# Introduction to SQL

- Data Definition Language (DDL)
    - Create table
    - Drop table
- Data Manipulation Language (DML)
    - Select
    - Insert
    - Delete
    - Update
- Other

# Basic Select, where Statement

1.

```
select S1.sID, S1.sName, S1.GPA, S2.sID, S2.sName, S2.GPA
from Student S1, Student S2
where S1.GPA = S2.GPA and S1.sID < S2.sID;
```

```
select cName as name from College
union
select sName as name from Student;
```

Sorted;
No duplicates;

| | name |
|---|---|
| ▶1 | Amy |
| 2 | Berkeley |
| 3 | Bob |
| 4 | Cornell |
| 5 | Craig |
| 6 | Doris |
| 7 | Edward |
| 8 | Fay |
| 9 | Gary |
| 10 | Helen |
| 11 | Irene |

```
select cName as name from College
union all
select sName as name from Student;
```

Not sorted;
Have duplicates;

sorted( order by name)

| | name |
|---|---|
| ▶1 | Stanford |
| 2 | Berkeley |
| 3 | MIT |
| 4 | Cornell |
| 5 | Amy |
| 6 | Bob |
| 7 | Craig |
| 8 | Doris |
| 9 | Edward |
| 10 | Fay |
| 11 | Gary |
| 12 | Helen |

```
select sID from Apply where major = 'CS'
intersect
select sID from Apply where major = 'EE';
```

| | sID |
|---|---|
| 1 | 123 |
| 2 | 345 |

```
select distinct A1.sID
from Apply A1, Apply A2
where A1.sID = A2.sID and A1.major = 'CS' and A2.major = 'EE';
```

| | sID |
|---|---|
| 1 | 123 |
| 2 | 345 |

```
select sID from Apply where major = 'CS'
except
select sID from Apply where major = 'EE';
```

students who apply to CS but not EE

Another way:

```
select sID, sName
from Student
where sID in (select sID from Apply where major = 'CS')
  and sID not in (select sID from Apply where major = 'EE');
```

Another way:

```
select sID, sName
from Student
where sID = any (select sID from Apply where major = 'CS')
  and not sID = any (select sID from Apply where major = 'EE');
```

| | sID |
|---|---|
| 1 | 543 |
| 2 | 876 |
| 3 | 987 |

```
select distinct A1.sID
from Apply A1, Apply A2
where A1.sID = A2.sID and A1.major = 'CS' and A2.major <> 'EE';
```

Different result from last one:
Apply to cs and cs

| | sID |
|---|---|
| 1 | 123 |
| 2 | 345 |
| 3 | 543 |
| 4 | 876 |
| 5 | 987 |

```
select sID, sName
from Student
where sID in (select sID from Apply where major = 'CS');
```

No duplicates

| | sID | sName |
|---|---|---|
| 1 | 123 | Amy |
| 2 | 345 | Craig |
| 3 | 987 | Helen |
| 4 | 876 | Irene |
| 5 | 543 | Craig |

```
select Student.sID, sName
from Student, Apply
where Student.sID = Apply.sID and major = 'CS';
```

Have duplicates
Like Amy applied to 2 schools with CS major in “Apply”

| | sID | sName |
|---|---|---|
| 1 | 123 | Amy |
| 2 | 123 | Amy |
| 3 | 345 | Craig |
| 4 | 987 | Helen |
| 5 | 987 | Helen |
| 6 | 876 | Irene |
| 7 | 543 | Craig |

```
select cName, state
from College C1
where exists (select * from College C2
              where C2.state = C1.state);
```

should add “C1.cName <> C2.cName”

| | cName | state |
|---|---|---|
| 1 | Stanford | CA |
| 2 | Berkeley | CA |
| 3 | MIT | MA |
| 4 | Cornell | NY |

```
select cName
from College C1
where not exists (select * from College C2
                  where C2.enrollment > C1.enrollment);
```

college with highest enrollment

| | cName |
|---|---|
| 1 | Berkeley |

```
select cName
from College S1
where enrollment > all (select enrollment from College S2
                        where S2.cName <> S1.cName);
```

another way: ”where not enrollment <= any ()”

| | cName |
|---|---|
| 1 | Berkeley |

```
select sName, GPA
from Student
where GPA >= all (select GPA from Student);
```

