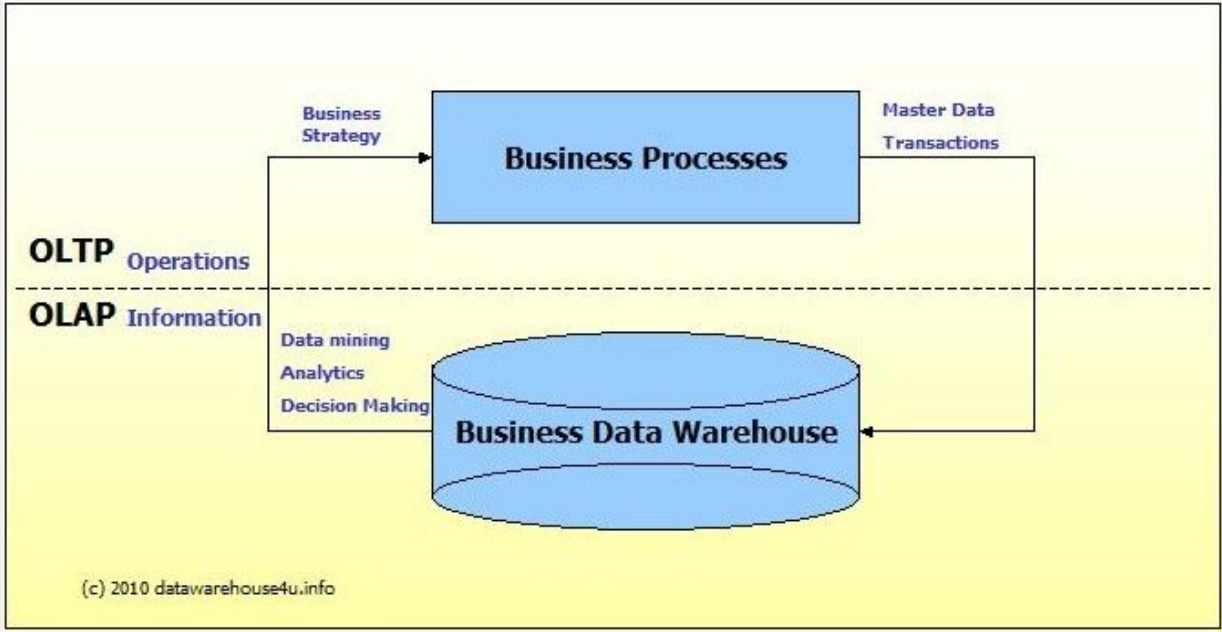
http://datawarehouse4u.info/OLTP-vs-OLAP.html



Analytical DBs

- Think historical information with lots of selects.
- Best for when you report a lot and insert once in awhile, but not for data that's altered more than a rare instance.
- Great for reporting.

http://datawarehouse4u.info/OLTP-vs-OLAP.html



Relational DBs

Relational DBs

Relational DBs are operational (transactional).

- Conceived in 1969 by IBM mathematician and scientist Dr. Edgar Codd.
- He wanted a better way to store data that would keep its integrity in check.
- Looked for mathematical approach and found it.
    - Based on set theory and first-order predicate logic.

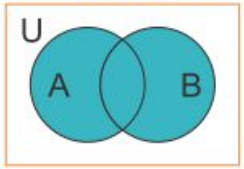
LC1711

Sets – Venn Diagrams

Venn Diagrams: Shows logical relations between a finite collection of sets.

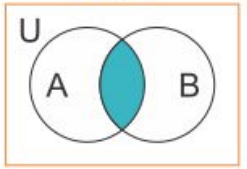
Union of Sets - Consists of all elements in sets A and B.

$A \cup B$



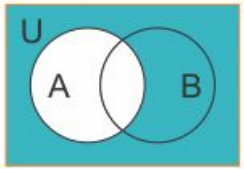
Intersection of Sets - Consists of only the common elements in sets A and B.

$A \cap B$



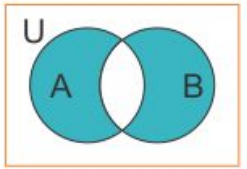
Complement of Set - Consists of elements which do not belong to set A.

$\overline{A}$, A^c or A'



Symmetric Difference of Sets - Consists of elements in sets A and B but not in their intersection.

$A \triangle B$ or $A \ominus B$



Relational Database Systems

- Relational Database Management Systems (RDBMS) are software programs to work with relational databases.
  - Microsoft SQL Server
  - PostgreSQL
  - MySQL
  - IBM DB2
  - Oracle
  - . . .

