SUNBEAM

MySQL RDBMS

Trainer: Mr. Nilesh Ghule

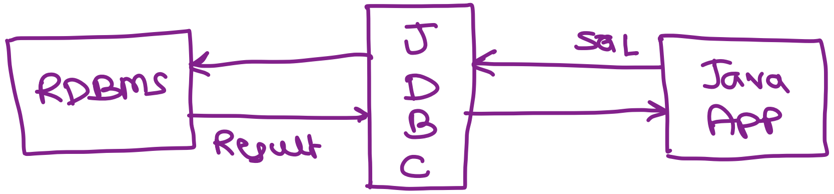
Row locking

- When an user update or delete a row (within a transaction), that row is locked and becomes read-only for other users.
- The other users see old row values, until transaction is committed by first user.
- If other users try to modify or delete such locked row, their transaction processing is blocked until row is unlocked. → or timeout.
- Other users can INSERT into that table. Also they can UPDATE or DELETE other rows.
- The locks are automatically released when COMMIT/ROLLBACK is done by the user. → first.
- This whole process is done automatically in MySQL. It is called as "OPTIMISTIC LOCKING".

Row locking

- Manually locking the row in advanced before issuing UPDATE or DELETE is known as "PESSIMISTIC LOCKING".
- This is done by appending FOR UPDATE to the SELECT query.
- It will lock all selected rows, until transaction is committed or rollbacked. or timeout
- If these rows are already locked by another users, the SELECT operation is blocked until rows lock is released.

- By default MySQL does table locking. Row locking is possible only when table is indexed on the column.



START TRANSACTION; → con.setAutoCommit(false);

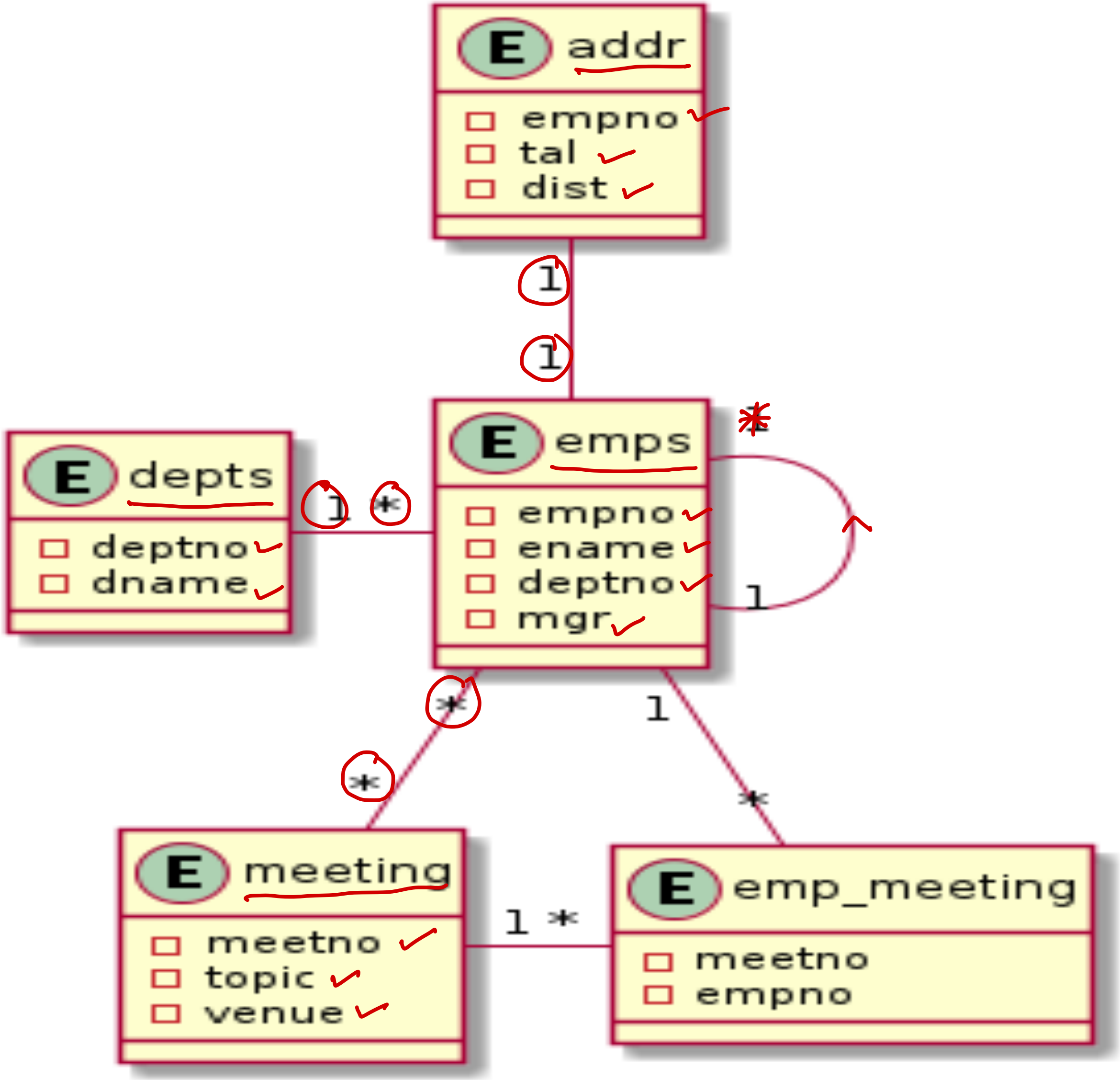
COMMIT; → con.commit();

