

Introduction to Data Access and Storage

Thomas Rosenthal - DSI @ UofT

Module 01

Introduction:

Welcome

What is SQL?

Data Modelling

Getting Started:

Welcome

What is SQL?

Data Modelling

Welcome

About Me

Course Content

Quick Technical Check

Welcome

About Me

Course Content

Quick Technical Check

About Me

- Fell into SQL in my first job out of undergraduate
- Worked as a Data Analyst in US healthcare until moving to Canada in 2018
 - Challenging data
 - Easy to see impact of good queries
 - I was pretty naive about data, probably lucky I didn't break anything 🤪
- Worked as a Data Engineer at Plan Canada managing a CRM Data Warehouse backend
- Went back to UofT to do my Master of Information
 - Wanted to do less SQL, more R and python
 - SQL seemed like it was on its way out...NOPE, I was just wrong
- Reluctant Data Scientist (*do we even need this model?*)
 - Currently doing Data Governance and Ethics implementation at Thomson Reuters
- Have a 2 year old Samoyed named Alto who takes up a lot of my free time 👉



About Me

- Recently graduated from UofT with majors in Computer Science and Statistics, Minor in Mathematics
 - Held co-op positions at different banks
 - Citi (Software Development)
 - Bank of Montreal(Business Analytics)
- Now working full-time at RBC Capital Markets in their Quant and Tech Services division
 - Currently a Business Systems Analyst (position is a cross between SDE and BA)
 - Working with salesforce CRM backend data in SQL server
 - Moving to a more technical role in a new team (Supervisory and Regulatory controls) next week (for my next rotation)
- Outside of work, strongly believe in advocating for the representation of women in tech
 - Was the president of Women in Computer Science at UofT for 2 years
 - Now getting involved with Toronto Wxmen in Data Science and RWomen (RBC's women in tech ERG)



About Me

- Undergrad from Paris-Saclay University in Applied Mathematics and Data Science -Discovered SQL, Python and R in an academic setting, and played with them in the industry after.
- A few internships in Machine Learning and Data Science:
 - Docugami (Incoming Research Scientist)
 - Ekimetrics (Data Scientist)
 - Stellantis (ML Engineer)
 - Randstad (ML Enginner)
- Currently pursuing the MscAC program at UofT - in the AI concentration
- Working on a lot of cool stuff at the intersection between Data Science and LLMs - very exciting but also a very fast-paced environment!
- Considering pursuing a PhD after to become a full-time professor 😎
- On my free time : play guitar, saxophone & bass and have been part of a few music bands 🎶



Welcome

About Me

Course Content

Quick Technical Check

Course Content

- Data Modelling, Data Structures, Schemas, Basic Data Management, Normal Forms
- Basic SQL Syntax
- Essential SQL Syntax
- Advanced Techniques
- Importing and Exporting Data to and from SQL
- SQL's relationship to the Machine Learning Pipeline
- Data Stewardship, Ethics, and SQL in the Wild

Github Repo

https://github.com/UofT-DSI/02-intro_sql

- Schedule
- These slides (HTML & PDF)
- Our database for live coding
- All in-class code
- Homework (and answers, posted afterwards)
- Assignment details and rubrics
- Policies, due dates, etc

It is crucial you visit the repo throughout the course, as I may make minor changes and push new content.

Course Content

- This course is an *Introduction* to SQL
- At the end of the course, I hope you will:
 - Feel comfortable with SQL
 - Know how to search for the right thing on Stack Overflow
 - Read documentation
- We won't cover advanced topics like:
 - Stored Procedures, Triggers, Jobs
 - DBA work (monitoring, server setup, etc)
 - Complex ETL or tooling



The screenshot shows the SQLite Database Browser interface. The main window displays a list of employees from a database. The columns shown are Employee ID, First Name, Last Name, and Hire Date. The data includes rows for employees like '48571 Alana Mccormick' and '48572 Jennifer Wicks'. The browser's toolbar at the top includes icons for file operations, zoom, and search. The title bar reads 'Software: DB Browser for SQLite'.

