Structured Query Language (SQL)

Lecture by
Binu jasim

SQL

- A special-purpose programming language designed for managing data held in a relational database management system (RDBMS)
- ANSI/ISO standard. Yet different DBMSystems implement SQL (slightly) differently
- Based on Codd's Relational model (Flat model)
- MySQL, SQLite, PostgreSQL etc. implement SQL

Create a Database

- A DBMS can manage several databases
- e.g. A MySQL system may manage the *Academics* database as well as the *Employee* database
- **sqlite3 Academics.db** - SQLite
- **CREATE DATABASE Academics;** - MySQL
- (Note: Names of db and schema are case sensitive)

Select a Database

- **ATTACH DATABASE Academics;** - SQLite
- **USE Academics** - MySQL

- Note: The commands are not case sensitive

- Note: MySQL has a client server architecture while SQLite is implemented using file systems. e.g. Web browsers, Android etc.

Create a Table

```
CREATE TABLE Student(
SID INT PRIMARY KEY,
Dept TEXT,
Name TEXT);
```

- .tables command to view all tables
- .schema Student to view the schema of the table

INSERT Command

```
INSERT INTO Student
VALUES (123, “CSE”, “Alice”);
```

```
INSERT INTO Student(SID,
Name) VALUES (201, “Bob”);
```

SID	Dept	Name
123	CSE	Alice
201	NULL	Bob

Querying a Database

```
SELECT A1, A2 …
FROM R1, R2 …
WHERE Condition;
```

SELECT statement

```
SELECT SID, Name
FROM Student
WHERE Dept = “CSE”;
```

```
SELECT *
FROM Student
WHERE Dept = “CSE”;
```

```
SELECT *
FROM Student;
```

Student

rollNo	name	dept	CGPA
123	Alice	CSE	8.2
201	Bob	EEE	5.6
399	Cherry	CSE	8.2

Q1. What will the following query return?

```
SELECT *
FROM Student
WHERE dept="CSE" and CGPA>8;
```

Joins

Course

rollNo	cName	dept	marks
123	DBMS	CSE	48
123	OS	CSE	36
399	DBMS	CSE	25
201	DBMS	CSE	40
123	Statistics	Maths	39
201	Control	EEE	35.5

Join between 2 or more relations is a subset of the cross product b/w those relations

Joins

Q2. Find students from EEE in the DBMS course?

Note that none of the tables has all the information required completely - So Join

Joins

Q2. Find students from EEE in the DBMS course?

Note that none of the tables has all the information required completely - So Join

```
select rollNo, name
from Student, Course
where
Student.rollNo=Course.rollNo
and Student.dept="EEE"
and cName="DBMS";
```

Joins

- Error: Ambiguous column name *rollNo*
- *Student.rollNo* to disambiguate
- Returns: 201 Bob

Q3. What does the following query return?

```
select *
from Student, Course;
```

Q4. What does the following query return?

```
select *
from Student, Course
where
Student.rollNo=Course.rollNo
```

Q3. Find all students who have taken a course in the CSE department?

Q3. Find all students who have taken a course in the CSE department?

```
select Student.rollNo, name
from Student, Course
where
Student.rollNo=Course.rollNo
and Course.dept = “CSE”;
```

rollNo	name
123	Alice
123	Alice
399	Cherry
201	Bob

DISTINCT keyword

- select distinct

```
select distinct name
from Student, Course
where
Student.rollNo=Course.rollNo
and Course.dept = “CSE”;
```

```
name
----------
Alice
Cherry
Bob
```

More keywords

- select name as Student_Name from Student

```
Student_Name
------------
Alice
Bob
Cherry
```

- order by keyword

More filters

- like %%

```
select *
from Course
where dept like “%E%”;
```

- Not equal to <>
- Q4. Find all Non CS departments offering some courses?

```
select distinct dept
from Course
where dept <> “CSE”;
```

Table Variables

```
select distinct S.name
from Student S, Course C
where
S.rollNo=C.rollNo
and C.dept = “CSE”;
```

Self Joins

Q5. Find all pairs students who are from the same department?

Q5. Find all pairs students who are from the same department?

```
select S1.name, S2.name, S1.dept
from Student S1, Student S2
where
S1.dept = S2.dept;
```

Q5. Find all pairs students who are from the same department?

```
select S1.name, S2.name, S1.dept
from Student S1, Student S2
where
S1.dept = S2.dept and
S1.rollNo <> S2.rollNo;
```

Q5. Find all pairs students who are from the same department?

```
select S1.name, S2.name, S1.dept
from Student S1, Student S2
where
S1.dept = S2.dept and
S1.rollNo < S2.rollNo;
```

Set operations

```
select distinct dept as name
from Course
union
select name
from Student;
```

Q6. Find all students who have taken a course from both CSE and Maths?

Q6. Find all students who have taken a course from both CSE and Maths?

```
select rollNo
from Course where dept="CSE"
intersect
select rollNo
from Course where dept="Maths"
```

Q6. Find all students who have taken a course from both CSE and Maths using self Join?

```
select C1.rollNo
from Course C1, Course C2
where
C1.rollNo = C2.rollNo
and
C1.dept="CSE" and C2.dept="Maths";
```

Set difference

Q7. Find all students who have taken a course from CSE but not Maths?

```
select rollNo
from Course where dept="CSE"
except
select rollNo
from Course where dept="Maths"
```

Set difference

Q8. Can we find all students who have taken a course from CSE but not Maths only using joins?

```
select C1.rollNo
from Course C1, Course C2
where
C1.rollNo = C2.rollNo
and
C1.dept="CSE" and C2.dept<>"Maths";
```

It doesn't work!