Basic Query Structure

- A typical SQL query has the form:

select $A_1, A_2, \ldots, A_n$
from $r_1, r_2, \ldots, r_m$
where P

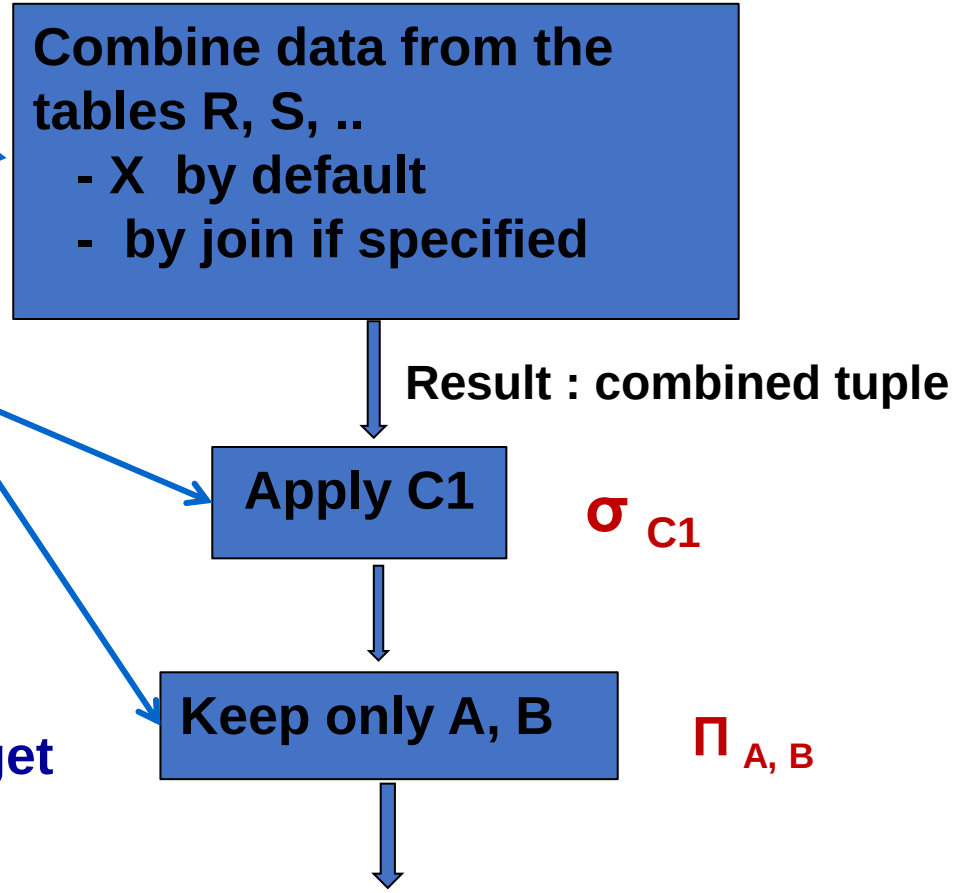
- The result of an SQL query is a relation
  - Made of columns selected in the query
  - May be one value, one tuple, or a set of tuples
  - May be empty too

Query interpretation

select A, B, …

from R, S, ..

where c1



Can test values in various columns
-- various comparison operations on data of different types
-- can NEST other SQL queries to get values for comparing

Can be computed results also

Duplicate handling in SQL

Nesting allowed in all clauses

Queries for University Database

Find students from CS with total credits > 100

```
select ID
from student
where dept_name = "CS" and tot_cred > 100
```

Use dot-notation for unique reference to attributes when using multiple tables

Find building of departments where Instructor Satish works

```
select building
from instructor, department
where instructor.dept_name = department.dept_name
    and name = "Satish"
```

What if there are two or more Satish ?!

The select Clause

- select * ….
- select distinct … (remove duplicates)
- select all … (keeps duplicates)
- select expression1, exp2, …

The where Clause

- Conditions using usual comparison operations : <, >, =, <>, …
- logical connectives and, or, and not

The from Clause

- Corresponds to the Cartesian product operation

  select *
  from *instructor, teaches*

- generates every possible instructor – teaches pair, with all attributes from both relations
- Cartesian product not very useful directly, but useful when combined with where-clause condition (== joins)

Joins

select *name, course_id*
from *instructor, teaches*
where *instructor.ID = teaches.ID*
and year = 2020

What is the query result ?

Joins.

