Database Design Process

- Requirements analysis
- Conceptual design: Entity-Relationship Model
- Logical design: transform ER model into relational schema
- Schema refinement: Normalization
- Physical tuning

Goals

- Understand:
  - The relational model
  - Relational model terminology

Relational Database

- A relation is a two-dimensional table
- Relation schema describes the structure for the table
  - Relation name
  - Column names
  - Column types
- A relational database is a set of relations

Relation Example

```plaintext
EMPLOYEE(EmployeeNumber:integer,
  FirstName:string,
  LastName:string,
  Department:string,
  Email:string,
  Phone:integer)
```

<table>
<thead>
<tr>
<th>EmployeeNumber</th>
<th>FirstName</th>
<th>LastName</th>
<th>Department</th>
<th>Email</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Jerry</td>
<td>Johnson</td>
<td>Accounting</td>
<td><a href="mailto:mail@somewhere.com">mail@somewhere.com</a></td>
<td>239-9957</td>
</tr>
<tr>
<td>300</td>
<td>Mary</td>
<td>Abernathy</td>
<td>Finance</td>
<td><a href="mailto:MBA@somewhere.com">MBA@somewhere.com</a></td>
<td>444-8898</td>
</tr>
<tr>
<td>360</td>
<td>Liz</td>
<td>Strothers</td>
<td>Finance</td>
<td><a href="mailto:LStrothers@somewhere.com">LStrothers@somewhere.com</a></td>
<td>777-0998</td>
</tr>
<tr>
<td>400</td>
<td>Tom</td>
<td>Cashiers</td>
<td>Accounting</td>
<td><a href="mailto:TJ@somewhere.com">TJ@somewhere.com</a></td>
<td>255-9957</td>
</tr>
<tr>
<td>500</td>
<td>Tom</td>
<td>Jackson</td>
<td>Production</td>
<td><a href="mailto:TJJackson@somewhere.com">TJJackson@somewhere.com</a></td>
<td>444-9988</td>
</tr>
<tr>
<td>600</td>
<td>Eleanor</td>
<td>Cadets</td>
<td>Legal</td>
<td><a href="mailto:EC@somewhere.com">EC@somewhere.com</a></td>
<td>767-0900</td>
</tr>
<tr>
<td>760</td>
<td>Richard</td>
<td>Bartolome</td>
<td>Legal</td>
<td><a href="mailto:PB@somewhere.com">PB@somewhere.com</a></td>
<td>787-0900</td>
</tr>
</tbody>
</table>
```

Relation

- All entries in a column are of the same kind
- Each column has a unique name
- Cells of the table hold a single value
- The order of the columns is not important
- The order of the rows is not important
- No two rows may be identical
- Rows contain data about entity instances
- Columns contain data about attributes of the entity
Tables That Are Not Relations

Alternative Terminology

- Although not all tables are relations, the terms table and relation are normally used interchangeably.
- The following sets of terms are equivalent:

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Row</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation</td>
<td>Attribute</td>
<td>Tuple</td>
</tr>
<tr>
<td>File</td>
<td>Field</td>
<td>Record</td>
</tr>
</tbody>
</table>

Integrity Constraints (IC)

- IC: condition that must be true for any instance of the database.
- ICs are specified when schema is defined.
- ICs are checked when relations are modified.
- A legal instance of a relation is one that satisfies all specified ICs.
- DBMS should not allow illegal instances.

Keys

- A key is a combination of one or more columns that is used to identify rows in a relation.
- A composite key is a key that consists of two or more columns.
- A set of columns is a key for a relation if:
  1. No two distinct rows can have same values in all key columns, and
  2. This is not true for any subset of the key.
- Part 2 false? A superkey.

Keys in EMPLOYEE Relation

Candidate and Primary Keys

- A candidate key is a key.
- A primary key is a candidate key selected as the primary means of identifying rows in a relation:
  - There is one and only one primary key per relation.
  - The primary key may be a composite key.
  - The ideal primary key is short, numeric and never changes.
Surrogate Keys

- A **surrogate key** as an artificial column added to a relation to serve as a primary key:
  - DBMS supplied
  - Short, numeric and never changes – an ideal primary key!
  - Has artificial values that are meaningless to users
  - Remember Access

**NOTE:** The primary key of the relation is **underlined** below:

- **RENTALPROPERTY** without surrogate key:
  
  RENTALPROPERTY (Street, City, StateProvince, ZipPostalCode, Country, RentalRate)

- **RENTALPROPERTY** with surrogate key:
  
  RENTALPROPERTY (PropertyID, Street, City, StateProvince, ZipPostalCode, Country, RentalRate)

Foreign Keys and Referential Integrity Constraints

- A **foreign key** is the primary key of one relation that is placed in another relation to form a link between the relations
- A **referential integrity constraint**: the values of the foreign key must exist as primary key values in the corresponding relation → No ‘dangling references’

**NOTE:** The primary key of the relation is **underlined** and any foreign keys are in **italics** in the relations below:

**DEPARTMENT** (DepartmentName, BudgetCode, ManagerName)

**EMPLOYEE** (EmployeeNumber, EmployeeName, DepartmentName)

Where EMPLOYEE.DepartmentName must exist in DEPARTMENT.DepartmentName

- Can you name a data model w/o referential integrity?

ICE: Find PK, AK

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Ryan</td>
<td>MD</td>
<td><a href="mailto:jr@gmail.com">jr@gmail.com</a></td>
</tr>
<tr>
<td>Bob</td>
<td>Smith</td>
<td>MD, VA, NY</td>
<td><a href="mailto:bsm@gmail.com">bsm@gmail.com</a></td>
</tr>
<tr>
<td>Alice</td>
<td>Brown</td>
<td>CA</td>
<td></td>
</tr>
<tr>
<td>Jane</td>
<td>Doe</td>
<td>WA</td>
<td><a href="mailto:jd@yahoo.com">jd@yahoo.com</a></td>
</tr>
<tr>
<td>John</td>
<td>Doe</td>
<td>WA</td>
<td><a href="mailto:jd@yahoo.com">jd@yahoo.com</a></td>
</tr>
</tbody>
</table>

ICE: Is This a Relation? Why?

<table>
<thead>
<tr>
<th>A</th>
<th>X</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Ryan</td>
<td>MD</td>
<td><a href="mailto:Jr@gmail.com">Jr@gmail.com</a></td>
</tr>
<tr>
<td>Bob</td>
<td>Smith</td>
<td>MD, VA, NY</td>
<td><a href="mailto:bsm@gmail.com">bsm@gmail.com</a></td>
</tr>
<tr>
<td>Alice</td>
<td>Brown</td>
<td>CA</td>
<td></td>
</tr>
<tr>
<td>Jane</td>
<td>Doe</td>
<td>WA</td>
<td><a href="mailto:jd@yahoo.com">jd@yahoo.com</a></td>
</tr>
<tr>
<td>John</td>
<td>Ryan</td>
<td>MD</td>
<td><a href="mailto:jr@gmail.com">jr@gmail.com</a></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

ICE: Find PK, AK
Summary – Relational Model

- 2-D tables
- Relational schema: structure of table
- Constraints
  - Domain
  - Key
    - Candidate, Primary, Alternate, Surrogate
    - Foreign key – Referential integrity constraint