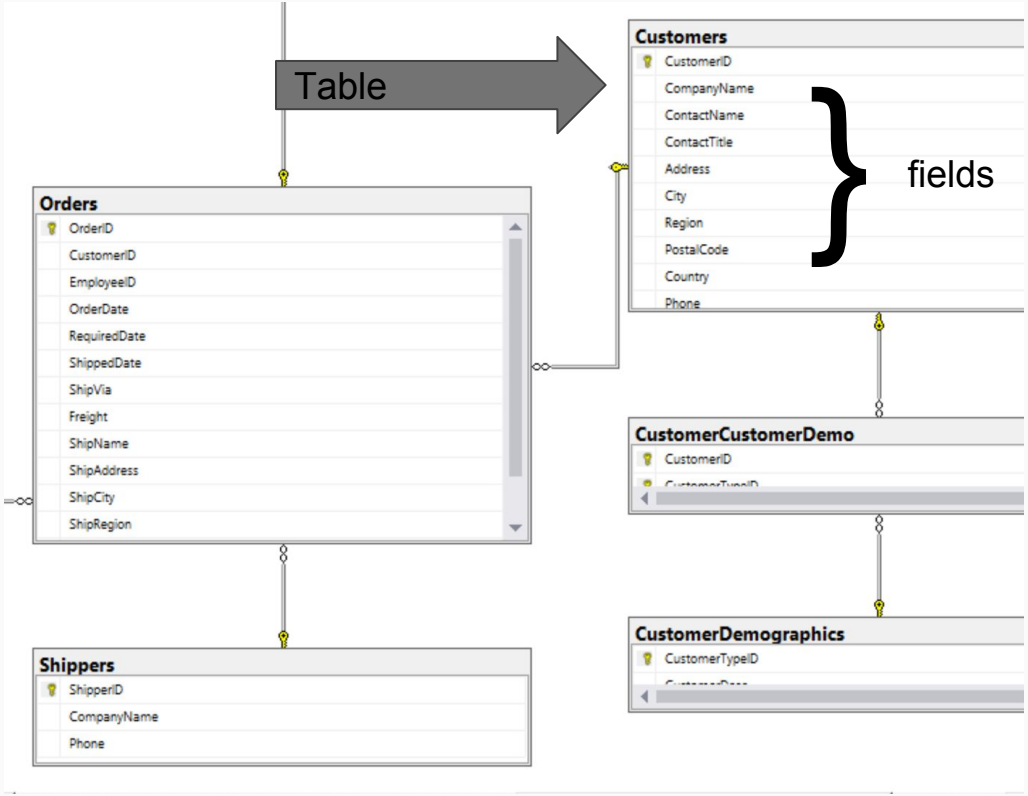
Anatomy of Relational DBs

Anatomy of Relational DBs Overview

- Tables
- Fields
- Records
- Keys
- Views
- Relationships

Tables

- Main structures in db
- Has at least 1 field
  - Primary Key
    - Unique Identifier
- Represents *single* specific subject
  - Object
    - E.g. “Customers”
  - Event
    - E.g. “Customer Orders”



Fields

- Smallest structure in a database.
- Represents a characteristic of the table's subject.
  - Like a "property" in OO.
- Contains ONE and only ONE value
  - Color : Brown 
    NOT
    ~~Color: Brown, Red, Green~~

Field Field Field Field

CustomerID	CompanyName	ContactName	ContactTitle
These are rows (a.k.a. records)			
AROUT	Around the Horn	Thomas Hardy	Sales Representative
BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative
BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager
BOLID	Bólido Comidas preparadas	Martín Sommer	Owner
BONAP	Bon app'	Laurence Lebihan	Owner
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager

