Relational Model

CSC365
Spring 2019

Data Model

- Three important components of a data model
  - Structure
  - Manipulation
  - Constraints
- Higher in the levels of abstraction compared to “data structure”

Kinds of Data Models

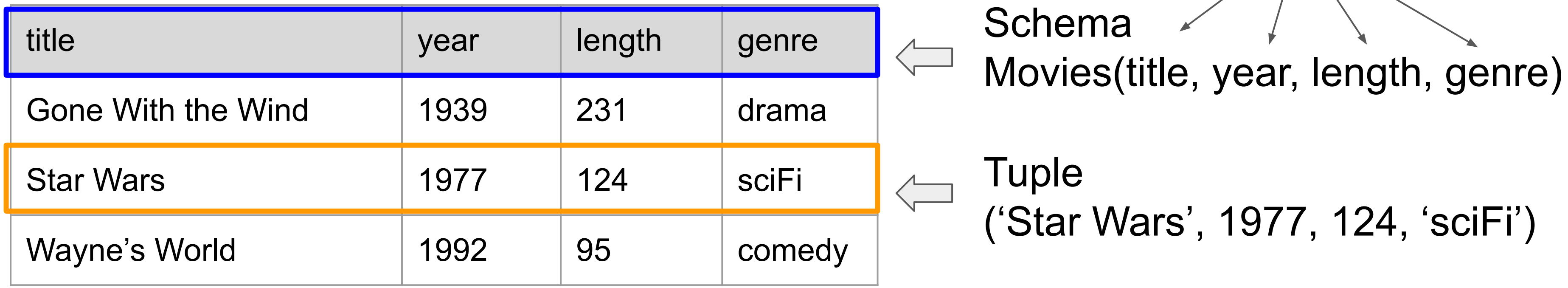
- The relational model
- The semistructured-data model

Relations

Relation - represented as a two-dimensional table

Movies

Attributes



title	year	length	genre
Gone With the Wind	1939	231	drama
Star Wars	1977	124	sciFi
Wayne’s World	1992	95	comedy

Schema
Movies(title, year, length, genre)

Tuple
(‘Star Wars’, 1977, 124, ‘sciFi’)

Domain

- Each component of each tuple is required to be atomic
  - The value can not be a structure such as set, list, and other data structures.
- Domains - elementary data types
  - Integer
  - String
  - Date
  - Null is a member of every domain

Schema
Movies(title:string, year:integer, length:integer, genre:string)

Equivalent Representations of a Relation

- Relations are sets of tuples, not lists of tuples

Movies

year	title	length	genre
1993	Gone With the Wind	231	drama
1977	Star Wars	124	sciFi
1992	Wayne’s World	95	comedy

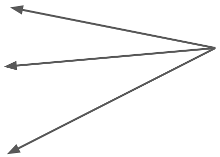
Movies(year, title, length, genre)

(1977, ‘Star Wars’, 124, ‘sciFi’)

Relation Instances

Movies

title	year	length	genre
Gone With the Wind	1939	231	drama
Star Wars	1977	124	sciFi
Wayne’s World	1992	95	comedy



instances

Keys

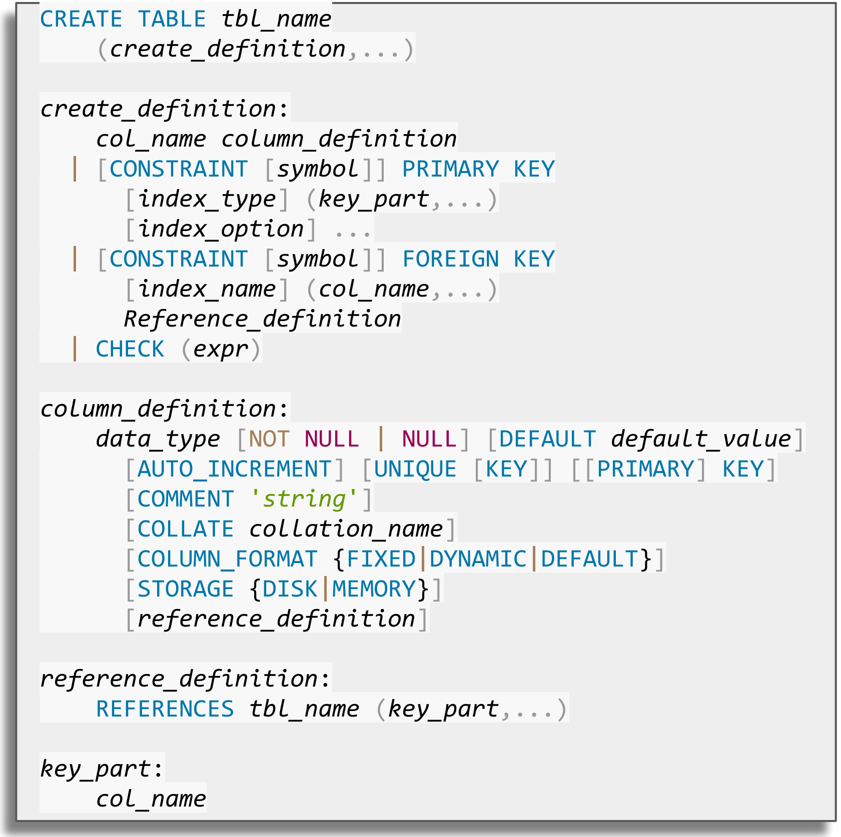
- Key Constraints
    - A set of attributes that no two tuples in a relation are allowed to have the same values
    - The set of attributes form a key
- Key
    - Uniquely identifies each row in a relation (must be minimal set)
    - Indicated by underlines
    - When there are more than one set of attributes that can be keys, we pick one as primary key.

