DLI Accelerated Data Science Teaching Kit

Lecture 2.6 - SQL Refresher

SQL Refresher: create table

```
>sqlite3 database.db


sqlite> create table student(id integer, name text);

sqlite> .schema

CREATE TABLE student(id integer, name text);
```

id	name

SQL Refresher: insert rows

```
insert into student values(111, "Smith");

insert into student values(222, "Johnson");

insert into student values(333, "Lee");

select * from student;
```

ssn	name
111	Smith
222	Johnson
333	Lee

SQL Refresher: create another table

```
create table takes
(id integer, course_id integer, grade integer);
```

```
sqlite>.schema
```

```
CREATE TABLE student(id integer, name text);
```

```
CREATE TABLE takes (id integer, course_id integer,
grade integer);
```

id	course_id	grade

SQL Refresher: joining 2 tables

More than one tables - **joins**

E.g., create roster for this course (6242)

id	name
111	Smith
222	Johnson
333	Lee

id	course_id	grade
111	6242	100
222	6242	90
222	4000	80

SQL Refresher: joining 2 tables + filtering

```
select name from student, takes
where
    student.id = takes.id and
    takes.course_id = 6242;
```

id	name
111	**Smith**
222	**Johnson**
333	Lee

id	course_id	grade
111	6242	100
222	6242	90
222	4000	80

Summarizing data:
Find **id** and **GPA** (a summary) for each student

```
select id, avg(grade)
from takes
group by id;
```

id	course_id	grade
111	6242	100
222	6242	90
222	4000	80

id	avg(grade)
111	100
222	85

Filtering Summarized Results

```
select id, avg(grade)
from takes
group by id
having avg(grade) > 90;
```

id	course_id	grade
111	6242	100
222	6242	90
222	4000	80

id	avg(grade)
111	100
~~222~~	~~85~~

SQL General Form

```
select a1, a2, ... an
from t1, t2, ... tm
where predicate
[order by ....]
[group by ...]
[having ...]
```

A lot more to learn! Oracle, MySQL, PostgreSQL, etc.
Highly recommend taking a **database course**.

DLI Accelerated Data Science Teaching Kit

Thank You