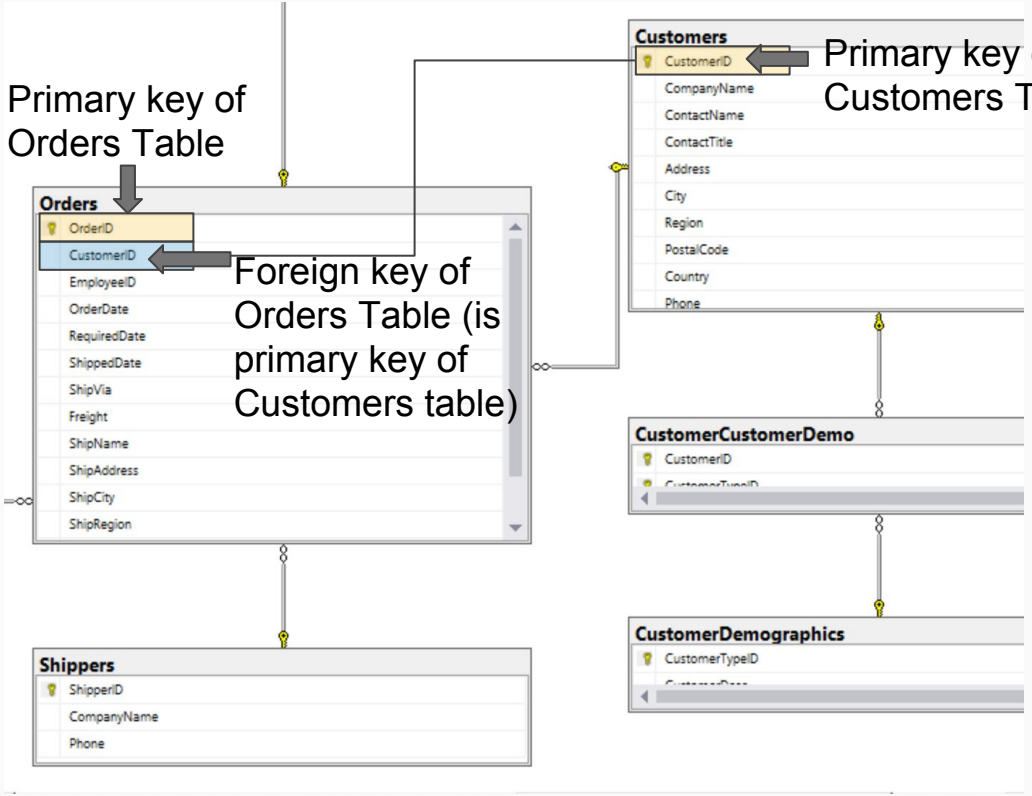
Records

- Represents a unique instance of the subject of a table
    - Think of these as an object instance, whereas the table would be like a class.
- Composed of the entire set of fields in a table (even if some fields in that row have null or empty values).
- On the right, each record represents a single customer.

CustomerID	CompanyName	ContactName	ContactTitle
AROUT	Around the Horn	Thomas Hardy	Sales Representative
BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative
BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager
BOLID	Bólido Comidas preparadas	Martín Sommer	Owner
BONAP	Bon app'	Laurence Lebihan	Owner
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager

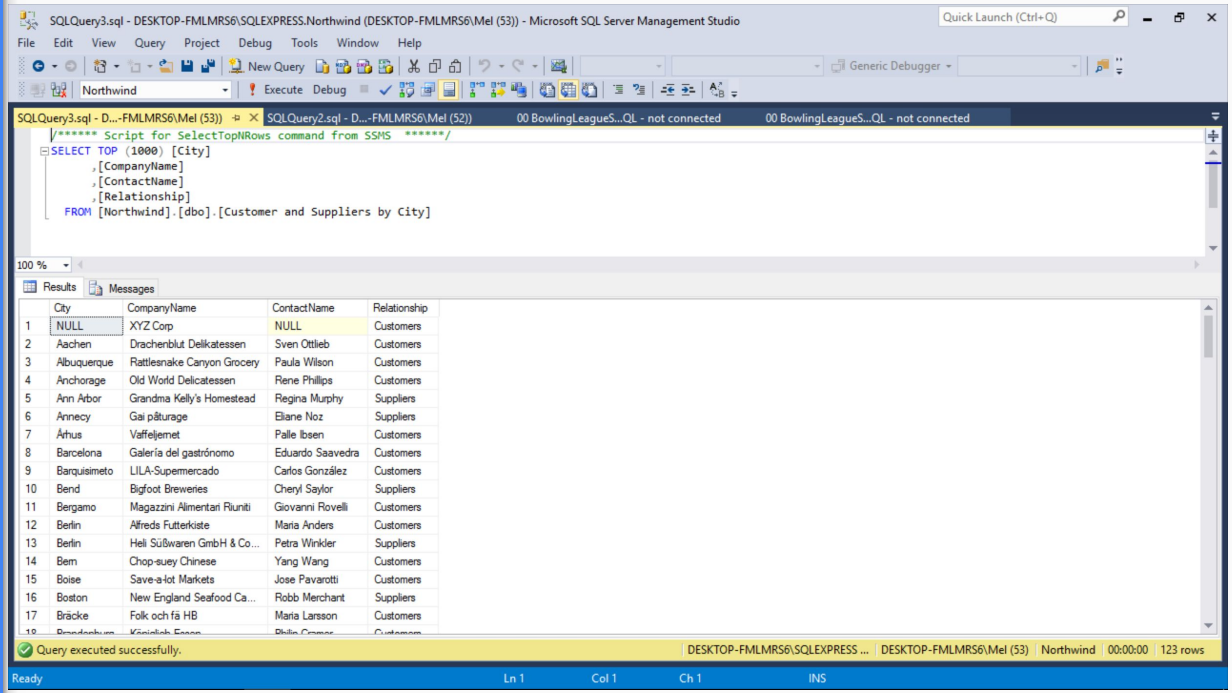
Keys

- Special fields within a table.
- Different types of Keys, but 2 most used are:
  - Primary keys
    - Unique identifier for a record in the table.
    - Like US citizens have Social Security number.
  - Foreign keys
    - Helps establish relationships between tables.
    - Ensures integrity.
    - It's a primary key from one table that resides in another.



