

Relational Data Model

- **Essential Elements**
 - **Data structure** - data are organized in the form of tables (relations) with rows and columns
 - **Data integrity** - Facilities are included to specify (business) rules that maintain the integrity of the data when they are manipulated.
 - **Data manipulation** - Powerful operations (using the SQL language) are used to manipulate data stored in the relations
- **How will this help you?**

Relation

- A **relation** is a named, two-dimensional table of data. Each relation (or table) consists of a set of named columns and an arbitrary number of unnamed rows. Each row of a relation corresponds to a record that contains data (attribute) values for a single entity.
 - **RECORD**: a row of a relation, data for single entity
 - Unique to relational database?
- Representing Relations

TEXT:

Employee1 (Emp_ID, Name, Dept, Salary)

GRAPHICAL:

Employee1

Emp_ID	Name	Dept	Salary
--------	------	------	--------

Relational Keys

- **Setting the Stage:**
 - Observation: Databases can have tens/hundreds of relations, Relations can have thousands of records
 - Challenge: Retrieving data, Capturing relationships among data
- **Primary Key** - Attribute that uniquely identifies each row/record in a relation

Employee1 (Emp_ID, Name, DeptName, Salary)

Department (DeptName, Location, Fax)

- **Question**: Can 2 relations have same primary key?
 - **Question**: Can 2 records in a relation have same primary key?
 - **Question**: How are “key” and “keyword” related (from classroom prep, “Key is similar to a keyword when doing a search. It is a word or attribute that represents the context of a particular article/table/information at the highest level of significance.”) Is this correct?
 - **Question**: Why important?
- **Foreign Key** - Attribute in a relation of a database that serves as the primary key of another relation in the same database.

Employee1 (Emp_ID, Name, DeptName, Salary)

- **Question**: Why important?

NOTE: This terminology is embedded in relational database software!

Integrity Constraints

- **Perspectives**

- **From the book:**

- Facilitate maintaining the accuracy / integrity of data in database.

- **From class prep:**

- ensure data have same nature throughout the system,
 - keep data from being inaccurate (bad),
 - prevents errors from occurring,
 - a rule that helps maintain the quality of data.

- **Integrity (from Merriam-Webster)**

- 1) firm adherence to a code (Incorruptibility)
 - 2) an unimpaired condition (Soundness)
 - 3) the quality or state of being complete or undivided (Completeness)

- **Important Types**

- **Defined as constraints**

- What are constraints on data to ensure “integrity”?

- **Domain Constraints**

- Clarifies allowable attribute values
 - Includes data type, length, allowable values

- **Entity Constraints**

- No primary key attribute may be null
 - Null is not the same as empty!

- **Referential Constraints**

- If there is a foreign key in one relation, either each foreign key value must match a primary key value in the other relation or else the foreign key value must be null.

→ **Bottom Line:** Integrity constraints ensure certain types of “accuracy” or soundness (domain, entity, referential) in data. However, just as a spell checker or grammar checker cannot ensure perfection in a document, integrity constraints cannot ensure accuracy in data (you can still enter inaccurate information and make mistakes). At the same time, integrity constraints ensure that certain rules are followed!