

MySQL - RDBMS

Agenda

- Table Relations
- Joins
- Views
- Constraints
- ALTER

Table Relations

- One To One
- One To Many
- Many To One
- Many To Many

Emp Table

empno	name	deptno
1	Nitin	10
2	Nilesh	20
3	Amit	20
4	Yogesh	20

Meeting Table

meetno	Topic	Venue
101	May21 Batch Planning	Zoom
102	Audit	Skype

Emp_Meeting Table (Aux)

empno	meetno
1	101
2	101
4	101
1	102
2	102
3	102

Joins

Cross Join

- SELECT columns FROM table1 t1 CROSS JOIN table2 t2;

```
SELECT e.ename, d.dname FROM emps e
CROSS JOIN depts d;
```

```
SELECT e.ename, d.dname FROM depts d
CROSS JOIN emps e;

SELECT emps.ename, depts.dname FROM depts
CROSS JOIN emps;

SELECT ename, dname FROM depts
CROSS JOIN emps;
```

Inner Join

- SELECT columns FROM table1 t1 INNER JOIN table2 t2 ON condition;

```
SELECT e.ename, d.dname FROM emps e
CROSS JOIN depts d
WHERE e.deptno = d.deptno;
-- will work, but not good practice -- instead use inner join -- it is designed
for matching rows

SELECT e.ename, d.dname FROM emps e
INNER JOIN depts d ON e.deptno = d.deptno;

SELECT e.ename, d.dname FROM emps e
INNER JOIN depts d
WHERE e.deptno = d.deptno;
-- will work, but not good practice -- ON clause is defined to given condition on
join.

SELECT e.ename, d.dname FROM emps e
INNER JOIN depts d ON e.deptno = d.deptno
WHERE e.ename IN ('Amit', 'Nilesh');
-- where should be used to filter output of join.
```

Outer Join

- Common rows + Extra rows in the table(s)

Left Outer Join

- Common rows + Extra rows in the left table
- Left table <-- Table appeared before JOIN keyword.

```
SELECT e.ename, d.dname FROM depts d
LEFT OUTER JOIN emps e ON e.deptno = d.deptno;

SELECT e.ename, d.dname FROM depts d
LEFT JOIN emps e ON e.deptno = d.deptno;
```

Right Outer Join

- Common rows + Extra rows in the right table
- Right table <-- Table appeared after JOIN keyword.

```
SELECT e.ename, d.dname FROM depts d
RIGHT OUTER JOIN emps e ON e.deptno = d.deptno;

SELECT e.ename, d.dname FROM depts d
RIGHT JOIN emps e ON e.deptno = d.deptno;
```

Full Outer Join

- Common rows + Extra rows in the both table

```
SELECT e.ename, d.dname FROM depts d
FULL OUTER JOIN emps e ON e.deptno = d.deptno;
-- error in MySQL -- not supported.
```

Self Join

```
SELECT e.ename, m.ename FROM emps e
INNER JOIN emps m ON e.mgr = m.empno;

SELECT e.ename, m.ename FROM emps e
LEFT JOIN emps m ON e.mgr = m.empno;
```

USING keyword

- If column names on which join condition is given is same in both tables, then we can use USING keyword to short-hand the join condition.
- USING keyword can only be used for equi-join i.e. equality condition in join.

```
SELECT e.ename, d.dname FROM depts d
INNER JOIN emps e ON e.deptno = d.deptno;

SELECT e.ename, d.dname FROM depts d
INNER JOIN emps e USING(deptno);

SELECT e.ename, d.dname FROM depts d
LEFT JOIN emps e USING(deptno);

SELECT e.ename, d.dname FROM depts d
```

```
RIGHT JOIN emps e USING(deptno);
```

Union Operators

- (query1) UNION ALL (query2);
 - Common results are duplicated.
- (query1) UNION (query2);
 - Common results are not duplicated.

```
SELECT ename FROM emp WHERE job = 'CLERK';
```

```
SELECT ename FROM emp WHERE deptno = 30;
```

```
(SELECT ename FROM emp WHERE job = 'CLERK')
UNION ALL
(SELECT ename FROM emp WHERE deptno = 30);
```

```
(SELECT ename FROM emp WHERE job = 'CLERK')
UNION
(SELECT ename FROM emp WHERE deptno = 30);
```

```
(SELECT ename FROM emp)
UNION
(SELECT dname FROM dept);
```

```
(SELECT ename, sal, job FROM emp)
UNION
(SELECT dname FROM dept);
-- error: two select statements have different number of columns
```

```
(SELECT sal FROM emp)
UNION
(SELECT dname FROM dept);
```

```
(SELECT e.ename, d.dname FROM depts d
LEFT OUTER JOIN emps e ON e.deptno = d.deptno)
UNION ALL
(SELECT e.ename, d.dname FROM depts d
RIGHT OUTER JOIN emps e ON e.deptno = d.deptno);
```

```
(SELECT e.ename, d.dname FROM depts d
LEFT OUTER JOIN emps e ON e.deptno = d.deptno)
UNION
(SELECT e.ename, d.dname FROM depts d
RIGHT OUTER JOIN emps e ON e.deptno = d.deptno);
```