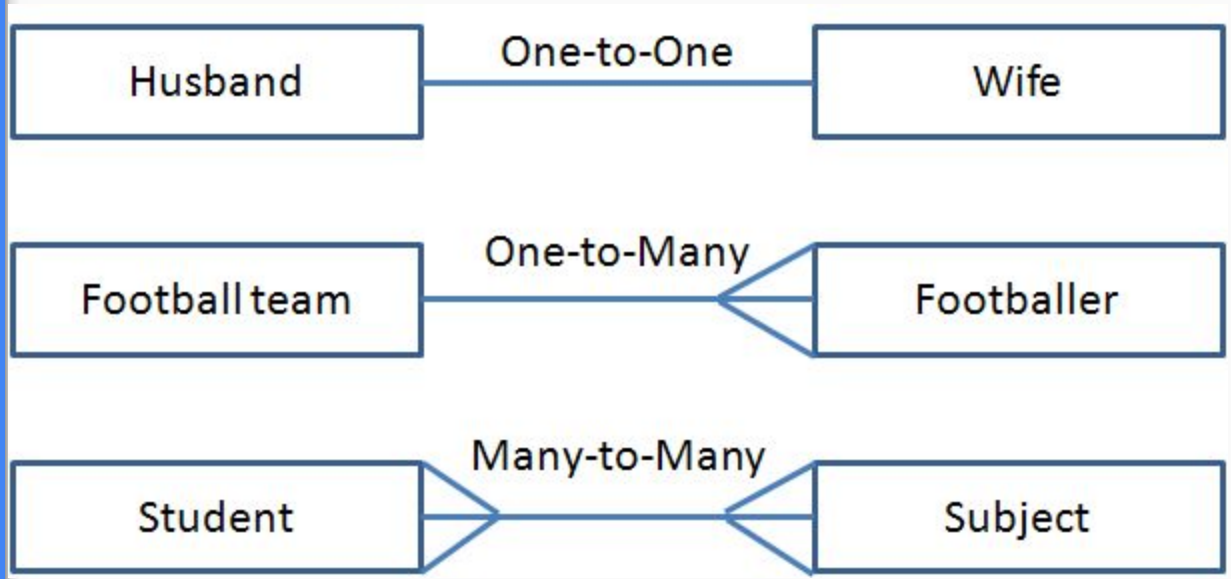
Views

- A virtual table made up of fields from other tables (one or more)
- Let's you see info from your database in other ways (other views).



Relationships

- How tables are linked or joined together.
- 3 Types:
  - One-to-One
  - One-to-Many
  - Many-to-Many



