CS 4650 – Topic 2

**Databases –** mechanism to manage data

1. Insert new data
2. Retrieve data
3. Modify data
4. Remove data

|  |  |
| --- | --- |
| Relational (SQL) | Non-relational (NoSQL) |
| 1. Highly structured 2. Elements linked through relations 3. Commands passed through SQL to DB servers 4. 50 year old tech | 1. Loosely structured 2. 4 main forms of data storage 3. Commands passed through ad-hoc/ proprietary languages 4. Less than 20 years old tech |

**Relational DB –** contain a number of tables (rows & columns). Similar to OOP (table vs class, row vs instance/object)

A table with text on it

Description automatically generated

Using SQL, a program can ask DB to return, e.g. the Name and Location for CS4650

Query: SELECT name, location FROM classes WHERE number=’CS4650’

Return: Big Data, 8-04

1. Apply design principles in OOP
   1. One table does one thing
   2. Don’t duplicate data (assign a value to a variable and reference it)
   3. Avoid errors by using strongly typed variables – and columns! (eg. Age can’t be a string

**Primary Key –** RBDMS needs a way to uniquely identify each row in a table.

* One of column values must be designated as the *primary* *key*
* Each row must have a value for the primary key, no duplicate
* In above table, the Number column is the primary key
* If situations arise that a field can’t be designated as primary key, designer can create additional column hidden from users and have unique integers as primary key (primary key autoincrement)

**SQL commands**

1. INSERT INTO classes (number, name, location, time) VALUES (‘CS3560’, ‘Object Oriented’,’6-04’,’8am’); this insert new row into table
2. UPDATE classes SET time = '2pm' WHERE number = 'CS3800'; this changes one of the rows of the table
3. DELETE FROM classes WHERE number='CS5250'; This removes one row from the table
4. SQL SELECT has options to JOIN data between tables.

For example, ask for a list of all classes Bob is taking

A table with numbers and time

Description automatically generated with medium confidence

1. SQL can group data, sort data, compute values

**DB Normalization –** reduce information stored and increase reliability (searchable, usable, changeable)

1. 3 main reasons – minimize duplicate data, minimize/avoid data modification issues, simplify queries. Check out this non-normalized DB
2. Primary normal forms – first normal form, second normal form, and third normal form

A close-up of a data

Description automatically generated

* This data is hard to update. To change a room for a class, we must find all places where it’s referenced, extract old location and insert new location at that spot. Missing a spot constitutes an U***pdate Anomaly***
* Adding a class is impossible until at least a student is taking the class aka ***Insert Anomaly***
* Bob may leave the school and since his row referencing CS5250 will be deleted, the whole class disappear. **Deletion Anomaly**

**First Normal Form**

Requirements:

1. Primary key that uniquely identifies each row (eg. Student ID)
2. No repeated column (eg. Class 1, Class 2, Class 3)
3. Each column should have atomic value (eg. Information in each cell can’t be one text string

A screenshot of a form

Description automatically generated

**Second Normal Form**

Requirement:

1. Be in first normal form
2. All non-key column are dependent upon the table’s primary key
   1. The primary identifies a particular row in the table, that row represents some ‘object’

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

Benefits:

1. Bidirectional linkage (eg. Find students in a class, find which classes a student is taking)
2. Remove redundancy
3. Remove update, insert, or delete anomalies
4. Faster searching

**Third Normal Form**

**A close-up of a form

Description automatically generated**

**Non-relational DB**