Find the course ID, semester, year and title of each course offered by the Comp. Sci. department

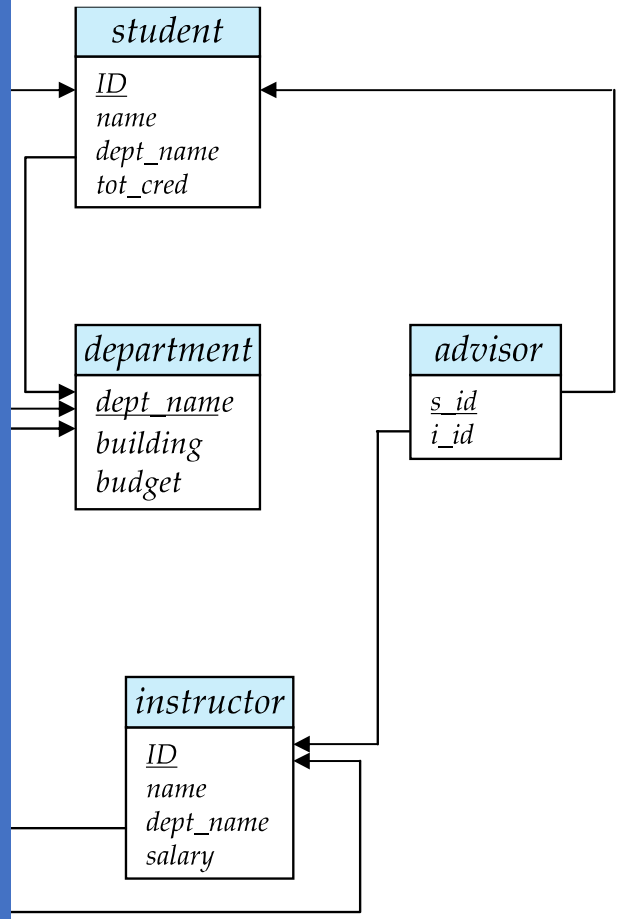
```
select section.course_id, semester, year, title
  from section, course
  where   section.course_id = course.course_id
                and dept_name = 'Comp. Sci.'
```

Queries for University Database

Find names of students advised by Sudarshan

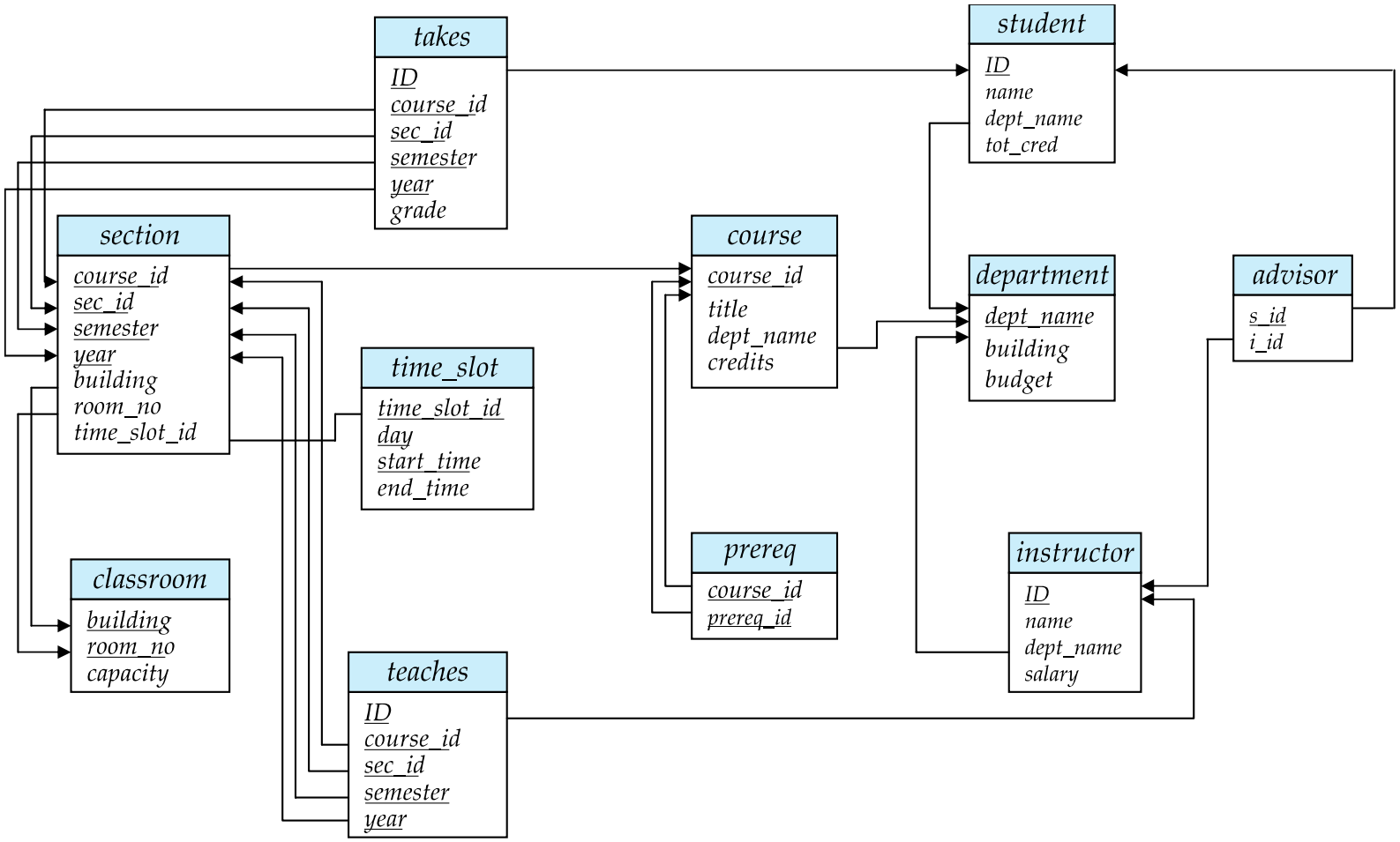
- What is output ?
- Which tables have required data
- How do you link data across the tables ? Use join

```
select student.ID
from student, advisor, instructor
where instructor.name = "Sudarshan"
    and instructor.ID = i_id
    and student.ID = s_id
```