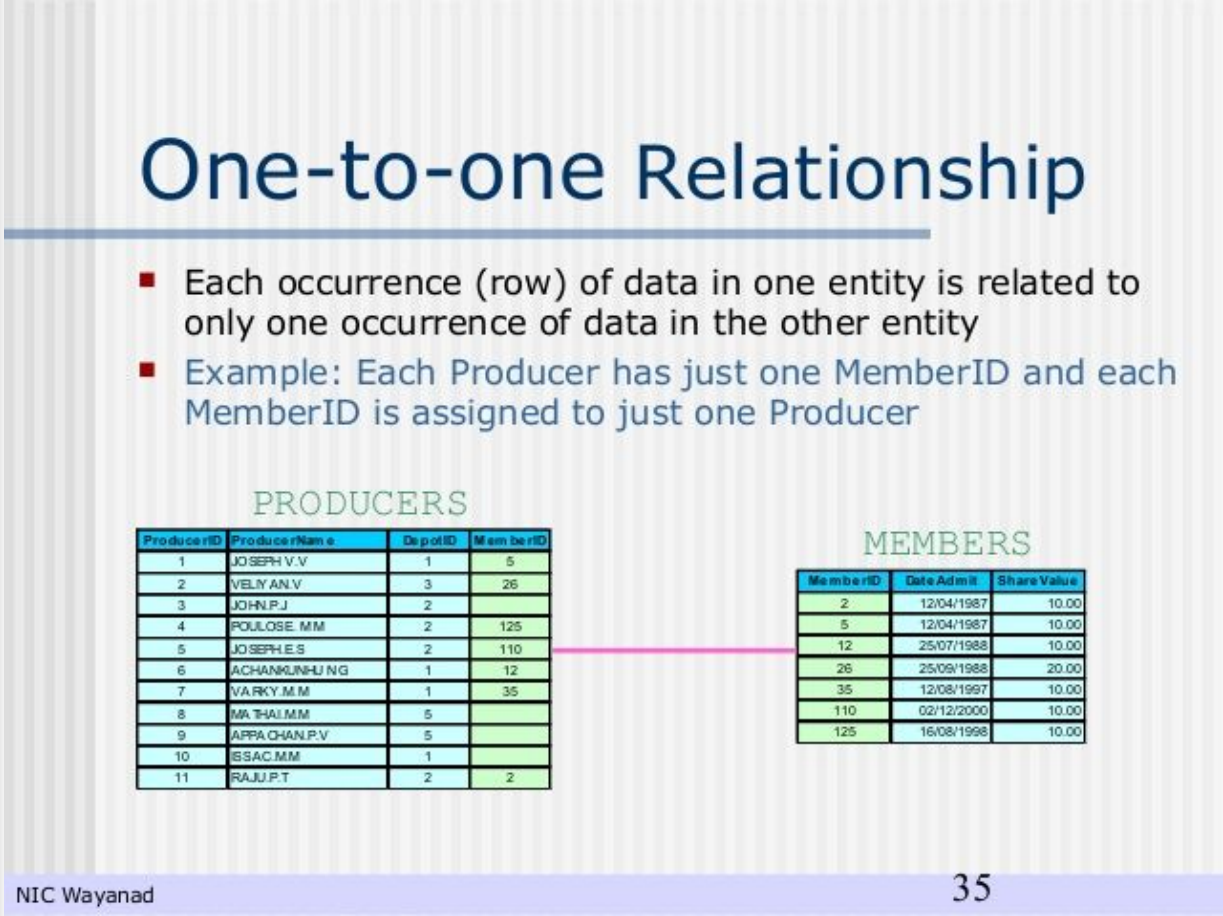
Types of Relationships

Types of Relationships Overview

- One-to-one
- One-to-many
- Many-to-many

One-to-One

- One record from the first table is related to one and only one record of the second table.
- Rare -- because these could really be in a single table (and often are)
- Use cases :
  - When security is important for certain fields.
  - When you're pulling data in from another source.



One-to-Many

- A single record from the first table can be related to many records from the second table, BUT a record from the second table can only be related to one record from the first table.
- Very common scenario.
- Example: One customer can have many orders, but a single order can only have one customer.

CUSTOMERS

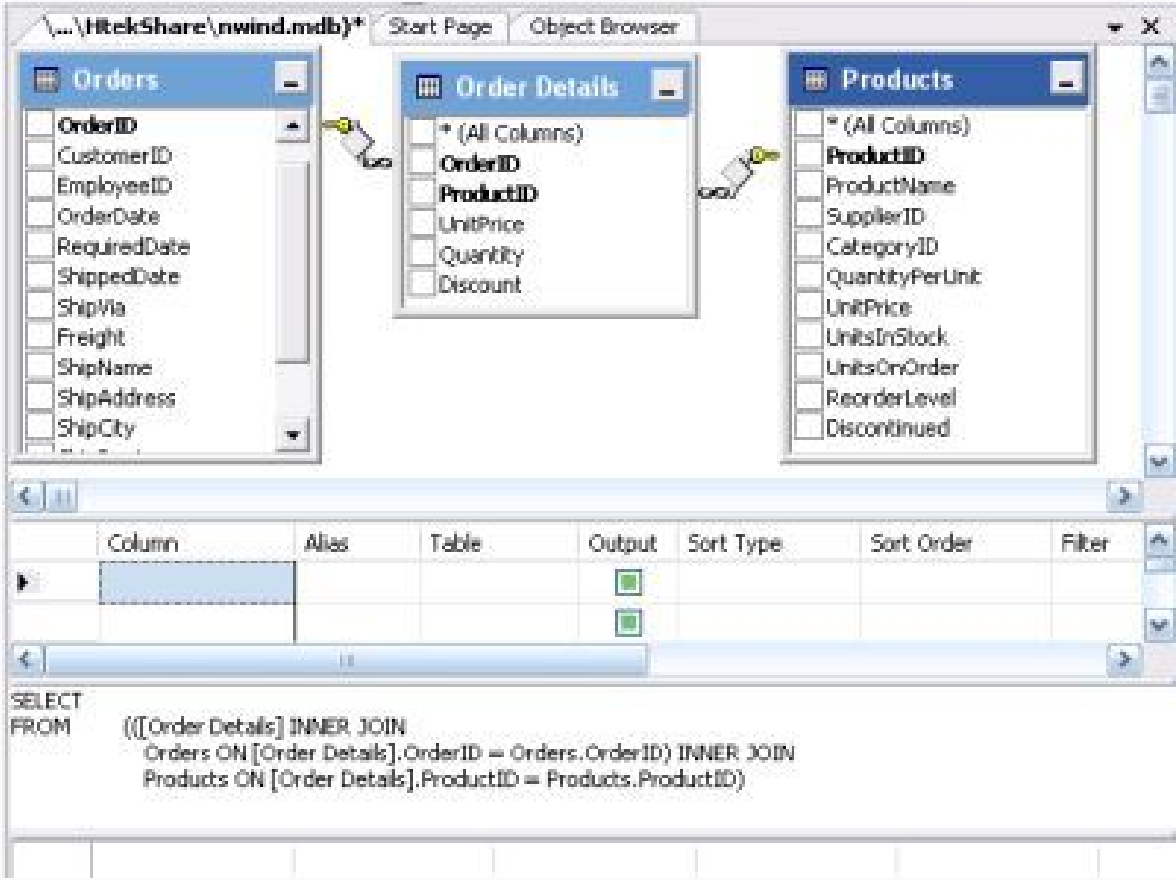
customer_id	customer_name
101	John Doe
102	Bruce Wayne

ORDERS

order_id	customer_id	order_date	amount
555	101	12/24/09	$156.78
556	102	12/25/09	$99.99
557	101	12/26/09	$75.00

Many-to-Many

- A single record from the first table can be related to many records from the second table, AND a single record from the second table can be related to many records from the first table.
- Very common.
- Example: One product can be in many orders, and many orders can have the same product.
- Note: You can't do this in 2 tables (try it yourself). You need a third joining or linking table in this scenario.