Employee ID	First Name	Last Name	Hire Date
48571	Alana	Mccormick	1997-06-11
48572	Jennifer	Wicks	1997-06-11
48573	Patricia	Yost	1997-06-11
48574	John	Whaley	1997-06-11
48575	Michael	Whaley	1997-06-11
48576	Patricia	Yost	1997-06-11
48577	John	Whaley	1997-06-11
48578	Michael	Whaley	1997-06-11
48579	Patricia	Yost	1997-06-11
48580	John	Whaley	1997-06-11
48581	Michael	Whaley	1997-06-11
48582	Patricia	Yost	1997-06-11
48583	John	Whaley	1997-06-11
48584	Michael	Whaley	1997-06-11
48585	Patricia	Yost	1997-06-11
48586	John	Whaley	1997-06-11
48587	Michael	Whaley	1997-06-11
48588	Patricia	Yost	1997-06-11
48589	John	Whaley	1997-06-11
48590	Michael	Whaley	1997-06-11
48591	Patricia	Yost	1997-06-11
48592	John	Whaley	1997-06-11
48593	Michael	Whaley	1997-06-11
48594	Patricia	Yost	1997-06-11
48595	John	Whaley	1997-06-11
48596	Michael	Whaley	1997-06-11
48597	Patricia	Yost	1997-06-11
48598	John	Whaley	1997-06-11
48599	Michael	Whaley	1997-06-11
48600	Patricia	Yost	1997-06-11
48601	John	Whaley	1997-06-11
48602	Michael	Whaley	1997-06-11
48603	Patricia	Yost	1997-06-11
48604	John	Whaley	1997-06-11
48605	Michael	Whaley	1997-06-11
48606	Patricia	Yost	1997-06-11
48607	John	Whaley	1997-06-11
48608	Michael	Whaley	1997-06-11
48609	Patricia	Yost	1997-06-11
48610	John	Whaley	1997-06-11
48611	Michael	Whaley	1997-06-11
48612	Patricia	Yost	1997-06-11
48613	John	Whaley	1997-06-11
48614	Michael	Whaley	1997-06-11
48615	Patricia	Yost	1997-06-11
48616	John	Whaley	1997-06-11
48617	Michael	Whaley	1997-06-11
48618	Patricia	Yost	1997-06-11
48619	John	Whaley	1997-06-11
48620	Michael	Whaley	1997-06-11
48621	Patricia	Yost	1997-06-11
48622	John	Whaley	1997-06-11
48623	Michael	Whaley	1997-06-11
48624	Patricia	Yost	1997-06-11
48625	John	Whaley	1997-06-11
48626	Michael	Whaley	1997-06-11
48627	Patricia	Yost	1997-06-11
48628	John	Whaley	1997-06-11
48629	Michael	Whaley	1997-06-11
48630	Patricia	Yost	1997-06-11
48631	John	Whaley	1997-06-11
48632	Michael	Whaley	1997-06-11
48633	Patricia	Yost	1997-06-11
48634	John	Whaley	1997-06-11
48635	Michael	Whaley	1997-06-11
48636	Patricia	Yost	1997-06-11
48637	John	Whaley	1997-06-11
48638	Michael	Whaley	1997-06-11
48639	Patricia	Yost	1997-06-11
48640	John	Whaley	1997-06-11
48641	Michael	Whaley	1997-06-11
48642	Patricia	Yost	1997-06-11
48643	John	Whaley	1997-06-11
48644	Michael	Whaley	1997-06-11
48645	Patricia	Yost	1997-06-11
48646	John	Whaley	1997-06-11
48647	Michael	Whaley	1997-06-11
48648	Patricia	Yost	1997-06-11
48649	John	Whaley	1997-06-11
48650	Michael	Whaley	1997-06-11
48651	Patricia	Yost	1997-06-11
48652	John	Whaley	1997-06-11
48653	Michael	Whaley	1997-06-11
48654	Patricia	Yost	1997-06-11
48655	John	Whaley	1997-06-11
48656	Michael	Whaley	1997-06-11
48657	Patricia	Yost	1997-06-11
48658	John	Whaley	1997-06-11
48659	Michael	Whaley	1997-06-11
48660	Patricia	Yost	1997-06-11
48661	John	Whaley	1997-06-11
48662	Michael	Whaley	1997-06-11
48663	Patricia	Yost	1997-06-11
48664	John	Whaley	1997-06-11
48665	Michael	Whaley	1997-06-11
48666	Patricia	Yost	1997-06-11
48667	John	Whaley	1997-06-11
48668	Michael	Whaley	1997-06-11
48669	Patricia	Yost	1997-06-11
48670	John	Whaley	1997-06-11
48671	Michael	Whaley	1997-06-11
48672	Patricia	Yost	1997-06-11
48673	John	Whaley	1997-06-11
48674	Michael	Whaley	1997-06-11
48675	Patricia	Yost	1997-06-11
48676	John	Whaley	1997-06-11
48677	Michael	Whaley	1997-06-11
48678	Patricia	Yost	1997-06-11
48679	John	Whaley	1997-06-11
48680	Michael	Whaley	1997-06-11
48681	Patricia	Yost	1997-06-11
48682	John	Whaley	1997-06-11
48683	Michael	Whaley	1997-06-11
48684	Patricia	Yost	1997-06-11
48685	John	Whaley	1997-06-11
48686	Michael	Whaley	1997-06-11
48687	Patricia	Yost	1997-06-11
48688	John	Whaley	1997-06-11
48689	Michael	Whaley	1997-06-11
48690	Patricia	Yost	1997-06-11
48691	John	Whaley	1997-06-11
48692	Michael	Whaley	1997-06-11
48693	Patricia	Yost	1997-06-11
48694	John	Whaley	1997-06-11
48695	Michael	Whaley	1997-06-11
48696	Patricia	Yost	1997-06-11
48697	John	Whaley	1997-06-11
48698	Michael	Whaley	1997-06-11
48699	Patricia	Yost	1997-06-11
48700	John	Whaley	1997-06-11
48701	Michael	Whaley	1997-06-11
