Agenda

```
-Joins Assignment discussion
Security
Transactions
ROW locking
```

Joins Assignment Questions

```
-- 4. Write a query that calculates the amount of the salesperson’s commission on
each  order by a customer with a rating above 100
salespeople -> 1002,1003,1007

-- 5. Display the Supplier name, Supplying Parts to a Job in the same City.
    --display sname,job,city of s,city of j
    s inner join sp inner join j

-- 9. Display all the Job names and City, which has bought more than 500 Parts.
-- SELECT `j#`,qty FROM sp WHERE qty>500;

-- 12. Display all the Parts which have more Weight than any Red parts.

-- 15. Display the name of the Supplier who has sold the maximum Quantity (in
onesale).
    s inner join sp a cross join sp b, group by, having
```

Security

Root

```
cmd> mysql -u root -p
mysql>PROMPT \u>
root> SELECT user,host FROM mysql.user;
root>CREATE USER 'magr' IDENTIFIED BY 'mgr';
root>CREATE USER 'teamlead'@'localhost' IDENTIFIED BY 'teamlead';
root>CREATE USER 'dev1'@'localhost' IDENTIFIED BY 'dev1';
root>CREATE USER 'dev2'@'localhost' IDENTIFIED BY 'dev2';
root> SELECT user,host FROM mysql.user;
root>SHOW GRANTS FOR mgr
root>SHOW GRANTS FOR teamlead@localhost
root> SHOW GRANTS
root>GRANT ALL PRIVILEGES ON classwork_db.* TO mgr;
root>GRANT ALL PRIVILEGES ON classwork_db.* TO mgr WITH GRANT OPTION;
root>...

-- If you want to delete the user then
```

```
DROP USER 'dev1'@'localhost';
DROP USER 'dev2'@'localhost';
```

Mgr

```
cmd> mysql -u mgr -p
mysql>PROMPT \u>
mgr>SHOW GRANTS;
mgr>SHOW DATABASES;
mgr>USE classwork_db;
mgr>SHOW TABLES;
mgr>GRANT SELECT,INSERT,UPDATE ON classwork_db.emps TO 'teamlead'@'localhost';
mgr>GRANT SELECT,INSERT,UPDATE ON classwork_db.depts TO 'teamlead'@'localhost';
mgr>GRANT SELECT ON classwork_db.emps TO 'dev1'@'localhost';
mgr>GRANT SELECT ON classwork_db.depts TO 'dev2'@'localhost';
mgr>....
REVOKE SELECT,INSERT,UPDATE ON classwork_db.emps FROM 'teamlead'@'localhost';
REVOKE SELECT,INSERT,UPDATE ON classwork_db.depts FROM 'teamlead'@'localhost';
REVOKE SELECT ON classwork_db.emps FROM 'dev1'@'localhost';
REVOKE SELECT ON classwork_db.depts FROM 'dev2'@'localhost';
```

teamlead

```
cmd> mysql -u teamlead -p
mysql>PROMPT \u>
teamlead>SHOW GRANTS;
teamlead>USE classwork_db;
teamlead>...
```

dev1

```
cmd> mysql -u dev1 -p
mysql>PROMPT \u>
dev1>SHOW GRANTS;
dev1>SHOW DATABASES;
dev1>USE classwork_db;
dev1>....
SHOW TABLES;
```

dev2

```
cmd> mysql -u dev2 -p
mysql>PROMPT \u>
dev2>SHOW GRANTS;
dev2>SHOW DATABASES;
dev2>USE classwork_db;
dev2>....
SHOW TABLES;
```

Transaction

```
CREATE TABLE accounts (
    id INT PRIMARY KEY,
    type VARCHAR(12),
    balance DECIMAL(9,2)
);

INSERT INTO accounts VALUES(1,"SAVINGS",10000);
INSERT INTO accounts VALUES(2,"CURRENT",20000);
INSERT INTO accounts VALUES(3,"SAVINGS",30000);
INSERT INTO accounts VALUES(4,"SAVINGS",40000);
INSERT INTO accounts VALUES(5,"CURRENT",50000);

START TRANSACTION;
UPDATE accounts SET balance = 8000 WHERE id = 1;
SELECT * FROM accounts;
COMMIT;
SELECT * FROM accounts;


START TRANSACTION;
UPDATE accounts SET balance =9000 WHERE id = 1;
SELECT * FROM accounts;
ROLLBACK;
SELECT * FROM accounts;

START TRANSACTION;
UPDATE accounts SET balance =5000 WHERE id = 3;
UPDATE accounts SET balance =15000 WHERE id = 4;
DELETE FROM accounts WHERE id=2;
SELECT * FROM accounts;
ROLLBACK;
SELECT * FROM accounts;

START TRANSACTION;
UPDATE accounts SET balance =25000 WHERE id = 3;
UPDATE accounts SET balance =35000 WHERE id = 4;
SAVEPOINT sp1;
UPDATE accounts SET balance =65000 WHERE id = 5;
SAVEPOINT sp2;
```

```
DELETE FROM accounts;
ROLLBACK TO sp2;
DELETE FROM accounts WHERE id = 1;
ROLLBACK TO sp1;
COMMIT;
```

ACID Properties

```
A - Atomicity
C - Consistant
I - Isolation
D - Durability
```

ROW LOCKING

```
- If your table have a primary key then your only specific row gets locked in a
transaction
```

```
root>START TRANSACTION;
root>UPDATE accounts SET balance = 15000 WHERE id = 1;

mgr>UPDATE accounts SET balance = 20000 WHERE id = 1;--error
--Lock wait timeout exceeded; try restarting transaction

root>COMMIT;

mgr>UPDATE accounts SET balance = 20000 WHERE id = 1;
```

Pessismetic Locking

```
- If you wish to lock the row manually then you can use this type of locking
```

```
root>START TRANSACTION;
root>SELECT * FROM accounts WHERE id = 2 FOR UPDATE;

mgr>UPDATE accounts SET balance = 40000 WHERE id = 2;--error
--Lock wait timeout exceeded; try restarting transaction

mgr>UPDATE accounts SET balance = 20000 WHERE id = 1;

root>ROLLBACK;
```

Table Locking

```
- If your table does not have a primary key then your entire table gets locked in
a transaction
```

```
root>START TRANSACTION;
root>UPDATE emps SET deptno = 60 WHERE empno=5;

mgr>UPDATE emps SET deptno=70 WHERE empno = 5;--error
--Lock wait timeout exceeded; try restarting transaction

mgr>UPDATE emps SET deptno=70 WHERE empno = 4; --error
--Lock wait timeout exceeded; try restarting transaction
```