Sound DB Structure

Sound DB Structure Overview

- Importance of sound DB architecture
- Good naming
- No duplicate fields
- Primary key
- Appropriate foreign keys
- No calculated values
- No multipart fields
- No multivalued fields

Importance of sound DB structure

- Without a properly designed database, SQL queries won’t be easy, and will often be incorrect.
- This is often left to a DBA (database administrator), but it’s important to understand how to spot a poorly designed db so you know what you’re dealing with.

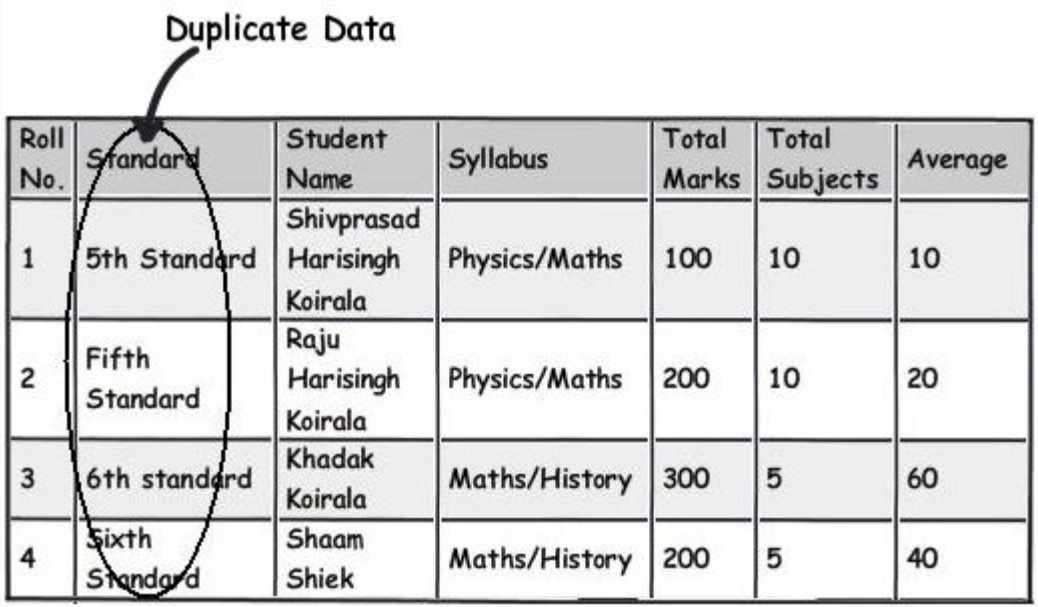
Good Naming

- Be thoughtful when naming your database, tables, fields, and views
  - Be specific
  - Think about your organization’s conventions
  - Examples
    - Good: CustomerCellPhoneNumber
    - Bad: PhoneNumber

NO Duplicate Fields

- If you see the same field throughout tables in the database, you have a problem. DON'T DO THIS.
- Hurts the integrity of data. What if you change it one place, but not the other?

Wrong

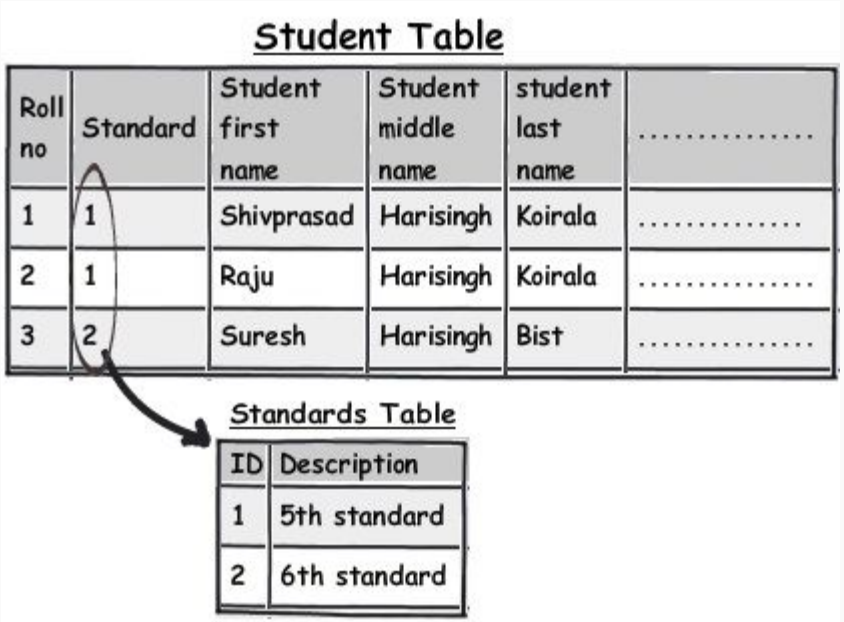


Roll No.	Standard	Student Name	Syllabus	Total Marks	Total Subjects	Average
1	5th Standard	Shivprasad Harisingh Koirala	Physics/Maths	100	10	10
2	Fifth Standard	Raju Harisingh Koirala	Physics/Maths	200	10	20
3	6th standard	Khadak Koirala	Maths/History	300	5	60
4	Sixth Standard	Shaam Shiek	Maths/History	200	5	40

