CS 220

Database Systems

Fall 2019

Lecture 2

Defining Relational Model

The relational model may be visualized as tables and fields, but it is formally defined in terms of sets and set operations.

A Relational Database Schema is a set of relation schemas

A Relation schema is a definition of a single relation

A relations schema **R** with attributes **A =<A1, A2, ..., An>**
is denoted ***R (A1, A2, ..., An)***
where each **Ai** is an attribute name that ranges over a domain **Di**
denoted **dom(Ai)**

Defining Relational Model

Example:

Product (id, name, supplierId, categoryId, price)

R = Product (relation name)

Set A = {id, name, supplierId, categoryId, price}

dom(**price**) is set of all possible positive currency values

dom(**name**) is set of all possible strings that represent people's names

Another Example:

Students(sid: string, name: string, batch: integer, age: integer, gpa: real)

Cartesian Product- Revising Set Theory

The Cartesian product written as **D1 x D2** is a set operation that takes two sets D1 and D2 and returns the set of all ordered pairs such that the first element is a member of D1 and the second element is a member of D2

Example:

D1 = {1,2,3}
D2 = {A,B}
D1 x D2 = {(1,A), (2,A), (3,A), (1,B), (2,B), (3,B)}

Total element in D1 x D2 = 3 x 2 = 6

Formal Definition of a Relation

A relation instance r(R) can also be defined as a subset of the Cartesian product of the domains of all attributes in the relation schema. That is

$$r(R) \subseteq (\mathrm{dom}(A_1) \times \mathrm{dom}(A_2) \times \ldots \times (\mathrm{dom}(A_n))$$

Cartesian product specifies all possible combinations of values from the underlying domains. Hence, This product of cardinalities of all domains represents the total number of possible instances or tuples that can ever exist in any relation state r(R)

$$|\mathrm{dom}(A_1)| \times |\mathrm{dom}(A_2)| \times \ldots \times |\mathrm{dom}(A_n)|$$

Set Theory for Relational Model

By defining relational model in terms of set we borrow set properties for defining databases

- Distinct Elements in set (No Duplicates):
    - Distinct Relations (tables) in a database
    - Distinct Attributes (columns) in Relation
    - Distinct Records (rows) in a Relation
- Sequence of elements in not important in set
    - Sequence of Relation not important
    - Sequence in Attribute not important
    - Sequence of Records not important

Properties of a Relation

- The relation has a name that is distinct from all other relation names in the relational schema
- Each cell of the relation contains exactly one atomic (single) value
- Each attribute has a distinct name
- The values of an attribute are all from the same domain
- Each tuple is distinct; there are no duplicate tuples
- The order of attributes has no significance
- The order of tuples has no significance

Relational Model Constraints

Domain Constraints

Within each tuple, the value of each attribute A must be an atomic value from the domain dom(A)

Domain constraints are specified using data types:

- integers (short integer, integer, and long integer)
- real numbers (float and double-precision float).
- Characters
- Booleans
- Strings (fixed-length strings, and variable-length strings)
- Date, Time
- Timestamp

Relational Model Constraints

Key Constraints (Uniqueness Constraint)

Do you remember ?

A relation is defined as a set of tuples : all tuples in a relation must also be distinct

Keys are used to uniquely identify a tuple in a relation

Minimality Property : A key is a minimal set of attributes that uniquely identifies a tuple in a relation.

Usually there is a set of attributes that uniquely identifies a tuple in a relation.

Such attributes will be collectively referred to as a **Superkey**

Note

Keys apply to the relational schema not to the relational instance.
That is, looking at the current data cannot tell you for sure if the set of attributes is a key.

Key Constraints continued

Candidate Key

One of the possible keys of a relation

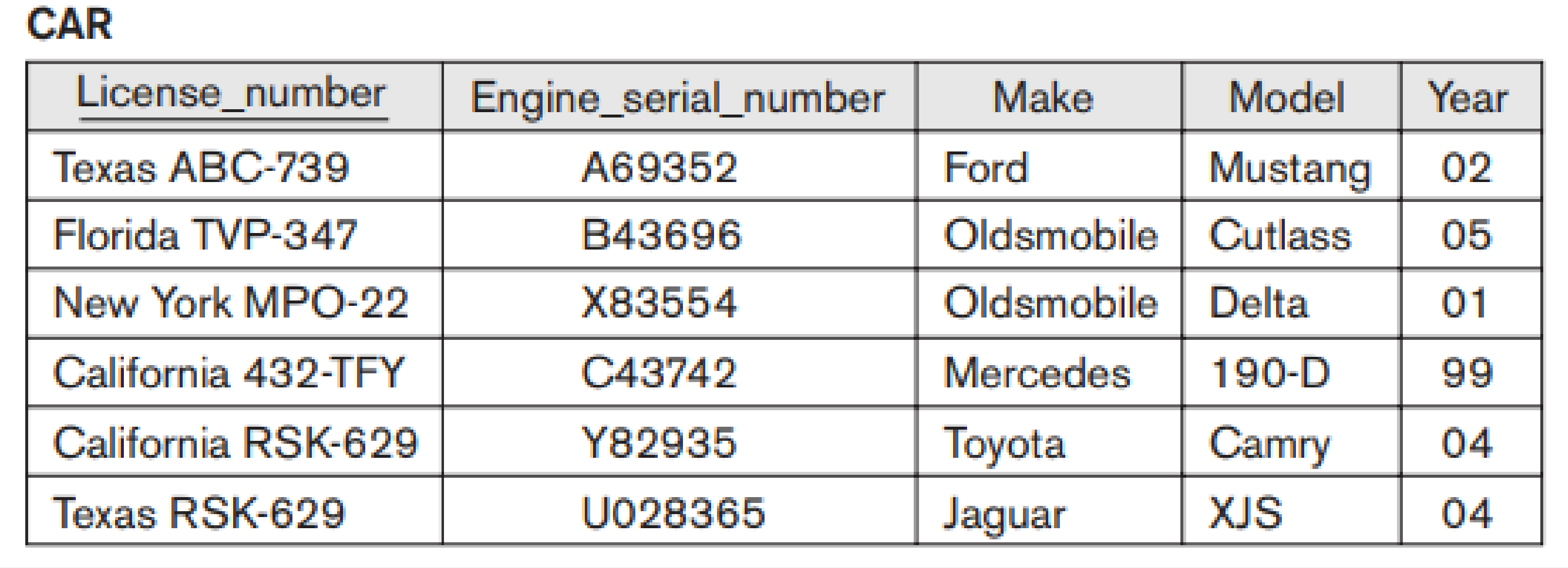
Primary Key

A candidate key designated as the distinguishing key of a relation

Foreign Key

A foreign key is a set of attributes in one relation referring to the primary key of another relation

Example



CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

QUESTIONS ???