**CS 31 Database Management Programming Lecture 1**

**Data** – recorded facts and numbers

**Database** – structure used to hold or store data

<http://www.northeastern.edu/levelblog/2016/05/13/how-much-data-produced-every-day/>

Databases are vital to most web, smartphone, and business applications today.

There are different types of databases. We will be focusing on the most commonly used type, the relational database. We will briefly discuss NoSQL databases towards the end of the semester.

<https://db-engines.com/en/ranking>

**Relational Databases** – used for structured data and transactions

**NoSQL Databases** – used for large amounts of data with less rigid structure

<http://www.ccs.neu.edu/home/kathleen/classes/cs3200/SQLvsNoSQL.pdf>

Relational databases store data in related tables. A table consists of rows and columns (think of a spreadsheet). Each row (record) has data about a particular occurrence or instance. Each column (field) stores a characteristic common to each row. These characteristics adhere to datatype designations that are made when creating the database.

Here is a table with student data.

**STUDENT\_DATA**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **StudentID** | **Name** | **Email** | **ClassID** | **ClassName** | **Term** | **Grade** |
| 1 | Joe | joe@ivc.edu | 10 | C Programming | Fall 2018 | A |
| 1 | Joe | joe@ivc.edu | 20 | Python Programming | Spring 2019 | B |
| 2 | Jo | jo@ivc.edu | 10 | C Programming | Fall 2018 | B |
| 2 | Jo | jo@ivc.edu | 20 | Python Programming | Spring 2019 | A |
| 2 | Jo | jo@ivc.edu | 30 | Database Management | Spring 2019 | C |

Including all this data in one table will likely cause inconsistent data due to entry errors. Breaking up the tables will help reduce entry errors. It will also provide a mechanism for validating certain data.

**STUDENT**

|  |  |  |
| --- | --- | --- |
| **StudentID** | **Name** | **Email** |
| 1 | Joe | joe@ivc.edu |
| 2 | Jo | jo@ivc.edu |

**CLASS**

|  |  |  |
| --- | --- | --- |
| **ClassID** | **ClassName** | **Term** |
| 10 | C Programming | Fall 2018 |
| 20 | Python Programming | Spring 2019 |
| 30 | Database Management | Spring 2019 |

**GRADE**

|  |  |  |
| --- | --- | --- |
| **StudentID** | **ClassID** | **GradeEarned** |
| 1 | 10 | A |
| 1 | 20 | B |
| 2 | 10 | B |
| 2 | 20 | A |
| 2 | 30 | C |

The tables in a relational database are related and these relationships are maintained using "keys."

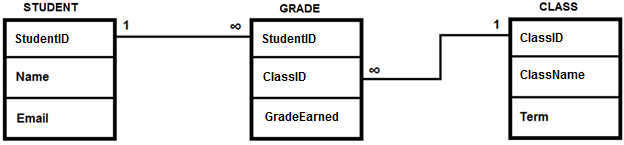
**Primary Key** – a unique identifier for each row in a table

**Composite Key** – a key composed of more than one column value

**Surrogate Key** – a primary key that is automatically generated and assigned in the database

**Foreign Key** – provides the link between two tables, i.e. creates a relationship between the two tables

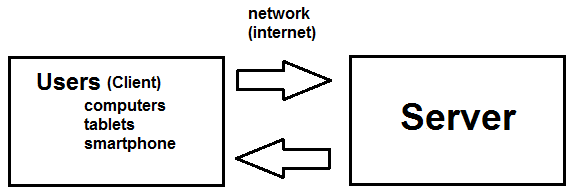
Below is a diagram showing how the tables above are related.

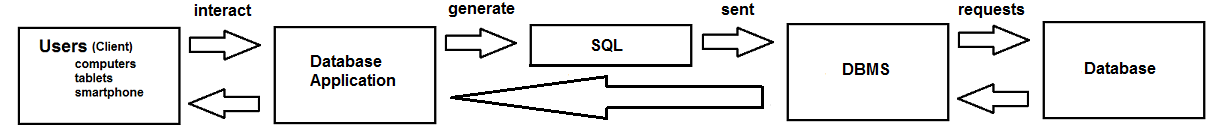


Because of the features of relational databases, data contained within the database can be used to produce accurate information.

**Information** – knowledge derived from data; data presented in a meaningful context; data processed by summing, ordering, averaging, grouping, comparing, or other similar operations

How do we interact with the database? We use DBMS software.





**Database** – collection of related tables and other structures

**Database Management System (DBMS)** – a computer program used to create, process, and administer the database

**Structured Query Language (SQL)** – an internationally recognized standard language that is understood by all commercial DBMS products

**Database Application** – a computer program that serves as an intermediary between the user and the DBMS

Different types of database applications include:

* Single-user – customer manager applications for salesperson to keep track of customer data (financial planner, real estate agent, freelance programmer)
* Multiuser – patient-scheduling application
* E-commerce – amazon.com
* Reporting and data mining – use operational data to help make business decisions

Users fill in forms to "write" data and generate reports to "read" information.

The DBMS provides functionality beyond the storage of data, including:

* Creating the database, tables, indexes (used to find data faster), and other supporting structures
* Modifying and reading database data
* Enforcing referential integrity constraints to eliminate data errors
* Controlling concurrency and performing backup and recovery

**Referential Integrity Constraints** – rules that enforce consistency of related data when making changes to data in a database

**Concurrency** – when multiple users modify the same data at the same time

A database is self-describing because it contains a description of itself

**Metadata** – data concerning the structure of data that are used to describe tables, columns, constraints, indexes, etc.

Metadata is data about data.

An important piece of column metadata is the datatype of the column. Some common datatypes are:

* String/character data – **CHAR[(M)]** e.g. phone number, **VARCHAR(M)** e.g. name of a book, **TEXT[(M)]** e.g. newspaper article, etc.
* Numeric data – **INT[(M)]** e.g. car mileage, **DECIMAL[(M[,D])]** e.g. money, etc.
* Date and time data – **DATE**, **DATETIME[(fsp)]**, **TIME[(fsp)]**, etc.
* Binary data – **BLOB[(M)]** e.g. images, multimedia, etc.

There are two classes of database systems and DBMS products: personal database system (e.g. Microsoft Access) and enterprise-class database systems (e.g. Oracle Database).

In class we will be using MySQL Workbench to connect to MariaDB. MariaDB is a fork of MySQL.

A major component of this class is database design. You can:

* Design a database from existing data (stored in excel, files) - Normalization
* Design a database for a new system – Data Modeling (E-R model)
* Redesign from an old system already implemented in a DBMS – (Correlated Subqueries)

We will focus on designing from existing data and new system design.

Possible database career paths include knowledge workers (users), databaseapplication programmers, and database administrators.

A knowledge worker prepares reports, mines data, and other types of data analysis. A database application programmer writes applications that process the database. A database administrator designs, constructs and manages the database.

Before relational database products there were file managers. Business information systems stored data by grouping similar data into separate files.

The problems with file-processing systems include:

* Data separated and isolated (accounting and sales information separate)
* Data is often duplicated (accounting and sales might duplicate customer info)
* Application program dependent (must be stored and accessed by same programming language)
* Slow (wait for other users to finish using file)
* Not secure (no way to hide private data)

The DBMS solutions for the problems associated with file-processing systems are:

* Data is integrated
* Data duplication is reduced
* DBMS is application independent
* Stored data can be shared
* Security restrictions