NO Duplicate Fields

- Instead, use a referential table and keys.

Right



Primary Key

- Each table should have a distinct identifier.
- You use this as a handle in order to know you are grabbing the right records.
- Typically a primary key is a single field, but in cases of joining (linking) tables, you'll sometimes see 2 foreign keys become a composite primary key for that table.

Primary Key

CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne

Primary Key

ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	$156.78
556	102	12/25/09	$99.99
557	101	12/26/09	$75.00

Foreign Key

- In order to create one-to-many and many-to-many relationships between tables, you'll use foreign keys.
- This will also help maintain the integrity of the data.
- You can have multiple foreign keys in a table, especially in joining (linking) tables.
- The example on the right shows a one-to-many relationship between customers and orders.

Primary Key

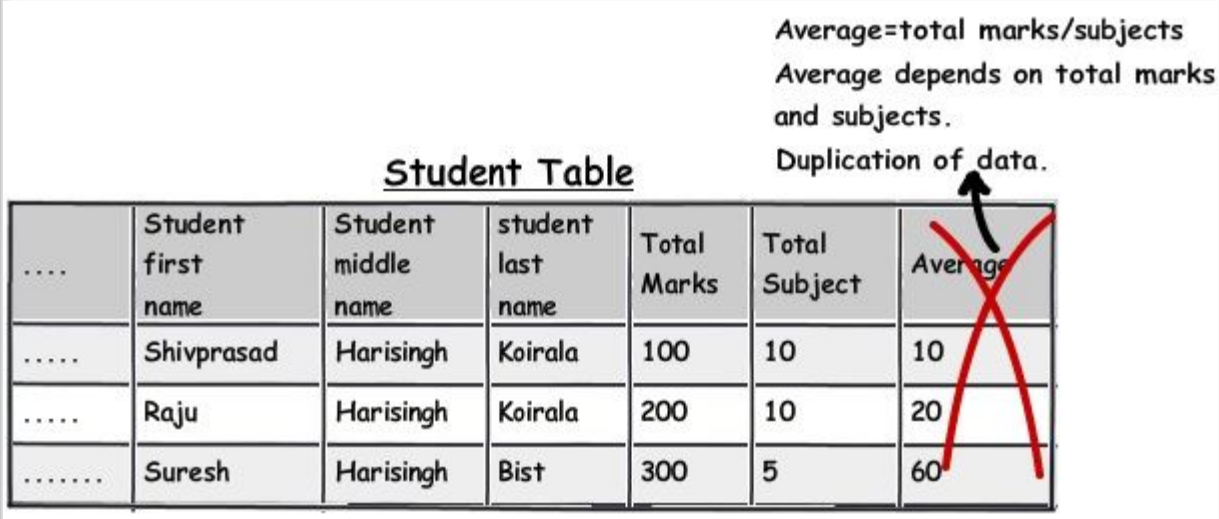
CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne

Primary Key Foreign Key

ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	$156.78
556	102	12/25/09	$99.99
557	101	12/26/09	$75.00

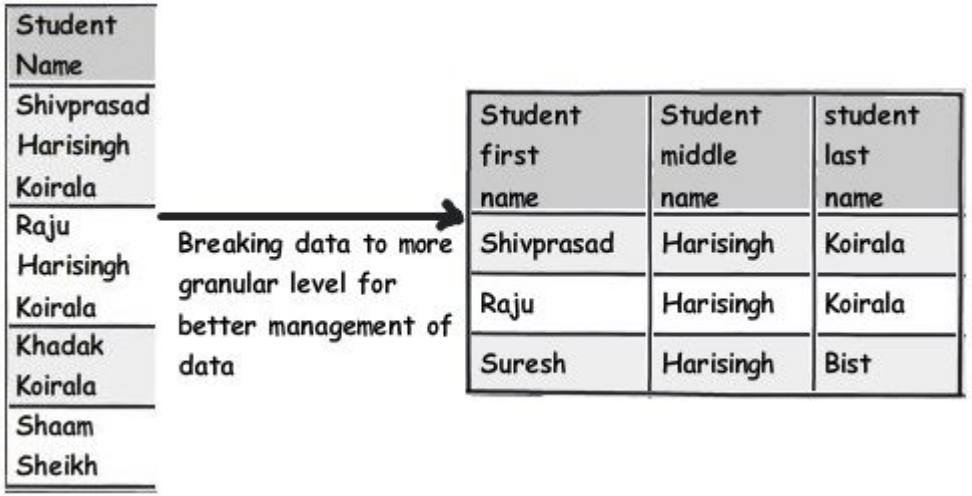
NO Calculated Fields

- No field should depend upon another, non-primary key field.
- The field called Average (right) depends on the 2 non-primary fields, Total Marks and Total Subject.
- This data duplication can lead to all sorts of issues. Think about if Total Subject is updated, for example. The Average would change. If it isn't updated, the integrity of the data is in jeopardy.



