

# 9/6 - Intro to Relational Model

## Announcements

- Form teams by Friday 9/8.
- Send email with team name in subject line to professor and all three TAs in order to be added to the github organization.

## Reading Quiz - Noteworthy Questions

1. Q1: B.
2. Q2: B. Every relation is a table in a relational database.
3. Q3: C. Database Management System
4. Q4: A. Think of a view as a virtual table but not an actual table that is stored. You can query views just as you would tables.
5. Q5: D.

## Basic Terminology and Concepts

An entity is an object that is represented in the database. In the database entities are represented as Relations and Tables. (i.e. A customer, an order, a book.)

**Field/Attribute/Column** - Each different field holds different data within tables. A table will have one or more fields to hold data.

**Row/Tuple/Record** - An entry in a table. For example, one specific employee record in the Employee table.

**Cell/Value** - The value of a specific attribute in a specific record.

**Keys** - Primary Keys (PK) and Foreign Keys (FK). A PK uniquely identifies each row within a table (usually underlined in diagrams). Keys can be a single field or a combination of multiple fields. The FK of a record in a child table points to the PK of a record in the parent table.

**Constraints** - you can place constraints on fields to disallow various inputs, for example uniqueness or disallowing null values.

**Schema** - the structure of a database (defining table names, datatypes and domains, fields, etc.).

## Data Types

**CHAR** - Character data with static length.

**VARCHAR** - Character data with dynamic length with a set upper limit.

**SERIAL** - Sequential Integer datatype which autoincrements.

**DATE vs TIMESTAMP** - Date has date without time. Timestamp includes the time.

**BLOB** - Arbitrary binary data for large amounts of data, usually used for documents, photos, or videos, etc.

## Relationships

An association between two Entities.

**Relationship Types:**

- **One-to-One:** For example, storing larger objects such as documents in another table while keeping a one-to-one relationship.
- **One-to-Many:** For example, a customer and order relationship. The customer FK would be stored in the order table, since an order can only have one customer. And then a customer can have many orders.
- **Many-to-Many:** For example, students and classes, a class can have multiple students and a student can enroll in multiple classes.
- **Many-to-One:** Same as one-to-many.

Note: The child table contains the foreign key which points back to the primary key in the parent table.

### Practice Problem 1

The relationship between customer and order is one to many because a customer can have many orders but each order can only be placed by (associated with) one customer.

## SQL

**Data Manipulation Language** (i.e. INSERT, UPDATE, DELETE) DML statements can be rolled back and they operate on records.

**Data Definition Language** statements (i.e. CREATE, DROP) DDL statements cannot be rolled back.

### SELECT

The select statement can select attributes from records in tables.

A basic select statement which lists out all the empids from the table called Employee.

```
SELECT empid, FROM Employee
```

### Practice Problem 2

```
SELECT * FROM Employee WHERE depid < 5;
```

This query returns zero rows from the table because there are no entries where the `depid` is less

than 5. If there query were to actually return records it would also include all the attributes (total of 5 attributes/fields) in the result since we use \* in the SELECT statement.

Note: If you want to retrieve null records use `WHERE depid IS NULL`.

### Practice Problem 3

Two rows will be returned from this query. We know this because the `IS NOT NULL` removes one row from the possible resultant records and then the ones in the remaining that have state of PA or TX are empid 2 and 5.