Schema
Movies(title, year, length, imdb, genre)

Defining a Relation Schema in SQL

- Structured Query Language (SQL)
    - Originally called SEQUEL (Structured English Query Language)
    - Based on Relational Algebra and Relational Calculus
    - Standardized in 1986 by ANSI
        - SQL - 2016 : The latest updates to the standard
        - Each DBMS has its own variation (Dialect)
    - Incorporates sublanguages
        - Data Definition Language (DDL)
        - Data Manipulation Language (DML)
        - Data Query Language (DQL)

Defining a Relation Schema in SQL



```
CREATE TABLE tbl_name
    (create_definition,...)

create_definition:
    col_name column_definition
  | [CONSTRAINT [symbol]] PRIMARY KEY
      [index_type] (key_part,...)
      [index_option] ...
  | [CONSTRAINT [symbol]] FOREIGN KEY
      [index_name] (col_name,...)
      Reference_definition
  | CHECK (expr)

column_definition:
    data_type [NOT NULL | NULL] [DEFAULT default_value]
      [AUTO_INCREMENT] [UNIQUE [KEY]] [[PRIMARY] KEY]
      [COMMENT 'string']
      [COLLATE collation_name]
      [COLUMN_FORMAT {FIXED|DYNAMIC|DEFAULT}]
      [STORAGE {DISK|MEMORY}]
      [reference_definition]

reference_definition:
    REFERENCES tbl_name (key_part,...)

key_part:
    col_name
```

```
CREATE TABLE Movies (
  title VARCHAR(50) NOT NULL,
  year INTEGER(4) NOT NULL,
  length INTEGER NOT NULL DEFAULT 0,
  genre ENUM(‘drama’,’comedy’,’sciFi’),
  PRIMARY KEY (title, year)
);
```

Data Types

- INTEGER
- FLOAT
- CHAR
- VARCHAR
- DATE
- BOOLEAN
    - Use TINYINT in MySQL instead

Constraints

- PRIMARY KEY (col_name[, …])
  - Key Constraints
- FOREIGN KEY (col_name[, …]) REFERENCES tbl_name (col_name[, …])
  - Foreign key
  - Referential Integrity Constraints

Kind of Keys

- Super Key
  - Super set of a key
  - Not necessarily minimal set
- Candidate Key
  - Minimal set
  - Candidate for a key - can be a key
- Primary Key
  - Chosen among keys
- Foreign Key
  - Uniquely identifies each row in other table

Movies

title	year	length	imdb
Footloose	1984	110	6.5
The Hunger Games	2012	142	7.2
Pompeii	2014	105	5.5

title, year, length, imdb
title, year, length
title, year
title

Examples

Stars

name	gender	birthdate	agent_id
Kit Harington	male	12/26/86	101
Kevin Bacon	male	07/08/58	201
Jennifer Lawrence	female	08/15/90	333

StarsIn

star	title	year
Kevin Bacon	Footloose	1984
Jennifer Lawrence	The Hunger Games	2012
Kit Harington	Pompeii	2014