48702	Patricia	Yost	1997-06-11
48703	John	Whaley	1997-06-11
48704	Michael	Whaley	1997-06-11
48705	Patricia	Yost	1997-06-11
48706	John	Whaley	1997-06-11
48707	Michael	Whaley	1997-06-11
48708	Patricia	Yost	1997-06-11
48709	John	Whaley	1997-06-11
48710	Michael	Whaley	1997-06-11
48711	Patricia	Yost	1997-06-11
48712	John	Whaley	1997-06-11
48713	Michael	Whaley	1997-06-11
48714	Patricia	Yost	1997-06-11
48715	John	Whaley	1997-06-11
48716	Michael	Whaley	1997-06-11
48717	Patricia	Yost	1997-06-11
48718	John	Whaley	1997-06-11
48719	Michael	Whaley	1997-06-11
48720	Patricia	Yost	1997-06-11
48721	John	Whaley	1997-06-11
48722	Michael	Whaley	1997-06-11
48723	Patricia	Yost	1997-06-11
48724	John	Whaley	1997-06-11
48725	Michael	Whaley	1997-06-11
48726	Patricia	Yost	1997-06-11
48727	John	Whaley	1997-06-11
48728	Michael	Whaley	1997-06-11
48729	Patricia	Yost	1997-06-11
48730	John	Whaley	1997-06-11
48731	Michael	Whaley	1997-06-11
48732	Patricia	Yost	1997-06-11
48733	John	Whaley	1997-06-11
48734	Michael	Whaley	1997-06-11
48735	Patricia	Yost	1997-06-11
48736	John	Whaley	1997-06-11
48737	Michael	Whaley	1997-06-11
48738	Patricia	Yost	1997-06-11
48739	John	Whaley	1997-06-11
48740	Michael	Whaley	1997-06-11
48741	Patricia	Yost	1997-06-11
48742	John	Whaley	1997-06-11
48743	Michael	Whaley	1997-06-11
48744	Patricia	Yost	1997-06-11
48745	John	Whaley	1997-06-11
48746	Michael	Whaley	1997-06-11
48747	Patricia	Yost	1997-06-11
48748	John	Whaley	1997-06-11
48749	Michael	Whaley	1997-06-11
48750	Patricia	Yost	1997-06-11
48751	John	Whaley	1997-06-11
48752	Michael	Whaley	1997-06-11
48753	Patricia	Yost	1997-06-11
48754	John	Whaley	1997-06-11
48755	Michael	Whaley	1997-06-11
48756	Patricia	Yost	1997-06-11
48757	John	Whaley	1997-06-11
48758	Michael	Whaley	1997-06-11
48759	Patricia	Yost	1997-06-11
48760	John	Whaley	1997-06-11
48761	Michael	Whaley	1997-06-11
48762	Patricia	Yost	1997-06-11
48763	John	Whaley	1997-06-11
48764	Michael	Whaley	1997-06-11
48765	Patricia	Yost	1997-06-11
48766	John	Whaley	1997-06-11
48767	Michael	Whaley	1997-06-11
48768	Patricia	Yost	1997-06-11
48769	John	Whaley	1997-06-11
48770	Michael	Whaley	1997-06-11
48771	Patricia	Yost	1997-06-11
48772	John	Whaley	1997-06-11
48773	Michael	Whaley	1997-06-11
48774	Patricia	Yost	1997-06-11
48775	John	Whaley	1997-06-11
48776	Michael	Whaley	1997-06-11
48777	Patricia	Yost	1997-06-11
48778	John	Whaley	1997-06-11
48779	Michael	Whaley	1997-06-11
48780	Patricia	Yost	1997-06-11
48781	John	Whaley	1997-06-11
48782	Michael	Whaley	1997-06-11
48783	Patricia	Yost	1997-06-11
48784	John	Whaley	1997-06-11
48785	Michael	Whaley	1997-06-11
48786	Patricia	Yost	1997-06-11
48787	John	Whaley	1997-06-11
48788	Michael	Whaley	1997-06-11
48789	Patricia	Yost	1997-06-11
48790	John	Whaley	1997-06-11
48791	Michael	Whaley	1997-06-11
48792	Patricia	Yost	1997-06-11
48793	John	Whaley	1997-06-11
48794	Michael	Whaley	1997-06-11
48795	Patricia	Yost	1997-06-11
48796	John	Whaley	1997-06-11
48797	Michael	Whaley	1997-06-11
48798	Patricia	Yost	1997-06-11
48799	John	Whaley	1997-06-11
48800	Michael	Whaley	1997-06-11
48801	Patricia	Yost	1997-06-11
48802	John	Whaley	1997-06-11
48803	Michael	Whaley	1997-06-11
48804	Patricia	Yost	1997-06-11
48805	John	Whaley	1997-06-11
48806	Michael	Whaley	1997-06-11
48807	Patricia	Yost	1997-06-11
48808	John	Whaley	1997-06-11
48809	Michael	Whaley	1997-06-11
48810	Patricia	Yost	1997-06-11
48811	John	Whaley	1997-06-11
48812	Michael	Whaley	1997-06-11
48813	Patricia	Yost	1997-06-11
48814	John	Whaley	1997-06-11
48815	Michael	Whaley	1997-06-11
48816	Patricia	Yost	1997-06-11
48817	John	Whaley	1997-06-11
48818	Michael	Whaley	1997-06-11
48819	Patricia	Yost	1997-06-11
48820	John	Whaley	1997-06-11
48821	Michael	Whaley	1997-06-11
48822	Patricia	Yost	1997-06-11
48823	John	Whaley	1997-06-11
48824	Michael	Whaley	1997-06-11
48825	Patricia	Yost	1997-06-11
48826	John	Whaley	1997-06-11
48827	Michael	Whaley	1997-06-11
48828	Patricia	Yost	1997-06-11
48829	John	Whaley	1997-06-11
48830	Michael	Whaley	1997-06-11
48831	Patricia	Yost	1997-06-11
48832	John	Whaley	1997-06-11
48833	Michael		