Advanced Joins

```
SELECT e.ename, d.dname FROM emps e, depts d
WHERE e.deptno = d.deptno;
```

```
-- print deptname and dist of all emps.
SELECT e.ename, d.dname FROM emps e
INNER JOIN depts d ON e.deptno = d.deptno;

SELECT e.ename, d.dname, a.dist FROM emps e
INNER JOIN depts d ON e.deptno = d.deptno
INNER JOIN addr a ON e.empno = a.empno;

SELECT e.ename, d.dname, a.dist FROM emps e
LEFT JOIN depts d ON e.deptno = d.deptno
INNER JOIN addr a ON e.empno = a.empno;
```

```
-- print emp name and meeting topic they are attending
SELECT e.ename, em.meetno FROM emp_meeting em
INNER JOIN emps e ON em.empno = e.empno;

SELECT e.ename, em.meetno, m.topic FROM emp_meeting em
INNER JOIN emps e ON em.empno = e.empno
INNER JOIN meeting m ON em.meetno = m.meetno;

-- emps are travelling for meeting from their hometown.
-- "also" print dist from where they are coming to meeting.
SELECT e.ename, em.meetno, m.topic, a.dist FROM emp_meeting em
INNER JOIN emps e ON em.empno = e.empno
INNER JOIN meeting m ON em.meetno = m.meetno
INNER JOIN addr a ON e.empno = a.empno;

-- emps are also representing their depts.
-- "also" print dnames in which they are working.
SELECT e.ename, em.meetno, m.topic, a.dist, d.dname FROM emp_meeting em
INNER JOIN emps e ON em.empno = e.empno
INNER JOIN meeting m ON em.meetno = m.meetno
INNER JOIN addr a ON e.empno = a.empno
LEFT JOIN depts d ON e.deptno = d.deptno;
```

```
-- (old classwork) print dname and num of emps working in each dept.
SELECT deptno, COUNT(empno) FROM emp
GROUP BY deptno;

SELECT d.dname, COUNT(e.empno) FROM emp e
INNER JOIN dept d ON e.deptno = d.deptno;
```

```
-- error: group fns need group by

SELECT d.dname, COUNT(e.empno) FROM emp e
INNER JOIN dept d ON e.deptno = d.deptno
GROUP BY e.deptno;

-- error: columns to be selected must be in group by

SELECT d.dname, COUNT(e.empno) FROM emp e
INNER JOIN dept d ON e.deptno = d.deptno
GROUP BY d.dname;

SELECT d.dname, COUNT(e.empno) FROM emp e
RIGHT JOIN dept d ON e.deptno = d.deptno
GROUP BY d.dname;

-- print dname and empcount in desc order of emp count.
SELECT d.dname, COUNT(e.empno) FROM emp e
RIGHT JOIN dept d ON e.deptno = d.deptno
GROUP BY d.dname
ORDER BY COUNT(e.empno) DESC;

-- print dname and empcount of dept having highest emps.
SELECT d.dname, COUNT(e.empno) FROM emp e
RIGHT JOIN dept d ON e.deptno = d.deptno
GROUP BY d.dname
ORDER BY COUNT(e.empno) DESC
LIMIT 1;

-- print dname and clerk count of dept having highest clerk.
SELECT d.dname, COUNT(e.empno) FROM emp e
RIGHT JOIN dept d ON e.deptno = d.deptno
WHERE job = 'CLERK'
GROUP BY d.dname
ORDER BY COUNT(e.empno) DESC
LIMIT 1;
```

Joins Syntax

```
SELECT columns FROM table1
XXX JOIN table2 ON condition1
XXX JOIN table3 ON condition2
WHERE condition
GROUP BY column
HAVING condition
ORDER BY column ASC/DESC
LIMIT m,n;
```

SQL

- SQL is Declarative Query Language

- Programmer decides what to do, but doesn't define how to do.
- Internal implementation depends on RDBMS (may differ in different RDBMS).

Sunbeam Infotech