NO Multipart Fields

- No field should depend upon another, non-primary key field.
- The field called Average (right) depends on the 2 non-primary fields, Total Marks and Total Subject.
- This data duplication can lead to all sorts of issues. Think about if Total Subject is updated, for example. The Average would change. If it isn't updated, the integrity of the data is in jeopardy.
- Makes searching and sorting easier.



NO Multivalue Fields

- Avoid repeating groups.
- Imagine if someone insert incorrectly. Imagine trying to search for all Maths. It becomes very difficult.
- Data integrity threatened.
- Instead, break this data out and create table relationships (shown on next slide).

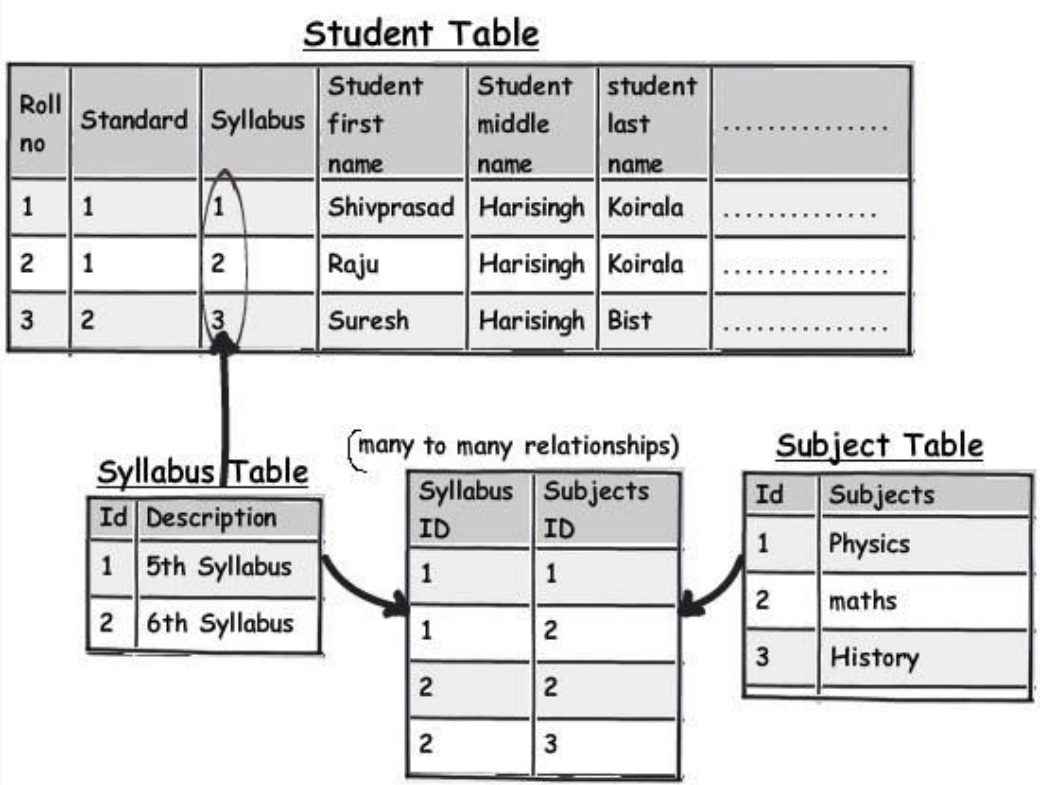
Wrong

Roll No.	Standard	Student Name	Syllabus	Total Marks	Total Subjects	Average
1	5th Standard	Shivprasad Harisingh Koirala	Physics/Maths	100	10	10
2	Fifth Standard	Raju Harisingh Koirala	Physics/Maths	200	10	20
3	6th standard	Khadak Koirala	Maths/History	300	5	60
4	Sixth Standard	Shaam Shiek	Maths/History	200	5	40

NO Multivalue Fields

- Ensure you break down data appropriately and don't stuff a bunch of data into a single field.

Right



Homework

Ensure a SQL Server is installed.

MS SQL Server with SQL Server Management Studio

OR

MySQL

Ensure you have run the download scripts to create and populate your databases. Script located in zip file on the Github repository.

https://github.com/MelMcGee/sql-in-7/blob/master/SQLQFMM3.zip

View the Readme file in the zipped folder for instructions.