Course Content

Homework (required)

- Six homeworks total
- At the end of each SQL module, I will provide you with 3-10 queries to write on your own
 - Covers the topics of the modules
- Review answers in Tutorials with Ananya and Vishnou
 - I may go over one or two queries at the start of class if desired
- Designed to be relatively easy
 - Reaffirms what we wrote together
 - Doing work on your own helps reinforce the learning
- ChatGPT probably won't help you much

Course Content

Grading

- Pass/Fail. Do the work, pass the course :)
- One Assignment: Data Model Design, 40%, due March 23
- Homework: 48% (8% each), due on Thursdays and Saturdays
- Class Attendance: 12% (2% each)
 - Let Ananya or myself know if you are unable to attend a lesson
 - Code along!! Best way to learn.

Welcome

About Me

Course Content

Quick Technical Check

Quick Technical Check

(a slide where we make sure everyone has DB Browser for SQLite installed)

If not, please download it here: <https://sqlitebrowser.org/dl/>

For live coding:

- Please download/fork the FarmersMarket.db from our GH repo:
 - https://github.com/UofT-DSI/02-intro_sql/tree/main/SQL
- Open it in SQLite with the "Open Database" button and navigate to wherever you have saved it

Good to go?

If not, please message Ananya

Getting Started:

Welcome

What is SQL?

Data Modelling

What is SQL?

SQL

Flavours

Environments for SQL

What is SQL?

SQL

Flavours

Environments for SQL

SQL

SQL Fundamentals

- SQL: Structured Query Language
 - Pronounced as either S.Q.L. (ess-cue-ell) or “sequel”
- SQL is a *query* language rather than a programming language
 - Querying is closer to telling a computer *what you want*, rather than *what is has to do*
 - SQL code is often less reproducible than other programming languages because it's domain specific
 - Some SQL code, especially more advanced procedural code, is reproducible within the same flavour
 - SQL's domain is databases and is based on set theory
- Designed to manage data within Relational Database Management Systems (RDBMs), e.g.
 - MSSQL
 - Oracle DB
 - MySQL/MariaDB
 - PostgreSQL

SQL

SQL Formatting

- Like other programming/query languages, SQL has reserved keywords/commands to perform instructional operations
 - Generally, these keywords are written in all caps: `SELECT`
 - Most modern interpreters no longer require this, but it is the expected standard
- All statements/queries should end with a semicolon
 - A few SQL constructs (like common table expressions, we'll get to these later) require them, otherwise they are optional
 - I'll almost certainly forget to use them
 - There's some debate over whether or not it's best practice

SQL

SQL Formatting

- In SQL, white space and/or line breaks do not matter
 - Readability is important
 - Try to keep SQL statements to a reasonable screen width
 - Use sensible line breaks
 - Offset subqueries with indents
- Code is commented in/out with `--` rather than `#`
- Code blocks can be commented out with `/* */`

```
/*
somecode spanning
multiple lines
*/
```

What is SQL?