OR OR

ROLLBACK; → con.rollback();

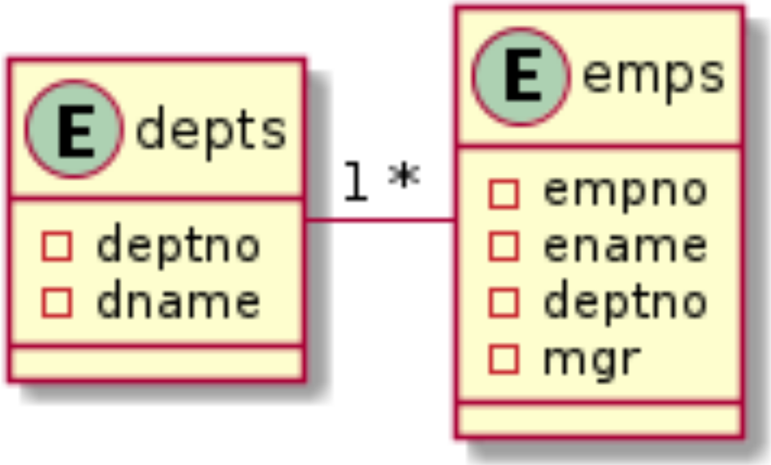
Entity Relations

- To avoid redundancy of the data, data should be organized into multiple tables so that tables are related to each other.
- The relations can be one of the following
  - One to One → emps → addr
  - One to Many → depts → emps
  - Many to One → emps → depts
  - Many to Many → emps ↔ meeting
- Entity relations is outcome of Normalization process.



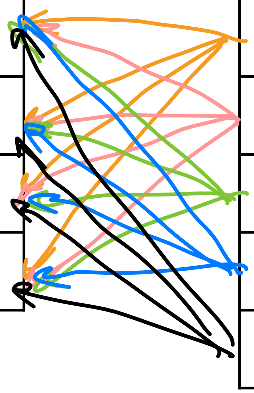
SQL Joins

- Join statements are used to SELECT data from multiple tables using single query.
- Typical RDBMS supports following types of joins:
  - Cross Join ✓
  - Inner Join ✓
  - Left Outer Join ✓
  - Right Outer Join ✓
  - Full Outer Join ✓
  - Self join ✓



Cross Join

deptno	dname
10	DEV
20	QA
30	OPS
40	ACC



empno	ename	deptno
1	Amit	10
2	Rahul	10
3	Nilesh	20
4	Nitin	50
5	Sarang	50

emps[] depts[]

print all possible depts in which each emp can be.

```
driving → foreach (e) in emps   ← 5 times
{
    driven → foreach d in depts   ← 4 times
    {
        print (e.ename, d.dname);   ← total 20 times
    }
}
```

column names

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

emps: table1

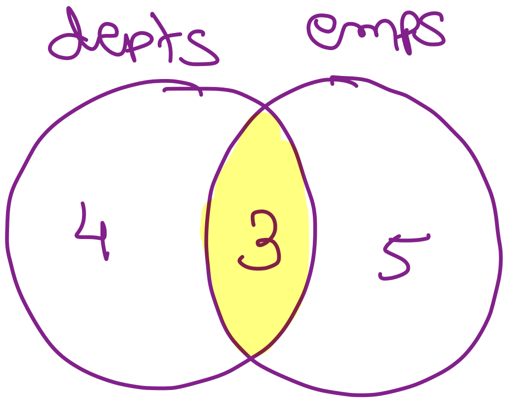
depts: table2

- Compares each row of Table1 with every row of Table2.
- Yields all possible combinations of Table1 and Table2.
- In MySQL, The larger table is referred as "Driving Table", while smaller table is referred as "Driven Table". Each row of Driving table is combined with every row of Driven table.
- Cross join is the fastest join, because there is no condition check involved.

Inner Join

deptno	dname
10	DEV
20	QA
30	OPS
40	ACC

empno	ename	deptno
1	Amit	10
2	Rahul	10
3	Nilesh	20
4	Nitin	50
5	Sarang	50



```
for each (e) in emps
{
    for each d in depts
    {
        if (e.deptno == d.deptno)
            print (e.ename, d.dname);
    }
}
```

columns

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

==

!=

- The inner JOIN is used to return rows from both tables that satisfy the join condition.
- Non-matching rows from both tables are skipped.
- If join condition contains equality check, it is referred as equi-join; otherwise it is non-equi-join.

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>