# SQL course notes (Stanford Lagunita)

**select**

**from**

**where**

**order by** ;

order by GPA **desc**, enrollment

where major **like** ‘%bio%’ => for example biology

select **\*** => All attributes

select **GPA\*(sizeHS/1000.0)**

select GPA\*(sizeHS/1000.0) **as scaledGPA**

Set operators: **union (all), intersect, except**

From Student **S**, College **C**, …

where (not) **exists**

where GPA >= **all** (subquery)

where not enrollment <= **any** (subquery)

from Student **(inner) join** Apply

**on** Student.sID = Apply.sID

from Student **natural join** Apply

better practice =>

from Student **join** Apply **using (sID)**

from Student **left join** Apply using(sID)

equivalent to (but not recommended)

from Student **natural left join** Apply

from Student **right outer join** Apply using(sID)

from Student **full outer join** Apply using(sID)

**group by** column

**having** condition

**avg**(GPA), **min**(GPA) , **max**(GPA) , **count**(\*), **count**(distinct sID)

**General SQL note:** QUICKLY VERY COMPLEX EXPRESSIONS! MESSY!

group by cName

having count(\*) < 5

Unknown values are represented by **NULL**

GPA **is null**

Inserting data:

**insert into** Table

values(A1, A2, …, An)

**insert into** Table

Select-Statement

**delete from** Table

where Condition

**update** Table

**set** Attr = Expression

where Condition

insert into College values (‘Carnegie Mellon’, ‘PA’, 11500);

insert into Apply

select sID, ‘Carnegie Mellon’, ‘CS’, null

from Student

where sID not in (select sID from Apply);

update Apply

set decision = ‘Y’, major = ‘economics’

where cName = ‘Carnegie Mellon’

and sID in (select sID from Student where GPA < 3.6);

|  |  |
| --- | --- |
| **Operator** | **Description** |
| = | Equal |
| <> | Not equal. Note: In some versions of SQL this operator may be written as != |
| >, < | Greater than, Less than |
| >= | Greater than or equal |
| <= | Less than or equal |
| BETWEEN | Between an inclusive range |
| LIKE | Search for a pattern |
| IN | To specify multiple possible values for a column |

select **top** number|percent column\_name(s)

or

select cols

from table

limit number

select top 50 percent \*

from Customers;

**Like operators:** % and \_ are wildcards. % is an arbitrary number of symbols, and \_ is only one

where column\_name **between** value1 **and** value2