SQL

Flavours

Environments for SQL

Flavours

- RDBMs differ from one to the next:
 - different keywords
 - e.g. return only 10 rows: `SELECT TOP 10...` vs `SELECT ... LIMIT 10`
 - different syntax
 - e.g. not equal: `!=` or `<>` (or both)
 - other, more nuanced/complex differences
 - e.g. architecture, data types, etc
- We are using SQLite:
 - Super easy to get setup
 - Requires almost no overhead
 - Open source, *free*
 - Similar enough in syntax to learn on
 - Used all over the world and in many applications
 - e.g. Firefox uses a SQLite backend to write a user's history locally

Flavours

- Broad observations about Open Source systems:
 - Excellent at what they are designed for
 - Varying data types (SQLite has some unique ones!)
 - Not every command exists, but workarounds are usually possible
 - Some utilize RDBMs that feel extremely outdated
- Broad observations about enterprise systems:
 - Powerful and designed to handle edge cases
 - Feel a bit more refined
 - Can be version dependent
 - Tend to "lock in" businesses/organizations
 - Migration is costly, sometimes outrageously so
 - Newer players (Snowflake, Databricks, etc) and cloud providers (Azure, AWS, GCP, etc) offer a lot more functionality than just database querying
 - Sometimes use different terminology to describe SQL tasks

What is SQL?

SQL

Flavours

Environments for SQL

Environments for SQL

Databases

- Relational databases are a collection of tables, views, procedural code, and other SQL-assisting artefacts
 - Generally the data stored in a database will be related to a real-world concept
 - Backends to data-collecting systems are often databases
 - e.g. CRMs, EMR software, ERPs, web-based applications
 - Usually not connected to other databases unless deemed necessary
 - Often transactional, meaning data is actively being written to by frontend systems
 - Tables are normalized
- There are also non-relational databases, often referred to as NoSQL
 - We won't cover these
 - Common tools include: Amazon DynamoDB, Azure CosmosDB, MongoDB, Google Cloud Datastore

Environments for SQL

Data Warehouses and Data Marts

- Data Warehouses are highly structured collections of (usually tabular) data
 - Data has been processed for a specific purpose, e.g. analytics
 - Data has been centralized
 - Often with the assistance of ETL (Extract, Transform, Load) tools
 - Have a lot of overhead, require governance, and strict rigidity
 - Tables are denormalized
 - Very common for enterprises, but losing traction in many industries
- Data Marts are created from Data Warehouses to focus on a single subject
 - Designed to make Data Warehouses easier to use
 - Data is structured, but flexibility is driven by the purpose of the Data Mart
 - Some denormalized tables may be normalized or undergo even greater normalization
 - *The subject/purpose of the Data Mart might drive these types of decisions*
 - Common for enterprises that have Data Warehouses

Environments for SQL

Data Lakes and Data Swamps

- Data Lakes allow on-demand access to raw, semi-structured, structured, and unstructured data
 - Not defined by a specific purpose
 - Highly scalable
 - Can be transactional, if systems are designed to produce outputs into Data Lakes
 - Often data sources for machine learning pipelines live in Data Lakes
 - Inexpensive compared to Data Warehouses
 - Increasingly common for enterprises to shift towards Data Lakes, especially with support from newer tools like Snowflake, Databricks, etc, which can maximize the analytical value of a Data Lake
- Data Swamps...are poorly governed Data Lakes
 - Lack of documentation, lack of governance, poorly designed Data Lakes become Swamps
 - Avoid building these

Getting Started:

Welcome

What is SQL?

Data Modelling

Data Modelling

Relational Database Management Systems

Data Models

Structure of Data

Constraints

(...)

Data Modelling (continued)

Entity Relationship Diagrams

Attributes of an ERD: Entities & Relationships

Relationship Examples

Conceptual, Logical, Physical Models

Assignment 1: Design a Logical Model

Data Modelling

Relational Database Management Systems

Data Models

Structure of Data

Constraints

(...)

Data Modelling

Relational Database Management Systems

- Relational Database Management Systems (RDBMs) are software designed to:
 - Store large amounts of data
 - Utilize a query language to allow easy retrieval of the data
 - Allow multiple users to access the data simultaneously
 - Manage permissions for data access
 - Mitigate data corruption and unauthorized access
- Generally, data is stored in a *database*
 - A database is a collection of information
 - Within a database, a collection of objects (e.g. tabular data "tables") is stored
- RDBMs allow users to define interactions between these objects, such as:
 - Establish the relationship between objects
 - Define procedural scripts to query specific data or trigger an action
 - Schedule routine work (e.g. procedures to run, maintenance, etc)

Data Modelling

Relational Database Management Systems

Data Models

Structure of Data

Constraints

(...)

Data Modelling

Data Models

- A data model is a notation for describing data or information
- Data models consist of:
 - Structure of the data
 - Operations
 - Constraints on the data
 - Relationships

Data Modelling

Relational Database Management Systems

Data Models

Structure of Data

Constraints

(...)