student with the highest GPA

| | sName | GPA |
|---|---|---|
| 1 | Amy | 3.9 |
| 2 | Doris | 3.9 |
| 3 | Irene | 3.9 |
| 4 | Amy | 3.9 |

```
select sID, sName, sizeHS
from Student
where sizeHS > any (select sizeHS from Student);
```

students not from the smallest HS

Another way:

```
select sID, sName, sizeHS
from Student S1
where exists (select * from Student S2
              where S2.sizeHS < S1.sizeHS);
```

| | sID | sName | sizeHS |
|---|---|---|---|
| 1 | 123 | Amy | 1000 |
| 2 | 234 | Bob | 1500 |
| 3 | 345 | Craig | 500 |
| 4 | 456 | Doris | 1000 |
| 5 | 567 | Edward | 2000 |
| 6 | 789 | Gary | 800 |
| 7 | 987 | Helen | 800 |
| 8 | 876 | Irene | 400 |
| 9 | 765 | Jay | 1500 |

# Subqueries in FORM & SELECT

```
select sID, sName, GPA, GPA*(sizeHS/1000.0) as scaledGPA
from Student
where GPA*(sizeHS/1000.0) - GPA > 1.0
or GPA - GPA*(sizeHS/1000.0) > 1.0;
```

another way:

```
select *
from (select sID, sName, GPA, GPA*(sizeHS/1000.0) as scaledGPA
from Student) G
where abs(G.scaledGPA - GPA) > 1.0;
```

| | sID | sName | GPA | scaledGPA |
|---|---|---|---|---|
| ▶1 | 234 | Bob | 3.6 | 5.4 |
| 2 | 345 | Craig | 3.5 | 1.75 |
| 3 | 567 | Edward | 2.9 | 5.8 |
| 4 | 678 | Fay | 3.8 | 0.76 |
| 5 | 876 | Irene | 3.9 | 1.56 |
| 6 | 765 | Jay | 2.9 | 4.35 |
| 7 | 543 | Craig | 3.4 | 6.8 |

```
select distinct College.cName, state, GPA
from College, Apply, Student
where College.cName = Apply.cName
  and Apply.sID = Student.sID
  and GPA >= all
          (select GPA from Student, Apply
           where Student.sID = Apply.sID
             and Apply.cName = College.cName);
```

college paired with the highest GPA of their applicants

Another way:

```
select cName, state,
(select distinct GPA
from Apply, Student
where College.cName = Apply.cName
  and Apply.sID = Student.sID
  and GPA >= all
          (select GPA from Student, Apply
           where Student.sID = Apply.sID
             and Apply.cName = College.cName)) as GPA
from College;
```

| | cName | state | GPA |
|---|---|---|---|
| ▶1 | Stanford | CA | 3.9 |
| 2 | Berkeley | CA | 3.9 |
| 3 | Cornell | NY | 3.9 |
| 4 | MIT | MA | 3.9 |

```
select cName, state,
(select distinct sName
from Apply, Student
where College.cName = Apply.cName
  and Apply.sID = Student.sID) as sName
from College;
```

| Description |
| --- |
| Subquery returns more than 1 row |

Since more than one student apply to one college

## The JOIN family of operators

```
select distinct sName, major
from Student, Apply
where Student.sID = Apply.sID;
```

Above is the nature join; below is inner join

```
select distinct sName, major
from Student inner join Apply
on Student.sID = Apply.sID;
```

| | sname | major |
| --- | --- | --- |
| 1 | Helen | CS |
| 2 | Jay | history |
| 3 | Jay | psychology |
| 4 | Irene | CS |
| 5 | Craig | bioengineering |
| 6 | Irene | marine biology |
| 7 | Amy | EE |

```
select sName, GPA
from Student, Apply
where Student.sID = Apply.sID
  and sizeHS < 1000 and major = 'CS' and cName = 'Stanford';
```

students who apply to cs and from small size HS

PS: inner join is the default

```
select sName, GPA
from Student join Apply
on Student.sID = Apply.sID
where sizeHS < 1000 and major = 'CS' and cName = 'Stanford';
```

Another way: change “where” to “and”;

| | sname | gpa |
| --- | --- | --- |
| 1 | Helen | 3.7 |
| 2 | Irene | 3.9 |

Another way: natural join:

```
select sName, GPA
from Student natural join Apply
where sizeHS < 1000 and major = 'CS' and cName = 'Stanford';
```

