Week 2 Tutoring

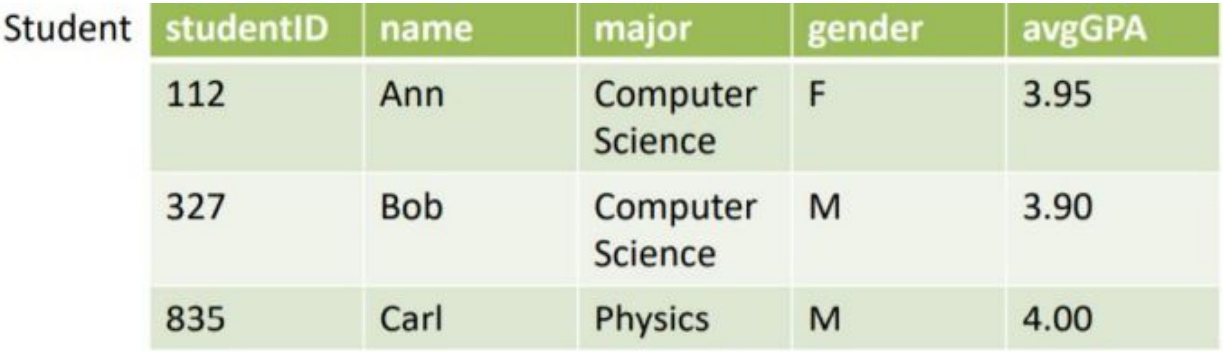
CSE 180

Relational Database Schema R

- A set of relation schemas
- A framework of a table

Instance of a Relational Database Schema R is one of the relations in the database schema

Eg. Student(studentID, name, major, gender, avgGPA)



Student

studentID	name	major	gender	avgGPA
112	Ann	Computer Science	F	3.95
327	Bob	Computer Science	M	3.90
835	Carl	Physics	M	4.00

Keys!

Superkey S: a subset of attributes of relation schema R, such that there can’t be any two different tuples in an instance of R that have the same superkey

Key K: a set of attributes that consists of the least number of attributes that satisfy Superkey’s principle mentioned on the previous page

Every key is a superkey, but not every superkey is a key

Superkey Example

Take the relation schema as an example

Student(studentID, name, major, gender, avgGPA)

Given a subset of attributes of the relation **{studentID, gender, avgGPA}**

We **cannot** find two different student/tuple (since **one student** can only have **one unique student ID**)

So the subset of attributes of Relation R is a Superkey.

Key Example

Student(studentID, name, major, gender, avgGPA)

Given a subset of attributes of the relation {*studentID*, gender, avgGPA}

We can **remove two attributes, {gender, avgGPA}**, from the set

We **can still find the same student/tuple** we found on the last page, **so {studentID, gender, avgGPA} is not a key.**

Data Types

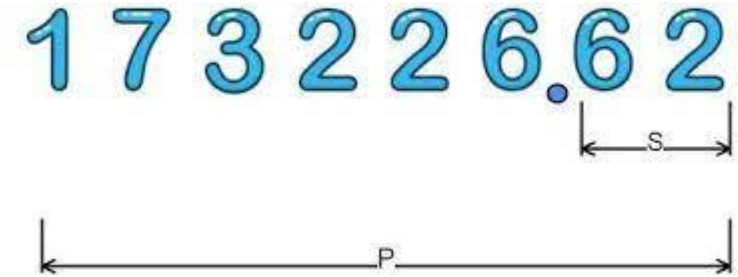
Primitive Data Types:

- CHAR(n), VARCHAR(n)
  - Char is **fixed** length; use when data entry size is constant
  - Varchar is **variable** length; use when data entry size varies
- BIT(n), BIT VARYING(n), BOOLEAN, INTEGER,
- SHORTINT, DECIMAL/NUMERIC(n, d), FLOAT(p), DOUBLE, PRECISION,
- DATE, TIME, TIMESTAMP, INTERVAL – Separate data types – Constants are character strings of a specific form, e.g., DATE ‘2017-09-13’ and TIME ‘16:45:33’

Decimal/Numeric Data Type

Decimal(p, s) aka Decimal(n,d)

Decimal(7, 2) => 25050.51



Interval Data Type

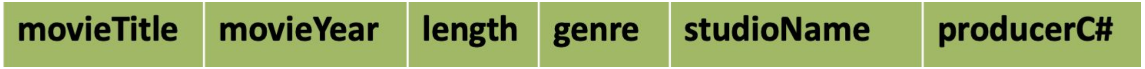
In lab 1, subscriptionInterval and readInterval should be of type interval

The INTERVAL data type stores a value that represents a span of time. INTERVAL types are divided into two classes: *year-month intervals* and *day-time intervals*.

- You do not need to specify this in your table declaration.

CREATE TABLE

```
CREATE TABLE Movies (
        movieTitle CHAR(100),
        movieYear INT, length INT,
        genre BOOLEAN,
        studioName CHAR(30),
        producerC# INT
);
```



movieTitle	movieYear	length	genre	studioName	producerC#

Foreign Key Constraint

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

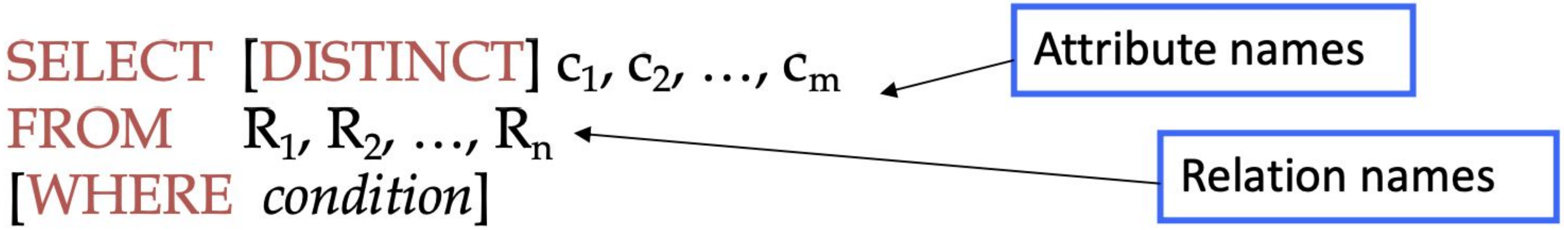
Eg: An election in Elections specifies the ID of the office for which the election is held, the date of the election, andthe start and end date for the term of office for the winner of the election.

- Any officeID that's in an Elections row must appear as an officeID in the ElectedOffices table.

This means that Elections is a child table of ElectedOffices

```
CREATE TABLE Elections (
        …
        FOREIGN KEY (officeID) REFERENCES ElectedOffices(officeID)
);
```

Simple SQL Query



SELECT [DISTINCT] $c_1, c_2, \ldots, c_m$
FROM $R_1, R_2, \ldots, R_n$
[WHERE *condition*]

Aliasing: Allows you to rename the attributes of the result

It will be important to know when to use DISTINCT and when it is not necessary for your queries!

Pattern Matching

LIKE keyword returns TRUE if the string is in the attribute

- '%' (stands for 0 or more arbitrary chars)
- '_' (stands for exactly one arbitrary char)
- WHERE movieTitle LIKE ‘Juras_ic%'
  - This returns TRUE if the string contains “Juras” + any singular atribrary character + “ic” + anything else or nothing
- Pattern matching for special characters like % and _ you can use Escape
  - Look at Lecture 3, slide 60 for more details and methods