Data Modelling

Structure of Data

- SQL is comprised of tables

Breed	Affectionate w/ Family	Good w/ Other Dogs	Shedding	Coat Type	Coat Length	Playfulness	Energy
Pugs	5	4	4	Smooth	Short	5	3
Akitas	3	1	3	Double	Medium	3	4
Samoyeds	5	3	3	Double	Long	5	4

- Tables have Attributes and Observations
 - In SQL we call Attributes "Columns"
 - e.g. Breed, Coat Type, Coat Length
 - and Observations "Rows"
 - e.g. Samoyed, Double, Long
- SQL databases require tables to be named
 - e.g. We can call this table "breed_traits"

Data Modelling

Structure of Data

- Columns are defined (and restricted, i.e. constrained) by data types
- Common ones include:
 - **INT** (integers: 1,2,3,-1,-2,-3)
 - most systems conserve storage space specifying their range
 - **FLOAT, DECIMAL, REAL** (decimal: 5.5, 3.333333)
 - **VARCHAR, NVARCHAR, TEXT** (text strings, with a maximum length associated: 'abc')
 - **DATE, DATETIME, TIME** (dates and times: '2023-01-09', '11:11:11.000')
- These may vary slightly by flavour (in SQLite they are simpler and less restricted)
- Data types are important:
 - They affect operation speed, storage size, data validity
 - Speed: it's computationally less expensive to compute smaller values
 - Storage: small is usually better, but the wrong size will affect systems; e.g. Psy's Gangnam Style exceeded 2,147,483,647 (32-bit signed, $2^{32}/2-1$) views, causing YouTube to expand the view counter to 64-bit (~9.2b if signed)
 - Validity: ensures columns contain the right type of data for operations, e.g. avoiding $5 + \text{'ten'} = ??$

Data Modelling

Relational Database Management Systems

Data Models

Structure of Data

Constraints

(...)

Data Modelling

Constraints

- Data Models also specify constraints
- Constraints are rules that must be followed:
 - Referential-Integrity constraints
 - Ensure that values in one table have corresponding values in another table
 - Attribute Constraints
 - Ensure that certain types of values are always consistent within columns
 - May also ensure whether values are unique, not missing, etc

Data Modelling

Constraints

- **NULL** and **NOT NULL**
 - If a value can be missing or not
- **UNIQUE**
 - All values are different
- **PRIMARY KEY (PK)**
 - Ensures each value in a column is unique within the table (e.g. an ID field)
 - One PK per table
 - Cannot be NULL
 - Ensures database integrity by restricting record deletion
- **FOREIGN KEY (FK)**
 - Creates a linkage between a column in one table and a column in another table
 - Generally, foreign keys are linked to primary keys
 - Sometimes share the same name as the linked column, but this isn't required
 - Linkage requires data types to be the same
 - As many FKS as needed per table
 - May be NULL
 - Record can be deleted

Data Modelling (continued)

Entity Relationship Diagrams

Attributes of an ERD: Entities & Relationships

Relationship Examples

Conceptual, Logical, Physical Models

Assignment 1: Design a Logical Model

Data Modelling

Entity Relationship Diagrams

- Entity Relationship Diagrams (ERDs) are diagrams depicting the structure of tables within a database
 - This both *identifies the tables* and *describes their relationships*
- ERDs are useful for:
 - Database design
 - Debugging
 - Writing logical, consistent, and efficient queries
- There are three levels of detail for ERD depictions:
 - Conceptual model
 - Logical model
 - Physical model

Data Modelling (continued)

Entity Relationship Diagrams

Attributes of an ERD: Entities & Relationships

Relationship Examples

Conceptual, Logical, Physical Models

Assignment 1: Design a Logical Model

Data Modelling

Attributes of an ERD Entity

- For a given table:
 - Name
 - Relationship to another table
 - Column Names
 - Column Types
 - Primary Keys (if present)
 - Foreign Keys (if present)

Attributes of an ERD Relationship

- Defines which columns are related
- Defines what type of relationship exists:
 - One-to-One
 - One-to-Many
 - Many-to-Many

Data Modelling (continued)

Entity Relationship Diagrams (ERDs)

Attributes of an ERD: Entities & Relationships

Relationship Examples

Conceptual, Logical, Physical Models

Assignment 1: Design a Logical Model

Data Modelling

Relationship Examples

One-to-One: where a given row within a table is associated with only a single row in another table

Table 1: Country – Table 2: Capital City

Table 1:Country	Table 2:Capital
Canada	1:1 Ottawa
USA	1:1 Washington DC
Mexico	1:1 Mexico City

Data Modelling

Relationship Examples

One-to-Many: where a given row within a table can be referenced by multiple rows in another table

Table 1: Country – Table 2: States

Table 1:Country	Table 2:States
Canada	1: ∞ Alberta
Canada	1: ∞ British Columbia
Canada	1: ∞ ... (11 more rows)
USA	1: ∞ Alabama
USA	1: ∞ Alaska
USA	1: ∞ ... (48 more rows)
Mexico	1: ∞ Aguascalientes
Mexico	1: ∞ Baja California
Mexico	1: ∞ ... (30 more rows)

Data Modelling

Relationship Examples

Many-to-Many: where multiple rows within a table can be referenced by multiple rows in another table

Table 1: Employee – Table 2: Employee Type

For this example, imagine the following:

Previously, A was an instructor at DSI (python) *and* a TA at DSI (SQL). Thomas remains the instructor for SQL. Ananya is now the TA. If a table contained Employee ID and Employee Type ID, A would be listed twice.