Another way: “using”:

```
select sName, GPA
from Student join Apply using(sID)
where sizeHS < 1000 and major = 'CS' and cName = 'Stanford';
```

```
select Apply.sID, sName, GPA, Apply.cName, enrollment
from Apply, Student, College
where Apply.sID = Student.sID and Apply.cName = College.cName;
```

another way:

```
select Apply.sID, sName, GPA, Apply.cName, enrollment
from (Apply join Student on Apply.sID = Student.sID) join College
on Apply.cName = College.cName;
```

| | sid | sname | gpa | cname | enrollment |
|---|---|---|---|---|---|
| 1 | 123 | Amy | 3.9 | Stanford | 15000 |
| 2 | 123 | Amy | 3.9 | Cornell | 21000 |
| 3 | 123 | Amy | 3.9 | Berkeley | 36000 |
| 4 | 123 | Amy | 3.9 | Stanford | 15000 |
| 5 | 234 | Bob | 3.6 | Berkeley | 36000 |
| 6 | 345 | Craig | 3.5 | MIT | 10000 |
| 7 | 345 | Craig | 3.5 | Cornell | 21000 |

```
select distinct sName, major
from Student inner join Apply
on Student.sID = Apply.sID;
```

above is inner join & below is natural join

```
select distinct sName, major
from Student natural join Apply;
```

| | sname | major |
|---|---|---|
| 1 | Helen | CS |
| 2 | Jay | history |
| 3 | Jay | psychology |
| 4 | Irene | CS |
| 5 | Craig | bioengineering |
| 6 | Irene | marine biology |

Pair of students get the same GPA:

```
select S1.sID, S1.sName, S1.GPA, S2.sID, S2.sName, S2.GPA
from Student S1, Student S2
where S1.GPA = S2.GPA and S1.sID < S2.sID;
```

another way by “join”:

```
select S1.sID, S1.sName, S1.GPA, S2.sID, S2.sName, S2.GPA
from Student S1 join Student S2 using(GPA)
where S1.sID < S2.sID;
```

PS: cannot use “using” and “on” on the same time

| | sid | sname | gpa | sid1 | sname1 | gpa1 |
|---|---|---|---|---|---|---|
| 1 | 567 | Edward | 2.9 | 765 | Jay | 2.9 |
| 2 | 543 | Craig | 3.4 | 789 | Gary | 3.4 |
| 3 | 456 | Doris | 3.9 | 876 | Irene | 3.9 |
| 4 | 456 | Doris | 3.9 | 654 | Amy | 3.9 |
| 5 | 123 | Amy | 3.9 | 456 | Doris | 3.9 |
| 6 | 123 | Amy | 3.9 | 876 | Irene | 3.9 |
| 7 | 123 | Amy | 3.9 | 654 | Amy | 3.9 |
| 8 | 654 | Amy | 3.9 | 876 | Irene | 3.9 |

```
select *
from Student S1 natural join Student S2;
```

Same as:

```
select *
from Student;
```

Any relation joined with itself gives you back the original relation.

```
select sName, sID, cName, major
from Student left outer join Apply using(sID);
```

It takes any tuples on the left side of the join, if they don’t have a matching tuple from the right(aka.dangling), it still

| | | | | |
|---|---|---|---|---|
| 9 | Craig | 345 | MIT | bioengineer |
| 10 | Doris | 456 | | |
| 11 | Edward | 567 | | |
| 12 | Fay | 678 | Stanford | history |
| 13 | Gary | 789 | | |
| 14 | Helen | 987 | Berkeley | CS |

added to the result and it's padded with null values.
PS: left outer = left
Another way to replace "using" by natural:

```
select sName, sID, cName, major
from Student natural left outer join Apply;
```

Another way to do without join:

```
select sName, Student.sID, cName, major
from Student, Apply
where Student.sID = Apply.sID
union
select sName, sID, NULL, NULL
from Student
where sID not in (select sID from Apply);
```

```
select sName, sID, cName, major
from Student left outer join Apply using(sID)
union
select sName, sID, cName, major
from Student right outer join Apply using(sID);
```

Another way by "full outer join":

```
select sName, sID, cName, major
from Student full outer join Apply using(sID);
```

