

1. What is a relational database?
  - a. "a database structured to recognize relations among stored items of information."
  - b. Tables of columns and rows, with a unique key per row.
  - c. Each table typically represents one "entity type"
2. Why a relational database?
  - a. Strengths of relational databases
    - i. Data is more compact because it isn't repeated
    - ii. Enforces relationships
    - iii. Flexible and well-established.
    - iv. SQL
  - b. Weaknesses of relational databases
    - i. You have to plan ahead and organize your data which can be hard with complex datasets (Fixed Schema)
    - ii. Cannot be directly exposed to the internet need a layer to keep it secure
    - iii. Not good for large records > 2GB
    - iv. Speed?
      1. Other data formats without joins can be faster
      2. But this means that sometimes unplanned for queries cannot be performed
    - v. Scalability?
      1. Can't really scale to multiple servers...
    - vi. cite: <http://it.toolbox.com/blogs/enterprise-solutions/some-pros-cons-of-relational-databases-24144>
      1. Performance problems associated with re-assembling simple data structures into their more complicated real-world representations.
      2. Lack of support for complex base types, e.g., drawings.
      3. SQL is limited when accessing complex data.
      4. Knowledge of the database structure is required to create ad hoc queries.
3. What is SQL? (ess-que-el)
  - a. Standard Database language
  - b. Came from [Edgar F. Codd](#)'s influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks."
  - c. A little history, originally it was SEQUEL (*Structured English Query Language*) which was already trademarked. Now officially it is SQL (ess-que-el) Structured Query Language.
  - d. Lots of versions:
    - i. MySQL
    - ii. SQLite
    - iii. Oracle
    - iv. Microsoft SQL Server

#### 4. Primary Keys and Foreign Keys

- a. One to Many
- b. Many to Many
- c. One to One

#### 5. Creating Tables

##### a. DataTypes

- i. CHARACTER [(length)] or CHAR [(length)]
  - 1. List of characters which are exactly the length specified, will be truncated or lengthened with whitespace to fit length
- ii. VARCHAR (length)
  - 1. List of characters with a max of length
- iii. BOOLEAN
  - 1. True/False
- iv. SMALLINT
  - 1. integer between  $2^{-15}$  and  $2^{15} - 1$  (16 bytes, last byte for +/-)
  - 2. Digits to the right of the decimal point are truncated
- v. INTEGER or INT
  - 1. Integer between  $2^{-31}$  and  $2^{31} - 1$  (32 bytes)
  - 2. Digits to the right of the decimal point are truncated
- vi. DECIMAL [(p[,s])] or DEC [(p[,s])]
  - 1. Number with decimal
  - 2. Precision == number of digits it will contain
  - 3. Scale == number of digits to the right of the decimal point
  - 4. Precision defaults to 38, Scale defaults to 0
  - 5. If you exceed on the left an error is thrown
  - 6. If you excess on the right it is truncated
- vii. NUMERIC [(p[,s])]
  - 1. Same as decimal
- viii. REAL
  - 1. Approximate
  - 2. 6 decimal digits precision
- ix. FLOAT(p)
  - 1. Approximate
  - 2. Default Precision 64
- x. DOUBLE PRECISION
  - 1. Approximate
  - 2. 15 decimal digits precision
- xi. DATE
  - 1. Format `DATE '1999-01-01'`
- xii. TIME
  - 1. Format `TIME '00:00:00'`
- xiii. TIMESTAMP
  - 1. Format `TIMESTAMP '1999-12-31 23:59:59.99'`
- xiv. CLOB [(length)] or CHARACTER LARGE OBJECT [(length)] or CHAR LARGE OBJECT [(length)]
  - 1. Allows for objects longer than a CHAR, max size 2 GB
- xv. BLOB [(length)] or BINARY LARGE OBJECT [(length)]

1. Allows for large Binary data, max size 2 GB
- b. CREATE TABLE EXAMPLE
- c. Designating Primary Keys, Foreign Keys, Incrementing
  - i. Add Primary Keys and Foreign Keys to EXAMPLE

```
mysql -u root -p
```

```
CREATE DATABASE students;  
use students;
```

```
show tables;
```

```
CREATE TABLE Students (  
    id INT AUTO_INCREMENT,  
    firstName VARCHAR(30),  
    lastName VARCHAR(30),  
    birth DATE,  
    grad INT,  
    PRIMARY KEY(id));
```

```
show columns from Students
```

```
CREATE TABLE Department  
(id INT AUTO_INCREMENT,  
    department CHAR(20),  
    PRIMARY KEY(id));
```

```
CREATE TABLE Courses  
(id INT AUTO_INCREMENT,  
    class CHAR(20),  
    departmentId INT,  
    PRIMARY KEY(id),  
    FOREIGN KEY (departmentId) REFERENCES Department(id));
```

```
CREATE TABLE Student_Courses  
(studentId INT,  
    courseId INT,  
    FOREIGN KEY (studentId) REFERENCES Students(id),  
    FOREIGN KEY (courseId) REFERENCES Courses(id));
```

```
INSERT INTO Students(firstName, lastName, birth, grad)  
VALUES ('Buffy', 'Summers', '1987-10-22', 2009),  
('Xander', 'Harris', '1990-4-16', 2012),
```

```

('Willow','Rosenberg', '1989-8-12', 2011),
('Cordelia','Chase', '1990-2-4', 2011),
('Amy','Madison', '1987-10-22', 2009),
('Cassie','Newton', '1990-4-16', 2012),
('Devon','MacLeish', '1989-8-12', 2011),
('Jesse','McNally', '1990-2-4', 2011),
('Larry','Blaisdell', '1987-10-22', 2009),
('Percy','West', '1990-4-16', 2012),
('Scott','Hope', '1989-8-12', 2011);

```

```

INSERT INTO Department(department)
VALUES
('English'),('Math'),('Science'),('History'),('Language'),('Elective'
);

```

```

INSERT INTO Courses(class, departmentID)
VALUES ('English 9', 1),('English 10', 1),('English 11',
1),('English 12', 1),
('Algebra', 2),('Geometry', 2),('Pre-Calculus', 2),('Calculus',
2),
('Biology', 3),('Chemistry', 3),('Physics', 3),('Earth
Science', 3),
('US History', 4),('World History', 4),('Geography',
4),('Government', 4),
('French', 5),('Latin', 5),('Spanish', 5),('Chinese', 5),
('Art', 6),('Music', 6),('Acting', 6),('P.E.', 6);

```

```

INSERT INTO Student_Courses(studentId, courseId)
VALUES (1,1),(1,5),(1,9),(1,13),(1,17),(1,21),
(2,2),(2,6),(2,10),(2,14),(2,18),(2,22),
(3,3),(3,7),(3,11),(3,15),(3,19),(3,23),
(4,4),(4,8),(4,12),(4,16),(4,20),(4,24),
(5,1),(5,5),(5,9),(5,13),(5,17),(5,21),
(6,2),(6,6),(6,10),(6,14),(6,18),(6,22),
(7,3),(7,7),(7,11),(7,15),(7,19),(7,23),
(8,4),(8,8),(8,12),(8,16),(8,20),(8,24),
(9,1),(9,5),(9,9),(9,13),(9,17),(9,21),
(10,2),(10,6),(10,10),(10,14),(10,18),(10,22),
(11,3),(11,7),(11,11),(11,15),(11,19),(11,23);

```

```

select * from Students;
select * from Students where grad = 2009;

```