Employee	Employee ID	Type	Type ID
Thomas	001	Instructor	10
A	002	TA	11
Ananya	003		

Employee ID	Type ID
001	∞:∞ 10
002	∞:∞ 10
002	∞:∞ 11
003	∞:∞ 11

We'd create additional Many-to-Many relationships if we created a course table, because A and Ananya are involved in multiple courses!

Data Modelling (continued)

Entity Relationship Diagrams

Attributes of an ERD: Entities & Relationships

Relationship Examples

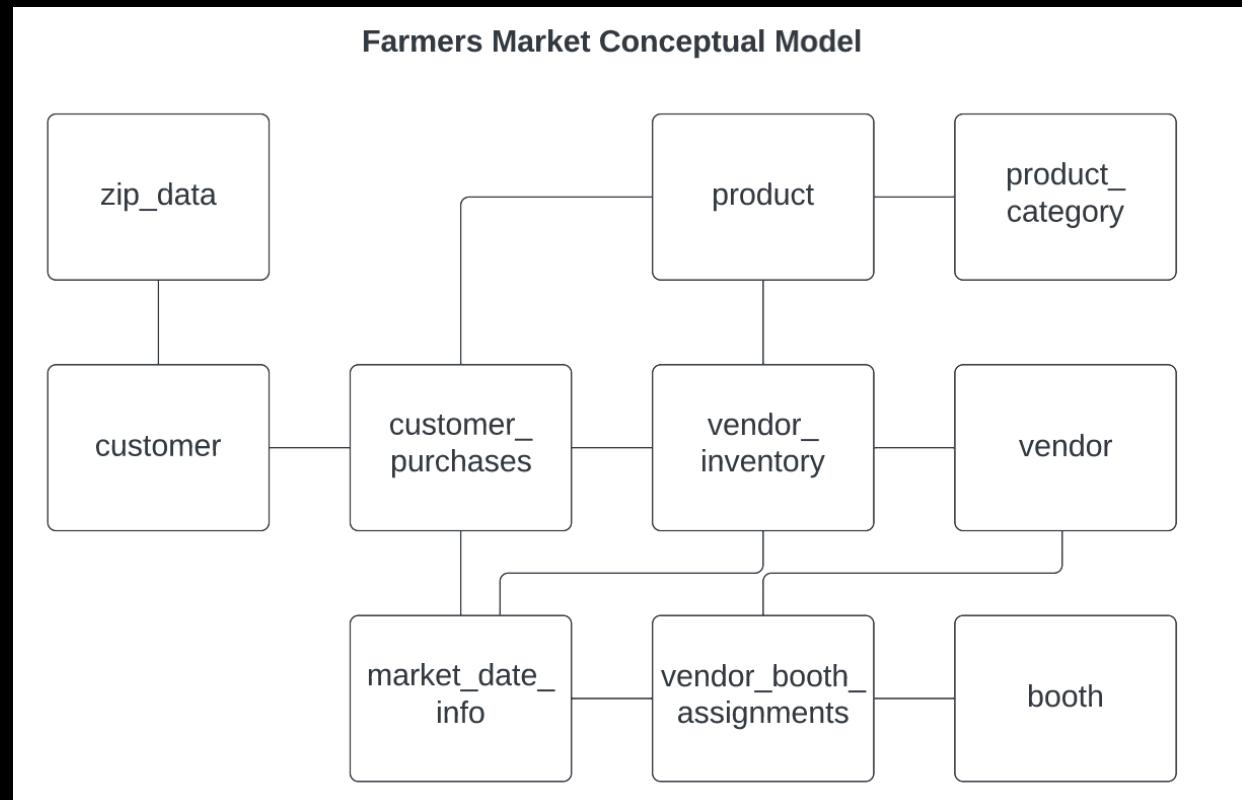
Conceptual, Logical, Physical Models

Assignment 1: Design a Logical Model

Data Modelling

Conceptual Models

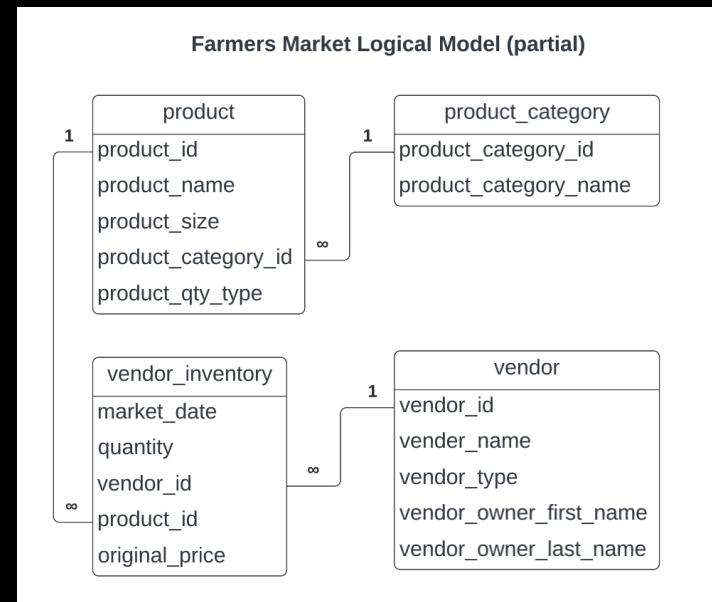
- Define the tables (objects/entities) and their relationships
- Our Farmers Market database:
 - 10 tables
 - Relationships between these tables
 - e.g. product and product_category: *what type of thing a product is*
 - product and customer_purchases: *what products a customer has bought*
 - product and vendor_inventory: *what products each vendor has available*
 - Not all tables share a relationship to one another, but all tables have at least one relationship



Data Modelling

Logical Models

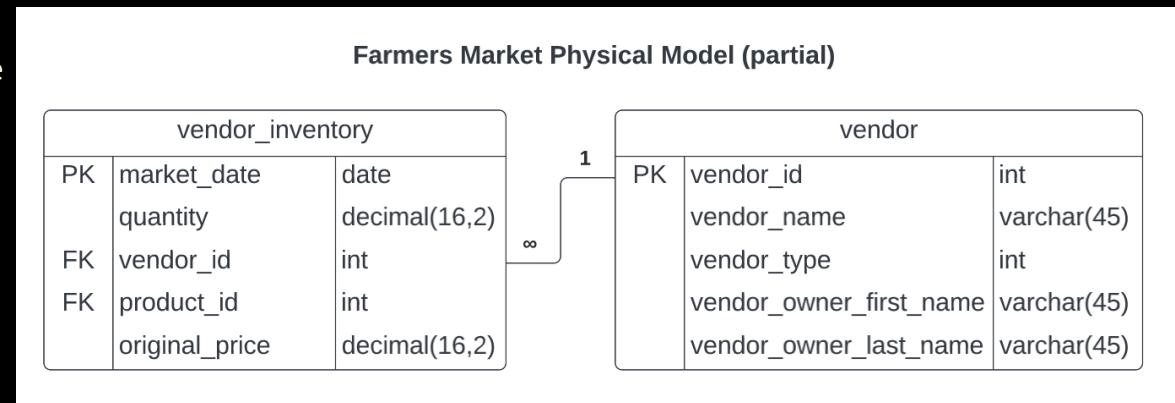
- Add additional detail to the conceptual model by adding column names for each table
- Often indicate the type of relationship
 - One-to-One
 - One-to-Many
 - Many-to-Many
- Our (partial) Farmers Market database:
 - product (5 columns) shares a One-to-Many relationship with vendor_inventory (5 columns) on product_id
 - product_category (2 columns) shares a One-to-Many relationship with product (5 columns) on product_category_id
 - _____ shares a _____ relationship with _____ on vendor_id



Data Modelling

Physical Models