Another way without "join":

```
select sName, Student.sID, cName, major
from Student, Apply
where Student.sID = Apply.sID
union
select sName, sID, NULL, NULL
from Student
where sID not in (select sID from Apply)
union
select NULL, sID, cName, major
from Apply
where sID not in (select sID from Student);
```

| | sname | sid | cname | major |
|---|---|---|---|---|
| 1 | Craig | 345 | MIT | bioenginee |
| 2 | Amy | 123 | Cornell | EE |
| 3 | Gary | 789 | | |
| 4 | Bob | 234 | Berkeley | biology |
| 5 | Irene | 876 | MIT | biology |
| 6 | Doris | 456 | | |
| 7 | | 321 | MIT | psychology |

T1: (a,b) = (1,2)
T2: (b,c) = (2,3)
T3: (a,c) = (4,5)

Out join is not associated:

```
select A,B,C
from (T1 natural full outer join T2) natural full outer join T3;
```

| | a | b | c |
|---|---|---|---|
| 1 | 1 | 2 | 3 |
| 2 | 4 | | 5 |

```
select A,B,C
from T1 natural full outer join (T2 natural full outer join T3);
```

| | a | b | c |
|---|---|---|---|
| 1 | 4 | | 5 |
| 2 | | 2 | 3 |
| 3 | 1 | 2 | |

## Aggregation


```
select avg(GPA)
from Student;
```

| avg(GPA) |
|---|
| 3.566666666666... |

Lowest GPA:

```
select min(GPA)
from Student, Apply
where Student.sID = Apply.sID and major = 'CS';
```

| min(GPA) |
|---|
| 3.4 |

```
select avg(GPA)
from Student
where sID in (select sID from Apply where major = 'CS');
```

Avg GPA of students who apply to CS

```
select count(*)
from College
where enrollment > 15000;
```

Number of colleges which enrollment is larger than 15k;

```
select count(distinct sID)
from Apply
where cName = 'Cornell';
```

Number of students applying to Cornell

```
select *
from Student S1
where (select count(*) from Student S2
       where S2.sID <> S1.sID and S2.GPA = S1.GPA) =
      (select count(*) from Student S2
       where S2.sID <> S1.sID and S2.sizeHS = S1.sizeHS);
```

Students s.t. number of other students with the same GPA is equal to number of other students with the same sizeHS

```
select CS.avgGPA - NonCS.avgGPA
from (select avg(GPA) as avgGPA from Student
      where sID in (
         select sID from Apply where major = 'CS')) as CS,
     (select avg(GPA) as avgGPA from Student
      where sID not in (
         select sID from Apply where major = 'CS')) as nonCS;
```

another way:

```
select distinct (select avg(GPA) as avgGPA from Student
      where sID in (
         select sID from Apply where major = 'CS')) -
     (select avg(GPA) as avgGPA from Student
      where sID not in (
         select sID from Apply where major = 'CS')) as d
from Student;
```

Amount by which avg. GPA of students applying to CS exceeds avg. of students not applying to CS

CS: 7
Total: 19
Non-cs:12
d: 12

The reason for the duplicates is that we compute this result once for each tuple in students.

```
select cName, count(*)
from Apply
group by cName;
```

Number of applications to each college

| cName | count(*) |
|---|---|
| Berkeley | 3 |
| Cornell | 6 |
| MIT | 4 |
| Stanford | 6 |

```
select state, sum(enrollment)
from College
group by state;
```

College enrollments by state

| | state | sum(enrollment) |
|---|---|---|
| ▶1 | CA | 51000 |
| 2 | MA | 10000 |
| 3 | NY | 21000 |

```
select cName, major, min(GPA), max(GPA)
from Student, Apply
where Student.sID = Apply.sID
group by cName, major;
```

min and max GPAs of applicants to each college & major

| cName | major | min(GPA) | max(GPA) |
|---|---|---|---|
| Berkeley | biology | 3.6 | 3.6 |
| Berkeley | CS | 3.7 | 3.9 |
| Cornell | bioengineering | 3.5 | 3.5 |
| Cornell | CS | 3.5 | 3.5 |
| Cornell | EE | 3.5 | 3.9 |
| Cornell | history | 2.9 | 2.9 |

```
select mx-mn
from (select cName, major, min(GPA) as mn, max(GPA) mx
from Student, Apply
where Student.sID = Apply.sID
group by cName, major) M;
```

max GPA - min GPA of applicants to each college & major

| | mx-mn |
|---|---|
| 1 | 0 |
| 2 | 0.19999999 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0.39999999 |
| 6 | 0 |

```
select Student.sID, count(distinct cName)
from Student, Apply
where Student.sID = Apply.sID
group by Student.sID;
```

