Uni DB

ER Model

Relational Model

- intstructor(ID, name, dept_name → department, salary)
- **course**(id, title, dept_name → department, credits)
- **prereq**(course_id → course, prereq_id → course)
- **department**(name, building, budget)
- **section**(course_id, id, semester, year, (building, room_number) → classroom, time_slot_id)
- **teaches**(instructor_ID → instructor, (course_id, sec_id, semester, year) → section)
- **student**(ID, name, dept_name → department, total_credit)
- **takes**(student_ID → student, (course_id, section_id, semester, year) → section, grade)
- **advisor**(student_id → student, instructor_id → instructor)
- **classroom**(building, room_number, capacity)
- **time_slot**(id, day, start_time, end_time)

SQL

DDL

- **Definitions**:

```
drop table prereq;
drop table time_slot;
drop table advisor;
drop table takes;
drop table student;
drop table teaches;
drop table section;
drop table instructor;
drop table course;
drop table department;
drop table classroom;


create table classroom
    (building       varchar(15),
     room_number        varchar(7),
```

```
     capacity       numeric(4,0),
     primary key (building, room_number)
    );

create table department
    (name       varchar(20),
     building       varchar(15),
     budget             numeric(12,2) check (budget > 0),
     primary key (name)
    );

create table course
    (id     varchar(8),
     title          varchar(50),
     dept_name      varchar(20),
     credits        numeric(2,0) check (credits > 0),
     primary key (id),
     foreign key (dept_name) references department (name)
        on delete set null
    );

create table instructor
    (ID         varchar(5),
     name           varchar(20) not null,
     dept_name      varchar(20),
     salary         numeric(8,2) check (salary > 29000),
     primary key (ID),
     foreign key (dept_name) references department (name)
        on delete set null
    );

create table section
    (course_id      varchar(8),
         id         varchar(8),
     semester       varchar(6)
        check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
     year           numeric(4,0) check (year > 1701 and year < 2100),
     building       varchar(15),
     room_number        varchar(7),
     time_slot_id       varchar(4),
     primary key (course_id, id, semester, year),
```

```
     foreign key (course_id) references course (id)
        on delete cascade,
     foreign key (building, room_number) references classroom (building,
      ↪  room_number)
        on delete set null
    );

create table teaches
    (instructor_ID          varchar(5),
     course_id      varchar(8),
     sec_id         varchar(8),
     semester       varchar(6),
     year           numeric(4,0),
     primary key (instructor_ID, course_id, sec_id, semester, year),
     foreign key (course_id, sec_id, semester, year) references section
      ↪  (course_id, id, semester, year)
        on delete cascade,
     foreign key (instructor_ID) references instructor (ID)
        on delete cascade
    );

create table student
    (ID         varchar(5),
     name           varchar(20) not null,
     dept_name      varchar(20),
     tot_cred       numeric(3,0) check (tot_cred >= 0),
     primary key (ID),
     foreign key (dept_name) references department (name)
        on delete set null
    );

create table takes
    (student_ID         varchar(5),
     course_id      varchar(8),
     sec_id         varchar(8),
     semester       varchar(6),
     year           numeric(4,0),
     grade              varchar(2),
     primary key (student_ID, course_id, sec_id, semester, year),
     foreign key (course_id, sec_id, semester, year) references section
      ↪  (course_id, id, semester, year)
```

```
        on delete cascade,
     foreign key (student_ID) references student (ID)
        on delete cascade
    );

create table advisor
    (student_ID         varchar(5),
     instructor_ID          varchar(5),
     primary key (student_ID),
     foreign key (instructor_id) references instructor (ID)
        on delete set null,
     foreign key (student_ID) references student (ID)
        on delete cascade
    );

create table time_slot
    (id     varchar(4),
     day            varchar(1),
     start_hr       numeric(2) check (start_hr >= 0 and start_hr < 24),
     start_min      numeric(2) check (start_min >= 0 and start_min <
 ↪  60),
     end_hr         numeric(2) check (end_hr >= 0 and end_hr < 24),
     end_min        numeric(2) check (end_min >= 0 and end_min < 60),
     primary key (id, day, start_hr, start_min)
    );

create table prereq
    (course_id      varchar(8),
     prereq_id      varchar(8),
     primary key (course_id, prereq_id),
     foreign key (course_id) references course (id)
        on delete cascade,
     foreign key (prereq_id) references course (id)
    );
```

- **Data**:

```
delete from prereq;
delete from time_slot;
delete from advisor;
delete from takes;
```

```
delete from student;
delete from teaches;
delete from section;
delete from instructor;
delete from course;
delete from department;
delete from classroom;
insert into classroom values ('Packard', '101', '500');
...
insert into department values ('Biology', 'Watson', '90000');
...
insert into course values ('BIO-101', 'Intro. to Biology', 'Biology',
 ↪  '4');
...
insert into instructor values ('10101', 'Srinivasan', 'Comp. Sci.',
 ↪  '65000');
...
insert into section values ('BIO-101', '1', 'Summer', '2017', 'Painter',
 ↪  '514', 'B');
...
insert into teaches values ('10101', 'CS-101', '1', 'Fall', '2017');
...
insert into student values ('00128', 'Zhang', 'Comp. Sci.', '102');
...
insert into takes values ('00128', 'CS-101', '1', 'Fall', '2017', 'A');
...
insert into advisor values ('00128', '45565');
...
insert into time_slot values ('A', 'M', '8', '0', '8', '50');
...
insert into prereq values ('BIO-301', 'BIO-101');
...
```

Example Querries