- Add additional detail to the logical model by adding key type and column data type
- Our (partial) Farmers Market database:
 - vendor_id (int) is the PK for vendor, which shares a One-to-Many relationship with vendor_inventory on vendor_id (FK)
 - product_id (int) is a FK for vendor_inventory (*so elsewhere in this diagram, we'd connect this to a PK of another table*)
 - market_date (date) is the PK for vendor_inventory
 - why? 🌸💬 Think, Pair, Share



Data Modelling (continued)

Entity Relationship Diagrams

Attributes of an ERD: Entities & Relationships

Relationship Examples

Conceptual, Logical, Physical Models

Assignment 1: Design a Logical Model

Data Modelling

Assignment 1: Design a Logical Model

Q1) Create a logical model for a small bookstore. 

At the minimum it should have employee, order, sales, customer, and book entities (tables). Determine sensible column and table design based on what you know about these concepts. Keep it simple, but work out sensible relationships to keep tables reasonably sized. Include a date table. There are several tools online you can use, I'd recommend [Lucidchart](#) or [Draw.io](#).

Q2) We want to create employee shifts, splitting up the day into morning and evening. Add this to the ERD.

Q3) The store wants to keep customer addresses. Propose two architectures for the CUSTOMER_ADDRESS table, one that will retain changes, and another that will overwrite. Which is type 1, which is type 2? *Hint, search type 1 vs type 2 slowly changing dimensions.*

Bonus: Are there privacy implications to this, why or why not?

Q4) Review the AdventureWorks ERD here: [link](#)

Highlight at least two differences between it and your ERD. Would you change anything in yours?