Try Writing these Queries in SQL

1. Get course-ids of courses running in room SIC-201 of building called Rekhi building.
2. Get the course names also in the above query



The University database for online querying :
https://www.db-book.com/db7/university-lab-dir/sqljs.html

https://www.db-book.com/db7/university-lab-dir/sqljs.html

Search

Online SQL interpreter

Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Note: page may take a few seconds to load initially.). All query processing is done right in your browser using the SQLite database. Save the database and load it later, if you want your data to persist when you close the browser tab.
Click here for tips on using SQLite including SQL syntax variations.

Enter SQL commands here

```
1 -- enter your commands here
```

Execute | Save the db | Load an SQLite database file: Browse... No file selected.

name	sql
classroom	CREATE TABLE classroom (building varchar(15), room_number varchar(7), capacity numeric(4,0), primary key (building, room_number))
department	CREATE TABLE department (dept_name varchar(20), building varchar(15), budget numeric(12,2) check (budget > 0), primary key (dept_name))

Simple query demo

Online SQL interpreter

Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Note: page may take a few seconds to load initially.). All query processing is done right in your browser using the SQLite database. Save the database and load it later, if you want your data to persist when you close the browser tab.
Click here for tips on using SQLite including SQL syntax variations.

Enter SQL commands here

```
1 select * from department
```

Execute | Save the db | Load an SQLite database file: Browse... No file selected.

dept_name	building	budget
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Original work by kripken (sql.js). C to Javascript compiler by kripken (emscripten). Project now maintained by lovasoa

Natural Join

- Natural join matches tuples with the same values for all common attributes, and retains only one copy of each common column

```
select *
from instructor natural join teaches;
```

- Same as

```
select *
 from instructor, teaches
 where instructor.ID = teaches.ID;
```

- Warning : beware of unrelated attributes with same name which get equated incorrectly

Join using …

- Give condition for joining in where itself
- List names of instructors along with the titles of courses they teach

  - Correct version
    - **select** *name*, *title*
      from *instructor* **natural join** *teaches*, *course*
      where *teaches.course_id* = *course.course_id*;
  - Another correct version
    - **select** *name*, *title*
      from (*instructor* **natural join** *teaches*)
      join *course* **using**(*course_id*);

The Rename Operation

- SQL allows renaming relations and attributes using the as clause:
  old-name as *new-name*
- Simplifies query writing and in resolving ambiguity
- Find names and monthly salaries of all instructors who have a higher salary than some instructor in 'Comp. Sci'.

```
select distinct T.name, T.salary/12 as monthly_salary
from instructor as T, instructor as S
  where T.salary > S.salary and
        S.dept_name = 'Comp. Sci.'
```

- Keyword as is optional and may be omitted
  instructor as *T* ≡ *instructor T*