SQL and Relational Databases

* Database
  + A way to store information
  + Persist information into some kind of physical medium
    - You can turn the power off and the information is still there
    - Magnetic tape, ssd card, disk drive, cuneiform tablet
  + Relational database
    - Stores information in tables referencing other tables
    - Oracle
    - MariaDB
    - MySQL
    - Postgres
    - Microsoft SQL server
  + NoSQL databases
    - Document based
    - MongoDB

Relational Database

* Data is stored in tables
* Tables can reference other tables
* ERD (Entity Relationship Diagram)
  + Stitches together how the different entities relate to each other
* Tables store information
  + Columns are the attributes
  + Rows are the records

|  |  |  |  |
| --- | --- | --- | --- |
| FIRST\_NAME | LAST\_NAME | AGE | GRADE |
| Tim | Stevens | 12 | 5 |
| Jessica | Smith | 15 | 9 |

Normalization

* Process of eliminating redundancy in your database
* Often increases optimization of things like insertions and deletions but can make your data more difficult to retrieve
* Will decrease the overall size of your database (in gigabytes)

1nf (first normalized form)

* All records are uniquely identifiable
  + All records should have a primary key
  + A primary key is a unique value for every record in the table
  + Used to enforce that each record is distinct
* Fields do not contain array-like information

|  |  |  |
| --- | --- | --- |
| NAME | AGE | PHONE\_NUMBER |
| Adam | 12 | 555-5555,333-33333,222-2222 |
|  |  |  |

* All data in a column should be atomic
  + Cannot be broken down into more useful bit of information

(Not a normalized table)

|  |  |  |
| --- | --- | --- |
| NAME | AGE | GRADE |
| Timothy Smith | 12 | 5 |
| Timothy Smith | 12 | 5 |
| Janet Harrison | 13 | 6 |

(A table in 1nf)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STUDENT\_ID | FIRST\_NAME | LAST\_NAME | AGE | GRADE |
| 101 | Timothy | Smith | 12 | 5 |
| 102 | Timothy | Smith | 12 | 5 |
| 202 | Janet | Harrison | 13 | 6 |

2nf

* In 1nf
* You have no functional dependencies in your table
  + You should have no column that you could calculate using other columns
* Could I calculate this value?

(not in 2nd nf because shooting percentage could be calculated)

|  |  |  |  |
| --- | --- | --- | --- |
| PLAYER\_ID | ATTEMPTS | MADE | SHOOTING\_PERCENTAGE |
| 1001 | 50 | 25 | 50 |
| 1005 | 200 | 20 | 10 |
| 1006 | 50 | 40 | 80 |
| 1002 | 100 | 30 | 30 |
|  |  |  |  |

3nf

* Has to be in 2nd normalized form
* No transitive dependencies
  + You cannot find that information elsewhere in the database

SQL (Structured Query Language)

* Programming language used by relational databases
* From the 60’s \*\*\* Not sure
* Scripting language
  + Not compiled
  + Programming language
  + \*Scripting language tells the machine what to do\*
  + You can run it in separate chunks in any order
* Englishy
* Sub languages of SQL
  + DQL (Data Query Language)
    - SELECT
  + DML (Data Manipulation Language)
    - INSERT, UPDATE, DELETE, SET
  + DDL (Data Definition Language)
    - Creates the schema which is the rules and relationships in your database. Creates table
    - CREATE, DROP, ALTER, CONSTRAINT
  + TCL (Transaction Control Language)
    - COMMIT
    - ROLLBACK
  + DCL (Data Control Language)
    - Used for creating users for your database
      * Users are the software and database developers
      * What privileges do they have?
    - GRANT
    - REVOKE

Multiplicities

* Relationships between your entities
* 3 types
  + 1 -1
    - Every record in one tables matches to one and only one record in another table
    - Ex tax to employee
  + 1-many
    - One record can match to many records in one table.
    - School – Student (1-many)
    - Team – player
    - Department - employee
  + Many-many
    - Many records in one table match to many records in another table
    - Junction table
    - Student – class
    - Game - player

Placing Foreign Keys

* The foreign key goes on the child record
  + The many in a one to many
* A Parent must exist before a child record can be add
  + You must have a sales department before you can have a child record reference the sales department
* The column that you place a foreign key does not have to be unique. However, the foreign key must reference a column that is unique
* CONSTRAINT FK\_KEY\_NAME FOREIGN KEY CHILD(SOMETHING\_ID) REFERENCES PARENT(SOMETHING\_ID);

Isolation levels

|  |  |  |  |
| --- | --- | --- | --- |
| Isolation Level | Dirty Reads | Non-repeatable | phantoms |
| Read Uncommitted | yes | yes | yes |
| Read Committed | no | yes | yes |
| Repeatable read | no | no | yes |
| Serializable | no | no | no |

PL/SQL (ProceduraL SQL)

* Procedures
  + Capable of performing DML
  + They can have 0 to many inputs
  + They can have outputs
* Functions
  + Cannot edit information
  + Must have at least one input
  + Always give you an output
  + Scalar
    - Can be applied to a single field
    - UPPERCASE()
    - ROUND()
  + Aggregate
    - Are applied to a column
    - COUNT()
    - SUM()
    - AVG()

Joins and Unions (Set Operators)

* In normalized relational databases queries can require you to need multiple tables
* The necessary information is spread across more than one table
* Joins and Unions denormalize data be combining tables
  + Joins
    - Combine tables horizontally based on predicate
    - INNER JOIN
    - LEFT JOIN
    - RIGHT JOIN
    - CROSS JOIN
    - THETA JOIN
    - OUTER JOIN
  + Set Operators
    - Combine tables vertically
    - Must be the same number of columns
    - Do not require a predicate
    - UNION ALL
    - MINUS