Recap

- Data-savviness is the future!
- Notion of a DBMS
- The relational data model and algebra: bags and sets
- SQL Queries
- SQL Modifications
- Next: SQL DDL

So far…

- We considered SQL queries and modifications
- This is the so-called "DML" part of SQL
  - Data Manipulation Language
- Next, we will cover the "DDL" part of SQL
  - Data Definition Language
- That is, setting up the relations (or "schema") so that you can then then query (SFWGH) or modify (Insert/Delete/Update) as need be

Defining a Relation

- Simplest form of adding a relation is:

```
CREATE TABLE <name> (
    <list of attributes and domains>
);
```

- And removing a relation is:

```
DROP TABLE <name>;
```

List of Attributes and Domains

- For each attribute in the relation we list the corresponding domain
- Common types include:
  - INTEGER
  - REAL
  - CHAR (n) = fixed length string of n characters
  - VARCHAR (n) = variable length string of up to n characters
  - DATE = ‘yyyy-mm-dd’
  - TIME = ‘hr:min:sec’

Example

- Create a table called Manager

```
CREATE TABLE manager (
    manager_name VARCHAR(20),
    age INTEGER
);
```

The Notion of Keys

- It is very useful to be able to be able to identify tuples in a relation
- Also true in the real world:
  - StudentID or email address to identify students
  - SSN or passport number to identify people
- A set of attributes is said to form a *key* if no two tuples can have the same value for all those attributes

- We do so using the PRIMARY KEY or UNIQUE keyword
  - (distinction later)

Example Primary Key Usage

```
CREATE TABLE manager (
    manager_name VARCHAR(20),
    age INTEGER,
    manager_id CHAR(4),
    PRIMARY KEY (manager_id)
);
```


UNIQUE vs. PRIMARY KEY

- Difference 1:
  - There can be only one PRIMARY KEY (hence the name!)
  - There can be many UNIQUEs

- Difference 2:
  - No attribute of a PRIMARY KEY can be NULL
  - Attributes within UNIQUEs can be NULL, and there can be several tuples with NULL
    - (So NULL is excluded from enforcement of UNIQUE)

Example Unique Usage

```
CREATE TABLE manager (
    manager_name VARCHAR(20),
    age INTEGER,
    manager_id CHAR(4),
    PRIMARY KEY (manager_id),
    UNIQUE (manager_name)
);
```


Other Declarations

- Two other declarations we can make for attributes:
  - NOT NULL means that the value for the attribute can never be NULL
  - DEFAULT <value> means that if there is no specific value known for this attribute's component in the tuple, use the stated <value> (as opposed to NULL)

Example Usage of Declarations

```
CREATE TABLE manager (
    manager_name VARCHAR(20),
    age INTEGER
        DEFAULT 50,
    manager_id CHAR(4),
    PRIMARY KEY (manager_id),
    UNIQUE (manager_name)
);
```


Adding Attributes

- We can add new columns to a relation by using the following:

```
ALTER TABLE <name>
ADD <attribute declaration>,
ADD <attribute declaration>, …;

ALTER TABLE Manager
ADD address VARCHAR (20),
ADD income REAL DEFAULT 10000.0;
```


Deleting Attributes

- We can delete columns from a relation by using the following:

```
ALTER TABLE <name>
DROP <attribute>,
DROP <attribute>, …;

ALTER TABLE Manager
DROP address,
DROP income;
```


Quick Demo

- Creating the manager relation

  CREATE TABLE manager (manager_name VARCHAR(20), age INTEGER DEFAULT 50, manager_id CHAR(4), PRIMARY KEY (manager_id), UNIQUE (manager_name));

- Adding a tuple (Success!)

  INSERT INTO Manager (age, manager_id)

  VALUES (23, 'sd45');

- Adding a tuple (Failure)

  INSERT INTO Manager (manager_name, age)

  VALUES ('John Smith', 123);

- Changing the schema by adding attribs

  ALTER TABLE Manager

  ADD address VARCHAR (20),

  ADD income REAL DEFAULT 10000.0;

- Changing the schema by dropping attribs

  ALTER TABLE Manager DROP address, DROP income;

- Deleting the relation entirely

  DROP TABLE Manager;