Number of colleges applied to by each student

| sID | count(distinct cName) |
|---|---|
| 123 | 3 |
| 234 | 1 |
| 345 | 2 |
| 543 | 1 |
| 678 | 1 |

```
select Student.sID, count(distinct cName)
from Student, Apply
where Student.sID = Apply.sID
group by Student.sID
union
select sID, 0
from Student
where sID not in (select sID from Apply);
```

Include students who did not apply to any college

| | sID | coun cNam |
|---|---|---|
| 7 | 876 | 2 |
| 8 | 987 | 2 |
| 9 | 456 | 0 |
| 10 | 567 | 0 |
| 11 | 789 | 0 |
| 12 | 654 | 0 |

```
select cName
from Apply
group by cName
having count(*) < 5;
```

Colleges with fewer than 5 applications.

Another way:

```
select distinct cName
from Apply A1
where 5 > (select count(*) from Apply A2 where A2.cName = A1.cName);
```

| | cName |
|---|---|
| 1 | Berkeley |
| 2 | MIT |

```
select cName
from Apply
group by cName
having count(distinct sID) < 5;
```

Colleges with fewer than 5 applicants

| | cName |
|---|---|
| 1 | Berkeley |
| 2 | Cornell |
| 3 | MIT |

```
select major
from Student, Apply
where Student.sID = Apply.sID
group by major
having max(GPA) < (select avg(GPA) from Student);
```

Majors whose applicant's max GPA is below the avg.

| | major |
|---|---|
| 1 | bioengineering |
| 2 | psychology |

# Nulls

| | |
|---|---|
| `insert into Student values (432, 'Kevin', null, 1500);` | Insert value to “Student” table |
| `select sID, sName, GPA`<br>`from Student`<br>`where GPA > 3.5 or GPA <= 3.5 or GPA is null;` | Including null GPA |
| `select count(distinct GPA)`<br>`from Student;` | Not count the null value |
| `select distinct GPA`<br>`from Student;`<br>Do include the null value in the result |  |

# Data Modification Statements

| | |
|---|---|
| Insert Into Table<br>Values($A_1$,$A_2$,…,$A_n$) | Insert Into Table<br>Select-Statement |
| Delete From Table<br>Where Condition | Update Table<br>Set Attr = Expression<br>Where Condition |

| | |
|---|---|
| `insert into College values ('Carnegie Mellon', 'PA', 11500);` | |
| `insert into Apply`<br>`select sID, 'Carnegie Mellon', 'CS', null`<br>`from Student`<br>`where sID not in (select sID from Apply);` | Have all students who didn't apply anywhere apply to CS at CMU |
| `insert into Apply`<br>`select sID, 'Carnegie Mellon', 'EE', 'Y'`<br>`from Student`<br>`where sID in (select sID from Apply where major = 'EE' and decision='N');` | Admit to CMU EE all students who were turned down in EE elsewhere |
| `delete from Student`<br>`where sID in`<br>`(select sID`<br>`from Apply`<br>`group by sID`<br>`having count(distinct major) > 2);` | Delete all students who applied to more than two different majors |

| | |
|---|---|
| `delete from College`<br>`where cName not in (select cName from Apply where major = 'CS');` | Delete colleges with no CS applicants |
| `update Apply`<br>`set decision = 'Y', major = 'economics'`<br>`where cName = 'Carnegie Mellon'`<br>`and sID in (select sID from Student where GPA < 3.6);` | Accept applicants to CMU with GPA < 3.6 but turn them into economics majors |
| `update Apply`<br>`set major = 'CSE'`<br>`where major = 'EE'`<br>`and sID in`<br>`(select sID from Student`<br>`where GPA >= all`<br>`(select GPA from Student`<br>`where sID in (select sID from Apply where major = 'EE')));` | Turn the highest-GPA EE applicant into a CSE applicant |
| `update Student`<br>`set GPA = (select max(GPA) from Student),`<br>`sizeHS = (select min(sizeHS) from Student);` | Update the GPA to max and sizeHS to min |