**Lesson 2 –**

* Hw #1 solutions to fixing Primary Key violations
  + You can either:

1. Fix the table (remove the duplicate value that is causing the problem)
2. Get rid of the constraint

* Best practices:
  + Read the error message
  + Consult the slides and your notes
  + Google it
  + Leave enough time for it
* Coding interview consist of:
  + Sharing your screen (walk through your code and explain what you have done).
  + Speak Clearly (try and state things simply at the same time be precise about your vocabulary)
    - Schema instead of table
    - Attributes instead of columns
  + Be prepared to answer questions (you interviewer might want to ask clarifying questions)
* Do:
  + State the problem clearly
  + State the assumptions you made.
    - Like this hw, I thought the data was corrupt so I fixed the table or I removed the constraint (changed the schema).
  + State your tools and methods you picked.
  + Explain how the code solves the problem.
  + Address the potential blind spots.
    - **Like is there any duplicates in the data that may effect the constraints**?
  + Practice it.
* Donts:
  + Be afraid
  + Rush through it
  + Go into too much detail
  + Feel the need to fill dead air.
  + Try and wing it.
* **RDBMS a closer look:**
  + FS does not:
    - Handle resource limitation
    - Have robust query language to retrieve data
    - Handle concurrent file access
    - Not resilient for system crashes
    - Password Protection
  + RDBMS:
    - Handles it for CPU, RAM, and disk
    - Has SQL
    - Comes with it out of the box
    - Fail safe
    - Complex security policies around tables.
* Advantages of RDBMS:
  + Data Independence
  + Efficient Data Access
  + Data Integrity and Security
  + Data Administration
  + Concurrent Access and Crash Recovery
  + Reduced App dev time?
* Data Model:
  + **Abstraction** – Hides most of the storage related complexities of the data
  + **Model** – As such it is decoupled from reality.
  + **Relational Model** – Wide-spread data model that RDBMS supports.
* **Relational Algebra, a Primer –** 
  + Relational Algebra is a procedural quert language that:
    - Takes a relation as input
    - Applies relational algebra operator
    - Outputs another relaion
  + Projection, Selection, Union, Set Difference, Cross Product (SYMBOLS FROM 331)
  + **Projection** – Used to project required attributes(columns) from a relation
    - Removes duplicate values

(**check slides for example) 🡪 or look at Database class notes**

* + **Selection -** Used to select required tuples from a relation
    - Using the same relation

**check slides for example) 🡪 or look at Database class notes**

* **SQL Restraints:**