Data Model

- **Schema**: description of data at some level
  - e.g., tables, attributes, constraints, domains

- **Model**: tools and language for describing:
  - Conceptual and external *schema*
    - Data definition language (DDL)
  - Integrity *constraints*, domains (DDL)
  - *Operations* on data
    - Data manipulation language (DML)
  - Directives that influence the physical schema (affects performance, not semantics)
    - Storage definition language (SDL)
Relational Model

- A particular way of structuring data (using relations)
- Simple
- Mathematically based
  - Expressions (\(=\) queries) can be analyzed by DBMS
  - Queries are transformed to equivalent expressions automatically (query optimization)
    - Optimizers have limits

Relation Instance

- Relation is a set of tuples
  - Atomic values
  - Tuple ordering is immaterial
  - No duplicates
  - **Cardinality** of relation = number of tuples
- All tuples in a relation have the same structure; constructed from the same set of attributes
  - Attributes are named (ordering is immaterial)
  - Value of an attribute is drawn from the attribute’s **domain**
    - There is also a special value **null** (value unknown or undefined), which belongs to no domain
  - **Arity** (or degree) of relation = number of attributes
Relation Instance (Example)

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Address</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>111111</td>
<td>John</td>
<td>123 Main</td>
<td>freshman</td>
</tr>
<tr>
<td>2345678</td>
<td>Mary</td>
<td>456 Cedar</td>
<td>sophomore</td>
</tr>
<tr>
<td>4433322</td>
<td>Art</td>
<td>77 So. 3rd</td>
<td>senior</td>
</tr>
<tr>
<td>7654321</td>
<td>Pat</td>
<td>88 No. 4th</td>
<td>sophomore</td>
</tr>
</tbody>
</table>

Student

Relation Schema

- Relation name
- Attribute names & domains
- Integrity constraints like
  - The values of a particular attribute in all tuples are unique
  - The values of a particular attribute in all tuples are greater than 0
- Default values
Relational Database

- Finite set of relations
- Each relation consists of a schema and an instance

- **Database schema** = set of relation schemas constraints among relations (inter-relational constraints)
- **Database instance** = set of (corresponding) relation instances

**Database Schema (Example)**

- Professor (*Id*: INT, *Name*: STRING, *DeptId*: DEPTS)
- Course (*DeptId*: DEPTS, *CrsName*: STRING, *CrsCode*: COURSES)
- Department(*DeptId*: DEPTS, *Name*: STRING)
Integrity Constraints

- Part of schema
- Restriction on state (or of sequence of states) of database
- Enforced by DBMS
- **Intra-relational** - involve only one relation
  - Part of relation schema
  - e.g., all IDs are unique
- **Inter-relational** - involve several relations
  - Part of relation schema or database schema

Constraint Checking

- **Automatically** checked by DBMS
- Protects database from errors
- Enforces enterprise rules
Kinds of Integrity Constraints

- **Static** – restricts legal states of database
  - Syntactic (structural)
    - e.g., all values in a column must be unique (atomic values)
  - Semantic (involve meaning of attributes)
    - e.g., cannot register for more than 18 credits
- **Dynamic** – limitation on sequences of database states
  - e.g., cannot raise salary by more than 5%

---

Key Constraint

- **A key constraint** is a sequence of attributes $A_1,\ldots,A_n$ of a relation schema, $S$, with the following property:
  - A relation instance $s$ of $S$ satisfies the key constraint iff at most one row in $s$ can contain a particular (or unique) set of values, $a_1,\ldots,a_n$, for the attributes $A_1,\ldots,A_n$
  - **Minimality**: no subset of $A_1,\ldots,A_n$ satisfies the key constraint

- **Key**
  - Set of attributes mentioned in a key constraint
    - e.g., Id in Student,
    - e.g., (StudId, CrsCode, Semester) in Transcript
  - It is minimal: no subset of a key is a key
    - (Id, Name) is not a key of Student
Key Constraint (cont’d)

- **Superkey** - set of attributes containing key
  - (Id, Name) is a superkey of Student

- Every relation has a key

- Relation can have several keys:
  - **Primary key**: Id in Student *(can’t be null)*
  - **Candidate key**: (Name, Address) in Student

Foreign Key Constraint

- **Referential integrity**: Item named in one relation must refer to tuples that describe that item in another
  - Transcript (CrsCode) references Course (CrsCode)
  - Professor(DeptId) references Department (DeptId)

- Attribute A₁ is a **foreign key** of R₁ referring to attribute A₂ in R₂, if whenever there is a value v of A₁, there is a tuple of R₂ in which A₂ has value v, and A₂ is a key of R₂
  - This is a special case of referential integrity: *A₂ must be a candidate key of R₂* *(e.g., CrsCode is a key of Course in the above)*
  - If no row exists in R₂ => violation of referential integrity
  - Not all rows of R₂ need to be referenced: relationship is not symmetric *(e.g., some course might not be taught)*
  - Value of a foreign key might not be specified *(DeptId column of some professor might be null)*
Foreign Key Constraint (Example)

Foreign Key (cont’d)

- Names of the attributes $A_1$ and $A_2$ can be different.
  - With tables:
    Teaching($CrsCode$: COURSES, $Sem$: SEMESTERS, $ProfId$: INT)
    Professor($Id$: INT, $Name$: STRING, $DeptId$: DEPTS)
  - $ProfId$ attribute of Teaching references $Id$ attribute of Professor

- $R1$ and $R2$ need not be distinct.
  - Employee($Id$:INT, $MgrId$:INT, ....)
    - Employee($MgrId$) references Employee($Id$)
    - Every manager is also an employee and hence has a unique row in Employee
Foreign Key (cont’d)

- Foreign key might consist of several columns
  \((\text{CrsCode, Semester})\) of Transcript references
  \((\text{CrsCode, Semester})\) of Teaching

- \(R1(A_1, ..., A_n)\) references \(R2(B_1, ..., B_n)\)
  - \(A_i\) and \(B_i\) must have same domains (although not necessarily
    the same names)
  - \(B_1, ..., B_n\) must be a candidate key of \(R2\)

Inclusion Dependency

- Referential integrity constraint that is not a foreign key
  constraint
  \((\text{CrsCode, Semester})\) of Teaching references
  \((\text{CrsCode, Semester})\) of Transcript

- Target attributes is not a CK in Transcript

- No simple enforcement mechanism for inclusion dependencies in SQL (requires assertions)