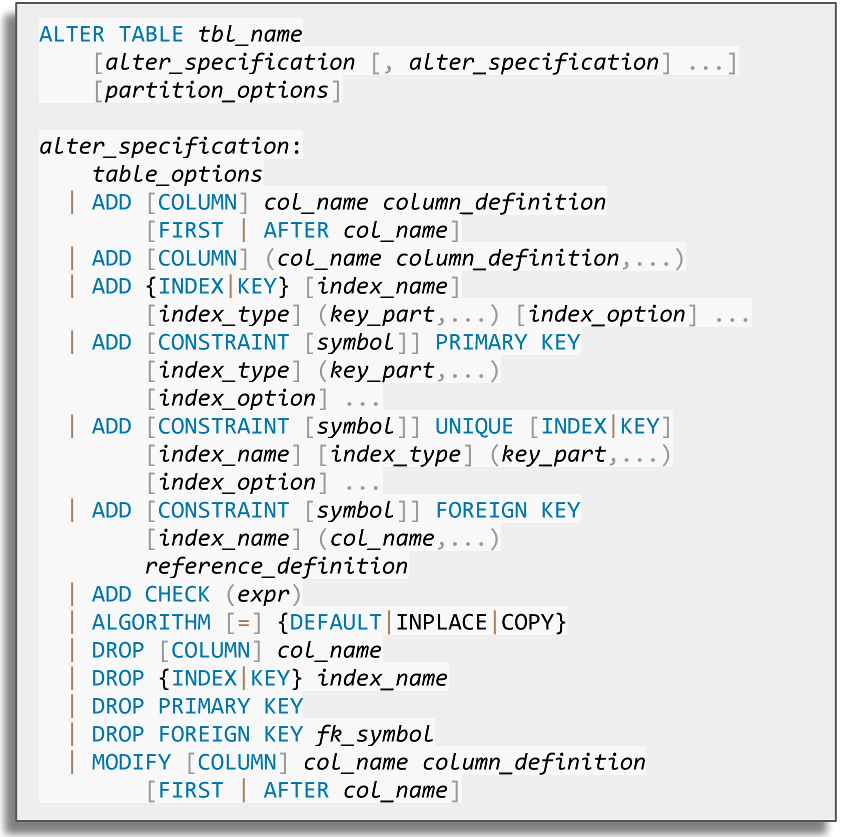
Agents

name	phone	agent_id
Star Agent A	1112223333	101
Star Agent B	2223334444	201
Star Agent C	4445556666	333

Movies

title	year	length	imdb
Footloose	1984	110	6.5
The Hunger Games	2012	142	7.2
Pompeii	2014	105	5.5

Altering a Relation Schema in SQL



```
ALTER TABLE tbl_name
    [alter_specification [, alter_specification] ...]
    [partition_options]

alter_specification:
    table_options
  | ADD [COLUMN] col_name column_definition
        [FIRST | AFTER col_name]
  | ADD [COLUMN] (col_name column_definition,...)
  | ADD {INDEX|KEY} [index_name]
        [index_type] (key_part,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]] PRIMARY KEY
        [index_type] (key_part,...)
        [index_option] ...
  | ADD [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY]
        [index_name] [index_type] (key_part,...)
        [index_option] ...
  | ADD [CONSTRAINT [symbol]] FOREIGN KEY
        [index_name] (col_name,...)
       reference_definition
  | ADD CHECK (expr)
  | ALGORITHM [=] {DEFAULT|INPLACE|COPY}
  | DROP [COLUMN] col_name
  | DROP {INDEX|KEY} index_name
  | DROP PRIMARY KEY
  | DROP FOREIGN KEY fk_symbol
  | MODIFY [COLUMN] col_name column_definition
        [FIRST | AFTER col_name]
```

ALTER TABLE Movies ADD imdb FLOAT;
ALTER TABLE Movies MODIFY genre ENUM('drama', 'comedy', 'sciFi', 'action');

Deleting a Relation in SQL

```
DROP [TEMPORARY] TABLE [IF EXISTS]
    tbl_name [, tbl_name] ...
    [RESTRICT | CASCADE]
```

```
DROP TABLE Movies;
```

DML

Data Manipulation Language

- Create
  - INSERT
- Update
  - UPDATE
- Delete
  - DELETE

INSERT

- Standard
  - INSERT INTO tbl_name (col_name, ...) VALUES (value, ...);
  - INSERT INTO Movies (title, year, length, imdb) VALUES (‘Skyfall’, 2012, 143, 7.8);
- MySQL dialect
  - INSERT INTO tbl_name SET col_name=value, …;
  - INSERT INTO Movies SET title=’Skyfall’, year=2012, length=143, imdb=7.8;

UPDATE

```
UPDATE [LOW_PRIORITY] [IGNORE]
table_reference
    SET assignment_list
    [WHERE where_condition]
    [ORDER BY ...]
    [LIMIT row_count]

value:
    {expr | DEFAULT}

assignment:
    col_name = value

assignment_list:
    assignment [, assignment] ...
```

```
UPDATE Movies SET imdb=8.6
  WHERE title='Star Wars'
  and year=1977;
```

DELETE

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
FROM tbl_name
    [PARTITION (partition_name [,
partition_name] ...)]
    [WHERE where_condition]
    [ORDER BY ...]
    [LIMIT row_count]
```

```
DELETE FROM Movies WHERE
  title='Star Wars' and year=1977;
```

Data Query Language (DQL)

READ (SELECT)

```
SELECT
    [ALL | DISTINCT | DISTINCTROW ]
   select_expr [, select_expr ...]
    [FROM table_references
    [WHERE where_condition]
    [GROUP BY {col_name | expr | position}
      [ASC | DESC], ... [WITH ROLLUP]]
    [HAVING where_condition]
    [ORDER BY {col_name | expr | position}
      [ASC | DESC], ...]
    [LIMIT {[offset,] row_count | row_count
OFFSET offset}]
```

```
SELECT * FROM Movies
  WHERE length >= 120;
```