Relational Model
- tables (relations)
- columns (attributes)
- rows (tuples)

Relation Schema
- R(A1, A2, ..., An)

Domain

Logical Definition
- Ssn_Numbers = Nine digit character string of integer values between 0 and 9 inclusive (can't start with 0).

Data Type
- String.

Format
- XXXXXXXXX where first X is not 0.

Unit of Measurement (Optional)
- i.e. pounds or kilograms for Weight attribute.

Relation State

r(R) - list of tuples t = {t1, t2, ..., tm}
tuple t = list of values v = {v1, v2, ..., vn}

v1 in the dom(A1)

Types of Constraints
- Model based constraints - relational model (see definition on left i.e. domains)
- Explicit constraints - defined by DDL i.e. SQL part of our database schema.
- - Semantic Constraints - Business Rules (Logic), implemented in the database applications (API).

Explicit Constraints
- - Domain Constraints (Data Types)
- - Key Constraints -
- - Entity Integrity
- - Referential Integrity

Key Constraints

Superkey
SK - subset of attributes such that the following holds for tuples t1, t2.

t1[SK] != t2[SK]

Keys
key is a superkey with the added property that it's minimal superkey.

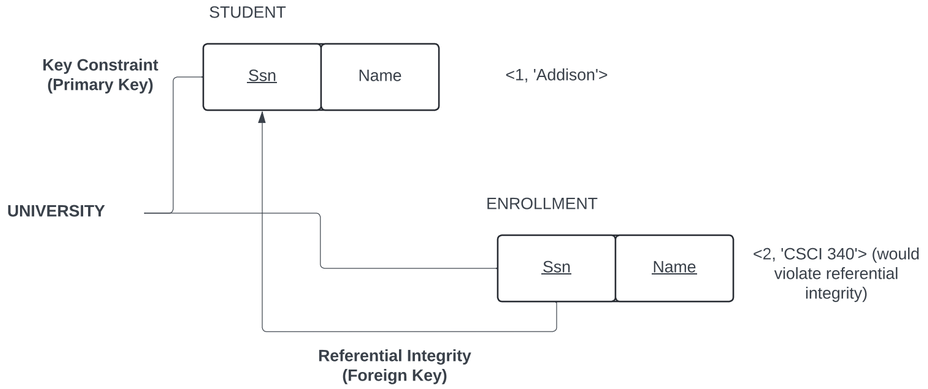
key K has the property if I remove A from K then I have K' that is not a superkey.

t1[K'] = t2[K']

Multiple keys?
- Arbitrarily choose one to be the *Primary Key*.
- Candidate keys - keys, but not the primary key.

Entity Integrity Constraint
- Primary Keys don't have null values.

Referential Integrity Constraint
- Foreign Key (FK) - Subset of Attributes on g1 that refers to the Primary Key (PK) of a tuple t1, or is NULL.



Relational Database Schema
S = {R1, R2, ..., Rn} and integrity constraints IC.

UNIVERSITY = {STUDENT, ENROLLMENT}

Relational Database State
DB = {r1, r2, ..., rn} relation states satisfy the constraints specified in IC.

Valid database state - follows from definition above.

Retrievals and *Updates (Insert, Update, Delete)*.

Domain, Key, Entity Integrity, Referential Integrity.

Insert - entity integrity (<NULL, 'Addison'>), referential integrity (<2, 'CSCI 340'>), domain, key (insert the same tuple twice)*

Update - not a foreign key or a primary key attribute (can only violate domain constraint). Otherwise see case of insert*
REJECT - default.

Delete - referential integrity (see cascading delete in SQL Section)